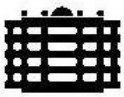
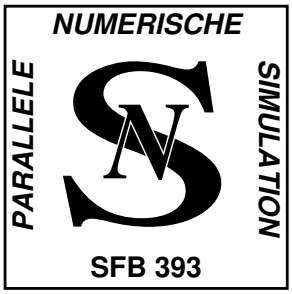


TECHNISCHE UNIVERSITÄT CHEMNITZ

Maharavo Randrianarivony Guido Brunnett
Reinhold Schneider

Constructing a diffeomorphism between a trimmed domain and the unit square

Preprint SFB393/03-20



Sonderforschungsbereich 393
Parallele Numerische Simulation für Physik und Kontinuumsmechanik

TECHNISCHE UNIVERSITÄT CHEMNITZ

Sonderforschungsbereich 393

Parallele Numerische Simulation für Physik und Kontinuumsmechanik

Maharavo Randrianarivony Guido Brunnett

Reinhold Schneider

Constructing a diffeomorphism between a trimmed domain and the unit square

Preprint SFB393/03-20

Abstract

This document has two objectives: decomposition of a given trimmed surface into several four-sided subregions and creation of a diffeomorphism from the unit square onto each subregion. We aim at having a diffeomorphism which is easy and fast to evaluate. Throughout this paper one of our objectives is to keep the shape of the curves delineating the boundaries of the trimmed surfaces unchanged. The approach that is used invokes the use of transfinite interpolations. We will describe an automatic manner to specify internal cubic Bézier-spline curves that are to be subsequently interpolated by a Gordon patch. Some theoretical criterion pertaining to the control points of the internal curves is proposed and proved so as to ensure that the resulting Gordon patch is a diffeomorphism. Numerical results are reported to illustrate the approaches. Our benchmarks include CAD objects which come directly from IGES files.

Preprintreihe des Chemnitzer SFB 393

ISSN 1619-7178 (Print) ISSN 1619-7186 (Internet)

SFB393/03-20 December 2003

Contents

Author's addresses:

Maharavo Randrianarivony, Guido Brunnett
TU Chemnitz, Fakultät für Informatik
D-09107 Chemnitz
http://www.tu-chemnitz.de/~ranh
http://www.informatik.tu-chemnitz.de/~gdv

Reinhold Schneider
TU Chemnitz, Fakultät für Mathematik
D-09107 Chemnitz
http://www-user.tu-chemnitz.de/~reinhold/

1 Introduction

We address the problem of design with trimmed surfaces ([17, 26]) which are fundamental entities in CAGD. The majority of CAD objects, even among the simplest ones such as closed cylinders, are partly or completely composed of trimmed surfaces. Therefore completely ignoring treatment of trimmed surfaces is a major inacceptable restriction if one wants to deal with real-world CAD data.

In this document we want to consider the problem of decomposing the surface of a CAD object into subregions that are images of parametric functions from the unit square. For untrimmed surfaces , such as Bézier, polynomial and rational B-splines ([9]), that objective generally does not induce any difficulty. For a trimmed surface we have to face two main problems. First we have to split the trimmed surface into four sided subregions. Then for each subregion, we are to find a parametric function $\mathbf{x}$, which is a diffeomorphism, from the unit square onto the subregion. The property that $\mathbf{x}$ should be a diffeomorphism is not required during the design process. It is only used in a subsequent application (which is not treated in this paper) of numerical solution of integral equations ([6, 7, 23, 15]) on CAD objects. We need also that the function $\mathbf{x}$ is easy to evaluate because it will have to be invoked repeatedly in future applications.

The general approach of obtaining a splitting into four-sided subregions is done in several stages. First one takes coarse approximations of the curves which bound the trimmed surface. Then one splits the resulting polygon $\mathbf{P}$ into convex sets. We will describe a way to have a convex splitting in which all boundary nodes reside on the boundaries of the initial trimmed surface. The boundaries of the polygon $\mathbf{P}$ might thus change during the course of the convex splitting. We will describe also an efficient way to make the number of nodes of each convex set even by adding internal extra nodes. Our approach tries to use very few extra nodes by using a weighted graph corresponding to the convex splitting and by utilizing shortest paths in the graph. A quadrangulation ([18, 20]) of the convex sets is the next stage. In this paper we will not describe any method for achieving the quadrangulation. Finally the quadrilateral cells are transformed into four-sided regions by replacing the straight edges which are next to the boundaries by the boundaries of the initial trimmed surface.

The major technique, upon which our generation of parametrization from the unit square onto a four-sided domain is based, is the use of transfinite interpolations ([11, 12, 9]). In most practical cases using the Coons interpolation already gives a diffeomorphism. But sometimes Coons interpolation gives an undesired overspill phenomenon. We recourse to the idea of Gordon in order to find a remedy for the overspill phenomenon. We will describe a way to automatically find the internal curves which have to be interpolated by a Gordon patch.

In the next section we will describe the problem to be solved more specifically. An outline of the method for tessellating a trimmed surface will be seen in the third section. In the following two sections, we will treat in detail the shifted convex splitting approach and the converting of odd convex sets into even ones. After recalling the main idea of transfinite interpolation using Coons and Gordon patches in section six, we will describe in detail a

way of automatic elimination of the overspill phenomenon in section seven. A theoretical result which serves as recognizing whether the resulting Gordon patch is diffeomorphism will be provided and proved. Further we will give some benchmarks which result directly from CAD data stored in IGES files ([26]). Runtimes about tessellation, parameterization and evaluations of the resulting mapping will also be reported.

2 Problem setting

Definition 1 A composite curve $\mathcal{C}$ with N constituents is a set of planar parametric curves $\mathcal{C}_i$ defined on $[a_i, b_i] \subset \mathbf{R}$ $(i = 1, ..., N)$ such that

$$\mathcal{C}_i(b_i) = \mathcal{C}_{i+1}(a_{i+1}) \quad \text{for } i = 1, ..., N-1\,.$$

By defining

$$\begin{cases} t_0 & := & 0 \\ t_k & := & t_{k-1} + (b_k - a_k) \qquad k = 1, ..., N, \end{cases}$$

the curve $\mathcal{C}$ is defined on $[t_0, t_N]$ by

$$\mathcal{C}(t) := \mathcal{C}_i(t - t_i) \qquad \text{if} \quad t \in [t_i, t_{i+1}]. \tag{1}$$

Remark 1 Throughout this document, we consider only closed composite curves which do not have any double point, that is, there do not exist two different parameters $a, b \in]t_0, t_N[$ such that $\mathcal{C}(a) = \mathcal{C}(b)$. In practice composite curves are composed of line segments, circular arcs, polynomial or rational B-splines.

Definition 2 A trimmed surface S is a multiply connected region in $\mathbf{R}^2$ such that each boundary curve is the image of a closed composite curve.

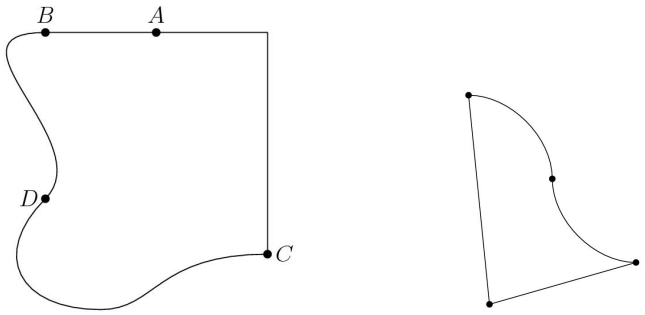
For a given trimmed surface S, the objective of this paper is twofold

1. Tessellate S into m four-sided domains F_i

$$S = \bigcup_{i=1}^{m} F_i\,. \tag{2}$$

2. For each four sided subregion F_i, $i = 1, ..., m$, generate a diffeomorphism $\mathbf{x}_i$ which maps the unit square $[0, 1]^2$ onto F_i.

Remark 2 In the first problem, we try to keep the number m of the four-sided domains small. The second problem is a very well known one in the context of complex analysis (see [16] and reference there). A common approach is to approximate the boundaries of S_i by a polygon and generate a conformal mapping from the unit square to the polygon. That method is numerically supported by the method of Schwarz-Christofel ([25, 16]) where



(a) (b)

Figure 1: (a) B and D are G^1 vertices. A and C are not (b) a four sided domain having a G^1 vertex

the mapping is given in the form of an integral requiring numerical evaluation. Some coefficients in the integral should be determined in which one has to solve a nonlinear system of equations.
Some generalization of the Schwarz-Christofel approach by approximating the boundary with piecewise polynomials is described in [4, 5]. The following two points are of great importance in our objective

- Retaining the exact geometry of the trimmed surface S.
- Easy and fast evaluation of the resulting mappings $\mathbf{x}_i$ $i = 1, ..., m$.

We will achieve those two objectives by following the idea of transfinite interpolation described by Gordon and Hall in [12] and by giving an efficient approach of finding the curves to be inserted.

3 Outline for multiply connected regions

In this section, we would like to outline our way of splitting a multiply connected trimmed surface S into several four-sided domains F_i (see relation (2)). Before doing that, let us see some few interesting properties.

Definition 3 Consider a parametric curve $\mathcal{C}(t)$ which is supposed to be a composite curve. A point $\mathbf{v} = \mathcal{C}(t_0)$ on the curve is called a G^1 vertex if we have G^1-geometric continuity (see [17, 9]) there and $\mathbf{v}$ does not reside strictly within a line segment. In Fig. 1(a) B and D are G^1 vertices. A and C are not.

Remark 3 We do not want to have a four sided subregion F in which one (or more) of its four vertices is a G^1 vertex (as in Fig. 1(b)). That is due to the fact that we cannot find any diffeomorphism $\mathbf{x}$ from the unit square $D = [0, 1] \times [0, 1]$ to such an F such that $\mathbf{x}$ transforms the four vertices of D to those of F.

We want to outline now our general approach in which we will denote the boundary of S by $\mathbf{B}_{\text{out}}$, and $\mathbf{B}_{\text{in}}^i$ $i = 0, ..., N-1$ where N is the number of internal boundary curves. The algorithm has four steps:
Step 1: Take a coarse polygonal approximations $\mathbf{P}_{\text{out}}$, $\mathbf{P}_{\text{in}}^i$ of the boundaries of S and call $\mathbf{P}$ the resulting multiply connected polygon.
Step 2: Take a "shifted" convex partitioning of $\mathbf{P}$. We will show a method for generating the convex partition in which all boundary nodes lie on the boundaries $\mathbf{B}_{\text{out}}$, $\mathbf{B}_{\text{in}}^i$ of the initial trimmed surface S. In other words, we avoid the idea of splitting the polygon $\mathbf{P}$ and then shifting any newly generated boundary nodes of $\mathbf{P}$ to the boundaries of S. Our method will demonstrate to be efficient when there are G^1 vertices.
Step 3: Quadrangulation of the resulting convex splitting. After completing this step we have m four-sided domains Q_i whose edges are still line segments.
Step 4: We generate the four-sided subregions F_i from the quadrilateral Q_i in the following way. For each quadrilateral Q_i, we test if two consecutive nodes A and B of Q_i belong to the same boundary curve of S. That is $A = \mathbf{B}_{\text{out}}(t_A)$ and $B = \mathbf{B}_{\text{out}}(t_B)$ (resp. $A = \mathbf{B}_{\text{in}}^j(t_A)$ and $B = \mathbf{B}_{\text{in}}^j(t_B)$ for some j). In that case, we replace the straight edge $[A, B]$ of Q_i by the curve portion of $\mathbf{B}_{\text{out}}$ (resp. $\mathbf{B}_{\text{in}}^j$) between the parameter values t_A and t_B.

Remark 4 There are already a lot of papers treating the quadrangulation in step 3. See for example [8, 3, 18, 20] and the reference there. So we will not give any detail about quadrangulation. In our implementation we have used the quadrangulation in [18] for three reasons. First, it does not use additional boundary nodes. Second, it keeps the number of quadrilaterals small. Finally, it is comparatively easier to implement it than other methods.

In the next sections, we will give details about how to efficiently treat step 2. We will give an accurate description of how to make the number of vertices of each convex subregion even. That is an essential preliminary step before quadrangulation.

4 Shifted convex splitting

Let us describe a way of tessellating a polygon $\mathbf{P}$ into convex sets R_i. For now we assume that $\mathbf{P}$ is simply connected. We will present a way of decomposing a multiply connected polygon into simply connected ones at the end of this section. The main idea behind having a shifted convex partitioning is that instead of inserting a new boundary node on the simply connected polygon $\mathbf{P}$, we choose a node which lies exactly on the curved boundaries $\mathbf{B}_{\text{out}}$, $\mathbf{B}_{\text{in}}^j$ of the initial trimmed surface. Of course, the polygon $\mathbf{P}$ will undergo a modification in the course of this process. The objective is to join a G^1 vertex or a reflex vertex (i.e. a vertex where the internal angle is strictly greater than π) to another suitably chosen point of $\mathbf{P}$ or to another point on the boundaries of the initial trimmed surface S. We will follow a similar idea as Joe ([19]) who has treated domains with line segments as boundaries. The main difference is twofold. First, we remove also G^1 vertices. Second, we allow the corresponding points to lie on the curved boundaries of S. We want to split $\mathbf{P}$

into $\cup R_i$ recursively. The convex splitting is complete when there is no G^1 or reflex vertex left in the splitting. The algorithm is as follows. As initialization we have $R_1 := P$ and the number of subregions is $p = 1$. Suppose now that we have p subregions R_i $i = 1, ..., p$. Test if R_i contains a G^1 or a reflex vertex.
In the positive case, we split R_i into two subregions S_1, S_2. We overwrite R_i by S_1 and we append S_2 to the end of the subregion list. We have certainly to increment the number of subregions p. Note that a G^1 vertex could be a reflex one. In this case we consider it as a reflex vertex and we consider only G^1 vertices those which cannot be treated as reflex ones. In the next discussion. we will show how to split the subregion R_i into S_1 and S_2.

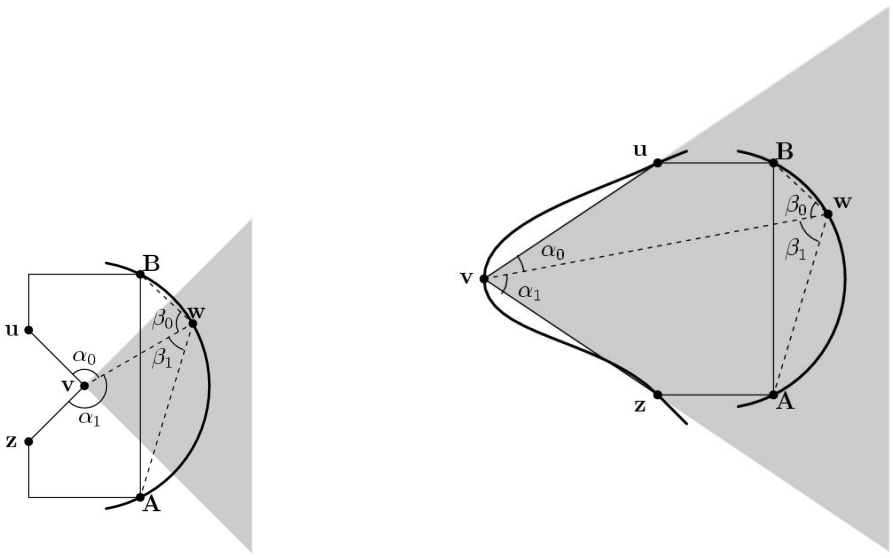
Remark 5 Throughout the whole algorithm, we assign flags to vertices. A vertex $\mathbf{v}$ has flag -1 if it lies on the exterior boundary curve $\mathbf{B}_{\text{out}}$. It has flag $j \geq 0$ if it lies on the j-th interior curve $\mathbf{B}^j_{\text{in}}$. Otherwise its flag is -2. Further, for a vertex $\mathbf{v}$ having flag -1 or $j \geq 0$, we consider $\mathbf{v}$ as a temporal entity, that is, we specify the time t at which it resides on the curve $\mathbf{v} = \mathbf{B}_{\text{out}}(t)$ or $\mathbf{v} = \mathbf{B}^j_{\text{in}}(t)$.

Suppose now that R_i has a reflex vertex or a G^1 vertex $\mathbf{v}$. We want to split R_i by inserting a separating edge from $\mathbf{v}$ to another vertex $\mathbf{w}$ which we want to specify in the next description.
Suppose the vertices of R_i are given in counter-clockwise orientation. Let $\mathbf{u}$ (resp. $\mathbf{z}$) be the vertex which precedes (resp. follows) $\mathbf{v}$. Define F to be the region in the plane which is in the intersection of the half-planes $(\mathbf{uv})^-$ and $(\mathbf{zv})^+$. In Fig. 2, F is graphically shown by the shaded areas. The desired point $\mathbf{w}$ should lie in F and it is determined as follows. We traverse the edges of R_i. For each edge e, let A and B be its endpoints. We distinguish two main cases for the edge $e = [A, B]$.
Case 1: A, B have the same flag which is not equal to -2 (see Fig. 2(a)). That is to say, they both reside on the same curve $\mathbf{D} := \mathbf{B}_{\text{out}}$ or $\mathbf{D} := \mathbf{B}^j_{\text{in}}$ for some $j = 0, ..., N - 1$.
Case 2: A, B have different flags or they have the same flag but it is -2 (see Fig. 2(b)).
If case 1 applies, suppose $A = \mathbf{D}(t_A)$ and $B = \mathbf{D}(t_B)$. We search for $\mathbf{w} = \mathbf{D}(t)$ in $F \cap \mathbf{D}([t_A, t_B])$ which is visible from $\mathbf{v}$ and which maximizes the expression

$$\gamma_e := \min\{\alpha_0, \alpha_1, \beta_0, \beta_1\}. \tag{3}$$

The new node $\mathbf{w}$ will then have the same flag as A and B.
If case 2 applies, we do the same thing but $\mathbf{w}$ should now lie on $F \cap [A, B]$. We repeat the same thing to all edges e and choose $\mathbf{w}$ which maximizes γ_e.

Remark 6 In order to decompose a multiply connected polygon $\mathbf{P}$ into simply connected ones, one can use similar idea as [19] in which the author inserts a line segment between the top node N_t of each internal curve and some point $\mathbf{w} \in \mathbf{P}$ lying above N_t and another line segment between the bottom node N_b and another node $\mathbf{z} \in \mathbf{P}$ below N_b. Contrarily to [19], we allow the points $\mathbf{w}$ and $\mathbf{z}$ not to belong to the polygon $\mathbf{P}$, that is, they could be placed on the boundary of the initial trimmed surface S following the same spirit as we have done in the shifted convex splitting.



(a) (b)

Figure 2: (a) Finding a corresponding point to a reflex vertex (b) Finding a corresponding point to a G^1 vertex

5 Even convex

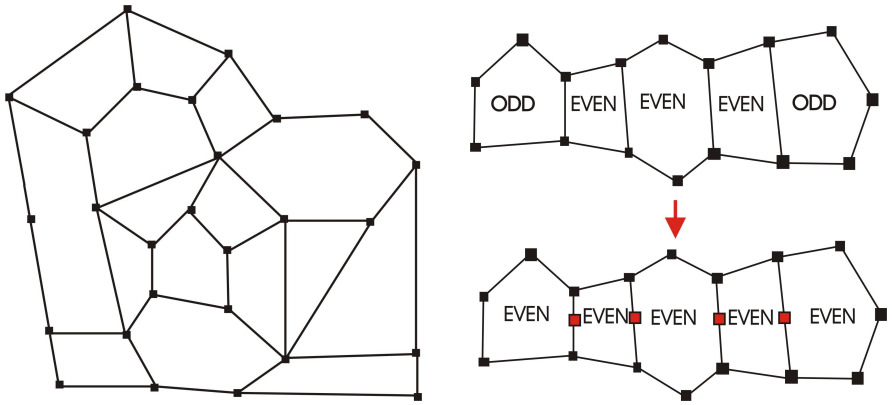
Suppose that we have a polygon **P** which has possibly some polygonal holes. We make the assumption that the number of its vertices n is even. At this point **P** is supposed to have been split into several convex subregions R_i (Fig. 3(a))

$$\mathbf{P} = \bigcup_{i=1}^{p} R_i.$$

We would like to immediately note that the number of vertices of the subregions R_i is not necessarily equal to n. That is because non-boundary nodes may be generated in the course of the former convex splitting.
Although we do not give any quadrangulation method in this paper, we would like to note the following characterization.

Theorem 1 (see [18, 3]) A convex set can be split into quadrilaterals if and only if the number of its vertices is even.

A prerequisite of being able to quadrilate is therefore to have convex sets R_i which all have even number of boundary nodes. The problem that we want to solve in this section is how to make the number of points of each convex set R_i even. At first sight, this problem is apparently very easy to solve. Considering the fact that we do not want to insert any new vertices in the boundary of **P** and that there might exist convex sets R_i which have no contact at all with the boundary of **P** (see Fig. 3(a)) and that we do not want to introduce



(a) (b)

Figure 3: (a)A convex splitting with some internal odd convex sets (b)Process of making two odd convex sets even

many new vertices, solving the former problem efficiently is not straightforward. Before describing our approach for solving such a problem, let us see the following facts

Definition 4 A polygon is called even (resp. odd), if the number of its vertices is even (resp. odd).

Remark 7 Suppose that we have two odd convex sets that are separated by s convex regions which are all even (see Fig. 3(b)). Adding one extra vertex between every two adjacent sets will make the parities of the boundary convex sets even. The parities of the intermediate convex sets are kept unchanged (i.e. even). That is due to the simple fact that adding a vertex in a convex set will invert its parity.

Theorem 2 Let $\mathbf{P}$ be a polygonal region which may have some holes (that are also polygonal). Suppose that the number n of nodes of $\mathbf{P}$ is even. Let R_i $i = 1, ..., p$ be a convex splitting of $\mathbf{P}$

$$\mathbf{P} = \bigcup_{i=1}^{p} R_i$$

Let r be the number of odd convex sets among R_i. Then r must be an even number.

Proof
Let us prove it by contradiction. Suppose that r were odd. Without loss of generality we can assume that $r = 1$ (if $r = 2k + 1$ with k nonzero then apply remark 7 to the first $2k$ odd convex sets). We also assume that R_p is the only convex set which is odd. Consider E the union of the first $(p-1)$ convex sets and let us compute the parity of e which is the number of boundary edges of E. Since the number of edges of each R_i $i = 1, ..., p-1$ is even and internal edges of E cancel in pair, e must be even. We want to consider now the

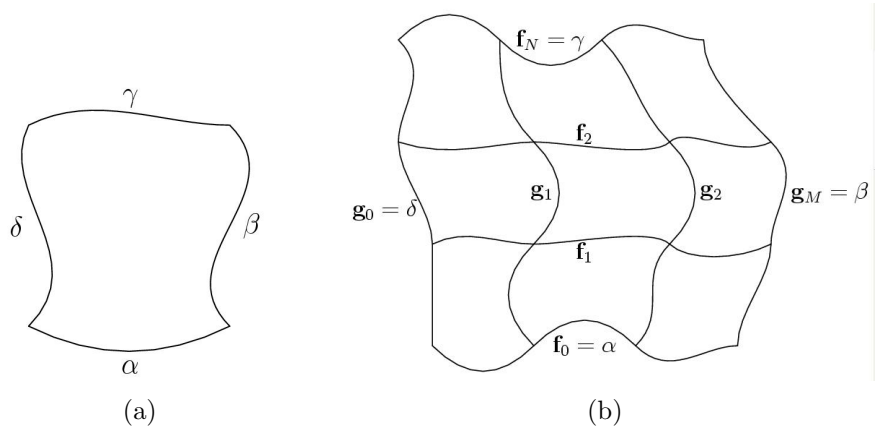
union $E \cup R_p$. Let m denote the number of common edges to E and R_p and c the number of edges of R_p. The number of boundary edges of $E \cup R_p$ is therefore

$$e + c - 2m = \text{even} + \text{odd} - \text{even} = \text{odd}, \tag{4}$$

which is in contradiction to the fact that the number of edges of $\mathbf{P}$ is even, $E \cup R_p$ being $\mathbf{P}$ itself.
Having all these properties in mind, we want to describe now our approach of making all convex sets even. The fundamental idea consists in joining the odd convex sets two by two by using the process described in remark 7. Because we do not want to introduce many extra points, we introduce the following definition.

Definition 5 To a set of convex regions R_i $i = 1, ..., p$, we associate a weighted graph $\mathcal{G}$. Each node N_i of the graph corresponds to a convex region R_i and two nodes N_i and N_j are linked if R_i and R_j are adjacent. Each node in the graph will be assigned a flag ODD or EVEN depending on the parity of the corresponding convex region. Each edge E_i in the graph is also assigned a flag KEEP or INSERT. Such a graph will be called the convex graph of R_i.

Algorithm:
We use the following algorithm to transform the parity of the odd polygons.
Step 0: Initialization: Set all flags of the edges in the convex graph $\mathcal{G}$ to KEEP and the weights to unity.
Step 1: Find the closest pair of ODD nodes N_i and N_j in the convex graph and let P be the closest path in the convex graph which joins those two nodes.
Step 2: For each edge E_i on the path P, invert its flag.
Step 3: Edges E_i having flag KEEP are assigned a weight 1.0 and those having flag INSERT a weight $\varepsilon << 1.0$. Change the flags of the nodes N_i and N_j to EVEN and go to step 1 if there are still ODD nodes in the convex graph.
After running this algorithm on the convex graph, we want to resume to the convex splitting. By considering all adjacent convex sets R_i and R_j, if the flag of the edge between the nodes N_i and N_j is INSERT in the convex graph, we insert one node in one common edge of the convex sets R_i and R_j.
Searching for the closest nodes in the graph and the corresponding path can be efficiently done with the help of the Dijkstra algorithm (see among others [2, 21]) or a similar one. Throughout our description, lengths of paths depends on the weights. The shortest path is the one which has the smallest weight. At the beginning where all weights are unity, shortest path means having the fewest number of edges. We assign very small weights ε to INSERT edges so that paths including these edges are short. That gives thus the likelihood for those INSERT edges to be inverted into KEEP edges. Recall that we want as few INSERT edges as possible because the number of INSERT edges is the same as the number of extra points to be inserted.



(a) (b)

Figure 4: (a)A four sided domain for Coons patch, (b)A network of curves for Gordon patch

6 Transfinite interpolation

Since our method of generating a mapping from the unit square to a four sided domain F is based on transfinite interpolation, let us introduce briefly some definitions and interesting properties. We will only state things which are immediately related to our problems. We direct the readers to [11, 12, 13, 14, 24] for more complete information.

6.1 Coons patch

Let us consider four continuously differentiable parametric curves (See Fig. 4(a)) α, β, γ, δ defined on the interval $[0, 1]$. They are supposed to fulfill the compatibility condition:

$$\alpha(0) = \delta(0)\,, \quad \alpha(1) = \beta(0)\,, \quad \gamma(0) = \delta(1)\,, \quad \gamma(1) = \beta(1). \tag{5}$$

We are interested in generating a parametric surface $\mathbf{x}(u, v)$ defined on the unit square $[0, 1]^2$ such that the boundary of the image of $\mathbf{x}$ coincides with the given four curves:

$$\begin{array}{rcl} \mathbf{x}(u,0) & = & \alpha(u) \quad \mathbf{x}(u,1) = \gamma(u) \quad \forall\, u \in [0,1] \\ \mathbf{x}(0,v) & = & \delta(v) \quad \mathbf{x}(1,v) = \beta(v) \quad \forall\, v \in [0,1]\,. \end{array} \tag{6}$$

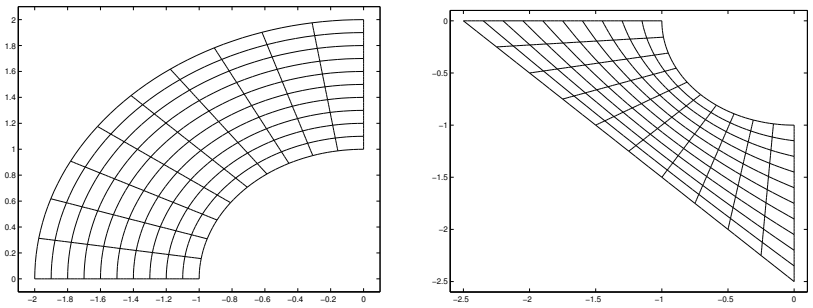
Such a mapping is usually referred to as transfinite interpolation which can be created by means of Coons patch. Its construction is done first by considering two functions ϕ and η

$$\phi(0) = \eta(0) = 0 \qquad \phi(1) = \eta(1) = 1 \tag{7}$$

and then by introducing two operators $\mathcal{P}$ and $\mathcal{Q}$:

$$(\mathcal{P}\mathbf{x})(u,v) \;:=\; (1 - \phi(v))\mathbf{x}(u,0) + \phi(v)\mathbf{x}(u,1) \tag{8}$$

$$(\mathcal{Q}\mathbf{x})(u,v) \;:=\; (1 - \eta(u))\mathbf{x}(0,v) + \eta(u)\mathbf{x}(1,v). \tag{9}$$



(a) (b)

Figure 5: Diffeomorph Coons patches

These operators are known completely if one knows the boundary values in (6). One verifies easily that the following Boolean sum satisfies the boundary conditions (6):

$$\mathcal{P} \oplus \mathcal{Q} := \mathcal{P} + \mathcal{Q} - \mathcal{P}\mathcal{Q}. \tag{10}$$

The functions ϕ, η which are better known as blending functions can be chosen in several ways (see [9, 11, 17, 24]). The simplest case is to take the bilinear blending function:

$$\phi(t) = \eta(t) = t\,. \tag{11}$$

We suppose in the sequel that the blending functions are sufficiently smooth. According to (10) we can express the solution to (6) in matrix form as:

$$\begin{aligned} \mathbf{x}(u,v) \;=\; & \begin{bmatrix} 1-\eta(u) & \eta(u) \end{bmatrix} \begin{bmatrix} \delta(v) \\ \beta(v) \end{bmatrix} + \\ & \begin{bmatrix} \alpha(u) & \gamma(u) \end{bmatrix} \begin{bmatrix} 1-\phi(v) \\ \phi(v) \end{bmatrix} - \\ & \begin{bmatrix} 1-\eta(u) & \eta(u) \end{bmatrix} \begin{bmatrix} \alpha(0) & \gamma(0) \\ \alpha(1) & \gamma(1) \end{bmatrix} \begin{bmatrix} 1-\phi(v) \\ \phi(v) \end{bmatrix}. \end{aligned} \tag{12}$$

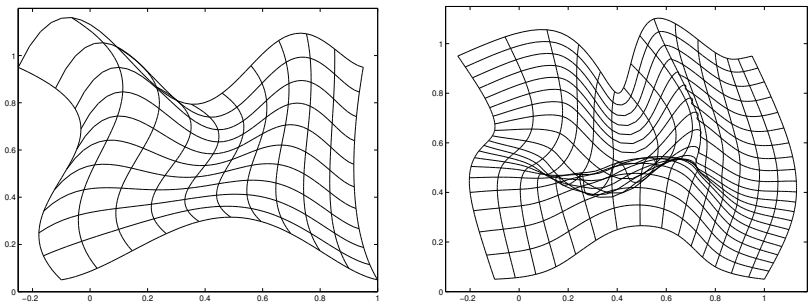
By using (6), we can still reduce it in a more compact way:

$$\mathbf{x}(u,v) = -\begin{bmatrix} -1 \\ 1-\eta(u) \\ \eta(u) \end{bmatrix}^T \begin{bmatrix} \mathbf{0} & \mathbf{x}(u,0) & \mathbf{x}(u,1) \\ \mathbf{x}(0,v) & \mathbf{x}(0,0) & \mathbf{x}(0,1) \\ \mathbf{x}(1,v) & \mathbf{x}(1,0) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} -1 \\ 1-\phi(v) \\ \phi(v) \end{bmatrix} \tag{13}$$

which is more suitable for implementation purpose.

6.2 Overspill phenomenon

The Coons patch $\mathbf{x}$ given by (12) interpolates the prescribed boundary curves. Our objective is not simply to have a mapping which interpolates the boundary but also to have



(a) (b)

Figure 6: Undesired overspill phenomena

a diffeomorphism. The boundary curves α, β, γ, δ being continuously differentiable, the differentiability of $\mathbf{x}$ is guaranteed if the blending functions are chosen to be differentiable as well. The real problem in having a diffeomorphism is then the invertibility and nonvanishing of the Jacobian. In most practical cases (like in Fig. 5) a Coons patch is already a diffeomorphism. Still, sometimes two unsatisfactory situations may occur. First, some isoline curves may reside partly outside the boundary like the case in Fig. 6(a). Another problem is that all isolines reside inside the domain but some of them overlap as in Fig. 6(b). These situations better known as "overspill phenomenon" are not suitable to our objective. Before describing some remedy let us review the idea of Gordon patch.

6.3 Gordon patch

Consider two sequences $\{u_0, ..., u_M\} \subset [0,1]$ and $\{v_0, ..., v_N\} \subset [0,1]$. In our case we have $u_0 = v_0 = 0$ and $u_M = v_N = 1$. Suppose we have two families of curves $\mathbf{f}_j$, $\mathbf{g}_i$, $j = 0, ..., N$, $i = 0, ..., M$ (see Fig. 4(b)) satisfying the compatibility condition:

$$\mathbf{x}_{ij} := \mathbf{g}_i(v_j) = \mathbf{f}_j(u_i) \quad (i,j) \in \{0, ..., M\} \times \{0, ..., N\}. \tag{14}$$

A Gordon patch ([12]) is a parametric surface $\mathbf{x}$ which is defined on $[0,1] \times [0,1]$ and which interpolates all the curves $\mathbf{f}_j$ and $\mathbf{g}_i$:

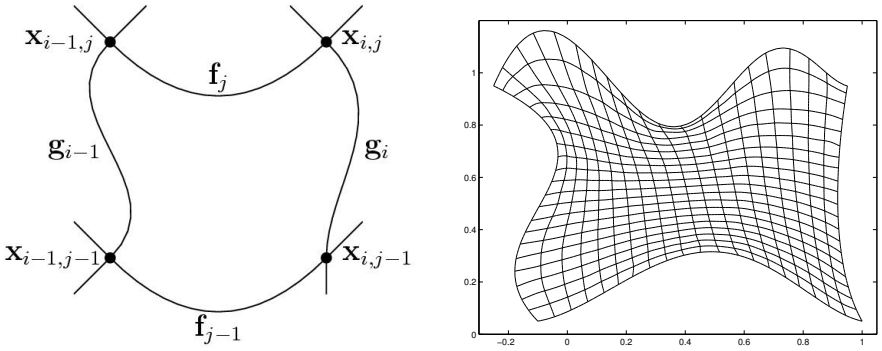
$$\begin{array}{rcll} \mathbf{x}(u, v_j) & = & \mathbf{f}_j(u) & \forall\, j = 0, ..., N \qquad \forall u \in [0,1], \\ \mathbf{x}(u_i, v) & = & \mathbf{g}_i(v) & \forall\, i = 0, ..., M \qquad \forall v \in [0,1]. \end{array} \tag{15}$$

By choosing two sets of functions $\varphi_i(u)$ and $\psi_j(v)$ $i = 0, ..., M$ and $j = 0, ..., N$ satisfying:

$$\varphi_i(u_k) = \delta_{ik} \qquad \psi_j(v_l) = \delta_{jl}, \tag{16}$$

we define the Gordon patch ([14]) as:

$$\mathbf{x}(u,v) := \sum_{i=0}^{M} \mathbf{g}_i(v)\varphi_i(u) + \sum_{j=0}^{N} \mathbf{f}_j(u)\psi_j(v) - \sum_{i=0}^{M}\sum_{j=0}^{N} \mathbf{x}_{ij}\varphi_i(u)\psi_j(v). \tag{17}$$



(a) (b)

Figure 7: (a)Local structure of Gordon patch, (b)Final Gordon patch of Fig. 6(a) without overspill

7 Generation of internal curves

We eliminate the overspill phenomena by first inserting some curves (see [12]) $\mathbf{f}_j$, $j = 1, ..., N-1$ and $\mathbf{g}_i$, $i = 1, ..., M-1$ inside the four-sided domain F and then by considering the Gordon patch $\mathbf{x}$ which interpolates those curves. Since we only know the equations of the curves delineating the boundary of F, we need an automatic method of generating the internal curves. Our approach consists of three stages:

- Find suitable points $\mathbf{x}_{ij}$ in the four-sided domain F. These points will be the future intersection of the curves $\mathbf{f}_j$ and $\mathbf{g}_i$ (see relation (14)).
- Determine the parameter values $u_0, ..., u_M$ and $v_0, ..., v_N$ (see relation (15)).
- Interpolate the points $\mathbf{x}_{ij}$ with cubic Bézier-splines to obtain $\mathbf{f}_j$ and $\mathbf{g}_i$:

$$\mathbf{g}_i(v) = \sum_{k=0}^{3} \mathbf{g}_{ik} B_k^3((v - v_{j-1})/(v_j - v_{j-1})) \quad \forall\, v \in [v_{j-1}, v_j]\,, \tag{18}$$

$$\mathbf{f}_j(u) = \sum_{l=0}^{3} \mathbf{f}_{jl} B_l^3((u - u_{i-1})/(u_i - u_{i-1})) \quad \forall\, u \in [u_{i-1}, u_i]\,. \tag{19}$$

We are going to give in the next section a rigorous method of recognizing if a Gordon patch is a diffeormorphism.
Determining the internal points $\mathbf{x}_{ij}$ is done in three stages. First one generates a mesh $\mathcal{M}$ on the four-sided domain. Then, one uses the shape preserving parameterization method described in [10] to find a mesh $\mathcal{T}$ on the unit square and a function κ such that κ transforms $\mathcal{T}$ to $\mathcal{M}$. Finally $\mathbf{x}_{ij}$ is defined to be $\kappa(i/M, j/N)$ for $i = 1, ..., M-1$ and $j = 1, ..., N-1$. Let the intersection of $\mathbf{g}_i$ and α (resp. γ) be $\alpha(\overline{u}_i)$ (resp. $\gamma(\tilde{u}_i)$) and that of $\mathbf{f}_j$ and δ (resp. β) be $\delta(\overline{v}_j)$ (resp. $\beta(\tilde{v}_j)$).

Define $u_i := 0.5(\overline{u}_i + \tilde{u}_i)$ and $v_j := 0.5(\overline{v}_j + \tilde{v}_j)$. In order that the compatibility condition (14) is fulfilled at the boundaries, one has to reparameterize α, β, γ, and δ. To that end, one searches for reparameterization functions s_α, s_β, s_γ, and s_δ which are monotonically increasing functions from $[0,1]$ to $[0,1]$ and which satisfy

$$s_\alpha(u_i) = \overline{u}_i\,, \quad s_\gamma(u_i) = \tilde{u}_i\,, \quad s_\delta(v_j) = \overline{v}_j\,, \quad s_\beta(v_j) = \tilde{v}_j. \tag{20}$$

One replaces afterward the boundary curves α, β, γ, and δ by:

$$\tilde{\alpha} := \alpha \circ s_\alpha\,, \quad \tilde{\beta} := \beta \circ s_\beta\,, \quad \tilde{\gamma} := \gamma \circ s_\gamma\,, \quad \tilde{\delta} := \delta \circ s_\delta. \tag{21}$$

In order to determine the unknown control points $\{\mathbf{g}_{ik}\}$, $\{\mathbf{f}_{jl}\}$ in (18) and (19), one solves some linear system which depends only on $\mathbf{x}_{ij}$, u_i and v_j and which is diagonal dominant (see [17] for details).

8 Termination test

In this section we would like to investigate a condition in terms of the control points from relations (18) and (19) so that the corresponding Gordon patch is a diffeomorphism. That could serve in practice as a termination test about how many curves $\mathbf{f}_j$ and $\mathbf{g}_i$ should be inserted. Let us see first some preliminary properties pertaining to Coons patch.

8.1 Preliminaries about Coons patch being diffeomorphism

Suppose that the boundary curves α, β, γ, δ for the Coons patch are Bézier curves of degree n and that their control points are α_i, β_i, γ_i, δ_i $i = 0, ..., n$ respectively. The blending functions are supposed also to be a polynomial:

$$\phi(t) = \eta(t) = \sum_{i=0}^{n} \phi_i B_i^n(t)\ . \tag{22}$$

We suppose that the degrees n are the same (otherwise one can apply degree elevation technique as in [9]).

Theorem 3 Consider the following three conditions:

(C1) There exists some $F > 0$ such that for $i = 0, ..., n$

$$\|(\beta_i - \delta_i) + \phi_i(\gamma_0 - \gamma_n + \alpha_n - \alpha_0) + (\alpha_0 - \alpha_n)\| \ \leq \ F/\rho \tag{23}$$
$$\|(\gamma_i - \alpha_i) + \phi_i(\alpha_n - \gamma_n + \gamma_0 - \alpha_0) + (\alpha_0 - \gamma_0)\| \ \leq \ F/\rho \tag{24}$$

where $\rho := \sup_{t\in[0,1]} |\phi'(t)|$.

(C2) There exists some $\kappa > 0$ such that for all ζ, u, χ, $v \in [0,1]$

$$\det(K_{u,\zeta}, L_{v,\chi}) \geq \kappa \qquad \text{where}$$

$$\begin{aligned} K_{u,\zeta} &:= (1-\zeta)\alpha'(u) + \zeta\gamma'(u)\,, && (25)\\ L_{v,\chi} &:= (1-\chi)\delta'(u) + \chi\beta'(u)\,. && (26) \end{aligned}$$

(C3) There is some M with

$$\|K_{u,\zeta}\| \leq M \quad \text{and} \quad \|L_{v,\chi}\| \leq M \qquad \forall \zeta, u, \chi, v \in [0,1]. \tag{27}$$

Under these conditions the Coons patch with respect to α, β, γ, δ is a diffeomorphism if

$$2MF + F^2 < \kappa. \tag{28}$$

Proof
Some few computations reveal that the partial derivatives of the Coons patch are

$$\mathbf{x}_u(u,v) = \phi'(u)S_u + C_u\,, \qquad \mathbf{x}_v(u,v) = \phi'(v)S_v + C_v \qquad \text{where} \tag{29}$$

$$\begin{aligned} S_u &:= \beta(v) - \delta(v) + [(1-\phi(v))\alpha(0) + \phi(v)\gamma(0)] - [(1-\phi(v))\alpha(1) + \phi(v)\gamma(1)]\,,\\ S_v &:= \gamma(u) - \alpha(u) + [(1-\phi(u))\alpha(0) + \phi(u)\alpha(1)] - [(1-\phi(u))\gamma(0) + \phi(u)\gamma(1)]\,,\\ C_u &:= (1-\phi(v))\alpha'(u) + \phi(v)\gamma'(u)\,,\\ C_v &:= (1-\phi(u))\delta'(v) + \phi(u)\beta'(v)\,. \end{aligned}$$

Therefore we obtain

$$\begin{aligned} S_u &= \sum_{i=0}^{n}(\beta_i - \delta_i)B_i^n(v) + \phi(v)(\gamma_0 - \gamma_n + \alpha_n - \alpha_0) + (\alpha_0 - \alpha_n)\,, && (30)\\ S_v &= \sum_{i=0}^{n}(\gamma_i - \alpha_i)B_i^n(u) + \phi(u)(\alpha_n - \gamma_n + \gamma_0 - \alpha_0) + (\alpha_0 - \gamma_0)\,. && (31) \end{aligned}$$

After a few rearrangements one obtains

$$\begin{aligned} S_u &= \sum_{i=0}^{n}[(\beta_i - \delta_i) + \phi_i(\gamma_0 - \gamma_n + \alpha_n - \alpha_0) + (\alpha_0 - \alpha_n)]B_i^n(v)\,, && (32)\\ S_v &= \sum_{i=0}^{n}[(\gamma_i - \alpha_i) + \phi_i(\alpha_n - \gamma_n + \gamma_0 - \alpha_0) + (\alpha_0 - \gamma_0)]B_i^n(u)\,. && (33) \end{aligned}$$

By using (C1) one obtains

$$|\phi'(u)|.\|S_u\| \leq F \quad \text{and} \quad |\phi'(v)|.\|S_v\| \leq F. \tag{34}$$

Because of multilinearity of the determinant function we have

$$\begin{aligned}
\det(\mathbf{x}_u, \mathbf{x}_v) &= \phi'(u)\phi'(v)\det(S_u, S_v) + \phi'(u)\det(S_u, C_v) + && (35)\\
&\quad +\phi'(v)\det(C_u, S_v) + \det(C_u, C_v). && (36)\\
&\geq \det(C_u, C_v) - \{|\phi'(u)\phi'(v)\det(S_u, S_v)| + && (37)\\
&\quad +|\phi'(u)\det(S_u, C_v)| + |\phi'(v)\det(C_u, S_v)|\} && (38)\\
&\geq \kappa - (F^2 + 2FM) > 0 \quad \text{due to (34) (C2) and (C3).} && (39)
\end{aligned}$$

That means the Jacobian is nowhere zero. The inverse function theorem ensures therefore that the Coons patch is a diffeomorphism.

Remark 8 Condition (C2) has some geometric interpretation. If ones consider any convex combination K of the tangent vectors $\alpha'(u)$ and $\gamma'(u)$ and L of $\delta'(v)$ and $\beta'(v)$, then K and L are bounded away from being collinear and they are never zero in norm.

Remark 9 Condition (C3) can be replaced by $(\tilde{C}3)$: For $i = 0, ..., n-1$ and $j = 0, ..., n$

$$\begin{aligned}
n\|\phi_j(\gamma_{i+1} - \gamma_i + \alpha_i - \alpha_{i+1}) + (\alpha_{i+1} - \alpha_i)\| \leq M && (40)\\
n\|\phi_j(\beta_{i+1} - \beta_i + \delta_i - \delta_{i+1}) + (\delta_{i+1} - \delta_i)\| \leq M. && (41)
\end{aligned}$$

The idea of the proof remains fundamentally unchanged. Condition $(\tilde{C}3)$ is easier to check than $(C3)$ because $(\tilde{C}3)$ involves only discrete information. Effectively $(\tilde{C}3)$ implies the following bounds

$$\begin{aligned}
\|(1 - \phi(v))\alpha'(u) + \phi(v)\gamma'(u)\| &\leq M\,, && (42)\\
\|(1 - \phi(u))\delta'(v) + \phi(u)\beta'(v)\| &\leq M\,. && (43)
\end{aligned}$$

8.2 Diffeomorph Gordon patch

The usual way ([9]) of generating the function φ_i, $i = 0, ..., M$ and ψ_j, $j = 0, ..., N$ of relation (16) is with the help of the Lagrange polynomials. The support of such functions is therefore the whole interval $[0, 1]$. That method and similar ones are called *elastic* in [14] because a perturbation of one curve $\mathbf{f}_j$ or $\mathbf{g}_i$ propagates to the whole domain. For our method, we prefer to use a local method in which φ_i and ψ_j are defined by

$$\varphi_i(u) := \begin{cases} \phi[(u - u_{i-1})/(u_i - u_{i-1})] & \text{if} \quad u \in [u_{i-1}, u_i] \\ 1 - \phi[(u - u_{i-1})/(u_i - u_{i-1})] & \text{if} \quad u \in [u_i, u_{i+1}] \\ 0 & \text{otherwise,} \end{cases} \qquad (44)$$

$$\psi_j(v) := \begin{cases} \phi[(v - v_{j-1})/(v_i - v_{j-1})] & \text{if} \quad v \in [v_{j-1}, v_j] \\ 1 - \phi[(v - v_{j-1})/(v_j - v_{j-1})] & \text{if} \quad v \in [v_j, v_{j+1}] \\ 0 & \text{otherwise,} \end{cases} \qquad (45)$$

where ϕ is a function defined on $[0, 1]$ with $\phi(0) = 0$ and $\phi(1) = 1$.

The resulting Gordon patch is known to be *plastic* because every perturbation is kept local due to local support of the blending functions. We want to analyze the Gordon patch in the cell $R_{ij} := [u_{i-1}, u_i] \times [v_{j-1}, v_j]$ whose image is bounded by $\mathbf{f}_{j-1}$, $\mathbf{f}_j$, $\mathbf{g}_{i-1}$ and $\mathbf{g}_i$ (see Fig. 7(a)).

Corollary 1 Consider the following three conditions:

(G1) There exists some $F > 0$ such that for $k = 0, ..., 3$

$$\begin{aligned} \|(g_{ik} - g_{i-1,k}) + \phi_k(\mathbf{x}_{i-1,j} - \mathbf{x}_{ij} + \mathbf{x}_{i,j-1} - \mathbf{x}_{i-1,j-1}) + (\mathbf{x}_{i-1,j-1} - \mathbf{x}_{i,j-1})\| &\leq F/\rho\,, \\ \|(f_{jk} - f_{j-1,k}) + \phi_k(\mathbf{x}_{i,j-1} - \mathbf{x}_{ij} + \mathbf{x}_{i-1,j} - \mathbf{x}_{i-1,j-1}) + (\mathbf{x}_{i-1,j-1} - \mathbf{x}_{i-1,j})\| &\leq F/\rho\,, \end{aligned}$$

where $\rho := \sup_{t\in[0,1]} |\phi'(t)|$.

(G2) There exists some $\kappa > 0$ such that for all $u \in [u_{i-1}, u_i]$, $v \in [v_{j-1}, v_j]$, $\zeta, \chi \in [0, 1]$

$$\det(K_{u,\zeta}, L_{v,\chi}) \geq \kappa \qquad \text{where}$$

$$K_{u,\zeta} := (u_i - u_{i-1})[(1-\zeta)\mathbf{f}'_{j-1}(u) + \zeta\mathbf{f}'_j(u)]\,, \tag{46}$$

$$L_{v,\chi} := (v_j - v_{j-1})[(1-\chi)\mathbf{g}'_{i-1}(v) + \chi\mathbf{g}'_i(v)]\,. \tag{47}$$

(G3) There is some M such that for all $u \in [u_{i-1}, u_i]$, $v \in [v_{j-1}, v_j]$, $\zeta, \chi \in [0, 1]$

$$\|K_{u,\zeta}\| \leq M \quad \text{and} \quad \|L_{v,\chi}\| \leq M. \tag{48}$$

Under these conditions the Gordon patch is a diffeomorphism in R_{ij} if

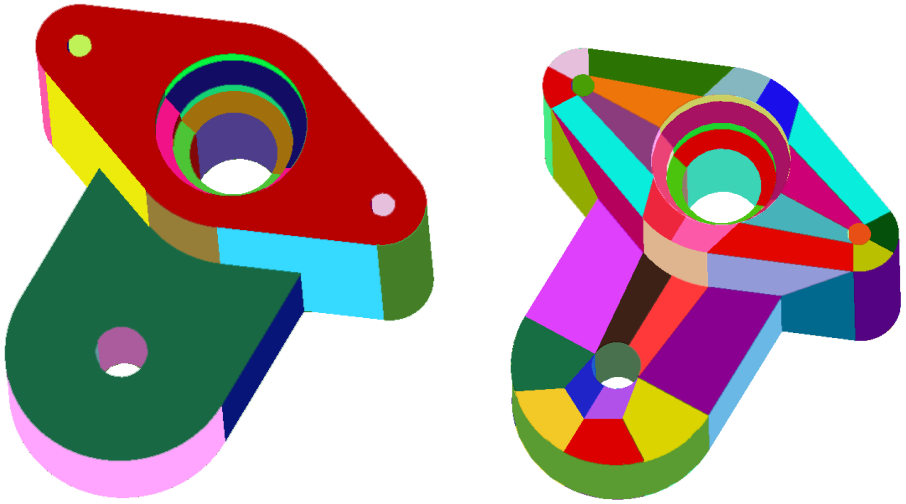
$$2MF + F^2 < \kappa. \tag{49}$$

Proof
One can check that in the domain $[u_{i-1}, u_i] \times [v_{j-1}, v_j]$ the Gordon patch, which has (44) and (45) as blending functions and which interpolates the internal curves in (18), (19), coincides with the Coons patch with respect to the boundary curves α, β, γ, δ defined for $u, v \in [0, 1]$ by:

$$\begin{aligned} \alpha(u) &:= \mathbf{f}_{j-1}(u(u_i - u_{i-1}) + u_{i-1})\,, & \beta(v) &:= \mathbf{g}_i(v(v_j - v_{j-1}) + v_{j-1})\,, \\ \gamma(u) &:= \mathbf{f}_j(u(u_i - u_{i-1}) + u_{i-1})\,, & \delta(v) &:= \mathbf{g}_{i-1}(v(v_j - v_{j-1}) + v_{j-1}). \end{aligned}$$

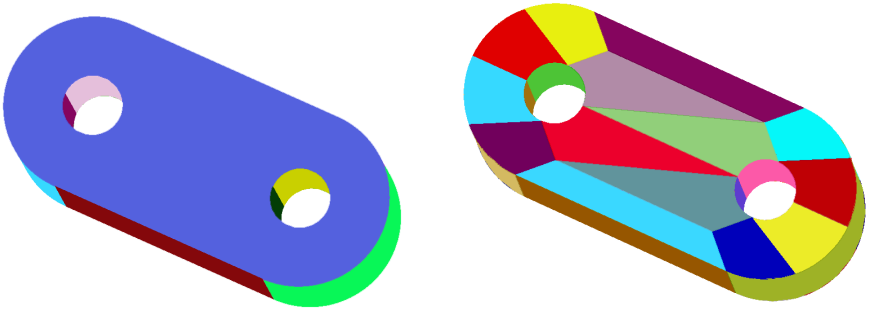
We need to apply the results in theorem 3 to complete the proof.

Remark 10 The objects which are used in the following numerical tests stem from IGES files ([26]). So far we have only treated *planar* trimmed surfaces. *Spatial* trimmed surfaces are provided in IGES format in the following way. An initial parametric surface $\mathbf{R}$ is given from a rectangular (which can supposed to be the unit square after scaling and shifting) parameter domain D to the space $\mathbf{R}^3$. Afterward, a planar trimmed surface S is defined inside D. The eventual spatial trimmed surface is then the image of S by $\mathbf{R}$. Splitting a spatial trimmed surface $\mathbf{R}$ into four-sided domains can therefore be done by splitting the planar trimmed surface S and then by taking the image by $\mathbf{R}$. In general the base surface $\mathbf{R}$ is supposed to be already a diffeomorphism. Hence, the final diffeomorphism in case of spatial trimmed surface is the composition of a Gordon patch and the base surface $\mathbf{R}$.



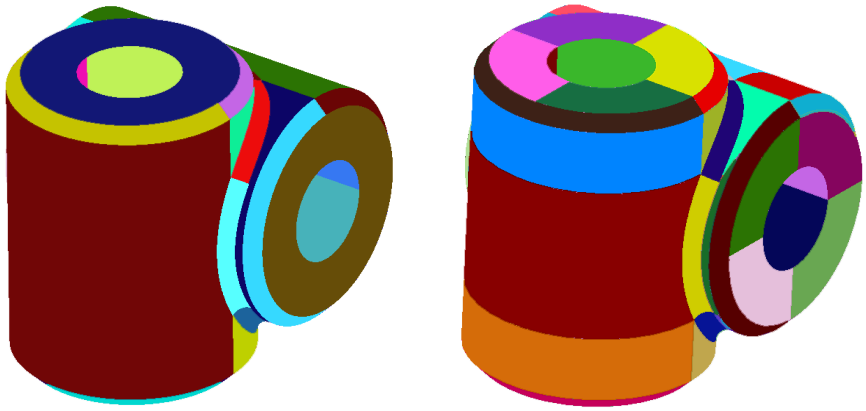
(a) (b)

Figure 8: (a)Object with 5 trimmed surfaces and 25 untrimmed, (b)Object with 79 four-sided subregions



(a) (b)

Figure 9: (a)Object with 2 trimmed surfaces and 8 untrimmed, (b)Object with 36 four-sided subregions



(a) (b)

Figure 10: (a)Object with 10 trimmed surfaces and 20 untrimmed, (b)Object with 54 four-sided subregions

9 Numerical tests

In this section, we want to present some practical samples obtained from the formerly presented methods. We want particularly to present the runtimes of the approaches. The following tests have been done with an Intel Pentium 4 processor 2.66 GHz running Windows XP. First we will present numerical results pertaining to the tessellation. Then we will give some for the parameterization.
We consider three objects which have 30, 10 and 30 patches respectively (see Fig. 8(a), 9(a), 10(a)). We have used the afore mentioned methods to tessellate them into four-sided subregions. In Fig. 8(b), 9(b), 10(b) we see the final tessellations. After the tessellations, the objects have respectively 79, 36 and 54 four-sided subregions.

	Information extraction	Tessellation
Object 1	0.301 sec	2.073 sec
Object 2	0.291 sec	0.801 sec
Object 3	0.381 sec	1.512 sec

Table 1: Runtimes for tessellations (CPU)

In Table 1 we gather the time needed to perform the tessellation. The information extraction includes automatic loading of the corresponding IGES files and establishment of the information pertaining to the trimmed as well as untrimmed surfaces.
As for the second test, we want to generate a Gordon patch for the domain in Fig. 6(a). In Table 2, we find the time needed to generate the gridpoints $\mathbf{x}_{ij}$, and the time for finding the internal curves $\{\mathbf{f}_j\}$ and $\{\mathbf{g}_i\}$. The result of the Gordon patch is found in Fig. 7(b). One notes that determining the internal curves needs only to be done once and the values

Tasks	CPU time
Searching for the gridpoints	4.306 sec
Finding the internal curves	0.080 sec
Evaluation at 100 positions	0.090 sec

Table 2: Runtimes for parameterization

of the control points in equations (18) and (19) can be stored for any future purpose. We are also interested in the computational cost of evaluating the resulting Gordon patch. In the last row of Table 2, we report the time needed to evaluate the Gordon patch at 100 parameter values $(u_k, v_k) \in [0,1] \times [0,1]$.

References

[1] R. Barnhill, "*Computer aided surface representation and design*", In R. Barnhill and W. Boehm, editors, Surfaces in CAGD, 1-24. North-Holland, Amsterdam, 1986.

[2] H. Bast, K. Mehlhorn, G. Schfer and H. Tamaki "*A heuristic for Dijkstra's algorithm with many targets and its use in weighted matching algorithms*", Algorithmica 36, No.1, 75-88, 2003.

[3] P. Bose and G. Toussaint, "*Characterizing and efficiently computing quadrangulations of planar point sets*", Comput. Aided Geom. Des. 14, No.8, 763-785 1997.

[4] M. Chaudhry, "*An extended Schwarz-Christoffel transformation for numerical mapping of polygons with curved segments*",

COMPEL 11, No.2, 277-293 ,1992.

[5] M. Chaudhry, "*Numerical computation of the Schwarz-Christoffel transformation parameters for conformal mapping of arbitrarily shaped polygons with finite vertices*", COMPEL 11, No.2, 263-275, 1992.

[6] W. Dahmen, "*Wavelet and multiscale methods for operator equations*", Acta Numerica Vol. 6, 55-228, 1997.

[7] W. Dahmen and R. Schneider, "*Wavelets on manifolds. I: Construction and domain decomposition*", SIAM J. Math. Anal. 31, No.1, 184-230, 1999.

[8] H. Everett, W. Lenhart, M. Overmars, T. Shermer and J. Urrutia, "*Strictly convex quadrilateralizations of polygons*", In Proc. of the fourth Canadian conference on computational geometry, 77-82, St. Johns, Newfoundland, 1992.

[9] G. Farin "*Curves and surfaces for computer aided geometric design. A practical guide.*", 2nd ed., Academic Press, Boston, 1990.

[10] M. Floater and K. Hormann "*Parameterization of triangulations and unorganized points*", Iske, Armin (ed.) et al., Tutorials on multiresolution in geometric modelling. European summer school lecture notes, Munich Univ. of Technology, Germany, August 22-30, 2001.

[11] A. Forrest, "*On Coons and other methods for the representation of curved surfaces*", Computer Graphics and Image Processing 1, 1972.

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

[13] W. Gordon and C. Hall, "*Transfinite element methods: Blending-function interpolation over arbitrary curved element domains*", Numer. Math. 21, 109-129, 1973.

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

[15] H. Harbrecht and R. Schneider, "*Biorthogonal wavelet bases for the boundary element method*", Preprint SFB393/03-10, Technische Universität Chemnitz, Sonderforschungsbereich 393, 2003.

[16] P. Henrici, "*Applied and computational complex analysis*", vol. 3, John Wiley & sons, New York, 1986.

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

[18] B. Joe, "*Quadrilateral mesh generation in polygonal regions*", Comput.-Aided Des. 27, No.3, 209-222,1995.

[19] B. Joe and R. Simpson, "*Triangular Meshes for Regions of Complicated Shape*", Int. J. Numer. Methods Eng. 23, 751-778, 1986.

[20] S. Ramaswami, P. Ramos and G. Toussaint, "*Converting triangulations to quadrangulations*", Comput. Geom. 9, No.4, 257-276, 1998.

[21] S. Saunders and T. Takaoka "*Improved shortest path algorithms for nearly acyclic graphs*", Theor. Comput. Sci. 293, No.3, 535-556, 2003.

[22] R. Schinzinger and P. Laura, "*Conformal mapping: methods and applications*", Elsevier, Amsterdam, 1991.

[23] R. Schneider, "*Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur Lösung grosser vollbesetzter Gleichungssysteme*", B.G. Teubner, Stuttgart, 1998.

[24] G. Schulze, "*Blending-Function-Methoden im CAGD*", Diplomarbeit, Universität Dortmund, 1986.

[25] L. Trefethen, "*Numerical computation of the Schwarz-Christoffel transformation*", SIAM J. Sci. Stat. Comput. 1, 82-102, 1980.

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

Other titles in the SFB393 series:

01-01 G. Kunert. Robust local problem error estimation for a singularly perturbed problem on anisotropic finite element meshes. January 2001.

01-02 G. Kunert. A note on the energy norm for a singularly perturbed model problem. January 2001.

01-03 U.-J. Görke, A. Bucher, R. Kreißig. Ein Beitrag zur Materialparameteridentifikation bei finiten elastisch-plastischen Verzerrungen durch Analyse inhomogener Verschiebungsfelder mit Hilfe der FEM. Februar 2001.

01-04 R. A. Römer. Percolation, Renormalization and the Quantum-Hall Transition. February 2001.

01-05 A. Eilmes, R. A. Römer, C. Schuster, M. Schreiber. Two and more interacting particles at a metal-insulator transition. February 2001.

01-06 D. Michael. Kontinuumstheoretische Grundlagen und algorithmische Behandlung von ausgewählten Problemen der assoziierten Fließtheorie. März 2001.

01-07 S. Beuchler. A preconditioner for solving the inner problem of the p-version of the FEM, Part II - algebraic multi-grid proof. March 2001.

01-08 S. Beuchler, A. Meyer. SPC-PM3AdH v 1.0 - Programmer's Manual. March 2001.

01-09 D. Michael, M. Springmann. Zur numerischen Simulation des Versagens duktiler metallischer Werkstoffe (Algorithmische Behandlung und Vergleichsrechnungen). März 2001.

01-10 B. Heinrich, S. Nicaise. Nitsche mortar finite element method for transmission problems with singularities. March 2001.

01-11 T. Apel, S. Grosman, P. K. Jimack, A. Meyer. A New Methodology for Anisotropic Mesh Refinement Based Upon Error Gradients. March 2001.

01-12 F. Seifert, W. Rehm. (Eds.) Selected Aspects of Cluster Computing. March 2001.

01-13 A. Meyer, T. Steidten. Improvements and Experiments on the Bramble–Pasciak Type CG for mixed Problems in Elasticity. April 2001.

01-14 K. Ragab, W. Rehm. CHEMPI: Efficient MPI for VIA/SCI. April 2001.

01-15 D. Balkanski, F. Seifert, W. Rehm. Proposing a System Software for an SCI-based VIA Hardware. April 2001.

01-16 S. Beuchler. The MTS-BPX-preconditioner for the p-version of the FEM. May 2001.

01-17 S. Beuchler. Preconditioning for the p-version of the FEM by bilinear elements. May 2001.

01-18 A. Meyer. Programmer's Manual for Adaptive Finite Element Code SPC-PM 2Ad. May 2001.

01-19 P. Cain, M.L. Ndawana, R.A. Römer, M. Schreiber. The critical exponent of the localization length at the Anderson transition in 3D disordered systems is larger than 1. June 2001

01-20 G. Kunert, S. Nicaise. Zienkiewicz-Zhu error estimators on anisotropic tetrahedral and triangular finite element meshes. July 2001.

01-21 G. Kunert. A posteriori H^1 error estimation for a singularly perturbed reaction diffusion problem on anisotropic meshes. August 2001.

01-22 A. Eilmes, Rudolf A. Römer, M. Schreiber. Localization properties of two interacting particles in a quasi-periodic potential with a metal-insulator transition. September 2001.

01-23 M. Randrianarivony. Strengthened Cauchy inequality in anisotropic meshes and application to an a-posteriori error estimator for the Stokes problem. September 2001.

01-24 Th. Apel, H. M. Randrianarivony. Stability of discretizations of the Stokes problem on anisotropic meshes. September 2001.

01-25 Th. Apel, V. Mehrmann, D. Watkins. Structured eigenvalue methods for the computation of corner singularities in 3D anisotropic elastic structures. October 2001.

01-26 P. Cain, F. Milde, R. A. Römer, M. Schreiber. Use of cluster computing for the Anderson model of localization. October 2001. Conf. on Comp. Physics, Aachen (2001).

01-27 P. Cain, F. Milde, R. A. Römer, M. Schreiber. Applications of cluster computing for the Anderson model of localization. October 2001. Transworld Research Network for a review compilation entitled "Recent Research Developments in Physics", (2001).

01-28 X. W. Guan, A. Foerster, U. Grimm, R. A. Römer, M. Schreiber. A supersymmetric Uq[(osp)(2—2)]-extended Hubbard model with boundary fields. October 2001.

01-29 K. Eppler, H. Harbrecht. Numerical studies of shape optimization problems in elasticity using wavelet-based BEM. November 2001.

01-30 A. Meyer. The adaptive finite element method - Can we solve arbitrarily accurate? November 2001.

01-31 H. Harbrecht, S. Pereverzev, R. Schneider. An adaptive regularization by projection for noisy pseudodifferential equations of negative order. November 2001.

01-32 G. N. Gatica, H. Harbrecht, R. Schneider. Least squares methods for the coupling of FEM and BEM. November 2001.

01-33 Th. Apel, A.-M. Sändig, S. I. Solov'ev. Computation of 3D vertex singularities for linear elasticity: Error estimates for a finite element method on graded meshes. December 2001.

02-01 M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen - Basisroutinen für Kommunikation und Grafik. Januar 2002.

02-02 M. Pester. Visualization Tools for 2D and 3D Finite Element Programs - User's Manual. January 2002.

02-03 H. Harbrecht, M. Konik, R. Schneider. Fully Discrete Wavelet Galerkin Schemes. January 2002.

02-04 G. Kunert. A posteriori error estimation for convection dominated problems on anisotropic meshes. March 2002.

02-05 H. Harbrecht, R. Schneider. Wavelet Galerkin Schemes for 3D-BEM. February 2002.

02-06 W. Dahmen, H. Harbrecht, R. Schneider. Compression Techniques for Boundary Integral Equations - Optimal Complexity Estimates. April 2002.

02-07 S. Grosman. Robust local problem error estimation for a singularly perturbed reaction-diffusion problem on anisotropic finite element meshes. May 2002.

02-08 M. Springmann, M. Kuna. Identifikation schädigungsmechanischer Materialparameter mit Hilfe nichtlinearer Optimierungsverfahren am Beispiel des Rousselier Modells. Mai 2002.

02-09 S. Beuchler, R. Schneider, C. Schwab. Multiresolution weighted norm equivalences and applications. July 2002.

02-10 Ph. Cain, R. A. Römer, M. E. Raikh. Renormalization group approach to energy level statistics at the integer quantum Hall transition. July 2002.

02-11 A. Eilmes, R. A. Römer, M. Schreiber. Localization properties of two interacting particles in a quasiperiodic potential with a metal-insulator transition. July 2002.

02-12 M. L. Ndawana, R. A. Römer, M. Schreiber. Scaling of the Level Compressibility at the Anderson Metal-Insulator Transition. September 2002.

02-13 Ph. Cain, R. A. Römer, M. E. Raikh. Real-space renormalization group approach to the quantum Hall transition. September 2002.

02-14 A. Jellal, E. H. Saidi, H. B. Geyer, R. A. Römer. A Matrix Model for $\nu_{k_1 k_2} = \frac{k_1+k_2}{k_1 k_2}$ Fractional Quantum Hall States. September 2002.

02-15 M. Randrianarivony, G. Brunnett. Parallel implementation of curve reconstruction from noisy samples. August 2002.

02-16 M. Randrianarivony, G. Brunnett. Parallel implementation of surface reconstruction from noisy samples. September 2002.

02-17 M. Morgenstern, J. Klijn, Chr. Meyer, R. A. Römer, R. Wiesendanger. Comparing measured and calculated local density of states in a disordered two-dimensional electron system. September 2002.

02-18 J. Hippold, G. Rünger. Task Pool Teams for Implementing Irregular Algorithms on Clusters of SMPs. October 2002.

02-19 H. Harbrecht, R. Schneider. Wavelets for the fast solution of boundary integral equations. October 2002.

02-20 H. Harbrecht, R. Schneider. Adaptive Wavelet Galerkin BEM. October 2002.

02-21 H. Harbrecht, R. Schneider. Wavelet Galerkin Schemes for Boundary Integral Equations - Implementation and Quadrature. October 2002.

03-01 E. Creusé, G. Kunert, S. Nicaise. A posteriory error estimation for the Stokes problem: Anisotropic and isotropic discretizations. January 2003.

03-02 S. I. Solov'ëv. Existence of the guided modes of an optical fiber. January 2003.

03-03 S. Beuchler. Wavelet preconditioners for the p-version of the FEM. February 2003.

03-04 S. Beuchler. Fast solvers for degenerated problems. February 2003.

03-05 A. Meyer. Stable calculation of the Jacobians for curved triangles. February 2003.

03-06 S. I. Solov'ëv. Eigenvibrations of a plate with elastically attached load. February 2003.

03-07 H. Harbrecht, R. Schneider. Wavelet based fast solution of boundary integral equations. February 2003.

03-08 S. I. Solov'ëv. Preconditioned iterative methods for monotone nonlinear eigenvalue problems. March 2003.

03-09 Th. Apel, N. Düvelmeyer. Transformation of hexahedral finite element meshes into tetrahedral meshes according to quality criteria. May 2003.

03-10 H. Harbrecht, R. Schneider. Biorthogonal wavelet bases for the boundary element method. April 2003.

03-11 T. Zhanlav. Some choices of moments of refinable function and applications. June 2003.

03-12 S. Beuchler. A Dirichlet-Dirichlet DD-pre-conditioner for p-FEM. June 2003.

03-13 Th. Apel, C. Pester. Clément-type interpolation on spherical domains - interpolation error estimates and application to a posteriori error estimation. July 2003.

03-14 S. Beuchler. Multi-level solver for degenerated problems with applications to p-version of the fem. *(Dissertation)* July 2003.

03-15 Th. Apel, S. Nicaise. The inf-sup condition for the Bernardi-Fortin-Raugel element on anisotropic meshes. September 2003.

03-16 G. Kunert, Z. Mghazli, S. Nicaise. A posteriori error estimation for a finite volume discretization on anisotropic meshes. September 2003.

The complete list of current and former preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.

Preprintreihe des Chemnitzer SFB 393 – ISSN 1619-7178 (Print)