FROM COMPUTER AIDED DESIGN TO WAVELET BEM

H. HARBRECHT AND M. RANDRIANARIVONY

Abstract. It is known that the wavelet Galerkin scheme is an efficient method for the numerical solution of boundary integral equations. The parametric description of the given surface is required in order to create wavelets on manifolds. Technical surfaces generated by CAD tools are represented in this form. But often trimmed surfaces appear when thinking of the standard IGES format. An algorithm to parametrize trimmed surfaces in order to apply the wavelet Galerkin scheme is presented. Numerical results are reported to illustrate the approach. In particular, the decomposition techniques are applied to real CAD data which come from IGES files.

Introduction

Various problems in science and engineering can be formulated as boundary integral equations which are generally solved by the boundary element method (BEM). For instance, BEM is a favourable approach for the treatment of exterior boundary value problems, in particular for problems in electromagnetics or in case of the Helmholtz equation. Nevertheless, traditional discretizations will lead in general to very large linear systems with densely populated and possibly ill-conditioned matrices. This makes the computation very costly in both respects, the computation time and computer memory requirements.

In recent years several ideas for the efficient approximation of the discrete system have been developed. Most prominent examples of such methods are fast multipole [17], panel clustering [19], wavelet Galerkin methods [2, 5], and the hierarchical matrix approach [18]. Such fast discretization methods end up with linear or almost linear complexity with respect to the number of unknowns.

The present paper is concerned with the wavelet Galerkin scheme. In fact, a Galerkin discretization with wavelet bases results in quasi-sparse matrices, i.e., most matrix entries are negligible and can be treated as zero. Discarding these nonrelevant matrix entries is called matrix compression. In [23] a fully discrete wavelet Galerkin method has been proposed that produces approximate solutions within discretization accuracy in linear complexity.

Key words and phrases. Computer Aided Design, trimmed surfaces, wavelets, boundary element method.

A strong effort has been spent on the construction of appropriate wavelet bases on surfaces, see e.g. [8, 9, 22, 24, 26]. In order to achieve linear complexity in the wavelet Galerkin scheme, wavelet bases are required which provide sufficiently many vanishing moments. Our realization is based on biorthogonal spline wavelets derived from the multiresolution developed in [4]. The surface is subdivided into smooth four-sided patches which are images of the unit square under smooth diffeomorphisms. Then, the wavelets are constructed by lifting wavelets from the unit square to the surface via parametrization. Gluing along the patch boundaries yields globally continuous wavelet bases.

The surface representation is in contrast to the usual approximation of the surface by panels. It has the advantage that the rate of convergence is not limited by this approximation. Technical surfaces generated by CAD tools are often represented in this form. The most common format of geometry representation is the IGES standard. Since it is very difficult to implement all IGES entities, we have restricted ourselves to IGES 144. Then, the initial CAD object is a solid bounded by a closed surface that is given as a collection of parametric surfaces. Each surface can be trimmed or untrimmed [3]. Untrimmed surfaces are four-sided patches, parametrized over a rectangle. There are several representations [25] including B-splines, NURBS, surfaces of revolution and tabulated cylinders.

Since the treatment of untrimmed surfaces is straightforward, we concentrate on trimmed surfaces. Here, we have a supporting surface parametrized over a reference rectangle. However, since only a piece of the supporting surface enters the given geometry, the remaining part has to be trimmed off. This piece is the image of a multiply connected region delineated by composite curves on the reference rectangle. We will construct diffeomorphisms between the trimmed surface and the unit square which enable us to apply our wavelet Galerkin scheme.

The paper is organized as follows. In Sec. 1 we describe the prerequisites for constructing wavelets on manifolds. Sec. 2 briefly surveys on the features of the wavelet Galerkin scheme for boundary integral equations. The following sections are dedicated to the preparation of the CAD objects for the wavelet Galerkin scheme. Sec. 3 is concerned with the preliminaries on the description of surfaces by the IGES 144 standard which includes in particular trimmed surfaces. The decomposition of trimmed surfaces into four-sided patches is explained in Sec. 4. In Sec. 5 diffeomorphisms between the unit square and the patches are constructed by means of transfinite interpolation. In Sec. 6 we realize global continuity by using a chord length reparametrization. Finally in Sec. 7 we present numerical results to demonstrate the capability of the algorithms.

1. Requirements for the wavelet construction

We shall consider a simply connected domain $\Omega \subset \mathbb{R}^3$ with piecewise smooth boundary $\Gamma = \partial\Omega$. We assume that the boundary manifold Γ is given as a parametric surface consisting of smooth patches. More precisely, let $\square := [0,1]^2$ denote the unit square. The manifold $\Gamma \subset \mathbb{R}^3$ is partitioned into a finite number of *patches*

$$\Gamma = \bigcup_{i=1}^{M} \Gamma_i, \qquad \Gamma_i = \boldsymbol{\gamma}_i(\square), \qquad i = 1, 2, \ldots, M, \tag{1.1}$$

where each $\boldsymbol{\gamma}_i : \square \to \Gamma_i$ defines a diffeomorphism of $\square$ onto Γ_i. The intersection $\Gamma_i \cap \Gamma_{i'}$, $i \neq i'$, of the patches Γ_i and $\Gamma_{i'}$ is supposed to be either $\emptyset$, or a common edge or vertex.

A mesh of level j on Γ is induced by dyadic subdivisions of depth j of the unit square into 4^j cubes $C_{j,\mathbf{k}} \subseteq \square$, where $\mathbf{k} = (k_1, k_2)$ with $0 \leq k_1, k_2 < 2^j$. This generates $4^j M$ *elements* (or elementary domains) $\Gamma_{i,j,\mathbf{k}} := \boldsymbol{\gamma}_i(C_{j,\mathbf{k}}) \subseteq \Gamma_i$, $i = 1, \ldots, M$.

In order to ensure that the collection of elements $\{\Gamma_{i,j,\mathbf{k}}\}$ on the level j forms a regular mesh on Γ, the parametric representation is subjected to the following *matching condition*: A bijective, affine mapping $\boldsymbol{\Xi} : \square \to \square$ exists such that for all $\mathbf{x} = \boldsymbol{\gamma}_i(\mathbf{s})$ on a common edge of Γ_i and $\Gamma_{i'}$ it holds that

$$\boldsymbol{\gamma}_i(\mathbf{s}) = (\boldsymbol{\gamma}_{i'} \circ \boldsymbol{\Xi})(\mathbf{s}). \tag{1.2}$$

In other words, the diffeomorphisms $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_{i'}$ coincide at common edges except for orientation.

We emphasize that the above definitions require patchwise smoothness but *not* global smoothness of the geometry. The surface itself needs to be only Lipschitz.

The nested trial spaces

$$V_{j_0} \subset V_{j_0+1} \subset \cdots \subset V_j \subset V_{j+1} \subset \cdots \subset H^q(\Gamma)$$

that we shall employ in the Galerkin scheme are the spaces of piecewise constant or bilinear functions on the given partition. These trial spaces have the *approximation order* $d = 1$ and $d = 2$ in the case of the piecewise constants and bilinears, respectively, i.e.,

$$\inf_{v_j \in V_j} \|v - v_j\|_0 \lesssim 2^{-jd} \|v\|_d.$$

The trial spaces are spanned by so called *single-scale bases* $\Phi_j = \{\phi_{j,\mathbf{k}} : \mathbf{k} \in \Delta_j\}$ which can be specified as follows: On the level j, we find for each element $\Gamma_{i',j,\mathbf{k}'}$ a piecewise constant scaling function $\phi_{j,\mathbf{k}}$ with

$$\phi_{j,\mathbf{k}}\big|_{\Gamma_{i',j,\mathbf{k}'}} \equiv 2^j$$

and $\phi_{j,\mathbf{k}}(\mathbf{x}) = 0$ elsewhere. To describe the canonical piecewise bilinear scaling functions, we define four bilinear shape functions on the unit square

$$p_1(\mathbf{s}) := (1-s_1)(1-s_2), \qquad p_2(\mathbf{s}) := s_1(1-s_2),$$
$$p_3(\mathbf{s}) := s_1 s_2, \qquad p_4(\mathbf{s}) := (1-s_1)s_2.$$

Then, $\phi_{j,\mathbf{k}}$ is equal to 1 in one node and equal to zero in the remaining nodes and, if its support contains the element $\Gamma_{i',j,\mathbf{k}'}$, we find an $m \in \{1,2,3,4\}$ such that

$$\phi_{j,\mathbf{k}}\big|_{\Gamma_{i',j,\mathbf{k}'}}(\mathbf{x}) = p_m(\mathbf{s}), \qquad \mathbf{x} = \boldsymbol{\gamma}_{i'}\big(2^{-j}(k_1'+s_1), 2^{-j}(k_2'+s_2)\big).$$

Continuity is supposed on each patch, but along the interfaces of the patches we may consider double nodes or continuity. Notice that our definition yields the L^2-normalization $\|\phi_{j,\mathbf{k}}\|_0 \sim 1$.

Wavelets for the present ansatz spaces have been constructed in several papers, see e.g. [8, 9, 24]. The wavelets $\Psi_j = \{\psi_{j,\mathbf{k}} : \mathbf{k} = \nabla_j\}$ ($\nabla_j := \Delta_{j+1} \setminus \Delta_j$) which have been implemented in our code are specified in [22]. Especially they provide *vanishing moments* in terms of

$$|\langle v, \psi_{j,\mathbf{k}}\rangle| \lesssim 2^{-j(\widetilde{d}+n/2)} |v|_{W^{\widetilde{d},\infty}(\mathrm{supp}\,\psi_{j,\mathbf{k}})}. \tag{1.3}$$

Here $|v|_{W^{\widetilde{d},\infty}(\Omega)} := \sup_{|\alpha|=\widetilde{d},\, x\in\Omega} |\partial^\alpha v(x)|$ denotes the semi-norm in $W^{\widetilde{d},\infty}(\Omega)$. Since we use biorthogonal wavelets the order $\widetilde{d}$ of vanishing moments can be chosen higher than the approximation order d. This is essential for deriving optimal compression rates and could not be realized by orthonormal bases.

2. Wavelet Galerkin BEM

We briefly overview on the principles of wavelet Galerkin BEM. We consider the boundary integral equation

$$A u(\mathbf{x}) = \int_\Gamma k(\mathbf{x},\mathbf{y}) u(\mathbf{y}) d\Gamma_{\mathbf{y}} = f(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{2.4}$$

where the boundary integral operator is assumed to be an operator of order $2q$, that is $A : H^q(\Gamma) \to H^{-q}(\Gamma)$. The kernel functions under consideration are supposed to be smooth as functions in the variables $\mathbf{x}, \mathbf{y}$, apart from the diagonal $\{(\mathbf{x},\mathbf{y}) \in \Gamma \times \Gamma : \mathbf{x} = \mathbf{y}\}$ and may have a singularity on the diagonal. Such kernel functions arise, for instance, by applying a boundary integral formulation to a second order elliptic problem. In general, they decay like a negative power of the distance of the arguments which depends on the order $2q$ of the operator.

We shall be concerned with the Galerkin method for the solution of the given boundary integral equation (2.4): find $u_J \in V_J$ solving the variational problem

$$\langle A u_J, v_J\rangle = \langle f, v_J\rangle \quad \text{for all} \quad v_J \in V_J.$$

Traditionally this equation is discretized by the single-scale basis of V_J which yields a densely populated system matrix. Using instead wavelets with a sufficiently strong cancellation property (1.3), the system matrix becomes quasi-sparse and the most matrix coefficients are negligible without compromising the order of convergence of the Galerkin scheme [5, 30].

But before we formulate this result, we introduce the following abbreviation

$$\Omega_{j,\mathbf{k}} := \mathrm{convhull}(\mathrm{supp}\psi_{j,\mathbf{k}}), \qquad \Omega'_{j,\mathbf{k}} := \mathrm{sing\,supp}\psi_{j,\mathbf{k}}. \tag{2.5}$$

Notice that the first expression denotes the convex hull of the support of a wavelet with respect to the Euclidean space $\mathbb{R}^3$. The second expression indicates the *singular support*, i.e. that subset of Γ where the wavelet is not smooth.

Theorem 2.1 (A-priori compression)**.** *Let $\Omega_{j,\mathbf{k}}$ and $\Omega'_{j,\mathbf{k}}$ be given as in* (2.5) *and define the compressed system matrix $\mathbf{A}_J$, corresponding to the boundary integral operator A, by*

$$[\mathbf{A}_J]_{(j,\mathbf{k}),(j',\mathbf{k}')} := \begin{cases} 0, & \mathrm{dist}(\Omega_{j,\mathbf{k}},\Omega_{j',\mathbf{k}'}) > \mathcal{B}_{j,j'} \text{ and } j,j' \geq j_0, \\ 0, & \mathrm{dist}(\Omega_{j,\mathbf{k}},\Omega_{j',\mathbf{k}'}) \lesssim 2^{-\min\{j,j'\}} \text{ and} \\ & \mathrm{dist}(\Omega'_{j,\mathbf{k}},\Omega_{j',\mathbf{k}'}) > \mathcal{B}'_{j,j'} \text{ if } j' > j \geq j_0 - 1, \\ & \mathrm{dist}(\Omega_{j,\mathbf{k}},\Omega'_{j',\mathbf{k}'}) > \mathcal{B}'_{j,j'} \text{ if } j > j' \geq j_0 - 1, \\ \langle A\psi_{j',\mathbf{k}'},\psi_{j,\mathbf{k}}\rangle, & \text{otherwise.} \end{cases} \tag{2.6}$$

Fixing

$$a > 1, \qquad d < \delta < \widetilde{d} + 2q, \tag{2.7}$$

the cut-off parameters $\mathcal{B}_{j,j'}$ and $\mathcal{B}'_{j,j'}$ are set as follows

$$\begin{aligned} \mathcal{B}_{j,j'} &= a\ \max\left\{2^{-\min\{j,j'\}}, 2^{\frac{2J(\delta-q)-(j+j')(\delta+\widetilde{d})}{2(\widetilde{d}+q)}}\right\}, \\ \mathcal{B}'_{j,j'} &= a \max\left\{2^{-\max\{j,j'\}}, 2^{\frac{2J(\delta-q)-(j+j')\delta-\max\{j,j'\}\widetilde{d}}{\widetilde{d}+2q}}\right\}. \end{aligned} \tag{2.8}$$

Then, the system matrix $\mathbf{A}_J$ has only $\mathcal{O}(N_J)$ nonzero coefficients. Moreover, the error estimate

$$\|u - u_J\|_{2q-d} \lesssim 2^{-2J(d-q)}\|u\|_d \tag{2.9}$$

holds for the solution u_J of the compressed Galerkin system provided that u and Γ are sufficiently regular.

The compressed system matrix can be assembled in linear complexity if one employs the exponentially convergent *hp*–quadrature method proposed in [23]. If the entries

of the compressed system matrix $\mathbf{A}_J$ have been computed, we may apply an a-posteriori compression by setting all entries to zero, which are smaller than a level dependent threshold. That way, a matrix $\widehat{\mathbf{A}}_J$ is obtained which has less nonzero entries than the matrix $\mathbf{A}_J$. Clearly, this does not accelerate the calculation of the matrix coefficients. But if the linear system of equations has to be solved for several right hand sides, like for instance in shape optimization (see e.g. [20]) or in inverse obstacle problems (see e.g. [21]), the faster matrix-vector multiplication pays off. To our experience the a-posteriori compression reduces the number of nonzero coefficients by a factor 2–5.

Theorem 2.2 (A-posteriori compression)**.** *We define the a-posteriori compression by*

$$\big[\widehat{\mathbf{A}}_J\big]_{(j,\mathbf{k}),(j',\mathbf{k}')} = \begin{cases} 0, & \text{if } \big|\big[\mathbf{A}_J\big]_{(j,\mathbf{k}),(j',\mathbf{k}')}\big| \leq \varepsilon_{j,j'}, \\ \big[\mathbf{A}_J\big]_{(j,\mathbf{k}),(j',\mathbf{k}')}, & \text{if } \big|\big[\mathbf{A}_J\big]_{(j,\mathbf{k}),(j',\mathbf{k}')}\big| > \varepsilon_{j,j'}. \end{cases}$$

Herein, the level dependent threshold $\varepsilon_{j,j'}$ is given by

$$\varepsilon_{j,j'} \sim \min\Big\{2^{-|j-j'|n/2}, 2^{-2n(J-\frac{j+j'}{2})\frac{\delta-q}{d+q}}\Big\}2^{2Jq}2^{-2\delta(J-\frac{j+j'}{2})}$$

with $d < \delta < \widetilde{d} + r$ from (2.7)*. Then, the optimal order of convergence* (2.9) *of the Galerkin scheme is not compromised.*

If the order q of the boundary integral operator A is $\neq 0$, the compressed system matrix $\mathbf{A}_J$ becomes more and more ill-conditioned when J increases. However, as a consequence of the norm equivalences of wavelet bases, the diagonally scaled system matrix has uniformly bounded spectral condition numbers ([6, 30]).

Theorem 2.3 (Preconditioning)**.** *Let the diagonal matrix $\mathbf{D}_J^r$ defined by*

$$\big[\mathbf{D}_J^r\big]_{(j,\mathbf{k}),(j',\mathbf{k}')} = 2^{rj}\delta_{j,j'}\delta_{\mathbf{k},\mathbf{k}'}, \quad \mathbf{k} \in \nabla_j, \quad \mathbf{k}' \in \nabla_{j'}, \quad j_0 \leq j, j' < J.$$

Then, if the regularity $\widetilde{\gamma}$ of the dual wavelets satisfies $\widetilde{\gamma} > -q$, the diagonal matrix $\mathbf{D}_J^{2q}$ defines an asymptotically optimal preconditioner to $\mathbf{A}_J$, i.e.,

$$\operatorname{cond}_{\ell^2}(\mathbf{D}_J^{-q}\mathbf{A}_J\mathbf{D}_J^{-q}) \sim 1.$$

Remark 2.4. *The entries on the main diagonal of $\mathbf{A}_J$ satisfy $\langle \mathcal{A}\psi_{j,\mathbf{k}}, \psi_{j,\mathbf{k}}\rangle \sim 2^{2qj}$. Therefore, the above preconditioning can be replaced by a diagonal scaling. In fact, the diagonal scaling improves and even simplifies the wavelet preconditioning.*

3. Geometry representation in CAD

In this section, we want to summarize the geometry representation using the IGES format. It is a CAD standard written in structured records specified as IGES entities which are stored into five sections. Since it is very difficult to implement all IGES entities, we have restricted ourselves to IGES 144 where the most important geometric items are summarized in Tab. 3.1.

IGES Entities	ID numbers	IGES-codes
Line	110	`LINE`
Circular arc	100	`ARC`
Polynomial/rational B-spline curve	126	`B_SPLINE`
Composite curve	102	`CCURVE`
Surface of revolution	120	`SREV`
Tabulated cylinder	122	`TCYL`
Polynomial/rational B-spline surface	128	`SPLSURF`
Trimmed parametric surface	144	`TRM_SRF`
Transformation matrix	124	`XFORM`

Table 3.1. Most important IGES entities.

The initial CAD object is a solid $\Omega \subset \mathbb{R}^3$ bounded by a closed surface Γ which is given as a collection of N parametric surfaces $S_1, \ldots, S_N$. Each surface S_i can be trimmed or untrimmed [3]. For untrimmed surfaces, we have several representations [25] including B-splines, NURBS, surfaces of revolution and tabulated cylinders defined on rectangular domains which will be supposed to be the unit square. Note that Bézier surfaces are represented in their B-spline forms. As for each trimmed surface S_i, we have a rectangle $\mathcal{R}_i := [a_i, b_i] \times [c_i, d_i]$ containing a multiply connected region $\mathcal{D}_i$. The external and internal (when relevant) boundary curves of the domain $\mathcal{D}_i$ are supposed to be composite curves as in Fig. 3.1. That is, there are univariate smooth functions $\boldsymbol{\kappa}_i^j$ defined on $[e_i^j, f_i^j]$ such that

$$\partial \mathcal{D}_i = \bigcup_j \mathrm{Im}(\boldsymbol{\kappa}_i^j). \tag{3.10}$$

We have a parametric function $\boldsymbol{\psi}_i : \mathcal{R}_i \to \mathbb{R}^3$ which is bijective, smooth and regular in the sense that its Jacobian has maximal rank such that $\boldsymbol{\psi}_i(\mathcal{D}_i) = S_i$. Furthermore, we suppose the following conditions for each $\mathcal{D}_i$:

(B1) For all j, we have $\dot{\boldsymbol{\kappa}}_i^j(\tau) \neq \mathbf{0}$.
(B2) Each $\boldsymbol{\kappa}_i^j$ is r-times continuously differentiable for sufficiently large r.

(B3) If the terminating point $\boldsymbol{\kappa}_i^j(f_i^j)$ of $\boldsymbol{\kappa}_i^j$ and the starting point $\boldsymbol{\kappa}_i^k(e_i^k)$ of $\boldsymbol{\kappa}_i^k$ coincide, we must have the next relation to forbid cusps:

$$\lim_{t\to f_i^j-} \dot{\boldsymbol{\kappa}}_i^{j_1}(t) \neq -\lambda \lim_{t\to e_i^k+} \dot{\boldsymbol{\kappa}}_i^{j_2}(t) \qquad \forall\, \lambda > 0. \tag{3.11}$$

Our objective is to tessellate the surface Γ into a collection of four-sided domains Γ_i, i.e., (1.1), where the splitting is conforming according to Sec. 1.

4. Decomposition procedure

4.1. **Polyhedral approximation.** To facilitate the next description, we will treat trimmed as well as untrimmed surfaces in a unified approach where $\mathcal{D}_i := \square$ for each untrimmed surface S_i. As initial step of the decomposition, we approximate the curved boundaries of $\{S_i\}$ by straight line segments separated by nodes $\mathcal{A} = \{\mathbf{x}_k\} \subset \mathbb{R}^3$ as in Fig. 4.2(b). That consists in finding polygonal approximations of the planar domains $\mathcal{D}_i$ as follows. For each surface S_i, we generate a polygon $P^{(i)}$ whose nodes $\mathbf{x}_k^{(i)}$ are taken from the curved boundary of $\mathcal{D}_i$. That is, each vertex $\mathbf{x}_k^{(i)}$ of $P^{(i)}$ is the image of some boundary curve parametrization $\boldsymbol{\kappa}_i^j$ of $\mathcal{D}_i$:

$$\mathbf{x}_k^{(i)} = \boldsymbol{\kappa}_i^j(t_{k,i}) \qquad \text{where} \qquad t_{k,i} \in [e_i^j, f_i^j]. \tag{4.12}$$

The polygons should be generated in such a way that for two adjacent different surfaces S_i and S_q which share a curve $\mathcal{C}$, if $\boldsymbol{\psi}_i(\mathbf{x}_k^{(i)}) \in \mathcal{C}$, then there must exist a vertex $\mathbf{x}_l^{(q)} \in P^{(q)}$ with

$$\boldsymbol{\psi}_i(\mathbf{x}_k^{(i)}) = \boldsymbol{\psi}_q(\mathbf{x}_l^{(q)}). \tag{4.13}$$

This relation is important in order to ensure conformity of the surface decomposition. If a four-sided patch $\Gamma_i \subset S_i$ and another one $\Gamma_q \subset S_q$ are adjacent then they should share a complete edge. In particular, since the endpoints of that shared edge are corners of Γ_i and Γ_q, we must have relation (4.13) for those corners.

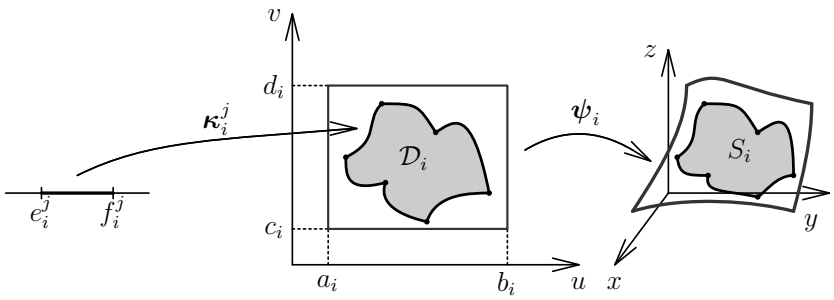
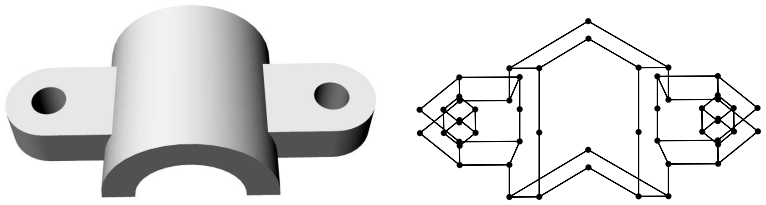
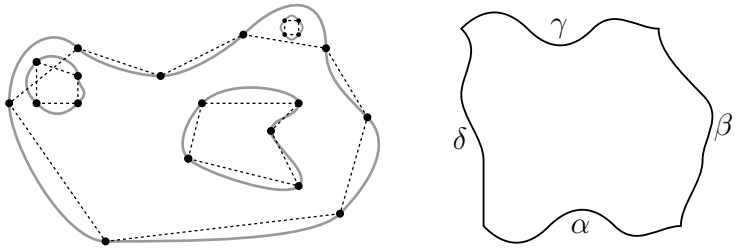


FIGURE 3.1. The boundary of $\mathcal{D}_i \subset \mathbb{R}^2$ is the image of several curves $\boldsymbol{\kappa}_i^j$. Apply $\boldsymbol{\psi}_i$ to have the trimmed surface S_i.



(a) (b)

FIGURE 4.2. (a) Given 3D model (b) Polyhedral approximation.



(a) (b)

FIGURE 4.3. (a) Imperfections in polygonal approximation (b) Four-sided region.

Let us note that if we take too few vertices, the resulting polygon may have imperfections such as different edges which intersect. For instance in Fig. 4.3(a), the dark dashed lines describe the polygonal approximation of a multiply connected domain with curved boundaries which are drawn in gray solid lines. The leftmost internal polygonal boundary intersects with the external polygonal approximation which is too coarse. But if the polygonal approximation is too fine, then it ends up with too many four-sided patches. Therefore, one has to split the curved edges adaptively while trying to maintain relation (4.13) which involves some preimage computations. Let us emphasize that only polygons having an even number of boundary vertices can be decomposed into quadrilaterals. In order to convert odd faces into even ones while inserting only few nodes, we assemble the adjacency graph and we use Dijkstra's algorithm to search for the shortest path connecting two odd polygons. One can show [28] that the number of odd faces is always even for a closed model and the odd faces can thus be converted to even ones pairwise.

4.2. **Decomposing in 2D and lifting.** Our main approach of achieving (1.1) consists in splitting the 2D regions $\mathcal{D}_i$ into four-sided regions $Q_{k,i}$ as $\mathcal{D}_i = \bigcup_k Q_{k,i}$. The final four-sided patches Γ_k are therefore the images by $\boldsymbol{\psi}_i$ of the 2D domains $Q_{k,i}$. Toward that end, we decompose each polygon $P^{(i)}$ into convex quadrilaterals $q_{k,i}$.

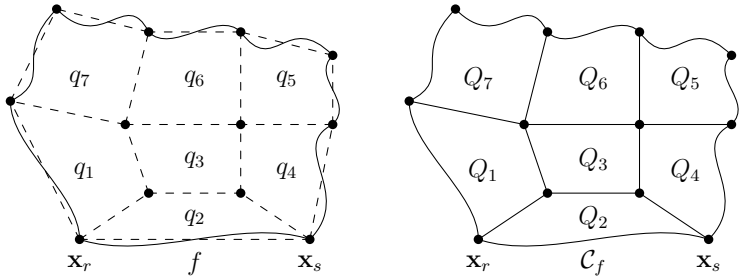


FIGURE 4.4. Subregions with curved sides from quadrangulation.

We have developed in [28] an approach that decomposes a polygon with n boundary vertices into $\mathcal{O}(n)$ convex quadrilaterals. During that quadrangulation, we do not use any boundary vertices other than the preimages $\boldsymbol{\psi}_i^{-1}(\mathbf{x}_k)$ of some $\mathbf{x}_k \in \mathcal{A}$. The four-sided domains $Q_{k,i}$ are obtained from $q_{k,i}$ by replacing the straight boundary edges of $q_{k,i}$ by the corresponding curve portion of $\mathcal{D}_i$ as shown in Fig. 4.4. Let us note that the process of curve replacement may give rise to three serious problems. First, the curve may intersect an internal edge. Second, sharp corners might be smoothened out by such a replacement. Third, it is possible that the Coons patch is not regular (see next section). In such cases, we have to make a polygonal refinement. We developed in [28] a method for making only a local repairment while keeping the large part of the quadrangulation in which we guarantee relation (4.13) when inserting new nodes. According to our experience, there are special trimmed surfaces which occur often in mechanical parts. As a result, we should treat them separately. That special treatment consists in detecting those cases and in designing a particular tessellation for each one of them.

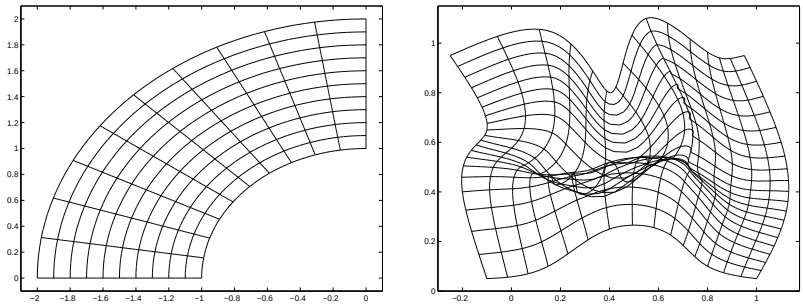
5. Transfinite interpolation

Let us consider four sufficiently smooth curves $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta} : \mathbb{R} \longrightarrow \mathbb{R}^2$. We are interested in their restriction on $[0,1]$ and we suppose that they fulfill the compatibility conditions at the corners as in Fig. 4.3(b):

$$\boldsymbol{\alpha}(0) = \boldsymbol{\delta}(0)\,, \quad \boldsymbol{\alpha}(1) = \boldsymbol{\beta}(0)\,, \quad \boldsymbol{\gamma}(0) = \boldsymbol{\delta}(1)\,, \quad \boldsymbol{\gamma}(1) = \boldsymbol{\beta}(1). \tag{5.14}$$

We assume that besides the common points in (5.14), there are no further intersection points. We are interested in generating a parametric surface $\mathbf{c}(u,v)$ defined on $\square$ such that the image of $\partial\square$ by $\mathbf{c}$ coincides with the four curves. That is, we have:

$$\begin{aligned} &\mathbf{c}(u,0) = \boldsymbol{\alpha}(u), \qquad \mathbf{c}(u,1) = \boldsymbol{\gamma}(u) \qquad \forall\, u \in [0,1], \\ &\mathbf{c}(0,v) = \boldsymbol{\delta}(v), \qquad \mathbf{c}(1,v) = \boldsymbol{\beta}(v) \qquad \forall\, v \in [0,1]. \end{aligned}$$



(a) (b)

FIGURE 5.5. (a) Regular Coons patch (b) Undesired overspill phenomenon.

This transfinite interpolation problem [13, 14] can be solved by a first order Coons patch which is defined in matrix form as

$$\mathbf{c}(u,v) := -\begin{bmatrix} -1 \\ F_0(u) \\ F_1(u) \end{bmatrix}^T \begin{bmatrix} \mathbf{0} & \mathbf{c}(u,0) & \mathbf{c}(u,1) \\ \mathbf{c}(0,v) & \mathbf{c}(0,0) & \mathbf{c}(0,1) \\ \mathbf{c}(1,v) & \mathbf{c}(1,0) & \mathbf{c}(1,1) \end{bmatrix} \begin{bmatrix} -1 \\ F_0(v) \\ F_1(v) \end{bmatrix}, \tag{5.15}$$

where F_0 and F_1 denote two arbitrary smooth functions satisfying [16, 31]:

$$F_i(j) = \delta_{ij}, \quad i,j = 0,1 \qquad \text{and} \qquad F_0(t) + F_1(t) = 1 \quad \forall\, t \in [0,1].$$

In practical cases, the blending functions F_0, F_1 are chosen as in Tab. 5.2.

Type	$F_0(t)$	$F_1(t)$
Linear	$1-t$	t
Cubic	$B_0^3(t) + B_1^3(t)$	$B_2^3(t) + B_3^3(t)$
Trigonometric	$\cos^2(0.5\pi t)$	$\sin^2(0.5\pi t)$

TABLE 5.2. Blending functions.

For most cases, a Coons patch is already regular as in Fig. 5.5(a). However, when the boundary curves become too wavy, we observe overlapping isolines indicating that the mapping is not invertible as in Fig. 5.5(b). Our next goal is to find an efficient method which can quickly verify if a Coons patch is regular.

5.1. **Bézier case.** Before describing the general case, we will examine different conditions that guarantee the regularity of a planar Coons map with Bézier boundaries. That is, we suppose that the boundary curves $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$ are expressed in terms of

their respective control points $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i$ $(i = 0, \dots, n)$ and the Bernstein polynomials $B_i^n(t) := \binom{n}{i} t^i (1-t)^{n-i}$ as follows

$$\boldsymbol{\alpha}(t) = \sum_{i=0}^{n} \boldsymbol{\alpha}_i B_i^n(t), \qquad \boldsymbol{\beta}(t) = \sum_{i=0}^{n} \boldsymbol{\beta}_i B_i^n(t),$$
$$\boldsymbol{\gamma}(t) = \sum_{i=0}^{n} \boldsymbol{\gamma}_i B_i^n(t), \qquad \boldsymbol{\delta}(t) = \sum_{i=0}^{n} \boldsymbol{\delta}_i B_i^n(t).$$

The polynomial blending function F_1 is also expressed in Bézier form $F_1(t) = \sum_{i=0}^{n} \phi_i B_i^n(t) = 1 - F_0(t)$. Furthermore, we suppose that the range of F_0 and F_1 is $[0,1]$ and we define

$$\mu := \max\{|F_1'(t)| : t \in [0,1]\}.$$

In order to express the next result, we define τ as the minimum of the following expressions over $i, j = 0, \cdots, n$

$$A_{ij} := n^2 \det[\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i, \boldsymbol{\delta}_{j+1} - \boldsymbol{\delta}_j], \quad B_{ij} := n^2 \det[\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i, \boldsymbol{\beta}_{j+1} - \boldsymbol{\beta}_j],$$
$$C_{ij} := n^2 \det[\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i, \boldsymbol{\delta}_{j+1} - \boldsymbol{\delta}_j], \quad D_{ij} := n^2 \det[\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i, \boldsymbol{\beta}_{j+1} - \boldsymbol{\beta}_j].$$

Introduce also $G := \max\{G_1, G_2\}$ where

$$\begin{aligned} G_1 &:= \max_i\{\mu\|(\boldsymbol{\beta}_i - \boldsymbol{\delta}_i) + \phi_i(\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_n + \boldsymbol{\alpha}_n - \boldsymbol{\alpha}_0) + (\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}_n)\|\}, \\ G_2 &:= \max_i\{\mu\|(\boldsymbol{\gamma}_i - \boldsymbol{\alpha}_i) + \phi_i(\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_n + \boldsymbol{\alpha}_n - \boldsymbol{\alpha}_0) + (\boldsymbol{\alpha}_0 - \boldsymbol{\gamma}_0)\|\}. \end{aligned} \tag{5.16}$$

Theorem 5.1. *Let M be a constant such that*

$$\begin{aligned} n\|\phi_j(\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i + \boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i+1}) + (\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i)\| &\le M, \\ n\|\phi_j(\boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}_i + \boldsymbol{\delta}_i - \boldsymbol{\delta}_{i+1}) + (\boldsymbol{\delta}_{i+1} - \boldsymbol{\delta}_i)\| &\le M, \end{aligned} \tag{5.17}$$

for all $i = 0, \dots, n-1$ and $j = 0, \dots, n$. If $2MG + G^2 < \tau$ and $\tau > 0$, then $\mathbf{c}$ is regular.

Proof. The partial derivatives of the Coons patch $\mathbf{c}$ are

$$\mathbf{c}_u(u,v) = F_1'(u)\mathbf{S}_1(v) + \mathbf{C}_1(u,v),$$
$$\mathbf{c}_v(u,v) = F_1'(v)\mathbf{S}_2(u) + \mathbf{C}_2(u,v),$$

where

$$\begin{aligned} \mathbf{S}_1(v) &:= \boldsymbol{\beta}(v) - \boldsymbol{\delta}(v) + F_0(v)\big(\boldsymbol{\delta}(0) - \boldsymbol{\beta}(0)\big) + F_1(v)\big(\boldsymbol{\delta}(1) - \boldsymbol{\beta}(1)\big), \\ \mathbf{S}_2(u) &:= \boldsymbol{\gamma}(u) - \boldsymbol{\alpha}(u) + F_0(u)\big(\boldsymbol{\alpha}(0) - \boldsymbol{\gamma}(0)\big) + F_1(u)\big(\boldsymbol{\alpha}(1) - \boldsymbol{\gamma}(1)\big), \end{aligned} \tag{5.18}$$

and

$$\mathbf{C}_1(u,v) := F_0(v)\boldsymbol{\alpha}'(u) + F_1(v)\boldsymbol{\gamma}'(u),$$
$$\mathbf{C}_2(u,v) := F_0(u)\boldsymbol{\delta}'(v) + F_1(u)\boldsymbol{\beta}'(v).$$

Some computations using the control points yield

$$\mathbf{S}_1(v) = \sum_{i=0}^{n}[(\boldsymbol{\beta}_i - \boldsymbol{\delta}_i) + \phi_i(\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_n + \boldsymbol{\alpha}_n - \boldsymbol{\alpha}_0) + (\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}_n)]B_i^n(v),$$

$$\mathbf{S}_2(u) = \sum_{i=0}^{n}[(\boldsymbol{\gamma}_i - \boldsymbol{\alpha}_i) + \phi_i(\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_n + \boldsymbol{\alpha}_n - \boldsymbol{\alpha}_0) + (\boldsymbol{\alpha}_0 - \boldsymbol{\gamma}_0)]B_i^n(u).$$

By using the Bézier representation, the function $\mathbf{C}_1$ can be expressed as

$$\mathbf{C}_1(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{n}[(\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i) + \phi_j(\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i - \boldsymbol{\alpha}_{i+1} + \boldsymbol{\alpha}_i)]B_i^{n-1}(u)B_j^n(v).$$

Additionally, the expression of $\mathbf{C}_2$ can be formulated in a similar way. Thus, relation (5.17) implies the following bounds

$$\|\mathbf{C}_1(u,v)\| \leq M, \qquad \|\mathbf{C}_2(u,v)\| \leq M \qquad \forall\,(u,v) \in \square.$$

On account of the multilinearity of the determinant and the fact that

$$\boldsymbol{\alpha}'(u) = \sum_{i=0}^{n-1} n(\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i)B_i^{n-1}(u)$$

with similar relations for $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, we deduce from the definition of τ that

$$\det[\mathbf{C}_1(u,v), \mathbf{C}_2(u,v)] \geq \sum_{i,j=0}^{n-1} \tau B_i^{n-1}(u)B_j^{n-1}(v) = \tau.$$

The next inequality concludes the proof

$$\det[\mathbf{c}_u, \mathbf{c}_v] \geq \det[\mathbf{C}_1(u,v), \mathbf{C}_2(u,v)] - (G^2 + 2GM).$$

□

In order to employ the technique of adaptive subdivision, let us introduce two notions. First, one can show (cf. [28]) that a Bézier surface $\sum_{i,j=0}^{n} \mathbf{E}_{ij}B_i^n(u)B_j^n(v)$ has as Jacobian a Bézier function of degree $2n$ with the next control coefficients:

$$J_{pq} := \sum_{\substack{i+k=p\\ j+l=q}} C(i,j,k,l)\frac{\binom{n}{i}\binom{n}{k}}{\binom{2n}{i+k}}\frac{\binom{n}{j}\binom{n}{l}}{\binom{2n}{j+l}}, \qquad p,q = 0,\ldots,2n, \tag{5.19}$$

where

$$C(i,j,k,l) := \frac{l}{n}\left[\frac{i}{n}D(i-1,j,k,l-1) + \left(1 - \frac{i}{n}\right)D(i,j,k,l-1)\right] + \left(1 - \frac{l}{n}\right)\left[\frac{i}{n}D(i-1,j,k,l) + \left(1 - \frac{i}{n}\right)D(i,j,k,l)\right]$$

and

$$D(i,j,k,l) := n^2 \det[\mathbf{E}_{i+1,j} - \mathbf{E}_{ij}, \mathbf{E}_{k,l+1} - \mathbf{E}_{kl}].$$

On the other hand, a Bézier surface F defined on $[a,b]\times[c,d]$ can be subdivided into four Bézier surfaces F^A, F^B, F^C, F^D which are respectively defined on

$$I^A := [a,(a+b)/2]\times[c,(c+d)/2], \qquad I^B := [a,(a+b)/2]\times[(c+d)/2,d],$$
$$I^C := [(a+b)/2,b]\times[c,(c+d)/2], \qquad I^D := [(a+b)/2,b]\times[(c+d)/2,d],$$

by using the following recursions. Suppose the control points of F are F_{ij}, $i,j = 0,\dots,n$. We define for $i,j=0,\dots,n$, $\quad k\geq 1$

$$\left\{\begin{array}{lll} F_{ij}^{[0]} := F_{ij} & \text{and} & F_{ij}^{[k]} := 0.5\big(F_{i-1,j}^{[k-1]} + F_{ij}^{[k-1]}\big) \\ P_{ij}^{[0]} := F_{ij}^{[i]} & \text{and} & P_{ij}^{[k]} := 0.5\big(P_{i,j-1}^{[k-1]} + P_{ij}^{[k-1]}\big) \\ Q_{ij}^{[0]} := F_{nj}^{[n-i]} & \text{and} & Q_{ij}^{[k]} := 0.5\big(Q_{i,j-1}^{[k-1]} + Q_{ij}^{[k-1]}\big) \end{array}\right. \tag{5.20}$$

The control points of F^A, F^B, F^C and F^D are respectively $A_{ij} := P_{ij}^{[j]}$, $B_{ij} := P_{in}^{[n-j]}$, $C_{ij} := Q_{ij}^{[j]}$, $D_{ij} := Q_{in}^{[n-j]}$. We have in particular

$$F(u,v) = F^r(u,v) \quad \text{for} \quad (u,v)\in I^r, \quad \text{where} \quad r = A,B,C,D.$$

We can apply the same subdivision technique to each of the resulting 4 Bézier surfaces. A recursive application of that subdivision on the unit square generates a uniform grid consisting of σ^2 little squares as illustrated in Fig. 5.6(a).

Theorem 5.2. *Suppose that the Coons patch* $\mathbf{c}$ *defined with* $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}$ *is regular. Suppose that its Jacobian function* J *has been subdivided into* σ^2 *functions* J^{ij} *defined on*

$$I^{ij} := [(i-1)/\sigma, i/\sigma]\times[(j-1)/\sigma, j/\sigma], \qquad i,j = 1,\dots,\sigma,$$

and with Bézier coefficients J_{pq}^{ij}, $p,q=0,\dots,2n$. *Then, for a sufficiently large* σ *all coefficients* J_{pq}^{ij} *have the same sign.*

Proof. Since $\mathbf{c}$ is regular, the Jacobian $J(u,v)$ must be of constant sign because it is never zero. Without loss of generality we suppose that it is positive:

$$J(u,v) > 0 \qquad \forall (u,v)\in\square.$$

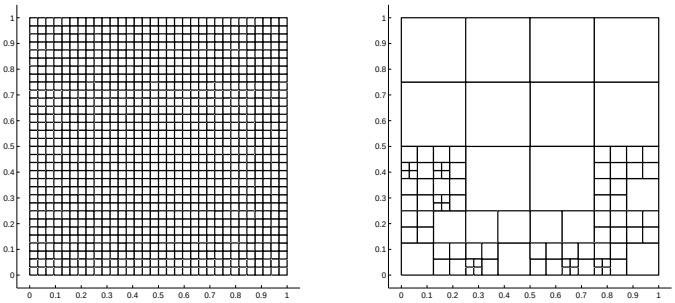
Since the function J is continuous on the compact $\square$, there must exist some $\lambda>0$ such that

$$J(u,v)\geq\rho \qquad \forall\,(u,v)\in\square. \tag{5.21}$$

To simplify notation we fix (i,j) and denote by $[a,b]\times[c,d]$ the domain I^{ij}. Let $P^{ij}(u_1,\dots,u_{2n};v_1,\dots,v_{2n})$ be the blossom [27] of J^{ij}:

$$J^{ij}(u,v) = P^{ij}(u,..[2n]..,u;v,..[2n]..,v)$$

where we use the notation $s,..[m]..,s$ for an m-times repetition of s.



(a) (b)

FIGURE 5.6. Multiple subdivisions: (a) uniform (b) adaptive.

Define $h := 1/(2n\sigma)$ and $a_p := a+ph$, $c_q := c+qh$ for $p = 0, \ldots, 2n$ and $q = 0, \ldots, 2n$. By applying the multivariate Taylor expansion of P^{ij} at $(a_p, ..[2n].., a_p; c_q, ..[2n].., c_q)$ and the symmetry property [27] of the blossom function, we obtain

$$J^{ij}_{pq} = J^{ij}(a_p, c_q) + \mathcal{O}(h^2). \tag{5.22}$$

Combining (5.21) and (5.22), there must exist some constant $C > 0$ such that

$$J^{ij}_{pq} = J^{ij}(a_p, c_q) + J^{ij}_{pq} - J^{ij}(a_p, c_q) \geq J(a_p, c_q) - Ch^2 \geq \lambda - Ch^2.$$

Since $Ch^2 = C/(2n\sigma)^2$ tends to 0 as σ tends to infinity, we deduce that $J^{ij}_{pq} > 0$ for σ sufficiently large. □

We omit the proof of the following results which can be proved in a very similar fashion.

Theorem 5.3. *Let* **c** *be a Coons patch that is not regular. Then, in the situation of Theorem 5.2 and for sufficiently large* σ*, there must exist* (i_1, j_1) *and* (i_2, j_2) *such that*

$$\left.\begin{array}{l} J^{i_1,j_1}_{pq} > 0 \\ J^{i_2,j_2}_{pq} < 0 \end{array}\right\} \quad \forall\, p, q = 0, \ldots, 2n.$$

In practice, we do not need to subdivide the Jacobian uniformly because we can perform adaptive subdivision. We start from a single Jacobian function written in Bézier form defined on the unit square. Then, we split it recursively by adaptively using the former subdivision techniques. That is, we subdivide only those Bézier functions that have Bézier coefficients J^{ij}_{pq} with different signs. The preceding two theorems serve as abortion conditions for that recursion. An instance of that adaptive subdivision process is illustrated in Fig. 5.6(b).

Note that the adaptive refinement is exclusively done for the purpose of verifying whether a Coons map is regular. In particular, the new corner points which are

observed in Fig. 5.6(b) do not become hanging nodes in the final four-sided decomposition. When considering an individual four-sided patch, the adaptive refinement is only performed in order to avoid the unnecessarily fine uniform refinement in Fig. 5.6(a). This adaptive refinement approach can be better understood if we consider the pipeline in Fig. 5.7.

5.2. **General case.** In CAD models, the bounding curves $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$ are not necessarily in Bézier forms. But it is always advantageous to have discrete criteria which can be easily verified in practice. As a first step, we approximate them by Bézier curves. Then, we apply the methods of the previous section to the resulting Bézier curves. Now, we want to know to which extent that approximation affects the former results. The following discussion might be obvious for specialists in approximation theory but we include them anyway for the sake of completeness. Let us recall [27] the following property of the Bernstein polynomials.

Theorem 5.4. *Let f be a continuously differentiable function. Define the Bézier function $A^n f$ by*

$$(A^n f)(t) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_i^n(t).$$

Then, we have uniform convergence: $\|f - (A^n f)\|_\infty \to 0$ and $\|f' - (A^n f)'\|_\infty \to 0$ as $n \to \infty$.

Let us denote by $\mathbf{c}^n$ the Coons patch corresponding to the Bézier curves $\widetilde{\boldsymbol{\alpha}} := A^n \boldsymbol{\alpha}$, $\widetilde{\boldsymbol{\beta}} := A^n \boldsymbol{\beta}$, $\widetilde{\boldsymbol{\gamma}} := A^n \boldsymbol{\gamma}$, $\widetilde{\boldsymbol{\delta}} := A^n \boldsymbol{\delta}$.

Corollary 5.5. *If $\det(\mathbf{c}_u, \mathbf{c}_v)$ is of fixed sign (say positive) on $\square$, then $\det(\mathbf{c}_u^n, \mathbf{c}_v^n)$ is also of the same fixed sign for sufficiently large n.*

Proof. We are only sketching the proof because it is not very difficult. Since the functions F_0, F_0', F_1, F_1' are continuous on the compact interval $[0,1]$, there exists $R \in (0, \infty)$ such that

$$\|F_0\|_\infty < R, \qquad \|F_0'\|_\infty < R, \qquad \|F_1\|_\infty < R, \qquad \|F_1'\|_\infty < R.$$

Let us denote by $\widetilde{\mathbf{S}}_1$ and $\widetilde{\mathbf{S}}_2$ the same relation as (5.18) but with respect to $\widetilde{\boldsymbol{\alpha}}$, $\widetilde{\boldsymbol{\beta}}$, $\widetilde{\boldsymbol{\gamma}}$, $\widetilde{\boldsymbol{\delta}}$. After a few computations, one can show that

$$\begin{aligned}\mathbf{a}_u^n := \mathbf{c}_u - \mathbf{c}_u^n = F_1'(u)[\mathbf{S}_1(v) - \widetilde{\mathbf{S}}_1(v)] + F_0(v)\big(\boldsymbol{\alpha}'(u) - \widetilde{\boldsymbol{\alpha}}'(u)\big) \\ + F_1(v)\big(\boldsymbol{\gamma}'(u) - \widetilde{\boldsymbol{\gamma}}'(u)\big).\end{aligned}$$

A similar relation holds for $\mathbf{a}_v^n := \mathbf{c}_v - \mathbf{c}_v^n$. We have

$$\|\mathbf{c}_u - \mathbf{c}_u^n\|_\infty = R\|\mathbf{S}_1(v) - \widetilde{\mathbf{S}}_1(v)\|_\infty + R\|\boldsymbol{\alpha}'(u) - \widetilde{\boldsymbol{\alpha}}'(u)\|_\infty + R\|\boldsymbol{\gamma}'(u) - \widetilde{\boldsymbol{\gamma}}'(u)\|_\infty$$

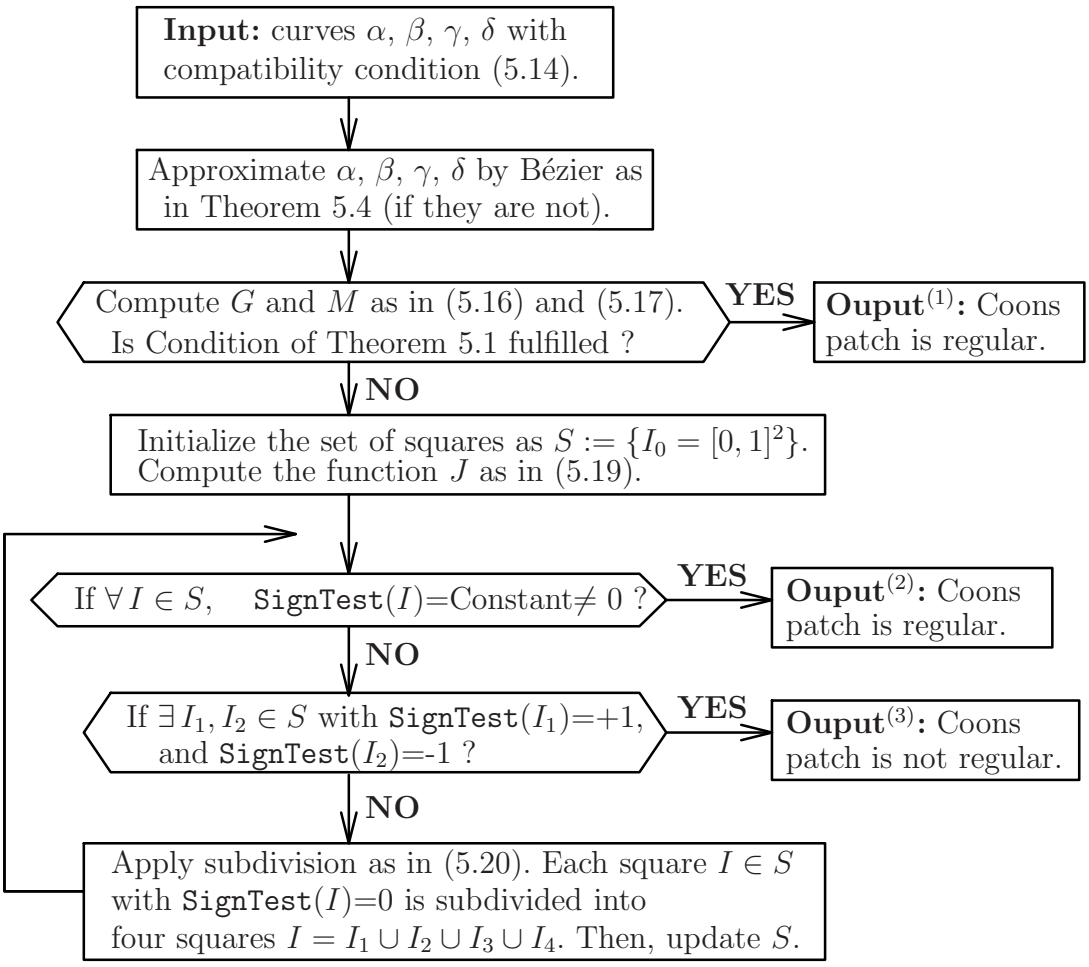


FIGURE 5.7. Pipeline for the parametrization

where

$$\mathbf{S}_1(v) - \widetilde{\mathbf{S}}_1(v) = \boldsymbol{\beta}(v) - \widetilde{\boldsymbol{\beta}}(v) + \boldsymbol{\delta}(v) - \widetilde{\boldsymbol{\delta}}(v).$$

Since $\mathbf{c}$ is continuously differentiable, there should exist some $0 < \lambda < 1$ such that $\det[\mathbf{c}_u, \mathbf{c}_v] > \lambda$ for all $(u, v) \in \square$. Let $B \geq 1$ be an upper bound of $\|\mathbf{c}_u(u, v)\|$ and $\|\mathbf{c}_v(u, v)\|$ for all $(u, v) \in \square$. Due to the former theorem, for $\varepsilon := \lambda/(4B)$ and for sufficiently large n, we have $\|\mathbf{a}_u^n\|_\infty = \|\mathbf{c}_u - \mathbf{c}_u^n\|_\infty < \varepsilon/2$. Similarly, we can show $\|\mathbf{a}_v^n\|_\infty = \|\mathbf{c}_v - \mathbf{c}_v^n\|_\infty < \varepsilon/2$. Since

$$\begin{aligned}\det[\mathbf{c}_u^n, \mathbf{c}_v^n] &= \det[\mathbf{c}_u, \mathbf{c}_v] + \det[\mathbf{a}_u^n, \mathbf{a}_v^n] - \det[\mathbf{a}_u^n, \mathbf{c}_v] - \det[\mathbf{c}_u, \mathbf{a}_v^n] \\ &\geq \det[\mathbf{c}_u, \mathbf{c}_v] - (\|\mathbf{a}_u^n\|_\infty \|\mathbf{a}_v^n\|_\infty + B\|\mathbf{a}_u^n\|_\infty + B\|\mathbf{a}_v^n\|_\infty),\end{aligned}$$

we have $\det[\mathbf{c}_u^n, \mathbf{c}_v^n] > \lambda/4 > 0$. □

A similar Corollary can surely be proved for the case where the determinant does not have any fixed sign. For practical purposes, we consider some threshold $\Delta > 0$ and we suppose that we have regularity only if $\det[\mathbf{c}_u, \mathbf{c}_v] \geq \Delta$. That is, if the determinant

is strictly positive but it is very small, we suppose that the Coons patch is not regular. In other words, we choose some large number n and we apply the theorems in Sec. 5.1 to $\det[\mathbf{c}_u^n, \mathbf{c}_v^n]$. When we do not have regularity for the approximation $\mathbf{c}^n$, we make refinement of the CAD surfaces as described in Sec. 4.

In order to summarize the whole process, we can see in Fig. 5.7 a flowchart which displays the parametrization pipeline. For a Bézier surface defined on a square I, $\texttt{SignTest}(I)$ is a routine which produces three values: $+1$, -1, 0. It is $+1$ if all control points with respect to I are positive. It is -1 if they are all negative. Otherwise, it is zero. Theorem 5.2 and Theorem 5.3 ensure that the loop in the flowchart is not infinite. Thus, the loop will break and exit at the second or third output gate.

6. Global continuity

The mapping $\boldsymbol{\gamma}_i$ from relation (1.1) will be the composition of the base surface $\boldsymbol{\psi}_i$ from Sec. 3 and the Coons map of each 2D four sided domain. The boundaries $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ of the Coons maps are either straight lines or restrictions of the images of the curves $\boldsymbol{\kappa}_i^j$ as illustrated in Fig. 6.8. Since the Coons patch (5.15) requires $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ to be defined on $[0,1]$, we use one of the next two representations for $t \in [0,1]$:

$$\begin{aligned} \boldsymbol{\mu}(t) &= t\mathbf{B} + (1-t)\mathbf{A}, && \boldsymbol{\mu} = \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \\ \boldsymbol{\mu}(t) &= \boldsymbol{\kappa}_i^j(t\theta + (1-t)s), && \boldsymbol{\mu} = \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}. \end{aligned} \tag{6.23}$$

The first representation is used when $\boldsymbol{\mu}$ is a straight internal edge joining two internal nodes $\mathbf{A}$ and $\mathbf{B}$ while the second one applies if $\boldsymbol{\mu}$ is a restriction of $\boldsymbol{\kappa}_i^j$ on some $[s, \theta] \subset [e_i^j, f_i^j]$.

An arbitrary parametrization of the curves $\boldsymbol{\kappa}_i^j$ from (3.10) does not guarantee the global continuity in (1.2). Since we cannot modify the base surfaces $\boldsymbol{\psi}_i$, we want to replace the 2D curves $\boldsymbol{\kappa}_i^j$ by $\widetilde{\boldsymbol{\kappa}}_i^j$ such that they have the same shapes ($\mathrm{im}(\boldsymbol{\kappa}_i^j) = \mathrm{im}(\widetilde{\boldsymbol{\kappa}}_i^j)$)

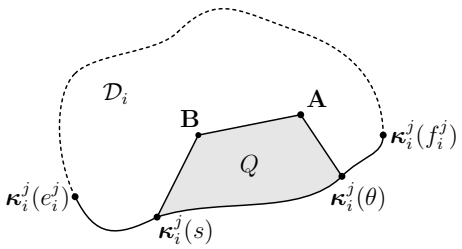
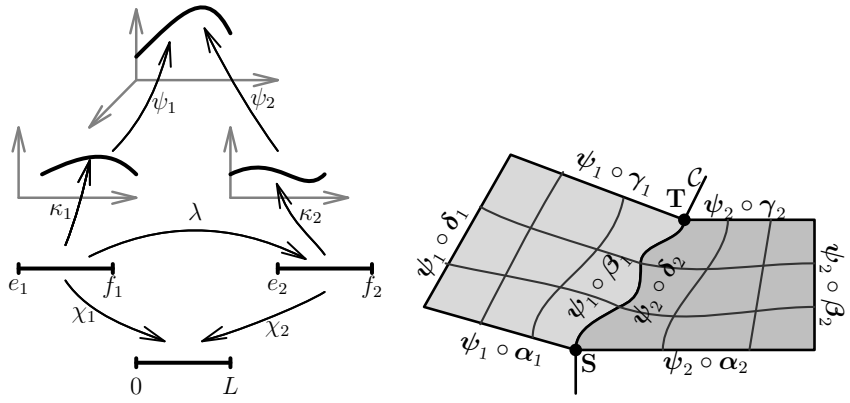


Figure 6.8. Restriction of $\boldsymbol{\kappa}_i^j$ to have boundary of a Coons patch inside the domain $\mathcal{D}_i$



(a) (b)

FIGURE 6.9. (a) Involved functions for adjacent Coons maps (b) Two Coons maps match well at interface curve $\mathcal{C}$ for chord length parametrization.

but they have different parametrizations. Let us introduce the length function

$$\chi_i^j(t) := \int_{e_i^j}^t \left\| \frac{d\boldsymbol{\rho}_i^j}{dt}(\theta) \right\| d\theta \quad \text{where} \quad \boldsymbol{\rho}_i^j := \boldsymbol{\psi}_i \circ \boldsymbol{\kappa}_i^j.$$

On account of the properties of $\boldsymbol{\kappa}_i^j$ and $\boldsymbol{\psi}_i$, let us observe that

$$\frac{d\chi_i^j}{dt}(t) = \left\| \frac{d\boldsymbol{\rho}_i^j}{dt}(t) \right\| \neq 0 \qquad \forall t \in [e_i^j, f_i^j]. \tag{6.24}$$

Hence, there is an inverse function $\phi_i^j := (\chi_i^j)^{-1}$ and our method consists in replacing the function $\boldsymbol{\kappa}_i^j$ by the chord length parametrization $\widetilde{\boldsymbol{\kappa}}_i^j := \boldsymbol{\kappa}_i^j \circ \phi_i^j$.

Let us consider two adjacent trimmed surfaces S_i and S_j. In order to facilitate the presentation, we may suppose that they are S_1 and S_2 and we omit the superscripts. In the following, we denote two Coons maps by $\mathbf{c}_1$ and $\mathbf{c}_2$ which are respectively incident upon $\mathrm{im}(\boldsymbol{\kappa}_1)$ and $\mathrm{im}(\boldsymbol{\kappa}_2)$. In addition, we will assume that we have coincidence of the images but we do not necessarily have *pointwise* agreement. That is, one of the following relations hold for some $u_1, u_2, v_1, v_2 \in \{0, 1\}$

$$\begin{aligned}
&\mathrm{Im}(\boldsymbol{\psi}_1[\mathbf{c}_1(u_1, \cdot)]) = \mathrm{Im}(\boldsymbol{\psi}_2[\mathbf{c}_2(u_2, \cdot)]),\\
&\mathrm{Im}(\boldsymbol{\psi}_1[\mathbf{c}_1(u_1, \cdot)]) = \mathrm{Im}(\boldsymbol{\psi}_2[\mathbf{c}_2(\cdot, v_2)]),\\
&\mathrm{Im}(\boldsymbol{\psi}_1[\mathbf{c}_1(\cdot, v_1)]) = \mathrm{Im}(\boldsymbol{\psi}_2[\mathbf{c}_2(u_2, \cdot)]),\\
&\mathrm{Im}(\boldsymbol{\psi}_1[\mathbf{c}_1(\cdot, v_1)]) = \mathrm{Im}(\boldsymbol{\psi}_2[\mathbf{c}_2(\cdot, v_2)]).
\end{aligned} \tag{6.25}$$

Further, we denote by $[e_1, f_1]$ and $[e_2, f_2]$ the intervals of definition of $\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1$ and $\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2$ which have coinciding images $\mathcal{C} := (\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1)([e_1, f_1]) = (\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)([e_2, f_2])$.

Note that the curved edge which is shared by the adjacent images of the two Coons maps is not necessarily the whole $\mathcal{C}$ (but it is included in $\mathcal{C}$) because of relation (4.12) and (4.13) (see also Fig. 6.8).

Theorem 6.1. *Suppose that we use the chord length parametrization $\widetilde{\boldsymbol{\kappa}}_1 = \boldsymbol{\kappa}_1 \circ \phi_1$ and $\widetilde{\boldsymbol{\kappa}}_2 = \boldsymbol{\kappa}_2 \circ \phi_2$ and that one of the relations in* (6.25) *is fulfilled. Then, the images of the Coons maps $\widetilde{\mathbf{c}}_1$, $\widetilde{\mathbf{c}}_2$ have sides which agree pointwise irrespective of the blending functions F_0 and F_1. That is, one of the following relations must hold for all $t \in [0,1]$:*

$$\begin{aligned} &\boldsymbol{\psi}_1[\widetilde{\mathbf{c}}_1(u_1,t)] = \boldsymbol{\psi}_2[\widetilde{\mathbf{c}}_2(u_2,t)], \qquad &&\boldsymbol{\psi}_1[\widetilde{\mathbf{c}}_1(u_1,t)] = \boldsymbol{\psi}_2[\widetilde{\mathbf{c}}_2(t,v_2)], \\ &\boldsymbol{\psi}_1[\widetilde{\mathbf{c}}_1(t,v_1)] = \boldsymbol{\psi}_2[\widetilde{\mathbf{c}}_2(u_2,t)], \qquad &&\boldsymbol{\psi}_1[\widetilde{\mathbf{c}}_1(t,v_1)] = \boldsymbol{\psi}_2[\widetilde{\mathbf{c}}_2(t,v_2)], \end{aligned} \tag{6.26}$$

where $u_1, u_2, v_1, v_2 \in \{0,1\}$.

Proof. Since $\boldsymbol{\rho}_2$ is invertible, we may define $\lambda := \boldsymbol{\rho}_2^{-1} \circ \boldsymbol{\rho}_1$. By definition of the chord length parametrization, we have

$$\chi_1(t) = \int_{e_1}^{t} \left\| \frac{d\boldsymbol{\rho}_1}{dt}(\theta) \right\| d\theta \qquad \chi_2(t) = \int_{e_2}^{t} \left\| \frac{d\boldsymbol{\rho}_2}{dt}(\theta) \right\| d\theta.$$

From the definition of λ, we obtain by chain rule

$$\frac{d\boldsymbol{\rho}_1}{dt}(t) = \frac{d\boldsymbol{\rho}_2}{dt}\big(\lambda(t)\big)\lambda'(t).$$

As a consequence, we deduce (cf. Fig. 6.9):

$$\chi_1(t) = \int_{e_1}^{t} \left\| \frac{d\boldsymbol{\rho}_1}{dt}(\theta) \right\| d\theta = \int_{e_2}^{\lambda(t)} \left\| \frac{d\boldsymbol{\rho}_2}{dt}(\sigma) \right\| d\sigma = \chi_2\big(\lambda(t)\big).$$

Hence $\chi_2^{-1} \circ \chi_1 = \lambda = \boldsymbol{\rho}_2^{-1} \circ \boldsymbol{\rho}_1$. Therefore, if we denote the total length by $L := \chi_1(f_1) = \chi_2(f_2)$, we have

$$(\boldsymbol{\rho}_2 \circ \phi_2)(t) = (\boldsymbol{\rho}_1 \circ \phi_1)(t) \qquad \forall\, t \in [0, L]. \tag{6.27}$$

Let $\mathbf{S}, \mathbf{T} \in \mathcal{C}$ be the starting and terminating points of the common edge. As explained in relation (6.23) (see Fig. 6.8), there are $s_1, \theta_1 \in [e_1, f_1]$ and $s_2, \theta_2 \in [e_2, f_2]$ such that

$$\begin{aligned} \mathbf{S} &= (\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1)(s_1) = (\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)(s_2) \\ \mathbf{T} &= (\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1)(\theta_1) = (\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)(\theta_2). \end{aligned}$$

From (6.27), we deduce $\chi_1(s_1) = \chi_2(s_2) =: s \in [0, L]$ and $\chi_1(\theta_1) = \chi_2(\theta_2) =: \theta \in [0, L]$. We assume only the first relation of (6.25) with $u_1 = 1$ and $u_2 = 0$ and we are proving the first equality of (6.26) while the other 15 cases can be treated in a similar manner. This means, let us assume that the sides of the Coons maps $\mathbf{c}_1$

and $\mathbf{c}_2$ are $\boldsymbol{\beta}_1$ and $\boldsymbol{\delta}_2$. That is, $\boldsymbol{\beta}_1$ is the restriction of $\boldsymbol{\kappa}_1$ on $[s_1, \theta_1]$ and $\boldsymbol{\delta}_2$ is the restriction of $\boldsymbol{\kappa}_2$ on $[s_2, \theta_2]$:

$$\boldsymbol{\beta}_1(t) = \boldsymbol{\kappa}_1[t\theta_1 + (1-t)s_1], \quad \boldsymbol{\delta}_2(t) = \boldsymbol{\kappa}_2[t\theta_2 + (1-t)s_2], \quad t \in [0,1].$$

Hence, $\boldsymbol{\psi}_1 \circ \boldsymbol{\beta}_1 \circ \phi_1(t) = (\boldsymbol{\rho}_1 \circ \phi_1)[t\theta + (1-t)s]$ for $t \in [0,1]$. Similarly, for $\boldsymbol{\delta}_2$ we have $\boldsymbol{\psi}_2 \circ \boldsymbol{\delta}_2 \circ \phi_2(t) = (\boldsymbol{\rho}_2 \circ \phi_2)[t\theta + (1-t)s]$ for $t \in [0,1]$. Relation (6.27) yields

$$\boldsymbol{\psi}_1 \circ \boldsymbol{\beta}_1 \circ \phi_1(t) = \boldsymbol{\psi}_2 \circ \boldsymbol{\delta}_2 \circ \phi_2(t) \qquad \forall\, t \in [0,1]. \tag{6.28}$$

We denote $\widetilde{\boldsymbol{\beta}}_1 := \boldsymbol{\beta}_1 \circ \phi_1$ and $\widetilde{\boldsymbol{\beta}}_2 := \boldsymbol{\beta}_2 \circ \phi_2$. Since we have $\widetilde{\mathbf{c}}_1(1, v) = \widetilde{\boldsymbol{\beta}}_1(v)$ and $\widetilde{\mathbf{c}}_2(0, v) = \widetilde{\boldsymbol{\delta}}_2(v)$ independently of F_0, F_1, we can deduce the result from (6.28). □

7. Numerical Results

7.1. **Examples of geometries.** We first like to demonstrate the parametric representation that is constructed by our algorithm. We consider four CAD models consisting respectively of 14, 34, 28, 26 faces. After using our decomposition techniques, the resulting tessellation have respectively 38, 78, 212, 93 patches. For all geometries, we present in Fig. 7.10(a)–7.10(d) the patches with the mesh that is obtained on refinement level 3.

In order to measure the performance of our algorithm in practice, we have done two tests. The first one specifies the time for the tessellation operation: file loading, IGES entity parsing and the four-sided decomposition. The second test is the evaluation of the points on a tensor product grid with dyadic stepsize 2^{-J}. We have used level $J = 6$ which requires 4225 points per patch. The computation measurement has been performed on a machine with processor Intel Core 2.16GHz running Windows Vista. The results of the tests are gathered in Tab. 7.3 that shows that the decomposition process is in general very fast. It is worth mentioning that the time for the evaluation depends on two factors. First, it depends on the number of four-sided patches and the level. On the other hand, the properties of the underlying surface $\boldsymbol{\psi}_i$ that we met in Sec. 3 affects the evaluation performance. For instance, a planar patch and a NURBS patch having high smoothness have different performance of point evaluation. Averagely, about 0.051 seconds per patch is need for the evaluation process in practical CAD models at level $J = 6$.

In order to avoid the dependence of the computing times on the patch representation we transfer only the grid points of the discretization level J to the wavelet BEM solver. Especially this decouples both codes. Arbitrary points on the surface

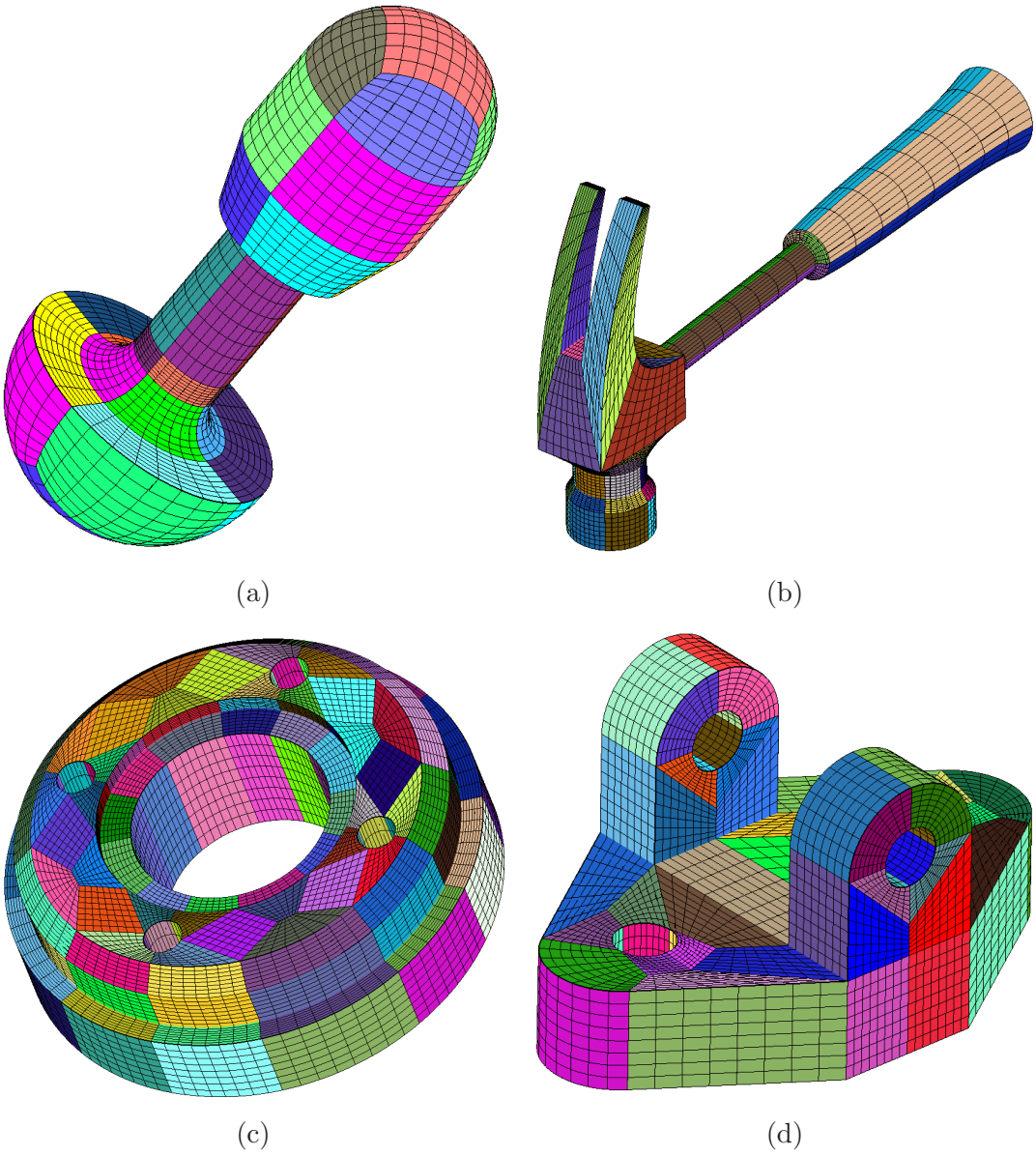


FIGURE 7.10. Constructed parametrizations with mesh on the level 3.

Model	#trim surf	#untrim surf	#patches	Decomposition	Evaluation
(a)	4	10	38	0.203 sec.	1.716 sec.
(b)	6	28	78	0.655 sec.	4.883 sec.
(c)	6	22	212	0.686 sec.	10.124 sec.
(d)	6	20	93	0.670 sec	5.913 sec

TABLE 7.3. Results about CAD models.

are evaluated via interpolation with sufficiently high order. That is quadratic interpolation in case of piecewise constant and fourth-order interpolation in case of piecewise bilinear wavelets.

7.2. **BEM calculations.** In order to demonstrate the efficiency of the wavelet Galerkin method we present here only a representative example which demonstrates our algorithm quantitatively.

Let Ω be the tool shown in Fig. 7.10(c), represented by 212 patches. We shall seek the function $U \in H^1_{loc}(\Omega^c)$ satisfying the following exterior Dirichlet problem for the Laplacian:

$$\begin{aligned} \Delta U &= 0 && \text{in } \Omega^c, \\ U &= f && \text{on } \Gamma, \\ U &= \mathcal{O}(\|\mathbf{x}\|^{-1}) && \text{as } \|\mathbf{x}\| \to \infty. \end{aligned} \tag{7.29}$$

The ansatz

$$U(\mathbf{x}) = \frac{1}{4\pi} \int_\Gamma \frac{u(\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|} d\sigma_{\mathbf{y}}, \quad \mathbf{x} \in \Omega, \tag{7.30}$$

yields the Fredholm boundary integral equation of the first kind $\mathcal{V}u = f$ for the unknown density function $u \in H^{-1/2}(\Gamma)$. Herein, $\mathcal{V} : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$ denotes the *single layer operator* given by

$$(\mathcal{V}u)(\mathbf{x}) := \frac{1}{4\pi} \int_\Gamma \frac{u(\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|} d\sigma_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma. \tag{7.31}$$

The function

$$U(\mathbf{x}) = \frac{1}{\sqrt{(x-0.5)^2 + (y-0.3)^2 + (z-0.5)^2}},$$

where $(0.5, 0.3, 0.5) \in \Omega$, satisfies (7.29) if we choose $f := U|_\Gamma$. That way we constructed a problem where the solution is known analytically.

We discretize the given boundary integral equation by piecewise constant wavelets with three vanishing moments which is consistent to (2.7). We compute the discrete solution $\mathbf{U}_J := [U_J(\mathbf{x}_i)]$ according to (7.30) from the approximated density u_J, where the evaluation points $\mathbf{x}_i$ are on the sphere with radius 1.5. If the density u is in $H^1(\Gamma)$ we obtain for $\mathbf{x} \in \Omega^c$ the pointwise estimate $|U(\mathbf{x}) - U_J(\mathbf{x})| \le c_{\mathbf{x}} \|u - u_J\|_{H^{-2}(\Gamma)} \lesssim 2^{-3J} \|u\|_{H^1(\Gamma)}$, cf. [33]. Nevertheless, we can only expect a reduced rate of convergence since $u \notin H^1(\Gamma)$ due to the presence of corner and edge singularities.

J	N_J	$\|\mathbf{U} - \mathbf{U}_J\|_\infty$		a-priori	a-posteriori	cpu-time	#iterations
1	848	1.8e-2		28 %	20 %	1 sec.	37
2	3392	8.3e-3	(2.2)	11 %	7.4 %	4 sec.	40
3	13568	8.6e-4	(9.6)	4.0 %	1.9 %	38 sec.	83
4	54272	5.1e-5	(17)	1.2 %	3.9e-1 %	248 sec.	70
5	217088	1.4e-5	(3.7)	3.2e-1 %	8.5e-2 %	1215 sec.	81

TABLE 7.4. Numerical results with respect to the tool.

We present in Tab. 7.4 the results produced by the wavelet Galerkin scheme. The 3rd column refers to the absolute ℓ^∞-error of the point evaluations $\mathbf{U}_J$. In the 4th and 5th columns we tabulate the number of relevant coefficients. Note that on the level 5 only 700 and 200 relevant matrix coefficients per degree of freedom remain after the a-priori and a-posteriori compression, respectively. One figures out of the 6th column the over-all computing times, including compressing, assembling and solving the linear system of equations. Since the single layer operator is of order -1 preconditioning becomes an issue. Therefore, in the last column we specified the number of cg-iterations, preconditioned by a diagonal scaling (cf. Thm. 2.3 and the remark below). The computations have been performed on a single processor of a Sun FireV20z Server with two 2.2 MHz AMD Opteron processors and 4 GB main memory per processor.

References

[1] J. Ahlberg, E. Nilson, and J. Walsh. *The theory of splines and their applications.* Academic Press, New York, 1967.

[2] G. Beylkin, R. Coifman, and V. Rokhlin. The fast wavelet transform and numerical algorithms. *Comm. Pure and Appl. Math.* **44** (1991) 141–183.

[3] G. Brunnett. Geometric design with trimmed surfaces. *Comput. Suppl.* **10** (1995) 101–115.

[4] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Pure Appl. Math.* **45** (1992) 485–560.

[5] W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for boundary integral equations — optimal complexity estimates. *SIAM J. Numer. Anal.* **43** (2006) 2251–2271.

[6] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.* **63** (1992) 315–344.

[7] W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal spline-wavelets on the interval – stability and moment conditions. *Appl. Comp. Harm. Anal.* **6** (1999) 259–302.

[8] W. Dahmen and R. Schneider. Composite wavelet bases for operator equations. *Math. Comp.* **68** (1999) 1533–1567.

[9] W. Dahmen and R. Schneider. Wavelets on manifolds I. Construction and domain decomposition. *SIAM J. Math. Anal.* **31** (1999) 184–230.

[10] G. Farin. Discrete Coons patches. *Comput. Aided Geom. Des.* **16**, No. 7 (1999) 691–700.

[11] G. Farin, S. Hahmann, and G. Brunnett. in: Geometric Modelling, Dagstuhl 2002, *Computing* **72**, (2004) 1–246.

[12] M. Floater. Chordal cubic spline interpolation is fourth order accurate. *IMA J. Numer. Anal.* **26** (2006) 25–33.

[13] A. Forrest. On Coons and other methods for the representation of curved surfaces, *Comput. Graph. Img. Process.* **1** (1972) 341–359.

[14] W. Gordon and C. Hall. Construction of curvilinear co-ordinate systems and applications to mesh generation. *Int. J. Numer. Methods Eng.* **7** (1973) 461–477.

[15] W. Gordon and C. Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numer. Math.* **21** (1973) 109–129.

[16] W. Gordon. Sculptured surface interpolation via blending-function methods. *Research Report, Department of Mathematics and Computer Science, Drexel University*, Philadelphia, 1982.

[17] L. Greengard and V. Rokhlin. A fast algorithm for particle simulation. *J. Comput. Phys.* **73** (1987) 325–348.

[18] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing* **64** (1999) 89–108.

[19] W. Hackbusch and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.* **54** (1989) 463–491.

[20] H. Harbrecht. A Newton method for Bernoulli's free boundary problem in three dimensions. *Preprint 321*, Berichtsreihe des SFB 611, Universitt Bonn, Germany, 2007. submitted to Computing.

[21] H. Harbrecht and T. Hohage. Fast Methods for Three-Dimensional Inverse Obstacle Scattering. *J. Integral Equations Appl.* **19** (2007) 237–260.

[22] H. Harbrecht and R. Schneider. Biorthogonal wavelet bases for the boundary element method. *Math. Nachr.* **269–270** (2004) 167–188.

[23] H. Harbrecht and R. Schneider. Wavelet Galerkin Schemes for Boundary Integral Equations – Implementation and Quadrature. *SIAM J. Sci. Comput.* **27** (2002) 1347–1370.

[24] H. Harbrecht and R. Stevenson. Wavelets with patchwise cancellation properties. *Math. Comput.* **75** (2006) 1871–1889.

[25] J. Hoschek and D. Lasser. *Grundlagen der geometrischen Datenverarbeitung.* Teubner, Stuttgart, 1989.

[26] T. von Petersdorff, R. Schneider, and C. Schwab. Multiwavelets for second kind integral equations. *SIAM J. Num. Anal.* **34** (1997) 2212–2227.

[27] H. Prautzsch, W. Boehm, and M. Paluszny. *Bézier and B-Spline techniques.* Springer, Berlin, 2002.

[28] M. Randrianarivony. Geometric processing of CAD data and meshes as input of integral equation solvers. *PhD thesis*, Technische Universität Chemnitz, 2006.

[29] J.A. Roulier. Specifying the arc length of Bézier curves. *Computer Aided Geometric Design* **10** (1993) 25–56.

[30] R. Schneider. *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur Lösung großer vollbesetzter Gleichungssysteme.* Teubner, Stuttgart, 1998.

[31] G. Schulze. *Segmentation operators on Coons' patches.* in: Mathematical methods in computer aided geometric design, edts. T. Lyche, L. Schumaker, Academic Press, Boston, 1989, pp. 561–572.

[32] U.S. Product Data Association. Initial Graphics Exchange Specification. IGES 5.3. Trident Research Center, SC, 1996.

[33] W.L. Wendland. On asymptotic error analysis and underlying mathematical principles for boundary element methods. In C.A. Brebbia, editor, *Boundary Element Techniques in Computer Aided Engineering, NATO ASI Series E-84*, pages 417–436, Martinus Nijhoff Publ., Dordrecht-Boston-Lancaster, 1984.

Helmut Harbrecht, Institut für Numerische Simulation, Universität Bonn, Wegelerstr. 6, 53115 Bonn, Germany.

E-mail address: harbrecht@ins.uni-bonn.de

Maharavo Randrianarivony, Institut für Informatik und Praktische Mathematik, Christian–Albrechts–Universität zu Kiel, Olshausenstr. 40, 24098 Kiel, Germany.

E-mail address: mara@informatik.uni-kiel.de