

MOX-Report No. 20/2017

**Fair surface reconstruction through rational triangular
cubic Bézier patches**

Albrecht G.; Caliò F.; Miglio E.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<http://mox.polimi.it>

Fair surface reconstruction through rational triangular cubic Bézier patches

G. Albrecht [‡], F. Calì [†], E. Miglio [◇]

March 10, 2017

[‡] Escuela de Matemáticas,
Universidad Nacional de Colombia,
Sede Medellín, Calle 59 A No 63-20, Medellín, Colombia

galbrecht@unal.edu.co

[†] Dept. of Mathematics,
Politecnico di Milano,
P.zza L. da Vinci 32, 20133 Milan, Italy,

franca.calio@polimi.it

[◇] MOX, Dept. of Mathematics,
Politecnico di Milano,
P.zza L. da Vinci 32, 20133 Milan, Italy,

edie.miglio@polimi.it

Keywords: triangular Bézier patches, G^1 interpolant, rational blending

Abstract

We consider the problem from reverse engineering to construct a G^1 continuous interpolant to a triangulated set of 3D points and corresponding normals by fitting a composite surface consisting of rational triangular Bézier patches by using the so-called rational blend technique. The solution depends on free shape parameters which are fixed by minimizing different functionals depending on suitable surface metrics.

1 Introduction

To reconstruct a virtual model of an object, starting from a cloud of points, is one of the main problems in the context of creative or reverse design or reverse engineering. Several approaches for accomplishing this task may be distinguished.

First of all, the input data are either approximated or interpolated. We are concerned with interpolating methods. Here, we further distinguish between global versus local interpolating reconstruction methods.

As far as global approaches are concerned, a global fitting surface is constructed such that its parametrization is optimal with respect to specific requirements. In this context we can, e.g., cite the article [3], where an algorithm is developed to reconstruct, without any kind of subdivision, a free form surface controlled by a set of points, whose positions are chosen in order to preserve the shape of the input data and to satisfy geometrical and/or functional properties, imposed by the designer. The used integral tensor product B-Spline surface depends on free shape parameters which are determined by minimizing a suitable user-defined functional. The functional depends on a specific surface metric chosen according to the application needs.

As far as local approaches are concerned the most popular methods are based on first triangulating the given input data and then locally replacing each planar triangle in the input mesh by a curved shape thus forming a composite surface. These methods are sufficiently general to represent surfaces of arbitrary genus. We are concerned with methods using parametric triangular Bézier patches interpolating vertex positions and associated normal in order to achieve positional and tangential continuity between the patches. In this context the survey [6] presents the main solutions and methods proposed up to the beginning of the 1990's. The more recent article [1] compares several approaches using a particular technique called rational blend providing G^1 continuous surfaces for visualization purposes. Here we focus on the rational cubic Gregory patch presented in [1], which showed very good performance especially regarding computational cost as well as the approximation of well known surfaces such as the sphere and the torus. On arbitrary triangular meshes it suffers from a certain flatness of the patches' boundary curves. The purpose of the present article is thus to improve the above rational blend cubic Gregory patch from [1] by introducing free shape parameters which are chosen to optimize the shape of the patches by minimizing certain functionals as in [3].

The remainder of the article is organized as follows. In section 2 we give the preliminaries to recall the terminology and the algorithm for the rational blend Gregory patch from [1]. In section 3 we propose a new scheme to determine a parametric form of the patch based on rational cubic boundary curves thus leaving some free shape parameters. In section 4 we show how to use these parameters to minimize locally suitable functionals with respect to a particular metric. Finally we describe in detail the algorithm used and we give some numerical results to confirm the good behavior of the proposed method in section 5.

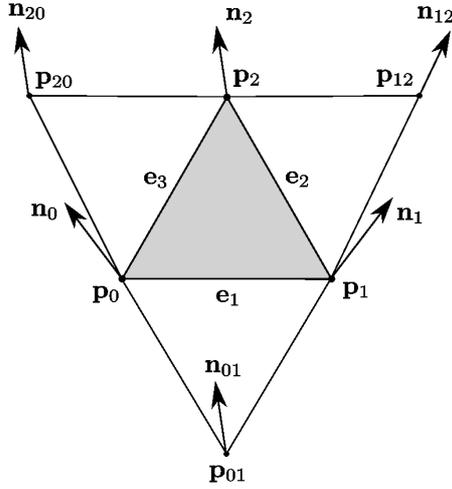


Figure 1: Notation for the vertexes and normals of the input flat triangles.

2 Preliminaries and recalls

Constructing two patches that meet with tangent plane continuity, called G^1 continuity, is rather straight-forward. On the contrary, a complex problem called vertex consistency problem arises when constructing a closed network of G^1 joined patches incident to a vertex. Different methods have been developed to deal with this problem. In [1] is presented a method, inspired by the work of Walton and Meek [9] which uses cubic Gregory patches with the rational blend technique.

We now recall the notation and the algorithm from [1]. A subset of 4 triangles is considered and as illustrated in Figure 1, the notation $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ is used for the vertices of the central triangle, $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$ for the respective unit normal vectors in these vertices, and $\mathbf{e}_0 = \mathbf{p}_1 - \mathbf{p}_0, \mathbf{e}_1 = \mathbf{p}_2 - \mathbf{p}_1, \mathbf{e}_2 = \mathbf{p}_0 - \mathbf{p}_2$ for the triangle's edge vectors. Considering the neighboring triangle adjacent to the edge \mathbf{e}_1 , the notation \mathbf{p}_{01} for its remaining vertex and \mathbf{n}_{01} for its associated normal is used. Analogously, are defined $\mathbf{p}_{12}, \mathbf{n}_{12}$ with respect to the edge \mathbf{e}_2 and $\mathbf{p}_{20}, \mathbf{n}_{20}$ with respect to the edge \mathbf{e}_3 . Additionally, the tangent plane in \mathbf{p}_i , which is defined by the normal vector \mathbf{n}_i , is denoted by τ_i for $i = 0, 1, 2, 01, 12, 20$.

Now, in order to introduce the G^1 rational blend interpolatory schemes, using a triangular network of control points $\mathbf{b}_{i,j,k}, (i + j + k = n, i, j, k \geq 0)$ and degree- n bivariate Bernstein polynomials $B_{i,j,k}^n(u, v, w) = \frac{n!}{i!j!k!} u^i v^j w^k$ ($u + v + w = 1$), a triangular Bézier patch is defined by

$$\mathbf{b}(u, v, w) = \sum_{i+j+k=n} \mathbf{b}_{i,j,k}(u, v, w) B_{i,j,k}^n(u, v, w) \quad (1)$$

It maps a triangular domain $D \subset \mathbb{R}^2$ to an affine space, typically \mathbb{R}^3 , where u, v and w are the barycentric coordinates in D . The approach from [1] is based on the creation of a triangular Bézier patch by means of rational blends, i.e. where the control points are affine combinations of the points using rational blending functions. Precisely (see [1]), for $n=3$, six points \mathbf{b}_{111}^{11} , \mathbf{b}_{111}^{12} (referred to edge \mathbf{e}_1) \mathbf{b}_{111}^{21} , \mathbf{b}_{111}^{22} (referred to edge \mathbf{e}_2) and \mathbf{b}_{111}^{31} , \mathbf{b}_{111}^{32} (referred to edge \mathbf{e}_3) are blended to define the interior control point $\mathbf{b}_{111}(u, v, w)$ of their rational blend cubic triangular Bézier patch (1) as follows:

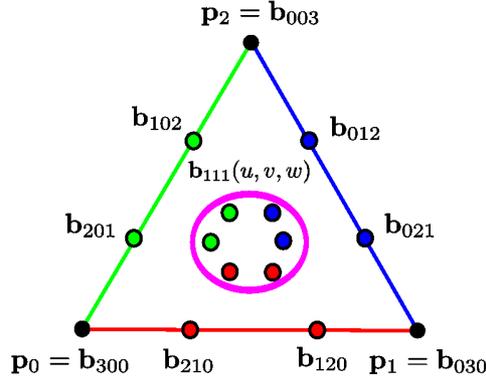


Figure 2: Points defining $\mathbf{b}_{111}(u, v, w)$ for the cubic version of Walton and Meek.

$$\begin{aligned} \mathbf{b}_{111}(u, v, w) = & u \left(\frac{v \cdot \mathbf{b}_{111}^{11} + w \cdot \mathbf{b}_{111}^{32}}{v + w} \right) \\ & + v \left(\frac{w \cdot \mathbf{b}_{111}^{21} + u \cdot \mathbf{b}_{111}^{12}}{w + u} \right) \\ & + w \left(\frac{u \cdot \mathbf{b}_{111}^{31} + v \cdot \mathbf{b}_{111}^{22}}{u + v} \right). \end{aligned} \quad (2)$$

Now, using the process (see [1]) to impose the G^1 continuity, the points \mathbf{b}_{111}^{11} , \mathbf{b}_{111}^{12} , \mathbf{b}_{111}^{21} , \mathbf{b}_{111}^{22} , \mathbf{b}_{111}^{31} and \mathbf{b}_{111}^{32} are obtained to be blended according to formula (2).

Finally from (1) the surface patch is created, but, analyzing the reconstructed surface, a problem from a certain flatness of boundary curves arises. In the next section we will propose, in order to overcome this problem, a rational version of the described method, where the parametric equations of boundary curves have some free weights, that can act as free shape parameters.

3 A rational cubic Gregory patch

The idea is to work with *rational* triangular Bézier patches, i.e., to define the interior control point $\mathbf{b}_{111}(u, v, w)$ of the patch by means of *rational blends* as well as the interior weights.

Let $\mathbf{s}(u, v, w)$ be a rational cubic triangular Bézier patch defined by

$$\mathbf{s}(u, v, w) = \frac{\sum_{i+j+k=3} \bar{\mathbf{b}}_{ijk}(u, v, w) B_{ijk}^3(u, v, w)}{\sum_{i+j+k=3} \omega_{ijk}(u, v, w) B_{ijk}^3(u, v, w)} = \frac{\mathbf{p}(u, v, w)}{\omega(u, v, w)}. \quad (3)$$

Here the points

$$\mathbf{b}_{ijk}(u, v, w) = \frac{\bar{\mathbf{b}}_{ijk}(u, v, w)}{\omega_{ijk}(u, v, w)} \quad (4)$$

are the control points and $\omega_{ijk}(u, v, w)$ the so-called weights of the patch.

Its boundary curves in rational cubic Bézier form are

$$\mathbf{c}_i(t) = \frac{\sum_{k=0}^3 \bar{\mathbf{b}}_k^i B_k^3(t)}{\sum_{k=0}^3 \omega_k^i B_k^3(t)} = \frac{\mathbf{p}_i(t)}{\omega_i(t)}, \quad (5)$$

where obviously the control points $\mathbf{b}_k^i = \frac{\bar{\mathbf{b}}_k^i}{\omega_k^i}$ and the weights ω_k^i are, respectively, the boundary control points and boundary weights of \mathbf{s} . For example, to construct the curve $\mathbf{c}_1(t)$ corresponding to $u = t, v = 1 - t, w = 0$, we determine the coefficients $\mathbf{b}_0^1 = \mathbf{b}_{300}, \mathbf{b}_1^1 = \mathbf{b}_{210}, \mathbf{b}_2^1 = \mathbf{b}_{120}$ and $\mathbf{b}_3^1 = \mathbf{b}_{030}$, using Walton and Meek's method [9]. According to this method, we denote:

$d_1 = \|\mathbf{b}_3^1 - \mathbf{b}_0^1\|$, $\mathbf{g}_1 = \frac{(\mathbf{b}_3^1 - \mathbf{b}_0^1)}{d_1}$, $f_1 = \mathbf{n}_0 \cdot \mathbf{n}_1$, $f_{1,0} = \mathbf{n}_0 \cdot \mathbf{g}_1$ and $f_{1,1} = \mathbf{n}_1 \cdot \mathbf{g}_1$. We also consider $r_1 = \frac{6(2f_{1,0} + f_1 f_{1,1})}{4 - f_1^2}$ and $s_1 = \frac{6(2f_{1,1} + f_1 f_{1,0})}{4 - f_1^2}$.

Then the cubic rational Bézier curve $\mathbf{c}_1(t)$ with $\mathbf{b}_0^1 = \mathbf{p}_0$, $\mathbf{b}_3^1 = \mathbf{p}_1$, $\mathbf{b}_1^1 = \mathbf{b}_0^1 + \frac{d_1}{18}(6\mathbf{g}_1 - 2r_1\mathbf{n}_0 + s_1\mathbf{n}_1)$ and $\mathbf{b}_2^1 = \mathbf{b}_3^1 - \frac{d_1}{18}(6\mathbf{g}_1 + r_1\mathbf{n}_0 - 2s_1\mathbf{n}_1)$ joins \mathbf{p}_0 to \mathbf{p}_1 and has its principal normal direction parallel to \mathbf{n}_0 at $t = 0$ and parallel to \mathbf{n}_1 at $t = 1$, respectively, according to Walton and Yeung [10]. Its weights remain to be freely chosen. In an analogous way the boundary curves $\mathbf{c}_2(t)$ and $\mathbf{c}_3(t)$ are constructed.

The derivatives of these curves are given by

$$\dot{\mathbf{c}}_i(t) = \frac{1}{\omega_i^2(t)} [\omega_i(t) \dot{\mathbf{p}}_i(t) - \mathbf{p}_i(t) \dot{\omega}_i(t)], \quad (6)$$

where

$$\begin{aligned} \dot{\mathbf{p}}_i(t) &= 3 \sum_{k=0}^2 (\bar{\mathbf{b}}_{k+1}^i - \bar{\mathbf{b}}_k^i) B_k^2(t), \\ \dot{\omega}_i(t) &= 3 \sum_{k=0}^2 (\omega_{k+1}^i - \omega_k^i) B_k^2(t). \end{aligned} \quad (7)$$

Now we introduce the vector:

$$\mathbf{h}_i(t) = \sum_{k=0}^2 \mathbf{a}_k^i B_k^2(t), \quad 0 \leq t \leq 1$$

where

$$\mathbf{a}_0^i = \mathbf{n}_{i-1} \times \frac{\mathbf{w}_0^i}{\|\mathbf{w}_0^i\|},$$

$$\mathbf{a}_2^i = \mathbf{n}_i \times \frac{\mathbf{w}_2^i}{\|\mathbf{w}_2^i\|},$$

$$\mathbf{a}_1^i = \frac{\mathbf{a}_0^i + \mathbf{a}_2^i}{\|\mathbf{a}_0^i + \mathbf{a}_2^i\|}$$

with $\mathbf{n}_3 = \mathbf{n}_0$ and $\mathbf{w}_k^i = \mathbf{b}_{k+1}^i - \mathbf{b}_k^i$, $k = 0, 1, 2$.

The plane spanned by the vectors $\dot{\mathbf{c}}_i(t)$ and $\mathbf{h}_i(t)$ creates a tangent ribbon along the edge \mathbf{e}_i , as shown in Figure 3 for the edge \mathbf{e}_1 .

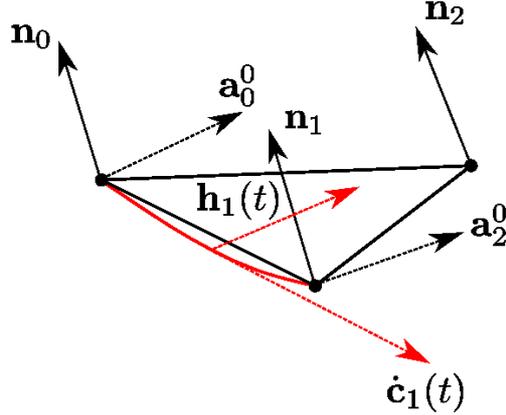


Figure 3: The plane spanned by the tangent vector $\dot{\mathbf{c}}_1(t)$ and the vector $\mathbf{h}_1(t)$.

To create our surface patch and to insure G^1 -continuity with the neighboring triangle $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_{01}$ along the edge \mathbf{e}_1 , we firstly consider the cross-boundary directional derivative.

The directional derivatives of \mathbf{s} , in the directions $\mathbf{d}_1 = (-1/2, -1/2, 1)$, $\mathbf{d}_2 = (1, -1/2, -1/2)$ and $\mathbf{d}_3 = (-1/2, 1, -1/2)$ are defined by

$$\mathbf{s}_i^{\mathbf{d}_i}(t) = \frac{1}{\omega_i^2(t)} [\omega_i(t) \mathbf{p}^{\mathbf{d}_i}(t) - \mathbf{p}_i(t) \omega^{\mathbf{d}_i}(t)], \quad (8)$$

where, after degree elevation,

$$\begin{aligned}
\mathbf{p}^{\mathbf{d}_i}(t) &= 3 \sum_{k=0}^3 \bar{\Delta}_k^i B_k^3(t), \\
\omega^{\mathbf{d}_i}(t) &= 3 \sum_{k=0}^3 \Omega_k^i B_k^3(t).
\end{aligned} \tag{9}$$

As already mentioned in [2], related to the edge \mathbf{e}_1 , we obtain the $\bar{\Delta}_k^1$ and their corresponding weights, as it follows:

$$\begin{aligned}
\bar{\Delta}_0^1 &= -\frac{1}{2}\bar{\mathbf{b}}_{300} - \frac{1}{2}\bar{\mathbf{b}}_{210} + \bar{\mathbf{b}}_{201}, \\
\bar{\Delta}_1^1 &= \frac{1}{3} \left(-\frac{1}{2}\bar{\mathbf{b}}_{300} - \frac{1}{2}\bar{\mathbf{b}}_{210} + \bar{\mathbf{b}}_{201} \right) + \frac{2}{3} \left(-\frac{1}{2}\bar{\mathbf{b}}_{210} - \frac{1}{2}\bar{\mathbf{b}}_{120} + \bar{\mathbf{b}}_{111}^1 \right), \\
\bar{\Delta}_2^1 &= \frac{2}{3} \left(-\frac{1}{2}\bar{\mathbf{b}}_{210} - \frac{1}{2}\bar{\mathbf{b}}_{120} + \bar{\mathbf{b}}_{111}^2 \right) + \frac{1}{3} \left(-\frac{1}{2}\bar{\mathbf{b}}_{120} - \frac{1}{2}\bar{\mathbf{b}}_{030} + \bar{\mathbf{b}}_{021} \right), \\
\bar{\Delta}_3^1 &= -\frac{1}{2}\bar{\mathbf{b}}_{120} - \frac{1}{2}\bar{\mathbf{b}}_{030} + \bar{\mathbf{b}}_{021},
\end{aligned} \tag{10}$$

$$\begin{aligned}
\Omega_0^1 &= -\frac{1}{2}\omega_{300} - \frac{1}{2}\omega_{210} + \omega_{201}, \\
\Omega_1^1 &= \frac{1}{3} \left(-\frac{1}{2}\omega_{300} - \frac{1}{2}\omega_{210} + \omega_{201} \right) + \frac{2}{3} \left(-\frac{1}{2}\omega_{210} - \frac{1}{2}\omega_{120} + \omega_{111}^1 \right), \\
\Omega_2^1 &= \frac{2}{3} \left(-\frac{1}{2}\omega_{210} - \frac{1}{2}\omega_{120} + \omega_{111}^2 \right) + \frac{1}{3} \left(-\frac{1}{2}\omega_{120} - \frac{1}{2}\omega_{030} + \omega_{021} \right), \\
\Omega_3^1 &= -\frac{1}{2}\omega_{120} - \frac{1}{2}\omega_{030} + \omega_{021}.
\end{aligned} \tag{11}$$

The conditions to be fulfilled to ensure G^1 -continuity with the neighboring triangles are:

$$\mathbf{s}_i^{\mathbf{d}_i}(t) = \frac{1}{3}\alpha_i(t)\dot{\mathbf{c}}_i(t) + \beta_i(t)\mathbf{h}_i(t), \quad i = 1, 2, 3. \tag{12}$$

Since in equation (12) the derivatives are complicated, we use their homogeneous representations (see, e.g., [5]). In projective coordinates the representation of $\mathbf{s}(u, v, w)$ is

$$\mathbf{S}(u, v, w) = \{\mathbf{p}(u, v, w), \omega(u, v, w)\} = \sum_{i+j+k=3} \mathbf{x}_{ijk} B_{ijk}^3(u, v, w), \tag{13}$$

where the control net in homogeneous coordinates is given by

$$\mathbf{x}_{ijk} = \{\bar{\mathbf{b}}_{ijk}, \omega_{ijk}\}. \tag{14}$$

The conditions to ensure G^1 -continuity by using the homogeneous representation of the patch are derived in the following proposition.

Proposition 1. *A sufficient condition in order for (12) to be fulfilled is the existence of two polynomials $\alpha(t)$ and $\beta(t)$ such that*

$$\begin{cases} \mathbf{p}^d(t) = \alpha(t)\dot{\mathbf{p}}(t) + \beta(t)\mathbf{h}(t), \\ \omega^d(t) = \alpha(t)\dot{\omega}(t), \end{cases} \quad (15)$$

where $\mathbf{p}^d(t)$, $\omega^d(t)$, $\dot{\mathbf{p}}(t)$ and $\dot{\omega}(t)$ are, respectively, defined in (9) and (7) and, for the sake of simplicity, the index i has been omitted.

Proof: In the affine coordinate system the G^1 condition (12) is equivalent to

$$\det(\mathbf{s}^d(t), \dot{\mathbf{c}}(t), \mathbf{h}(t)) = 0. \quad (16)$$

We substitute (6) and (8) in the left hand side of (16) and thus obtain

$$\begin{aligned} \det(\mathbf{s}^d, \dot{\mathbf{c}}, \mathbf{h}) &= \frac{1}{\omega^4} \det(\omega\mathbf{p}^d - \mathbf{p}\omega^d, \omega\dot{\mathbf{p}} - \mathbf{p}\dot{\omega}, \mathbf{h}) = \\ &= \frac{1}{\omega^4} \det \begin{pmatrix} 1 & 0 & 0 & 0 \\ \mathbf{p} & \omega\mathbf{p}^d - \mathbf{p}\omega^d & \omega\dot{\mathbf{p}} - \mathbf{p}\dot{\omega} & \mathbf{h} \end{pmatrix} = \\ &= \frac{\omega^d\dot{\omega}}{\omega^4} \det \begin{pmatrix} 1 & 1 & 1 & 0 \\ \mathbf{p} & \frac{\omega}{\omega^d}\mathbf{p}^d & \frac{\omega}{\dot{\omega}}\dot{\mathbf{p}} & \mathbf{h} \end{pmatrix} = \\ &= \frac{1}{\omega^3} \det \begin{pmatrix} \omega & \omega^d & \dot{\omega} & 0 \\ \mathbf{p} & \mathbf{p}^d & \dot{\mathbf{p}} & \mathbf{h} \end{pmatrix} = \\ &= -\frac{1}{\omega^3} \det \begin{pmatrix} \mathbf{p} & \omega \\ \mathbf{p}^d & \omega^d \\ \dot{\mathbf{p}} & \dot{\omega} \\ \mathbf{h} & 0 \end{pmatrix}. \end{aligned}$$

Equation (16) is thus equivalent to the existence of four scalar functions $f(t)$, $g(t)$, $m(t)$ and $n(t)$ such that

$$f(t) \begin{pmatrix} \mathbf{p}(t) \\ \omega(t) \end{pmatrix} + g(t) \begin{pmatrix} \mathbf{p}^d(t) \\ \omega^d(t) \end{pmatrix} + m(t) \begin{pmatrix} \dot{\mathbf{p}}(t) \\ \dot{\omega}(t) \end{pmatrix} + n(t) \begin{pmatrix} \mathbf{h}(t) \\ 0 \end{pmatrix} = 0. \quad (17)$$

We choose $f(t) = 0$ and we obtain

$$\begin{pmatrix} \mathbf{p}^d(t) \\ \omega^d(t) \end{pmatrix} = -\frac{m(t)}{g(t)} \begin{pmatrix} \dot{\mathbf{p}}(t) \\ \dot{\omega}(t) \end{pmatrix} - \frac{n(t)}{g(t)} \begin{pmatrix} \mathbf{h}(t) \\ 0 \end{pmatrix}. \quad (18)$$

Therefore, by defining $\alpha(t) = -\frac{m(t)}{g(t)}$ and $\beta(t) = -\frac{n(t)}{g(t)}$, (12) is equivalent to (15). \square

We thus obtain the desired G^1 -continuity conditions by using the linear functions:

$$\alpha_1(t) = \alpha_0^1(1-t) + \alpha_1^1t \quad \text{and} \quad \beta_1(t) = \beta_0^1(1-t) + \beta_1^1t,$$

and by equaling the right hand sides of (15) and (9).

For $i = 1$ this yields the following two systems of equations:

$$\begin{aligned}
\bar{\Delta}_0^1 &= \alpha_0^1(\bar{\mathbf{b}}_{210} - \bar{\mathbf{b}}_{300}) + \beta_0^1 \mathbf{a}_0^1, \\
\bar{\Delta}_1^1 &= \frac{2}{3}(\alpha_0^1(\bar{\mathbf{b}}_{120} - \bar{\mathbf{b}}_{210}) + \beta_0^1 \mathbf{a}_1^1) + \frac{1}{3}(\alpha_1^1(\bar{\mathbf{b}}_{210} - \bar{\mathbf{b}}_{300}) + \beta_1^1 \mathbf{a}_0^1), \\
\bar{\Delta}_2^1 &= \frac{1}{3}(\alpha_0^1(\bar{\mathbf{b}}_{030} - \bar{\mathbf{b}}_{120}) + \beta_0^1 \mathbf{a}_2^1) + \frac{2}{3}(\alpha_1^1(\bar{\mathbf{b}}_{120} - \bar{\mathbf{b}}_{210}) + \beta_1^1 \mathbf{a}_1^1), \\
\bar{\Delta}_3^1 &= \alpha_1^1(\bar{\mathbf{b}}_{030} - \bar{\mathbf{b}}_{120}) + \beta_1^1 \mathbf{a}_2^1,
\end{aligned} \tag{19}$$

and

$$\begin{aligned}
\Omega_0^1 &= \alpha_0^1(\omega_{210} - \omega_{300}), \\
\Omega_1^1 &= \frac{2}{3}\alpha_0^1(\omega_{120} - \omega_{210}) + \frac{1}{3}\alpha_1^1(\omega_{210} - \omega_{300}), \\
\Omega_2^1 &= \frac{1}{3}\alpha_0^1(\omega_{030} - \omega_{120}) + \frac{2}{3}\alpha_1^1(\omega_{120} - \omega_{210}), \\
\Omega_3^1 &= \alpha_1^1(\omega_{030} - \omega_{120}),
\end{aligned} \tag{20}$$

Once the two systems (19) and (20) are derived, the problem is how to define the weights of the boundary curves such that α_0^1 , α_1^1 , β_0^1 and β_1^1 can be computed from their first and last equations.

From the first equation of (20) we can compute

$$\alpha_0^1 = \frac{\Omega_0^1}{\omega_{210} - \omega_{300}}, \tag{21}$$

and its substitution in the first equation of (19) gives

$$\beta_0^1 \mathbf{a}_0^1 = \mathbf{v}, \tag{22}$$

where

$$\mathbf{v} = \frac{(\omega_{201} - \omega_{210})\bar{\mathbf{b}}_{300} + (\omega_{300} - \omega_{201})\bar{\mathbf{b}}_{210} + (\omega_{210} - \omega_{300})\bar{\mathbf{b}}_{201}}{\omega_{210} - \omega_{300}} \tag{23}$$

β_0^1 can be calculated from (22) if \mathbf{v} and \mathbf{a}_0^1 are collinear. A necessary condition for this is the equation

$$\mathbf{v} \cdot (\bar{\mathbf{b}}_{210} - \bar{\mathbf{b}}_{300}) = 0,$$

which we solve for ω_{201} and obtain

$$\omega_{201}^{sol1} = \frac{(x - 2y + z)\omega_{210}\omega_{300}}{(k - h - y + z)\omega_{210} + (h - k + x - y)\omega_{300}}, \tag{24}$$

where

$$\begin{aligned} x &= \bar{\mathbf{b}}_{300} \cdot \bar{\mathbf{b}}_{300}, \quad y = \bar{\mathbf{b}}_{210} \cdot \bar{\mathbf{b}}_{300}, \quad z = \bar{\mathbf{b}}_{210} \cdot \bar{\mathbf{b}}_{210}, \\ k &= \bar{\mathbf{b}}_{201} \cdot \bar{\mathbf{b}}_{300}, \quad h = \bar{\mathbf{b}}_{201} \cdot \bar{\mathbf{b}}_{210} \end{aligned} \quad (25)$$

By imposing the G^1 -continuity condition along the edge \mathbf{e}_3 and by repeating the above procedure we obtain a second solution for the same weight, i.e.,

$$\omega_{201}^{sol2} = \frac{(h - k + x - y)\omega_{210}\omega_{300}}{(h + k - l - y)\omega_{210} + (l - 2k + x)\omega_{300}}, \quad (26)$$

where additionally to the abbreviations from (25) we have $l = \mathbf{b}_{201} \cdot \mathbf{b}_{201}$.

By solving $\omega_{201}^{sol1} = \omega_{201}^{sol2}$ with respect to ω_{210} we obtain that the only two possible solutions are $\omega_{210} = \omega_{201} = 0$ or $\omega_{210} = \omega_{201} = \omega_{300}$.

We are not interested in the zero valued weights and thus retain the solution

$$\omega_{210} = \omega_{201} = \omega_{300} \quad (27)$$

which trivially satisfies the equation (22), i.e.,

$$\begin{aligned} (\omega_{210} - \omega_{300})\beta_0^1 \mathbf{a}_0^1 &= (\omega_{201} - \omega_{210})\bar{\mathbf{b}}_{300} + (\omega_{300} - \omega_{201})\bar{\mathbf{b}}_{210} \\ &\quad + (\omega_{210} - \omega_{300})\bar{\mathbf{b}}_{201}, \end{aligned} \quad (28)$$

but does not allow to calculate β_0^1 from it.

By applying the above choice (27) of the weights in all the patch's corners, i.e.,

$$\omega_{210} = \omega_{201} = \omega_{300}, \quad \omega_{120} = \omega_{021} = \omega_{030}, \quad \omega_{102} = \omega_{012} = \omega_{003} \quad (29)$$

the equation systems derived from the G^1 -continuity requirements along the borders can now be solved. In fact, along the edge \mathbf{e}_1 for example, the first equation of (19) becomes

$$\omega_{300}\Delta_0^1 = \omega_{300}\alpha_0^1(\bar{\mathbf{b}}_{210} - \bar{\mathbf{b}}_{300}) + \beta_0^1 \mathbf{a}_0^1 \quad (30)$$

Thus, α_0^1 can be computed as

$$\alpha_0^1 = \frac{[\omega_{300}\Delta_0^1] \cdot [\omega_{300}\mathbf{w}_0^1]}{[\omega_{300}\mathbf{w}_0^1] \cdot [\omega_{300}\mathbf{w}_0^1]} = \frac{\Delta_0^1 \cdot \mathbf{w}_0^1}{\mathbf{w}_0^1 \cdot \mathbf{w}_0^1}. \quad (31)$$

Analogously,

$$\beta_0^1 = \omega_{300}(\Delta_0^1 \cdot \mathbf{a}_0^1). \quad (32)$$

The first and the last equation in (20) are identically satisfied. From the two central equations the weights ω_{111}^{11} and ω_{111}^{12} can be computed as

$$\begin{aligned} \omega_{111}^{11} &= \left(\frac{1}{2} - \alpha_0^1\right)\omega_{300} + \left(\frac{1}{2} + \alpha_0^1\right)\omega_{030}, \\ \omega_{111}^{12} &= \left(\frac{1}{2} - \alpha_1^1\right)\omega_{300} + \left(\frac{1}{2} + \alpha_1^1\right)\omega_{030}. \end{aligned} \quad (33)$$

Finally, once $\alpha_0^1, \alpha_1^1, \beta_0^1, \beta_1^1$ and $\omega_{111}^{11}, \omega_{111}^{12}$ are calculated, the points \mathbf{b}_{111}^{11} and \mathbf{b}_{111}^{12} can be obtained from the two central equations of (19).

Repeating this procedure for the three borders we obtain the six points $\mathbf{b}_{111}^{11}, \mathbf{b}_{111}^{12}, \mathbf{b}_{111}^{21}, \mathbf{b}_{111}^{22}, \mathbf{b}_{111}^{31}$ and \mathbf{b}_{111}^{32} and the six weights $\omega_{111}^{11}, \omega_{111}^{12}, \omega_{111}^{21}, \omega_{111}^{22}, \omega_{111}^{31}$ and ω_{111}^{32} , to be blended, respectively, for the interior control point \mathbf{b}_{111} and the interior control weight ω_{111} as in (2).

4 An automatic approach to define the three free weights

In the previous sections we described how to construct a rational cubic Gregory patch in which there are three free weights $\omega_{300}, \omega_{030}, \omega_{003}$ that act as shape parameters.

The next step is to investigate how these three weights influence the shape of the patch.

Actually the weights act as shape parameters in correspondence of each mesh vertex. Leaving them free allows the designer to decide how to modify the surface in correspondence to the interpolating points, increasing or decreasing the corresponding vertex weights.

Nonetheless, we tried to find a way to define these vertex weights automatically, namely by using suitably (as designer decides) the information from the points and normals of the mesh.

To this aim it's necessary to develop geometric models and algorithms that automatically create shapes close as much as possible to the given data and assuring at the same time a good representation of the geometric and functional properties of the required virtual model.

4.1 Fairness metrics

We will now discuss some functionals, involving the acquired information, that can be used in order to achieve different aims, according to application needs. From the mathematical point of view, different results about the patch fairness can be obtained using different functionals involving geometrical surface properties. A discussion on the meaning of different functionals can be found in [7] and in [8].

We recall that it is possible to introduce metrics used to measure the smoothness of the surface at interest and to produce, consequently, the required shape adjustment. These metrics must depend only on invariants such that a re-parametrization of the surface does not affect the value of the measure.

In the following, taking into account $w = 1 - u - v$, we consider the expression of $\mathbf{s}(u, v, w)$ in (3) depending only on two parameters u, v .

Let G and H be the Gaussian and the mean curvature respectively of the surface and \mathbf{n} the normal vector; the following three derived surfaces are considered as metrics:

1. the *flattening metric* which is the surface area of the derived surface

$$\mathbf{S}(u, v) = G(u, v)\mathbf{n}(u, v). \quad (34)$$

This metric tends to minimize the magnitude of the Gaussian curvature G as well as extreme changes in the Gaussian curvature along the lines of curvature: the tendency is to flatten the surface.

2. the *rounding metric* which is the surface area of the derived surface

$$\mathbf{S}(u, v) = \mathbf{s}(u, v) + [H(u, v)/G(u, v)]\mathbf{n}(u, v). \quad (35)$$

This metric tends to pull the surface towards a sphere.

3. the *rolling metric* which is the surface area of the derived surface

$$\mathbf{S}(u, v) = [G(u, v) + H^2(u, v)]\mathbf{n}(u, v). \quad (36)$$

This metric tends to make the surface more cylindrical or conical.

The functionals used in the algorithm we are proposing are obtained computing the surface area A of the above mentioned derived surface, i.e.:

$$A = \int_R \left| \frac{\partial \mathbf{S}}{\partial u} \times \frac{\partial \mathbf{S}}{\partial v} \right| dudv, \quad (37)$$

where R is the parameter domain of $\mathbf{s}(u, v)$.

4.2 Discrete functional definition

However, as we have a scattered cloud of points defining the patch, we have to compute the discrete Gaussian and mean curvatures, using the notations in Figure 4 and remembering that in Figure 4 $\mathbf{p}_0^{T_i}, \mathbf{p}_1^{T_i}, \mathbf{p}_2^{T_i}$ are respectively $\mathbf{b}_{300}, \mathbf{b}_{030}, \mathbf{b}_{003}$ of triangle T_i .

Precisely we can compute the mean curvature normal operator H (see [7]) using the following expression:

$$\mathbf{K}(\mathbf{p}_0^{T_i}) = \frac{1}{2A_p} \sum_{i \in N(i)} (\cot \zeta_i + \cot \zeta_{i+1})(\mathbf{p}_0^{T_i} - \mathbf{p}_1^{T_i}), \quad (38)$$

where A_p is a suitable area surrounding point $\mathbf{p}_0^{T_i}$ and can be chosen in different ways (see [7]), $N(i)$ is the number of the set of 1-ring neighbor vertexes of vertex $\mathbf{p}_0^{T_i}$ and ζ_i and ζ_{i+1} are defined in Figure 4.

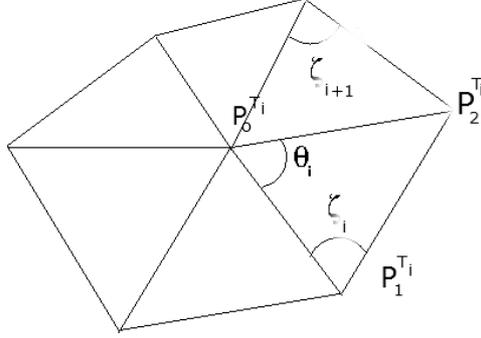


Figure 4: Notations used in the definition of the discrete curvatures for a surface.

The mean curvature H can be easily computed

$$H(\mathbf{p}_0^{T_i}) = \frac{1}{2} \|\mathbf{K}(\mathbf{p}_0^{T_i})\|. \quad (39)$$

Moreover the normal direction vector \mathbf{n} is computed by normalizing the normal vector $\mathbf{K}(\mathbf{p}_0^{T_i})$.

As for Gaussian curvature G , the following expression can be used:

$$G(\mathbf{p}_0^{T_i}) = \left(2\pi - \sum_{j=1}^{N(i)} \theta_j \right) / A_p \quad (40)$$

Calling T_{ij} the j -th triangle of the set of triangles surrounding the vertex $\mathbf{p}_0^{T_i}$, the discretized version of functional (37) becomes:

$$DA = \sum_{i=1}^m \sum_{j=1}^{N(i)} \|(\mathbf{S}(\mathbf{p}_1^{T_{ij}}) - \mathbf{S}(\mathbf{p}_0^{T_{ij}})) \times (\mathbf{S}(\mathbf{p}_2^{T_{ij}}) - \mathbf{S}(\mathbf{p}_0^{T_{ij}}))\| \quad (41)$$

4.3 The minimization algorithm

Now we have the background to describe the used minimization algorithm.

Actually the problem is: to minimize the chosen functionals depending on some parameters with the aim to improve the surface fairness.

Precisely, as $\mathbf{s}, H, G, \mathbf{n}$ and, consequently, \mathbf{S} depend on the $\omega_{300}^{T_i}, \omega_{030}^{T_i}, \omega_{003}^{T_i}$ $i = 1, \dots, m$ parameters, related to the vertexes of triangular patch T_i , we could proceed to functional minimization with respect to these parameters.

At this point we want to reduce the problem dimension, to improve the efficiency and stability of the algorithm. In this sense, for the T_i triangle, we propose a suitable relation between the generic triple $\omega_{300}^{T_i}, \omega_{030}^{T_i}, \omega_{003}^{T_i}$ $i = 1, \dots, m$ and a qualitative indicator of the behavior of all the triangles of

the one-ring neighbors of the vertex of triangle under consideration, using only one parameter.

The idea of our approach is the following:

- firstly we set the vertex weight depending on the average of angles between the normals of two adjacent triangles, namely depending on

$$K^{T_i} = \frac{1}{N(i)} \sum_{j=1}^{N(i)} \gamma_j \quad (42)$$

where:

$$\gamma_j = \mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{T_{i(j+1)}}. \quad (43)$$

$\mathbf{n}_{T_{ij}}$ the normals of the triangles T_{ij} ($j = 1, \dots, N(i)$).

- secondly we propose the relation: (for example referred to $\omega_{300}^{T_i}$)

$$\omega_{300}^{T_i} = W + (1 - W)K_{300}^{T_i} \quad (44)$$

- and then, we insert (44) into the functional (41), by using the discretized version of \mathbf{S} , with one of the metrics (34) - (36). In this way we reduce the multivariate minimization problem to minimization with respect to only one parameter.

We use the MATLAB function FMINSEARCH in our implementation.

After the minimization procedure we compute $\omega_{300}^{T_i}$, $\omega_{030}^{T_i}$, $\omega_{003}^{T_i}$ $i = 1, \dots, m$ depending on optimal W , according to (44) and determine the parametric representation (3) of the surface patch.

The surface described by the final set of parameters, that result from the minimization process, is the faired surface.

5 Numerical results

In this Section we present some examples with the aim of showing the quality and the usefulness of the proposed method.

Two methods for surface reconstruction, Rational Cubic Gregory patches (RCG) with shape parameters, presented in this paper, and tensor integral VDS functions (TIS) with shape parameter, presented in [3], are considered.

The first example has the aim of showing that the RCG method is correctly defined and implemented. Precisely, from a cloud of measured data derived from a spherical surface and perturbed by random noise, the surface is reconstructed by means of the RCG method on a Cartesian mesh and regularized by a fairness functional, defined on a rounding metric (see (35)) and depending on three parameters. Figure 5 illustrates this process.

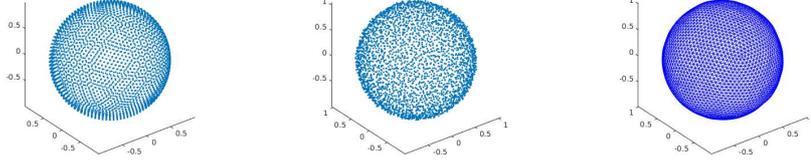


Figure 5: First Example. Data detected from a sphere (left) are perturbed by random noise (center), then reconstructed by means of the RCG method (right).

As a second example, an analytical surface (an hyperbolic paraboloid or “saddle” whose equation is $x^2 - y^2 = z$) is reconstructed starting from a cloud of points on it, perturbed by random noise. The methods used and compared are TIS and RCG respectively. The fairness functional in both cases is based on a flattening metric (see (34)).

The two methods are further compared using two alternate distributions of points for the cloud. Namely, a regularly distributed set of points, giving raise to a rectangular mesh over the xy (horizontal) plane is counter posed to a set of points randomly distributed over the surface, corresponding to an unstructured mesh over the xy -plane. The two meshes are depicted by Figure 6. Results for the regular against the unstructured mesh case are depicted by Figure 7.

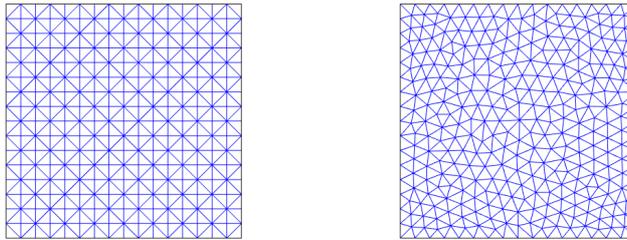


Figure 6: xy -meshes corresponding to two different distribution of points over the saddle: regular mesh (left) versus unstructured mesh (right).

It appears that for a regular mesh there is no significant difference, while an irregular one gives better results with RCG. This last remark orients towards the choice of the RCG method when dealing with the reconstruction of a complex shaped surface, since in this case an unstructured-type mesh is obviously expected.

As a third example the reconstruction through the RCG method of a

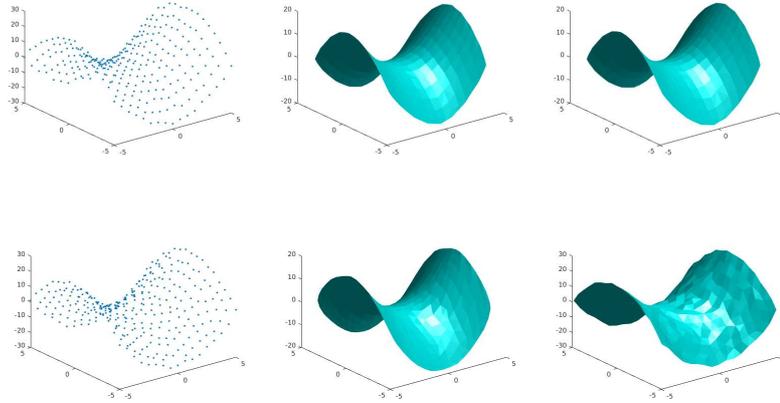


Figure 7: Second Example. Two xy -meshes corresponding to two different distribution of points over the saddle. The upper row corresponds to the regular mesh case, lower row to the unstructured one. The left column shows the surface points, central and right column the RCG and TIS results respectively.

complex shape is considered, as illustrated by Figure 8.

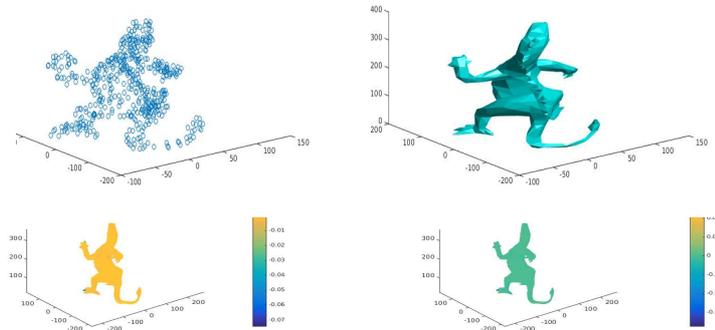


Figure 8: Third Example. A complex shape reconstructed through the RCG method. The cloud of points (upper left) is paralleled by the reconstructed surface (upper right). The lower row shows the Gaussian curvatures respectively: left related to the original cloud and right related to reconstructed surface.

Conclusions

We propose a local interpolation method for reconstructing a smooth surface from given vertex positions and associated normals. It uses parametric triangular Bézier patches in the form of rational blend cubic Gregory patches and achieves positional and tangential continuity between the patches by using free shape parameters. The minimization of some functionals linked to a particular metric on the reconstructed surface and depending on the free parameters allows to optimize the shape of the patches. The proposed method (RCG) is compared with a global method based on bivariate integral tensor spline with shape parameters (TIS). The better behavior of RCG is particularly evident when the point cloud comes from an unstructured mesh.

References

- [1] M. Boschioli, C. Fünzig, L. Romani, G. Albrecht, *G^1 rational blend interpolatory schemes: A comparative study*. Graphical Models 74, pages 29-49, 2012.
- [2] M. Boschioli, *Local Parametric Bézier Interpolants for Triangular Meshes: from Polynomial to Rational Schemes*. PhD thesis, University of Valenciennes, France, and Università Milano–Bicocca, Italy, 2011.

- [3] F. Calì, M. Miglio, *Constrained reconstruction of 3D curves and surfaces using integral spline operators*. CAIM 74, pages 29- 49, 2013.
- [4] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press Inc., Boston (1990).
- [5] G. Farin, *NURB curves and surfaces: from projective geometry to practical use*. A. K. Peters, Ltd., Natick, MA, USA, 1995.
- [6] S. Mann, C. Loop, M. Lounsbury, D. Meyers, J. Painter, T. DeRose, K. Sloan, *A survey of parametric scattered data fitting using triangular interpolants*, in: H. Hagen (Ed.), *Curve and Surface Design*, SIAM, 1992, pp. 145–172.
- [7] R.R. Martin, *Estimation of principal curvatures from range data*. Int. J. of Shape Modelling, vol. 4, pages 99-109,1998.
- [8] T. Rando and J.A. Roulier, *Designing faired parametric surfaces*. Computer Aided Design, vol.23, pages 492-497,1991.
- [9] D.J. Walton and D.S. Meek, *A triangular G^1 patch from boundary curves*. Computer Aided Design, vol. 28, no. 2, pages 113-123, 1996.
- [10] D.J. Walton and M. Yeung, *Geometric modelling from CT scans for stereolithography apparatus*. In Z. Tang, editor, *New Advances in CAD & Computer Graphics*, pages 417-422, 1993.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 18/2017** Ambartsumyan, I.; Khattatov, E.; Yotov, I.; Zunino, P.
A Lagrange multiplier method for a Stokes-Biot fluid-poroelastic structure interaction model
- 19/2017** Giovanardi, B.; Formaggia, L.; Scotti, A.; Zunino P.
Unfitted FEM for modelling the interaction of multiple fractures in a poroelastic medium
- 17/2017** Agosti, A.
Error Analysis of a finite element approximation of a degenerate Cahn-Hilliard equation
- 13/2017** Gigante, G.; Vergara, C.
Optimized Schwarz Methods for circular flat interfaces and geometric heterogeneous coupled problems
- 14/2017** Bruggi, M.; Parolini, N.; Regazzoni, F.; Verani, M.
Finite Element approximation of an evolutionary Topology Optimization problem
- 15/2017** Tagliabue, A.; Dede', L.; Quarteroni A.
Complex blood flow patterns in an idealized left ventricle: a numerical study
- 16/2017** Ghiglietti, A.; Scarale, M.G.; Miceli, R.; Ieva, F.; Mariani, L.; Gavazzi, C.; Paganoni, A.M.; E
Urn models for response-adaptive randomized designs: a simulation study based on a non-adaptive randomized trial
- 11/2017** Ferro, N.; Micheletti, S.; Perotto, S.
Anisotropic Mesh Adaptation for Crack Propagation Induced by a Thermal Shock
- 12/2017** Gasperoni, F.; Ieva, F.; Barbati, G.; Scagnetto, A.; Iorio, A.; Sinagra, G.; Di Lenarda, A.
Multi state modelling of heart failure care path: a population-based investigation from Italy
- 10/2017** Pini, A.; Stamm, A.; Vantini, S.
Hotelling's T^2 in separable Hilbert spaces