KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING NUMERIEKE ANALYSE EN
TOEGEPASTE WISKUNDE
Celestijnenlaan 200A – B-3001 Heverlee

POWELL-SABIN SPLINES AND MULTIRESOLUTION TECHNIQUES

Promotor:
Prof. Dr. A. Bultheel

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Evelyne VANRAES

April 2004

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING NUMERIEKE ANALYSE EN
TOEGEPASTE WISKUNDE
Celestijnenlaan 200A – B-3001 Heverlee

POWELL-SABIN SPLINES AND MULTIRESOLUTION TECHNIQUES

Jury:
Prof. Dr. ir. H. Van Brussel, voorzitter
Prof. Dr. A. Bultheel, promotor
Prof. Dr. ir. P. Dierckx
Prof. Dr. ir. S. Vandewalle
Dr. ir. M. Jansen
(Technische Universiteit Eindhoven)
Prof. Dr. P. Sablonnière
(Institut National des Sciences Appliquées, Rennes)
Dr. ir. W. Sweldens
(Bell Laboratories)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Evelyne VANRAES

U.D.C. 681.3*F21

April 2004

D/2004/7515/12

ISBN 90-5682-474-0

Powell-Sabin Splines and Multiresolution Techniques

Evelyne Vanraes
Katholieke Universiteit Leuven
Department of Computer Science

Abstract

Powell-Sabin splines are piecewise quadratic polynomials with global C^1 continuity. In contrast to the widely used tensor product B-splines and NURBS, they are defined on triangular patches which has certain advantages. In this dissertation we place Powell-Sabin splines in a multiresolution context. Multiresolution techniques are concerned with the generation, representation and manipulation of geometric objects at different levels of detail. The main ingredient here is a subdivision scheme to compute a representation of a surface on a refined triangulation. Such a refined triangulation has more vertices and more and smaller triangles than the original triangulation. The new basis functions after subdivision have smaller support and give the designer more local control. It makes it possible to represent a surface on different levels of detail. A standard dyadic scheme cannot be used for non uniform triangulations where the vertices are not regularly spaced. Instead we propose a $\sqrt{3}$ scheme. Applying this scheme twice results in a triadic scheme. When going back from a fine resolution level to a coarser resolution level, fine detail is lost. We develop a wavelet decomposition algorithm that transforms a fine scale surface into a coarse scale approximation and a detail part that lives in a complement space. We use the lifting paradigm with the triadic subdivision scheme as prediction step. Because the domain triangles need not to be uniform, we pay special attention to stability in the design of the update step. The multiresolution techniques for splines can only be applied in the functional case or when the object is defined as a parametric surface. We propose an extension of a dyadic subdivision algorithm voor uniform Powell-Sabin splines to arbitrary topologies. In this setting a surface is defined procedurally as the limit of a refinement process.

Keywords: Powell-Sabin split, B-splines, subdivision, $\sqrt{3}$ subdivision, wavelets, lifting, stability, generalised subdivision, arbitrary topology.

AMS 2000 Mathematics Subject Classification: 65D17, 65D18, 68U05, 68U07.

Preface

I first came in contact with multiresolution techniques during my Master's Thesis about second generation wavelets with Maarten Jansen as my daily advisor. Thanks to the many lively discussions with Maarten, I decided do start a PhD and joined the Numerical Approximation and Linear Algebra Group of the Department of Computer Science at the KULeuven under the direction of Prof. Adhemar Bultheel. I want to thank Prof. Bultheel for becoming the advisor of my PhD thesis and for carefully proofreading all my manuscripts.

The idea of placing Powell-Sabin splines in a multiresolution context is inspired by my former colleague Joris Windmolders who developed a subdivision scheme for the uniform case under supervision of Prof. Paul Dierckx. Splines on triangulations were new to me and Joris spent many hours answering my questions. Also the help of Prof. Dierckx was valuable to gain insight in the theory of splines.

I also would like to thank Prof. Stefan Vandewalle for being a member of the reading committee together with Prof. Bultheel and Prof. Dierckx. They did an excellent job reading and commenting on an early draft of this text. Prof. Paul Sablonnière, Maarten Jansen and Wim Sweldens kindly accepted to join them as a member of the jury.

I enjoyed working together with Joris Windmolders and Jan Maes with whom I had fruitful collaborations. My colleagues on the fourth floor created the stimulating and pleasant environment in which this work could grow. I also owe many thanks to my friends and family for their support during the last years. Last but not least, I am most grateful to Roel. His genuine concern and his endless patience have been a great help to me.

Evelyne Vanraes
April 2004

Nederlandse Samenvatting

Powell-Sabin Splines en Multiresolutietechnieken

Inhoudsopgave

1 Inleiding

1.1 Modelleren van curves en oppervlakken

Het voorstellen van complexe geometrische vormen op de computer is belangrijk in verschillende toepassingen. Een voorstellingswijze die nog steeds courant gebruikt wordt is de Bernstein-Bézier veeltermvoorstelling. Ze verschilt van andere technieken door het gebruik van controleveelhoeken. In plaats van een curve te definiëren aan de hand van punten die erop liggen, worden punten gebruikt die in de buurt liggen. Als de controleveelhoek verandert, verandert de curve op een intuïtieve manier mee.

Een ‘spline’ was een houten lat die door middel van gewichten gebogen werd om een vloeiende lijn te tekenen door een stel opgegeven punten. De potentiële energie van de lat is minimaal en omdat de kromming niet discontinu kan veranderen zijn de tweede afgeleiden continu. De term spline werd later gebruikt om curves aan te duiden die bestaan uit kubische segmenten die twee keer afleidbaar zijn omdat deze een functionaal minimaliseren gelijkaardig aan de fysische eigenschappen van de houten spline. Nu wordt de term splinecurve in het algemeen gebruikt om stuksgewijze veeltermen met een bepaalde graad van continuïteit aan te duiden.

B-splines of ‘basic’ splines zijn genormaliseerde basisfuncties voor een splineruimte. Initieel werden ze gedefinieerd aan de hand van gedeelde differenties, maar later werd een recursieve definitie ontdekt die het werken met B-splines aanzienlijk vereenvoudigde. Het bleek dat de Bézier curves een deelverzameling zijn van de B-spline curves.

Om oppervlakken voor te stellen worden meestal tensorprodukt-splines gebruikt. Een nadeel van deze voorstelling is echter dat ze beperkt is tot rechthoekige roosters. Een alternatief is het oppervlak uit te drukken in termen van barycentrische coördinaten ten opzichte van een driehoekig domein. We spreken dan van Bézier driehoeken.

Oppervlakken voorstellen over een domein dat bestaat uit meerdere driehoeken is heel wat moeilijker omdat de dimensie van een dergelijke splineruimte niet alleen afhangt van het aantal punten en het aantal driehoeken, maar ook van de topologie en de geometrie. Hoewel er voor verschillende specifieke gevallen bovenen ondergrenzen en zelfs exacte resultaten gevonden zijn, is het in het geval van veeltermen van lage graad nog steeds onmogelijk de dimensie van de splineruimte te bepalen.

Om dat op te lossen worden soms splits-procedures gebruikt die elke driehoek in de driehoeksverdeling opsplitsen in meerdere kleinere driehoeken. Om het oppervlak over een oorspronkelijke driehoek te beschrijven wordt dan meer dan één veelterm gebruikt. Voor kwadratische splines bestaat er het Powell-Sabin element waarin elke driehoek wordt gesplitst in zes nieuwe driehoeken met een gemeenschappelijk punt in het midden. De dimensie van de ruimte van C^1 continue kwadratische veeltermen over dit soort driehoeken is exact bepaald. P. Dierckx et al. construeerden een B-spline basis voor deze ruimte die een convexe eenheidspartitie vormt.

1.2 Multiresolutie

Subdivisie is een iteratieve techniek om zachtverlopende curves en oppervlakken te construeren door steeds nieuwe punten toe te voegen aan een initiële verzameling punten. De eerste subdivisieschema's waren methodes om knooppunten toe te voegen aan splinecurves. De nieuwe verzameling knooppunten stelt dan exact dezelfde splinefunctie voor. Later werden subdivisieschema's bestudeerd buiten de context van splines. In het algemeen is er geen mathematische uitdrukking voorhanden voor de limietfunctie en is de curve proceduraal gedefinieerd.

In het geval van oppervlakken werden ook subdivisieregels ontwikkeld voor onregelmatige topologieën, zoals bijvoorbeeld in het geval van driehoeksroosters voor punten met een valentie verschillend van zes. In deze punten, die buitengewone punten genoemd worden, is de convergentie en de continuïteit niet voor de hand liggend. Het bewijs daarvan gebeurt aan de hand van eigenwaardenanalyse.

Een functie kan gezien worden als een zachtverlopend gedeelte dat de algemene vorm bepaalt met daarbovenop details of fluctuaties. Het splitsen van een functie in die twee gedeeltes wordt een wavelettransformatie genoemd. Het zachtverlopend gedeelte kan iteratief terug opgesplitst worden in een grove benaderende functie en details. De resolutie bepaalt hoeveel details worden opgenomen in de benadering. Hoe meer details worden toegevoegd, hoe beter de benadering van de oorspronkelijke functie zal zijn.

Subdivisie en wavelets zijn met elkaar verbonden doordat elk subdivisieschema een sequentie van geneste ruimtes bepaalt die de grove functies bevat. Dit inzicht is bijzonder nuttig in het ontwerp van tweedegeneratiewavelets op onregelmatige roosters, waar de klassieke technieken voor regelmatige roosters niet toepasbaar zijn.

1.3 Opbouw van het proefschrift

In Hoofdstuk 2 bespreken we enkele basisbegrippen in verband met Bernstein-Bézier veeltermen en Powell-Sabin splines. We vestigen ook de aandacht op de constructie van de B-spline basis en de geometrische interpretatie met controle-driehoeken. In Hoofdstuk 3 bewijzen we de stabiliteit van de basis, zowel voor de oneindig norm, als in het algemeen voor de p-norm. Hierbij blijkt het belangrijk te zijn dat de oppervlakte van de PS-driehoeken begrensd is. We observeren ook dat hoe groter de PS-driehoeken zijn hoe meer de basisfuncties lineair afhankelijk zijn en we geven een praktisch algoritme om snel een kleine PS-driehoek te vinden.

Hoofdstuk 4 is een overzicht van de multiresolutietechnieken die aan bod komen in deze thesis. We definiëren subdivisie als een iteratieve procedure en bouwen daarop voort om de begrippen multiresolutie-analyse en wavelets te introduceren. In Hoofdstuk 5 bekijken we subdivisie voor Powell-Sabin splines. Een dyadische splitsing kan niet gebruikt worden in het onregelmatige geval. Daarom stellen we een $\sqrt{3}$ schema voor dat als het twee keer wordt toegepast leidt tot een triadische splitsing. Hoofdstuk 6 behandelt splinewavelets met een multiresolutie-analyse gebaseerd op het triadische subdivisieschema. We maken gebruik van het lifting schema, en bij het ontwerp van de bijsturingsstap besteden we aandacht aan de stabiliteit.

Een nadeel van spline-oppervlakken is dat ze moeten kunnen geparametriseerd worden over een vlak. Daarom onderzoeken we in Hoofdstuk 7 oppervlakken met willekeurige topologieën en ontwerpen een veralgemeend subdivisieschema dat gebaseerd is op uniforme dyadische Powell-Sabin splinesubdivisie. Het limietoppervlak interpoleert niet alleen bepaalde punten maar ook hun opgegeven normalen. We bewijzen de convergentie van dit nieuwe schema aan de hand van de eigenwaarden van de subdivisiematrix en de eigenschappen van een karakteristieke functie.

Tenslotte geven we een overzicht van de nieuwe bijdragen en enkele suggesties voor verder onderzoek in Hoofdstuk 8.

2 Powell-Sabin splines

In dit hoofdstuk geven we een overzicht van de fundamentele theorie rond Powell-Sabin splines. Dit zijn stuksgewijs C^1 continue kwadratische veeltermen op Powell-Sabin driehoeksverdelingen. We gebruiken ze om oppervlakken en geometrische objecten voor te stellen. We geven ook een samenvatting van de Bernstein-Bézier voorstelling omdat de Powell-Sabin voorstelling daarop verder bouwt.

2.1 Bernstein-Bézier veeltermen

Zij $\mathcal{T}$ een niet ontaarde driehoek met hoekpunten $T_i(x_i, y_i), i = 1, 2, 3$. Elk punt $P(x, y)$ kan voorgesteld worden door middel van zijn barycentrische coördinaten $\tau = (\tau_1, \tau_2, \tau_3)$ ten opzichte van $\mathcal{T}$: $P = \sum_{i=1}^{3} \tau_i T_i$, met τ_i, $i = 1, 2, 3$ de unieke oplossing van het stelsel

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

De Bernstein veeltermen van graad m over een driehoek zijn gedefinieerd als

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

waarbij $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ een multi-index is, en $|\lambda| = \lambda_1 + \lambda_2 + \lambda_3$, $\lambda_i \in \{0, 1, \ldots, m\}$. Deze veeltermen vormen een convexe eenheidspartitie, dat wil zeggen, ze zijn niet negatief en ze sommeren tot één.

Zij Π_m de ruimte van veeltermen met graad $\leq m$ in twee veranderlijken. Een Bernstein-Bézier veelterm $b(\tau)$ is de unieke voorstelling van een veelterm $p(x, y), (x, y) \in \mathbb{R}^2$, in Π_m, in termen van de Bernstein veeltermen van graad m op $\mathcal{T}$

$$p(x, y) := b(\tau) = \sum_{|\lambda|=m} b_\lambda B_\lambda^m(\tau).$$

De coëfficiënten b_λ worden Bézier ordinaten genoemd en worden geassocieerd met bepaalde punten ξ_λ in $\mathcal{T}$ met barycentrische coördinaten $\frac{\lambda}{m}$.

Een Bernstein-Bézier oppervlak is de grafe van een Bernstein-Bézier veelterm

$$\boldsymbol{b}(\tau) := \{(x, y, z) \mid (x, y) = \tau \in \mathcal{T},\ z = b(\tau)\}.$$

De punten $\boldsymbol{b}_\lambda(\xi_\lambda, b_\lambda)$ zijn de controlepunten. Het Bernstein-Bézier oppervlak ligt binnen de convex omhullende van haar controlepunten. Hetzelfde geldt voor de uitbreiding naar parametrische oppervlakken

$$\boldsymbol{b}(\tau) = \sum_{|\lambda|=m} \boldsymbol{b}_\lambda B_\lambda^m(\tau),$$

met controlepunten $\boldsymbol{b}_\lambda = (b_\lambda^x, b_\lambda^y, b_\lambda^z)$.

Om een Bernstein-Bézier veelterm te evalueren gebruiken we het de Casteljau algoritme. De waarde in een punt met barycentrische coördinaten τ is

$$b(\tau) = b^m_{000}(\tau)$$

met

$$b^0_{\lambda_1,\lambda_2,\lambda_3}(\tau) = b_{\lambda_1,\lambda_2,\lambda_3},$$
$$b^r_{\lambda_1,\lambda_2,\lambda_3}(\tau) = \tau_1 b^{r-1}_{\lambda_1+1,\lambda_2,\lambda_3}(\tau) + \tau_2 b^{r-1}_{\lambda_1,\lambda_2+1,\lambda_3}(\tau) + \tau_3 b^{r-1}_{\lambda_1,\lambda_2,\lambda_3+1}(\tau),$$
$$r = 1,\ldots,m, \quad |\lambda| = m - r.$$

Dit algoritme is ook nuttig voor het berekenen van afgeleiden, raakvectoren en normalen.

Continuïteitsvoorwaarden tussen verschillende Bézier driehoeken kunnen dankzij het de Casteljau algoritme uitgedrukt worden in termen van de respectievelijke Bézier ordinaten. Beschouw bijvoorbeeld de Bernstein-Bézier veeltermen $b(\tau)$ op de driehoek $\mathcal{T}(T_1,T_2,T_3)$ en $\hat{b}(\tau)$ op de driehoek $\mathcal{T}(\hat{T}_1,T_2,T_3)$ waarbij het punt $\hat{T}_1$ barycentrische coördinaten σ heeft ten opzichte van driehoek $\mathcal{T}$. Dan is een nodige en voldoende voorwaarde voor C^1 continuïteit langs de gemeenschappelijke zijde

$$C^0 : \ \hat{b}_{0,\lambda_2,\lambda_3} = b_{0,\lambda_2,\lambda_3},$$
$$C^1 : \ \hat{b}_{1,\lambda_2,\lambda_3} = \sigma_1 b_{1,\lambda_2,\lambda_3} + \sigma_2 b_{0,\lambda_2+1,\lambda_3} + \sigma_3 b_{0,\lambda_2,\lambda_3+1}.$$

2.2 Powell-Sabin splines

De continuïteitsvoorwaarden tussen aangrenzende Bézier driehoeken zijn moeilijk te implementeren, en bovendien is het niet mogelijk voorspelbare lokale veranderingen te doen aan het oppervlak. Daarom beschouwen we op driehoeksverdelingen liever veeltermen met globale continuïteitseigenschappen. Omdat in het algemeen de dimensie van een splineruimte over een driehoeksverdeling niet exact te bepalen is, beperken we ons tot kwadratische veeltermen met C^1 continuïteit over Powell-Sabin verfijningen.

Zij Δ een conforme driehoeksverdeling van $\Omega \subset \mathbb{R}^2$ met punten $V_k(x_k,y_k)$, $k = 1,\ldots,n$ en driehoeken $\mathcal{T}_j, j = 1,\ldots,t$. Een Powell-Sabin verfijning Δ^* van Δ splitst elke driehoek in zes kleinere driehoeken als volgt:

1. Kies een punt Z_j in elke driehoek $\mathcal{T}_j$, zodat als twee driehoeken $\mathcal{T}_i$ en $\mathcal{T}_j$ een gemeenschappelijke zijde hebben, de lijn die de punten Z_i en Z_j verbindt, deze zijde snijdt in een punt R_{ij} tussen de hoekpunten.

2. Verbind elk punt Z_j met de hoekpunten van $\mathcal{T}_j$.
3. Voor elke zijde van $\mathcal{T}_j$
   - die tot de rand $\delta\Omega$ behoort, verbind Z_j met een willekeurig punt tussen de hoekpunten.
   - die gemeenschappelijk is met een driehoek $\mathcal{T}_i$, verbind Z_j met R_{ij}.

De ruimte van stuksgewijs kwadratische veeltermen met C^1 continuïteit, de Powell-Sabin splines, wordt gedefinieerd als

$$S_2^1(\Delta^*) := \{s \in C^1(\Omega) : s|_{\mathcal{T}^*} \in \Pi_2,\ \mathcal{T}^* \in \Delta^*\}.$$

Elk van de $6t$ driehoeken van de Powell-Sabin verfijning wordt het domein van een kwadratische Bernstein-Bézier veelterm. Powell en Sabin hebben aangetoond dat er een oplossing $s(x,y) \in S_2^1(\Delta^*)$ bestaat voor het interpolatieprobleem

$$s(V_k) = f_k, \quad \frac{\partial s}{\partial x}(V_k) = f_{x,k}, \quad \frac{\partial s}{\partial y}(V_k) = f_{y,k}, \quad k = 1, \ldots, n.$$

De dimensie van $S_2^1(\Delta^*)$ is dus $3n$.

Beschouw drie lineair onafhankelijke drietallen $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$, $j = 1, 2, 3$ voor elk punt V_i in de driehoeksverdeling. De basisfunctie $B_{ij}(x,y)$ is de oplossing van het interpolatieprobleem met alle $(f_k, f_{x,k}, f_{y,k}) = (0,0,0)$ behalve $(f_i, f_{x,i}, f_{y,i}) = (\alpha_{ij}, \beta_{ij}, \gamma_{ij})$. Deze basisfuncties zijn lokaal, het domein van $B_{ij}(x,y)$ bestaat slechts uit de driehoeken waarvan de vertex V_i deel uitmaakt. Elke $s(x,y) \in S_2^1(\Delta^*)$ heeft een unieke voorstelling

$$s(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{3} c_{ij} B_{ij}(x,y) = \mathbf{B}\ \mathbf{c}.$$

In deze thesis kiezen we de drietallen $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ zodanig dat de basisfuncties een convexe eenheidspartitie vormen, dat wil zeggen dat ze niet negatief zijn en opsommen tot één. Daartoe zoeken we voor elk hoekpunt een driehoek $t_i(Q_{i1}, Q_{i2}, Q_{i3})$ die een gegeven stel punten, die we PS-punten noemen, bevat. Er bestaat een één-één relatie tussen de drietallen $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ en de hoekpunten $Q_{ij}(X_{ij}, Y_{ij})$ van de PS-driehoek. Wanneer we als PS-driehoek de kleinste mogelijke driehoek kiezen spreken we van een optimale voorstelling. Dat de PS-driehoek de betrokken PS-punten bevat is een nodige en voldoende voorwaarde voor de convexiteit van de basisfuncties.

Een Powell-Sabin spline-oppervlak is de grafe van een Powell-Sabin spline-functie

$$\boldsymbol{s}(x,y) = \begin{cases} x &= x \\ y &= y, \\ z &= s(x,y) \end{cases} \qquad (x,y) \in \Omega$$

met als controlepunten $\boldsymbol{c}_{ij}(X_{ij}, Y_{ij}, c_{ij})$. De controledriehoeken worden gedefinieerd als $T_i(\boldsymbol{c}_{i1}, \boldsymbol{c}_{i2}, \boldsymbol{c}_{i3})$ en zijn rakend aan het oppervlak in $\boldsymbol{s}(V_i)$. De projectie van de controledriehoeken T_i in het domein geeft de PS-driehoeken t_i.

De uitbreiding naar parametrische Powell-Sabin spline-oppervlakken is

$$\boldsymbol{s}(u,v) = \sum_{i=1}^{n} \sum_{j=1}^{3} \boldsymbol{c}_{ij} B_{ij}(u,v), \qquad (u,v) \in \Omega$$

met controlepunten $\boldsymbol{c}_{ij} = (c_{ij}^x, c_{ij}^y, c_{ij}^z)$. Het oppervlak ligt binnen de convex omhullende van de controlepunten. Uit de Powell-Sabin voorstelling met controlepunten kan gemakkelijk de Bernstein-Bézier voorstelling gevonden worden.

3 De B-spline basis

Stabiliteit is een wenselijke eigenschap voor een spline basis. In praktische toepassingen is het van groot belang dat een kleine perturbatie van de coëfficiënten niet leidt tot een grote perturbatie van de functie zelf. Het blijkt een nodige voorwaarde te zijn dat de oppervlakte van de PS-driehoeken begrensd is. Bovendien zijn bij een kleine PS-driehoek de basisfuncties lineair onafhankelijk. We bespreken ook een praktisch algoritme om een kleine PS-driehoek te vinden.

3.1 Stabiliteit

We bewijzen dat de basisfuncties $\{B_{ij}(x,y)\}$ een stabiele basis vormen voor de splineruimte $S_2^1(\Delta^*)$. Er bestaan constanten k_1 en k_2 die enkel afhangen van de kleinste hoek θ_Δ in de driehoeksverdeling, zodanig dat voor alle mogelijke keuzes van coëfficiënten geldt dat

$$k_1 \|\mathbf{c}\|_\infty \leq \| \sum_{i=1}^{n} \sum_{j=1}^{3} c_{ij} B_{ij}(x,y) \|_\infty \leq k_2 \|\mathbf{c}\|_\infty$$

met $\|\mathbf{c}\|_\infty := \max_{ij} |c_{ij}|$ en $\|f\|_\infty := \max_\Omega |f(x,y)|$.

Een belangrijk element in het bewijs is de keuze van de PS-driehoeken. Er zijn eigenlijk oneindig veel mogelijkheden omdat de enige voorwaarde is dat deze driehoek de juist PS-punten bevat. Het blijkt echter dat de stabiliteitsconstanten k_1 en k_2 afhankelijk zijn van de grootte van de PS-driehoek. De oppervlakte moet beperkt zijn.

We vinden dat

$$k_1 = \frac{1}{K_1},$$
$$k_2 = 1,$$

met

$$K_1 = \left[1 + \frac{384\sqrt{3}K}{\sin(\theta_\Delta \sin(\theta_\Delta)/4)^4 \tan(\theta_\Delta \sin(\theta_\Delta)/8)}\right].$$

In deze uitdrukking is K een factor die de grootte van de PS-driehoek weerspiegelt en θ_Δ de kleinste hoek in de driehoeksverdeling.

Het conditiegetal van een basis is gedefinieerd als de verhouding van de stabiliteitsconstanten: $\kappa = k_2/k_1$. Het best mogelijke conditiegetal is $\kappa = 1$ voor een orthonormale basis. In het geval van de B-spline basis voor de oneindig norm vinden we $\kappa_\infty = K_1$. Als de kleinste hoek in de driehoeksverdeling θ_Δ kleiner wordt of als de PS-driehoek en dus ook K groter wordt zal het conditiegetal stijgen. We besluiten dat kleine hoeken en grote PS-driehoeken ongunstig zijn voor de stabiliteit.

Om stabiliteit te bewijzen voor andere normen dan de oneindig norm, moeten we eerst de basisfuncties normaliseren met de oppervlakte van hun domein. Dit is analoog aan ééndimensionale B-splines op de reële lijn. Deze normalisatie komt voort uit het feit dat de basisfuncties een eenheidspartitie vormen en daarom hun volume afhankelijk is van de oppervlakte van hun domein.

De genormaliseerde basisfuncties zijn

$$B'_l := A_l^{-1/p} B_{ij}(x, y), \quad l = 3(i-1) + j$$

en de genormaliseerde coëfficiënten zijn

$$c'_l := A_l^{1/p} c_{ij}, \quad l = 3(i-1) + j.$$

Hierin staat A_l voor de oppervlakte van het domein van de overeenkomstige basisfunctie. We noteren met $\mathbf{B}'$ en $\mathbf{c}'$ de rij- en kolomvector die respectievelijk de genormaliseerde basisfuncties en de genormaliseerde coëfficiënten bevatten.

3.2 Keuze van de PS-driehoeken

In de oorspronkelijke constructie van de B-splines werd de driehoek met de kleinst mogelijke oppervlakte gekozen. Dit komt neer op het oplossen van een klein kwadratisch optimalisatieprobleem voor ieder punt. We noemen dit de optimale oplossing. We kunnen ook gemakkelijk nagaan dat dit overeenkomt met lineair onafhankelijke basisfuncties en dat als de PS-driehoek groter wordt, de drie basisfuncties in een punt afhankelijk worden.

We stellen een suboptimale procedure voor die snel een kleine PS-driehoek berekent zonder optimalisatieroutines te gebruiken. Het is gebaseerd op de observatie dat als een punt slechts grenst aan drie driehoeken in de driehoeksverdeling, de PS-punten zelf een driehoek vormen. Indien er meer driehoeken in het spel zijn vormen de PS-punten een veelhoek. We zoeken dan een driehoek die die veelhoek omvat.

Het algoritme in pseudocode is:

```
A = 0
for i = 1 to n do
    L1 = rechte door p_i en p_i+1
    for j = i+1 to n do
        L2 = rechte door p_j en p_j+1
        if !(L1 // L2) do
           Q1 = doorsnede(L1,L2)
           P = rechte loodrecht op bissectrice van L1
              en L2 en door het verste PS-punt
           Q2 = doorsnede(L2,P)
           Q3 = doorsnede(P,L1)
        end if
        if Opp < A or A = 0 do A = Opp
    end for j
end for i
```

We hebben zowel de optimale als de suboptimale procedure toegepast op een voorbeeld met 21 punten en 31 driehoeken. De gemiddelde oppervlakte van de suboptimale PS-driehoeken is 4.05% groter dan de gemiddelde oppervlakte van de optimale PS-driehoeken.

4 Multiresolutietechnieken

Multiresolutietechnieken omvatten het genereren, voorstellen en manipuleren van geometrische objecten op verschillende niveaus van detail. In dit hoofdstuk geven we een overzicht van de multiresolutietechnieken die we gebruiken in deze thesis.

4.1 Subdivisie

Subdivisie is een proceduraal mechanisme om curven en oppervlakken te genereren. We starten van een initiële verzameling coëfficiënten $\mathbf{s}^0$ die gedefinieerd zijn op een grof rooster in het domein. Dan passen we een filter van lineaire combinaties toe en verkrijgen een grotere verzameling coëfficiënten die leven op een verfijnd rooster.

$$\mathbf{s}^{j+1} = P^j \mathbf{s}^j .$$

Een rooster wordt regelmatig genoemd als alle punten in het domein op regelmatige afstanden van elkaar liggen. Een uniform subdivisieschema werkt op een regelmatig rooster en gebruikt dezelfde gewichten voor elk niveau en binnen een niveau. Een semiregelmatig rooster start van een onregelmatig rooster maar wordt verder gevormd door nieuwe punten toe te voegen op een regelmatige manier. De gewichten van een overeenkomstig semi-uniform subdivisieschema kunnen verschillen afhankelijk van de plaats in het rooster. Dergelijk schema wordt ook wel stationair genoemd omdat de regels niet verschillen van niveau tot niveau. In volledig onregelmatige roosters zijn de nieuwe punten niet gebalanceerd ten opzichte van de oude. De niet uniforme regels kunnen telkens verschillen. Ook de topologie kan onregelmatig zijn, maar in deze thesis beschouwen we enkel onregelmatige roosters met subdivisieconnectiviteit, dat wil zeggen met regelmatige topologie.

De valentie van een punt in een driehoeksrooster is het aantal driehoeken waar het punt toe behoort. In een regelmatig rooster hebben alle punten een valentie zes. Punten in een semiregelmatig of onregelmatig rooster met een valentie verschillend van zes worden buitengewone punten genoemd. Het stencil van een punt is de verzameling van omliggende punten die worden gebruikt om een nieuwe waarde te berekenen.

Een subdivisieschema is interpolerend als de oorspronkelijke waarden in de oude punten behouden blijven. In het andere geval spreken we van benaderende subdivisie. Schema's die afgeleid zijn van spline subdivisieschema's zijn typisch benaderend.

Met elk subdivisieschema kunnen basisfuncties ϕ_k^j geassocieerd worden, de zogenaamde schaalfuncties. Ze zijn gedefinieerd als de limietfunctie als we het

subdivisieschema toepassen op een impuls bestaande uit allemaal nullen en één 1 op positie k op een bepaald niveau j. Op een fijner niveau zijn er meer schaalfuncties dan op een grover niveau.

Een subdivisieschema is lokaal als een coëfficiënt op een fijn niveau een lineaire combinatie is van slechts een beperkt aantal coëfficiënten op het grove niveau. In dat geval zijn de schaalfuncties ook lokaal, hun domein is beperkt. Elke schaalfunctie kan geschreven worden als een lineaire combinatie van schaalfuncties op het volgende fijnere niveau

$$\Phi^j = \Phi^{j+1} P^j,$$

waarbij de vector Φ^j alle schaalfuncties bevat op het niveau j.

Door deze verfijningsrelatie kunnen we deelruimten V^j associëren met het initieel grof rooster

$$V^j = span\{\phi_k^j\}$$

en

$$V_0 \subset V^1 \subset V^2 \ldots$$

Dit wordt een multiresolutie-analyse genoemd. De orde van een multiresolutie-analyse is de maximale graad van veeltermen welke vervat zitten in de ruimten V^j.

4.2 Wavelets

Tussen twee opeenvolgende ruimten V^j en V^{j+1} is een verschil dat kan worden voorgesteld in een complementruimte W^j

$$V^j \oplus W^j = V^{j+1}.$$

De basisfuncties ψ_k^j van W^j worden wavelets genoemd. De complementruimte W^j is niet noodzakelijk orthogonaal op V^j. Dikwijls is het zo dat W^j orthogonaal staat op veeltermen van lage graad die in V^j vervat zitten. De maximale graad van deze veeltermen is het aantal nulmomenten.

Elke functie in V^{j+1} kan geschreven worden als een gedeelte in V^j en een gedeelte in W^j

$$\Phi^{j+1}\mathbf{s}^{j+1} = \Phi^j\mathbf{s}^j + \Psi^j\mathbf{w}^j.$$

De schaalfuncties en waveletfuncties zijn zodanig gekozen dat V^j het laagfrequente gedeelte van V^{j+1} bevat en W^j het hoogfrequente gedeelte.

Analoog aan de verfijningsrelatie voor schaalfuncties kunnen ook de waveletfuncties geschreven worden als lineaire combinaties van schaalfuncties op het fijnere niveau

$$\Psi^j = \Phi^{j+1} Q^j.$$

Door al de vorige vergelijkingen te combineren vinden we dat de coëfficiënten op niveau $j+1$ kunnen gereconstrueerd uit de benadering en de details

$$\mathbf{s}^{j+1} = P^j \mathbf{s}^j + Q^j \mathbf{w}^j.$$

Dit is de synthese of de inverse wavelettransformatie. De decompositie of analyse gebeurt met de inverse filters A^j en B^j

$$\begin{aligned} \mathbf{s}^j &= A^j \mathbf{s}^{j+1} \\ \mathbf{w}^j &= B^j \mathbf{s}^{j+1}. \end{aligned}$$

De voorwaartse wavelet transformatie kan herhaald worden op het laagfrequente gedeelte $\mathbf{s}^j$ en heeft als resultaat $\mathbf{s}^{j-1}$ en $\mathbf{w}^{j-1}$. Dit kan herhaald worden tot op niveau 0.

Analyse en synthese zijn elkaars inverse. De oorspronkelijke informatie $\mathbf{s}^{j+1}$ op niveau $j+1$ kan, afrondingsfouten buiten beschouwing gelaten, exact gereconstrueerd worden uit de twee componenten $\mathbf{s}^j$ en $\mathbf{w}^j$ op niveau j. In praktische toepassingen zullen $\mathbf{s}^j$ en $\mathbf{w}^j$ echter gefilterd worden, en de reconstructie is een benadering van de oorspronkelijke data.

Voor de praktische uitvoering van de wavelettransformatie hebben we de schaalfuncties en de waveletfuncties zelf niet nodig, enkel de coëfficiënten en de lineaire combinaties om van het ene niveau naar het andere te gaan. We willen echter vermijden de filters P^j, Q^j, A^j en B^j expliciet te moeten berekenen. Daarom gebruiken we het lifting schema. De belangrijkste eigenschap van dit schema is dat het niet gebaseerd is op Fourier analyse, maar een volledige ruimtelijke interpretatie heeft. Daarom is het geschikt om wavelets te ontwerpen op onregelmatige roosters.

Eerst splitsen we de verzameling coëfficiënten in twee, een groep die overeenkomt met de oude punten in het subdivisieschema, en een groep die overeenkomt met de nieuwe punten. Dan proberen we de waarden in de nieuwe punten te voorspellen aan de hand van de waarden in de oude punten. Hiervoor gebruiken we de subdivisieregels. De voorspellingen worden afgetrokken van de werkelijke waarden. De verschillen zijn de waveletcoëfficiënten. Dan passen we op die waveletcoëfficiënten een bijsturingsfilter toe en tellen het resultaat op bij de waarden

in de oude punten, wat ons de schaalcoëfficiënten geeft. De bijsturingsstap wordt bijvoorbeeld gebruikt om het gemiddelde in het laagfrequente gedeelte te bewaren.

Een bijkomend voordeel van het lifting schema is dat de inverse transformatie voor de hand liggend is. We kunnen gemakkelijk de voorspelling en de bijsturing in de andere richting uitvoeren.

Een beperking van de hier voorgestelde wavelettransformatie is dat ze enkel kan toegepast worden op roosters met subdivisieconnectiviteit, dus semiregelmatige roosters, of onregelmatige roosters met regelmatige topologie.

5 Spline subdivisie

We willen de B-spline voorstelling berekenen van een Powell-Sabin spline op een verfijning Δ^1 van de gegeven driehoeksverdeling Δ^0. De nieuwe basisfuncties hebben een kleiner domein. Als we dan één coëfficiënt van een controlepunt aanpassen wordt het oppervlak slechts beïnvloedt in een kleinere omgeving van het punt. In de limiet convergeren de controlepunten naar het oppervlak zelf. Spline subdivisie wordt daarom gebruikt als visualisatietechniek.

5.1 Dyadische subdivisie

De meest gebruikte procedure om driehoeksroosters te verfijnen is dyadische subdivisie. Hierbij wordt een nieuw punt toegevoegd tussen elke twee oude punten en elke oorspronkelijke driehoek wordt gesplitst in vier nieuwe driehoeken.

Voor uniforme Powell-Sabin splines bestaat reeds een dyadisch subdivisieschema. Dit is bijzonder eenvoudig dankzij het feit dat we in het uniforme geval een vaste vorm kunnen kiezen voor de PS-driehoeken. Dit vereenvoudigt de berekeningen aanzienlijk.

Als we de posities kiezen van de nieuwe punten is het noodzakelijk dat de lijnen van C^2 discontinuïteit in het Powell-Sabin element bewaard blijven in de verfijnde driehoeksverdeling. Daarom moeten de nieuwe punten geplaatst worden op de intersecties R_{ij} van de PS-lijnen en de zijden van de driehoeken. Dit heeft als gevolg dat soms het inwendige punt Z niet ligt in de middelste van de vier nieuwe driehoeken. In dat geval is het onmogelijk de lijnen van het Powell-Sabin element te bewaren in de verfijnde driehoeksverdeling.

Concreet moet aan de volgende voorwaarden voldaan zijn

$$\begin{aligned}
\frac{1}{2} &< \frac{(1-\lambda_{ij})\lambda_{ki}}{b_{ijk}\lambda_{ki} + c_{ijk}(1-\lambda_{ij})} < 1 \\
\frac{1}{2} &< \frac{(1-\lambda_{jk})\lambda_{ij}}{a_{ijk}(1-\lambda_{jk}) + c_{ijk}\lambda_{ij}} < 1 \\
\frac{1}{2} &< \frac{\lambda_{jk}(1-\lambda_{ki})}{a_{ijk}\lambda_{jk} + b_{ijk}(1-\lambda_{ki})} < 1
\end{aligned}$$

waarbij

$$\begin{aligned}
R_{ij} &= \lambda_{ij}V_i + (1-\lambda_{ij})V_j \\
R_{jk} &= \lambda_{jk}V_j + (1-\lambda_{jk})V_k \\
R_{ki} &= \lambda_{ki}V_k + (1-\lambda_{ki})V_i \\
Z_{ijk} &= a_{ijk}V_i + b_{ijk}V_j + c_{ijk}V_k.
\end{aligned}$$

in een driehoek (V_i, V_j, V_k).

Dyadische verfijning is dus niet altijd mogelijk voor Powell-Sabin splines op onregelmatige driehoeksverdelingen. Daarom nemen we dit niet verder in beschouwing in deze thesis.

5.2 $\sqrt{3}$ subdivisie

Een ander verfijningsschema is $\sqrt{3}$ subdivisie. Hierbij wordt een nieuw punt toegevoegd in het midden van elke oorspronkelijke driehoek en de punten worden opnieuw getriangulariseerd. De oude zijden van de driehoeken worden niet behouden, maar worden vervangen door nieuwe zijden die het nieuwe punt verbinden met de oorspronkelijke hoekpunten en met de nieuwe punten in de aangrenzende oude driehoeken. Bij een $\sqrt{3}$ schema gaat de verfijning trager, het aantal driehoeken verdriedubbeld slechts in elke stap. Er is een fijnere gradatie van niveaus.

Om een geldige verfijning te bekomen moet het nieuwe punt in elke driehoek gekozen worden op de plaats van het inwendige punt Z in de PS-verfijning. Voor de PS-verfijning van de nieuwe driehoeken moet dan wel nog een nieuw inwendig punt gekozen worden. Het voordeel van $\sqrt{3}$ subdivisie is dat er altijd een mogelijkheid is om die nieuwe inwendige punten zodanig te kiezen dat de nieuwe PS-verfijning geldig is en compatibel met de oude.

We noteren met V_{ijk} het nieuwe punt dat toegevoegd wordt in een oorspronkelijke driehoek (V_i, V_j, V_k). De nieuwe controledriehoek $T_{ijk}^{\sqrt{3}}$ in V_{ijk} is

$$\begin{aligned}
\boldsymbol{c}_{ijk,1}^{\sqrt{3}} &= \tilde{L}_{i1}\boldsymbol{c}_{i1} + \tilde{L}_{i2}\boldsymbol{c}_{i2} + \tilde{L}_{i3}\boldsymbol{c}_{i3} \\
\boldsymbol{c}_{ijk,2}^{\sqrt{3}} &= \tilde{L}_{j1}\boldsymbol{c}_{j1} + \tilde{L}_{j2}\boldsymbol{c}_{j2} + \tilde{L}_{j3}\boldsymbol{c}_{j3} \\
\boldsymbol{c}_{ijk,3}^{\sqrt{3}} &= \tilde{L}_{k1}\boldsymbol{c}_{k1} + \tilde{L}_{k2}\boldsymbol{c}_{k2} + \tilde{L}_{k3}\boldsymbol{c}_{k3}.
\end{aligned}$$

In deze formules zijn de $\tilde{L}$ de barycentrische coördinaten van bepaalde PS-punten ten opzichte van de PS-driehoek van het punt waartoe ze behoren.

Om de nieuwe controledriehoeken te berekenen gebruiken we enkel convexe combinaties van de data op het grovere niveau. De barycentrische coördinaten $\tilde{L}$ hebben immers een waarde tussen 0 en 1. Dit betekent dat het subdivisieschema numeriek stabiel is.

5.3 Triadische subdivisie

Als we de $\sqrt{3}$ procedure een tweede keer toepassen resulteert dit in een triadische verfijning van de oorspronkelijke driehoeksverdeling. Elke oorspronkelijke rand wordt in drie verdeeld en elke oorspronkelijke driehoek wordt gesplitst in negen nieuwe driehoeken.

Aan de randen van een begrensd domein hebben we niet automatisch een triadische verdeling. In de eerste $\sqrt{3}$ stap werden daar de oude zijden behouden omdat er geen aanliggende oorspronkelijke driehoek is in de welke ook een nieuw punt werd gekozen. In de tweede $\sqrt{3}$ stap moeten we in zulk een driehoek met nog een oude zijde twee nieuwe punten kiezen in plaats van één, en wel op die zijde zelf.

Zij die zijde V_iV_j dan duiden we de twee nieuwe punten aan met V_{ij} en V_{ji}. Om de overeenkomstige controledriehoeken te vinden kunnen we in twee van de drie hoekpunten van de controledriehoek dezelfde regels als in het inwendige van het domein gebruiken. Voor het derde hoekpunt geldt een aangepaste regel:

$$\begin{aligned}
\boldsymbol{c}_{ij,2} = &(\omega_{ij} + \lambda_{ij} - \omega_{ij}\lambda_{ij})(L_{i1}\boldsymbol{c}_{i1} + L_{i2}\boldsymbol{c}_{i2} + L_{i3}\boldsymbol{c}_{i3}) \\
&+ (1 - \omega_{ij} - \lambda_{ij} + \omega_{ij}\lambda_{ij})(L_{j1}\boldsymbol{c}_{j1} + L_{j2}\boldsymbol{c}_{j2} + L_{j3}\boldsymbol{c}_{j3}).
\end{aligned}$$

Ook hier worden enkel convexe combinaties gebruikt, en het subdivisieschema blijft numeriek stabiel.

Voor de oude punten kunnen we steeds de bestaande PS- en controledriehoek behouden. De betrokken PS-punten zullen dichter bij het punt liggen dan voor de

subdivisie, en zullen dus zeker in de PS-driehoek liggen. Uit de stabiliteitsanalyse van de B-spline basis was evenwel gebleken dat het opportuun is om een zo klein mogelijke PS-driehoek te hebben. Daarom stellen we voor in iedere stap de oude PS-driehoeken te scaleren.

Een courante toepassing van spline subdivisie is de visualisatie van de spline oppervlakken. In de limiet convergeren de controlepunten naar het oppervlak zelf. Bij Powell-Sabin splines is het zelfs zo dat de punten waarin de controledriehoeken het oppervlak raken op het limiet oppervlak liggen. Het driedimensionaal rooster dat de raakpunten verbindt wordt gebruikt voor de grafische voorstelling.

Een belangrijke motivatie voor de studie van subdivisie algoritmen is de connectie met multiresolutie- analyse en wavelets. De opeenvolgende spline ruimten vormen een geneste rij en de basisfuncties voldoen aan een verfijningsrelatie. In het volgende hoofdstuk gebruiken we het triadische schema om Powell-Sabin spline wavelets te ontwikkelen.

6 Spline wavelets

In de literatuur vinden we verschillende spline gebaseerde wavelet constructies terug. Er is echter weinig gepubliceerd over spline wavelets op driehoeksverdelingen. Het ontwerp dat we hier voorstellen maakt gebruik van het lifting schema met de triadische spline subdivisie als voorspellingsstap.

6.1 Multischaal decompositie

Als schaalruimten V^j gebruiken we de Powell-Sabin ruimte $S_2^1(\Delta^{j*})$ op de driehoeksverdeling Δ^j op niveau j. Voor elke ruimte V^j hebben we een stabiele basis nodig. De genormaliseerde B-splines uit Hoofdstuk 3 voldoen hieraan. We kiezen dus

$$\begin{aligned} \mathbf{s} &= \mathbf{c}', \\ \Phi &= \mathbf{B}', \end{aligned}$$

waarbij $\mathbf{s}$ staat voor de kolomvector met schaalcoëfficiënten en Φ voor de rijvector met schaalfuncties.

Het verband tussen de opeenvolgende niveaus wordt gegeven door het triadische subdivisieschema. De matrix P^j in de vergelijking

$$\mathbf{s}^{j+1} = P^j \mathbf{s}^j,$$

bevat dus de coëfficiënten uit de subdivisieregels in hoofdstuk 5. We splitsen deze matrix in twee stukken, een eerste gedeelte voor de oude punten en een tweede gedeelte voor de nieuwe punten

$$P^j = \begin{bmatrix} O^j \\ N^j \end{bmatrix}.$$

Op dezelfde manier splitsen we de schaalfuncties in functies $\mathcal{O}^j$ en functies $\mathcal{N}^j$

$$\mathbf{\Phi}^{j+1} = [\mathcal{O}^{j+1}\mathcal{N}^{j+1}].$$

De verzameling schaalfuncties Φ^j op niveau j en de deelverzameling schaalfuncties $\mathcal{N}^j$ bij de nieuwe punten op niveau $j+1$ tesamen spannen de ruimte V^{j+1} op. Bijgevolg kunnen de $\mathcal{N}^{j+1}$ gebruikt worden als waveletfuncties, of met andere woorden als basisfuncties voor de complement ruimte W^j

$$\Psi^j = \mathcal{N}^{j+1} \quad \text{en} \quad Q^j = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

Dit is echter niet altijd een goede keuze als we denken aan bijvoorbeeld stabiliteit. Daarom projecteren we de $\mathcal{N}^{j+1}$ op de gewenste complementruimte W^j

$$\Psi^j = \mathcal{N}^{j+1} - \Phi^j U^j.$$

Deze projectie is niet noodzakelijk orthogonaal.

De reconstructiefilters zijn dan

$$\begin{bmatrix} P^j & Q^j \end{bmatrix} = \begin{bmatrix} O^j & -O^j U^j \\ N^j & I - N^j U^j \end{bmatrix}.$$

Dit kan gefactoriseerd worden als volgt

$$\begin{bmatrix} P^j & Q^j \end{bmatrix} = \begin{bmatrix} O^j & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ N^j & I \end{bmatrix} \cdot \begin{bmatrix} I & -U^j \\ 0 & I \end{bmatrix}.$$

Hierin herkennen we het lifting schema. Eerst worden de coëfficiënten $\mathbf{s}^{j+1}$ opgesplitst in twee deelverzamelingen. De eerste deelverzameling bevat de coëfficiënten $\mathbf{s}_o^{j+1}$ die overeenkomen met de oude punten in het subdivisieschema, en de tweede deelverzameling $\mathbf{s}_n^{j+1}$ bevat die coëfficiënten die overeenkomen met de nieuwe punten.

Dan beschouwen we de $\mathbf{s}_o^{j+1}$ als controlepunten van een oppervlak gedefinieerd op Δ^j en passen het subdivisieschema toe. De gevonden waarden in de nieuwe punten gebruiken we als voorspelling en trekken we af van de werkelijke waarden $\mathbf{s}_n^{j+1}$. Dit geeft ons de waveletcoëfficiënten $\mathbf{w}^j$. In een laatste stap passen we de coëfficiënten $\mathbf{s}_o^{j+1}$ bij de oude punten aan door er een lineaire combinatie van de reeds gevonden waveletcoëfficiënten $\mathbf{w}^j$ bij op te tellen. Dit geeft ons de schaalcoëfficiënten $\mathbf{s}^j$.

6.2 Eigenschappen van de complementruimte

Een voor de hand liggend criterium voor de decompositie is dat de grove benadering zo dicht mogelijk tegen de oorspronkelijke functie op het fijnere resolutieniveau moet liggen. Optimaal zou zijn als de complement ruimte W^j orthogonaal staat op V^j en we dus een orthogonale decompositie hebben. Als de schaalfuncties in V^j niet alleen orthogonaal zijn ten opzichte van de wavelets in W^j maar ook ten opzichte van de andere schaalfuncties op niveau j noemen we transformatie volledig orthogonaal. Een orthogonale decompositie zonder bijkomende orthogonaliteit tussen de schaalfuncties noemen we semi-orthogonaal.

Zowel semi-orthogonale als orthogonale wavelettransformaties zijn moeilijk te ontwerpen. Daarom wordt dikwijls vereist dat de wavelets orthogonaal zijn ten opzichte van veeltermen van lage graad. We noemen deze graad het aantal nulmomenten. De achterliggende reden hiervan is dat de schaalruimtes meestal veeltermen tot en met een bepaalde graad bevatten.

Als we een wavelettransformatie uitvoeren zou er geen verlies aan nauwkeurigheid mogen optreden, of met andere woorden, het conditiegetal moet beperkt blijven. We noemen de transformatie dan stabiel. In het geval van een orthogonale of een semi-orthogonale transformatie is dit automatisch voldaan. Uit de literatuur weten we ook dat de transformatie niet stabiel kan zijn als ze niet minstens één nulmoment heeft.

De stabiliteitsvereiste kan ook uitgedrukt worden als de eis dat de multischaalbasis bestaande uit de schaalfuncties op het grofste niveau aangevuld met de waveletfuncties op alle niveaus een stabiele basis is. Een nodige voorwaarde daarvoor is dat elke basis Ψ^j op zich een stabiele basis is. Dit is geen voldoende voorwaarde omdat in theorie de multischaalbasis in de limiet nog onstabiel kan zijn. Deze vorm van stabiliteit noemen we complement stabiliteit.

6.3 Ontwerp van de bijsturingsstap

Een eenvoudige bijsturingsstap zou eruit kunnen bestaan alle vrijheidsgraden te gebruiken om nulmomenten te creëren. Het aantal vrijheidsgraden hangt af van het aantal schaalfuncties dat wordt betrokken in de de bijsturingsstap. We noemen dit het stencil. De grootte van de bijsturingscoëfficiënten is echter heel gevoelig aan de keuze van het stencil, en stabiliteit is niet gegarandeerd.

Een andere mogelijkheid is de bijsturingsoperator te kiezen als de orthogonale projectie van W^j op V^j. We hebben dan de semi-orthogonale configuratie. Het nadeel van deze bijsturingsstap is echter dat alle schaalfuncties erbij moeten betrokken worden en dat de wavelets niet lokaal zijn.

Een derde mogelijkheid zou zijn een stencil vast te leggen dat slechts een beperkt aantal schaalfuncties bevat en elke wavelet te orthogonaliseren ten opzichte van zijn stencil. De grootte van het stencil is dan een afweging tussen semi-orthogonaliteit en lokaliteit. Een nadeel hier is dan weer dat er geen nulmomenten gegarandeerd zijn.

In ons voorstel leggen we ook het stencil op voorhand vast. Voor een nieuw punt op een zijde van een oorspronkelijke driehoek gebruiken we de schaalfuncties bij de twee buren op die zijde. Voor een nieuw punt in het midden van een driehoek gebruiken we de schaalfuncties bij de drie hoekpunten van die driehoek.

We gebruiken één van de vrijheidsgraden om het eerste nulmoment te garanderen. De overblijvende vrijheidsgraden gebruiken we dan om de wavelets orthogonaal te maken ten opzichte van hun stencil. Hieraan kan niet exact voldaan worden, het stelsel is overgetermineerd. We lossen het op met een kleinstekwadraten methode.

6.4 Resultaten

De waveletfuncties van een wavelettransformatie met één primaal nulmoment hebben een integraal gelijk aan nul. Of met andere woorden het volume onder en boven het nulvlak zijn aan elkaar gelijk. Dit is duidelijk te zien in de tekeningen.

We bewijzen complement stabiliteit, of met andere woorden dat het conditiegetal van een transformatie van één niveau begrensd is. We doen dit door een afschatting te maken voor de grootte van de bijsturingscoëfficiënten. Uit een praktisch voorbeeld blijkt dat de conditiegetallen inderdaad begrensd zijn.

Met elk punt in het domein komt een controledriehoek overeen, dus drie controlepunten in plaats van één. De Powell-Sabin wavelets zijn daarom in feite een voorbeeld van multiwavelets. In het regelmatige geval zou dit betekenen dat de schaalfuncties en waveletfuncties translaties en dilataties zijn van drie verschillende functies in plaats van van één functie. In de onregelmatige configuratie die we hier beschouwen komt het multiwavelet aspect tot uiting in de vorm van de filters. Deze bestaan uit submatrices van telkens drie op drie.

Om oppervlakken te beschouwen die niet gedefinieerd zijn over een vlak, moeten we overgaan naar een parametrische voorstelling. In dat geval is elk van de coördinaten gedefinieerd als een aparte scalaire functie. We passen dan de wavelettransformatie toe op elk van de coördinaten tegelijk. De betekenis van de details als een verschil in de z-richting gaat dan wel verloren, en dit kan tot onverwachte en ongewenste resultaten leiden.

De Powell-Sabin wavelettransformatie zet een verzameling coëfficiënten om naar een andere verzameling met exact evenveel coëfficiënten. Het totaal aantal coëfficiënten blijft gelijk. Omwille van de stabiliteitsproblemen werden in het verleden echter dikwijls constructies gebruikt waarin het aantal coëfficiënten groeit.

7 Veralgemeende subdivisie

Spline-oppervlakken hebben als nadeel dat ze een parametrisatie hebben over een vlak. Voor meer algemene oppervlakken is een dergelijke parametrisatie niet altijd mogelijk. Een mogelijke oplossing is het oppervlak in stukken op te delen, maar continuïteitsvoorwaarden tussen de verschillende deeloppervlakken bemoeilijken dit. Een heel andere aanpak is het gebruik van veelhoeken, bijvoorbeeld driehoekige netten. Deze kunnen wel algemene topologieën voorstellen, maar hebben als nadeel dat heel veel veelhoeken nodig zijn, wat het oppervlak moeilijk handelbaar maakt, en dat ze niet geschikt zijn om zachtverlopende veranderingen over grote oppervlakken te modelleren.

Subdivisie-oppervlakken proberen de voordelen van beide te combineren. Ze worden proceduraal gedefinieerd als de limiet van een verfijningsprocedure op een initieel controlenet. De verfijningsregels kunnen een spline of andere wiskundige achtergrond hebben, maar die wordt niet meer gebruikt in de procedurale definitie. Op die manier kunnen willekeurige topologieën gemodelleerd worden. Ook zijn subdivisie-oppervlakken geschikt om multiresolutietechnieken op toe te passen.

7.1 Subdivisie-oppervlakken

Veralgemeende subdivisieschema's die onstaan als een uitbreiding van uniforme B-spline subdivisie zijn uitermate interessant. In de regelmatige gebieden zal het limietoppervlak een uniform spline-oppervlak zijn. Spline gebaseerde subdivisieschema's zijn typisch benaderend. Interpolerende schema's verdienen echter ook aandacht omdat het in verschillende toepassingen nodig is het limietoppervlak exact te controleren. Er bestaan aanpassingen van benaderende splineschema's zodat het limietoppervlak bepaalde punten en normalen interpoleert, maar de stencils van deze schema's zijn niet beperkt, of de interpolatievoorwaarden zijn enkel voldaan in de limiet en het oppervlak wordt in de praktijk nooit tot in de limiet berekend.

In dit hoofdstuk stellen we een subdivisieschema voor dat is gebaseerd op uniforme Powell-Sabin spline subdivisie. Het limietoppervlak interpoleert de initiële punten en hun normalen. We gebruiken hier niet het triadische subdivisieschema voor niet uniforme driehoeksverdelingen omdat de coëfficiënten in de lineaire combinaties gebruik maken van kennis van het parameterdomein. Dit willen we expliciet vermijden.

Voor elk punt is er een controledriehoek in plaats van één controlepunt, en de driehoek is rakend aan het oppervlak in het barycenter. De hoeken van de controledriehoek veranderen elke iteratie, dus strikt gezien is het schema benaderend. Het schema is echter interpolerend in het raakpunt en de normale.

7.2 Theoretische achtergrond

Het aantal zijden dat toekomt in een punt kan verschillend zijn van punt tot punt. De regels die ontworpen zijn aan de hand van splines of interpolatievoorwaarden gelden enkel voor de regelmatige stukken, dus voor punten met valentie zes in driehoekige schema's. In de buitengewone punten met andere valenties hebben we andere regels nodig. We zitten in het geval van semiregelmatige subdivisie. Het definiëren van de speciale regels is de belangrijkste uitdaging in het ontwerp van veralgemeende subdivisieschema's.

Het controlenet bestaat uit regelmatige stukken en geïsoleerde onregelmatige gebieden met buitengewone punten. In de regelmatige gebieden gelden standaard continuïteitsvoorwaarden. Het sleutelelement in de analyse is nu dat de regelmatige gebieden steeds groter worden en de onregelmatige gebieden kleiner. De regels die gebruikt worden in de omgeving van de buitengewone punten zullen het aantal buitengewone punten niet veranderen, maar voegen enkel regelmatige punten toe. Omdat elke subdivisieregel slechts informatie gebruikt van de punten in de onmid-

dellijke omgeving kunnen we de analyse beperken tot controlenetten bestaande uit een buitengewoon punt omringd met gewone punten.

We noteren die beperkte verzameling punten in de buurt van het buitengewone punt op niveau j met $\mathbf{d}^j$. Het subdivisie-algoritme kan dan beschreven worden door een vierkante matrix S^j

$$\mathbf{d}^j = S^j \mathbf{d}^{j+1}.$$

We noemen deze matrix de subdivisiematrix. S^j is een submatrix van de volledige matrix P^j uit de vorige hoofdstukken. Na subdivisie is de configuratie van de punten in $\mathbf{d}^{j+1}$ dezelfde als de configuratie van de punten in $\mathbf{d}^j$ en we kunnen dezelfde nummering gebruiken. We beschouwen hier enkel stationaire subdivisie-algoritmen die op elk niveau dezelfde regels gebruiken: $S = S^j$ onafhankelijk van j.

Zij $\lambda_0, \ldots, \lambda_K$ de eigenwaarden van S geordend volgens hun absolute waarde

$$|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_K|$$

en zij $\mathbf{v}_0, \ldots, \mathbf{v}_K$ de overeenkomstige eigenvectoren. Als $|\lambda_0| \geq |\lambda_1| = |\lambda_2| > |\lambda_3|$ dan definiëren we de karakteristieke functie Υ als de functie die het domein van de punten in $\mathbf{d}^j$ afbeeldt op het tweedimensionale oppervlak dat bepaald wordt door het tweedimensionale controlenet $[\mathbf{v}_1, \mathbf{v}_2]$. Het geval waarin λ_1 verschillend is van λ_2 hebben we hier niet nodig.

Als $\lambda_1 = \lambda_2, 1 > |\lambda| > |\lambda_3|$ een reële eigenwaarde is met multipliciteit 2 en als de karakteristieke functie regelmatig en injectief is, dan is het limietoppervlak C^1 continu voor bijna alle initiële controlenetten $\mathbf{d}^0$.

7.3 Het Tangent Schema

Voor de gewone punten met valentie zes gebruiken we de subdivisieregels voor uniforme Powell-Sabin splines. Voor de nieuwe controledriehoeken is dit

$$\boldsymbol{c}_{ij}^{j+1} = B\,\boldsymbol{c}_i^j + C\,\boldsymbol{c}_j^j$$

met

$$B = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 1/4 & 0 \\ 1/2 & 0 & 1/4 \end{bmatrix}.$$

De controledriehoek voor de oude punten moet gescaleerd worden

$$\boldsymbol{c}_i^{j+1} = A\,\boldsymbol{c}_i^j$$

met

$$A = \begin{bmatrix} 2/3 & 1/6 & 1/6 \\ 1/6 & 2/3 & 1/6 \\ 1/6 & 1/6 & 2/3 \end{bmatrix}.$$

Om deze regels af te leiden werd verondersteld dat alle controledriehoeken in dezelfde richting georiënteerd zijn en dat de hoeken in dezelfde volgorde genummerd zijn. In de buurt van een buitengewoon punt echter komen verschillende stukken regelmatig oppervlak samen en is dit niet mogelijk. Daarom leiden we ook regels af voor andere configuraties, zoals bijvoorbeeld voor een zijde waarop de twee PS-driehoeken $60°$ ten opzichte van elkaar gedraaid zijn

$$\boldsymbol{c}_{jk}^{j+1} = K\,\boldsymbol{c}_{j}^{j} + L\,\boldsymbol{c}_{k}^{j}$$

met

$$K = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 \end{bmatrix}, \quad L = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \end{bmatrix}.$$

Deze regel kan gemakkelijk gevonden worden aan de hand van de onderliggende Bézier voorstelling.

Om de verdere analyse te vergemakkelijken introduceren we de rotatiematrix Q_6 en verminderen zo het aantal regels dat expliciet moet worden afgeleid

$$Q_6 = \begin{bmatrix} 2/3 & 2/3 & -1/3 \\ -1/3 & 2/3 & 2/3 \\ 2/3 & -1/3 & 2/3 \end{bmatrix}.$$

We kunnen nu, na eventueel de rotatiematrix het gepast aantal keren toegepast te hebben, steeds dezelfde reeds genoemde regels gebruiken.

Voor de buitengewone punten vinden we nieuwe regels door een gelijkaardige rotatiematrix te definiëren

$$Q_n = \begin{bmatrix} a & b & c \\ c & a & b \\ b & c & a \end{bmatrix} \quad \text{met} \quad \begin{aligned} a &= \tfrac{1}{3} + \tfrac{2}{3}\cos\left(\tfrac{2\pi}{n}\right) \\ b &= \tfrac{1}{3} - \tfrac{1}{3}\cos\left(\tfrac{2\pi}{n}\right) + \tfrac{1}{\sqrt{3}}\sin\left(\tfrac{2\pi}{n}\right). \\ c &= \tfrac{1}{3} - \tfrac{1}{3}\cos\left(\tfrac{2\pi}{n}\right) - \tfrac{1}{\sqrt{3}}\sin\left(\tfrac{2\pi}{n}\right) \end{aligned}$$

We gebruiken verder dezelfde regels als voor de regelmatige punten.

7.4 Resultaten

Om de convergentie te analyseren kijken we naar de structuur van de subdivisiematrix.

$$S = \begin{bmatrix} S_0 & S_1 & \cdots & S_{n-1} \\ S_{n-1} & S_0 & \cdots & S_{n-2} \\ \vdots & & \ddots & \vdots \\ S_1 & \cdots & S_{n-1} & S_0 \end{bmatrix}$$

met

$$S_0 = \begin{bmatrix} \frac{1}{n}A & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}B & \frac{1}{2}C & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}A & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}K & 0 & 0 & \frac{1}{2}L & 0 \\ \frac{1}{n}BQ_n^{-n+1} & 0 & 0 & 0 & \frac{1}{2}C & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}A & 0 \end{bmatrix},$$

$$S_1 = \begin{bmatrix} \frac{1}{n}AQ_n^{-1} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}BQ_n^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}L & 0 & 0 & 0 & 0 \\ \frac{1}{n}B & \frac{1}{2}C & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}A & 0 & 0 & 0 & 0 \end{bmatrix}$$

en

$$S_k = \begin{bmatrix} \frac{1}{n}AQ_n^{-k} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}BQ_n^{-k} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}BQ_n^{-k+1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{voor} \quad k = 2, \ldots, n-1.$$

Met behulp van een computeralgebra programma vinden we dat de eerste eigenwaarde gelijk is aan 1, en dat de twee volgende eigenwaarden gelijk zijn aan $1/2$. We kijken ook na of de karakteristieke functie regelmatig en injectief is.

Een voldoende voorwaarde is dat de directionele afgeleide van de karakteristieke functie overal positief is. Dit bewijzen we door over te gaan naar de Bézier voorstelling, en alle Bézier ordinaten blijken inderdaad positief te zijn. We kunnen dus besluiten dat het tangent schema leidt tot limietoppervlakken die C^1 continu zijn voor bijna alle initiële controlenetten $\mathbf{d}^0$.

Welke subdivisieregels moeten gebruikt worden om een nieuwe controledriehoek te vinden tussen twee oude controledriehoeken hangt af van de specifieke configuratie. Als de oude controledriehoeken bijvoorbeeld niet dezelfde oriëntatie hebben moeten ze een aantal keer gedraaid worden. Deze informatie zit opgeslagen in een vlaggetje voor elke zijde in het driedimensionale net.

Het tangent schema biedt meer mogelijkheden om oppervlakken te modelleren dan de traditionele subdivisieschema's omdat we de raakvector in de beginpunten kunnen kiezen. Ook de grootte van de controledriehoek speelt een rol, kleine controledriehoeken geven scherpe randen en grote controledriehoeken leiden tot rondere randen.

Een nadeel is dat dit schema, zoals de meeste schema's, geen C^2 continuïteit maar slechts C^1 continuïteit biedt. Ook zijn soms de golfjes die typisch zijn voor interpolerende schema's, aanwezig op het limietoppervlak. Wat het geheugengebruik betreft moet voor elk raakpunt drie punten opgeslagen worden in plaats van één.

Dankzij de kennis van de normalen en de raakvectoren is het mogelijk het oppervlak exacter te controleren. Het interpolerend zijn vergemakkelijkt ook de ontwikkeling van subdivisiewavelets of multiresolutietoepassingen. Voor de grafische voorstelling op een computerscherm is de normale nodig, die moeten we met dit schema niet meer benaderen. Het schema is spline gebaseerd. Daardoor hebben we een gesloten mathematische uitdrukking voor het limietoppervlak in de regelmatig gebieden, en de limiet posities zijn exact gekend. Het schema gebruikt enkel informatie van de direct omliggende punten. Er zijn dus geen speciale regels nodig voor de randen van het oppervlak of object.

8 Besluit

8.1 Overzicht van de resultaten

- We hebben bewezen dat de verzameling basisfuncties $\{B_{ij}\}$ een stabiele basis vormt voor de Powell-Sabin splineruimte $S_2^1(\Delta^*)$ voor zowel de oneindig norm als de p-norm. Dit is onderzoek tesamen met Jan Maes.

- We hebben een algoritme gegeven om snel een kleine PS-driehoek te berekenen zonder een optimalisatieprobleem op te lossen.
- Het dyadisch subdivisieschema uitbreiden naar het niet uniforme geval bleek niet altijd mogelijk te zijn. Daarom zijn we er niet verder op ingegaan.
- Als alternatief stelden we $\sqrt{3}$ subdivisie voor. Hierbij is het altijd mogelijk de nieuwe punten zodanig te kiezen dat de verfijning geldig is. Het bewijs voor de nieuwe controledriehoek is in samenwerking met Joris Windmolders.
- Het $\sqrt{3}$ schema twee keer toepassen leidt tot triadische subdivisie. We hebben speciale regels gegeven die moeten toegepast worden aan de randen, en we stelden een optimalisatie voor voor de oude controledriehoeken.
- Het eerste ontwerp voor Powell-Sabin splinewavelets was voor het uniforme geval tesamen met Joris Windmolders. We gebruikten het dyadische subdivisieschema als voorspelling in het lifting schema.
- In het niet uniforme geval gebruiken we het triadische subdivisieschema als voorspelling. In tegenstelling tot de uniforme wavelets is de stabiliteit hier niet voor de hand liggend. We argumenteren dat het belangrijk is tenminste één nulmoment en een zo orthogonaal mogelijke tranformatie te hebben. We bereiken dit door een kleinstekwadraten probleem op te lossen,
- We geven een bewijs van de stabiliteit voor de 1-norm en voor de oneindig norm. Dit is onderzoek tesamen met Jan Maes. We berekenen ook conditiegetallen voor een praktisch voorbeeld.
- We veralgemenen het dyadisch subdivisie-algoritme voor uniforme Powell-Sabin splines naar algemene topologieën. Voor de gewone punten leiden we nog bijkomende regels af aan de hand van de Bézier voorstelling omdat de controledriehoeken niet altijd in dezelfde richting moeten georiënteerd zijn en we introduceren de rotatiematrix. Voor de buitengewone punten ontwerpen we nieuwe regels door middel van een veralgemening van de rotatiematrix.
- We berekenen de eigenwaarden van de subdivisiematrix voor veralgemeende dyadische subdivisie en analyseren de karakteristieke functie. Daaruit blijkt dat het tangent schema convergeert.
- We hebben al de algoritmes geïmplementeerd in C++. We bouwden voort op het pakket PSSURF van Joris Windmolders en pasten het aan. Voor de implementatie van de triadische subdivisie en het tangent schema hebben we andere datastructuren gekozen en die in PSSURF geïntegreerd. We deden ook aanpassingen die de uitbreidbaarheid van het programma verbeterden.

8.2 Toekomstig onderzoek

- Eventueel kunnen de beschreven algoritmen uitgebreid worden naar rationale Powell-Sabin spline-oppervlakken. Rationale voorstellingen geven de ontwerpers meer vrijheidsgraden door middel van gewichten die met elk punt geassocieerd worden.

- Het limietoppervlak van het tangent schema wordt niet alleen bepaald door de richting van de raakvectoren maar ook door de grootte van de controledriehoek. Het zou interessant zijn na te gaan wat er gebeurt als de controledriehoek een lijn of een punt wordt. Zal er nog convergentie zijn? Kunnen we hiermee speciale effecten modelleren?

  Voor de meeste subdivisieschema's bestaan speciale regels om 'cusps' te modelleren. Met het tangent schema zou dat kunnen gedaan worden door de controledriehoek loodrecht op het intiële controlenet te zetten. Zoals dikwijls het geval is in praktische implementaties moeten de hoekpunten van de controledriehoeken in tegenwijzerszin geordend zijn zodat de normale naar buiten wijst. Dit is een praktisch probleem voor het modelleren van cusps.

- Het veelvlak dat dient als initieel controlenet wordt meestal gegeven als een verzameling verbonden driedimensionale punten. In het geval van het tangent schema hebben we ook informatie nodig over de raakvectoren. Deze invoer is niet altijd beschikbaar. Het zou nuttig zijn over een algoritme te beschikken dat automatisch initiële controledriehoeken berekent.

- Het tangent schema zou kunnen gebruikt worden als onderdeel van een 'remeshing' techniek op basis van normalen. Bij deze procedure wordt subdivisie toegepast op een grove resolutie benadering van het onregelmatig net, en dan wordt een benadering berekend voor de normale op die plaats. Het nieuwe punt wordt vervangen door de doorsnede van de normale en het onregelmatige net. De afstand over de normale is de detail- of waveletcoëfficiënt. Het voordeel hiervan is dat slechts één waveletcoëfficiënt nodig is in plaats van de gebruikelijke drie in een parametrische voorstelling. Het bijkomend voordeel van het tangent schema is dat de informatie omtrent de normale vervat zit in de voorstelling.

Contents

Notations and Abbreviations

List of Symbols

Symbol	:	Description
$(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$	:	function values and partial derivatives of B_{ij} at V_i
A^j	:	low pass analysis filter
$Ar_{\mathcal{T}}$	:	area of $\mathcal{T}$
$b(\tau)$	:	Bernstein-Bézier polynomial
$\boldsymbol{b}(\tau)$	:	Bernstein-Bézier surface
b_λ	:	Bézier ordinate
$\boldsymbol{b}_\lambda$	:	Bézier control point
B^j	:	high pass analysis filter
B_λ^m	:	Bernstein polynomial of degree m
B_{ij}	:	B-spline
B'_l	:	normalized B-spline
$\mathbf{B}$	:	rowvector of B-splines
$\mathbf{B}'$	:	rowvector of normalized B-splines
c_{ij}	:	B-spline coefficient
c'_l	:	normalized B-spline coefficient
$\boldsymbol{c}_{ij}$	:	B-spline control point
$\boldsymbol{c'}_{ij}$	:	normalized B-spline control point
$\hat{\boldsymbol{c}}_{ij}$	:	optimized B-spline control point
$\mathbf{c}$	:	column vector of B-spline control points
$\mathbf{c}'$	:	column vector of normalized B-spline control points
C_i	:	circle that contains PS-triangle
d	:	barycentric direction
$\mathbf{d}^j$	:	subset of control net in neighborhood of extraordinary vertex
D_d	:	directional derivative
D	:	diagonal matrix

Δ	:	triangulation
Δ^*	:	Powell-Sabin refinement of Δ
$e_{max}(\mathcal{T})$	:	longest edge in $\mathcal{T}$
I	:	identity matrix
K_1, K_2, K_3	:	constants in stability theorems
$L, \tilde{L}, L'$	:	barycentric coordinates of PS-points $S, \tilde{S}, S'$
λ	:	multi index
λ_i	:	eigenvalue
m_i	:	molecule number
M_i	:	molecule
M^j	:	single scale wavelet transform matrix
n	:	number of vertices in triangulation
N	:	number of vanishing moments
$\tilde{N}$	:	order of multiresolution analysis
N^j	:	part of P^j corresponding to new vertices
$\tilde{N}^j$	:	factor of N^j for unnormalized coefficients
$\mathcal{N}^{j+1}$	:	row vector of scaling functions corresponding to new vertices
O^j	:	part of P^j corresponding to old vertices
$\tilde{O}^j$	:	factor of O^j for unnormalized coefficients
$\mathcal{O}^{j+1}$	:	row vector of scaling functions corresponding to old vertices
$\mathbf{p}^j$	:	control net
ϕ_k^j	:	scaling function
ψ_k^j	:	wavelet function
$\mathbf{\Phi}$	:	rowvector of scaling functions
$\mathbf{\Psi}$	:	rowvector of wavelet functions
P^j	:	low pass synthesis filter, subdivision matrix
Π_m	:	space of bivariate polynomials of degree m
Q_{ij}	:	corner of PS-triangle
Q^j	:	high pass synthesis filter
Q_n	:	rotation matrix
$r_{\mathcal{T}}$	:	radius of largest disk contained in $\mathcal{T}$
$\boldsymbol{r}$	:	surface ring
R_{ij}	:	intersection of Powell-Sabin split line with edge V_iV_j
$R_{\mathcal{T}}$	:	diameter of smallest disk containing $\mathcal{T}$
$s(x, y)$	:	Powell-Sabin spline function
$\boldsymbol{s}(x, y)$	:	Powell-Sabin spline surface
$\mathbf{s}$	:	column vector of scaling coefficients
$S, \tilde{S}, S'$	:	PS-points
$S_m^r(\Delta)$	:	space of degree m splines with C^r continuity on Δ
$S_2^1(\Delta^*)$	:	space of Powell-Sabin splines
S^j	:	square subdivision matrix
t_i	:	PS-triangle

T_i	:	control triangle
T_{C_i}	:	equilateral triangle that contains PS-triangle
T^n	:	multiscale wavelet transform matrix
τ	:	barycentric coordinate
$\mathcal{T}$	:	triangle
$\theta_{\mathcal{T}}$	:	smallest angle in $\mathcal{T}$
Ω	:	parameter domain
$\mathbf{u}$	:	column of U^j
U^j	:	update filter
Υ	:	characteristic map
Υ_i	:	segment of characteristic map
$\mathbf{v}_i$	:	eigenvector
$\hat{\mathbf{v}}$	:	eigenvector of subdominant eigenvalue
V_i	:	vertex
V^j	:	approximation space
$\mathbf{w}$	:	column vector of wavelet coefficients
W^j	:	wavelet space
ξ	:	Bézier triangle point
X_{ij}	:	x-coordinate of Q_{ij}
Y_{ij}	:	y-coordinate of Q_{ij}
Z	:	interior point

List of Abbreviations

CAD	:	Computer Aided Design
CAGD	:	Computer Aided Geometric Design
MRA	:	Multiresolution analysis
PS	:	Powell-Sabin
DC	:	Direct current

Chapter 1

Introduction

1.1 Curve and surface representations

The ability to represent complex shapes on the computer is important in many applications. Basic primitives that are still often encountered are Bernstein polynomials. They were introduced by P. de Casteljau at Citroën in the late fifties [37]. His breakthrough insight was to work with control polygons, a technique that was never used before. Instead of defining a curve through interpolation points, a control polygon utilizes points close to it and instead of changing the curve or surface directly, the control polygon is changed, and the curve follows in a very intuitive way.

Around the same time P. Bézier at Renault represented a basic curve as the intersection of two elliptic cylinders [7]. Later he moved to polynomial formulations and the result turned out to be identical to de Casteljau's curves, only the mathematics involved was different. It was A.R. Forrest who realized that Bézier curves could also be expressed in terms of Bernstein polynomials [52].

A spline was originally a flexible strip of wood that was bent with metal weights to interpolate certain points. The resulting shape has minimal potential energy and is visually pleasing because the curvature cannot change discontinuously. At the US aircraft company Boeing, J. Ferguson and D. MacLaren had the idea to piece cubic patches together so that they formed composite curves which were overall twice differentiable [50, 83]. These curves were referred to as spline curves since they minimize a functional similar to the physical properties of mechanical splines. The meaning of the term spline curve has since undergone a change: spline curves are now mostly thought of as piecewise polynomial curves with certain smoothness properties.

B-splines or basic splines go back to I. Schoenberg who introduced them in 1946 for the case of uniform knots [103]. They were initially defined using a tedious divided difference approach which was numerically very unstable. It was the recursive definition and the recursive evaluation algorithm of C. de Boor [33] in the early seventies that made B-splines a popular tool. Later it was discovered that the recursive evaluation was the natural generalization of the de Casteljau algorithm and it was realized that B-spline curves include Bézier curves as a proper subset.

The most popular of all surface methods is the tensor product surface. It was first introduced by C. de Boor for the case of bicubic spline interpolation [32]. Tensor product B-spline surfaces and their rational extension to NURBS (Non Uniform Rational B-Spline) surfaces are today the most commonly used surface types in commercial CAD packages and computer graphics applications. However, a disadvantage is that they are restricted to rectangular grids. While they have proven themselves an excellent tool for the modeling of largely regular surfaces, e.g. outer car bodies, they have drawbacks if the modeling of largely irregular objects, e.g. inner car panels, is required. Indeed many 3D models require the use of more general surface patches that not necessarily have four edges. Therefore, not surprisingly, the need for representations over non rectangular regions has been expressed quite early [101].

An alternative for the tensor product representation on a rectangular domain, is to write it in terms of barycentric coordinates with respect to a triangular domain. While tensor product surfaces are far more often encountered, triangular ones have been around for a long time also. The first uses of such triangular surfaces go back to finite elements. The simplest type is the linear element, which is simply a planar triangular facet. From a historical perspective, it is interesting to observe that early finite element research did not make use of the elegant formalism offered by the use of barycentric coordinates or the Bernstein-Bézier form. Triangular patches in Bernstein form (called Bézier triangles although Bezier never made any mention of them) are due to de Casteljau; however, that work was never published.

In contrast to univariate spline theory, it is a non trivial problem to determine the dimension of bivariate spline spaces on arbitrary triangulations. Historically G. Strang [109] was the first to pose the problem. In general, the dimension cannot be expressed in terms of the numbers of vertices and triangles of the triangulation because it also depends on the topology and geometry. A lower bound on the dimension was given in [104] and upper bounds in [105, 99]. For several specific choices for the polynomial degree and the smoothness exact results are found [88, 29, 30, 62, 2, 3], but in general and specially for low degree polynomials the problem remains open.

An idea inspired by finite elements and macro elements is to modify the given triangulation by applying a certain splitting procedure to each triangle in the triangulation. More than one polynomial piece is then needed to describe every original triangle. A well-known element is given by R. Clough and J. Tocher [17] for cubic spline spaces. For quadratic splines M. Powell and M. Sabin [92] constructed an element by splitting every triangle into six subtriangles with a common vertex. The dimension of the space of C^1 continuous quadratics polynomials over this split only depends on the number of vertices in the original triangulation. P. Sablonnière proved that the corresponding Hermite interpolating spline has optimal approximation order [102]. Further results on interpolation and approximation using this element were obtained in [61, 24].

P. Dierckx et al. [42] proposed a partially orthonormalized basis for C^1 continuous quadratic splines on Powell-Sabin splits with respect to data points in a least-squares computation problem. Also, the Bézier ordinates on each of the subtriangles resulting from the PS-refinement have been derived explicitly. Later he constructed a normalized B-spline basis that forms a convex partition of unity [41].

1.2 Multiresolution

In 1974, on the CAGD conference at the University of Utah, one of the presenters was the graphics artist G. Chaikin. He presented a curve generation method that was different from all the other methods discussed at the conference. Starting from a closed two-dimensional polygon, and using a process of continual 'chopping off corners' he arrived at a smooth limit curve [13]. It turned out that Chaikin had invented an iterative way to generate uniform quadratic B-spline curves [98].

Such an iterative procedure to construct smooth curves and surfaces by inserting new points and filtering the old ones is now called a subdivision scheme. Other examples are the de Casteljau algorithm and its natural generalization to the recursive B-spline evaluation algorithm. In 1987, C. de Boor discovered that generalizations of corner cutting also produce continuous curves [35]. He also pointed out that Chaikin's algorithm was a special case of a class of algorithms described by G. de Rham much earlier [38, 39]. Spline based subdivision is typically approximating; the original control points change each iteration. Interpolating subdivision schemes were discovered by N. Dyn, D. Levin and J. Gregory [46], S. Dubuc [45], G. Deslauriers and S. Dubuc [40].

The corner cutting algorithm of Chaikin was also the starting point for the work on generalized subdivision surfaces, going back to two articles by D. Doo

and M. Sabin [44] and E. Catmull and J. Clark [9]. They first generalized a one-dimensional B-spline subdivision algorithm to tensor product surfaces. Then they reformulated it such that it could also be applied to control meshes of arbitrary topology. Therefore they defined additional rules for the extraordinary vertices in the irregular parts. For example in a triangular mesh an extraordinary vertex belongs to a number of edges that is different from six. Both schemes yield smooth surfaces though not spline surfaces anymore. In 1987, Loop generalized triangular box spline surfaces to a new subdivision surface type where the input control polygon can be any triangular mesh [78] and N. Dyn, J. Gregory and D. Levin generalized their 4-point interpolation subdivision scheme to surfaces, the so-called Butterfly scheme [47].

The main mathematical challenge in understanding the nature of subdivision surfaces is determining the behavior near the extraordinary points. This was first analyzed for quadrilateral schemes by D. Doo and M. Sabin [44] using Fourier transforms and an eigenanalysis of the subdivision process. This work was expanded upon by A. Ball and D. Storry [5] who derived some necessary conditions for tangent plane continuity near the extraordinary points. However, these are not sufficient because self intersections are still possible. U. Reif [96] derived necessary and sufficient smoothness conditions by taking into account the eigenproperties of the subdivision matrix as well as the properties of the regular basis functions.

Classically the term multiresolution has been intimately connected with the study of wavelets [25]. A function can be viewed as composed of a smooth background and fluctuations or details on top of it. The distinction between the smooth part and the details is determined by the resolution, that is, by the scale below which the details of a signal cannot be discerned. At a given resolution, a function is approximated by ignoring all fluctuations below that scale. Progressively increasing the resolution and adding finer details to the coarser description at each stage, provides a successively better approximation to the function. With each resolution a subspace of the function space is associated. The sequence of nested spaces is called a multiresolution analysis.

The first recorded mention of what we now call a wavelet seems to be in 1909, in a thesis by A. Haar [59]. The concept of wavelets in its present theoretical form was first proposed by J. Morlet and the team at the Marseille Theoretical Physics Center working under A. Grossmann in France [55, 56]. The methods of wavelet analysis have been developed mainly by Y. Meyer and his colleagues [77], who have ensured the methods' dissemination. The formal definition of multiresolution analysis was introduced by S. Mallat [85, 87, 86]. The particular appeal of wavelets derives from their representational and computational efficiency: most data sets exhibit correlation both in time or space and in frequency. These can be modeled with high accuracy through sparse combinations of wavelets.

With every subdivision scheme a multiresolution analysis can be associated. This connection was discovered by M. Lounsbery, T. DeRose and J. Warren [80]. Their subdivision wavelets are an example of second generation wavelets which in contrast to the traditional setting are not defined as translates and dilates of one particular function. The classical techniques work well in a regular setting but they are less useful in the more general settings that engineers and computer scientists encounter in real life situations.

A more general framework to construct second generation wavelets is given by W. Sweldens with the lifting scheme [113, 112, 111]. It starts with a very simple or trivial multiresolution analysis and gradually builds a multiresolution analysis with particular improved properties. All existing transforms in the first generation setting can be factored into lifting steps [27]. The main feature of lifting is that it provides an entirely spatial interpretation of the wavelet transform as opposed to the more traditional Fourier based constructions. This makes it possible to define wavelets on non uniform grids.

1.3 Outline of the thesis

In Chapter 2 we recall the fundamental theory of Bernstein-Bézier polynomials and Powell-Sabin splines. First we review basic concepts such as barycentric coordinates, polynomials on triangles and the de Casteljau algorithm. Besides evaluation of Bernstein-Bézier polynomials the de Casteljau algorithm also gives continuity conditions between neighboring triangles. Second, we define polynomial spline spaces and in particular the space of C^1 continuous quadratic splines over Powell-Sabin splits. We summarize the construction of the normalized B-spline basis and show how control triangles can be associated with the B-spline coefficients. We also give the conversion to the Bernstein-Bézier form.

In Chapter 3 we focus on some properties of the B-spline basis. First, we prove that it is a stable basis. We start with some properties of and lemmas on Powell-Sabin refinements. A key element is the fact that the area of the PS-triangles must be bounded. To prove stability for all p-norms, the basis functions have to be rescaled with respect to their support because the B-splines form a partition of unity and therefore their volume depends on their support. Second, we show that choosing a small PS-triangle corresponds to choosing three linearly independent basis functions for one vertex. We also propose a practical algorithm that quickly yields a nearly optimal solution for the PS-triangles. We depict the sequence of steps, discuss a special case and give the outline in pseudo code. We also compare the optimal and the nearly optimal solution on the basis of an example.

Chapter 4 is an overview of the multiresolution techniques used in this dissertation. We introduce subdivision as an iterative procedure to construct smooth surfaces and we settle some terminology. We define the concept of multiresolution analysis starting from basis functions that satisfy a refinement equation. Then we explain the wavelet filter bank algorithm and build the multiresolution representation on the coarsest grid also starting from subdivision. Every filter bank can be factored into lifting steps using the technique of the lifting scheme. We use this paradigm for the design and the practical execution of the wavelet transform. The subdivision based multiresolution representation applies to surfaces with a specific structure called subdivision connectivity. The structure of the grid is determined by the coarsest level.

In Chapter 5 we discuss spline subdivision schemes for the Powell-Sabin spline surfaces. The goal is to calculate the B-spline representation of a surface on a refinement of the given triangulation. After recalling the dyadic scheme for uniform Powell-Sabin splines, we show why this principle cannot be used in the non uniform case. Conditions on the initial triangulation and its Powell-Sabin split are necessary to ensure the existence of the refined triangulation. As an alternative we propose a $\sqrt{3}$ scheme. It can be used for both the uniform and the non uniform setting. The proof is based on the underlying Bézier representation. Applying the $\sqrt{3}$ scheme twice yields a triadic scheme, but special care is needed at the boundaries. In the light of the discussion on stability and the choice of the PS-triangles, we propose an optimization for the control points introduced in the first $\sqrt{3}$ step.

Chapter 6 is devoted to spline wavelets. The multiresolution representation is based on the triadic subdivision scheme which means that the subdivision algorithm appears as the prediction step in the lifting scheme. A second lifting step, the update step, is used to build basis functions for the complement spaces. The choice of the complement space depends on approximation and stability considerations. When doing the wavelet transform there should be no significant loss of accuracy in the data. We must design the update so that the transform is numerically stable. The ideal setting would be an orthogonal transform and we know that at least one vanishing moment is required. Also we want the stencil to be limited to avoid globally supported wavelets. We propose a least squares solution of an overdetermined system of equations. We prove the stability for the 2-norm and give condition numbers for an example.

In Chapter 7 we leave the functional setting and move to surfaces of arbitrary topology. A drawback of spline surfaces is that they have to be homomorphic to pieces of the plane. Generalized subdivision techniques give procedural definitions for surfaces that do not necessarily have a planar parameterization. We propose a new subdivision scheme that is a generalization of the dyadic scheme for uniform

Powell-Sabin splines. The limit surface interpolates a particular point and its unit normal for each initial vertex. For each vertex there is a control triangle that is tangent to the surface instead of a control point. The scheme is interpolating in the tangent point, though the corners of the triangle change each iteration. We explicitly construct the subdivision matrix in the neighborhood of the extraordinary points and calculate its eigenvalues. We give a convergence analysis based on the eigenanalysis and the properties of the characteristic map.

Finally we give an overview of the contributions, conclusions and suggestions for further research in Chapter 8.

Chapter 2

Powell-Sabin Splines

2.1 Introduction

In this chapter we summarize the fundamental theory of Powell-Sabin splines. These are piecewise C^1 continuous quadratic polynomials on Powell-Sabin split triangulations. We use them to represent surfaces and geometric objects. We also give a short overview of the Bernstein-Bézier representation because the Powell-Sabin representation is build on it.

Until now it has been impossible to determine the dimension of spline spaces on arbitrary triangulations for polynomials of low degree in terms of the number of vertices and the number of triangles in the triangulation. A common solution is the use of split triangulations: each triangle is split in smaller triangles in a structured manner and the dimension on that refined triangulation can uniquely be determined. For the Powell-Sabin split other bases than the one described in this chapter are around in the literature [73, 4], but their main drawback is that they do not form a convex partition of unity.

Section 2.2 reviews the formalism of barycentric coordinates, the Bernstein-Bézier representation of polynomials on triangles and the de Casteljau algorithm [49] and its applications. Section 2.3 is devoted to Powell-Sabin splines. They overcome some disadvantages of the Bernstein-Bézier representation. The Powell-Sabin split [92] defines the class of triangulations on which we are working. We recall the normalized B-spline representation of [41], the geometric interpretation with control triangles and the B-spline to Bézier conversion [42].

2.2 Bernstein-Bézier polynomials

This section gives the definition and some properties of Bernstein-Bézier polynomials and triangular Bézier patches. In some texts the blossom notation is used, but this notation tends to obscure the concepts and is therefore not used here. Unless stated otherwise, the proofs and further details can be found in the comprehensive work of Farin [49].

2.2.1 Barycentric coordinates

Bézier triangles are constructed on a triangular domain. Barycentric coordinates provide an elegant tool for defining points in a plane with respect to the triangular reference frame.

Definition 2.1 *Let $\mathcal{T}(T_1, T_2, T_3)$ be a non degenerate triangle. The barycentric coordinates of an arbitrary point P with respect to $\mathcal{T}$ are $\tau = (\tau_1, \tau_2, \tau_3)$ with*

$$\tau_1 + \tau_2 + \tau_3 = 1 \tag{2.1}$$

and P can be expressed as

$$P = \sum_{i=1}^{3} \tau_i T_i. \tag{2.2}$$

Denote the Cartesian coordinates of the involved points by $T_i(x_i, y_i)$, $i = 1, 2, 3$ and $P(x, y)$, then the barycentric coordinates τ_i, $i = 1, 2, 3$ are the unique solution of the system

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2.3}$$

If the point $P(x, y)$ lies inside $\mathcal{T}$, then $0 \leq \tau_i \leq 1$, $i = 1, 2, 3$. The barycentric coordinates of the vertices T_1, T_2, T_3 are $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ respectively.

Definition 2.2 *Let τ and σ be the barycentric coordinates of two arbitrary points P_1 and P_2, then their difference d defines a barycentric direction.*

A barycentric direction always satisfies $\sum_{i=1}^{3} d_i = 0$.

Definition 2.3 *The directional derivative of a function $f(\tau)$ with respect to a barycentric direction d is*

$$D_d f(\tau) = \sum_{i=1}^{3} \frac{\partial f}{\partial \tau_i} d_i. \tag{2.4}$$

2.2.2 Bernstein-Bézier polynomials

Let $\lambda = (\lambda_1, \lambda_2, \lambda_3),\ |\lambda| = \lambda_1 + \lambda_2 + \lambda_3,\ \lambda_i \in \{0, 1, \ldots, m\}$.

Definition 2.4 *The Bernstein polynomials of degree m over a triangle $\mathcal{T}$ are defined as*

$$B_\lambda^m(\tau) = \frac{m!}{\lambda_1!\lambda_2!\lambda_3!} \tau_1^{\lambda_1} \tau_2^{\lambda_2} \tau_3^{\lambda_3}, \quad |\lambda| = m. \tag{2.5}$$

The Bernstein polynomials form a convex partition of unity on $\mathcal{T}$

$$B_\lambda^m(\tau) \geq 0 \tag{2.6}$$

$$\sum_{|\lambda|=m} B_\lambda^m(\tau) \equiv 1 \tag{2.7}$$

with $\tau \in \mathcal{T}$.

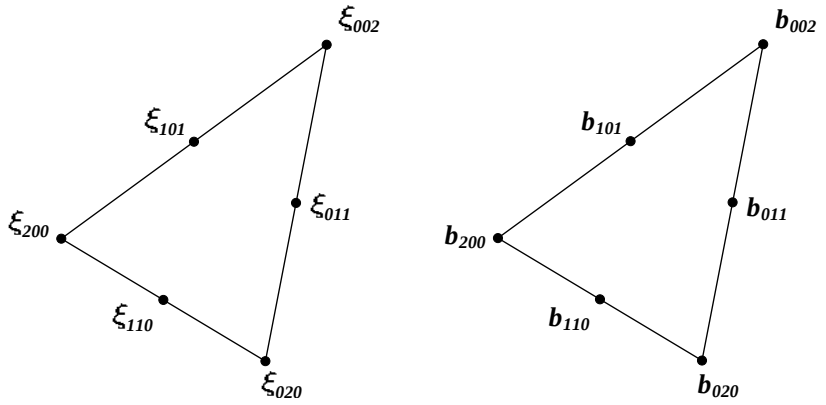
Definition 2.5 *Let Π_m denote the space of bivariate polynomials of total degree $\leq m$ on $\mathcal{T}$. A Bernstein-Bézier polynomial $b(\tau)$ is the unique representation of a polynomial $p(x, y)$, $(x, y) \in \mathbb{R}^2$, in Π_m, in terms of the Bernstein polynomials of degree m on $\mathcal{T}$*

$$p(x, y) := b(\tau) = \sum_{|\lambda|=m} b_\lambda B_\lambda^m(\tau). \tag{2.8}$$

The coefficients b_λ are called the Bézier ordinates. The Bézier triangle points ξ_λ of $\mathcal{T}$ are the points with barycentric coordinates $\frac{\lambda}{m}$ with respect to $\mathcal{T}$ for $|\lambda| = m$, or

$$\xi_\lambda = \frac{\lambda_1 T_1 + \lambda_2 T_2 + \lambda_3 T_3}{m}, \quad |\lambda| = m. \tag{2.9}$$

Figure 2.1(a) shows the Bézier triangle points of a triangle for $m = 2$.



(a) (b)

Figure 2.1: (a) Bézier triangle points of a triangle for $m = 2$. (b) Schematic representation of the Bézier ordinates of a quadratic Bernstein-Bézier polynomial.

Definition 2.6 *A Bernstein-Bézier surface is the graph of a Bernstein-Bézier polynomial. That is the set of points*

$$\boldsymbol{b}(\tau) := \{(x, y, z) \mid (x, y) = \tau \in \mathcal{T},\ z = b(\tau)\}. \tag{2.10}$$

The points $\boldsymbol{b}_\lambda(\xi_\lambda, b_\lambda)$ are the Bézier control points. Each Bézier ordinate b_λ is associated with a Bézier triangle point ξ_λ. This is illustrated schematically in Figure 2.1(b). The linear interpolant of the Bézier control points is called the Bézier control net. Figure 2.2 shows an example of a quadratic Bernstein-Bézier surface with its control points and control net.

Definition 2.7 *A parametric Bernstein-Bézier surface is the set of points*

$$\boldsymbol{b}(\tau) = \begin{cases} x &= \sum_{|\lambda|=m} b_\lambda^x B_\lambda^m(\tau) \\ y &= \sum_{|\lambda|=m} b_\lambda^y B_\lambda^m(\tau), \\ z &= \sum_{|\lambda|=m} b_\lambda^z B_\lambda^m(\tau) \end{cases} \qquad \tau \in \mathcal{T}, \tag{2.11}$$

$$= \sum_{|\lambda|=m} \boldsymbol{b}_\lambda B_\lambda^m(\tau). \tag{2.12}$$

The $\boldsymbol{b}_\lambda = (b_\lambda^x, b_\lambda^y, b_\lambda^z)$ are now the Bézier control points.

Figure 2.2: A quadratic Bernstein-Bézier surface with its control net.

2.2.3 The de Casteljau algorithm

Bernstein-Bézier polynomials can be evaluated using the de Casteljau algorithm [36]. It resolves into repeated linear interpolation.

Theorem 2.1 *The value of a Bernstein-Bézier polynomial in a point with barycentric coordinates τ is*

$$b(\tau) = b^m_{000}(\tau) \tag{2.13}$$

with

$$\begin{aligned} b^0_{\lambda_1,\lambda_2,\lambda_3}(\tau) &= b_{\lambda_1,\lambda_2,\lambda_3}, \\ b^r_{\lambda_1,\lambda_2,\lambda_3}(\tau) &= \tau_1 b^{r-1}_{\lambda_1+1,\lambda_2,\lambda_3}(\tau) + \tau_2 b^{r-1}_{\lambda_1,\lambda_2+1,\lambda_3}(\tau) + \tau_3 b^{r-1}_{\lambda_1,\lambda_2,\lambda_3+1}(\tau), \\ & \qquad r = 1,\ldots,m, \quad |\lambda| = m - r. \end{aligned} \tag{2.14}$$

The b^r_λ are called intermediate de Casteljau ordinates and are actually values of a Bernstein-Bézier polynomial of degree r.

Besides for evaluating a Bernstein-Bézier polynomial, the de Casteljau algorithm can also be used for derivatives, subdivision, continuity conditions,.. We state some important consequences in the next corollaries.

Corollary 2.1 *The r-th directional derivative of a Bernstein-Bézier polynomial $b(\tau)$ with respect to a barycentric direction d is given by*

$$D_d^r(\tau) = \frac{m!}{(m-r)!} \sum_{|\lambda|=r} b_\lambda^{m-r}(\tau) B_\lambda^r(d) \tag{2.15}$$

$$= \frac{m!}{(m-r)!} \sum_{|\lambda|=m-r} b_\lambda^r(d) B_\lambda^{m-r}(\tau), \quad r = 0, \ldots, m. \tag{2.16}$$

To compute the directional derivative $D_d^r(\tau)$, one has to perform $m-r$ de Casteljau steps with respect to τ and r de Casteljau steps with respect to d, but they may be carried out in any order.

This has a nice geometric interpretation: the three control points in the final step define the tangent plane and thus the normal at the domain point with barycentric coordinates τ.

Corollary 2.2 *The tangent plane at the Bernstein-Bézier surface $\boldsymbol{b}(\tau)$ is spanned by the triangle with corners $\boldsymbol{b}_{100}^{m-1}(\tau)$, $\boldsymbol{b}_{010}^{m-1}(\tau)$ and $\boldsymbol{b}_{001}^{m-1}(\tau)$ and the tangent point has barycentric coordinates τ with respect to this triangle.*

For example, if $m = 2$, the tangent plane of $\boldsymbol{b}(\tau)$ at $\tau = (0,0,1)$ is spanned by $\boldsymbol{b}_{002}$, $\boldsymbol{b}_{101}$ and $\boldsymbol{b}_{011}$ and the tangent point is $\boldsymbol{b}_{002}$.

Corollary 2.3 *Let $b(\tau)$ be a Bernstein-Bézier polynomial on $\mathcal{T}(T_1, T_2, T_3)$, and $\hat{b}(\hat{\tau})$ a Bernstein-Bézier polynomial on $\hat{\mathcal{T}}(\hat{T}_1, T_2, T_3)$ where $\hat{T}_1$ has barycentric coordinates σ with respect to $\mathcal{T}(T_1, T_2, T_3)$. The two polynomials $b(\tau)$ and $\hat{b}(\hat{\tau})$ are identical if*

$$\hat{b}_{r,\lambda_2,\lambda_3} = b_{0,\lambda_2,\lambda_3}^r(\sigma), \quad r + \lambda_2 + \lambda_3 = m. \tag{2.17}$$

If $\hat{T}_1$ lies inside $\mathcal{T}$, Corollary 2.3 can be used to find a representation of a Bernstein-Bézier polynomial over a subtriangle of the triangle on which it is defined.

Corollary 2.4 *Let $b(\tau)$ be a Bernstein-Bézier polynomial on $\mathcal{T}(T_1, T_2, T_3)$, and $\hat{b}(\hat{\tau})$ a Bernstein-Bézier polynomial on $\hat{\mathcal{T}}(\hat{T}_1, T_2, T_3)$ where $\hat{T}_1$ has barycentric*

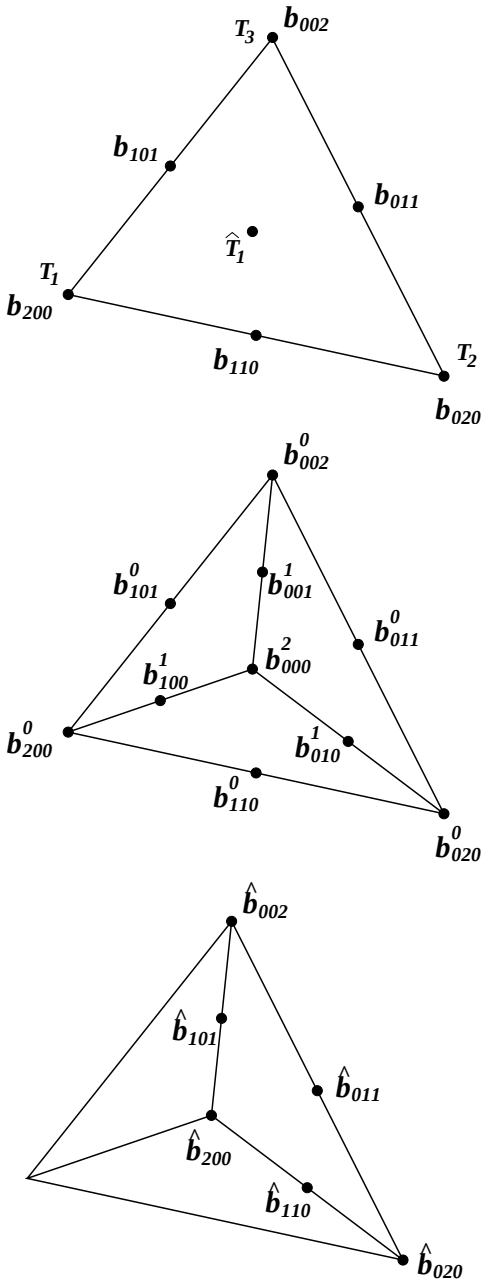


Figure 2.3: Subdivision of a quadratic Bernstein-Bézier polynomial. The intermediate Bézier ordinates of the de Casteljau algorithm are the Bézier ordinates of the polynomial on the subtriangles.

coordinates σ with respect to $\mathcal{T}(T_1, T_2, T_3)$. Suppose $\hat{T}_1$ is inside $\mathcal{T}$, then the intermediate Bézier ordinates $b^r_\lambda(\sigma), |\lambda| = m - r$ are the Bézier ordinates of this polynomial over the three subtriangles $(\hat{T}_1, T_1, T_2)$, $(\hat{T}_1, T_2, T_3)$ and $(\hat{T}_1, T_3, T_1)$.

This is illustrated in Figure 2.3. The Bernstein-Bézier polynomials defined by the intermediate Bézier ordinates $b^r_\lambda(\sigma)$ are a representation of the same original polynomial but with respect to another domain triangle. Corollary 2.4 thus gives a subdivision algorithm for Bernstein-Bézier surfaces.

Corollary 2.5 *Let $b(\tau)$ be a Bernstein-Bézier polynomial on $\mathcal{T}(T_1, T_2, T_3)$, and $\hat{b}(\tau)$ a Bernstein-Bézier polynomial on $\hat{\mathcal{T}}(\hat{T}_1, T_2, T_3)$ where $\hat{T}_1$ has barycentric coordinates σ with respect to $\mathcal{T}(T_1, T_2, T_3)$. A necessary and sufficient condition for $b(\tau)$ and $\hat{b}(\tau)$ to be C^r continuous across the common boundary is*

$$\hat{b}_{\lambda_1,\lambda_2,\lambda_3} = b^{\lambda_1}_{0,\lambda_2,\lambda_3}(\sigma), \quad \lambda_1 = 0, 1, \ldots, r, \quad |\lambda| = m. \tag{2.18}$$

Continuity conditions between adjacent triangles can be expressed as relations between the Bézier ordinates. The triangle pairs along the boundaries should be coplanar. In the particular case of C^1 continuity this becomes

$$C^0 : \ \hat{b}_{0,\lambda_2,\lambda_3} = b_{0,\lambda_2,\lambda_3}, \tag{2.19}$$

$$C^1 : \ \hat{b}_{1,\lambda_2,\lambda_3} = \sigma_1 b_{1,\lambda_2,\lambda_3} + \sigma_2 b_{0,\lambda_2+1,\lambda_3} + \sigma_3 b_{0,\lambda_2,\lambda_3+1}. \tag{2.20}$$

This means that for the example in Figure 2.4 the control points $\boldsymbol{b}_{110}$, $\boldsymbol{b}_{020} = \hat{\boldsymbol{b}}_{020}$, $\boldsymbol{b}_{011} = \hat{\boldsymbol{b}}_{011}$ and $\hat{\boldsymbol{b}}_{110}$ are coplanar. The same is valid for $\boldsymbol{b}_{101}$, $\boldsymbol{b}_{002} = \hat{\boldsymbol{b}}_{002}$, $\boldsymbol{b}_{011} = \hat{\boldsymbol{b}}_{011}$ and $\hat{\boldsymbol{b}}_{101}$.

2.3 Powell-Sabin splines

Although Bernstein-Bézier polynomials provide a nice theoretical framework for dealing with piecewise polynomials on triangulations, they have two main disadvantages. First, they only give the designer pseudo-local control because a change in one control point changes the surface over the whole domain. Second, continuity conditions between polynomials on adjacent triangles result in nontrivial relations between their Bézier coefficients. These relations are hard to implement and make it even more difficult for a designer to make predictable local changes. Therefore, we look for piecewise polynomials on triangulations with global continuity properties.

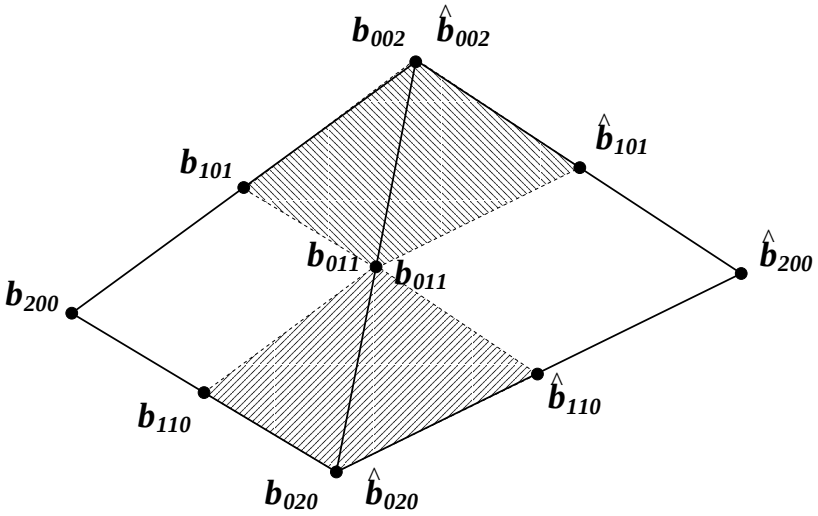


Figure 2.4: C^1 continuity between adjacent Bézier patches: the two pairs of marked triangles have to be coplanar.

2.3.1 Spline space dimension

Consider a simply connected subset $\Omega \subset \mathbb{R}^2$ with polygonal boundary $\delta\Omega$ and a conforming triangulation Δ of Ω, consisting of triangles $\mathcal{T}_j,\ j = 1, \ldots, t$ and vertices V_k with Cartesian coordinates $(x_k, y_k),\ k = 1, \ldots, n$. A triangulation is conforming if for any two triangles $\mathcal{T}_i$ and $\mathcal{T}_j$ with $\mathcal{T}_i \neq \mathcal{T}_j$, the intersection $\mathcal{T}_i \cap \mathcal{T}_j$ is either empty or contains only a common vertex or a common edge.

Definition 2.8 *The polynomial spline space $S_m^r(\Delta)$ is defined as*

$$S_m^r(\Delta) = \{s \in C^r(\Omega) : s|_{\mathcal{T}} \in \Pi_m \ \ , \ \ \mathcal{T} \in \Delta\}, \tag{2.21}$$

where $m > r \geq 0$ are given integers and Π_m is the space of bivariate polynomials of total degree $\leq m$.

Finding suitable bases for spline spaces $S_m^r(\Delta)$ is a non trivial task for $r > 0$, and for general triangulations this can only be done when $m \geq 3r + 2$ [64]. The first constructions were for very special superspline subspaces of $S_m^r(\Delta)$, and can be found in [14, 72]. A general construction for spline spaces $S_m^r(\Delta)$ with $m \geq 3r + 2$ was discovered only recently [31].

For the dimension of spline spaces with $m < 3r+2$ there is no general formula available in terms of the numbers of vertices and triangles of the triangulation because it also depends on its topology and geometry. Therefore we restrict ourselves to a class of triangulations with a special structure. We choose Powell-Sabin triangulations which are obtained from an arbitrary triangulation by splitting each triangle into six subtriangles.

2.3.2 The linear space $S_2^1(\Delta^*)$

Definition 2.9 *A Powell-Sabin refinement Δ^* of Δ divides each triangle into six smaller triangles with a common vertex. It is defined algorithmically as follows (Figure 2.5)*

1. *Choose an interior point Z_j in each triangle $\mathcal{T}_j$, so that if two triangles $\mathcal{T}_i$ and $\mathcal{T}_j$ have a common edge, then the line joining these interior points 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 $\mathcal{T}_j$.*

3. *For each edge of the triangle $\mathcal{T}_j$*

   - *which belongs to the boundary $\delta\Omega$, join Z_j to an arbitrary inner point of the edge.*
   - *which is common to a triangle $\mathcal{T}_i$, join Z_j to R_{ij}.*

Choosing Z_j as the incentre of each triangle $\mathcal{T}_j$ ensures the existence of the points R_{ij}. Other choices, such as the barycenter, may be more appropriate from the practical point of view. The number of triangles in the PS-refinement is equal to six times the number of triangles in the original triangulation.

Definition 2.10 *Given a conforming triangulation Δ of Ω and a PS-refinement Δ^*, the linear space of piecewise C^1 continuous quadratic polynomials on Δ^* is defined as*

$$S_2^1(\Delta^*) := \{s \in C^1(\Omega) : s|_{\mathcal{T}^*} \in \Pi_2,\ \mathcal{T}^* \in \Delta^*\}. \tag{2.22}$$

Each of the $6t$ triangles resulting from the PS–refinement becomes the domain triangle of a quadratic Bernstein–Bézier polynomial as indicated schematically for one subtriangle in Figure 2.5.

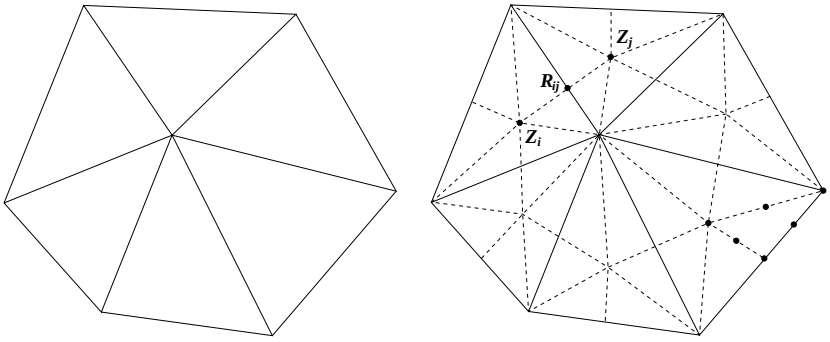


Figure 2.5: PS-refinement. Each triangle $\mathcal{T}_j$ is split into six smaller triangles with a common vertex Z_j.

Powell and Sabin [92] proved that the dimension of the space $S_2^1(\Delta^*)$ equals $3n$ with n the number of vertices. There exists a unique solution $s(x,y) \in S_2^1(\Delta^*)$ for the interpolation problem

$$s(V_k) = f_k, \quad \frac{\partial s}{\partial x}(V_k) = f_{x,k}, \quad \frac{\partial s}{\partial y}(V_k) = f_{y,k}, \quad k = 1, \ldots, n. \tag{2.23}$$

2.3.3 A normalized B-spline representation

Every $s(x,y) \in S_2^1(\Delta^*)$ has a unique representation

$$s(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{3} c_{ij} B_{ij}(x,y). \tag{2.24}$$

There are three basis functions for each vertex as the dimension of $S_2^1(\Delta^*)$ equals $3n$. In the sequel we will sometimes find it convenient to write Equation (2.24) in matrix form as

$$s(x,y) = \mathbf{B}\ \mathbf{c}, \tag{2.25}$$

where $\mathbf{B}$ denotes the row vector of basis functions $B_{ij}(x,y)$ and $\mathbf{c}$ the column vector with the coefficients c_{ij} in the expansion.

Some properties for the basis $\{B_{ij}\}$ such as stability, convex partition of unity and locality are highly desirable in CAGD applications.

Definition 2.11 *The basis $\{B_{ij}\}$ is called stable if there exist constants k_1 and k_2 such that for all choices of the coefficient vector* $\mathbf{c}$

$$k_1\|\mathbf{c}\|_\infty \leq \|\sum_{i=1}^{n}\sum_{j=1}^{3} c_{ij}B_{ij}\|_\infty \leq k_2\|\mathbf{c}\|_\infty \tag{2.26}$$

with $\|\mathbf{c}\|_\infty = \max_{ij}|c_{ij}|$ and $\|f\|_\infty = \max_\Omega |f(x,y)|$ and $0 < k_1, k_2 < \infty$.

Such a stable basis is also called a Riesz basis. The constants k_1 and k_2 are called the lower and upper Riesz bounds. They may depend on the actual geometry of the triangulation, for example on the smallest angle in Δ. The ratio k_2/k_1 is the condition number $\kappa(\mathbf{B})$ of the basis. This property guarantees that a small change in a coefficient c_{ij} corresponds to a small change in the function itself.

Definition 2.12 *The basis $\{B_{ij}\}$ forms a convex partition of unity if*

$$B_{ij}(x,y) \geq 0, \tag{2.27}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{3} B_{ij}(x,y) = 1 \tag{2.28}$$

for $(x,y) \in \Omega$.

This is an important property in CAGD applications. The fact that the basis functions sum up to 1, ensures that the surface is affine invariant: applying an affine transformation to the control points yields the same result as applying the transformation on the surface itself. The nonnegativity of the basis functions guarantees that the surface lies in the convex hull of its so-called control points.

Definition 2.13 *The basis functions B_{ij} have local support if*

$$B_{ij}(x,y) = 0, \quad (x,y) \notin M \subset \Omega \tag{2.29}$$

where M is a connected subset of Ω.

The locality ensures that when the coefficient c_{ij} is changed, the function will only be modified on the subset M of Ω. Note that Bernstein-Bézier polynomials do not possess this property.

Consider three linearly independent triplets $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$, $j = 1, 2, 3$ for each vertex V_i. The basis function $B_{ij}(x, y)$ is the solution of the interpolation problem (2.23) with all $(f_k, f_{x,k}, f_{y,k}) = (0, 0, 0)$ except for $(f_i, f_{x,i}, f_{y,i}) = (\alpha_{ij}, \beta_{ij}, \gamma_{ij})$.

We call the set of triangles $\mathcal{T}_j \in \Delta$ that have V_i as a vertex the molecule M_i. The molecule number m_i is the number of triangles in M_i. The basis function $B_{ij}(x, y)$ obviously vanishes outside the molecule M_i of the vertex V_i: the basis functions are local. If the stability and convex partition of unity property are satisfied depends on the specific choice for $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$.

A historical choice of Shi et al. [107] for the triplets $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ is

$$\begin{aligned} (\alpha_{i1}, \beta_{i1}, \gamma_{i1}) &= (\frac{1}{4}, 0, \epsilon), \\ (\alpha_{i2}, \beta_{i2}, \gamma_{i2}) &= (\frac{1}{4}, \epsilon, 0), \\ (\alpha_{i3}, \beta_{i3}, \gamma_{i3}) &= (\frac{1}{2}, \epsilon, \epsilon). \end{aligned} \tag{2.30}$$

where $\epsilon \in [1/(4h), 1/(2h)]$ and h is the length of the longest edge of Δ. The resulting basis functions form a convex partition of unity, but the basis can be very poor from the stability point of view. For instance, if h is large, then ϵ will be small and the basis functions $B_{ij}, j = 1, 2, 3$ will tend to be numerically linearly dependent.

Hence, even if there is only one triangle with a long edge in the triangulation, the same small value for ϵ will be used for the whole triangulation. Therefore we prefer a choice for $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ that only depends on the molecule M_i and not on the whole triangulation Δ.

Dierckx et al. [42] proposed a partially orthonormalized basis with respect to data points in a least-squares computation problem. Also, the Bézier ordinates on the subtriangles resulting from the PS-refinement have been derived explicitly. This basis has been used for least-squares data fitting by Willemans in [123]. These basis functions however do not form a convex partition of unity.

Lai and Schumaker [73] and Alfeld and Schumaker [4] constructed stable local bases for certain spline spaces $S_d^r(\Delta^*)$ on Powell-Sabin splits. Their construction is based on the principle of minimal determining sets [1]. The basis is stable and locally linearly independent, but the basis functions also do not form a convex partition of unity [118].

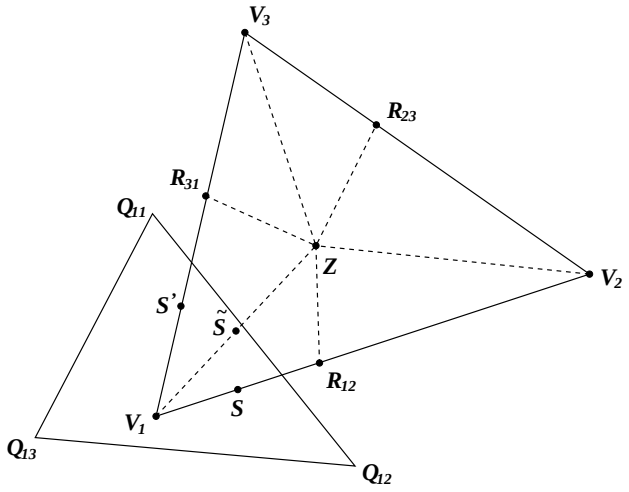


Figure 2.6: The PS-triangle contains all the PS-points of a vertex.

The basis that we will use in this thesis is a normalized B-spline basis suggested by Dierckx in [41] that forms a convex partition of unity. It is local and we prove the stability in Chapter 3. We briefly recall the construction of this basis and its main properties.

1. For each vertex $V_i \in \Delta$, identify its so-called PS-points. To this end consider all edges in the PS-refinement Δ^* that have V_i as a vertex. On each such edge, the Bézier triangle point that is located closest to V_i is a PS-point of V_i. V_i itself is also a PS-point of V_i. Figure 2.6 shows the PS-points S, $\tilde{S}$, S' and V_1 for the vertex V_1 in the triangle $\mathcal{T}(V_1, V_2, V_3)$.

2. For each vertex V_i, find a triangle $t_i(Q_{i1}, Q_{i2}, Q_{i3})$ containing all the PS-points of V_i (from all the triangles $\mathcal{T}_l$ in the molecule M_i). Denote its vertices $Q_{ij}(X_{ij}, Y_{ij})$. The triangles t_i, $i = 1, \ldots, n$ are called PS-triangles. Figure 2.6 also shows the PS-triangle $t_1(Q_{11}, Q_{12}, Q_{13})$ for the vertex V_1.

   We denote the barycentric coordinates of the PS-points $S_i, \tilde{S}_i$ and $S_i^{'}$ with respect to the triangle $t_i(Q_{i1}, Q_{i2}, Q_{i3})$ with (L_{i1}, L_{i2}, L_{i3}), $(\tilde{L}_{i1}, \tilde{L}_{i2}, \tilde{L}_{i3})$ and $(L_{i1}^{'}, L_{i3}^{'}, L_{i3}^{'})$.

3. Given the PS-triangle t_i of a vertex V_i, the three linearly independent triplets of real numbers can be found as follows:

   $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \alpha_{i3})$ are the barycentric coordinates of V_i with respect to t_i,

   $$V_i = \alpha_{i1} Q_{i1} + \alpha_{i2} Q_{i2} + \alpha_{i3} Q_{i3}, \tag{2.31}$$

   $$\beta_i = (\beta_{i1}, \beta_{i2}, \beta_{i3}) = \left(\frac{Y_{i2} - Y_{i3}}{e}, \frac{Y_{i3} - Y_{i1}}{e}, \frac{Y_{i1} - Y_{i2}}{e} \right), \tag{2.32}$$

   $$\gamma_i = (\gamma_{i1}, \gamma_{i2}, \gamma_{i3}) = \left(\frac{X_{i3} - X_{i2}}{e}, \frac{X_{i1} - X_{i3}}{e}, \frac{X_{i2} - X_{i1}}{e} \right), \tag{2.33}$$

   where

   $$e = \begin{vmatrix} X_{i1} & Y_{i1} & 1 \\ X_{i2} & Y_{i2} & 1 \\ X_{i3} & Y_{i3} & 1 \end{vmatrix}. \tag{2.34}$$

   We have $|\alpha_i| = 1$ and $|\beta_i| = |\gamma_i| = 0$.

4. The basis function $B_{ij}(x, y)$ is the unique solution of the interpolation problem (2.23) with all $(f_k, f_{x,k}, f_{y,k}) = (0, 0, 0)$ except for $(f_i, f_{x,i}, f_{y,i}) = (\alpha_{ij}, \beta_{ij}, \gamma_{ij})$.

The fact that the PS-triangle t_i contains the PS-points of the vertex V_i guarantees property (2.27) of the convex partition of unity property. It is also a necessary condition. Figure 2.7 shows an example of a B-spline basis function.

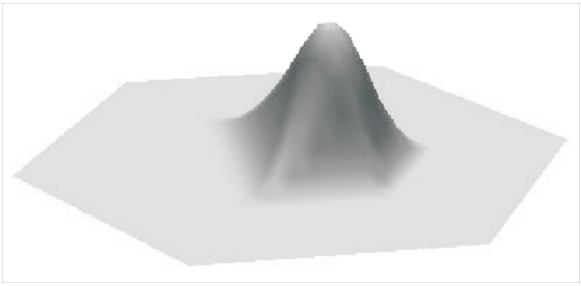


Figure 2.7: B-spline basis function.

Definition 2.14 *A Powell-Sabin spline surface is the graph of a Powell-Sabin polynomial. That is the set of points*

$$\boldsymbol{s}(x,y) = \begin{cases} x &= x \\ y &= y, \\ z &= s(x,y) \end{cases} \qquad (x,y) \in \Omega. \tag{2.35}$$

We define the control points as

$$\boldsymbol{c}_{ij} = (Q_{ij}, c_{ij}) = (X_{ij}, Y_{ij}, c_{ij}) \tag{2.36}$$

and the control triangles as

$$T_i(\boldsymbol{c}_{i1}, \boldsymbol{c}_{i2}, \boldsymbol{c}_{i3}). \tag{2.37}$$

The projection of the control triangles T_i in the (x,y) plane are the PS-triangles t_i. One can prove that the control triangle T_i is tangent to the surface at V_i. The tangent point is $(x_i, y_i, s(V_i))$.

Definition 2.15 *A parametric Powell-Sabin spline surface is the set of points*

$$\boldsymbol{s}(u,v) = \begin{cases} x &= \sum_{i=1}^{n} \sum_{j=1}^{3} c_{ij}^x B_{ij}(u,v) \\ y &= \sum_{i=1}^{n} \sum_{j=1}^{3} c_{ij}^y B_{ij}(u,v), \\ z &= \sum_{i=1}^{n} \sum_{j=1}^{3} c_{ij}^z B_{ij}(u,v) \end{cases} \qquad (u,v) \in \Omega, \tag{2.38}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{3} \boldsymbol{c}_{ij} B_{ij}(u,v), \qquad (u,v) \in \Omega. \tag{2.39}$$

The $\boldsymbol{c}_{ij} = (c_{ij}^x, c_{ij}^y, c_{ij}^z)$ are here the B-spline control points.

2.3.4 Bernstein-Bézier representation

The Bézier representation of a PS-spline surface can be calculated from the B-spline representation [42]. Consider a domain triangle $\mathcal{T}(V_i, V_j, V_k) \in \Delta$ with its PS-refinement as on the top picture in Figure 2.8, where

$$\begin{aligned} R_{ij} &= \lambda_{ij} V_i + (1-\lambda_{ij}) V_j, \\ R_{jk} &= \lambda_{jk} V_j + (1-\lambda_{jk}) V_k, \\ R_{ki} &= \lambda_{ki} V_k + (1-\lambda_{ki}) V_i, \\ Z_{ijk} &= a_{ijk} V_i + b_{ijk} V_j + c_{ijk} V_k. \end{aligned} \tag{2.40}$$

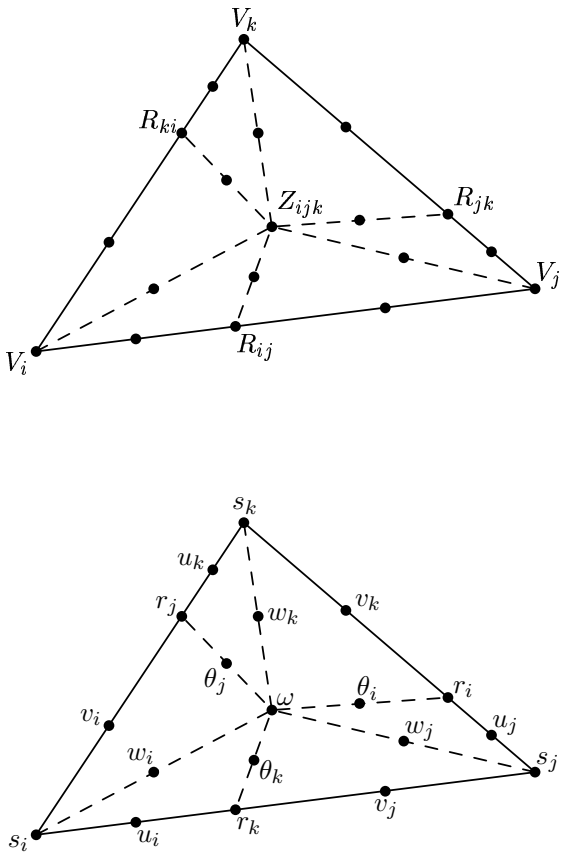


Figure 2.8: Top: PS-refinement of a triangle $\mathcal{T}(V_i, V_j, V_k)$. Bottom: Schematic representation of Bézier ordinates.

Denote the Bézier ordinates as on the bottom picture in Figure 2.8. They can be written as the following unique convex barycentric combinations of the B-spline coefficients:

$$\begin{aligned} s_i &= \alpha_{i1}c_{i1} + \alpha_{i2}c_{i2} + \alpha_{i3}c_{i3}, \\ u_i &= L_{i1}c_{i1} + L_{i2}c_{i2} + L_{i3}c_{i3}, \\ v_i &= L'_{i1}c_{i1} + L'_{i2}c_{i2} + L'_{i3}c_{i3}, \\ w_i &= \tilde{L}_{i1}c_{i1} + \tilde{L}_{i2}c_{i2} + \tilde{L}_{i3}c_{i3}. \end{aligned} \tag{2.41}$$

where the L, $\tilde{L}$ and L' are the barycentric coordinates of certain PS-points with respect to their PS-triangle as described in the construction of the basis functions. The coefficients in these formulae depend on the geometry of the PS-refinement, and on the choice of the PS-triangles.

Similar expressions hold for (s_j, u_j, v_j, w_j) and (s_k, u_k, v_k, w_k). The other Bézier ordinates can be found from the C^1 continuity conditions (2.19) and (2.20), e.g.,

$$\begin{aligned} r_k &= \lambda_{ij}u_i + (1-\lambda_{ij})v_j, \\ \theta_k &= \lambda_{ij}w_i + (1-\lambda_{ij})w_j, \\ \omega &= a_{ijk}w_i + b_{ijk}w_j + c_{ijk}w_k. \end{aligned} \tag{2.42}$$

2.4 Concluding remarks

The Bernstein-Bézier representation underlies the Powell-Sabin B-spline representation and is theoretically well elaborated. We will need this theory and the corollaries of the de Casteljau algorithm in the proofs in the remaining chapters. For example, the new control triangle for $\sqrt{3}$ subdivision in Chapter 5 and the analysis of the characteristic map of the generalized subdivision scheme in Chapter 7 are based on the Bernstein-Bézier representation.

In the remainder of this thesis, the Powell-Sabin splines and their B-spline representation play a central role. We use them as the main tool to represent surfaces and geometric objects. In Chapter 3 we discuss the stability of the B-spline basis more in detail. Stability will be important for the design of wavelets in Chapter 6. We also give an algorithm to calculate suitable PS-triangles for the B-spline representation.

The other main topic of this thesis is multiresolution techniques, more specifically we design multiresolution algorithms for Powell-Sabin splines. This will enable us to use Powell-Sabin splines on different levels of resolution. The basis of all discussed techniques is given by the subdivision algorithms to calculate the B-spline representation of a Powell-Sabin spline on a refined triangulation. An additional advantage of subdivision is that it can be used for the visualization of the splines, whereas before we had to evaluate the spline function through the Bernstein-Bézier representation in certain points.

Because the control triangles are tangent to the spline surface, the B-spline representation also gives information on the first derivatives in the domain vertices, and not only on the function values. We will exploit this property in Chapter 7. The control triangles are used as a modeling tool.

Chapter 3

The B-spline basis

3.1 Introduction

The B-spline basis has already proven to be useful in different CAGD applications. Windmolders et al. studied the uniform case and obtained results on subdivision [125], wavelets [129], data structures for graphical display and the polygonal hole problem [128]. Also a rational extension for the general case was investigated [126] and used for representing quadrics and special effects [127].

Stability is a highly desirable property for a spline basis. In practical applications it is of critical importance that a small perturbation on the coefficients does not lead to a large perturbation on the function itself. Furthermore if a spline space of a certain degree has a stable local basis, it provides optimal order approximation of smooth functions [72]. This is of particular importance in applications as data fitting and the solution of boundary value problems.

If properly normalized the Powell-Sabin B-spline basis is an absolutely stable basis. We first prove this for the infinity norm. Because we have a partition of unity, the volume of the basis functions differs with the size of their domain molecule. In order to form a stable basis for the p-norm we have to normalize the basis functions with respect to their support. The section on stability is joint work with Jan Maes.

In the stability proofs we assume that the area of the PS-triangle is bounded. When the PS-triangle becomes infinitely large, the basis becomes unstable. This indicates that it is important to choose a small PS-triangle in the construction of the basis functions. In [41] Dierckx proposes to solve a quadratic optimization problem to find a PS-triangle that is as small as possible.

We show that choosing a small PS-triangle is equivalent to choosing linearly independent basis functions. We also give a practical algorithm that quickly yields a nearly optimal result. We depict the sequence of steps, discuss special cases and give the outline in pseudo code. The results in this chapter are original contributions.

3.2 Stability

In this section we will prove that the basis functions $\{B_{ij}\}$ form a stable basis for $S_2^1(\Delta^*)$. We follow a similar approach as in [72], where it is proven that the Bernstein polynomials of degree m on a triangle $\mathcal{T}$ form a stable basis for Π_m. Related work has been done for a Hermite basis for quadratic splines. Upper bounds were derived for the Hermite basis functions and for their first derivatives in [102]. Section 3.2.1 is devoted to some useful lemmas on Powell-Sabin splits that will facilitate the proof of the main stability theorem in section 3.2.2. Section 3.2.3 extends the result for the infinity norm to the p-norm.

3.2.1 Properties of Powell-Sabin refinements

Let $\mathcal{T}$ be a triangle in a triangulation Δ and denote by

$$\begin{aligned}
R_{\mathcal{T}} &:= \text{diameter of the smallest disk containing } \mathcal{T},\\
r_{\mathcal{T}} &:= \text{radius of the largest disk contained in } \mathcal{T},\\
\theta_{\mathcal{T}} &:= \text{smallest angle in } \mathcal{T},\\
Ar_{\mathcal{T}} &:= \text{area of } \mathcal{T},\\
e_{\max}(\mathcal{T}) &:= \text{longest edge in } \mathcal{T},\\
e_{\min}(\mathcal{T}) &:= \text{shortest edge in } \mathcal{T},
\end{aligned}$$

and

$$\begin{aligned}
R_\Delta &:= \min_{\mathcal{T}\in\Delta} R_{\mathcal{T}},\\
r_\Delta &:= \min_{\mathcal{T}\in\Delta} r_{\mathcal{T}},\\
\theta_\Delta &:= \min_{\mathcal{T}\in\Delta} \theta_{\mathcal{T}},\\
Ar_\Delta &:= \sum_{\mathcal{T}\in\Delta} Ar_{\mathcal{T}},\\
e_{\max}(\Delta) &:= \max_{\mathcal{T}\in\Delta} e_{\max}(\mathcal{T})\\
e_{\min}(\Delta) &:= \min_{\mathcal{T}\in\Delta} e_{\min}(\mathcal{T}).
\end{aligned}$$

Δ^* en $\mathcal{T}^*$ are the PS-refinements of Δ and $\mathcal{T}$ respectively. These are triangulations and similar definitions as for Δ apply.

The following lemmas give estimates of the above quantities.

Lemma 3.1

$$\frac{R_{\mathcal{T}}}{r_{\mathcal{T}}} \leq \frac{4}{\tan(\theta_{\mathcal{T}}/2)} \tag{3.1}$$

Proof: It is known that [65]

$$r_{\mathcal{T}} = \tan(\theta_{\mathcal{T}}/2) \cdot \frac{a+b-c}{2}, \tag{3.2}$$

in which a, b and c are the lengths of the sides of $\mathcal{T}$. Side length c corresponds to the side opposite to the smallest angle $\theta_{\mathcal{T}}$, and thus has the smallest value. The following inequalities hold:

$$\frac{2}{\tan(\theta_{\mathcal{T}}/2)} = \frac{a+b-c}{r_{\mathcal{T}}} \geq \frac{|e_{\max}(\mathcal{T})|}{r_{\mathcal{T}}} \geq \frac{R_{\mathcal{T}}/2}{r_{\mathcal{T}}}. \tag{3.3}$$

□

Lemma 3.2 *Choose the interior points Z in the PS-refinement as the incenters of the triangles. Then*

$$\theta_{\Delta^*} \geq \theta_{\Delta} \sin(\theta_{\Delta})/4. \tag{3.4}$$

Proof: This lemma is due to Lai and Schumaker [73]. □

This lemma is used each time at the end of the stability proofs to express the Riesz constants in terms of the smallest angle in Δ instead of the smallest angle in Δ^*. Apart from the estimate for θ_{Δ^*} in the last step, the stability proofs do not change if alternatives for Lemma 3.2 with other interior points are considered.

Lemma 3.3

$$\frac{1}{r_{\mathcal{T}^*}} \leq \frac{4}{\sin(\theta_{\mathcal{T}^*})^4 \tan(\theta_{\mathcal{T}^*}/2)|e_{\max}(\mathcal{T}^*)|}. \tag{3.5}$$

Proof: Let e and $\tilde{e}$ be two edges of the same triangle $\mathcal{T}_i^* \in \mathcal{T}^*$. Then

$$\sin(\theta_{\mathcal{T}^*})|e| \leq |\tilde{e}|. \tag{3.6}$$

Between two arbitrary edges e_1 and e_2 in $\mathcal{T}^*$, there always exists a path of maximum four edges in $\mathcal{T}^*$. Applying (3.6) to this path yields

$$|e_1| \leq \left(\frac{1}{\sin(\theta_{\mathcal{T}^*})}\right)^4 |e_2|. \tag{3.7}$$

This also holds for $e_{\min}(\mathcal{T}^*)$ and $e_{\max}(\mathcal{T}^*)$.
Lemma 3.1 implies

$$\frac{|e_{\min}(\mathcal{T}^*)|}{r_{\mathcal{T}^*}} \leq \frac{R_{\mathcal{T}^*}}{r_{\mathcal{T}^*}} \leq \frac{4}{\tan(\theta_{\mathcal{T}^*}/2)}. \tag{3.8}$$

Substitute (3.7) in (3.8) to complete the proof. □

Lemma 3.4 *Consider two arbitrary triangles $\mathcal{T}_1$ and $\mathcal{T}_2$ in the same molecule $M_i \in \Delta$. The fraction of their areas is bounded*

$$\frac{Ar_{\mathcal{T}_2}}{Ar_{\mathcal{T}_1}} \leq K_2, \tag{3.9}$$

where $K_2 := \left(\frac{1}{\sin(\theta_\Delta)}\right)^{\frac{2\pi}{\theta_\Delta}+2} \frac{16}{\tan(\theta_\Delta/2)^2}$.

Proof: The ratio of the areas of the two triangles satisfies

$$\frac{Ar_{\mathcal{T}_2}}{Ar_{\mathcal{T}_1}} \leq \frac{\pi |e_{\max}(\mathcal{T}_2)|^2}{\pi r_{\mathcal{T}_1}^2}. \tag{3.10}$$

Let e and $\tilde{e}$ be two edges of the same triangle $\mathcal{T} \in \Delta$. Then

$$\sin(\theta_\Delta)|e| \leq |\hat{e}|. \tag{3.11}$$

The maximum number of triangles in a molecule M_i is $2\pi/\theta_\Delta$. Between two arbitrary edges e_1 in $\mathcal{T}_1$ and e_2 in $\mathcal{T}_2$ there always exists a path of maximum $\frac{\pi}{\theta_\Delta}+1$ edges in M_i. Applying (3.11) to this path yields

$$|e_2| \leq \left(\frac{1}{\sin(\theta_\Delta)}\right)^{\frac{\pi}{\theta_\Delta}+1} |e_1)|. \tag{3.12}$$

This also holds for $e_2 = e_{\max}(\mathcal{T}_2)$ and $e_1 = e_{\max}(\mathcal{T}_1)$.
Lemma 3.1 implies

$$\frac{|e_{\max}(\mathcal{T})|}{r_{\mathcal{T}}} \leq \frac{R_{\mathcal{T}}}{r_{\mathcal{T}}} \leq \frac{4}{\tan(\theta_\Delta/2)}. \tag{3.13}$$

Combining (3.10), (3.12) and (3.13) completes the proof. □

3.2.2 Stability for the max norm

We want the basis functions $\{B_{ij}(x,y)\}$ to form a stable basis for $S_2^1(\Delta^*)$ according to Definition 2.11, i.e. that there exist constants k_1 and k_2 depending only on the smallest angle θ_Δ such that for all choices of the coefficient vector $\mathbf{c}$

$$k_1\|\mathbf{c}\|_\infty \leq \|\sum_{i=1}^{n}\sum_{j=1}^{3} c_{ij}B_{ij}(x,y)\|_\infty \leq k_2\|\mathbf{c}\|_\infty \tag{3.14}$$

with $\|\mathbf{c}\|_\infty := \max_{ij}|c_{ij}|$ and $\|f\|_\infty := \max_\Omega |f(x,y)|$.

Before we prove the main theorem, we introduce two lemmas. The first lemma gives an upper bound for both $\|D_x\, s(x,y)\|_{\infty,\mathcal{T}_i}$ and $\|D_y\, s(x,y)\|_{\infty,\mathcal{T}_i}$, where

$$\|D_u\, s(x,y)\|_{\infty,\mathcal{T}_i} := \max_{(x,y)\in\mathcal{T}_i} |D_u^1\, s(x,y)|, \tag{3.15}$$

and D_u^r stands for the rth directional derivative as in Definition 2.3.

Lemma 3.5 *Consider a spline function $s(x,y) \in S_2^1(\Delta^*)$ and a triangle $\mathcal{T}_i \in \Delta^*$. Then*

$$\|D_x\, s(x,y)\|_{\infty,\mathcal{T}_i} \leq \frac{12}{r_{\mathcal{T}_i}}\|s(x,y)\|_{\infty,\mathcal{T}_i} \tag{3.16}$$

and

$$\|D_y\, s(x,y)\|_{\infty,\mathcal{T}_i} \leq \frac{12}{r_{\mathcal{T}_i}}\|s(x,y)\|_{\infty,\mathcal{T}_i}. \tag{3.17}$$

Proof: For the proof we refer to the work of Maes et al. [84]. □

The second lemma deals with the choice of the PS-triangles. There are infinitely many possibilities for the PS-triangle because the only condition is that it contains the appropriate PS-points. In section 3.3 we will see that the three basis functions corresponding to a vertex become linearly dependent if the area of the PS-triangle is large. Not surprisingly the constants k_1 and k_2 in the stability definition (3.14) will also depend on the choice of the PS-triangle. If the PS-triangle becomes infinitely large, the basis will be unstable.

In order to make reasonable statements about the magnitude of the coefficients k_1 and k_2 we will assume that the PS-triangle is bounded as follows.

Definition 3.1 *Let C_i be the smallest circle with the vertex V_i as center that contains all the PS-points of V_i as in Figure 3.1 and denote its radius as r_i. An equilateral triangle $\mathcal{T}_{C_i}$ with barycenter V_i and inradius $K_i r_i$ with $K_i \geq 1$ obviously is a valid PS-triangle for V_i. We choose the value K_i such that this equilateral triangle contains the actual PS-triangle t_i. Define $K = \max_i K_i$ as the maximum of all constants K_i in the vertices V_i of Δ.*

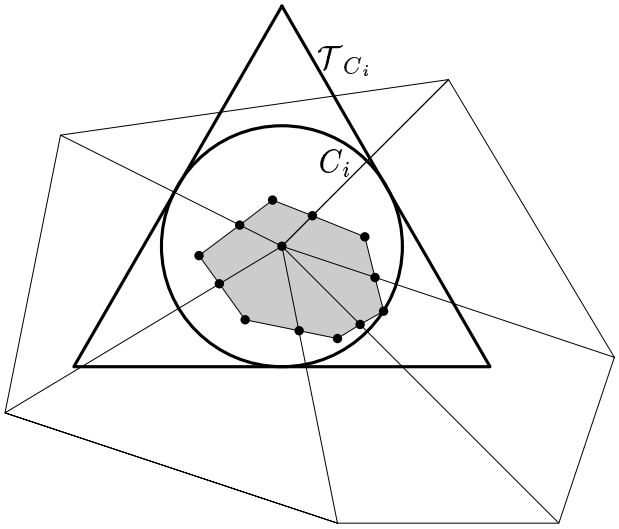


Figure 3.1: The circle C_i that contains all the PS-points of V_i and the equilateral triangle $\mathcal{T}_{C_i}$ for $K_i = 1$.

Lemma 3.6 *Denote by S the PS-point with the longest distance (r_i) to V_i and define $\mathcal{T}_S \in \Delta^*$ as either one of the two triangles of the PS-refinement that contains the PS-point S. Then*

$$\frac{|e_{\max}(t_i)|}{|e_{\max}(\mathcal{T}_S)|} \leq \sqrt{3}K. \tag{3.18}$$

with K and t_i as in Definition 3.1.

Proof: Because t_i is contained in $\mathcal{T}_{C_i}$ it is sufficient to prove that

$$\frac{|e_{\max}(\mathcal{T}_{C_i})|}{|e_{\max}(\mathcal{T}_S)|} \leq \sqrt{3}K. \tag{3.19}$$

Combining

$$|e_{\max}(\mathcal{T}_{C_i})| = 2\sqrt{3}Kr_i. \tag{3.20}$$

and

$$|e_{\max}(\mathcal{T}_S)| \geq 2r_i \tag{3.21}$$

proves the lemma. □

Now we come to the main theorem of this chapter which states that the normalized B-spline basis functions form a stable basis.

Theorem 3.1 *There exists a constant K_1 depending only on the constant K from Definition 3.1 and on the smallest angle θ_Δ in the triangulation such that for all Powell-Sabin splines $s(x,y) \in S_2^1(\Delta^*)$*

$$\|s(x,y)\|_\infty \leq \|\mathbf{c}\|_\infty \leq K_1\|s(x,y)\|_\infty. \tag{3.22}$$

Proof: The left inequality immediately follows from the partition of unity property (2.28).

We now establish the right inequality. From the construction of the B-spline basis functions as the solution of the interpolation problem (2.23) with triplets $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ we know

$$\begin{aligned} s(V_i) &= \alpha_{i1}c_{i1} + \alpha_{i2}c_{i2} + \alpha_{i3}c_{i3}, \\ D_x s(V_i) &= \beta_{i1}c_{i1} + \beta_{i2}c_{i2} + \beta_{i3}c_{i3}, \\ D_y s(V_i) &= \gamma_{i1}c_{i1} + \gamma_{i2}c_{i2} + \gamma_{i3}c_{i3}, \end{aligned}$$

or

$$\begin{bmatrix} s(V_i) \\ D_x s(V_i) \\ D_y s(V_i) \end{bmatrix} = A \begin{bmatrix} c_{i1} \\ c_{i2} \\ c_{i3} \end{bmatrix} \tag{3.23}$$

with

$$A = \begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \alpha_{i3} \\ \beta_{i1} & \beta_{i2} & \beta_{i3} \\ \gamma_{i1} & \gamma_{i2} & \gamma_{i3} \end{bmatrix}. \tag{3.24}$$

We find the inverse of A as

$$A^{-1} = \begin{bmatrix} 1 & \eta_{i1} & \tilde{\eta}_{i1} \\ 1 & \eta_{i2} & \tilde{\eta}_{i2} \\ 1 & \eta_{i3} & \tilde{\eta}_{i3} \end{bmatrix},$$

with

$$\begin{aligned} \eta_{ij} &= \frac{\alpha_{i2}\gamma_{i1} - \alpha_{i1}\gamma_{i2} + \delta_{j1}\gamma_{i2} - \delta_{j2}\gamma_{i1}}{\beta_{i1}\gamma_{i2} - \beta_{i2}\gamma_{i1}}, \\ \tilde{\eta}_{ij} &= \frac{\alpha_{i1}\beta_{i2} - \alpha_{i2}\beta_{i1} - \delta_{j1}\beta_{i2} + \delta_{j2}\beta_{i1}}{\beta_{i1}\gamma_{i2} - \beta_{i2}\gamma_{i1}}, \end{aligned}$$

and δ_{ij} the Kronecker delta.

Suppose that $\|c\|_\infty = |c_{ij}|$ for a certain i and j. Then

$$\|c\|_\infty = |s(V_i) + \eta_{ij} D_x s(V_i) + \tilde{\eta}_{ij} D_y s(V_i)|.$$

Define $\mathcal{T}_S$ as in Lemma 3.6 and apply Lemma 3.5 to it. Together with Equation (3.46) this yields

$$\begin{aligned} \|c\|_\infty &= \|s(x,y)\|_{\infty,\mathcal{T}_S} \left(1 + \eta_{ij}.\frac{12}{r_{\mathcal{T}_S}} + \tilde{\eta}_{ij}.\frac{12}{r_{\mathcal{T}_S}}\right) \\ &\leq \|s(x,y)\|_{\infty,\mathcal{T}_S} \Bigg(1 + \\ &\qquad |\alpha_{i2}\gamma_{i1} - \alpha_{i1}\gamma_{i2} + \delta_{j1}\gamma_{i2} - \delta_{j2}\gamma_{i1}| \cdot 2Ar_{t_i} \cdot \frac{12}{r_{\mathcal{T}_S}} + \\ &\qquad |\alpha_{i1}\beta_{i2} - \alpha_{i2}\beta_{i1} - \delta_{j1}\beta_{i2} + \delta_{j2}\beta_{i1}| \cdot 2Ar_{t_i} \cdot \frac{12}{r_{\mathcal{T}_S}}\Bigg). \end{aligned}$$

If we use the explicit formulas (2.32) and (2.33) for β_{ij} and γ_{ij} together with the fact that $|\beta_i| = |\gamma_i|$, we find

$$\begin{aligned} \|c\|_\infty \leq &\|s(x,y)\|_{\infty,\mathcal{T}_S} \cdot \Bigg[1 + \frac{24Ar_{t_i}}{r_{\mathcal{T}_S}} \cdot \\ &\left(\frac{2|X_{i3} - X_{i2}| + 2|X_{i1} - X_{i3}| + 2|Y_{i3} - Y_{i1}| + 2|Y_{i2} - Y_{i3}|}{2Ar_{t_i}}\right)\Bigg]. \end{aligned}$$

We simplify this equation to

$$\|c\|_\infty \leq \|s(x,y)\|_{\infty,\mathcal{T}_S} \cdot \left[1 + \frac{12}{r_{\mathcal{T}_S}} \cdot (8|e_{\max}(t_i)|)\right].$$

Applying Lemma 3.3 yields

$$\|c\|_\infty \leq \|s(x,y)\|_{\infty,\mathcal{T}_S} \left[1 + \frac{48 \cdot (8|e_{\max}(t_i)|)}{\sin(\theta_{\Delta *})^4 \tan(\theta_{\Delta *}/2)|e_{\max}(\mathcal{T}_S)|}\right],$$

and because of Lemma 3.6 and Lemma 3.2 it follows that

$$\|s(x,y)\|_\infty \leq \|c\|_\infty \leq K_1 \|s(x,y)\|_\infty$$

with

$$K_1 = \left[1 + \frac{384\sqrt{3}K}{\sin(\theta_\Delta \sin(\theta_\Delta)/4)^4 \tan(\theta_\Delta \sin(\theta_\Delta)/8)}\right]. \tag{3.25}$$

□

Theorem 3.1 establishes the stability of the B-spline basis for the max norm. It corresponds to the definition of stability (3.14) with

$$\begin{aligned} k_1 &= \frac{1}{K_1} \\ k_2 &= 1. \end{aligned} \tag{3.26}$$

The lower and upper Riesz bounds k_1 and k_2 determine the condition number $\kappa(\mathbf{B})$ of the basis

$$\kappa(\mathbf{B}) = \frac{k_2}{k_1}. \tag{3.27}$$

The best possible condition number occurs when we have an orthonormal basis with $\kappa = 1$.

For the B-spline basis with the max norm we find

$$\kappa_\infty = K_1. \tag{3.28}$$

From Equation (3.25) we see that if the smallest angle in the triangulation θ_Δ decreases, the condition number K_1 increases. We conclude that small angles in the triangulation are less favorable for stability. The factor K reflects the influence of the size of the PS-triangle. The larger K the larger the PS-triangle, and the larger the condition number K_1. This confirms that it is important to have small areas for the PS-triangles.

3.2.3 Stability for the p-norm

The l_p norm for the coefficient vector $\mathbf{c}$ is defined as

$$\|\mathbf{c}\|_p := \left(\sum_{i=1}^{n} \sum_{j=1}^{3} |c_{ij}|^p \right)^{1/p} \tag{3.29}$$

and the L_p norm for the function f as

$$\|f\|_p := \left(\int_\Omega |f(x,y)|^p dxdy \right)^{1/p}. \tag{3.30}$$

When proving stability for the p-norm [84], there appears a factor that depends on the support of the basis functions through the constants k_1 and k_2. At first sight it seams that the basis on a scaled version of the same domain has a higher condition number. However, this factor is a fraction of the area of a triangle compared to the area of the whole domain, and therefore does not change when a scaling operation is performed on the domain.

The appearance of this factor is due to the fact that the basis functions B_{ij} form a partition of unity and therefore their volume depends on the size of their molecules. We now introduce the normalized basis functions B'_l as

$$B'_l := A_l^{-1/p} B_{ij}(x,y), \quad l = 3(i-1)+j \tag{3.31}$$

and the normalized coefficients $\mathbf{c}'$ as

$$c'_l := A_l^{1/p} c_{ij}, \quad l = 3(i-1)+j \tag{3.32}$$

or in matrix form

$$\begin{aligned} \mathbf{B}' &:= \mathbf{B}\ D^{-1}, \\ \mathbf{c}' &:= D\ \mathbf{c}, \end{aligned} \tag{3.33}$$

where D is a diagonal matrix with as elements the pth root of the areas Ar_l of the support of the corresponding basis function. The representation (2.24) becomes

$$s(x,y) = \sum_{l=1}^{3n} c'_l B'_l, \tag{3.34}$$

or in matrix form

$$s(x,y) = \mathbf{B}\ \mathbf{c} = \mathbf{B}'\ \mathbf{c}'. \tag{3.35}$$

Theorem 3.2 *The functions B_l' form a p-stable basis in the sense that there exist constants k_1 and k_2 that depend only on the smallest angle θ_Δ such that*

$$k_1\|\mathbf{c}'\|_p \leq \|\mathbf{B}'\ \mathbf{c}'\|_p \leq k_2\|\mathbf{c}'\|_p \tag{3.36}$$

for all choices of the coefficient vector $\mathbf{c}'$ and $1 \leq p \leq \infty$.

Proof: We only treat the case $1 \leq p < \infty$. For $p = \infty$ the normalization factor equals 1 and we again find Theorem 3.1.

Choose a triangle $\mathcal{T} \in \Delta$, and define $I_\mathcal{T} := \{l : \text{supp}(\phi_l) \cap \mathcal{T} \neq 0\}$ as the index set of all normalized basis functions that are different from zero on that triangle.

Let $1/p + 1/q = 1$, then by Hölder's inequality for sums

$$\sum c_l' A r_l^{-1/p} B_{ij} \leq \sum |c_l' A r_l^{-1/p} B_{ij}| \tag{3.37}$$

$$\leq \left(\sum |c_l'|^p\right)^{1/p} \left(\sum |A r_l^{-1/p} B_{ij}|^q\right)^{1/q} \tag{3.38}$$

and

$$\begin{aligned}\int_\mathcal{T} |s(x,y)|^p dxdy &= \int_\mathcal{T} |\sum_{l\in I_\mathcal{T}} c_l'\ A r_l^{-1/p} B_{ij}|^p dxdy, \quad l = 3(i-1)+j \\ &\leq \left(\sum_{l\in I_\mathcal{T}} |c_l'|^p\right) \int_\mathcal{T} \left(\sum_{l\in I_\mathcal{T}} A r_l^{-q/p} (B_{ij})^q\right)^{\frac{p}{q}} dxdy. \end{aligned} \tag{3.39}$$

Since $\# I_\mathcal{T} \leq 9$ and B_{ij} is uniformly bounded

$$\int_\mathcal{T} |s(x,y)|^p dxdy \leq 9^{p/q} \max_{l\in I_\mathcal{T}} \frac{A r_\mathcal{T}}{A r_l} \sum_{l\in I_\mathcal{T}} |c_l'|^p \leq 9^p \sum_{l\in I_\mathcal{T}} |c_l'|^p. \tag{3.40}$$

We sum over all triangles $\mathcal{T} \in \Delta$. A certain coefficient s_l can appear more than once on the right-hand side and the number of times it appears can be bounded by the maximum number of triangles in the support of ϕ_l. We find

$$\|s(x,y)\|_p^p = \sum_{\mathcal{T}\in\Delta} \int_\mathcal{T} |s(x,y)|^p dxdy \leq \frac{2\pi}{\theta_\Delta}\ 9\|\mathbf{c}'\|_p^p, \tag{3.41}$$

which proves the upper bound in (3.36).

To prove the lower bound, we use the fact that all norms on a finite dimensional vector space are equivalent. Consider a triangle $\mathcal{T} \in \Delta$ and the standard simplex $\mathcal{T}_s := \{(x,y) : 0 \le x, y \le 1, x+y \le 1\}$. We know that

$$\|s(x,y)\|_{\infty,\mathcal{T}_s} \le K_3 \|s(x,y)\|_{p,\mathcal{T}_s}$$

and therefore also

$$\|s(x,y)\|_{\infty,\mathcal{T}} \le \frac{K_3}{Ar_{\mathcal{T}}^{1/p}} \|s(x,y)\|_{p,\mathcal{T}}.$$

Applying Theorem 3.1 gives

$$\begin{aligned}
\sum_{l \in I_{\mathcal{T}}} |c_l'|^p &\le \max_{l \in I_{\mathcal{T}}} Ar_l \sum_{l \in I_{\mathcal{T}}} |c_l' Ar_l^{-1/p}|^p = \max_{l \in I_{\mathcal{T}}} Ar_l \sum_{l \in I_{\mathcal{T}}} |c_{ij}|^p \\
&\le \max_{l \in I_{\mathcal{T}}} Ar_l \; 9\|\mathbf{c}\|_{\infty,\mathcal{T}}^p \le \max_{l \in I_{\mathcal{T}}} Ar_l \; 9K_1^p \|s(x,y)\|_{\infty,\mathcal{T}}^p \qquad (3.42) \\
&\le \max_{l \in I_{\mathcal{T}}} \frac{Ar_l}{Ar_{\mathcal{T}}} \; 9K_1^p K_3^p \|s(x,y)\|_{p,\mathcal{T}}^p.
\end{aligned}$$

Lemma 3.4 implies

$$\max_{l \in I_{\mathcal{T}}} \frac{Ar_l}{Ar_{\mathcal{T}}} \le \frac{2\pi}{\theta_\Delta} K_2. \qquad (3.43)$$

Summing over all triangles $\mathcal{T} \in \Delta$ yields

$$\|\mathbf{c}'\|_p^p \le \sum_{\mathcal{T} \in \Delta} \sum_{l \in I_{\mathcal{T}}} |c_l'|^p \le \frac{18\pi}{\theta_\Delta} K_1^p K_2 K_3^p \|s(x,y)\|_p^p, \qquad (3.44)$$

which proves the lower bound in (3.36). □

The condition number of the normalized B-spline basis for the p-norm is

$$\kappa_p = \sqrt[p]{\frac{324\pi^2 K_1^p K_2 K_3^p}{\theta_\Delta^2}} \qquad (3.45)$$

We again see the influence of the smallest angle in the triangulation and the size of the PS-triangle through θ_Δ and K_1. The factor K_2 originates from Lemma 3.4 that states that the fraction of two arbitrary triangles in the molecule of a vertex is bounded. We conclude that it is better to have 'balanced' triangulations with triangles of comparable sizes. The factor K_3 is only a correction factor depending on the norm p.

3.3 The choice of the PS-triangles

The construction of the normalized B-spline basis in section 2.3.3 leaves a question unanswered: how to choose the PS-triangles. From the previous section we know that for stability the area of the PS-triangles must be bounded. In this section we show that this also corresponds to choosing linearly independent basis functions in each vertex. In [41] the PS-triangle with the smallest area is chosen. The control points are then close to the surface, which makes it easier to manipulate the surface intuitively. We call this the optimal solution. We propose an alternative algorithm that avoids solving optimization problems and that quickly yields a nearly optimal result for the PS-triangles.

3.3.1 Linearly independent basis functions

The area of the PS-triangle t_i of a vertex V_i is given as

$$Ar(t_i(Q_{i1}, Q_{i2}, Q_{i3})) = \frac{1}{2|\beta_{i1}\gamma_{i2} - \gamma_{i1}\beta_{i2}|}, \tag{3.46}$$

with β_{ij} and γ_{ij} defined as in Equations (2.32) and (2.33).

Theorem 3.3 *Choosing a PS-triangle with bounded area leads to linearly independent basis functions $B_{ij}, j = 1, 2, 3$.*

Proof: This can easily be seen because

$$\begin{vmatrix} \alpha_{i1} & \alpha_{i2} & \alpha_{i3} \\ \beta_{i1} & \beta_{i2} & \beta_{i3} \\ \gamma_{i1} & \gamma_{i2} & \gamma_{i3} \end{vmatrix} = \beta_{i1}\gamma_{i2} - \gamma_{i1}\beta_{i2}. \tag{3.47}$$

This corresponds to linearly independent basis functions B_{i1}, B_{i2}, B_{i3}, because each triplet defines a basis function. □

Alternatively, we can show that the equation

$$a_1 B_{i1}(x, y) + a_2 B_{i2}(x, y) + a_3 B_{i3}(x, y) \equiv 0 \tag{3.48}$$

has only the trivial solution for unknowns a_1, a_2 and a_3. Plugging in three well chosen points gives a linear system of equations in a_1, a_2 and a_3. The determinant of this system has to be different from zero. For example, choosing the two PS-points S and S' and the vertex V_1 (Figure 2.6) in one of the triangles of the

molecule to evaluate Equation (3.48) leads to the following expression for the determinant

$$\frac{1}{4}\left(\beta_{11}\gamma_{12}-\gamma_{11}\beta_{12}\right)\left(x_{12}y_{13}-x_{13}y_{12}\right)\left(1-\lambda_{12}-\lambda_{13}+\lambda_{12}\lambda_{13}\right), \qquad (3.49)$$

where $x_{12} = x_2 - x_1$, $x_{13} = x_3 - x_1$, $y_{12} = y_2 - y_1$ and $y_{13} = y_3 - y_1$ and (x_i, y_i) are the cartesian coordinates of the vertex V_i. The numbers λ_{12} and λ_{13} defined in Equation (2.40) characterize the PS-refinement.

In this expression we recognize three factors:

1. The first factor, which we expected to see, is inversely proportional with the area of the PS-triangle. So the smaller the PS-triangle, the larger the expression for the determinant. When the area of the PS-triangle approaches towards infinity, the basis functions become linearly dependent.

2. The second factor depends on the triangulation and equals the dot product of the vectors $\overrightarrow{V_1V_2}$ and $\overrightarrow{V_1V_3}$. This is equivalent to the length of these vectors multiplied with the cosine of the angle between them. The appearance of this factor indicates that triangles with a small angle in a corner are less favorable. Also note that if the triangles of Δ become smaller, and thus the area of the PS-triangle is relatively large compared to the triangles of Δ, this factor will become smaller too.

3. The third factor is the influence of the PS-refinement. The numbers λ_{12} and λ_{13} have a value between 0 and 1. The higher these values are, the closer the PS-points are to vertex V_1. The area of the chosen PS-triangle is then relatively large and this is reflected by a higher value for the third factor of the determinant. On the other hand, when λ_{12} and λ_{13} have smaller values, the PS-points are further away from V_1 and the area of the chosen PS-triangle is closer to optimal. The third factor of the determinant is then larger.

3.3.2 The optimal solution

In the original construction of the B-splines of Dierckx [41], the PS-triangle with the smallest area is chosen. To minimize the area we have to maximize

$$\left|\beta_{i1}\gamma_{i2}-\gamma_{i1}\beta_{i2}\right| \;\rightarrow\; \max \qquad (3.50)$$

subject to the constraints that the PS-points must lie inside the PS-triangle. The barycentric coordinates of the PS-points with respect to the PS-triangle must be greater than or equal to zero. We have a quadratic optimization problem with linear inequality constraints.

3.3.3 A nearly optimal solution

In this section we describe a practical algorithm to find a suitable small PS-triangle that contains the PS-points without having to solve a small optimization problem for each vertex. The algorithm quickly yields a nearly optimal result.

When the molecule number m_i equals three, there is a trivial solution for the PS-triangle t_i. For vertices with a higher molecule number, we reduce the number of PS-points by taking them together in a certain way, until we have the trivial case again. We also give an example of a degenerate case in which the algorithm breaks down. The modification for the degenerate case can also be used in general and this yields the practical algorithm. We give the procedure in pseudo code. Finally we compare this nearly optimal PS-triangles with the optimal PS-triangles on the basis of an example.

Trivial case

When the molecule number m_i equals three, the six PS-points different from V_i form a triangle. This is the smallest triangle that contains the PS-points and thus it is an obvious choice for the PS-triangle t_i. No extra computation is needed to find this triangle. An example is shown in Figure 3.2.

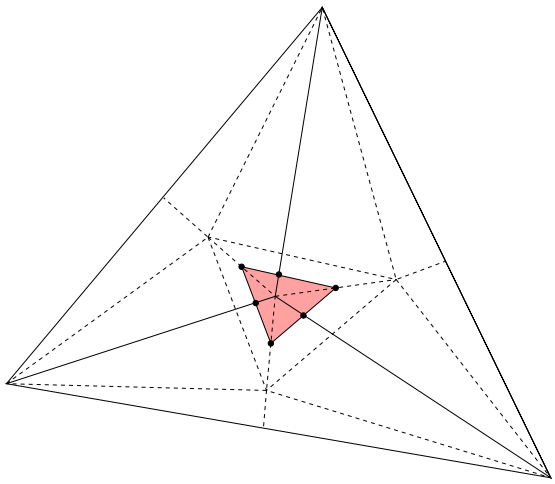


Figure 3.2: Trivial case: for molecule $m_i = 3$, the PS-points form a triangle.

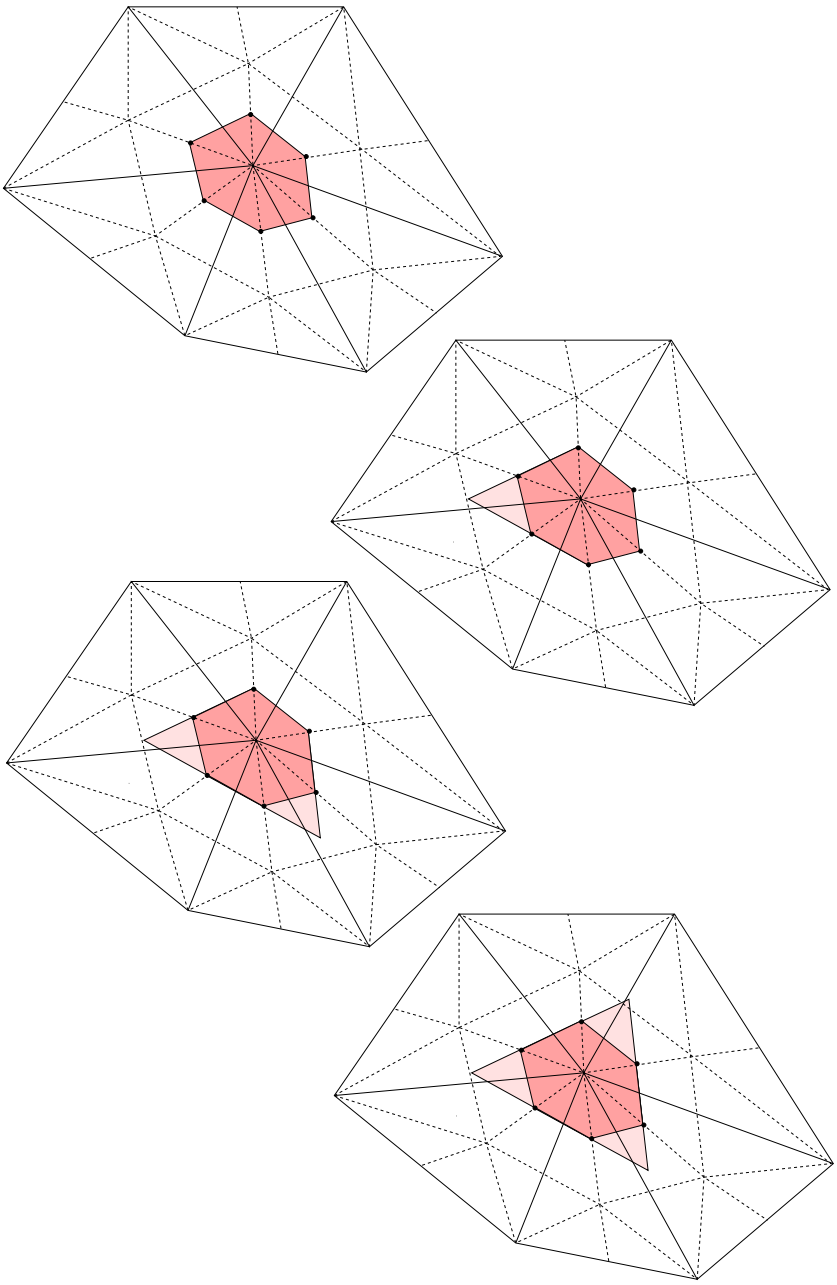


Figure 3.3: In each step we remove an edge from the polygon formed by the PS-points in such a way that the resulting triangle contains the original polygon and can be used as PS-triangle.

Reduction of other configurations to the trivial case

When there are more then three triangles in the molecule M_i, the PS-points do not form a triangle but a polygon with m_i corners. We now change the m_i-gon into a $(m_i - 1)$-gon that contains the m_i-gon. To do so we choose an edge of the original polygon and add a corner at the intersection of the neighboring edges. We drop the original edge and its two corners.

We repeat this procedure until we obtain a triangle that contains the original polygon and thus contains the PS-points of V_i. We use this triangle as the PS-triangle t_i. An example of the subsequent steps is shown in Figure 3.3. The middle vertex has six triangles in its molecule. In three steps we find a PS-triangle that contains all the PS-points.

Sequence of operations and degenerate case

The order in which the edges of the polygon are removed can be important. Figure 3.4 repeats the example of Figure 3.3, but another edge is chosen to be removed in the second step. This leads to a polygon with edges that are (almost) parallel. We cannot apply the above described procedure because the edges do not intersect or the intersection point lies too far to be useful. It is also possible that the polygon has two pairs of parallel edges already from the initial configuration. Figure 3.5 shows such a degenerate example.

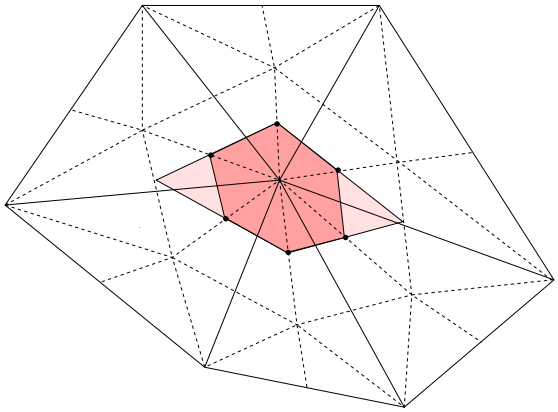


Figure 3.4: The order in which the edges of the polygon are removed is important.

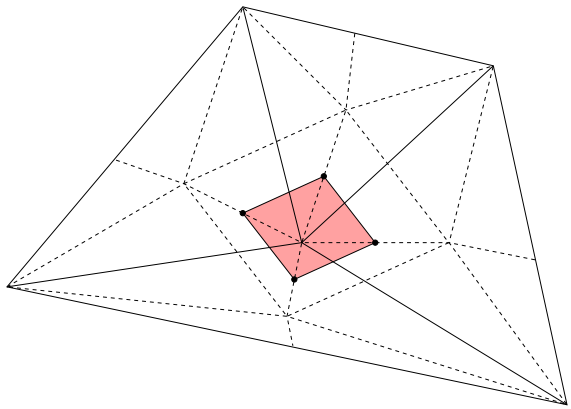


Figure 3.5: Degenerate case: the edges of the polygon are parallel, no intersection can be found.

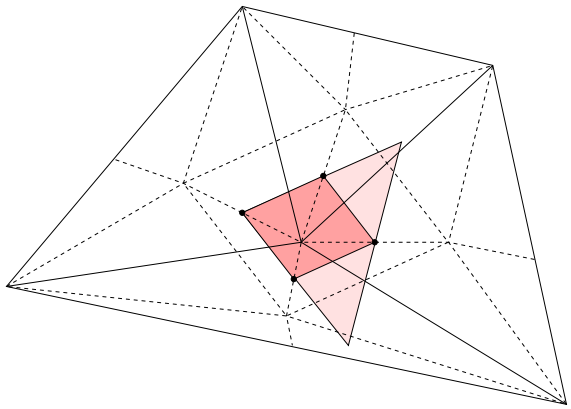


Figure 3.6: Practical solution. Only one corner of the PS-triangle is found as the intersection of the lines through two edges of the polygon.

Practical solution

In order to avoid the degenerate case we only compute one corner of the PS-triangle as the intersection of the lines through two edges of the polygon. In contrast to the first idea these two edges can be chosen arbitrary and can also be neighbors. The second and third corner of the PS-triangle are chosen somewhere on the lines through these two edges such that the whole polygon is inside the PS-triangle. This is illustrated in Figure 3.6.

To formulate a practical algorithm we need to specify how we choose the second and third PS-points, because there is not one unique solution. An approach that is easy to implement and that leads to sufficient results is to calculate a line through the PS-point farest a way from the first corner of the PS-triangle and perpendicular to the bisector of the two initial edges. The intersection of this line with the lines through the initial edges yields the second and third PS-point.

We repeat the described procedure systematically for all pairs of initial edges of the m_i-gon. Each time we also compute the area of the PS-triangle, and keep the smallest one so far.

The algorithm in pseudo-code:

```
A = 0
for i = 1 to n do
    L1 = line through p_i and p_i+1
    for j = i+1 to n do
        L2 = line through p_j and p_j+1
        if !(L1 // L2) do
           Q1 = intersection(L1,L2)
           P = line perpendicular to bisector of L1
              and L2 and through farest PS-point
           Q2 = intersection(L2,P)
           Q3 = intersection(P,L1)
        end if
        if Area < A or A = 0 do A = Area
    end for j
end for i
```

Boundary

At the boundary it can be necessary to use the vertex itself as a corner of the polygon. An example of this situation is shown in Figure 3.7 for the leftmost vertex. Indeed, if the vertex, which is also a PS-point, is not used it is not contained in the resulting PS-triangle. This adaptation of the algorithm is not needed when the vertex lies correctly inside the polygon as shown on the right of Figure 3.7.

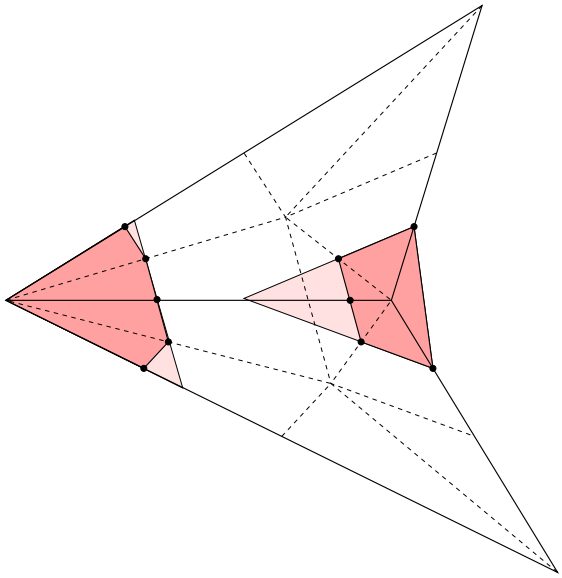


Figure 3.7: At the boundary of the domain it can be necessary to use the vertex also as a corner of the polygon.

Results

We applied both the optimal and the nearly optimal procedure to an example from [41]. The results are shown in Figure 3.8. The triangulation consists of 21 vertices and 31 triangles. The PS-triangles in the left picture are optimal and have the smallest possible area. The PS-triangles in the right picture are nearly optimal. The average area of the nearly optimal PS-triangles is 4.05% greater than the average

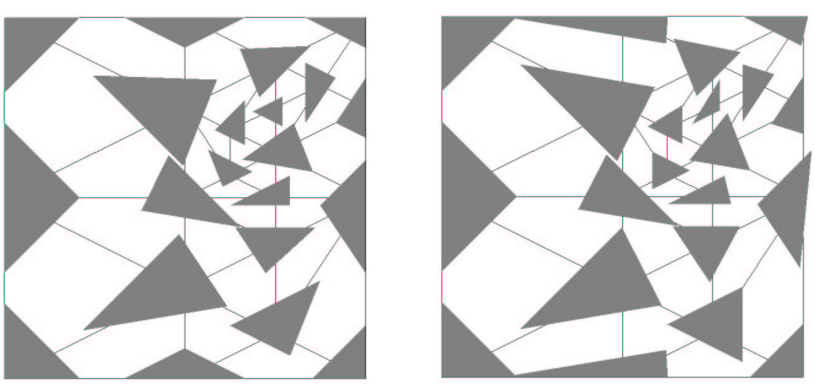


Figure 3.8: Left: Optimal PS-triangles. Right: Nearly optimal PS-triangles

area of the optimal triangles.

$$\frac{\sum_{i=1}^{21} Ar_{t_i'}}{\sum_{i=1}^{21} Ar_{t_i}} = 1.0405 \tag{3.51}$$

with t_i the optimal and t_i' the nearly optimal PS-triangles.

Of the 31 PS-triangles, four have exactly the same area, namely the three PS-triangles on the left boundary of the triangulation and the PS-triangle in the lower right corner. The largest differences occur for the PS-triangles in the middle of the uppermost and lowest edge of the triangulation with factors 1.2525 and 1.2526 respectively.

3.4 Concluding remarks

In this chapter we proved the stability of the B-spline basis. For the p-norm the basis functions should be normalized with respect to their support. An essential element is that the areas of the PS-triangles have to be bounded. Furthermore when the area of the PS-triangle goes to infinity the basis functions become linearly dependent. In the stability proofs we give estimates of the Riesz bounds. From these we can see that apart from large PS-triangles, small angles and unbalanced sizes of triangles in the triangulation should also be avoided.

Stable bases are of critical importance for both theoretical and practical purposes. In this thesis the stability results will be used for the design of wavelets

in Chapter 6. The first requirement to build a multiresolution pyramid is that the spline spaces are spanned by Riesz bases.

We also gave a practical algorithm to calculate small PS-triangles without solving optimization problems. We call the results nearly optimal, in contrast to the optimal solution, that is, the PS-triangles with the smallest area. The size of the nearly optimal triangles is on average 4.05% larger than the size of the optimal triangles.

Chapter 4

Multiresolution techniques

4.1 Introduction

Multiresolution techniques are concerned with the generation, representation and manipulation of geometric objects at different levels of detail. In this research field both mathematics and computer science play an important role. The standard approach to facilitate the handling of large amounts of data is to introduce hierarchical structures. Hierarchies usually provide fast access to relevant parts of a dataset which increases the efficiency of any algorithm processing the data. In the context of geometric objects hierarchical representations provide access to different resolutions of the underlying surface.

The representation of an object on several levels of detail gives also rise to other applications that exploit the hierarchical nature of the representation and include fast visualization and rendering, coding, compression, digital transmission and others. Multiresolution techniques use the additional information that comes available through a level of detail representation either by choosing an appropriate resolution for a given quality or complexity, or by considering the difference between two levels of detail as a separate frequency band which contains the detail information that is added or removed when switching between hierarchy levels.

In this chapter we give an overview of the multiresolution techniques that are used in this thesis. We avoid going into deep detail and only cover the aspects that are relevant for the remainder of the text.

4.2 Subdivision

Subdivision is a powerful mechanism for the construction of smooth curves and surfaces. The main idea is to start with a set of initial coefficients $\mathbf{s}^0$ which live on the coarsest domain grid and then iterate upsampling, that is inserting new vertices in the grid, and local averaging on the corresponding coefficients to build complex geometrical shapes.

4.2.1 Definitions

Definition 4.1 *A subdivision scheme computes iteratively coefficients $\mathbf{s}^{j+1}$ on finer grids as linear combinations of coefficients $\mathbf{s}^j$ on a coarser grid*

$$\mathbf{s}^{j+1} = P^j \mathbf{s}^j. \tag{4.1}$$

In this equation the $\mathbf{s}^j$ are column vectors, and the linear combinations on each level are described by the subdivision matrices P^j. The vector $\mathbf{s}^{j+1}$ is longer than the vector $\mathbf{s}^j$ because it lives on a finer grid and contains more coefficients.

The first uses of subdivision schemes were in the context of corner cutting [13] and for building piecewise polynomial curves. For example the de Casteljau algorithm for Bernstein-Bezier curves [36] and algorithms for the iterative construction of splines [34, 74]. Spline subdivision or knot insertion algorithms [8, 20, 81, 93] represent the limit curve in terms of a larger set of control points defined on a finer domain grid. Later subdivision was studied independently of spline functions [46, 45, 40, 10, 11, 12].

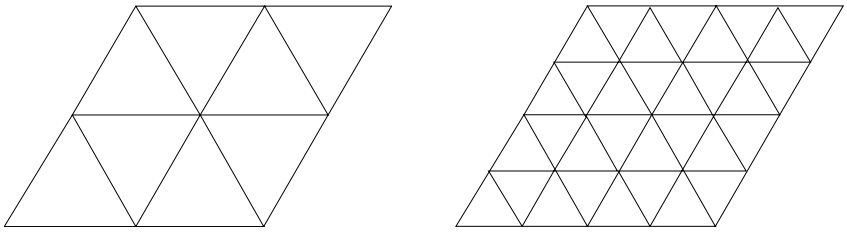


Figure 4.1: A regular domain grid. Uniform subdivision uses the same weights within and across levels.

A grid is called regular if the vertices in the domain are equally spaced and the intervals or faces all have the same size. The uniform subdivision scheme that corresponds with a regular grid uses the same weights within a level and across all levels. An example is given in Figure 4.1.

A semi-regular grid starts with an irregular coarse grid and adds new points at parameter locations regularly spaced between the old points. The example in Figure 4.2 is dyadic, the new points are midway between the old points. The weights of the semi-uniform subdivision scheme vary within each level and take into account the local neighborhood structure of a vertex. Such schemes are often referred to as stationary because the same rules can be used for each level.

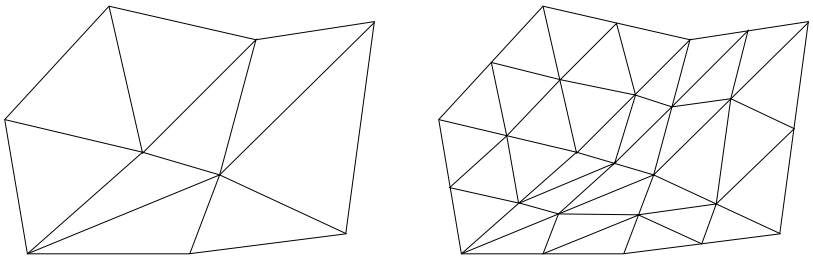


Figure 4.2: A semi-regular domain grid. The weights of semi-uniform subdivision can vary within in a level.

In an irregular grid the new vertices need not to be balanced with respect to the old ones. Non uniform subdivision schemes use varying weights within and across levels. Also there is not necessarily a new vertex for each edge or face, the topology can also be irregular. For example, not every triangle has to be split in exactly four new triangles. An example of a such a subdivision scheme is given in [26], it computes one new coefficient at the time. We do not consider the latter here and only work with irregular grids and non uniform subdivision with a regular topology. This is called subdivision connectivity and illustrated in Figure 4.3.

The valence of a vertex in a grid is the number of faces it belongs to. For example in a regular triangular grid, all vertices have valence six. A vertex in a semi-regular of irregular grid is called extraordinary if it has a valence different from the standard valence. The mask or stencil of a vertex is the set of surrounding vertices that are used to compute the new value.

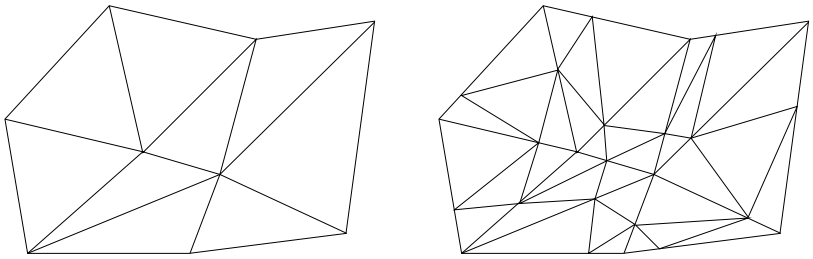


Figure 4.3: An irregular domain grid with subdivision connectivity. The weights of non uniform subdivision vary within and across levels.

The key point about subdivision as a surface definition technique is that if we forget about the underlying representation such as a spline representation, we get an enormous amount of additional freedom. We can use the refinement rules described by the linear combinations in P^j on three-dimensional control points and consider the subdivision rules as the way the surface is defined and interrogated. For example, although subdivision rules are often designed starting from the regular setting, we can still carry out a refinement operation on semi-regular three-dimensional meshes by using a special pattern in the neighborhood of the extraordinary points. The semi-regular setting is standard in most schemes described in the literature.

When building splines, the limit functions are piecewise polynomial, and there is no need to distinguish between the semiregular and the irregular case. Both cases are commonly referred to as non uniform. The smoothness of the result of spline subdivision is obvious because the analytic form of the polynomial pieces of the limit function is known. However, when the limit function is not a spline, convergence and smoothness are usually harder to prove.

We say that a subdivision scheme is interpolating if in each subdivision step the values at the old vertices are kept. The limiting function thus interpolates the original data on the coarsest grid. If the old values change each step, we call the scheme approximating. This is typically the case for spline based subdivision.

4.2.2 Multiresolution analysis

An important motivation for the study of subdivision algorithms is their relation to multiresolution analysis and wavelet bases. We can associate basis functions with a subdivision scheme:

Definition 4.2 *The scaling function ϕ_k^j is defined as the limit function if we start the subdivision scheme on level j with an impulse δ_k.*

The set of coefficients at level j are all zeroes except at one vertex where we have a 1 and we start the iterations from there. On level $j+1$ we have more scaling functions than on level j.

If the subdivision scheme is local in the sense that each coefficient on a finer level is a linear combination of only a limited number of coefficients on the coarser level, then the scaling functions are compactly supported. The limit function of a subdivision scheme started on level j can be written as a linear combination of scaling functions on level $j+1$.

Denote by Φ^j a row vector with the scaling functions ϕ_k^j on level j.

Theorem 4.1 *The scaling functions satisfy a refinement equation*

$$\Phi^j = \Phi^{j+1} P^j. \tag{4.2}$$

This follows immediately from the combination of Definition 4.1 and 4.2. It establishes refinebility since it states that each of the functions in Φ^j can be written as a linear combination of the functions in Φ^{j+1}.

Given this relation, a strictly increasing sequence of subspaces V^j can be associated with the initial coarse grid:

$$V^j = span\{\phi_k^j\} \tag{4.3}$$

and

$$V_0 \subset V^1 \subset V^2 \ldots \tag{4.4}$$

This is called a multiresolution analysis (MRA).

Definition 4.3 *The polynomial order of a MRA is the largest number $\tilde{N}$ for which $\Pi_{\tilde{N}-1}$, the space of polynomials of degree at most $\tilde{N}-1$, is contained in V^j*

$$\tilde{N} = \max\{n \mid \Pi_{n-1} \subset V^j\}. \tag{4.5}$$

The polynomial order is also sometimes called the number of dual vanishing moments.

4.3 Wavelets

The subdivision technique builds a smooth function on a fine grid starting from a description on a coarse grid. In the wavelet setting we want to start from a fine grid that is gradually coarsified. Subdivision then gives an approximation of the original data by extrapolating the reduced data on the coarser grid back to the original finer grid. Now subdivision is not used to refine the finest grid even more, but to build a multiresolution analysis on coarser grids. We are in the case of irregular grids and non uniform subdivision because the new vertices in which subdivision calculates a value are dictated by the already existing finer grid.

4.3.1 Filter bank algorithm

Equations (4.1) and (4.2) define an identity map from the coarse space V^j to the fine space V^{j+1}. Obviously this map is not surjective since V^{j+1} contains functions that are not in V^j. The difference or the additional details can be captured in in a complementary space W^j such that

$$V^j \oplus W^j = V^{j+1}. \tag{4.6}$$

There should be a set of functions that form a basis for W^j

$$W^j = span\{\psi_k^j\}. \tag{4.7}$$

The ψ_k^j are called wavelet functions. We collect them in a row vector Ψ. The complement space W^j is not necessarily orthogonal to V^j as long as we have a direct sum. In practice it is often required that W^j is orthogonal to some low degree polynomials that are contained in V^j. The maximum order of these polynomials is called the number of primal vanishing moments.

Definition 4.4 *The wavelets ψ_k^j are said to have N primal vanishing moments if the complement space W^j is orthogonal to Π_{N-1}.*

Every function in V^{j+1} can be written as a part in V^j and a part in W^j

$$\Phi^{j+1}\mathbf{s}^{j+1} = \Phi^j\mathbf{s}^j + \Psi^j\mathbf{w}^j. \tag{4.8}$$

The scaling functions and wavelet functions are typically designed such that V^j captures the low frequency component of V^{j+1} and W^j captures the high frequency parts. When the space W^j and its basis functions ψ_k^j are carefully chosen, $\mathbf{s}^j$ gives a coarse scale approximation of $\mathbf{s}^{j+1}$ and $\mathbf{w}^j$ describes the difference or

detail information. The combination of $\mathbf{s}^j$ and $\mathbf{w}^j$ should be a sparser representation than $\mathbf{s}^{j+1}$ or in other words the change of basis from Φ^{j+1} to Φ^j combined with Ψ^j should be decorrelating.

The inclusion $W^j \subset V^{j+1}$ links the basis of W^j to the basis of V^{j+1}. Since the wavelet functions ψ_k^j also lie in the space V^{j+1} there must exist a linear combination that satisfies

$$\Psi^j = \Phi^{j+1} Q^j. \tag{4.9}$$

Combining this equation with the refinement relation (4.2) of the scaling functions, we find that the coefficients on level $j+1$ can be reconstructed from the coarse scale approximation and the detail information

$$\mathbf{s}^{j+1} = P^j \mathbf{s}^j + Q^j \mathbf{w}^j. \tag{4.10}$$

This is called the reconstruction, inverse wavelet transform or synthesis.

The inverse of the reconstruction operation is called the decomposition, wavelet transform or analysis. This is done with two filters, a low pass filter A^j and a high pass filter B^j

$$\begin{aligned} \mathbf{s}^j &= A^j \mathbf{s}^{j+1} \\ \mathbf{w}^j &= B^j \mathbf{s}^{j+1}. \end{aligned} \tag{4.11}$$

The filter operations are represented schematically in Figure 4.4.

The basis Φ^{j+1} is linked to the combined basis of Φ^j and Ψ^j by the necessarily non singular change of basis matrix $M^j = [P^j \; Q^j]$,

$$[\Phi^j \; \Psi^j] = \; \Phi^{j+1} \; [P^j \; Q^j] = \; \Phi^{j+1} \; M^j, \tag{4.12}$$

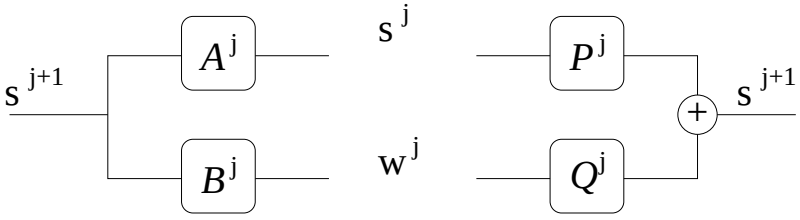


Figure 4.4: Filter bank algorithm.

and we can easily see that the analysis filters must be defined by the inverse relation

$$M^{-j} = \begin{bmatrix} A^j \\ B^j \end{bmatrix}^{-1} = \begin{bmatrix} P^j & Q^j \end{bmatrix} = M^j. \tag{4.13}$$

The decomposition procedure can now be repeated on the low pass coefficients $\mathbf{s}^j$ to yield $\mathbf{s}^{j-1}$ and $\mathbf{w}^{j-1}$. This can be repeated until level 0. Note that M^j can be different from level to level since we consider the non uniform setting. The matrix T^n that represents the inverse transform from the multiscale representation to the fine scale data on level n is

$$\mathbf{s}^n = T^n \begin{bmatrix} \mathbf{s}^0 \\ \mathbf{w}^0 \\ \vdots \\ \mathbf{w}^{n-1} \\ \mathbf{w}^n \end{bmatrix}, \tag{4.14}$$

$$T^n = \begin{bmatrix} M^n & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} M^{n-1} & 0 \\ 0 & I \end{bmatrix} \cdots \begin{bmatrix} M^0 & 0 \\ 0 & I \end{bmatrix}. \tag{4.15}$$

In these equations 0 stands for the zero matrix, and I for the identity matrix.

Decomposition and reconstruction are inverse processes, so that the original information can be reconstructed from its components, at least theoretically, when not taking into account any round off errors in the computations. In practical applications the approach is off course different. The coefficients obtained by the decomposition are filtered in some way, following prescribed criteria, that reflect certain goals like for example data compression, noise removal or edge detection. The original function is then not reconstructed exactly but approximated.

4.3.2 Lifting scheme

For the practical execution of decomposition and reconstruction we need not so much the underlying basis functions themselves, but the coefficients in the expansions together with the decomposition and reconstruction matrices. The construction of a basis for the complement space W^j thus typically means the determination of a matrix Q^j that describes how the wavelets Ψ^j can be expressed in terms of the higher level scaling functions Φ^j. Preferably the matrices P^j and Q^j have a simple structure. Then reconstruction means matrix multiplication and the decomposition is carried out by treating Equation (4.10) as a linear system of equations with the $\mathbf{s}^{j+1}$ as the known right hand side and the $\mathbf{s}^j$ and $\mathbf{w}^j$ as the unknowns.

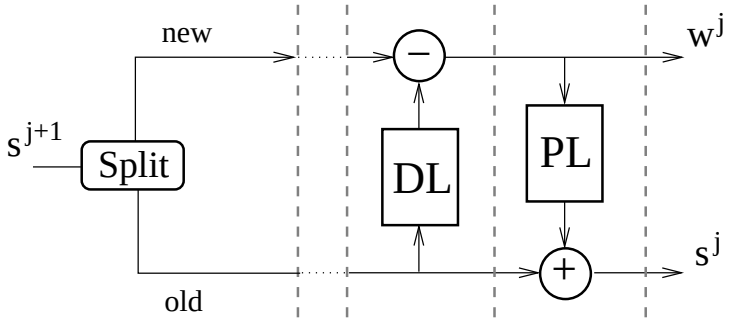


Figure 4.5: Decomposition of a filter bank into lifting steps. The first type of lifting is called dual lifting or prediction. The other type is primal lifting or update.

However, we wish to avoid building the matrix Q^j explicitly and solving a linear system of equations. Instead we use the lifting scheme technique. The lifting scheme decomposes the filter bank operation of Figure 4.4 in consecutive lifting steps [27, 113, 112, 111]. The main difference with the classical construction is that it does not rely on the Fourier transform but has an entirely spatial interpretation. The most important property of lifting is its generality. All classical wavelet transforms can be implemented using the lifting scheme. However, the lifting steps are by no means limited to the classical linear filter operations or regular settings and therefore lifting is an excellent tool to design wavelets on non uniform grids.

The basic idea is very simple. It starts with a trivial wavelet, the Lazy wavelet, that corresponds to just splitting the signal in 'old' points and 'new' points. We will use the terms old and new points in analogy with the hierarchy induced by subdivision. The lifting scheme then gradually builds a new wavelet with improved properties by combining and decorrelating the two sets. The building blocks are lifting steps (Figure 4.5).

Dual lifting substracts a filtered version of the old samples from the new samples. Primal lifting adds a filtered version of the dual lifting output to the so far untouched old samples. Dual and primal lifting are often called prediction and update.

Before prediction the old and new samples are highly correlated. We then try to predict the new samples with a filter on the old samples. Substracting this prediction from the new samples reduces the correlation. The differences are the detail

coefficients, the high pass information. It is this step that exploits the correlation between the two sets of points. The prediction formula can for example be an interpolating polynomial or a subdivision algorithm. The update step or primary lifting step can be interpreted as a way to preserve the average and higher moments in the low pass coefficients.

Inverting a lifted transform is straightforward: run through the scheme backwards replacing plus-signs with minus-signs, and merge what had been split. Unlike the classical filter bank setup, the same filters appear in the forward and inverse transforms and no inverse filters are needed. The inverse diagram is shown in Figure 4.6.

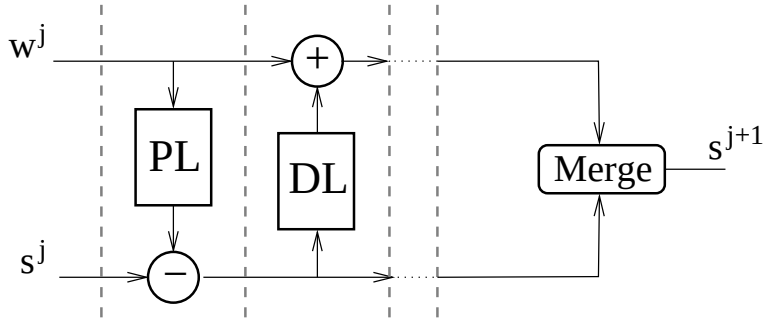


Figure 4.6: The inverse lifting scheme: run through the lifting scheme backwards, replacing plus-signs with minus-signs and merge what had been split.

4.4 Subdivision connectivity

A limitation of the subdivision based multiresolution representation is that it cannot be applied to arbitrary meshes. Because the decomposition is constructed for a prescribed operator, it only applies to the special output of that operator. This specific connectivity is called subdivision connectivity. Such meshes consist of patches with regular mesh structure and extraordinary points that do not have regular connectivity only occur at the corners of these patches. The extraordinary points correspond to the coarsest level grid, and all points introduced by the subdivision operator have regular connectivity.

The mathematical reason of subdivision connectivity for a multiresolution analysis lies in the need for a nested set of approximation spaces. Indeed, the condition $V^j \subset V^{j+}$ implies that the support of the each scaling function at one level of resolution must be the union of the supports of some finer scaling functions.

This type of hierarchy is sometimes called coarse-to-fine hierarchy because the structure of the meshes on the different levels is determined by the coarsest level which can be arbitrary. If a surface is given by an unstructured triangle mesh which does not have subdivision connectivity, remeshing techniques [48, 68, 75] have to be applied which resample the original surface to generate a subdivision connectivity mesh that approximates the original geometry.

Alternatively one can start with some decomposition operator, e.g mesh decimation algorithms [16] building the hierarchy from fine to coarse and then derive the matching reconstruction operator [63, 67, 57]. We do not consider these approaches in this dissertation.

4.5 Concluding Remarks

Subdivision is the main ingredient of the multiresolution techniques discussed in this chapter. In the case of spline subdivision the aim is to calculate a representation of the known spline function on a refinement of the domain. Subdivision is not used as a surface definition or construction method, since the limit surface is simply the spline surface defined by the initial control points. This is the topic of Chapter 5.

On the other hand, in Chapter 7, we use subdivision as a procedure to construct smooth surfaces. The surface is defined by the initial control points and the set of refinement rules. By doing so we can represent surfaces of arbitrary topological type.

The multiresolution analysis based on the spline subdivision algorithm will be used in Chapter 6 as the starting point to design wavelets. The complement space captures the difference between two successive spline spaces. We use the lifting paradigm for the design and the implementation.

Chapter 5

Spline Subdivision

5.1 Introduction

The goal of spline subdivision is to calculate the B-spline representation (2.24) of a PS-spline surface on a refinement Δ^1 of the given triangulation Δ^0. The new basis functions after subdivision have smaller support and give the designer more local control when manipulating surfaces. Adjusting one coefficient c_{ij} of a control point of a subdivided PS-spline surface, influences a smaller neighborhood of the involved vertex. The control triangles are tangent to the surface and in case of repeated subdivision, the linear interpolant of the tangent points converges to the surface itself. Therefore subdivision is a common technique for displaying surfaces graphically.

In Section 5.2 we discuss dyadic subdivision. This type of refinement scheme was already developed for uniform Powell-Sabin splines [125]. Because we will use these rules in Chapter 7, we briefly recall them here. The remainder of the chapter is original material. We show that dyadic refinement is not generally applicable in the non uniform case.

Therefore we switch to another scheme in Section 5.3, the $\sqrt{3}$ scheme. This refinement always exists. We give a proof for the new PS-triangles and control triangles based on the underlying Bézier representation.

Applying the $\sqrt{3}$ scheme twice yields a triadic scheme. This is the topic of Section 5.4. At the boundaries of the domain we need slightly different rules, namely in the second $\sqrt{3}$ step we have to introduce two new vertices instead of one on each boundary edge. Finally we propose an optimization that makes the PS-triangles smaller.

5.2 Dyadic Subdivision

The most obvious possibility for subdivision is dyadic subdivision. In this scheme a new vertex is inserted on every edge between two old vertices and every original triangle is split into four new triangles. This is illustrated in Figure 5.1. It shows one original triangle (V_i, V_j, V_k) on level 0. This triangle is split into six smaller triangles with a common vertex Z_{ijk}.

When choosing the positions of the new vertices on level 1, the lines of the PS-refinement Δ^{0*} of the initial triangulation Δ^0 must be maintained in the PS-refinement Δ^{1*} of the refined triangulation Δ^1. The aim is to represent exactly the same spline surface and so the lines of C^2 discontinuity may not change. To ensure this, the new vertices have to be placed on the intersections R_{ij} of the PS-refinement with the edges. We denote the new vertices with V_{ij}, V_{jk} and V_{ki}.

For the new triangle (V_{ij}, V_{jk}, V_{ki}) the PS-refinement is already fixed, the point Z_{ijk} is also the interior point for this new triangle. The interior points for the three new remaining triangles have to lie on the line of the old PS-refinement Δ^{0*} that crosses the new triangle.

5.2.1 Uniform Powel-Sabin splines

The dyadic scheme was used by Windmolders and Dierckx for uniform Powell–Sabin splines, this is on a regular triangulation with all equilateral triangles. The subdivision rules for this special case can be found in [125].

Due to the high degree of symmetry in a uniform triangulation, the PS-refinement is trivial and the form of the PS-triangles can be fixed. The PS-refinement (Figure 5.2) is constructed by drawing the bisector of each angle. Only a limited number of molecules are possible in convex domains, namely with molecule numbers one, two, three and six. Their optimal PS-triangles are shown in Figure 5.3(a). For molecule number six, two optimal solutions exist. In the sequel we use the one shown in the figure. We can see that the optimal PS-triangle for molecule number six is also a valid, though not optimal, choice for the smaller molecule number, because it contains the appropriate PS-points (Figure 5.3(b)). We use this PS-triangle for every vertex. All PS-triangles are fixed and known in advance, this facilitates the calculations with uniform Powell-Sabin splines considerably.

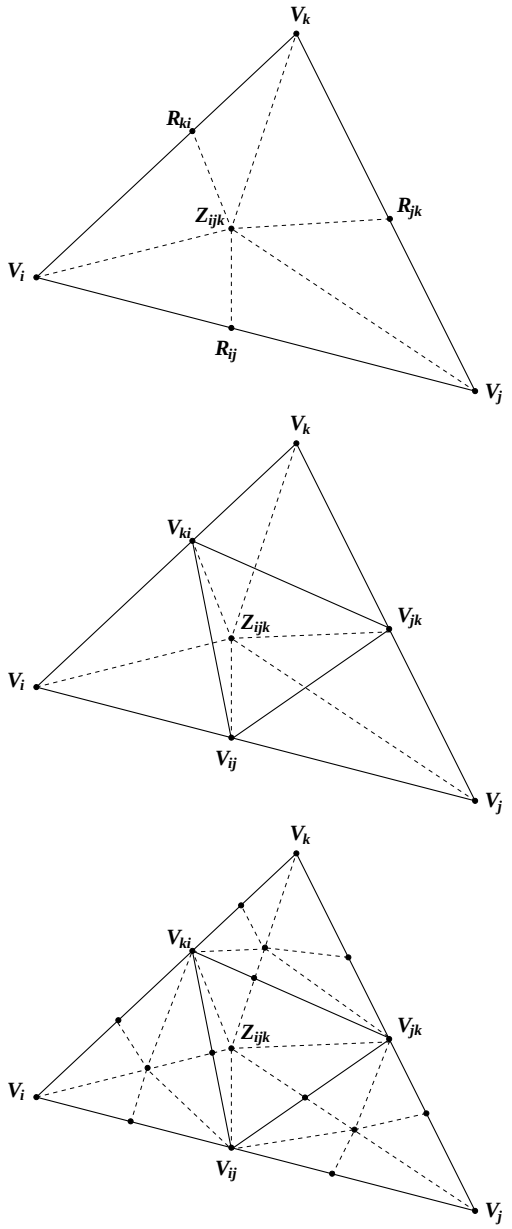


Figure 5.1: Positions of the new vertices and PS-refinement in the new triangles for dyadic subdivision.

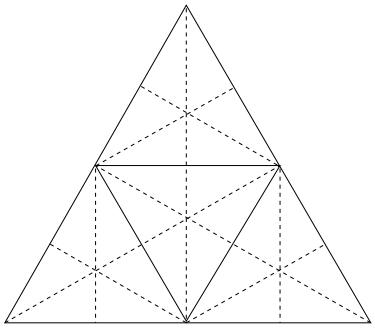


Figure 5.2: The PS-refinement of a uniform triangulation can be found by drawing the bisector of each angle.

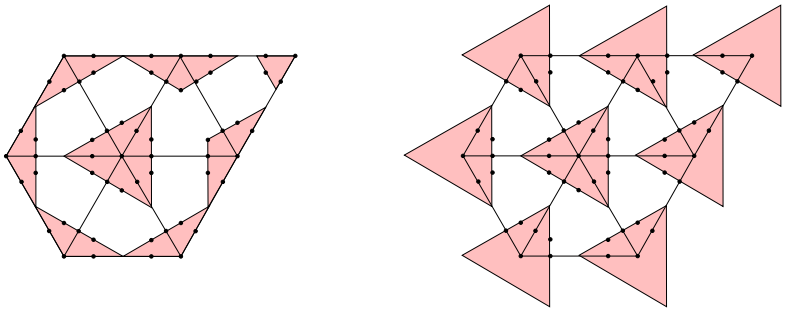


Figure 5.3: (a) The optimal PS-triangles for molecule numbers one, two, three and six. (b) A fixed form for the PS-triangles.

There are two types of subdivision rules, one for the old vertices and one for the newly introduced vertices.

Theorem 5.1 *The subdivision rule for an old vertex $V_i \in \Delta^0$ (Figure 5.4) is*

$$\boldsymbol{c}_{i1}^1 = \frac{2}{3}\boldsymbol{c}_{i1}^0 + \frac{1}{6}\left(\boldsymbol{c}_{i2}^0 + \boldsymbol{c}_{i3}^0\right), \tag{5.1}$$

$$\boldsymbol{c}_{i2}^1 = \frac{2}{3}\boldsymbol{c}_{i2}^0 + \frac{1}{6}\left(\boldsymbol{c}_{i3}^0 + \boldsymbol{c}_{i1}^0\right), \tag{5.2}$$

$$\boldsymbol{c}_{i3}^1 = \frac{2}{3}\boldsymbol{c}_{i3}^0 + \frac{1}{6}\left(\boldsymbol{c}_{i1}^0 + \boldsymbol{c}_{i2}^0\right). \tag{5.3}$$

The subdivision rule for a new vertex $V_{ij} \in \Delta^1 \setminus \Delta^0$ on an edge $V_iV_j \in \Delta^0$ (Figure 5.5) is

$$\boldsymbol{c}_{ij,1}^1 = \frac{1}{2}\left(\boldsymbol{c}_{i2}^0 + \boldsymbol{c}_{i3}^0\right), \tag{5.4}$$

$$\boldsymbol{c}_{ij,2}^1 = \frac{1}{2}\boldsymbol{c}_{j1}^0 + \frac{1}{4}\left(\boldsymbol{c}_{i2}^0 + \boldsymbol{c}_{j2}^0\right), \tag{5.5}$$

$$\boldsymbol{c}_{ij,3}^1 = \frac{1}{2}\boldsymbol{c}_{j1}^0 + \frac{1}{4}\left(\boldsymbol{c}_{i3}^0 + \boldsymbol{c}_{j3}^0\right). \tag{5.6}$$

Proof: see [125].

The new control triangle T_i^1 for the old vertex V_i is coplanar with the old control triangle T_i^0, the tangent plane and tangent point remain the same. Also the control triangle T_{ij}^1 for the new vertex V_{ij} is tangent to the surface. All the new PS-triangles have the same fixed form and the PS-refinement is trivial. The refined triangulation is immediately in the required form for a next subdivision step.

5.2.2 Conditions on initial triangulation

When the initial triangulation is not regular, the topology and connectivity is arbitrary and the triangles can have any size, the idea of a dyadic refinement can only be used under certain conditions. For example in Figure 5.1, the interior point Z_{ijk} of the PS-refinement of the original triangle, must lie inside the middle new triangle (V_{ij}, V_{jk}, V_{ki}).

Also, if we want to be able to use similar uniform formulas as in Theorem 5.1 the new configuration around an old vertex must be a scaled version of the old configuration. Therefore the new interior point in the new triangle (V_i, V_{ij}, V_{ki}) must be chosen in the old PS-point on the line V_iZ_{ijk}, and this is only possible if this PS-point lies inside (V_i, V_{ij}, V_{ki}).

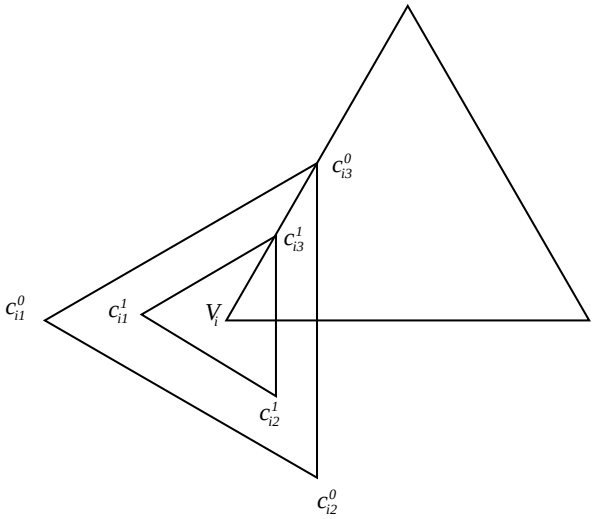


Figure 5.4: Subdivision of uniform Powell-Sabin splines at an old vertex $V_i \in \Delta^0$

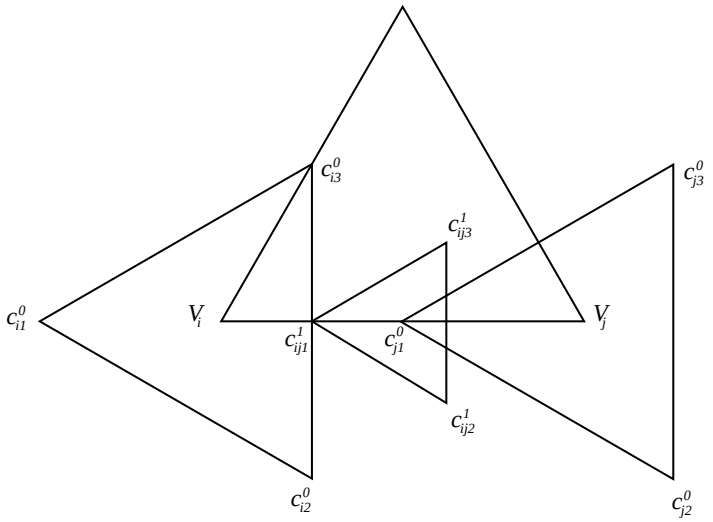


Figure 5.5: Subdivision of uniform Powell-Sabin splines at a new vertex $V_{ij} \in \Delta^1 \setminus \Delta^0$

These requirements lead to conditions on the initial triangulation Δ^0 and its PS-refinement Δ^{0*}, i.e. on the placement of the interior points Z_{ijk} and the resulting positions of the R_{ij}.

Recall from (2.40)

$$\begin{aligned} R_{ij} &= \lambda_{ij}V_i + (1-\lambda_{ij})V_j \\ R_{jk} &= \lambda_{jk}V_j + (1-\lambda_{jk})V_k \\ R_{ki} &= \lambda_{ki}V_k + (1-\lambda_{ki})V_i \\ Z_{ijk} &= a_{ijk}V_i + b_{ijk}V_j + c_{ijk}V_k. \end{aligned} \tag{5.7}$$

Theorem 5.2 *A uniform dyadic refinement procedure can only be used on irregular Powell-Sabin triangulations if and only if the following conditions are fulfilled*

$$\frac{1}{2} < \frac{(1-\lambda_{ij})\lambda_{ki}}{b_{ijk}\lambda_{ki} + c_{ijk}(1-\lambda_{ij})} < 1 \tag{5.8}$$

$$\frac{1}{2} < \frac{(1-\lambda_{jk})\lambda_{ij}}{a_{ijk}(1-\lambda_{jk}) + c_{ijk}\lambda_{ij}} < 1 \tag{5.9}$$

$$\frac{1}{2} < \frac{\lambda_{jk}(1-\lambda_{ki})}{a_{ijk}\lambda_{jk} + b_{ijk}(1-\lambda_{ki})} < 1. \tag{5.10}$$

Proof: The first pair of inequalities can be obtained by requesting that the intersection point of the line $V_{ij}V_{ki}$ and the line V_iZ_{ijk} lies between Z_{ijk} and the PS-point on V_iZ_{ijk}. The second and third pair of inequalities are found analogously. □

If the upper bounds are not satisfied the refined triangulation and its PS-split cannot be used to represent exactly the same surface as before subdivision. A dyadic refinement procedure is not applicable at all in that case. If the lower bounds are not satisfied it is still possible to resort to non uniform subdivision rules that can vary for each vertex and for each level.

Because the upper bounds are not generally fulfilled, the dyadic scheme is not generally applicable. Therefore we do not further consider the dyadic scheme for irregular triangulations and look for alternative subdivision schemes.

5.3 $\sqrt{3}$ subdivision

Another possibility is $\sqrt{3}$ subdivision. This kind of scheme was first introduced by Kobbelt [69] and Labsik and Greiner [71]. Instead of splitting each edge and performing a 1-to-4 split for each triangle, we compute a new vertex for each triangle and retriangulate the old and new vertices. A $\sqrt{3}$ scheme performs slower topological refinement, the number of triangles only triples each step. We have a finer gradation of hierarchy levels.

5.3.1 Refined triangulation

The new triangulation $\Delta^{\sqrt{3}}$ is constructed by inserting a new vertex V_{ijk} at the position of the interior point Z_{ijk} of each triangle. Except at the boundaries, the old edges are not preserved in the new triangulation. Instead new edges are introduced connecting every new vertex V_{ijk} with the vertices of the old triangle it lies in, and connecting every two new vertices that lie in neighboring old triangles. Figure 5.6 shows the result of $\sqrt{3}$ subdivision on an example triangulation. In this figure the PS-refinement is not shown, but note that the new edges in $\Delta^{\sqrt{3}}$ must coincide with the lines of the PS-refinement Δ^{0*} and that the original edges of Δ^0 are now part of the new PS-refinement $\Delta^{\sqrt{3}*}$.

Figure 5.7 shows a detail of one original triangle after $\sqrt{3}$ subdivision. In the new triangles new interior points must be chosen on the one line of the new PS-refinement $\Delta^{\sqrt{3}*}$ that is already fixed, that is, the original edge that crosses the triangle. For example

$$\begin{aligned} Z_{ij} &= \omega_{ij} V_i + (1 - \omega_{ij}) R_{ij}, \\ Z_{ji} &= \omega_{ji} V_j + (1 - \omega_{ji}) R_{ij}. \end{aligned} \tag{5.11}$$

For the resulting refinement to exist, the new vertex V_{ijk} has to lie inside the hexagon formed by the interior points $(Z_{ij}, Z_{ji}, Z_{jk}, Z_{kj}, Z_{ki}, Z_{ik})$. It is always possible to place these interior points, i.e. choose a value for ω_{ij} and ω_{ji} in equation (5.11), such that this condition is fulfilled: there are no conditions on the initial triangulation Δ^0 or its PS-refinement Δ^{0*}. Therefore we prefer to use the $\sqrt{3}$ scheme instead of the dyadic scheme.

In a practical implementation we use $\omega_{ij} = 1/3$ because this leads to the expected results in the uniform case when all triangles are equilateral and have the same size. We check whether the result satisfies the hexagon requirement and in the occasional case that is does not, we decrease the value incrementally, e.g. by dividing $1 - \omega_{ij}$ by a factor 2, until it yields a valid PS-refinement.

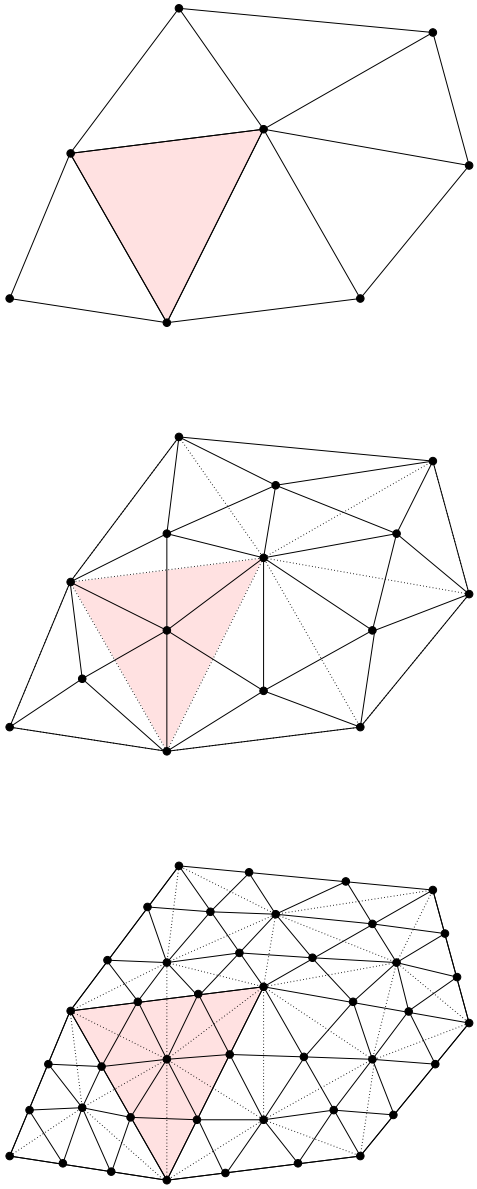


Figure 5.6: Principle of $\sqrt{3}$ subdivision. The PS-refinements are not shown. Applying $\sqrt{3}$ subdivision twice results in triadic subdivision.

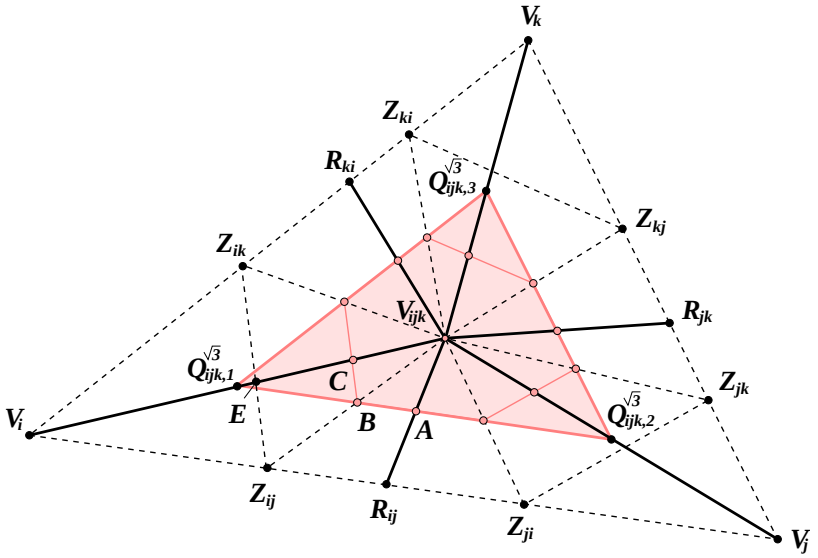


Figure 5.7: $\sqrt{3}$ subdivision: $t^{\sqrt{3}}_{ijk}$ is a valid PS-triangle for V_{ijk}. A valid choice for the new interior points and the PS-refinement always exist, there are no conditions on the initial triangulation.

5.3.2 The triangles $t_{ijk}^{\sqrt{3}}$ and $T_{ijk}^{\sqrt{3}}$

Consider the triangle $T_{ijk}^{\sqrt{3}}$ formed by the Bézier control points corresponding to the Bézier ordinates w_i, w_j and w_k in Figure 2.8. We denote its corners by $\boldsymbol{c}_{ijk,1}^{\sqrt{3}}$, $\boldsymbol{c}_{ijk,2}^{\sqrt{3}}$ and $\boldsymbol{c}_{ijk,3}^{\sqrt{3}}$. The projection of $T_{ijk}^{\sqrt{3}}$ in the (x,y)-plane is denoted by $t_{ijk}^{\sqrt{3}}$ with corners $Q_{ijk,1}^{\sqrt{3}}$, $Q_{ijk,2}^{\sqrt{3}}$ and $Q_{ijk,3}^{\sqrt{3}}$.

Theorem 5.3 *The triangle $T_{ijk}^{\sqrt{3}}$ is a valid control triangle for the new vertex V_{ijk}. The B-spline coefficients are*

$$\begin{aligned}
\boldsymbol{c}_{ijk,1}^{\sqrt{3}} &= \tilde{L}_{i1}\boldsymbol{c}_{i1} + \tilde{L}_{i2}\boldsymbol{c}_{i2} + \tilde{L}_{i3}\boldsymbol{c}_{i3} \\
\boldsymbol{c}_{ijk,2}^{\sqrt{3}} &= \tilde{L}_{j1}\boldsymbol{c}_{j1} + \tilde{L}_{j2}\boldsymbol{c}_{j2} + \tilde{L}_{j3}\boldsymbol{c}_{j3} \\
\boldsymbol{c}_{ijk,3}^{\sqrt{3}} &= \tilde{L}_{k1}\boldsymbol{c}_{k1} + \tilde{L}_{k2}\boldsymbol{c}_{k2} + \tilde{L}_{k3}\boldsymbol{c}_{k3}.
\end{aligned} \tag{5.12}$$

Proof: We first prove that $t_{ijk}^{\sqrt{3}}$ is a valid PS-triangle, or in other words, that $t_{ijk}^{\sqrt{3}}$ contains all the involved PS-points. The molecule number m_{ijk} of V_{ijk} is always six and there are twelve PS-points, shown as gray dots in Figure 5.7, plus the vertex V_{ijk} itself, that have to be contained in the new PS-triangle.

The proposed PS-triangle $t_{ijk}^{\sqrt{3}}$ is marked in gray. We can see that the corners $Q_{ijk}^{\sqrt{3}}$ of this triangle are PS-points of the original PS-refinement Δ^{0*} that lie in the middle between V_{ijk} and the old vertices. We now zoom in on the left bottom corner and check that the PS-points A, B and C of $\Delta^{\sqrt{3}*}$ are contained in the proposed PS-triangle $t_{ijk}^{\sqrt{3}}$.

Referring to Equations (2.40) and (5.11) it immediately follows that

$$A = \lambda_{ij} Q_{ijk,1}^{\sqrt{3}} + (1-\lambda_{ij}) Q_{ijk,2}^{\sqrt{3}} \tag{5.13}$$

and

$$\begin{aligned}
B &= \omega_{ij} Q_{ijk,1}^{\sqrt{3}} + (1-\omega_{ij}) A \\
&= (\omega_{ij} + \lambda_{ij} - \omega_{ij}\lambda_{ij}) Q_{ijk,1}^{\sqrt{3}} + (1 - \lambda_{ij} - \omega_{ij} + \omega_{ij}\lambda_{ij}) Q_{ijk,2}^{\sqrt{3}}.
\end{aligned} \tag{5.14}$$

Denote by E the intersection of $Z_{ij}Z_{ik}$ and $V_{ijk}V_i$. We can determine an r with

$0 < r < 1$ such that

$$E = rV_i + (1-r)V_{ijk}, \tag{5.15}$$

$$\begin{aligned} C &= \frac{1}{2}(E + V_{ijk}) \\ &= rQ^{\sqrt{3}}_{ijk,1} + (1-r)V_{ijk}, \\ &= ((1-r)a_{ijk} + r)Q^{\sqrt{3}}_{ijk,1} + (1-r)b_{ijk}Q^{\sqrt{3}}_{ijk,2} + (1-r)c_{ijk}Q^{\sqrt{3}}_{ijk,3}. \end{aligned} \tag{5.16}$$

From Equations (5.13), (5.14) and (5.16) we see that the barycentric coordinates of A, B and C with respect to $t^{\sqrt{3}}_{ijk}$ are all positive, which means that they lie inside or on the boundary of the triangle $t^{\sqrt{3}}_{ijk}$. This is independent of the value for ω_{ij} that was used in Equation (5.11) to determine the position of the interior point Z_{ij}. The same reasoning can be used for the remaining PS-points of V_{ijk} and we conclude that $t^{\sqrt{3}}_{ijk}$ is a valid PS-triangle for the vertex V_{ijk} after one $\sqrt{3}$ subdivision step.

For the corresponding triangle $T^{\sqrt{3}}_{ijk}$ to be a valid control triangle, it needs to be tangent to the surface at V_{ijk}. This requirement follows from Corollary 2.2, the tangent property of the Bézier control net, and Corollary 2.5 on the continuity across boundaries, for each of the Bézier triangles in the PS-refinement.

From Equation (2.41) we know that

$$\begin{aligned} c^{\sqrt{3}}_{ijk,1} &= \tilde{L}_{i1}c_{i1} + \tilde{L}_{i2}c_{i2} + \tilde{L}_{i3}c_{i3} \\ c^{\sqrt{3}}_{ijk,2} &= \tilde{L}_{j1}c_{j1} + \tilde{L}_{j2}c_{j2} + \tilde{L}_{j3}c_{j3} \\ c^{\sqrt{3}}_{ijk,3} &= \tilde{L}_{k1}c_{k1} + \tilde{L}_{k2}c_{k2} + \tilde{L}_{k3}c_{k3}. \end{aligned} \tag{5.17}$$

The same convex combinations also apply to $Q^{\sqrt{3}}_{ijk}$ and consequently also to $\boldsymbol{c}^{\sqrt{3}}_{ijk}$. □

The formulas in Equation (5.17) use only convex combinations of the old data. The barycentric coordinates $\tilde{L}$ have values between 0 and 1. This means that the subdivision scheme is numerically stable.

5.4 Triadic subdivision

5.4.1 Two steps of $\sqrt{3}$ subdivision

Applying the $\sqrt{3}$ subdivision operator a second time, results in new vertices that coincide with the interior points introduced in the first step. They lie on the edges of the initial triangulation. As can be seen in Figure 5.6, this causes a refinement trisecting every original edge and splitting each original triangle into nine subtriangles. Hence one single refinement step of this scheme can be considered as the square root of one step of a triadic scheme. Triadic schemes have been studied in [79, 43].

In order to achieve a triadic scheme for splines defined on a bounded domain, the boundary triangles have to be treated different in the second $\sqrt{3}$ step. For this reason we choose to do two $\sqrt{3}$ steps in a practical implementation.

5.4.2 Boundaries

If the triangle (V_i, V_j, V_k) has no neighboring triangle $(V_i, V_j, V_{k'})$, i.e. the edge V_iV_j is a boundary edge, we need different rules for the second $\sqrt{3}$ step to achieve a trisection of the edge V_iV_j. Instead of inserting one new vertex in the interior of the boundary triangle, we insert two new vertices on the boundary edge on the positions where they would have been if there was a neighboring triangle.

For these vertices, V_{ij} and V_{ji} on Figure 5.8 we propose a different PS-triangle that uses only information of the triangle (V_i, V_j, V_k).

Theorem 5.4 *In the second $\sqrt{3}$ step on the boundary of a triangulation the control points $\boldsymbol{c}_{ij,1}$ and $\boldsymbol{c}_{ij,3}$ can be calculated by applying the formulas from Theorem 5.3 twice and*

$$\begin{aligned}\boldsymbol{c}_{ij,2} = &(\omega_{ij} + \lambda_{ij} - \omega_{ij}\lambda_{ij})(L_{i1}\boldsymbol{c}_{i1} + L_{i2}\boldsymbol{c}_{i2} + L_{i3}\boldsymbol{c}_{i3}) \\ &+ (1 - \omega_{ij} - \lambda_{ij} + \omega_{ij}\lambda_{ij})(L_{j1}\boldsymbol{c}_{j1} + L_{j2}\boldsymbol{c}_{j2} + L_{j3}\boldsymbol{c}_{j3}).\end{aligned} \tag{5.18}$$

Proof: From the tangent property of the Bézier control net of the triangles in the PS-refinement we know that $c_{ij,2}$ can be written as a linear combination of the two Bézier ordinates u_i and v_j from Figure 2.8

$$\begin{aligned}c_{ij,2} &= \omega_{ij}u_i + (1 - \omega_{ij})r_k \\ &= \omega_{ij}u_i + (1 - \omega_{ij})(\lambda_{ij}u_i + (1 - \lambda_{ij})v_j) \\ &= (\omega_{ij} + \lambda_{ij} - \omega_{ij}\lambda_{ij})u_i + (1 - \omega_{ij} - \lambda_{ij} + \omega_{ij}\lambda_{ij})v_j\end{aligned}$$

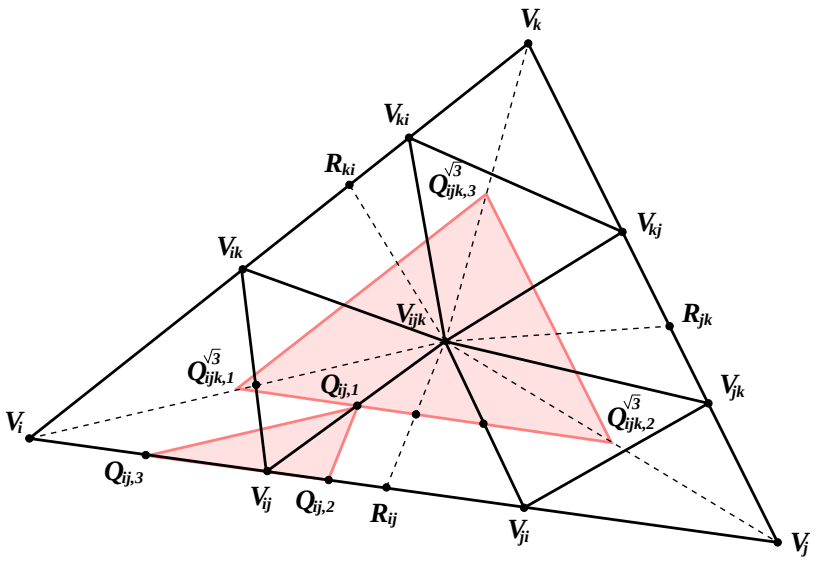


Figure 5.8: For vertices V_{ij} on a boundary edge we choose a different control triangle that only uses information of the triangle (V_i, V_j, V_k).

and finally

$$\begin{aligned} c_{ij,2} = &(\omega_{ij} + \lambda_{ij} - \omega_{ij}\lambda_{ij})(L_{i1}c_{i1} + L_{i2}c_{i2} + L_{i3}c_{i3}) \\ &+ (1 - \omega_{ij} - \lambda_{ij} + \omega_{ij}\lambda_{ij})(L_{j1}c_{j1} + L_{j2}c_{j2} + L_{j3}c_{j3}). \end{aligned} \tag{5.19}$$

The same convex combinations also apply to $Q_{ij,2}^{\sqrt{3}}$ and consequently also to $\boldsymbol{c}_{ij,2}^{\sqrt{3}}$. □

Note that because

$$\begin{aligned} \omega_{ij} + \lambda_{ij} - \omega_{ij}\lambda_{ij} &= \omega_{ij}(1 - \lambda_{ij}) + \lambda_{ij} > 0 \\ &= 1 - (1 - \omega_{ij})(1 - \lambda_{ij}) < 1. \end{aligned} \tag{5.20}$$

this special rule is a convex combination of the data on the previous level: the algorithm is also numerically stable at the boundaries.

5.4.3 Optimization

For the original vertices we can reuse their original control triangles. This is a valid choice because after subdivision, the PS-points are closer to the involved vertex and are therefore contained in the PS-triangle. However when doing subsequent subdivision steps this PS-triangle becomes larger relative to the size of the triangles in the refined triangulation. It is possible to choose a smaller PS-triangle, rescale t_i and denote the new PS-triangle by t_i'.

To find the appropriate scaling factor, we need to know the positions of the interior points of the PS-refinement $\Delta^{\sqrt{3}*}$. For example in Figure 5.7 in the triangle $(V_i, V_{ijk}, V_{ijk'})$ where $V_{ijk'}$ is the new vertex in the neighboring triangle $(V_i, V_j, V_{k'})$ we choose the interior point Z_{ij} between R_{ij} and V_i

$$Z_{ij} = \omega_i V_i + (1 - \omega_i)V_{ijk}, \qquad 0 < \omega_i < 1 \tag{5.21}$$

and we suggest to choose the same value ω_i for the interior points of all triangles that have V_i as a vertex. It is this value ω_i that we use to rescale t_i to t_i', i.e.

$$\begin{aligned} \hat{\boldsymbol{c}}_{i1} &= (\omega_i\alpha_{i1} + 1 - \omega_i)\,\boldsymbol{c}_{i1} + \omega_i\alpha_{i2}\,\boldsymbol{c}_{i2} + \omega_i\alpha_{i3}\,\boldsymbol{c}_{i3} \\ \hat{\boldsymbol{c}}_{i2} &= \omega_i\alpha_{i1}\,\boldsymbol{c}_{i1} + (\omega_i\alpha_{i2} + 1 - \omega_i)\,\boldsymbol{c}_{i2} + \omega_i\alpha_{i3}\,\boldsymbol{c}_{i3} \\ \hat{\boldsymbol{c}}_{i3} &= \omega_i\alpha_{i1}\,\boldsymbol{c}_{i1} + \omega_i\alpha_{i2}\,\boldsymbol{c}_{i2} + (\omega_i\alpha_{i3} + 1 - \omega_i)\,\boldsymbol{c}_{i3}. \end{aligned} \tag{5.22}$$

5.5 Concluding Remarks

A common application of spline subdivision is visualization. In the limit the control points converge to the spline surface itself. Depending on the resolution a certain number of subdivision steps is carried out and the control points are plotted. In the case of Powell-Sabin splines the control triangles are tangent to the surface, and the tangent point, often chosen as the barycenter, is a point on the surface. The connected tangent points form a triangular three dimensional mesh that is used for the graphical display.

An important motivation for the study of subdivision algorithms is their relation to multiresolution analysis and wavelet bases. The successive spline spaces are a nested sequence and the basis functions satisfy a refinement equation. In the next chapter we use the triadic scheme for non uniform Powell-Sabin splines to develop wavelets.

The dyadic subdivision scheme for uniform Powell-Sabin splines that we recalled in the first section, will be generalized to surfaces with arbitrary topology in Chapter 7. Subdivision is then used as a surface definition technique and the initial control points do not define a parametric spline surface. However, the spline background facilitates the design and the convergence analysis.

Chapter 6

Spline Wavelets

6.1 Introduction

Splines have had a significant impact on the theory of the wavelet transform. The earliest example is the Haar wavelet [59] which is a spline of degree 0. This construction was extended to higher order splines by Strömberg [110]. Other orthogonal spline wavelets are the Battle-Lemarié functions [6, 76]. Later the orthogonality requirement was dropped to achieve other desirable wavelet properties. The B-spline wavelets [15, 115] are compactly supported and achieve a near optimal time frequency localization. Also some representatives of the Cohen-Daubechies-Feauveau wavelets [18] are splines as well. Spline wavelets in the case of bounded intervals were studied in [94] and the non uniform case in [82].

Although different spline bases are constructed on triangulations, little is published on spline wavelets on triangulations. In [51] piecewise linear wavelets and orthogonal spaces are investigated. The construction we use here is inspired by [70] where the lifting scheme is applied in higher dimensions but on regular lattices and by [80] where the construction of wavelets is linked with subdivision.

In Section 6.2 we describe the filters that are used in the forward and inverse wavelet transform, and how they can be decomposed in lifting steps. Section 6.3 overviews properties of the complement space. The stability of wavelet transforms on irregular domain grids is not straightforward. In our design of the update step in Section 6.4 we pay special attention to this. In Section 6.5 we illustrate the result and give a proof for the stability. This is joint work with Jan Maes. Finally we give remarks on these results in Section 6.6.

6.2 Multiscale decomposition

The construction of spline wavelets starts with the specification of the underlying multiresolution spaces in terms of polynomial splines instead of only implicitly through the refinement equation. Each space V^j should be spanned by a Riesz basis as in Definition 2.11. The normalized B-spline basis functions satisfy this demand and consequently qualify as scaling functions and we choose

$$\mathbf{s} = \mathbf{c}' \tag{6.1}$$
$$\Phi = \mathbf{B}'. \tag{6.2}$$

Recall from Chapter 4 that $\mathbf{s}$ is a column vector that contains the scaling coefficients, and Φ is a row vector with scaling functions. The vector $\mathbf{c}'$, introduced in Chapter 3, contains the normalized B-spline control points and $\mathbf{B}'$ the B-spline basis functions, scaled with respect to a power p of their support depending on which norm is used. In wavelet theory one works with the 2-norm. The space V^j that is spanned by the scaling functions Φ^j is the spline space $S_2^1(\Delta^{j*})$.

The subdivision matrix P^j provided the connection between Φ^j and Φ^{j+1} in the refinement equation in Theorem 4.1. Clearly for our spline space the matrix P^j is already known to us, it consists of a combination of scaling factors proportional to the supports of the basis functions and the coefficients in the subdivision rules of the previous chapter. Consequently the subdivision matrix P^j is fully computable. It can be partitioned in two parts, one part is for the old vertices and the other for the new vertices

$$P^j = \begin{bmatrix} O^j \\ N^j \end{bmatrix}. \tag{6.3}$$

These submatrices can be further written as

$$\begin{bmatrix} O^j \\ N^j \end{bmatrix} = D^{j+1} \begin{bmatrix} \tilde{O}^j \\ \tilde{N}^j \end{bmatrix} (D^j)^{-1} = \begin{bmatrix} D_o^{j+1} \tilde{O}^j (D^j)^{-1} \\ D_n^{j+1} \tilde{N}^j (D^j)^{-1} \end{bmatrix}. \tag{6.4}$$

In these equations the matrix D^j is the diagonal matrix from Equation (3.33). The diagonal elements are the areas of the support of the basis functions on level j to the power p. The matrices $\tilde{O}^j$ and $\tilde{N}^j$ describe the subdivision rules for the unnormalized coefficients in matrix from. The subscripts o and n stand for old and new respectively. D_o^{j+1} is the upper part of the diagonal matrix D^{j+1} and D_n^{j+1} the lower part.

Analogously we also split Φ^{j+1} into functions $\mathcal{O}^{j+1}$ associated with the old vertices in Δ^j and functions $\mathcal{N}^{j+1}$ associated with the new vertices added when going from Δ^j to Δ^{j+1}:

$$\mathbf{\Phi}^{j+1} = [\mathcal{O}^{j+1} \mathcal{N}^{j+1}]. \tag{6.5}$$

6.2.1 Wavelet functions

The wavelet functions collected in Ψ^j span the space W^j that is the complement of V^j in V^{j+1}. The sets Φ^j and $\mathcal{N}^{j+1}$ together span V^{j+1}. Thus, the scaling functions on level $j+1$ in $\mathcal{N}^{j+1}$ that correspond to the new positions can be used as wavelet functions

$$\Psi^j = \mathcal{N}^{j+1} \quad \text{and} \quad Q^j = \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{6.6}$$

where Q^j is the analog for P^j for the wavelet functions in Equation (4.9).

By choosing the wavelet functions as the scaling functions on the finer level, we also made a choice for the complement space W^j. However, it will be desirable to have another complement space with certain properties related to stability. The wavelet functions can be found by projecting the $\mathcal{N}^{j+1}$ onto the desired complement space W^j along V^j

$$\Psi^j = \mathcal{N}^{j+1} - \Phi^j U^j. \tag{6.7}$$

This projection is not necessarily orthogonal. For each wavelet function there is a corresponding column in U^j. The nonzero entries in this column together will be called the stencil for that wavelet function. Remark that if there are no zero entries in U^j, the wavelets will have the whole domain Ω as their support.

6.2.2 Lifting

The reconstruction or synthesis filter in matrix form is now

$$\begin{bmatrix} P^j & Q^j \end{bmatrix} = \begin{bmatrix} O^j & -O^j U^j \\ N^j & I - N^j U^j \end{bmatrix}. \tag{6.8}$$

The matrix $O^j = D_o^{j+1} \tilde{O}^j (D^j)^{-1}$ describes the subdivision rules for the old vertices. As discussed in Section 5.4.3 the old PS-triangle is still valid because it contains the appropriate PS-points. In that case $\tilde{O}^j$ is simply the identity matrix. It was however possible to make the PS-triangle smaller. The new PS-triangle is then a rescaled version of the old PS-triangle and both triangles span the same plane. This scaling operation is easily invertible.

In both cases the matrix O^j is invertible and consequently the filter bank operation is also easily invertible and we find the analysis filters A^j and B^j

$$\begin{bmatrix} A^j \\ B^j \end{bmatrix} = \begin{bmatrix} (O^j)^{-1} - U^j N^j (O^j)^{-1} & U^j \\ -N^j (O^j)^{-1} & I \end{bmatrix}. \tag{6.9}$$

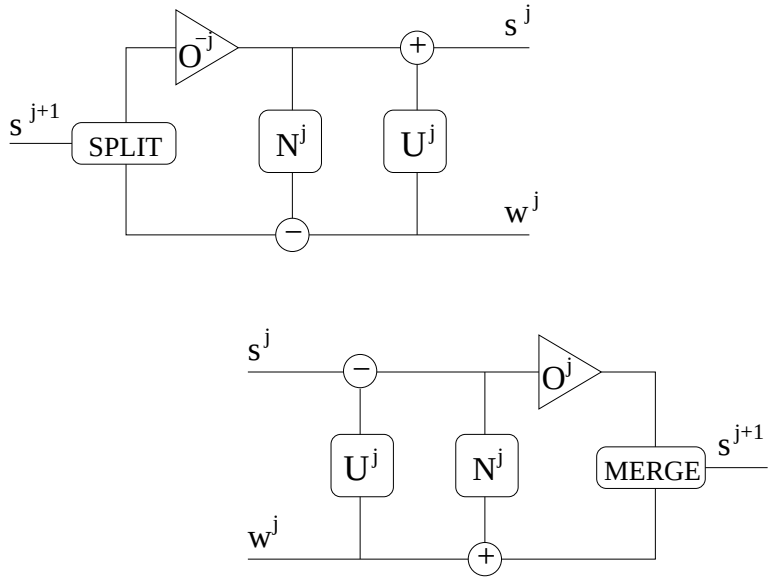


Figure 6.1: Filter bank factored in lifting steps.

These filters can be factored as follows

$$\begin{bmatrix} P^j & Q^j \end{bmatrix} = \begin{bmatrix} O^j & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ N^j & I \end{bmatrix} \cdot \begin{bmatrix} I & -U^j \\ 0 & I \end{bmatrix} \tag{6.10}$$

and

$$\begin{bmatrix} A^j \\ B^j \end{bmatrix} = \begin{bmatrix} I & U^j \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ -N^j & I \end{bmatrix} \cdot \begin{bmatrix} (O^j)^{-1} & 0 \\ 0 & I \end{bmatrix} \tag{6.11}$$

This relates to the concept of lifting [112]. Every factor in the factorization of the filters corresponds to a lifting or a scaling step. The filter bank factored in lifting steps is shown in Figure 6.1.

First the control points $\mathbf{s}^{j+1}$ are split into two sets. The first set, $\mathbf{s}_o^{j+1}$, contains the control points that correspond to vertices in Δ^j, and the second set, $\mathbf{s}_n^{j+1}$, contains control points that correspond to vertices that are in Δ^{j+1} but not in Δ^j. The splitting is depends on the connectivity induced by the subdivision scheme.

Then we want to treat the fine scale control points $\mathbf{s}_o^{j+1}$ corresponding to the old vertices as the control points of a surface defined on the coarse scale triangulation Δ^j. Therefore we first need to scale the control triangles in $\mathbf{s}_o^{j+1}$ with $(O^j)^{-1}$. We undo the scaling O^j that is used when going from coarse to fine.

After this we can apply to $\mathbf{s}_o^{j+1}$ the part N^j of the subdivision algorithm P^j that leads to the control points for the new vertices. The result is used as a prediction for $\mathbf{s}_n^{j+1}$ and substracted from this sequence. This yields the wavelet coefficients $\mathbf{w}^j$ on the lower branch in Figure 6.1.

Finally $\mathbf{s}_o^{j+1}$ is updated with a linear combination of wavelet coefficients $\mathbf{w}^j$. The coefficients in the linear combination are defined by the values in U^j. This yields the scaling coefficients $\mathbf{s}^j$ on the upper branch in Figure 6.1.

Reversing the lifting scheme is straightforward: we run through the scheme backwards, replace plus-signs with minus-signs, undo scaling operations and merge what had been split. So unlike the classical wavelet transform where A^j, B^j, P^j and Q^j are used explicitly, the same filters O^j, N^j and U^j appear now in the forward and inverse transform.

Note that we never compute the filters O^j and N^j explicitly nor do the matrix multiplication. Instead we multiply with the normalization factors and apply the subdivision rules. As will become clear later also for the filter U^j there is no matrix multiplication involved. The matrix U^j will be calculated column by column and the old coefficients are updated incrementally.

6.3 Properties of the complement space

6.3.1 Approximation

An obvious requirement for the decomposition is that the coarse scale approximation should be as close as possible to the fine scale function. The optimal solution can be obtained if we find basis functions ψ_k^j which are orthogonal to the basis functions ϕ_l^j and $V^j \oplus W^j = V^{j+1}$ is an orthogonal decomposition. In that case the approximation error is minimal in the least squares sense.

In an orthonormal basis each scaling function ϕ_k^j is not only orthogonal to the wavelet functions ψ_l^j but also to the other scaling functions ϕ_l^j. However, requiring an orthonormal basis is a very strong condition which eliminates most degrees of freedom. Additional properties such as smoothness, differentiability, symmetry and local support cannot be satisfied anymore.

Hence, in many applications, we prefer the orthogonal decomposition without the additional orthogonality between the scaling functions. We call this the semi-orthogonal setting. The additional degrees of freedom are used to obtain smooth basis functions with local support.

In practice, it often turns out that even the semi-orthogonal setting is quite difficult to establish. Therefore an even weaker condition is imposed on the basis functions. The motivation is that the space V^j contains some low degree polynomials up to a certain order $\tilde{N}$ as in Definition 4.3. Then instead of requiring that the wavelet functions should be orthogonal to all functions in V^j, we restrict ourselves to requiring that the wavelet functions are at least orthogonal to some of these low degree polynomials up to an order N. N is called the number of vanishing moments (Definition 4.4).

6.3.2 Stability

Aside from approximation properties, the stability of the transformation T^n is of central importance. More precisely, referring to Equations (4.14) and (4.15), one has to ensure that

$$\|\mathbf{s}^n\| \sim \|[(\mathbf{s}^0)^T\ (\mathbf{w}^0)^T\ \ldots\ (\mathbf{w}^{n-1})^T\ (\mathbf{w}^n)^T]^T\| \tag{6.12}$$

uniformly in n, which is equivalent to

$$\|T^n\|, \|(T^n)^{-1}\| = \mathcal{O}(1), \quad n \to \infty. \tag{6.13}$$

For an orthogonal wavelet transform this is automatically satisfied. However, it suffices to have an orthogonal multiresolution decomposition in the sense that all complement spaces W^j are orthogonal complements of V^j in V^{j+1} and therefore also orthogonal to W^i with $i < j$. Orthogonality of the basis functions need only hold between different levels and not within one level. This is again the semi-orthogonal setting.

From Dahmen [22] we know that the wavelet transform T^n is uniformly stable, i.e. (6.13) holds, if the multiscale basis $\bigcup_{j=1}^{\infty} \Psi^j$ is also a Riesz basis. This means that a small change in a wavelet coefficient should correspond to a small change in the function that is represented in the multiscale basis. A necessary condition is that the combined basis $\Phi^j \cup \Psi^j$ is uniformly stable or alternatively that

$$\|M^j\|, \|(M^j)^{-1}\| = \mathcal{O}(1). \tag{6.14}$$

W^j is then called a stable complement of V^j in V^{j+1}. Remark that this is not sufficient for the stability of the overall transform, because, in principle, the condition of the multiscale basis $\bigcup_{j=1}^{\infty} \Psi^j$ (6.34) may still become prohibitively large.

Dahmen establishes criteria for stability that are not based on Fourier techniques, so that they are useful for irregular and bounded domains. Certain Jackson and Bernstein inequalities must be satisfied from which we can see that the wavelets must have at least one vanishing moment. The meaning of one vanishing moment is that the DC-component must be conserved in the low pass signal. This is in accordance with results on the Sobolev regularity of refinable functions [19, 100]. The Sobolev exponent turns out to be strictly positive and proportional to the approximation order.

6.4 Design of the update step

From the above considerations we conclude that it is important to have at least one vanishing moment for the wavelets and that it is desirable to have a complement space W^j that is orthogonal to V^j. Without an update step the wavelet functions are the scaling functions on the next finer level corresponding to the new vertices. These unlifted wavelet functions do not have the above mentioned properties and we use the update step to achieve them. We first overview some common choices for the update step. Then we describe our concrete approach and, finally, we define the inner product and explain how to compute it efficiently.

6.4.1 Trivial updates

A simple update from [112] is to obtain lifted wavelets with N vanishing moments, where N depends on the size of the stencil. The lifting coefficients in U^j are found as the solution of a linear system that arises from the condition that the wavelet functions are orthogonal to a basis for Π_{N-1}. Unfortunately the magnitude of the lifting coefficients can be very sensitive to the choice of the stencil. The update coefficients in U^j appear to be unbounded, and therefore the transform is not stable. Simoens gives examples of this phenomenon in the one-dimensional case [108].

Another possibility would be to choose the update operator U^j as the orthogonal projection from W^j into V^j. We then have a semi-orthogonal multiresolution decomposition and the stability condition (6.34) is satisfied. Because

$\Pi_2 \subset V^j$ we immediately have two vanishing moments for free in this case. Unfortunately, to achieve semiorthogonality the width of the stencil cannot be limited, every scaling function in Φ^j is involved in the update step. The resulting wavelet functions are not locally supported but stretch out over the whole domain Ω.

A third possibility is to fix the stencil and the scaling functions that come in the update step in advance and then orthogonalise each wavelet function to the subset of V^j that is defined by the stencil. Choosing the stencil is then a trade off between semiorthogonality and local support. A disadvantage of this idea is that we loose the two vanishing moments we had for free in the semiorthogonal case. Alternatively we can minimize the maximum value of the update coefficients [119].

6.4.2 Least squares approach

The solution we propose also fixes the stencil in advance in order to have local support for the wavelet functions. The stencil is different for a new vertex on an edge or a new vertex in the interior of an old triangle. Both stencils are shown in Figure 6.2. A wavelet function of a new vertex on an edge is updated by the scaling functions of the two neighboring old vertices on that edge. Because there are three basis functions associated with each vertex, the size of this stencil is six. For the wavelet function of a new vertex in the interior of an old triangle the scaling functions of the corners of the triangle are used. This yields a stencil of size nine.

We want to orthogonalise the wavelets to their predefined set of scaling functions, but we also want at least one vanishing moment. This leads to an overdetermined system because we do not have enough degrees of freedom

$$A \cdot \mathbf{u} = \mathbf{b} \quad \text{with} \quad \begin{cases} A_{kl} &= \langle \phi_k^j, \phi_l^j \rangle \\ b_k &= \langle \phi_k^j, \mathcal{N}_l^{j+1} \rangle \end{cases} \tag{6.15}$$

$$\sum_{m=1}^{9} \langle 1, \phi_m^j \rangle \, u_m = \langle 1, \mathcal{N}_l^{j+1} \rangle. \tag{6.16}$$

In these equations the column vector $\mathbf{u}$ with elements u_m denotes the lth column of the update matrix U^j and $\mathcal{N}_l^{j+1}$ denotes the corresponding fine level scaling function.

We could solve this system with an approximating method, but then the vanishing moment condition is not exactly fulfilled. Therefore we treat the system as a least squares problem with an equality constraint. We first solve Equation (6.16) for u_1 and substitute u_1 in the remaining Equations (6.15). The resulting system

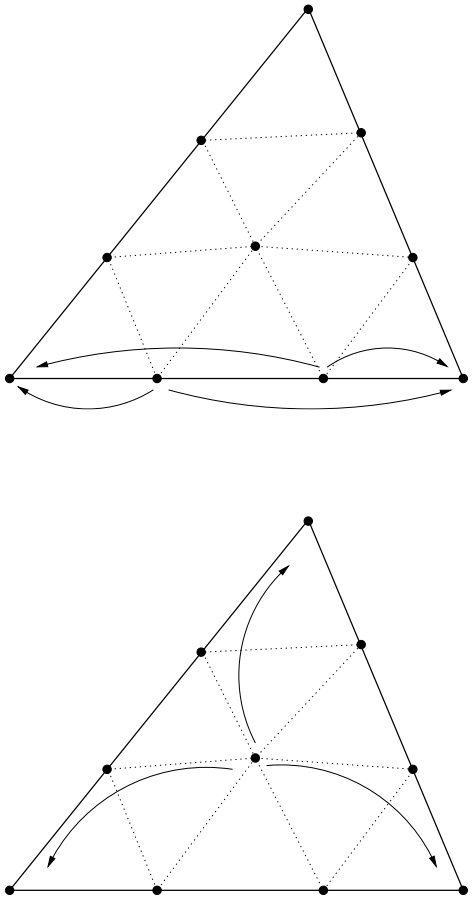


Figure 6.2: The update stencil is chosen a priori.

equals

$$A' \cdot \mathbf{u}' = \mathbf{b}' \quad \text{with} \quad \begin{cases} A'_{kl} &= \langle \phi_k^j, \phi_{l+1}^j \rangle - \frac{\langle \phi_k^j, \phi_1^j \rangle \langle 1, \phi_{l+1}^j \rangle}{\langle 1, \phi_1^j \rangle} \\ u'_l &= u_{l+1} \\ b'_k &= \langle \phi_k^j, \mathcal{N}_l^{j+1} \rangle - \frac{\langle \phi_k^j, \phi_1^j \rangle \langle 1, \mathcal{N}_l^{j+1} \rangle}{\langle 1, \phi_1^j \rangle}. \end{cases} \tag{6.17}$$

This overdetermined system is then solved for $\mathbf{u}'$ with a least squares method. The resulting wavelet function ψ_l^j has one vanishing moment and is close to orthogonal to its stencil of scaling functions in V^j.

6.4.3 Inner products

The inner product of two Powell-Sabin splines $f, g \in V^j$ is defined as

$$\langle f, g \rangle := \int_\Omega f(x,y) g(x,y) dxdy, \tag{6.18}$$

or, equivalently,

$$\langle f, g \rangle := \sum_{\mathcal{T}^j \in \Delta^{j*}} \int_{\mathcal{T}^j} f(x,y) g(x,y) dxdy. \tag{6.19}$$

We remark that some authors use the following alternative definition for the inner product

$$\langle f, g \rangle := \sum_{\mathcal{T}^j \in \Delta^j(\Omega)} \frac{1}{Ar_{\mathcal{T}^j}} \int_{\mathcal{T}^j} f(x,y) g(x,y) \; dxdy. \tag{6.20}$$

The latter definition implies that triangles of different geometric size and shape are weighted equally. The resulting wavelet spaces are independent of the geometry of the mesh and consequently a significant amount of precomputation of inner products is possible. However, in the case of Powell-Sabin subdivision the scaling functions also depend on the PS-triangle which can be different for each vertex. Therefore no precomputation is possible. Moreover, we prefer the first definition because it does not ignore the inherent irregularity of the triangulation.

The practical computation of the inner product is done by considering one triangle $\mathcal{T}^j \in \Delta^{j*}$ at the time. The functions $f|_{\mathcal{T}^j}$ and $g|_{\mathcal{T}^j}$ are bivariate polynomials of degree ≤ 2 and their Bézier representation (2.8) can easily be obtained from the B-spline representation (2.24) with the formulas in Equation (2.41) and (2.42). Denote the Bézier ordinates of $f|_{\mathcal{T}^j}$ and $g|_{\mathcal{T}^j}$ as b_λ and c_λ respectively, then $I = \langle f, g \rangle$ is

$$
\begin{aligned}
I &= \int_{\mathcal{T}^j} \sum_{|\lambda|=2} b_\lambda B_\lambda^2(\tau) \sum_{|\lambda|=2} c_\lambda B_\lambda^2(\tau)\, d\tau, \\
&= \int_0^1 \int_0^{1-\tau_1} |J(\tau)| \sum_{|\lambda|=2} b_\lambda B_\lambda^2(\tau) \sum_{|\lambda|=2} c_\lambda B_\lambda^2(\tau)\, d\tau_2 d\tau_1, \qquad (6.21)
\end{aligned}
$$

where the Jacobian $|J(\tau)|$ equals $2 \cdot Ar_{\mathcal{T}^j}$. Some elementary calculus yields

$$
I = Ar_{\mathcal{T}^j} \begin{bmatrix} c_{200} \\ c_{110} \\ c_{020} \\ c_{011} \\ c_{002} \\ c_{101} \end{bmatrix}^T \begin{bmatrix} \frac{1}{15} & \frac{1}{30} & \frac{1}{90} & \frac{1}{90} & \frac{1}{90} & \frac{1}{30} \\ \frac{1}{30} & \frac{2}{45} & \frac{1}{30} & \frac{1}{45} & \frac{1}{90} & \frac{1}{45} \\ \frac{1}{90} & \frac{1}{30} & \frac{1}{15} & \frac{1}{30} & \frac{1}{90} & \frac{1}{90} \\ \frac{1}{90} & \frac{1}{45} & \frac{1}{30} & \frac{2}{45} & \frac{1}{30} & \frac{1}{45} \\ \frac{1}{90} & \frac{1}{90} & \frac{1}{90} & \frac{1}{30} & \frac{1}{15} & \frac{1}{30} \\ \frac{1}{30} & \frac{1}{45} & \frac{1}{90} & \frac{1}{45} & \frac{1}{30} & \frac{2}{45} \end{bmatrix} \begin{bmatrix} b_{200} \\ b_{110} \\ b_{020} \\ b_{011} \\ b_{002} \\ b_{101} \end{bmatrix}. \qquad (6.22)
$$

Substituting (6.22) in (6.19) yields the inner product.

6.5 Results

6.5.1 Wavelets

Figure 6.3 shows a wavelet function corresponding to a new vertex in the interior of an old triangle. In the first picture we see the wavelet before the update step, it is the scaling function on the finer level. When a wavelet has one vanishing moment it is orthogonal to a constant function and the inner product with 1 equals zero. This means that the integral of the wavelet must be zero. This condition is obviously not satisfied for the unlifted wavelet.

In the second picture we see the wavelet after the update step. The wavelet now has a vanishing moment, the integral equals zero. Note the larger lob in the front because for every vertex there are three scaling functions or wavelet functions, each for one corner of the control triangle. The sum of the three basis functions has a balanced and symmetric shape.

The support of a wavelet function is the union of the supports of the scaling functions in its update stencil. Therefore the support of a wavelet with its vertex on an edge of the coarse scale triangulation is slightly smaller than the support of wavelet with its vertex inside an old triangle.

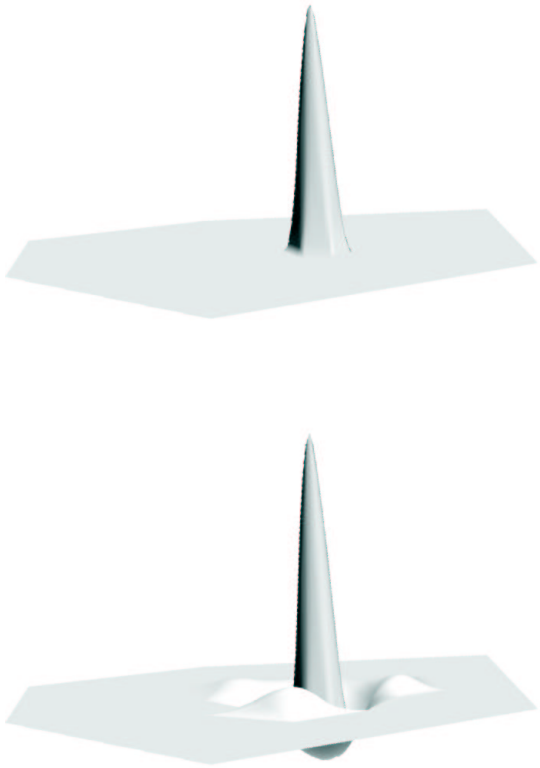


Figure 6.3: Top: a wavelet function before update is the scaling function on the finer level. Bottom: the integral of a wavelet function after update equals zero.

6.5.2 Stability

Theorem 6.1 *The one level wavelet transforms M^j are uniformly bounded in j*

$$\|M^j\|, \|(M^j)^{-1}\| = \mathcal{O}(1), \tag{6.23}$$

with $\|\cdot\|$ the l_2 norm.

Proof:

1. From the factorizations (6.10) and (6.11) we know it is sufficient to prove

   $$\|U^j\|, \|N^j\|, \|O^j\|, \|(O^j)^{-1}\| = \mathcal{O}(1). \tag{6.24}$$

   Since an arbitrary matrix F satisfies the property $\|F\|^2 \leq \|F\|_1 \|F\|_\infty$, we will look for upper bounds for $\|\cdot\|_1$ and $\|\cdot\|_\infty$.

2. From the construction of the update step it follows that U^j is a sparse matrix. Each column has at most nine non zero entries, and each row has at most $3 \cdot \frac{2\pi}{\theta_{\Delta^j}}$ non–zero entries. It follows that $\|U^j\|_1 \leq 9 \max_{kl} |\alpha^j_{kl}|$ and $\|U^j\|_\infty \leq \frac{6\pi}{\theta_{\Delta^j}} \max_{kl} |U^j_{kl}|$, or

   $$\|U^j\| \leq 3\sqrt{\frac{6\pi}{\theta_{\Delta^j}}} \max_{kl} |U^j_{kl}|. \tag{6.25}$$

3. To derive an upper bound for $\max_{kl} |U^j_{kl}|$ we define A, $\mathbf{u}$ and $\mathbf{b}$ as in Section 5.4. The column vector $\mathbf{u}$ that is part of the update matrix U^j satisfies

   $$A \cdot \mathbf{u} = \mathbf{b} + \boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \text{ the least squares error,} \tag{6.26}$$

   $$\sum_{m=1}^{9} \langle 1, \phi^j_m \rangle u_m = \langle 1, \mathcal{N}^{j+1}_l \rangle. \tag{6.27}$$

   Clearly A is a Gram matrix. Suppose that $\{\phi^{*j}_1, \ldots, \phi^{*j}_9\}$ is a dual base for $\{\phi^j_1, \ldots, \phi^j_9\}$, i.e. $\langle \phi^j_k, \phi^{*j}_l \rangle = \delta_{kl}$, and suppose that $\phi^{*j}_l =: \sum_{k=1}^9 b_{kl} \phi^j_k$, then $A^{-1} = (b_{kl})$ and

   $$\|\phi^{*j}_l\|^2 = \langle \phi^{*j}_l, \phi^{*j}_l \rangle = \langle \phi^{*j}_l, \sum_{k=1}^{9} b_{kl} \phi^j_k \rangle = b_{ll}. \tag{6.28}$$

   From the L_2 stability of Φ^j it follows that $k_1^2 \sum_{k=1}^9 b^2_{kl} \leq \|\phi^{*j}_l\|^2$, with k_1

a constant only depending on θ_{Δ^j}. So $k_1^2 \sum_{k=1}^{9} b_{kl}^2 \leq b_{ll}$ and we derive $\frac{b_{kl}^2}{b_{ll}} \leq \frac{1}{k_1^2}$. For $k = l$ this becomes $b_{ll} \leq \frac{1}{k_1^2}$. Combining these two yields

$$|b_{kl}| \leq \frac{1}{k_1^2}. \tag{6.29}$$

Because of (6.26)

$$\|\mathbf{u}\|_\infty \leq \|A^{-1}\|_\infty (\|\mathbf{b}\|_\infty + \|\boldsymbol{\epsilon}\|_\infty). \tag{6.30}$$

Combining (6.28), (6.29) and (6.30) gives

$$\max_i |u_i| \leq \frac{9}{k_1^2}(1 + \|\boldsymbol{\epsilon}\|_\infty). \tag{6.31}$$

4. Suppose that we choose for the column vector $\mathbf{u}$ a solution in which all the elements u_i are the same. From the equality constraint (6.27) we find

$$\forall i : \; u_i = \frac{\langle 1, \mathcal{N}_l^{j+1} \rangle}{\sum_{m=1}^{9} \langle 1, \phi_m^j \rangle}. \tag{6.32}$$

From a similar argument as in Lemma 3.4 we can see that the fraction of the areas of two molecules with a common triangle is bounded by a constant. Furthermore, before normalization with their support the scaling functions form a convex partition of unity. Consequently the fraction of inner products in the above equation is also bounded and $u_i = \mathcal{O}(1)$. The elements of the matrix A and the vector $\mathbf{b}$ consist of inner products of scaling functions. The same reasoning leads to $\|\boldsymbol{\epsilon}\|_2 = \|A \cdot \mathbf{u} - \mathbf{b}\|_2 = \mathcal{O}(1)$.

Since the least squares method finds an $\mathbf{u}$ that minimizes $\|A \cdot \mathbf{u} - \mathbf{b}\|_2$, we have a contradiction if we suppose that $\|\boldsymbol{\epsilon}\|_2$ is not at least $\mathcal{O}(1)$. Hence $\|\boldsymbol{\epsilon}\|_2 = \mathcal{O}(1)$ and because of the norm equivalence also $\|\boldsymbol{\epsilon}\|_\infty = \mathcal{O}(1)$.

Combining this with Equation (6.31) yields

$$\max_{kl} |U_{kl}^j| \leq \frac{9k_3}{k_1^2}, \tag{6.33}$$

with k_3 a constant.

5. According to (6.4), N^j can be written as $D_n^{j+1}\tilde{N}^j(D^j)^{-1}$, with $\tilde{N}^j$ a sparse matrix. Because the subdivision scheme only uses convex combinations, we know that all entries in $\tilde{N}^j$ are smaller than one. Every row in N^j corresponds to a function in $\mathcal{N}^{j+1}$ and every column to a function in Φ^j. The (k,l)th entry of N^j is non zero if and only if the support of $\mathcal{N}_k^{j+1}$ lies in the support of ϕ_l^j. Suppose that the (k,l)th entry is non zero, then it is bounded by $\sqrt{Ar_{\mathcal{N}_k^{j+1}}/Ar_{\phi_l^j}}$ which is $\mathcal{O}(1)$. Hence $\|N^j\| = \mathcal{O}(1)$.

6. We recall from (6.4) that O^j can be written as $D_o^{j+1}\tilde{O}^j(D^j)^{-1}$. To prove the boundedness of $\|O^j\|$ similar arguments can be used as in (5). We also need $\|(O^j)^{-1}\| = \mathcal{O}(1)$. Again we find that it is sufficient to prove that $\sqrt{Ar_{\phi_l^j}/Ar_{\phi_l^{j+1}}} = \mathcal{O}(1)$. This follows easily from Lemma 3.4.

□

6.5.3 Condition numbers

When doing the wavelet transform there should be no significant loss of accuracy in the data, or in other words the condition numbers should remain uniformly bounded

$$\|T^n\|.\|(T^n)^{-1}\| = \mathcal{O}(1), \quad n \to \infty. \tag{6.34}$$

This is a weaker condition than the stability condition (6.13). Analogously the condition numbers of the one level transforms M^j are uniformly bounded in j if

$$\|M^j\|.\|(M^j)^{-1}\| = \mathcal{O}(1). \tag{6.35}$$

Table 6.1 gives condition numbers for an example with six triangles on level 0 in the 1–norm and table 6.2 for the ∞–norm. The condition numbers for the scaling and the prediction operators are upper bounds, the condition numbers for the update are exact. For the 1-norm all condition numbers are of order 10^1. This confirms the stability of the wavelet transform. For the ∞-norm the condition numbers for the update step are slightly higher. To our knowledge no results on triangulations are available to compare with. We refer to [23] where the authors report higher condition numbers on the interval.

$\|\cdot\|_1$	level j				
	0	1	2	3	4
$\kappa\left(\begin{bmatrix} O^j & 0 \\ 0 & I \end{bmatrix}\right)$	43,821	43,666	43,702	43,714	43,714
$\kappa\left(\begin{bmatrix} I & 0 \\ N^j & I \end{bmatrix}\right)$	11,419	13,018	13,019	13,019	13,019
$\kappa\left(\begin{bmatrix} I & -U^j \\ 0 & I \end{bmatrix}\right)$	20,305	17,128	17,128	17,128	17,128

Table 6.1: Condition numbers in 1-norm

$\|\cdot\|_\infty$	level j				
	1	2	3	4	5
$\kappa\left(\begin{bmatrix} O^j & 0 \\ 0 & I \end{bmatrix}\right)$	2,4981	2,4935	2,4936	2,4936	2,4936
$\kappa\left(\begin{bmatrix} I & 0 \\ N^j & I \end{bmatrix}\right)$	11,431	15,021	15,022	15,022	15,022
$\kappa\left(\begin{bmatrix} I & -U^j \\ 0 & I \end{bmatrix}\right)$	212,31	262,49	262,50	262,50	262,51

Table 6.2: Condition numbers in ∞-norm

6.6 Discussion

6.6.1 Multiwavelets

For every domain vertex there are three control points and the set of control points can be regarded as consisting of three subsets each for a corner of the control triangles. The wavelets developed here are in fact an example of multiwavelets [91, 28, 21]. In the regular setting this means that the scaling and wavelet functions are not translates and dilates of one function but of several functions. In the case of uniform Powell-Sabin spline wavelets we have three basic scaling functions [129].

In the irregular setting that we consider here, the multiwavelet aspect is less obvious. We illustrate it by looking at the scaling step described by O^j more in detail. Recall from (6.4) that O^j can be written as $D_o^{j+1}\tilde{O}^j(D^j)^{-1}$. The matrices D_o^{j+1} and D^j are (part of) diagonal matrices, so their action on a vector is equivalent to multiplying each coefficient with its corresponding diagonal element.

The matrix $\tilde{O}^j$ on the other hand is a block diagonal matrix

$$\tilde{O}^j = \begin{bmatrix} \tilde{O}_1 & & & & \\ & \ddots & & & \\ & & \tilde{O}_i & & \\ & & & \ddots & \\ & & & & \tilde{O}_n \end{bmatrix} \tag{6.36}$$

with n the number of vertices in the domain triangulation. A block $\tilde{O}_i$ describes the scaling of the control triangle corresponding to vertex V_i with the formulas (5.22).

$$\tilde{O}_i = \begin{bmatrix} \omega_i\alpha_{i1} + 1 - \omega_i & \omega_i\alpha_{i2} & \omega_i\alpha_{i3} \\ \omega_i\alpha_{i1} & \omega_i\alpha_{i2} + 1 - \omega_i & \omega_i\alpha_{i3} \\ \omega_i\alpha_{i1} & \omega_i\alpha_{i2} & \omega_i\alpha_{i3} + 1 - \omega_i \end{bmatrix} \tag{6.37}$$

A new control is influenced by the values in each of the three control points of the vertex. Here we see again why it is important that O^j is invertible. In contrast to the prediction and update filter the scaling has to be inverted to undo the scaling.

Also in the prediction and the update step the multiwavelet aspect can be seen. Each new subset is determined by the three old subsets.

6.6.2 Parametric splines

In this chapter we considered the functional setting, i.e. the geometry was given as the graph of a scalar valued function defined over the plane. In a more general setup, we have to use parametric representations where the geometry is given by a vector valued function which maps some planar or non planar parameter domain into three dimensional space. The parametric Powell-Sabin spline of Definition 2.15 has a planar parameter domain. Each of the coordinates is defined by a separate scalar valued function. As a consequence the control coefficients and the detail coefficients are also vector valued.

If we consider decomposition and reconstruction only, the processing of vector valued functions is done by simply applying the same operators simultaneously to all three coordinate functions. However, if the result of the decomposition is filtered in some way for some application, i.e. if the position of the control points is changed then a more geometric definition of the detail information is necessary.

This is illustrated in Figure 6.4. The top picture consist of a global shape, the straight line, and detail information. The line can be regarded as the low pass information and is described by the scaling coefficients. The ondulations are the high pass information and are described by the wavelet coefficients. Then, in the middle figure, we modify the global shape and add the detail information back. This leads to counter intuitive results because the rotation of the global shape's tangent is neglected.

The detail is considered to be measured in the z-direction, but it would be better if it was interpreted as, for example, being orthogonal to the tangent vector at the global shape at that point. So in order to avoid this effect, one should use some kind of local frame representation [53, 54] for the detail coefficients that for example consist of the normal vector and the tangents. Then the intuitive result on the right of the figure can be obtained.

6.6.3 Oversampled hierarchical structure

The Powell-Sabin spline wavelets are critically sampled wavelets. This means that details are only computed in the new vertices and the total number of coefficients before and after the wavelet transform remains the same, namely one coefficient for each vertex. The length of $\mathbf{s}^j$ plus the length of $\mathbf{w}^j$ equals the length of $\mathbf{s}^{j+1}$. This kind of multiresolution representations have had limited use in computer graphics so far because of the discussed stability problems. We proposed a solution and proved the stability.

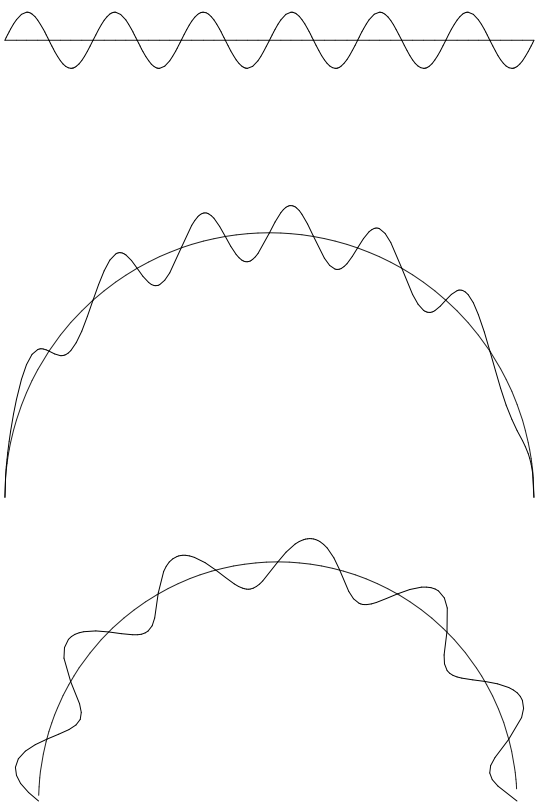


Figure 6.4: When the geometry is given as a vector valued function a more geometric definition of the detail information is necessary.

Another solution could be to use an oversampled multiresolution pyramid instead of a critically sampled pyramid. The low pass information $\mathbf{s}^j$ remains the same but the detail vector $\mathbf{w}^j$ then contains the same number of coefficients as the original fine level data $\mathbf{s}^{j+1}$. There is a detail coefficient for every new vertex and also for every old vertex instead of only for every new vertex. The total number of coefficients grows during the transform. The transform is redundant, but does not suffer from stability problems.

6.7 Concluding remarks

The complement space captures the detail that is lost when going from a fine scale representation to a coarser approximation. We found wavelet functions by projecting the scaling functions in the old vertices onto the desired complement space. The projection is not necessarily orthogonal, but it is known that an orthogonal decomposition automatically leads to a stable wavelet transform and to a minimal approximation error in the least squares sense. It is also important to have at least one primal vanishing moment. We proposed an update step for the lifting scheme where the update coefficients are the solution of a least squares problem with an equality constraint.

Chapter 7

Generalized Subdivision

7.1 Introduction

Spline surfaces have an inherent drawback: by definition, they have to be homomorphic to pieces of the plane. This forces designers to divide surfaces into patches that have to be stitched together. When building complex models with many patches the preservation of smoothness across patch boundaries can be quite cumbersome and expensive and it is difficult to avoid seams. Furthermore, when fine details have to be represented, more and smaller patches are needed.

Another popular choice is the use of polygon models. Given that most hardware rendering ultimately resolves into converting to triangles, even for patches, polygonal meshes are a very basic primitive. They can represent arbitrary topologies without stitching together patches and can describe fine details very accurately. But because of the enormous number of polygons needed to represent realistic surfaces, they are difficult to manipulate and not well suited for designing surfaces with smooth changes over larger regions.

Subdivision surfaces provide a possible solution that combines the advantages of these two approaches. They are defined procedurally as the limit of a refinement process on a coarse scale control polyhedron. Due to the procedural definition, freedom from the closed form mathematical expression is achieved and a wide variety of surface types can be expressed. These surfaces are no longer restricted to being images of bivariate functions, therefore they can easily represent shapes of arbitrary topology. Furthermore they support multiresolution algorithms as discussed in Chapter 4.

When the subdivision process is iterated, the control net will contain more and more vertices. In the case of spline subdivision the control net converges to the spline surface described by the original control net. However, by the procedural definition, we can also define a subdivision process by an arbitrary linear map in the space of control points without knowing something a priori about the limit surface. If the sequence of control nets converges, one can use this procedure to generate different surface types.

Subdivision schemes that generalize B-spline representations are particularly useful [78, 9]. In the regular regions away from the extraordinary vertices the limit surface will be a spline surface and the basis functions are known. Spline based subdivision schemes are typically approximating. However, interpolating subdivision schemes for which the limit surface interpolates the initial control points are also of interest because the ability to control the resulting surface exactly is important in many applications. Modifications of approximating schemes have been proposed to force the limit surface to interpolate particular points and normals, but the resulting masks have infinite size [116] or the interpolation constraints are only satisfied in the limit [89, 60]. Other schemes [47, 66] exist that are interpolating by design, but these are not spline based.

In this chapter we present a new subdivision scheme that is spline based, but the limit surface interpolates a particular point and its unit normal for each initial vertex. The unit normal is a normalized vector perpendicular to the surface. The scheme is a generalization of the dyadic uniform Powell-Sabin spline subdivision [125] that was summarized in Section 5.2.1.

The triadic subdivision for non uniform Powell-Sabin splines of Section 5.4 is not suited for generalization to arbitrary topology because the coefficients in the linear combinations depend on the parameter plane and can be different for every new vertex. This requires an extra amount of bookkeeping; we explicitly want to avoid this here. We are interested in semi-uniform subdivision and do not consider the non uniform setting. However, a generalization of non uniform spline schemes exists [106] and has the advantage that certain features and shapes can more easily be modeled.

For each vertex there is a control triangle instead of a control point, and this control triangle is tangent to the limit surface. The scheme is interpolating in the tangent point, though the corners of the triangle vary with each iteration. The new control triangle lies in the same plane as the old control triangle, and therefore the scheme also interpolates the normals. We also give a convergence analysis based on the eigenanalysis of the subdivision matrix and the properties of the characteristic map [96, 90].

7.2 Theoretical background

In this section we give an overview of the theoretical background on the analysis of subdivision schemes. We restrict ourselves to the aspects that are relevant for the convergence proof of the tangent scheme later in this chapter. The approach is inspired by papers by Reif [96], Peters [90] and Umlauf [114].

7.2.1 Base polyhedron and refinement rule

A subdivision surface consist of two main ingredients: a coarse base polyhedron that controls the overall shape of the surface, and a refinement rule to compute in-between control points on a finer scale. The surface is then defined as the limit of a sequence of successive refinements of the initial control polyhedron or polygonal mesh. Each control point on level $j+1$ can be computed as a linear combination of control points of the polyhedron on the previous level j. Figure 7.1 shows an example in which each triangle in the original mesh is split into four new triangles, quadrupling the number of triangles in the mesh.

The number of edges connected to a vertex may be different from vertex to vertex. As a result the rules derived from the spline basis functions or interpolation constraints can be applied only to those parts of the mesh that are locally regular; that is only to those vertices that have a valence of six for triangular schemes. We are dealing with semi-uniform subdivision. For the vertices with other valences, the extraordinary vertices, additional rules are needed. This is the main challenge in the design of subdivision schemes for surfaces: one has to define additional rules for irregular parts of the mesh in such a way that the limit surfaces have desired properties, in particular, are smooth.

7.2.2 Union of surface rings

A control net of three-dimensional control points will consist mostly of large regular regions, where standard continuity conditions can be used to describe a smooth surface. There are also isolated irregular regions, where these conditions are too restrictive. The key observation is that semi-uniform subdivision enlarges the regular regions of the control net and shrinks the irregular regions. The special rules used near the extraordinary vertices, do not change the number of extraordinary vertices in two consecutive meshes $\mathbf{p}^j$ and $\mathbf{p}^{j+1}$. The new vertices all have valence 6.

Since the subdivision masks have a fixed finite size, we can restrict the analysis to nets $\mathbf{p}^0$ with a single extraordinary vertex surrounded by r rings of ordinary vertices. Figure 7.2 shows an example of an extraordinary vertex with valence 5 surrounded by 3 rings of ordinary vertices. The number of rings r must be chosen

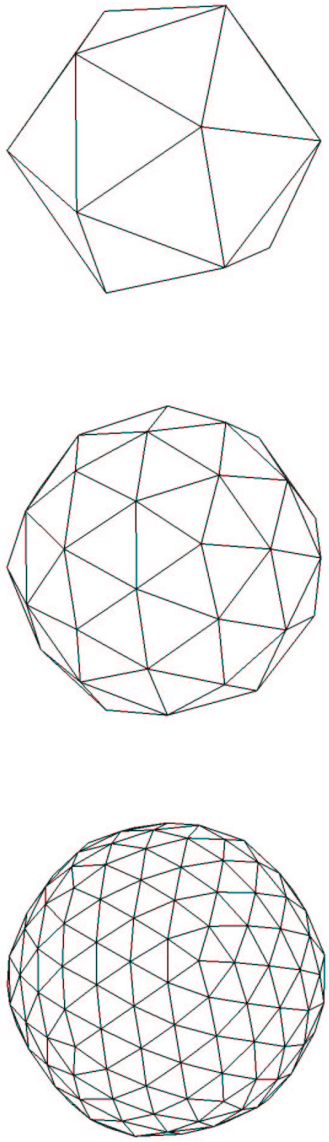


Figure 7.1: Example of a subdivision surface. The initial triangular mesh approximates the surface. Each triangle is split into 4, and the subdivision rule determines how the new vertices are computed.

such that the regular parts of $\mathbf{p}^0$ define at least a closed part of the surface that surrounds the extraordinary point. In the example of Figure 7.2 the regular parts of the domain are indicated in gray. The specific value of r that is needed to represent a surface part around the extraordinary point depends on the meaning of the points in $\mathbf{p}^0$, thus on the representation and the regular basis functions that underlie the subdivision scheme.

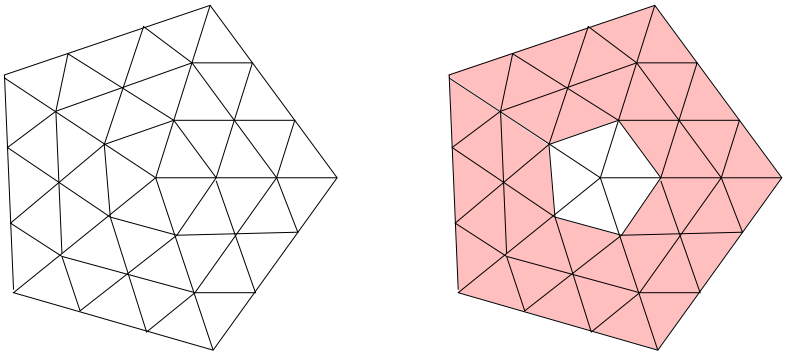


Figure 7.2: An extraordinary vertex of valence 5 surrounded by 3 rings of ordinary vertices. The regular parts marked in gray have to define a part of the surface that completely surrounds the extraordinary point.

If we denote by $\boldsymbol{s}^j$ the surface that corresponds to the regular parts of $\mathbf{p}^j$, then obviously $\boldsymbol{s}^{j-1}$ is part of $\boldsymbol{s}^j$ for $j \geq 1$, because in each iteration the regular part is extended towards the extraordinary vertex

$$\boldsymbol{s}^0 \subset \boldsymbol{s}^1 \subset \boldsymbol{s}^2 \subset \cdots \tag{7.1}$$

and the limiting surface is given by $\boldsymbol{s} = \cup \boldsymbol{s}^j$. Taking $\boldsymbol{s}^{j-1}$ away from $\boldsymbol{s}^j$ we obtain a surface ring $\boldsymbol{r}^j$ which is added to $\boldsymbol{s}^{j-1}$ in the jth iteration step. This yields

$$\boldsymbol{s} = \boldsymbol{s}^0 \cup \bigcup_{j \geq 1} \boldsymbol{r}^j. \tag{7.2}$$

Figure 7.3 illustrates the above definitions with an example for which $r = 2$. The shaded triangles on the left are the domain of the surface ring $\boldsymbol{r}^{j-1}$ and the shaded triangles on the right are the domain of the surface ring $\boldsymbol{r}^j$. The irregular part surrounding the extraordinary point shrinks and the hole gradually becomes filled with smaller surface rings that together in the limit form the subdivision surface.

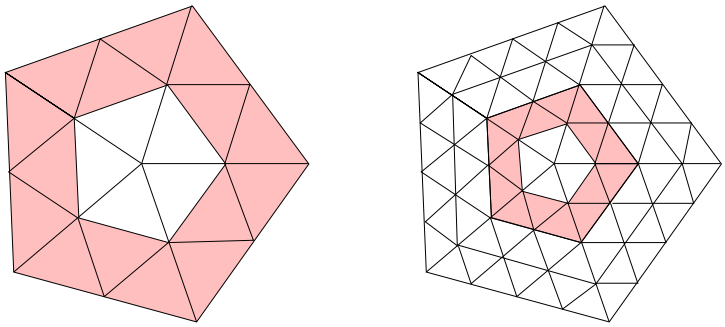


Figure 7.3: In each iteration a surface ring is added to the regular part of the surface.

At an extraordinary vertex of valence n the surface rings $\boldsymbol{r}^j$ can be parameterized over a common domain $\Omega \times \mathbb{Z}_n$ where Ω is (Figure 7.4)

$$\Omega = \{(u,v) | u,v \geq 0 \text{ and } 1 \leq u+v \leq 2\}. \tag{7.3}$$

and $\mathbb{Z}_n = \{0,1,\ldots,n-1\}$. It consist of n copies of the unit Ω. A vertex in this parameter domain is characterized by a triple (u,v,k) with $(u,v) \in \Omega$ and $k \in \mathbb{Z}_n$.

The control points that define the ring $\boldsymbol{r}^j$ form a subnet $\mathbf{d}^j \in \mathbf{p}^j$. For the regular parts there are basis functions that can be used to represent the surface. In the specific case that we will present in the next section, we use Powell-Sabin spline functions. Other examples that are based on spline basis function are the Loop scheme [78] and the Catmull-Clark scheme [9]. We arrange the basis functions corresponding to $\mathbf{d}^j$ in a row vector $\mathbf{B}^j$ and B_i^j is the ith basis function corresponding to the ith control point $\boldsymbol{d}_i^j$ on level j. The parameterization of a surface ring is

$$\boldsymbol{r}^j : \Omega \times \mathbb{Z}_n \to \mathbb{R}^3 : (u,v,k) \to \boldsymbol{r}^j(u,v,k) = \sum_i \boldsymbol{d}_i^j B_i^j(u,v,k) =: \mathbf{B}^j \mathbf{d}^j. \tag{7.4}$$

The parts $\boldsymbol{r}_k^j(u,v) = \boldsymbol{r}^j(u,v,k)$ are the image of each time one unit domain. They are referred to as segments and together they form the surface ring $\boldsymbol{r}^j$. The control points that define one segment of a surface ring are a subset of $\mathbf{d}^j$ and their number depends on the underlying representation.

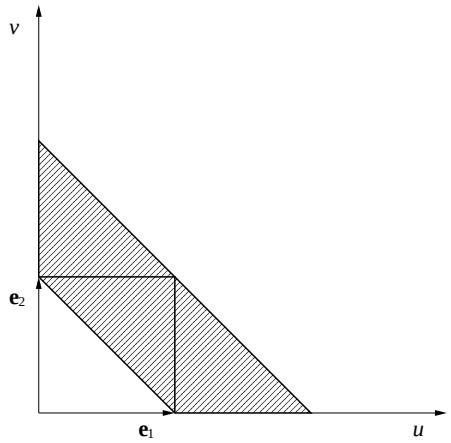


Figure 7.4: The unit domain Ω

7.2.3 The subdivision matrix and the characteristic map

The subdivision algorithm can be described by a square subdivision matrix S^j that relates the control points in the neighborhood of a vertex on two sequential levels

$$\mathbf{d}^j = S^j \mathbf{d}^{j+1}. \tag{7.5}$$

This matrix is a submatrix of the subdivision matrix P^j used in the previous chapters. A separate matrix is defined for each valence. The size of the square subdivision matrix S^j depends on the number of control points in $\mathbf{d}^j$ that is needed to define one surface ring.

We assign numbers to the vertices in $\mathbf{d}^j$. After one subdivision step the configuration of the vertices in $\mathbf{d}^{j+1}$ is exactly similar to the configuration of the old vertices in $\mathbf{d}^j$. We can establish a one-to-one correspondence between the vertices before and after subdivision and we can use the same numbering system. This is illustrated in Figure 7.5 around an extraordinary vertex with valence 3. The numbering in this figure is chosen arbitrary and only to illustrate the concept.

In general the subdivision matrix can change from level to level. Most schemes are stationary which means that the same affine combinations are used in every step j of the iteration and the subdivision matrix $S = S^j$ is independent of the level j.

Definition 7.1 *Let $\lambda_0, \ldots, \lambda_K$ be the eigenvalues of S ordered by their modulus*

$$|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_K| \tag{7.6}$$

and denote by $\mathbf{v}_0, \ldots, \mathbf{v}_K$ the corresponding eigenvectors.

In the remainder we shall assume that $\lambda_1 = \lambda_2$ with geometric and algebraic multiplicity 2. Such a double subdominant eigenvalue appears often in subdivision schemes, and also in the scheme proposed later in this chapter. We do not consider the case where λ_1 is different from λ_2.

Denote by $[\mathbf{v}_1, \mathbf{v}_2]$ the two dimensional mesh with as two dimensional control points each ith entry of $\mathbf{v}_1$ combined with the ith entry of $\mathbf{v}_2$.

Definition 7.2 *If $|\lambda_0| \geq |\lambda_1| = |\lambda_2| > |\lambda_3|$ we define the characteristic map Υ as the map from n times the unit domain to the two dimensional surface with $[\mathbf{v}_1, \mathbf{v}_2]$ as control mesh*

$$\Upsilon : \Omega \times \mathbb{Z}_n \to \mathbb{R}^2 : (u, v, k) \to \mathbf{B}(u, v, k)[\mathbf{v}_1, \mathbf{v}_2], \tag{7.7}$$

or in complex form as

$$\Upsilon^* : \Omega \times \mathbb{Z}_n \to \mathbb{C} : (u, v, k) \to \mathbf{B}(u, v, k)\mathbf{v}^*, \quad \mathbf{v}^* := \mathbf{v_1} + i\mathbf{v_2}. \tag{7.8}$$

Note that Υ can be regarded as consisting of n segments $\Upsilon_k(u, v) := \Upsilon(u, v, k)$, $k = 0, 1, \ldots, n-1$. We will need the following definitions.

Definition 7.3 *The characteristic map Υ is called regular if*

$$\det\left(\frac{\partial \Upsilon(u, v, j)}{\partial(u, v)}\right) \neq 0 \quad \textit{for all} \quad (u, v, j) \in \Omega \times \mathbb{Z}_n \tag{7.9}$$

and injective if it maps distinct points to distinct points

$$\Upsilon(u_1, v_1, k_1) = \Upsilon(u_2, v_2, k_2) \Leftrightarrow (u_1, v_1, k_1) = (u_2, v_2, k_2). \tag{7.10}$$

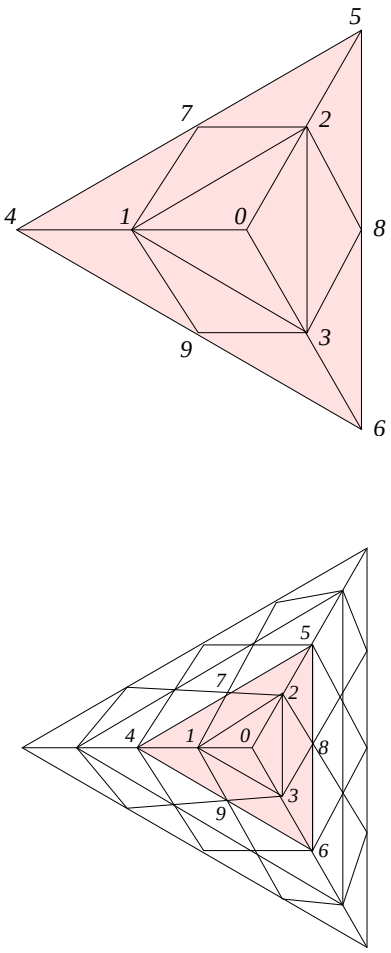


Figure 7.5: The vertices before and after subdivision have the same configuration.

7.2.4 Main theorem

Most subdivision schemes are constructed from regular schemes, which are known to produce at least C^1 continuous surfaces for almost any initial configuration of control points. For the spline based systems, we know that the exact form of the limit surface in regular regions must be the spline surface itself, because it consists of pieces each influenced by a finite number of control points that are all regularly connected. The level of continuity is thereby well defined. For systems which are not spline based we cannot make this deduction straight away, but an analogous statement can be made after applying the same analysis as for the irregular points.

The main mathematical challenge in understanding the nature of subdivision surfaces is determining the behavior around the extraordinary points. This was first analyzed for quadrilateral schemes by Doo and Sabin [44] using Fourier transforms and an eigenanalysis of the subdivision process. This work was expanded upon by Ball and Storry [5] who derived some necessary conditions for convergence to smooth surfaces. However, these are not sufficient because they only prove tangent plane continuity near the extraordinary point and self intersections are still possible. Reif [96] derived necessary and sufficient smoothness conditions by taking into account the eigenproperties of the subdivision matrix as well as the properties of the regular basis functions.

The crucial theorem for the analysis of subdivision algorithms can be stated in terms of the subdominant eigenvalue $\lambda := \lambda_1 = \lambda_2$ of S and the characteristic map Υ.

Theorem 7.1 *If λ is a real eigenvalue of the subdivision matrix S with geometric and algebraic multiplicity 2 with $1 > |\lambda| > |\lambda_3|$ and if the characteristic map Υ is regular and injective, then the limit surface is C^1 continuous for almost all initial control nets $\mathbf{d}^0$.*

Proof: see [96]. □

A generalization with a less severe restriction on the eigenvalues can be found in [95].

Checking if the characteristic map is regular and injective can be quite cumbersome. Therefore we first discuss sufficient conditions for regularity and injectivity in the case of symmetric schemes.

7.2.5 Symmetric schemes

Assuming a double subdominant eigenvalue is less restrictive than it might seem at first since this case appears in a natural way for symmetric schemes.

Definition 7.4 *A subdivision scheme is said to be symmetric, if it is invariant under shifts and reflections of the labelling of the vertices in* $\mathbf{d}_m$*. The permutation matrices* T *and* R *are characterized by*

$$\begin{aligned} \mathbf{B}(u,v,j+1)\mathbf{d}^j &= \mathbf{B}(u,v,j)T\mathbf{d}^j \\ \mathbf{B}(v,u,-j)\mathbf{d}^j &= \mathbf{B}(u,v,j)R\mathbf{d}^j \end{aligned} \tag{7.11}$$

Symmetry of the subdivision scheme means that the subdivision matrix S *commutes with* T *and* R*:*

$$ST = TS \quad \textit{and} \quad SR = RS. \tag{7.12}$$

Symmetry implies that the subdivision matrix S has a block-circulant structure with square blocks $S_k, k \in \mathbb{Z}_n$. This can be seen from the fact that choosing another segment as the first segment should not change the subdivision matrix for symmetric schemes. The size of the blocks S_k depends on the number of control points to represent one segment of a surface ring.

Thus S is unitary similar to a block-diagonal matrix $\hat{S}$. The diagonal blocks of $\hat{S}$ result from S_k by the discrete Fourier transform

$$\hat{S}_k = \sum_{l=0}^{n-1} \omega_n^{-kl} S_l \quad \text{for } k = 0, \cdots, n-1, \tag{7.13}$$

where $\omega_n = e^{\frac{2\pi i}{n}}$ denotes an n-th root of unity. If $\hat{\mathbf{v}}$ is an eigenvector of some block $\hat{S}_k$ corresponding to the eigenvalue μ, then μ is also an eigenvalue of S with complex eigenvector

$$\mathbf{v}^* = \begin{bmatrix} \omega_n^0 \hat{\mathbf{v}} \\ \omega_n^k \hat{\mathbf{v}} \\ \vdots \\ \omega_n^{k(n-1)} \hat{\mathbf{v}} \end{bmatrix}. \tag{7.14}$$

Because S is real, the blocks of $\hat{S}$ satisfy $\hat{S}_{n-k} = \overline{\hat{S}}_k$ for $k = 1, \cdots, \lfloor n/2 \rfloor$. Hence if μ is an eigenvalue of $\hat{S}_k$ it is also an eigenvalue of $\hat{S}_{n-k}$ and there are always two linear independent, real eigenvectors $\mathbf{v}_1 = Re(\mathbf{v}^*)$ and $\mathbf{v}_2 = Im(\mathbf{v}^*)$.

It can be proven [90] that if the characteristic map is injective, the subdominant eigenvalue stems form the blocks $\hat{S}_1$ and $\hat{S}_{n-1}$. From now on we denote with $\hat{\mathbf{v}}$ the eigenvector of the subdominant eigenvalue λ corresponding to $\hat{S}_1$.

Definition 7.5 *A triangular subdivision scheme is said to have a normalized characteristic map, if the eigenvector $\hat{\mathbf{v}}$ of $\hat{S}_1$ corresponding to λ is scaled such that $\Upsilon_0(\mathbf{e}_1 + \mathbf{e}_2) = (p, 0)$ with a scalar $p > 0$, where $\mathbf{e}_1$ and $\mathbf{e}_2$ are unit vectors in the u and v direction as in Figure 7.4.*

For symmetric schemes we can restrict the analysis of the characteristic map to a single segment Υ_0. Let $\Upsilon_0 = [\Upsilon_0^1, \Upsilon_0^2]$ and denote by $\Upsilon_{0,v} := [\Upsilon_{0,v}^1, \Upsilon_{0,v}^2]$ the partial derivative of Υ_0 with respect to v.

Theorem 7.2 *If the normalized characteristic map Υ of a symmetric subdivision scheme satisfies*

$$\Upsilon_{0,v}(u, v) > 0 \quad \textit{for all} \quad (u, v) \in \Omega \tag{7.15}$$

componentwise, then the characteristic map is regular and injective.

Proof: This was proven for quadrilateral schemes in [90] and extended to triangular schemes in [114]. □

7.3 The tangent scheme

In this section we introduce the subdivision rules for the tangent scheme. Because it is based on uniform Powell-Sabin splines it works with control triangles that are tangent to the limit surface in their barycenter. Hence the name tangent scheme.

7.3.1 Ordinary vertices

For the ordinary vertices with valence six we use the subdivision rules for uniform dyadic Powell-Sabin subdivision [125] that were summarized in Section 5.2.1. Here we will use these rules in matrix form.

For a new vertex inserted on the bottom edge as in Figure 7.6, top panel, the new control triangle is found as follows

$$\boldsymbol{c}_{ij}^{j+1} = B\,\boldsymbol{c}_i^j + C\,\boldsymbol{c}_j^j \tag{7.16}$$

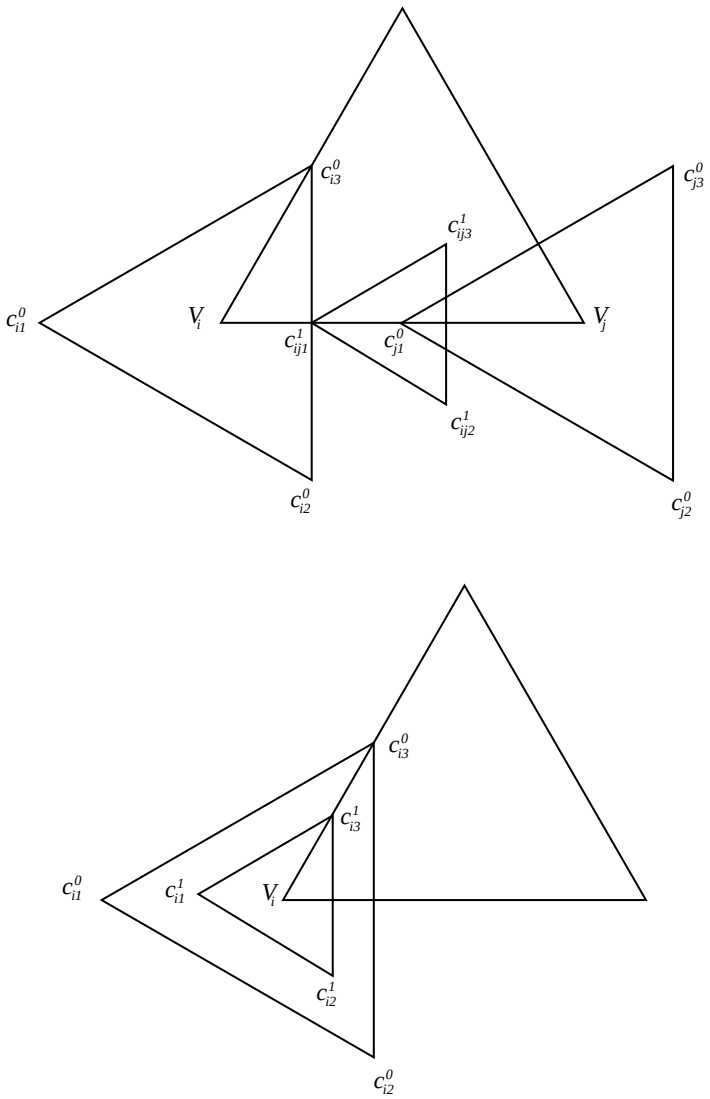


Figure 7.6: Subdivision of uniform Powell–Sabin splines. Top: the control triangle for a new vertex between two old vertices. Bottom: the control triangle for the old vertices is rescaled.

with

$$B = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 1/4 & 0 \\ 1/2 & 0 & 1/4 \end{bmatrix}. \tag{7.17}$$

The control triangle for the old vertices needs to be rescaled

$$\boldsymbol{c}_i^{j+1} = A\,\boldsymbol{c}_i^j \tag{7.18}$$

with

$$A = \begin{bmatrix} 2/3 & 1/6 & 1/6 \\ 1/6 & 2/3 & 1/6 \\ 1/6 & 1/6 & 2/3 \end{bmatrix}. \tag{7.19}$$

This is illustrated in Figure 7.6, bottom panel. As mentioned before, the tangent point remains the same.

For the derivation of these rules it was assumed that all control triangles point in the same direction and that their corners are numbered in the same order. However, in the neighborhood of the extraordinary vertices where the different regular parts come together, it becomes impossible to request this. Therefore we also derive rules for other configurations of the control triangles. For example the new control triangle in Figure 7.7 is found as

$$\boldsymbol{c}_{jk}^{j+1} = K\,\boldsymbol{c}_j + L\,\boldsymbol{c}_k \tag{7.20}$$

with

$$K = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 \end{bmatrix}, \quad L = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \end{bmatrix}. \tag{7.21}$$

These rules can easily be derived from the underlying Bézier representation. The matrix A for the rescaling of the old control triangle in Equation (7.18) is independent of the orientation. Equations (7.16), (7.18) and (7.20) are matrix combinations because there are three control points for each vertex.

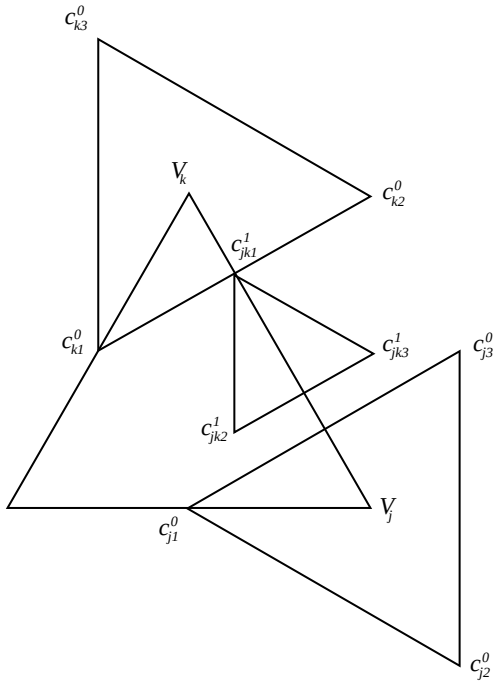


Figure 7.7: Rules for different orientations of the control triangles can be derived.

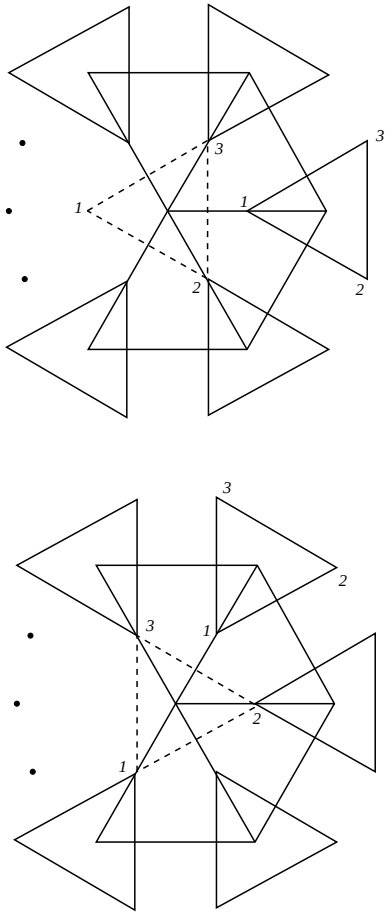


Figure 7.8: Rotation of the control triangle of the central vertex.

To simplify the analysis we reduce the number of rules that have to be derived by introducing the rotation matrix Q_6

$$Q_6 = \begin{bmatrix} 2/3 & 2/3 & -1/3 \\ -1/3 & 2/3 & 2/3 \\ 2/3 & -1/3 & 2/3 \end{bmatrix}. \tag{7.22}$$

The effect of this matrix on a control triangle is a rotation of $2\pi/6 = \pi/3$ radians. We can now, after applying Q_6 an appropriate number of times, use Equation (7.16) to compute a new vertex on every edge adjacent to the centroid. For example in Figure 7.8, top panel, the control triangle is chosen such that Equation (7.16) can be used for a new vertex on the horizontal edge between the numbered control triangles. In the bottom panel we have first rotated the central control triangle $\pi/3$ radians and now we can use the same equation for a new vertex on the diagonal edge between the numbered control triangles.

7.3.2 Extraordinary vertices

For the new rules in the extraordinary vertices we define a similar rotation matrix Q_n. It defines a rotation over an angle of $2\pi/n$ radians

$$Q_n = \begin{bmatrix} a & b & c \\ c & a & b \\ b & c & a \end{bmatrix} \quad \text{with} \quad \begin{aligned} a &= \tfrac{1}{3} + \tfrac{2}{3}\cos\left(\tfrac{2\pi}{n}\right) \\ b &= \tfrac{1}{3} - \tfrac{1}{3}\cos\left(\tfrac{2\pi}{n}\right) + \tfrac{1}{\sqrt{3}}\sin\left(\tfrac{2\pi}{n}\right). \\ c &= \tfrac{1}{3} - \tfrac{1}{3}\cos\left(\tfrac{2\pi}{n}\right) - \tfrac{1}{\sqrt{3}}\sin\left(\tfrac{2\pi}{n}\right) \end{aligned} \tag{7.23}$$

We now use exactly the same rules as in the uniform case, but with Q_n instead of Q_6.

7.4 Convergence analysis

To do the convergence analysis of the tangent scheme we first look at the structure of the subdivision matrix and calculate its eigenvalues. The subdominant eigenvalue will turn out to be equal to $1/2$ and stems from the blocks $\hat{S}_1$ and $\hat{S}_{n-1}$. For the analysis of the characteristic map we resort to the sufficient condition that the partial derivative of the first segment with respect to v should be positive. We prove that this condition is satisfied and conclude that the limit surface of the tangent scheme is C^1 continuous for almost all initial control nets $\mathbf{d}^0$.

7.4.1 The subdivision matrix

The form of the subdivision matrix depends on the number of the vertices in the control net $\mathbf{d}^j$ and their labelling. For the tangent scheme only two rings of ordinary vertices around the vertex of interest are needed to do the analysis. This follows from the fact that the support of a Powell-Sabin spline basis function only consists of the triangles that contain the involved vertex and none of the neighboring triangles further away influences the surface.

We label the control points in $\mathbf{d}^j$ segment after segment as in Figure 7.9. The central control point has number 0. In each segment their are 5 vertices. The first and second are also part of the previous segment and the fourth and the fifth are also part of the next segment. In each segment we number three vertices and continue to the next segment. So the total number of vertices equals three times the number of segments n.

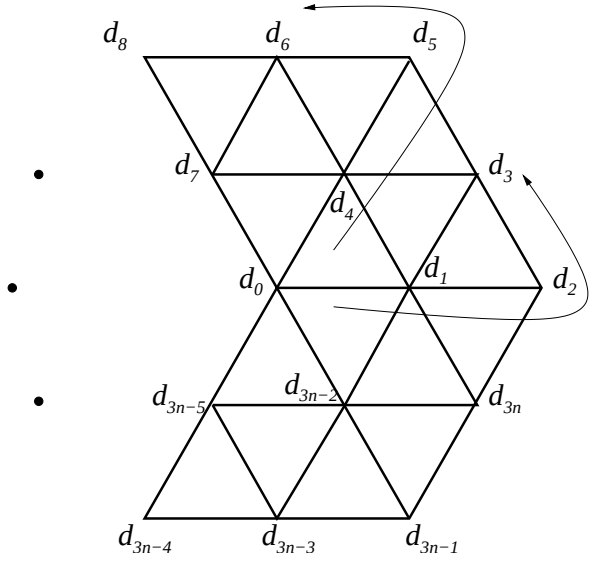


Figure 7.9: Labelling of the vertices

To achieve the desired periodic structure, we replace the central control point $\boldsymbol{d}_0$ by n rotated copies $\boldsymbol{d}_{0,k}, k = 1, \ldots, n$

$$\boldsymbol{d}_0^j = \frac{1}{n} \sum_{k=0}^{n-1} Q_n^{-j} \boldsymbol{d}_{0,k}^j. \tag{7.24}$$

Although not strictly necessary, we also replace the control points $\boldsymbol{d}_{3i+1}^j$ and $\boldsymbol{d}_{3i+2}^j$ by two identical copies to facilitate the analysis of only a single segment of the characteristic map.

Then S has a block circulant structure

$$S = \begin{bmatrix} S_0 & S_1 & \cdots & S_{n-1} \\ S_{n-1} & S_0 & \cdots & S_{n-2} \\ \vdots & & \ddots & \vdots \\ S_1 & \cdots & S_{n-1} & S_0 \end{bmatrix} \tag{7.25}$$

where

$$S_0 = \begin{bmatrix} \frac{1}{n}A & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}B & \frac{1}{2}C & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}A & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}K & 0 & 0 & \frac{1}{2}L & 0 \\ \frac{1}{n}BQ_n^{-n+1} & 0 & 0 & 0 & \frac{1}{2}C & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}A & 0 \end{bmatrix}, \tag{7.26}$$

$$S_1 = \begin{bmatrix} \frac{1}{n}AQ_n^{-1} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}BQ_n^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}L & 0 & 0 & 0 & 0 \\ \frac{1}{n}B & \frac{1}{2}C & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}A & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.27}$$

and

$$S_k = \begin{bmatrix} \frac{1}{n}AQ_n^{-k} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}BQ_n^{-k} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}BQ_n^{-k+1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for} \quad k = 2, \cdots, n-1. \tag{7.28}$$

Now we can use the discrete Fourier transform from Equation (7.13) to yield a unitary similar block-diagonal matrix $\hat{S}$ with blocks

$$\hat{S}_k = \begin{bmatrix} \frac{1}{n}A\sum_{l=0}^{n-1}\omega_n^{-kl}Q_n^{-l} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{n}B\sum_{l=0}^{n-1}\omega_n^{-kl}Q_n^{-l} & \frac{1}{2}C & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}A & 0 & 0 & \frac{1}{2}\omega_n^{-k(n-1)}A & 0 \\ 0 & \frac{1}{2}L + \frac{1}{2}\omega_n^{-k}K & 0 & 0 & \frac{1}{2}L + \frac{1}{2}\omega_n^{-k(n-1)}K & 0 \\ \frac{1}{n}B\sum_{l=0}^{n-1}\omega_n^{-kl}Q_n^{-(l-1)_n} & \frac{1}{2}\omega_n^{-k}C & 0 & 0 & \frac{1}{2}C & 0 \\ 0 & \frac{1}{2}\omega_n^{-k}A & 0 & 0 & \frac{1}{2}A & 0 \end{bmatrix} \tag{7.29}$$

for $k \in \mathbb{Z}_n$. With a computer algebra program we find the eigenvalues of the subdivision matrix S as in Table 7.1. The subdominant eigenvalue λ equals $1/2$ and stems from $\hat{S}_1$ and $\hat{S}_{n-1}$.

7.4.2 The characteristic map

What remains is the analysis of the characteristic map. Because of the convex hull property of the Powell-Sabin splines, the partial derivative of Υ_0 with respect to v will be positive, if all the Bézier coefficients in its Bézier representation are positive.

	multiplicity	eigenvalue
$\hat{S}_0$	1	1
	2	$\frac{1}{4}$
	15	0
$\hat{S}_1$ and $\hat{S}_{n-1}$	1	$\frac{1}{2}$
	2	$\frac{1}{4}$
	15	0
$\hat{S}_k$ for $k \neq 0, 1, n-1$	2	$\frac{1}{4}$
	16	0

Table 7.1: Eigenvalues of the blocks of the Fourier transformed subdivision matrix.

The normalized eigenvector of $\hat{S}_1$ corresponding to the subdominant eigenvalue $\lambda = \frac{1}{2}$ is $\hat{\mathbf{v}}$

$$\hat{\mathbf{v}} = \begin{bmatrix} -2 + 2\sqrt{3}i \\ -2 - 2\sqrt{3}i \\ 4 \\ 2 - 2\sqrt{3}i \\ 2 - 6\sqrt{3}i \\ 8 - 4\sqrt{3}i \\ 6 - 6\sqrt{3}i \\ 6 - 10\sqrt{3}i \\ 12 - 8\sqrt{3}i \\ 10 + 6\sqrt{3}i \\ 7 - 3\sqrt{3}i \\ 13 - 3\sqrt{3}i \\ 2 + 2\sqrt{3}i \\ 8 + 4\sqrt{3}i \\ 2 + 6\sqrt{3}i \\ 6 + 6\sqrt{3}i \\ 12 + 8\sqrt{3}i \\ 6 + 10\sqrt{3}i \end{bmatrix} \tag{7.30}$$

with i the imaginary unit.

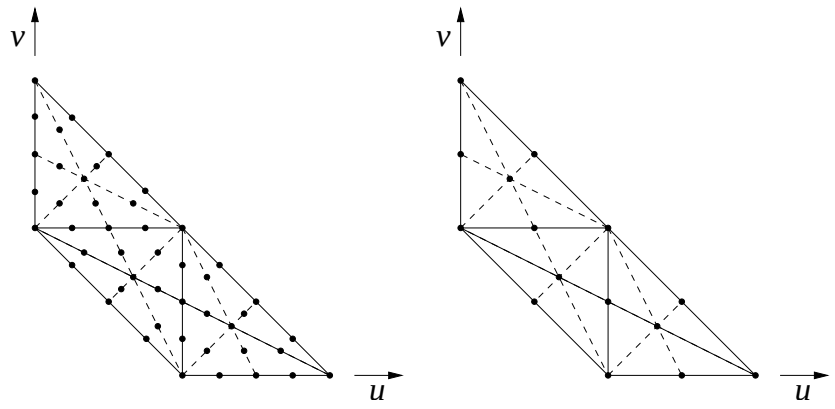


Figure 7.10: Left: the domain of the segment Υ_0^* of the characteristic map consist of 18 quadratic patches. Right: the domain of $\Upsilon_{0,v}^*$ consist of linear patches.

The segment Υ_0^* of the characteristic map consist of 18 quadratic patches. The domain of Υ_0^* is depicted in Figure 7.10 on the left. Each subtriangle is a quadratic Bézier triangle. The Bézier ordinates can be calculated from the B-spline coefficients of the normalized eigenvector $\hat{\mathbf{v}}$ in Equation (7.30). To this end we use the formulas (2.41) and (2.42). The result is given in Table 7.2. The ordering of the Bézier ordinates in the table corresponds to their position on Figure 7.10.

The partial derivative of a Bézier polynomial in the v direction can be calculated using Corollary 2.1 of the de Casteljau algorithm. The right panel of Figure 7.10 shows the domain of $\Upsilon_{0,v}^*$. It consists of 18 linear Bézier patches. The Bézier ordinates are given in Table 7.3.

We can see that both the real and the imaginary part of the Bézier coefficients of $\Upsilon_{0,v}^*$ are positive and condition (7.15) of Theorem 7.2 is satisfied. The result of the analysis is summarized in the following theorem.

Theorem 7.3 *The limit surface for the tangent scheme is C^1 continuous for almost all initial control nets $\mathbf{d}^0$.*

8 $+8i$								
7 $+7i$	8 $+\frac{2}{3}i$	9 $+7i$						
6 $+6i$	7 $+\frac{17}{3}i$	$\frac{23}{3}$ $+\frac{43}{9}i$	$\frac{17}{2}$ $+\frac{29}{6}i$	$\frac{19}{2}$ $+5i$				
5 $+5i$	6 $+\frac{14}{3}i$	$\frac{15}{2}$ $+236i$	9 $+3i$	10 $+3i$				
4 $+4i$	6 $+4i$	$\frac{29}{4}$ $+\frac{11}{4}i$	$\frac{17}{2}$ $+\frac{3}{2}i$	10				
5 $+3i$	6 $+\frac{10}{3}i$	7 $+\frac{5}{3}i$	8	$\frac{17}{2}$ $-\frac{3}{2}i$	9 $-3i$	10 $-3i$		
5	6	$\frac{23}{3}$	7 $-\frac{5}{3}i$	$\frac{29}{4}$ $\frac{11}{4}i$	$\frac{15}{2}$ $-\frac{23}{6}i$	$\frac{23}{3}$ $-\frac{43}{9}i$	$\frac{17}{2}$ $-\frac{29}{6}i$	$\frac{19}{2}$ $-5i$
		5 $-3i$	6 $-\frac{10}{3}i$	6 $-4i$	6 $-\frac{14}{3}i$	7 $-\frac{17}{3}i$	8 $-\frac{20}{3}i$	9 $-7i$
				4 $-4i$	5 $-5i$	6 $-6i$	7 $-7i$	8 $-8i$

Table 7.2: Bézier ordinates of Υ^*_0.

$2+2i$				
$2+2i$	$2+2i$	$\frac{5}{2}+\frac{5}{2}i$		
$2+2i$	$2+2i$	$3+3i$		
$3+3i$	$2+\frac{10}{3}i$	$\frac{5}{2}+\frac{5}{2}i$	$3+\frac{5}{3}i$	$\frac{7}{2}+\frac{3}{2}i$
		4	4	4

Table 7.3: Bézier ordinates of $\Upsilon^*_{0,v}$.

7.5 Implementation issues

In this section we briefly touch some practical implementation questions. We explain how the subdivision scheme knows which subdivision rule to apply. This happens with flags that contain information for each edge. We also argue that a preprocessing step that turns the input in the requested form would be useful.

7.5.1 Edgetags and numbering of points

The subdivision rules that have to be applied to find a new control triangle between two old control triangles depend on the specific configuration of the old control triangles. If the two control triangles do not have the same orientation, one of them has to be rotated an appropriate number of times. Furthermore it is possible that although the orientation is correct, the numbering of the three corners does not correspond. In that case the numbering of the corners in one of the old control triangles has to be changed.

All this information is coded bitwise in an integer edgetag for each edge of the control mesh. After subdivision a new edgetag has to be assigned to the newly created edges. Each original edge is split into two new edges, and the new edgetages are automatically computed based on the edgetag of the original edge.

As in most subdivision schemes the points of the control triangles and of the triangles in the mesh have to be ordered anti clockwise. We use this to know which are the neighboring triangles. For the visualization the normal has to point to the exterior of the surface.

7.5.2 Preprocessing

The initial input for the tangent scheme is not a connected set of control points but a connected set of control triangles. The barycenters of these control triangles form the triangular mesh. The advantage is that the control triangles are tangent to the limit surface. The drawback on the other hand is that this kind of input is not always available.

For the examples in the next section we have chosen the initial control triangles manually to illustrate some ideas. It would be interesting to have a preprocessing step that automatically calculates a possible control triangle for every point in a triangular mesh. Also the information that is contained in the edgetags has to be calculated. Such a preprocessing step is currently not available in our implementation.

7.6 Results

7.6.1 More design possibilities

Figure 7.11 shows the results of the tangent scheme with a tetraeder as the initial control polygon. The configuration of the control triangles in the top row leads to a surface one would expect from an interpolating subdivision scheme with the barycenters of the control triangles as initial control net. Note that the control triangles are not shown in the figures, only the connected barycenters in which the control triangles are tangent to the limit surface. The configuration of the bottom row starts with another steeper control triangle on the top of the tetraeder. The limit surface has a lob in the direction of the control triangle.

Also the size of the control triangles influences the limit surface. Figure 7.12 shows the same tetraeder with different sizes for the control triangles. The smaller the control triangle, the sharper the corners. When the control triangles grow the corners become more rounded.

7.6.2 Drawbacks

Like most schemes the tangent scheme only yields C^1 continuous surfaces. To achieve C^2 continuity the schemes either have very large support, or have zero curvature at extraordinary vertices [97]. Another problem is the presence of ripples in the neighbourhood of extraordinary vertices.

The tangent scheme is an expensive scheme when it comes to memory requirements. Instead of one control point we have three control points per vertex that need to be stored. Finally the question arises how the control triangles need to be determined in practical applications. The actual implementation of a preprocessing step will depend on the specific application.

7.6.3 Advantages

Because the tangent scheme interpolates the initial tangent points and normals, we can control the limit surface more exactly. The fact that, although the control points change each iteration, it is essentialy an interpolation scheme, facilitates the design of subdivision wavelets or applications like multiresolution editing.

For the graphical display of surfaces we need besides the points also estimates for the normals. These are used to calculate the color of the surface in the presence of light sources. With the tangent scheme we can easily calculate the exact

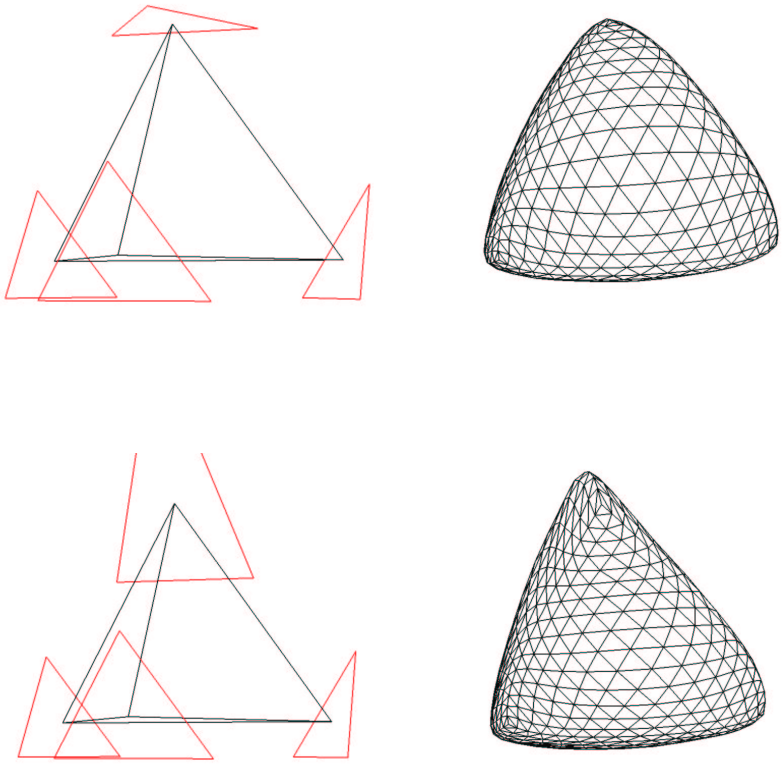


Figure 7.11: With the tangent scheme the normal on the limit surface in the inital points can be chosen in advance. On the second row, a steeper control triangle is used on top of the tetraeder. This yields a lob in the direction of the control triangle.

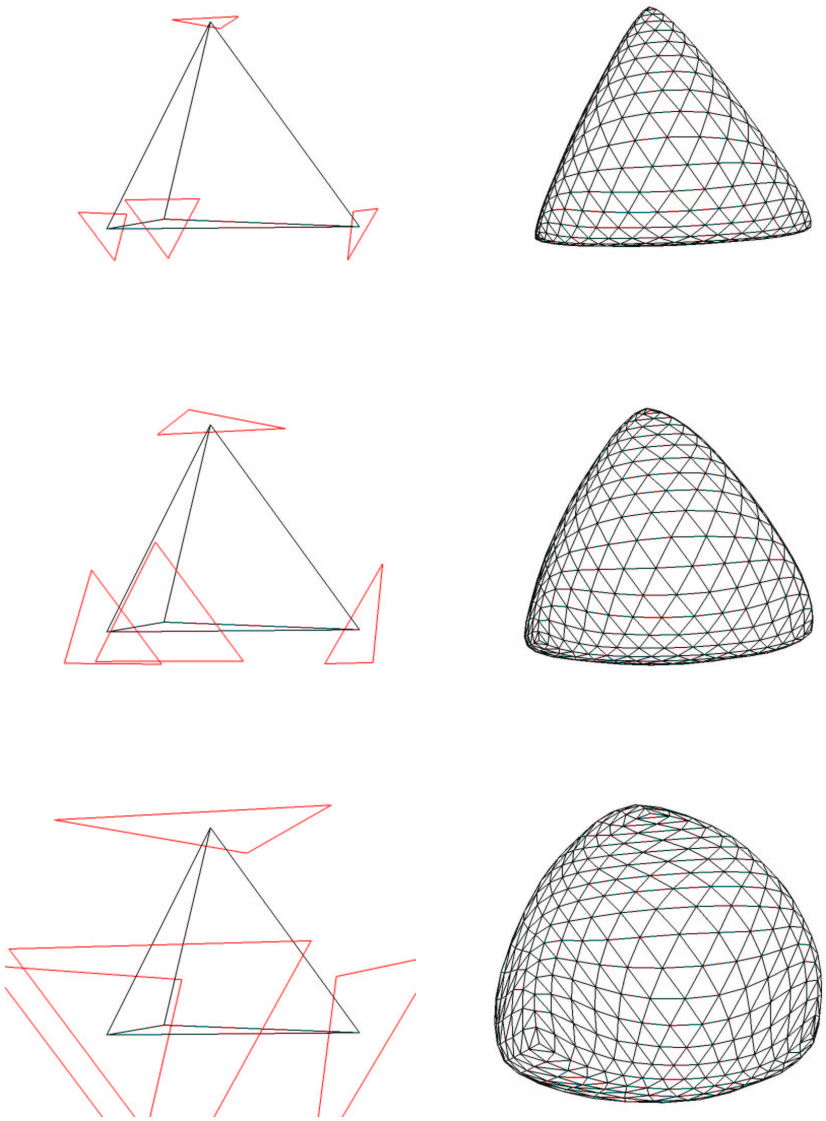


Figure 7.12: Effect of size of control triangles. Small control triangles yield sharp corners and large control triangles yiels rounded corners.

normales from the tangent planes. Using the true normals improves the rendering quality considerably.

The scheme is spline based. We have a closed mathematical expression for the surface in the regular regions and the limit positions on the surface are known exactly.

The stencil of the tangent scheme is compact, it uses only information of the two control triangles of the edge that is split in two new edges and no information of control points from neighbouring triangles. Therefore no special rules are needed for surfaces with a boundary.

7.7 Concluding Remarks

The tangent scheme is a new subdivision scheme that offers additional modeling functionality compared to traditional schemes. The control triangles are tangent to the surface, this means that the normal in a point can be chosen in advance. We think that this scheme will have many possibilities in CAGD and computer graphics applications.

Chapter 8

Conclusions

8.1 Overview of contributions

This dissertation investigated multiresolution techniques for Powell-Sabin splines in their normalized B-spline representation. After two chapters of introduction the description of our own results starts in Chapter 3 with some results on the B-spline basis. Chapter 4 overviews basic concepts on multiresolution techniques and we continue the discussion of our own results in Chapters 5, 6 and 7. In this chapter we recapitulate our contributions.

Stability of the B-spline basis

In Section 3.2 we proved that the set of basis functions $\{B_{ij}\}$ forms a stable basis for $S_2^1(\Delta^*)$

$$k_1\|\mathbf{c}\|_\infty \leq \|\sum_{i=1}^{n}\sum_{j=1}^{3} c_{ij}B_{ij}(x,y)\|_\infty \leq k_2\|\mathbf{c}\|_\infty$$

for all choices of the coefficient vector $\mathbf{c}$. We started with some lemma's on geometrical properties of Powell-Sabin refinements. Then we proved the stability for the max norm and finally for the p-norm. Essential in the proofs is that the area of the PS-triangle has to be bounded. We introduced the factor K that gives a measure of the size of the PS-triangle. For the p-norm the basis functions had to be normalized with respect to their support.

This section is joint work with colleague Jan Maes from the NALAG research group. We did not include his proof of Lemma 3.5, but refer to our paper [84]. In that paper he also gives an estimation of the constant K in the case of optimal PS-triangles.

A practical algorithm to compute PS-triangles

We gave an algorithm in Section 3.3 to calculate a PS-triangle with a bounded area that contains all the appropriate PS-points. From the stability analysis we know that it is important to have small PS-triangles. Furthermore we observed that when the size of the PS-triangles goes to infinity, the basis functions become linearly dependent. We denoted our result as nearly optimal in contrast to the optimal solution described by Dierckx in [41].

The nearly optimal algorithm yields PS-triangles that on average are 4.05% larger than the optimal PS-triangles. It was first published in [118] where we also described a connection between the representation of quadratic splines on Powell-Sabin splits with PS-triangles and control triangles, and a definition of basis functions based on minimal determining sets.

Dyadic subdivision not generally applicable

When trying to extend the uniform dyadic subdivision scheme [125] to non uniform settings we came across the limitations described in Theorem 5.2. The PS-refinement must satisfy inequality constraints of the form

$$\frac{1}{2} < \frac{(1-\lambda_{ij})\lambda_{ki}}{b_{ijk}\lambda_{ki} + c_{ijk}(1-\lambda_{ij})} < 1.$$

Otherwise the lines of C^2 discontinuity cannot be maintained in the refined triangulation. This is necessary because the aim of spline subdivision is to represent exactly the same surface on a finer level.

These conditions are not satisfied for every PS-refinement and therefore dyadic subdivision is not always possible. Furthermore, if a triangulation with its PS-refinement does satisfy the requirements, it is not guaranteed that the requirements are satisfied on the next finer level. We did not elaborate this further because of the limited usefulness.

$\sqrt{3}$ subdivision for non uniform splines

As an alternative for the dyadic refinement procedure we proposed a $\sqrt{3}$ scheme in Section 5.3. When applied to a regular triangulation of the plane, it produces a triangulation scaled by the factor $\sqrt{3}$ instead of the factor 2 for dyadic subdivision. The rules for uniform $\sqrt{3}$ subdivision are straightforward to derive starting from the underlying Bézier representation. We did not discuss them in this thesis but refer to [121].

The advantage of the $\sqrt{3}$ refinement lies in the application to non uniform triangulations [122]. The new vertices have to be chosen on the interior points, but this does not conflict with the existing PS-refinement. The new PS-refinement can always be chosen such the subdivided spline surface is a valid Powell-Sabin surface. The proof for the new control triangle (Theorem 5.3)

$$\begin{aligned}
\boldsymbol{c}_{ijk,1}^{\sqrt{3}} &= \tilde{L}_{i1}\boldsymbol{c}_{i1} + \tilde{L}_{i2}\boldsymbol{c}_{i2} + \tilde{L}_{i3}\boldsymbol{c}_{i3} \\
\boldsymbol{c}_{ijk,2}^{\sqrt{3}} &= \tilde{L}_{j1}\boldsymbol{c}_{j1} + \tilde{L}_{j2}\boldsymbol{c}_{j2} + \tilde{L}_{j3}\boldsymbol{c}_{j3} \\
\boldsymbol{c}_{ijk,3}^{\sqrt{3}} &= \tilde{L}_{k1}\boldsymbol{c}_{k1} + \tilde{L}_{k2}\boldsymbol{c}_{k2} + \tilde{L}_{k3}\boldsymbol{c}_{k3}
\end{aligned}$$

is based on the underlying Bézier representation and is in cooperation with former colleague Joris Windmolders.

Triadic subdivision

Applying the $\sqrt{3}$ scheme twice yields a triadic scheme, every original edge is split in three new edges and every original triangle is split in nine new triangles as explained in Section 5.4. On bounded domains we do not automatically have a triadic split and special rules have to used for the boundary triangles in the second $\sqrt{3}$ step. These rules use information on the control triangles in the initial configuration before the first $\sqrt{3}$ step. Therefore in practice we always do triadic subdivision and not one step of $\sqrt{3}$ subdivision.

We also suggested to scale the control triangles in the original vertices, although the original control triangles contain the new PS-points, because a smaller control triangle is favorable for the stability. To this end we must choose the new interior points in the refined triangulation in the same way on all the edges that are common to an original vertex.

Uniform spline wavelets

The first attempt to design Powell-Sabin spline wavelets was for uniform Powell-Sabin splines in collaboration with Joris Windmolders [129]. We used the lifting scheme with the uniform dyadic subdivision scheme as the prediction step. The update step added vanishing moments to the unlifted wavelets. We did not include these results in this thesis.

Non uniform wavelets

In Chapter 6 we designed non uniform Powell-Sabin spline wavelets [120]. Because the approximation spaces V^j have to be spanned by Riesz bases, we use the B-spline basis functions that are normalized with respect to their support as scaling functions. The triadic subdivision scheme provides the connection between

the different resolutions. The wavelet functions are found by projecting the scaling functions in the old vertices $\mathcal{N}^j$ onto the desired complement space W^j along V^j

$$\Psi^j = \mathcal{N}^{j+1} - \Phi^j U^j.$$

This corresponds to the update step in the lifting scheme. The triadic subdivision scheme is the prediction step.

Because we are dealing with the non uniform setting, stability is not straightforward and few results have been published on this topic. In Section 6.3 we argue that it is important to have at least one vanishing moment and that the transform should be as semi-orthogonal as possible. We achieve this by choosing the update stepin Section 6.4 as the least squares solution of an overdetermined system with an equality constraint.

Stability of the wavelet transform

In Section 6.5 we gave a proof of the stability of the one level wavelet transform

$$\|M^j\|, \|(M^j)^{-1}\| = \mathcal{O}(1),$$

for the 1-norm and for the ∞-norm. Combination of these norms gives an upper bound for the 2-norm. This is joint work with colleague Jan Maes. We also calculated condition numbers for a practical example to illustrate the results.

Generalized subdivision based on uniform PS-splines

In Chapter 7 we generalized the dyadic subdivision algorithm for uniform Powell-Sabin splines to arbitrary topologies [117] and use it as a surface definition technique. The resulting surface is only a Powell-Sabin spline surface in the regular regions, and the subdivision process is viewed as a linear map in the space of control points without knowing something a priori about the limit surface. In contrast to other subdivision schemes with a spline background, our scheme is interpolating, namely in the barycenter of each control triangle. The control triangle gives the tangent plane in the interpolation point.

In the regular vertices we can not always use the original rules from [125] because the control triangles are not assumed to be oriented all in the same direction. We derive new rules based on the underlying Bézier representation. We facilitate the notation and the analysis by introducing a rotation matrix Q. For the extraordinary vertices with a valence different from six we design new subdivision rules based on a variant of the rotation matrix (Section 7.3).

Convergence analysis of tangent scheme

We described the structure of the subdivision matrix in detail in Section 7.4. It depends in general on the number of rings needed to describe a surface ring. In this specific case we need two rings. It also depends on the labelling of the vertices. This happens in a structured manner, the vertices of each segment are labelled in the same sequence. The central point is replaced with n rotated copies of itself to achieve a periodic structure. This permits to restrict the analysis to one segment of the characteristic map.

The eigenvalues can be calculated by using a discrete Fourier transform for block circulant matrices. The double subdominant eigenvalue equals $1/2$. We calculated the partial derivative of the characteristic map in terms of its Bézier ordinates. They are all positive, which means that the partial derivative is positive and that the characteristic map is regular and injective. According to the theorem of Reif, we conclude that the tangent scheme converges to a C^1 continuous surface.

Implementation in C++

All the algorithms are implemented in C++. We have mainly built on the package PSSURF written by Joris Windmolders and adapted it to our needs. For a detailed description of the underlying data structures we refer to his PhD text [124]. For the implementation of the triadic subdivision and the tangent scheme we have chosen other data structures and integrated them in PSSURF. We also did some major changes on implementation level that improved the scalability.

8.2 Suggestions for further research

Rational Powell-Sabin splines

It is worth investigating whether the described algorithms can be extended to rational Powell-Sabin splines [126]. Rational surface representations give a designer extra degrees of freedom compared to their non rational counterparts through the weights that are associated with each control point.

A rational Powell-Sabin spline can be considered as a parametric surface with four dimensional points as control points. The multiresolution techniques can be applied to the four coordinate functions separately. However, it is not clear what the effect will be of algorithms processing the weight as a separate coordinate function in terms of stability and unwanted effects.

Modeling of special features with the tangent scheme

We observed that the limit surface of the tangent scheme is not only influenced by the direction of the tangents in the original vertices, but also by the size of the control triangles. Big control triangles yield smooth rounded edges, and smaller control triangles yield sharper corners.

It would be interesting to investigate what happens if a control triangle reduces to a line or a point. Will the scheme still converge? How do the newly introduced control triangles behave in the neighborhood of degenerate control triangles? Can we model special effects? On the other hand, when the control triangle grows, the corresponding basis functions become linearly dependent. It is an open question what effect this has on the limit surface and the convergence. Another open question is what happens when an initial control triangle is rotated without changing the ordering of the control points.

For most subdivision schemes special rules have been developed to model cusps. A cusp can be seen as the intersection of two curved surfaces. With the tangent scheme we can place a control triangles perpendicular to the initial control mesh to model a cusp. However, the corners of the control triangles must be ordered anti-clockwise so that the normal points to the exterior of the surface. This is essential for the practical implementation as it is now. We should note that this kind of requirement appears often in multiresolution techniques but then for the triangles of the control polyhedron themselves. It is unclear if we can easily overcome this practical problem.

Initial control triangles for the tangent scheme

The polyhedron that serves as the coarse base mesh for subdivision is typically given as a connected set of three dimensional points. In the case of the tangent scheme we need a control triangle, or in other words we also need the tangents in each point. This kind of input is not always available.

It would be useful to have an algorithm that automatically calculates control triangles for an initial base polyhedron. Such an algorithm could be based on the differences in gradients between the triangles in the molecule of a vertex. The resulting control triangles can be used as a starting point and adjusted interactively to achieve special effects.

Normal meshes with the tangent scheme

The goal of remeshing techniques is to approximate a mesh with irregular topology by a semi-regular mesh. This is useful because many multiresolution techniques require the input to have subdivision connectivity. A particular remeshing technique that can be interesting in the context of this thesis is the technique of the normal meshes [58].

First a mesh simplification procedure is applied to the irregular input mesh to yield a coarse base mesh. Then a subdivision algorithm is applied to the coarse mesh and an approximation of the normal in the new points is calculated. Finally a piercing procedure is applied to find the intersection of the irregular mesh and the normal line. The intersection point will be used instead of the point found by subdivision. The subdivision and piercing are repeated until the approximation criteria are satisfied.

We think it would be useful to build normal meshes with the tangent scheme as the subdivision step. The main advantage lies in the fact that no approximations of normals need to be calculated based on the coarse mesh. The normal is given for free because the control triangles are tangent to the subdivision surface. Furthermore, the approximation order of a normal remesher depends on the approximation order of the underlying subdivision scheme. In the original algorithm only a linear subdivision scheme is used.

The result of the piercing procedure only gives the tangent point and not the directions of the tangents. The procedure should be adapted to yield a new control triangle. The normal mesh technique was designed to produce detail or wavelet coefficients with no tangential components. The wavelet vectors perfectly align with the locally defined normal direction. Then only one scalar coefficient needs to be stored instead of the standard vector. For the tangent scheme we have three wavelet vectors, can we align them in a similar way such that only scalars are needed?

Bibliography

[1] P. Alfeld. Bivariate splines and minimal determining sets. *J. Comput. Appl. Math.*, 119:13–27, 2000.

[2] P. Alfeld, B. Piper, and L.L. Schumaker. An explicit basis for C^1 quartic bivariate splines. *SIAM J. Numer. Anal*, 24:891–911, 1987.

[3] P. Alfeld and L. Schumaker. The dimension of bivariate spline spaces of smoothness r for degree $d \geq 4r + 1$. *Constr. Approx.*, 3:189–197, 1987.

[4] P. Alfeld and L. Schumaker. Smooth macro–elements based on Powell–Sabin triangle splits. *Adv. Comput. Math.*, 16:29–46, 2002.

[5] A. Ball and D. Storry. Conditions for tangent plane continuity over recursively generated b-spline surfaces. *ACM Trans. Graph.*, 7(2):83–102, 1988.

[6] G. Battle. A block spin construction of ondelettes. Part I: Lemarié functions. *Commun. Math. Phys.*, 110:601–615, 1987.

[7] P. Bézier. *Essay de définition numérique des courbes et des surfaces expérimentales*. PhD thesis, University of Paris VI, France, 1977.

[8] W. Boehm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Comp. Aided Geom. Des.*, 1:1–60, 1984.

[9] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput.-Aided Des.*, 10(6):350–355, 1978.

[10] A. S. Cavaretta, W. Dahmen, and C. A. Michelli. Stationary subdivision. *Mem. Am. Math. Soc.*, 93(453), 1991.

[11] A.S. Cavaretta and C. A. Michelli. Computing surfaces invariant under subdivision. *Comput. Aided Geom. Des.*, 4(4):321–328, 1987.

[12] A.S. Cavaretta and C. A. Michelli. The design of curves and surfaces by subdivision algorithms. In T. Lyche and L.L. Schumaker, editors, *Mathematical Aspects of Computer Aided Geometric Design*. Academic Press, Tampa, 1989.

[13] G. Chaikin. An algorithm for high speed curve generation. *Comput. Graph. Im. Proc.*, 3:346–349, 1974.

[14] C.K. Chui, D. Hong, and R.Q. Jia. Stability of optimal order approximation by bivariate splines over arbitrary triangulations. *Trans. Am. Math. Soc.*, 347:3301–3318, 1995.

[15] C.K. Chui and J.Z Wang. On compactly supported spline wavelets and a duality principle. *Trans. Am. Math. Soc.*, 330(2):903–915, 1992.

[16] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Comp. Graph.*, pages 37–54, 1998.

[17] R.W. Clough and J.L Tocher. Finite element stiffness matrices for analysis of plates in bending. In *Proceedings of the first Conference on Matrix Methods in Structural Mechanics*, pages 515–545. Wright Patterson Air Force Base, Ohio, 1965.

[18] A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, 45:485–560, 1992.

[19] A. Cohen, K. Grochenig, and L.F. Villemoes. Regularity of multivariate refinable functions. *Constr. Approx.*, 15:241–255, 1999.

[20] E. Cohen, T. Lyche, and R. Riesenfield. Discrete B-splines and subdivision techniques in computer aided geometric design and computer graphics. *Comput. Graph. Im. Proc.*, 14(2):87–111, October 1980.

[21] M. Cotronei, L. Montefusco, and L. Puccio. Multiwavelet analysis and signal processing. *IEEE Trans. Circ. Syst. II*, 1997.

[22] W. Dahmen. Some remarks on multiscale transformations, stability and orthogonality. In P.J. Laurent, A. L. Méhauté, and L.L. Schumaker, editors, *Curves and Surfaces II*, pages 1–32. A.K. Peters, Boston, 1991.

[23] W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal spline-wavelets on the interval: Stability and moment conditions. *Appl. Comput. Harmon. Anal.*, 6:132–196, 1999.

[24] W.A. Dahmen, R.H.J. Gmelig Meyling, and J.H.M. Ursem. Scattered data interpolation by bivariate C^1 piecewise quadratic functions. *Approx. Theory Appl.*, 6:6–29, 1990.

[25] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[26] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens. Wavelets on irregular point sets. *Philos. Trans. R. Soc. Lond. Ser. A-Math. Phys. Eng. Sci.*, 357(1760):2397–2413, 1999.

[27] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.

[28] G. M. Davis, V. Strela, and R. Turcajová. Multiwavelets construction via the lifting scheme. In *Wavelet Analysis and Multiresolution Methods*. Marcel Dekker, 1999.

[29] O. Davydov. Locally linearly independent basis for C^1 bivariate splines of degree $q \geq 5$. In M. Daehlen, T. Lyche, and L.L Schumaker, editors, *Mathematical Methods for Curves and Surfaces II*, pages 71–78. Vanderbilt University Press, Nashville, 1998.

[30] O. Davydov and L. Schumaker. Locally linearly independent bases for bivariate polynomial spline spaces. *Adv. Comput. Math.*, 13:355–373, 2000.

[31] O. Davydov and L.L. Schumaker. On stable local bases for bivariate polynomial splines. *Constr. Approx.*, 18:87–116, 2002.

[32] C. de Boor. Bicubic spline interpolation. *J. Math. Phys.*, 41:212–218, 1962.

[33] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972.

[34] C. de Boor. *A practical guide to splines*. Number 27 in Applied Mathematical Sciences. Springer, New York, 1978.

[35] C. de Boor. Cutting corners always works. *Comp. Aided Geom. Design*, 4:125–131, 1987.

[36] P. de Casteljau. Outillages, méthodes de calcul. Technical report, André Citroën Automobiles S.A., Paris, 1959.

[37] P. de Casteljau. Courbes et surfaces à poles. Technical report, André Citroën Automobiles S.A., Paris, 1963.

[38] G. de Rham. Un peu de mathématique à propos d'une courbe plane. *Elem. Math,*, 2:73–76;89–97, 1947.

[39] G. de Rham. Sur une courbe plane. *J. Math. Pures Appl.*, 35:25–42, 1956.

[40] G. Deslauriers and S. Dubuc. Interpolation dyadique. In *Fractals, dimensions non entières et applications*, pages 44–55. Masson, Paris, 1987.

[41] P. Dierckx. On calculating normalized Powell-Sabin B-splines. *Comput. Aided Geom. Des.*, 15(1):61–78, 1997.

[42] P. Dierckx, S. Van Leemput, and T. Vermeire. Algorithms for surface fitting using Powell–Sabin splines. *IMA J. Numer. Anal.*, 12(2):271–299, 1992.

[43] N.A. Dodgson, M.A. Sabin, L. Barthe, and M.F. Hassan. Towards a ternary interpolating subdivision scheme for the triangular mesh. Technical Report 529, University of Cambridge Computer Laboratory, 2002.

[44] D. Doo and M. Sabin. Analysis of the behaviour of recursive division surfaces near extraordinary points. *Comput.-Aided Des.*, 10(6):356–360, 1978.

[45] S. Dubuc. Interpolation through an iterative scheme. *J. Math. Anal. Appl.*, 114:185–204, 1986.

[46] N. Dyn, D. Levin, and J. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Comp. Aided Geom. Des.*, 4:257–268, 1987.

[47] N. Dyn, D. levin, and J. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph.*, 9(2):160–169, April 1990.

[48] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Computer Graphics*, volume 29 of *Annual Conference Series*, pages 173–182, 1995.

[49] G. Farin. Triangular Bernstein-Bézier patches. *Comput. Aided Geom. Des.*, 3(2):83–127, 1986.

[50] J. Ferguson. Multivariable curve interpolation. *J. ACM*, 11(2):221–228, April 1964.

[51] M. Floater, E. Quak, and M. Reimers. Filter bank algorithms for piecewise linear prewavelets on arbitrary triangulations. *J. Comput. Appl. Math.*, 119:185–207, 2000.

[52] A.R. Forrest. Interactive interpolation and approximation by Bézier polynomials. *Comput. J.*, 15:71–79, 1972.

[53] D. Forsey and R. Bartels. Hierarchical B-spline refinement. In *Computer Graphics (SIGGRAPH 88 Proceedings)*, pages 205–212, 1988.

[54] D. Forsey and R. Bartels. Surface fitting with hierarchical splines. *ACM Trans. Graph.*, pages 134–161, 1995.

[55] A. Grossmann and J. Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15:723–736, 1984.

[56] A. Grossmann, J. Morlet, and T. Paul. Transforms associated to square integrable group representations I: General results. *J. Math. Phys.*, 26:2473–2479, 1985.

[57] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Computer Graphics (SIGGRAPH 99 Proceedings)*, pages 325–334, 1999.

[58] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal meshes. In *Computer Graphics Proceedings (SIGGRAPH 2000)*, pages 95–102, 2000.

[59] A. Haar. Zur Theorie der orthogonalen Funktionen-Systeme. *Math. Ann.*, 69:331–371, 1910.

[60] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using catmull-calrk surfaces. In *Computer Graphics Proceedings*, Annual Conference Series, pages 35–44. ACM Siggraph, 1993.

[61] G. Heindl. Interpolation and approximation by piecewise quadratic C^1 functions of two variables. In W. Schempp and K. Zeller, editors, *Multivariate Approximation Theory ISNM 51*, pages 146–161. Birkhäuser Verlag, Basel, 1979.

[62] D. Hong. Spaces of bivariate spline functions over triangulation. *Approx. Theory Appl.*, 7:56–75, 1991.

[63] H. Hoppe. Progressive meshes. In *Computer Graphics (SIGGRAPH 96 Proceedings)*, pages 99–108, 1996.

[64] R. Q. Jia. Approximation order from certain spaces of smooth bivariate splines on a three-direction mesh. *Trans. Am. Math. Soc.*, 295:199–212, 1986.

[65] R.A. Johnson. *Advanced Euclidean Geometry: An elementary treatise on the geometry of the triangle and the circle*. Dover New York, 1960.

[66] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Proceedings of Eurographics 96*, Computer Graphics Forum, pages 409–420, 1996.

[67] L. Kobbelt, S. Campagna, and J. Vorsatz. Interactive multi-resolution modeling on arbitrary meshes. In *Computer Graphics (SIGGRAPH 98 Proceedings)*, pages 105–114, 1998.

[68] L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. *Comput. Graph. Forum*, 18:C119–C130, 1999.

[69] Leif Kobbelt. $\sqrt{3}$–subdivision. In *Computer Graphics Proceedings*, Annual Conference Series. ACM SIGGRAPH, July 2000.

[70] J. Kovačević and W. Sweldens. Wavelet families of increasing order in arbitrary dimensions. *IEEE Trans. Image Proc.*, 9(3):480–496, March 2000.

[71] U. Labsik and G. Greiner. Interpolatory $\sqrt{3}$-subdivision. In Sabine Coquillart and Jr. Duke David, editors, *Proceedings of the 21th European Conference on Computer Graphics (Eurographics-00)*, volume 19(3) of *Computer Graphics Forum*, pages 131–138, Cambridge, August 21–25 2000. Blackwell Publishers.

[72] M. Lai and L. Schumaker. On the approximation power of bivariate splines. *Adv. Comput. Math.*, 9:251–279, 1998.

[73] M. Lai and L. Schumaker. Macro-elements and stable local bases for splines on Powell-Sabin triangulations. *Math. Comp.*, 72:335–354, 2003.

[74] J.M. Lane and R.F. Riesenfeld. A theoretical development for the computer generation of piecewise polynomial surfaces. *IEEE Trans. Patt. Anal. Mach. Intell.*, 1:35–46, 1980.

[75] A. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. Maps: Multiresolution adaptive parametrization of surfaces. In *Computer Graphics (SIGGRAPH 1998)*, pages 95–104, 1998.

[76] P.-G. Lemarié. Ondelettes à localisation exponentielles. *J. Math. Pures Appl.*, 67(3):227–236, 1988.

[77] P.G. Lemarié and Y. Meyer. Ondelettes et bases hilbertiennes. *Rev. Mat. Iberoam.*, 2:1–18, 1986.

[78] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[79] C. Loop. Smooth ternary subdivision of triangle meshes. In A. Cohen, J.-L Merrien, and L.L Schumaker, editors, *Curve and Surface Fitting, Saint Malo 2002*, pages 295–302. Nashboro Press, Brentwood, 2003.

[80] Michael Lounsbery, Tony D. DeRose, and Joe Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, 1997.

[81] T. Lyche and K. Morken. Making the oslo algorithm more efficient. *SIAM J. Numer. Anal.*, 23(3):663–675, June 1986.

[82] T. Lyche and K. Morken. Spline wavelets of minimal support. In D. Braess and L.L. Schumaker, editors, *Numerical methods in Approximation Theory*, pages 177–194. Birkhäuser, Basel, 1992.

[83] D. MacLaren. Formulas for fitting a spline curve through a set of points. Appl. Math. Report 2, Boeing, 1958.

[84] J. Maes, E. Vanraes, P. Dierckx, and A. Bultheel. On the stability of normalized Powell-Sabin B-splines. *J. Comput. Appl. Math.*, Accepted, 2004.

[85] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[86] S.G. Mallat. Multifrequency channel decomposition of images and wavelet models. *IEEE Trans. Acoust., Speech, Signal Proc.*, 37(12):2091–2110, December 1989.

[87] S.G. Mallat. Multiresolution approximations and wavelet orthonormal bases of l (ir). *Trans. Am. Math. Soc.*, 315:69–87, 1989.

[88] J. Morgan and R. Scott. A nodal basis for C^1 piecewise polynomials of degree $n \geq 5$. *Math. Comput.*, 29:736–740, 1975.

[89] A.H. Nasri. Surface interpolation on irregular networks with normal conditions. *Comput. Aided Geom. Des.*, 8:89–96, 1991.

[90] J. Peters and U. Reif. Analysis of algorithms generalizing B-spline subdivision. *SIAM J. Numer. Anal.*, 35(2):728–748, April 1998.

[91] G. Plonka and V. Strela. From wavelets to multiwavelets. In *Mathematical Methods for Curves and Surfaces II*. Vanderbilt University Press, Nashville, TN, 1998.

[92] M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximations on triangles. *ACM Trans. Math. Softw.*, 3:316–325, 1977.

[93] R. Qu and J. Gregory. A subdivision algorithm for non-uniform B-splines. In *Approximation Theory, Spline functions and Applications*, NATO ASI Series C: Mathematical and Physical Sciences 356, pages 423–436. 1992.

[94] E. Quak and N. Weyrich. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval. *Appl. Comput. Harmon. Anal.*, 1:217–231, 1994.

[95] U. Reif. Some new results on subdivision algorithms for meshes of arbitrary topology. In C.K. Chui and L.L Schumaker, editors, *Wavelets and MultiLevel Approximation*, Series in Approximations and Decompositions 2, pages 367–374. World Scientific, Singapore, 1995.

[96] U. Reif. A unified approach to subdivision algorithms near extraordinary points. *Comput. Aided Geom. Des.*, 12:153–174, 1995.

[97] U. Reif. A degree estimate for polynomial subdivision surface of higher regularity. In *Proceedings of the American mathematical society*, volume 124(7), July 1996.

[98] R. Riesenfeld. On Chiakin's algorithm. *Comput. Graph. Im. Proc.*, 4(3):304–310, 1975.

[99] D.J. Ripmeester. Upper bounds on the dimension of bivariate spline spaces and duality in the plane. In M. Daehlen, T. Lyche, and L.L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, pages 455–466. Vanderbilt University Press, Nashville, 1995.

[100] A. Ron and Z. Shen. The Sobolev regularity of refinable functions. *J. Approx. Theory*, 106(2):185–225, 1997.

[101] M. A. Sabin. *The use of piecewise forms for the numerical representation of shape*. PhD thesis, Hungarian Academy of Sciences, Budapest, Hungary, 1976.

[102] P. Sablonnière. Error bounds for hermite interpolation by quadratic splines on an α-triangulation. *IMA J. Numer. Anal.*, 7:495–508, 1987.

[103] I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Q. Appl. Math.*, 4:45–99,112–141, 1946.

[104] L.L. Schumaker. Lower bounds for the dimension of spaces of piecewise polynomials in two variables. In W. Schempp and K. Zeller, editors, *Multivariate Approximation Theory*, pages 396–412. Brikhäuser Verlag, Basel, 1979.

[105] L.L. Schumaker. Bounds on the dimension of spaces of multivariate piecewise polynomials. *Rocky Mt. J. Math.*, 14:251–264, 1984.

[106] T.W. Sederberg, J. Zheng, D. Sewell, and M. Sabin. Non-uniform recursive subdivision surfaces. *Computer Graphics*, 32:387–394, 1998.

[107] X. Shi, S. Wang, W. Wang, and R. H. Wang. The C^1 quadratic spline space on triangulations. *Report 86004, Department of Mathematics, Jilin University, Changchun*, 1986.

[108] J. Simoens and S. Vandewalle. A stabilized lifting construction of wavelets on irregular meshes on the interval. *SIAM J. Sci. Comput.*, 24(4):1356–1378, 2003.

[109] G. Strang. Piecewise polynomials and the finite element method. *Bull. Amer. Math. Soc.*, 79:1128–1137, 1973.

[110] J.O. Strömberg. A modified franklin system and higher order spline system of $\mathbb{R}^n$ as unconditional bases for Hardy spaces. In *Proc. Conference in Harmonic Analysis in honor of Antoni Zygmund*, pages 475–493, 1983.

[111] W. Sweldens. The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

[112] W. Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[113] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, ACM SIGGRAPH Course Notes, pages 15–87, 1996.

[114] G. Umlauf. Analyzing the characteristic map of triangular subdivision schemes. *Constr. Approx.*, 16(1):145–155, 2000.

[115] M. Unser, A. Aldroubi, and M. Eden. On the asymptotic convergence of B-spline wavelets to gabor functions. *IEEE Trans. Inf. Theory*, 38(2):864–872, March 1992.

[116] A. Z. Averbuch V. A. Zheludev and M. Gruzd. Interpolatory subdivision schemes generated by splines. In A. Cohen, J.-L. Merrien, and L.L. Schumaker, editors, *Curve and Surface Fitting: Saint-Malo 2002*, pages 393–402. Nashboro Press, 2003.

[117] E. Vanraes and A. Bultheel. Tangent subdivision. *ACM Trans. Graph.*, Submitted, 2004.

[118] E. Vanraes, P. Dierckx, and A. Bultheel. On the choice of the PS-triangles. TW Report 353, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 2003.

[119] E. Vanraes and M. Jansen. Stabilized wavelet transforms for non-equispaced data smoothing. *Signal Process.*, 82(12):1979–1990, 2002.

[120] E. Vanraes, J. Maes, and A. Bultheel. Powell-Sabin spline wavelets. *Int. J. Wav. Multires. Inf. Proc.*, 2(1):23–42, 2004.

[121] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Dyadic and $\sqrt{3}$–subdivision for Uniform Powell–Sabin splines. In *Information Visualisation Proceedings*. IEEE Computer Society, 2002.

[122] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Subdivision for powell-Sabin spline surfaces. *Comput. Aided Geom. Design*, Submitted, 2002.

[123] K. Willemans. *Smoothing scattered data with a constrained Powell-Sabin spline surface*. PhD thesis, Katholieke Universiteit Leuven, 1996.

[124] J. Windmolders. *Powell-Sabin splines for computer aided geometric design*. PhD thesis, K.U.Leuven, 2003.

[125] J. Windmolders and P. Dierckx. Subdivision of Uniform Powell-Sabin splines. *Comput. Aided Geom. Des.*, 16:301–315, May 1999.

[126] J. Windmolders and P. Dierckx. From PS-splines to NURPS. In A. Cohen, C. Rabut, and L.L. Schumaker, editors, *Proceedings of Curve and Surface Fitting, Saint Malo 1999*. Vanderbilt University Press, Nashville, 2000.

[127] J. Windmolders and P. Dierckx. Nurps for quadrics and special effects. In T. Lyche and L.L. Schumaker, editors, *Proceedings of Mathematical Methods for Curves and Surfaces, Oslo, 2000*. Vanderbilt University Press, Nashville, 2001.

[128] J. Windmolders and P. Dierckx. Uniform Powell-Sabin splines for the polygonal hole problem. In *Proceedings of algorithms for Approximation IV*, 2001.

[129] J. Windmolders, E. Vanraes, P. Dierckx, and A. Bultheel. Uniform Powell-Sabin spline wavelets. *J. Comput. Appl. Math.*, 154(1):125–142, 2003.

Curriculum vitae

Studies en onderzoeksloopbaan

- 1995 - 2000
  Burgerlijk ingenieur Computerwetenschappen, optie Toegepaste Wiskunde
  Geslaagd met grote onderscheiding
  KULeuven

- 01/09/2000 - 30/09/2000
  Overbruggingsmandaat
  Numerieke Approximatie en Lineaire Algebra Groep
  KULeuven

- 01/10/2000 - 30/09/2004
  Aspirant Fwo
  Numerieke Approximatie en Lineaire Algebra Groep
  KUleuven

Buitenlandse verblijven

- 25/05/2001 - 25/06/2001
  Digital Signal Processing Group
  Rice University

Publicaties

1. E. Vanraes, M. Jansen and A. Bultheel, Stabilized lifting steps in noise reduction for nonequispaced samples, Wavelets: Applications in Signal and Image Processing IX (A.F. Laine, M.A. Unser and A. Aldroubi, eds.), vol 4478, Proceedings of SPIE, pp. 105–116, 2001.

2. E. Vanraes, M. Jansen and A. Bultheel, Stabilised wavelet transforms for non-equispaced data smoothing, Signal Processing 82(12):1979–1990, 2002.

3. E. Vanraes, J. Windmolders, A. Bultheel and P. Dierckx, Dyadic and $\sqrt{3}$ subdivision for uniform Powell Sabin splines, Proceedings of Sixth International Conference on Information Visualisation (D. Williams, ed.), pp. 639–643, 2002.

4. J. Windmolders, E. Vanraes, P. Dierckx and A. Bultheel, Uniform Powell-Sabin Spline Wavelets, J. Comput. Appl. Math. 154(1):125–142, 2003.

5. E. Vanraes, P. Dierckx and A. Bultheel. Triadic subdivision for non-uniform Powell-Sabin splines. Advances in Geometric Modeling (M. Sarfraz, ed.), John Wiley, pp 107–119, 2003.

6. E. Vanraes, J. Maes and A. Bultheel, Powell-Sabin spline wavelets, Int. J. Wav. Multires. Inf. Proc. 2(1):23–42, 2004.

7. J. Maes, E. Vanraes, P. Dierckx and A. Bultheel, On the stability of normalized Powell-Sabin B-splines, J. Comput. Appl. Math., 2004. Ter perse.

8. E. Vanraes, J. Windmolders, A. Bultheel and P. Dierckx, Subdivision for Powell-Sabin spline surfaces, Comput. Aided Geom. Design, Aangeboden ter publicatie.

9. E. Vanraes and A. Bultheel, The tangent scheme, ACM Trans. Graph., 2004. Aangeboden ter publicatie.