

THE GENERATION OF UNIT P^2 MESHES: ERROR ESTIMATION AND MESH ADAPTATION

Ruili Zhang Amaury Johnen Jean-François Remacle François Henrotte
Arthur Bawin

Université catholique de Louvain, Louvain-la-Neuve, Belgium, ruili.zhang@uclouvain.be
Université catholique de Louvain, Louvain-la-Neuve, Belgium, amaury.johnen@uclouvain.be
Université catholique de Louvain, Louvain-la-Neuve, Belgium, jean-francois.remacle@uclouvain.be
Université catholique de Louvain, Louvain-la-Neuve, Belgium, francois.henrotte@uclouvain.be
Université catholique de Louvain, Louvain-la-Neuve, Belgium, arthur.bawin@uclouvain.be

ABSTRACT

We propose a new framework for the generation and adaptation of unit curvilinear P^2 meshes in dimension 2. In this approach, curvature is not only used to match curved boundaries but also to capture features of the interpolated solutions, and it results in meshes that would not have been achievable by simply curving *a posteriori* a straight-sided mesh. We proceed as follows. Starting with a smooth function $f(x, y)$, a metric field, based on f and its derivatives up to order 3, is constructed. A unit P^2 mesh is then generated, with edges within an adimensional length range of $[0.7, 1.4]$ with respect to this metric field. Points are then spawned in such a way that their geodesic distance corresponds to edges of unit size, and these points are then connected in a standard isotropic fashion. A curvilinear mesh quality criterion is then proposed to drive the mesh optimization process. The triangulation is subsequently modified using straight-sided edge swap, straight-sided edge curving, curvilinear edge swap and Curvilinear Small Polygon Reconnection (CSPR) to form the desired unit mesh. A unit curvilinear mesh containing only valid “Geodesic Delaunay triangles” is obtained this way. A number of application examples are presented in order to demonstrate the capabilities of the mesh adaptation procedure. The resulting adapted meshes allow, most of the times, a significant reduction of the approximation error compared with straight-sided P^2 meshes of the same density.

Keywords: curvilinear mesh generation, mesh adaptation, Riemannian metric field, geodesic, analytic functions, high-order error, finite element method

1. INTRODUCTION

Scientific computing is now an old science. Solving partial differential equations on a computer is a very common task for aerospace/chemical/ mechanical/electrical engineers. Still, numerical methods for PDEs that have reached a production level such as finite elements are, for most of them, based on numerical schemes that are of the second order of accuracy. Some applications in fluid mechanics or in electromagnetic nonetheless require numerical schemes that are of higher order of accuracy (those schemes are some-

times called high fidelity schemes). It has been proved in many contributions that high-order finite element schemes require high-order meshes, i.e., meshes that capture the curvilinear features of the geometry with a high fidelity as well [1]. In the last decade, a significant part of the research in mesh generation has thus been devoted to the generation of body fitted curvilinear meshes. The main issue of generating curved meshes is that there exists for now no algorithm that actually generates a P^2 mesh in a direct fashion. State-of-the-art methods generate a straight-sided mesh and place high-order points on the CAD geometry. Then, in-

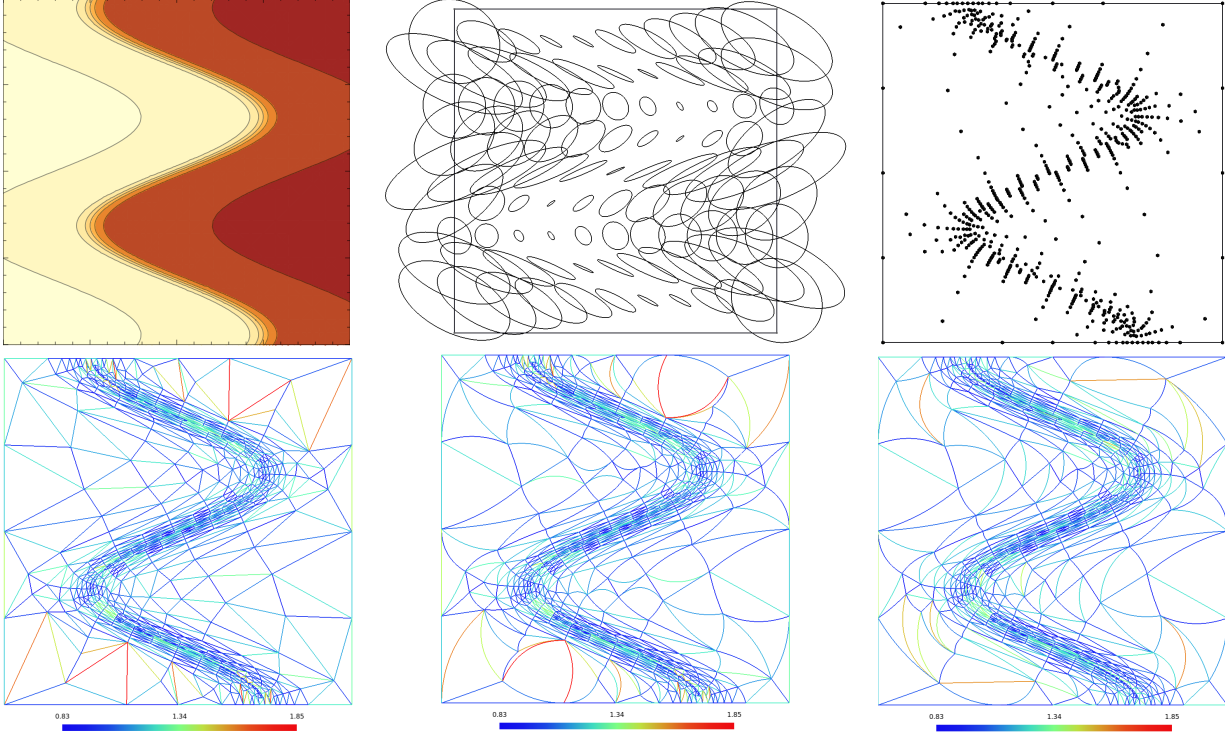


Figure 1: Illustration of the whole process of curvilinear mesh generation and adaptation based on a given analytic function. The top left image represents the function(19) on the unit square. The top center image depicts the metric field $\mathcal{M}(x)$ computed as explained in section 3. The top right image shows the point spawned according to this metric field. . The bottom left image is the straight-sided anisotropic mesh based on the spawned points. The bottom center image shows the initial curvilinear mesh, with the edges colored according to their adimensional lengths. Finally, the bottom right image is the adapted anisotropic mesh including the new Curvilinear Small Polygon Reconnection (CSPR) procedure.

valid elements are untangled using various approaches [2, 3, 4, 5, 6, 7, 8]. Nowadays, body fitted curvilinear meshes start to be used in an industrial context [9, 10].

High-order meshes have exclusively been used for increasing geometrical accuracy, i.e., to make the mesh represents the geometry of curved parts with high fidelity. The natural extension of the use of curvilinear meshes is *high-order/curvilinear mesh adaptation*. In the linear case, extensive work has been done in anisotropic mesh adaptation[11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. The concept of metric tensor is always central in anisotropic adaptation: it allows to define mesh sizes and directions that allow to minimize the interpolation error [22, 23, 24, 25, 26, 27]. Yet, all those methods end up with a straight-sided mesh.

This paper is in line with recent paper [28] which have provided an embryo of solution to the problem of curvilinear mesh adaptation. In [28], an analytical metric field was assumed and unit P^2 meshes were generated based on that metric. In the present paper, we essentially tackle two additional problems that al-

low to move forward to “true anisotropic curvilinear mesh adaptation” (i.e., optimizing a mesh based on a high-order finite element solution):

1. P^2 error estimates that are currently used in the literature are based on the implicit hypothesis that underlying meshes are straight-sided. Assuming a function $f(x, y) \in C^3$, we construct a metric field that does not assume mesh edges to be straight-sided.
2. The mesh generation procedure that we use for generating the curvilinear meshes is based on the one of [28]. Yet, we show that using edge swaps only does not always allow to reach a unit mesh. We propose a more general operator - curvilinear small polygon reconnection - that allows to reconnect points in a wider range and generate P^2 unit meshes very robustly.

The paper is structured as follows: in Section 2, a brief review of the interpolation error and algorithm that

compute geodesic parabolas is presented; Section 3 describes a new idea of the definition of the metric field that takes into account the curvilinear nature of the mesh; in Section 4, we give a simple illustrative example; in Section 5, we describe and detail the mesh generation approach. The whole process of curvilinear mesh adaptation will be explained based on a running example. All the stages of that process are illustrated in Figure 1. Interpolation error is analyzed at the end of the paper.

2. INTERPOLATION ERROR

2.1 A point of departure

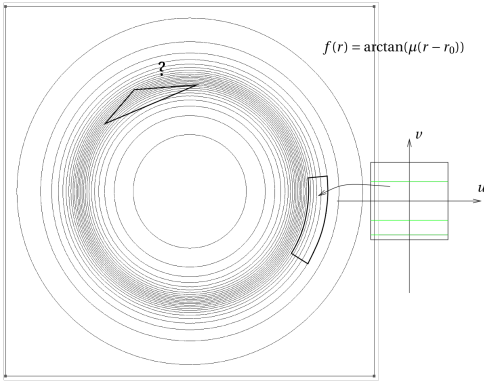


Figure 2: ideal elements

This section starts with a small reflection about a very interesting paper published in 2011 by Lorenzo Botti [29]. In his paper, Botti shows that, in a standard finite element context, curving a mesh may have a dramatic cost in terms of the quality of the finite element interpolation. When we first read this paper, we were already working in curvilinear meshing: we were thus quite puzzled by Botti's conclusions. Clearly, using P^2 meshes and P^2 finite elements (isoparametric finite elements) causes the interpolation to barely pass the patch test. Such an interpolation is not able to exactly represent a quadratic function in the \mathbf{x} plane which causes damages to the approximation properties of the element.

Now imagine a function $f(r, \theta)$ in polar coordinates that is, say, parabolic in r and θ . On the one hand, f is not polynomial in Euclidean coordinates. On the other hand, with a mesh whose edges are aligned with \mathbf{e}_r and \mathbf{e}_θ there would be no interpolation error at all with P^2 finite elements. In Fig. 2, the straight-sided anisotropic triangle does its best to align with the iso-lines of function f whereas the curvilinear quadrangle is able to align with the solution. With a same mesh size, a much lower interpolation error is expected with

this quadrangle than with the triangle. This is our starting point: optimal curvilinear edges should be adapted to match the local parametrization of f .

Let thus $f(\mathbf{x}) \in C^3$ be a three time differentiable function, with $\mathbf{x} = (x^1, x^2)$. Its derivatives up to order 3 are respectively its gradient

$$G_i = \frac{\partial f}{\partial x^i},$$

its hessian

$$H_{ij} = \frac{\partial G_i}{\partial x^j} = \frac{\partial^2 f}{\partial x^i \partial x^j},$$

and its third order derivative tensor

$$C_{ijk} = \frac{\partial H_{ij}}{\partial x^k} = \frac{\partial^3 f}{\partial x^i \partial x^j \partial x^k},$$

with $i, j, k = 1, 2$.

Based on these derivatives of $f(\mathbf{x})$, we present an approach to compute a metric field $\mathcal{M}(\mathbf{x})$ that takes into account the curvilinear nature of the P^2 mesh of $f(\mathbf{x})$.

2.2 Parabolic edges

This paper having the aim of building P^2 meshes, the mesh edges are going to be represented by parabolas.

In our approach, the set of parabolas that connect the points \mathbf{X}_1 and \mathbf{X}_2 is restricted to those with the midpoint \mathbf{X}_{12}

$$\mathbf{X}_{12} = \frac{\mathbf{X}_1 + \mathbf{X}_2}{2} + \alpha(\mathbf{X}_2 - \mathbf{X}_1) \times \mathbf{e}_3, \quad \alpha \in \mathbb{R}$$

located on the orthogonal bisector of the segment $\mathbf{X}_1\mathbf{X}_2$, as shown in Fig. 3.

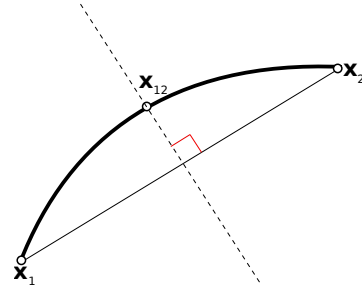


Figure 3: Midpoint \mathbf{X}_{12} of a parabola situated on the orthogonal bisector of the straight line $\mathbf{X}_1\mathbf{X}_2$.

The parametric equation of the parabola is then given by

$$\begin{aligned} \mathcal{C} &\equiv \mathbf{x}(t) \\ &= (1-t)(1-2t)\mathbf{X}_1 + t(2t-1)\mathbf{X}_2 + 4t(1-t)\mathbf{x}_{12}(\alpha) \\ &= \mathbf{X}_1 + t(\mathbf{X}_2 - \mathbf{X}_1) + 4t(1-t)\alpha(\mathbf{X}_2 - \mathbf{X}_1) \times \mathbf{e}_3 \\ &= \mathbf{X}_1 + th\mathbf{u} + 4th(1-t)\alpha\mathbf{b}, \end{aligned} \quad (1)$$

where $\mathbf{u} = (u^1, u^2)$ is the unit vector parallel to $\mathbf{X}_2 - \mathbf{X}_1$, $h = \|\mathbf{X}_2 - \mathbf{X}_1\|$ and $\mathbf{b} = \mathbf{u} \times \mathbf{e}_3$. Here, α is the only unknown coefficient that allows to move midpoint \mathbf{X}_{12} along \mathbf{b} and it is computed in such a way that the length of that parabola is minimized. This minimization is performed using a golden section algorithm.

2.3 Approximation error and mesh size

Assume a curve $\mathbf{x}(t)$, $t \in [0, 1]$, a function $f(\mathbf{x}(t))$, and its quadratic Lagrange interpolate $\pi^2 f(\mathbf{x}(t))$. On basis of a Taylor expansion, one knows that the interpolation error can be bounded as follows:

$$\max_{t \in [0,1]} |f(\mathbf{x}(t)) - \pi^2 f(\mathbf{x}(t))| \leq \frac{1}{6} \sup_{t \in [0,1]} \left| \frac{d^3 f(\mathbf{x}(t))}{dt^3} \right|. \quad (2)$$

When \mathcal{C} is a straight edge (i.e., $\alpha = 0$ in (1)), one has

$$\dot{\mathbf{x}}(t) = h\mathbf{u} \quad (3)$$

so that

$$\frac{d^3 f(\mathbf{x}(t))}{dt^3} = C_{ijk} \dot{x}^i \dot{x}^j \dot{x}^k = h^3 C_{ijk} u^i u^j u^k,$$

and (2) becomes

$$\max_{\mathbf{x} \in \mathcal{C}} |f(\mathbf{x}) - \pi^2 f(\mathbf{x})| \leq \frac{h^3}{6} \sup_{\mathbf{x} \in \mathcal{C}} \left| C_{ijk}(\mathbf{x}) u^i u^j u^k \right|, \quad (4)$$

where $C_{ijk} = C_{ijk}(\mathbf{x}(t))$, and repeated indices are implicitly summed over (Einstein summation).

Now, if \mathcal{C} is the parabola (1), we have

$$\dot{\mathbf{x}}(t) = h\mathbf{u} + \alpha h(4 - 8t)\mathbf{b}, \quad \ddot{\mathbf{x}}(t) = -8\alpha h\mathbf{b},$$

and it is easy to show that

$$\frac{d^3 f(\mathbf{x}(t))}{dt^3} = C_{ijk} \dot{x}^i \dot{x}^j \dot{x}^k + 3H_{ij} \dot{x}^i \ddot{x}^j \quad (5)$$

and

$$\max_{\mathbf{x} \in \mathcal{C}} |f(\mathbf{x}) - \pi^2 f(\mathbf{x})| \leq \frac{h^3}{6} \sup_{\mathbf{x} \in \mathcal{C}} \left| C_{ijk} \dot{x}^i \dot{x}^j \dot{x}^k + 3H_{ij} \dot{x}^i \ddot{x}^j \right|.$$

with

$$\begin{aligned} \frac{\partial f(\mathbf{x}(t))}{\partial t} &= \frac{\partial f(\mathbf{x}(t))}{\partial x_1(t)} \frac{\partial x_1(t)}{\partial t} + \frac{\partial f(\mathbf{x}(t))}{\partial x_2(t)} \frac{\partial x_2(t)}{\partial t} \\ &= \frac{\partial f(\mathbf{x}(t))}{\partial x_i(t)} \frac{\partial x_i(t)}{\partial t} \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 f(\mathbf{x}(t))}{\partial t^2} &= \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_1^2(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial x_1(t)}{\partial t} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_2^2(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial x_2(t)}{\partial t} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_1(t) \partial x_2(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial x_2(t)}{\partial t} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_2(t) \partial x_1(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial x_1(t)}{\partial t} \\ &+ \frac{\partial f(\mathbf{x}(t))}{\partial x_1(t)} \frac{\partial^2 x_1(t)}{\partial t^2} + \frac{\partial f(\mathbf{x}(t))}{\partial x_2(t)} \frac{\partial^2 x_2(t)}{\partial t^2} \\ &= \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_i(t) \partial x_j(t)} \frac{\partial x_i(t)}{\partial t} \frac{\partial x_j(t)}{\partial t} + \frac{\partial f(\mathbf{x}(t))}{\partial x_i(t)} \frac{\partial^2 x_i(t)}{\partial t^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial^3 f(\mathbf{x}(t))}{\partial t^3} &= \frac{\partial^3 f(\mathbf{x}(t))}{\partial x_1^3(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial x_1(t)}{\partial t} \frac{\partial x_1(t)}{\partial t} \\ &+ \frac{\partial^3 f(\mathbf{x}(t))}{\partial x_2^3(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial x_2(t)}{\partial t} \frac{\partial x_2(t)}{\partial t} \\ &+ \frac{\partial^3 f(\mathbf{x}(t))}{\partial x_1^2(t) \partial x_2(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial x_1(t)}{\partial t} \frac{\partial x_2(t)}{\partial t} \\ &+ \frac{\partial^3 f(\mathbf{x}(t))}{\partial x_2^2(t) \partial x_1(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial x_2(t)}{\partial t} \frac{\partial x_1(t)}{\partial t} \\ &+ 2 \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_1^2(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial^2 x_1(t)}{\partial t^2} \\ &+ 2 \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_2^2(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial^2 x_2(t)}{\partial t^2} \\ &+ 2 \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_1(t) \partial x_2(t)} \frac{\partial^2 x_1(t)}{\partial t^2} \frac{\partial x_2(t)}{\partial t} \\ &+ 2 \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_2(t) \partial x_1(t)} \frac{\partial^2 x_2(t)}{\partial t^2} \frac{\partial x_1(t)}{\partial t} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_1^2(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial^2 x_1(t)}{\partial t^2} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_1(t) \partial x_2(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial^2 x_1(t)}{\partial t^2} \\ &+ \frac{\partial f(\mathbf{x}(t))}{\partial x_1(t)} \frac{\partial^3 x_1(t)}{\partial t^3} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_2^2(t)} \frac{\partial x_2(t)}{\partial t} \frac{\partial^2 x_2(t)}{\partial t^2} \\ &+ \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_2(t) \partial x_1(t)} \frac{\partial x_1(t)}{\partial t} \frac{\partial^2 x_2(t)}{\partial t^2} \\ &+ \frac{\partial f(\mathbf{x}(t))}{\partial x_2(t)} \frac{\partial^3 x_2(t)}{\partial t^3} \\ &= \frac{\partial^3 f(\mathbf{x}(t))}{\partial x_i(t) \partial x_j(t) \partial x_k(t)} \frac{\partial x_i(t)}{\partial t} \frac{\partial x_j(t)}{\partial t} \frac{\partial x_k(t)}{\partial t} \\ &+ 3 \frac{\partial^2 f(\mathbf{x}(t))}{\partial x_i(t) \partial x_j(t)} \frac{\partial x_i(t)}{\partial t} \frac{\partial^2 x_j(t)}{\partial t^2} \\ &+ \frac{\partial f(\mathbf{x}(t))}{\partial x_i(t)} \frac{\partial^3 x_i(t)}{\partial t^3} \end{aligned}$$

Equation (5) shows that, even if $f(\mathbf{x})$ is only quadratic in \mathbf{x} , i.e., $C_{ijk} = 0$, the finite element approximation error of $f(\mathbf{x}(t))$ with quadratic isoparametric elements does not vanish, due to the curvature of the curve \mathcal{C} . This was the observation of Botti in [29] - say, in a standard finite element context, curving a mesh may have a dramatic cost in terms of the quality of the finite element interpolation.

A relationship similar to (3) is needed for the parabolic case, and it is convenient to derive it to reparametrize the curve $\mathbf{x}(t)$ by arc length, i.e., with the arc length

$$s(t) = \int_0^t |\dot{\mathbf{x}}(u)| du \quad (6)$$

as parameter. One has then

$$\partial_s \mathbf{x}(s) = \mathbf{g} + s\kappa \mathbf{e}_3 \times \mathbf{g},$$

where \mathbf{g} is the unit vector tangent to the curve, and κ the geodesic curvature, using the basis vectors of a Darboux frame [30]. One has now the relationship equivalent to (3) for the parabola,

$$\begin{aligned} \ddot{\mathbf{x}}(t=0) &= \partial_s \mathbf{x}(s=0) \partial_t s(t=0) \\ &= \mathbf{g} |\dot{\mathbf{x}}(0)| = \mathcal{L} \mathbf{g} \end{aligned}$$

with

$$\mathcal{L} = h\sqrt{1 + 16\alpha^2}.$$

Moreover, one can show that

$$(\mathbb{I} - \mathbf{g}\mathbf{g}^T)\ddot{\mathbf{x}}(t=0) = \kappa(\mathbf{e}_3 \times \mathbf{g})\mathcal{L}^2.$$

Assuming that $(\mathbb{I} - \mathbf{g}\mathbf{g}^T)\ddot{\mathbf{x}}(t=0)$ is not too different from $\ddot{\mathbf{x}}(t=0)$, and that $C_{ijk}(\mathbf{x})$ and $H_{ij}(\mathbf{x})$ do not vary too much over \mathcal{C} , one can write

$$\begin{aligned} \max_{\mathbf{x} \in \mathcal{C}} |f(\mathbf{x}) - \pi^2 f(\mathbf{x})| &\leq \\ \frac{\mathcal{L}^3}{6} |C_{ijk}(\mathbf{p}) g^i g^j g^k| + \frac{\mathcal{L}^3}{2} |H_{ij}(\mathbf{p}) g^i \kappa(\mathbf{e}_3 \times \mathbf{g})^j| \end{aligned}$$

although doing so may be too conservative in the sense that this estimate assumes that errors due to polynomial approximation and geometry add to each other *while they may actually balance each other*. We thus stick to

$$\begin{aligned} \max_{\mathbf{x} \in \mathcal{C}} |f(\mathbf{x}) - \pi^2 f(\mathbf{x})| & \quad (7) \\ &\leq \frac{\mathcal{L}^3}{6} \sup_{t \in [0,1]} |C_{ijk}(\mathbf{p}) g^i g^j g^k + 3H_{ij}(\mathbf{p}) g^i \kappa(\mathbf{e}_3 \times \mathbf{g})^j|. \end{aligned}$$

If we pose

$$E = |C_{ijk}(\mathbf{p}) g^i g^j g^k + 3H_{ij}(\mathbf{p}) g^i \kappa(\mathbf{e}_3 \times \mathbf{g})^j|, \quad (8)$$

the interpolation error is thus of the form $\epsilon \simeq \mathcal{L}^3/6 E$, and since the goal is to adapt the meshsize in order to have an error equidistribution ϵ among edges, we choose

$$\mathcal{L} = (6\epsilon/E)^{1/3}. \quad (9)$$

3. CONSTRUCTION OF THE METRIC FIELD $\mathcal{M}(\mathbf{x})$

A classical technique in anisotropic mesh generation consists in defining an auxiliary metric field $\mathcal{M}(\mathbf{x})$ under which the sought anisotropic mesh is a unit mesh, i.e., a mesh with edges of approximately unit length. This metric field is defined by two orthogonal unit vectors \mathbf{g}_1 and \mathbf{g}_2 and two mesh sizes h_1 and h_2 at every point \mathbf{p} of the domain. It is customary to choose the vectors \mathbf{g}_1 and \mathbf{g}_2 as the eigenvectors of the Hessian H_{ij} , so that one has

$$\mathcal{M}(\mathbf{x}) = (\mathbf{g}_1 \ \mathbf{g}_2) \begin{pmatrix} \frac{1}{h_1^2} & 0 \\ 0 & \frac{1}{h_2^2} \end{pmatrix} (\mathbf{g}_1 \ \mathbf{g}_2)^T. \quad (10)$$

This arbitrary choice has two principal virtues:

1. It allows a maximum amount of anisotropy in the P^1 case.
2. If the function f is C^2 , \mathbf{g}_1 and \mathbf{g}_2 are continuous and mesh orientation varies smoothly.

The meshsizes h_1 and h_2 are calculated so as to maintain the interpolation error below a prescribed target error ϵ . We shall try to keep up with these properties in the context of curvilinear meshes.

The main idea is to assume that the edges of the anisotropic mesh at point \mathbf{p} are oriented along either the iso-contour of f or the direction of the gradient ∇f , i.e., along the curves defined as follows:

1. Curve $\mathcal{C}_1 \equiv \mathbf{x}_1(t)$ is the quadratic approximation of the iso-contour $f(\mathbf{x}_1(t)) = f(\mathbf{p})$.
2. Curve $\mathcal{C}_2 \equiv \mathbf{x}_2(t)$ is the quadratic approximation of the local downhill gradient going through \mathbf{p} .

Here again, it is convenient to work with curves parametrized by arc-length. Let

$$\mathcal{C}_1 \equiv \mathbf{x}_1(s) = \mathbf{p} + \mathbf{g}_1 s + \kappa_1 \mathbf{g}_2 \frac{s^2}{2} \quad (11)$$

and

$$\mathcal{C}_2 \equiv \mathbf{x}_2(s) = \mathbf{p} + \mathbf{g}_2 s + \kappa_2 \mathbf{g}_1 \frac{s^2}{2}. \quad (12)$$

with, as explained above, the unit tangent vectors

$$\begin{aligned} \mathbf{g}_1 &= \frac{\nabla^\perp f|_{\mathbf{p}}}{\|\nabla^\perp f|_{\mathbf{p}}\|} = \frac{1}{(f_{x_1}^2 + f_{x_2}^2)^{1/2}} \begin{pmatrix} -f_{x_2} \\ f_{x_1} \end{pmatrix}, \\ \mathbf{g}_2 &= \frac{\nabla f|_{\mathbf{p}}}{\|\nabla f|_{\mathbf{p}}\|} = \frac{1}{(f_{x_1}^2 + f_{x_2}^2)^{1/2}} \begin{pmatrix} f_{x_1} \\ f_{x_2} \end{pmatrix} \end{aligned}$$

where the shorthand notations

$$f_{x^1} = \frac{\partial f}{\partial x^1}(\mathbf{p}) \quad , \quad f_{x^2} = \frac{\partial f}{\partial x^2}(\mathbf{p})$$

have been used.

Taylor expansion limited to order 2 of $f(\mathbf{x})$ around \mathbf{p} writes

$$f(\mathbf{x}) \simeq f(\mathbf{p}) + \nabla f|_{\mathbf{p}}(\mathbf{x} - \mathbf{p}) + \frac{1}{2}(\mathbf{x} - \mathbf{p})^T H|_{\mathbf{p}}(\mathbf{x} - \mathbf{p}). \quad (13)$$

If the curve $\mathbf{x}_1(s)$ runs along the isovalue of f , one can write $f(\mathbf{x}_1(s)) = f(\mathbf{p})$ and, using (11), one can write

$$\mathbf{x} - \mathbf{p} \equiv \mathbf{x}_1(s) - \mathbf{p} = \mathbf{g}_1 s + \kappa_1 \mathbf{g}_2 \frac{s^2}{2}$$

to be inserted into (13) to give

$$\underbrace{\nabla f|_{\mathbf{p}} \cdot \mathbf{g}_1}_{=0} s + \left[\kappa_1 \nabla f|_{\mathbf{p}} \cdot \mathbf{g}_2 + \mathbf{g}_1^T H|_{\mathbf{p}} \mathbf{g}_1 \right] \frac{s^2}{2} + \mathcal{O}(s^3) = 0$$

Since the equation is true in a neighborhood of $s = 0$, the quantity between bracket must vanish, and the identity

$$\kappa_1 \nabla f|_{\mathbf{p}} \cdot \mathbf{g}_2 + \mathbf{g}_1^T H|_{\mathbf{p}} \mathbf{g}_1 = 0$$

gives the expression of the curvature κ_1 needed to finalize the identification of the curve (11)

$$\begin{aligned} \kappa_1 &= -\frac{\mathbf{g}_1^T H|_{\mathbf{p}} \mathbf{g}_1}{\nabla f|_{\mathbf{p}} \cdot \mathbf{g}_2} \\ &= \frac{-f_{x_2}^2 f_{x_1 x_1} + 2f_{x_1} f_{x_2} f_{x_1 x_2} - f_{x_1}^2 f_{x_2 x_2}}{(f_{x_1}^2 + f_{x_2}^2)^{3/2}}. \end{aligned}$$

Next, we turn to the second curve (12), which is to be aligned with $\nabla f(\mathbf{p})$: A Taylor expansion around \mathbf{p} of the gradient this time gives

$$\nabla f(\mathbf{x}) \simeq \nabla f|_{\mathbf{p}} + H|_{\mathbf{p}}(\mathbf{x} - \mathbf{p}) \quad (14)$$

with, using (12),

$$\mathbf{x} - \mathbf{p} \equiv \mathbf{x}_2(s) - \mathbf{p} = \mathbf{g}_2 s + \kappa_2 \mathbf{g}_1 \frac{s^2}{2}. \quad (15)$$

On the other hand, one has

$$\partial_s \mathbf{x}_2(s) = \mathbf{g}_2 + \kappa_2 \mathbf{g}_1 s \quad (16)$$

and the alignment of the curve with the gradient reads

$$\nabla f(\mathbf{x}_2(s)) \times \partial_s \mathbf{x}_2(s) = 0, \quad (17)$$

where \times is the vector product. Substituting (14), (15) and (16) into (17) yields

$$\underbrace{\nabla f|_{\mathbf{p}} \times \mathbf{g}_2}_{=0} + [\kappa_2 \nabla f|_{\mathbf{p}} \times \mathbf{g}_1 + (H|_{\mathbf{p}} \mathbf{g}_2) \times \mathbf{g}_2] s + \mathcal{O}(s^2) = 0.$$

Again, as this equation is true in a neighborhood of $s = 0$, the quantity between bracket must vanish. Noting that, for any vector \mathbf{V} , one has $\mathbf{V} \times \mathbf{g}_2 = \mathbf{V} \cdot \mathbf{g}_1$ and

$\mathbf{V} \times \mathbf{g}_1 = -\mathbf{V} \cdot \mathbf{g}_2$, one obtains for the curvature κ_2 needed to finalize the identification of the curve (12) the expression

$$\begin{aligned} \kappa_2 &= \frac{\mathbf{g}_1^T H|_{\mathbf{p}} \mathbf{g}_2}{(f_{x_1}^2 + f_{x_2}^2)^{1/2} \mathbf{g}_2 \cdot \mathbf{g}_2} \\ &= \frac{f_{x_1} f_{x_2} (f_{x_2 x_2} - f_{x_1 x_1}) + (f_{x_1}^2 - f_{x_2}^2) f_{x_1 x_2}}{(f_{x_1}^2 + f_{x_2}^2)^{3/2}}. \end{aligned}$$

Using (8), we can thus finally write

$$E_{1,2} = \left| C_{ijk}(\mathbf{p}) g_{1,2}^i g_{1,2}^j g_{1,2}^k + 3\kappa_{1,2} H_{ij}(\mathbf{p}) g_{1,2}^i g_{2,1}^j \right|.$$

Using (9), the mesh sizes are finally defined by

$$h_{1,2} = \mathcal{L}_{1,2} = (6\epsilon/E_{1,2})^{1/3}. \quad (18)$$

Two remarks need to be made regarding the error estimate that has just been proposed. At first, some modification is clearly needed whenever $f_{x_1} = 0$ or $f_{x_2} = 0$, i.e., when the function is locally constant. In order to define \mathcal{M} everywhere in the domain, orthogonal directions \mathbf{g}_1 and \mathbf{g}_2 are computed everywhere where ∇f does not vanish. Then, well-defined directions are extended by a smoother borrowed from our cross field solver [31]. When the function f is constant, mesh sizes are limited to a user defined maximal size h_{max} .

Finally, we can note that the orthogonal directions \mathbf{g}_1 and \mathbf{g}_2 that has been chosen here are somewhat arbitrary. There might be other choices that have some advantages over the eigenvectors of H . Yet, the analytical example described in §4 shows that our choice makes sense.

4. A SIMPLE ILLUSTRATIVE EXAMPLE

Assume function $f(x^1, x^2) = (x^1)^2 + (x^2)^2$ and P^1 interpolation on triangles, on the standard context of straight-sided mesh adaptation, we clearly see f as an isotropic function, and its hessian being

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

In the context of standard P^1 adaptation, the optimal mesh is isotropic and its size for having an interpolation error of ϵ^2 is $\mathcal{L} = \epsilon$.

Now, the same function, which is isotropic in Euclidian coordinates (x^1, x^2) , is actually anisotropic in polar coordinates. There, $f(r, \theta) = r^2$ is independent of θ . If the mesh ‘‘mimics’’ polar coordinates, we could potentially have a significant gain by going anisotropic.

Now, let us allow the use of quadratic triangles and consider point $(x^1, x^2) = (a, 0)$, we have

$$\kappa_1 = -\frac{1}{\sqrt{(x^1)^2 + (x^2)^2}} \quad \text{and} \quad \kappa_2 = 0$$

and then

$$\mathbf{x}_1(t) = \begin{pmatrix} a - \frac{t^2}{2a} \\ t \end{pmatrix} \quad \text{and} \quad \mathbf{x}_2(t) = \begin{pmatrix} a + t \\ 0 \end{pmatrix}.$$

Along $\mathbf{x}_1(t)$, the function

$$f(\mathbf{x}_1(t)) = t^2 + \left(a - \frac{t^2}{2a}\right)^2 = a^2 + \frac{t^4}{4a^2},$$

is a constant up to order $\mathcal{O}(t^4)$. Assuming a linear interpolation of f , i.e., $\pi^1 f(\mathbf{x}_1(t)) = f(0) + f'(0)t = a^2$, the exact error is bounded by

$$|f(\mathbf{x}_1(t)) - \pi^1 f(\mathbf{x}_1(t))| = \left| \frac{t^4}{4a^2} \right| < \epsilon^2$$

so that $|t| < \sqrt{2\epsilon a} = \mathcal{L}_1$.

Along the perpendicular curve $\mathbf{x}_2(t)$, we have $f(\mathbf{x}_2(t)) = (a + t)^2$ and

$$\begin{aligned} |f(\mathbf{x}_2(t)) - \pi^1 f(\mathbf{x}_2(t))| &= |(a + t)^2 - a^2 - 2at| \\ &= |t^2| < \epsilon^2 \end{aligned}$$

which is independent of a . Thus, the optimal mesh has a constant mesh size $\mathcal{L}_2 = \epsilon$. The optimal mesh is thus anisotropic with an anisotropic scale factor equal to

$$\frac{\mathcal{L}_1}{\mathcal{L}_2} = \sqrt{\frac{2a}{\epsilon}}.$$

So, far away from the origin ($a \gg 0$), the optimal mesh is highly anisotropic.

Now, whenever the exact solution is unknown, but we can still estimate the derivatives of the numerical approximation, an estimator similar to the one explained in the previous section can be used. We have

$$\frac{d^2 f(\mathbf{x}_2(t))}{dt^2} = H_{ij} \dot{x}_2^i \dot{x}_2^j + 2G_i \ddot{x}_2^i = 2.$$

Our proposed estimate leads to

$$|f(\mathbf{x}_2(t)) - \pi^1 f(\mathbf{x}_2(t))| \simeq \frac{\mathcal{L}_2^2}{2} \left| \frac{d^2 f(\mathbf{x}_2(0))}{dt^2} \right| = \mathcal{L}_2^2 < \epsilon^2$$

which yields $\mathcal{L}_2 < \epsilon$.

The second order derivative along $\mathbf{x}_1(t)$, on the other hand, is computed as

$$\frac{d^2 f(\mathbf{x}_1(t))}{dt^2} = \underbrace{H_{ij} \dot{x}_1^i \dot{x}_1^j}_{2\left(\frac{t}{a}\right)^2 + 2} + \underbrace{2G_i \ddot{x}_1^i}_{-2 + \left(\frac{t}{a}\right)^2} = 3 \left(\frac{t}{a}\right)^2$$

and vanishes for $t = 0$. This indicates that our estimate allows choosing \mathcal{L}_1 arbitrarily large, which should come as no surprise, since in this case, the ‘‘true’’ anisotropic ratio is large, and probably beyond the capabilities of the mesh generator in terms of anisotropic elements.

5. MESH ADAPTATION

This section describes our mesh generation and adaptation approach. For illustrating the different steps of the procedure, the following analytic function

$$f(x) = \arctan(10(\sin(3\pi y/2) - 2x)) \quad (19)$$

will serve as running example. Figure 1 shows the general behavior of $f(x)$ on the unit square as well as the different steps of the procedure.

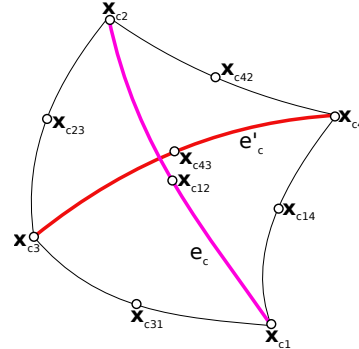


Figure 4: Curvilinear edge swap.

5.1 Generation of corner points

Our meshing approach is not the usual one, for which points and triangles are generated at the same time. In our approach, points are generated first, and then connected in a second step. Points are generated using the frontal algorithm described in [32, 33]. In short, one proceeds as follows. The points of the domain boundary are inserted in a queue. Then, the point at the end of the queue is popped out, and 4 neighbor points are created, located at unit distance from it along the parabolas $\pm \mathbf{x}_1$ and $\pm \mathbf{x}_2$, provided they are not too close to already existing points. This algorithm ensures thus that (i) two points \mathbf{x}_i and \mathbf{x}_j are never too close to each other and (ii) that there exist four points \mathbf{x}_{ij} ($j = 1, 2, 3, 4$) at unit distance from each point, i.e., that can form with it edges whose length is in the range $[0.7, 1.4]$. Figure 1 shows the points generated with a metric field based on a target error $\epsilon = 0.02$.

5.2 Generation of a straight-sided anisotropic mesh

The generated points are then connected together using a standard anisotropic mesh generator based on the metric field $\mathcal{M}(\mathbf{x})$ [34]. Figure 1 shows the resulting straight-sided mesh. The connectivity of this mesh is however not optimal. Yet, it constitutes a

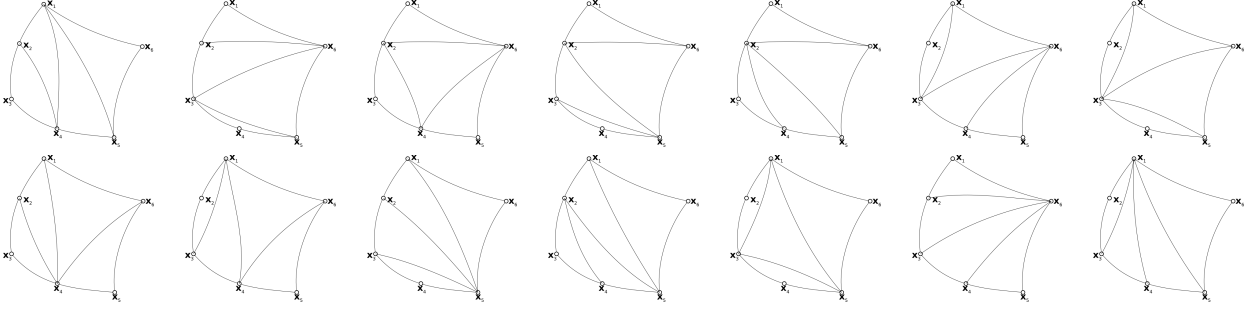


Figure 5: All curvilinear triangulations of a 4-cavity. There are 20 distinct triangles $\{0, 1, 2\}$, $\{0, 1, 3\}$, $\{0, 2, 3\}$, $\{1, 2, 3\}$, $\{0, 1, 4\}$, $\{0, 2, 4\}$, $\{1, 2, 4\}$, $\{0, 3, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$, $\{0, 1, 5\}$, $\{0, 2, 5\}$, $\{1, 2, 5\}$, $\{0, 3, 5\}$, $\{1, 3, 5\}$, $\{2, 3, 5\}$, $\{0, 4, 5\}$, $\{1, 4, 5\}$, $\{2, 4, 5\}$, $\{3, 4, 5\}$. in the $T_4 = 14$ triangulations shown in the Figure.

good starting point for constructing the unit curvilinear mesh that will further reduce the interpolation error.

5.3 Curving the straight sided mesh

The straight edges of the mesh are now to be transformed into parabolas. This operation is however endowed with the risk of creating invalid P^2 triangles. The following backtracking procedure allows however to provide a provably valid P^2 mesh. At first, invalid triangles are identified using the simple and robust validity criterion described in [35]. Since straight-sided triangles are always valid, the 3 mid-edge points of a given invalid triangle T are moved simultaneously backwards towards the mid points of the straight edges, until the triangle becomes valid again. All triangles sharing an edge with T are then checked, and if needed added to the list of invalid triangles. This algorithm always terminates, and as for limit case the recovery of a straight-sided mesh. Yet, in general, mild modifications of the initial curvilinear mesh are sufficient to restore a valid mesh.

It is very important at this point to ensure that one has a valid curvilinear mesh, since all subsequent optimisation operations will improve the mesh quality, and thus per definition always preserve the validity of the mesh.

The first mesh improvement method is a basic curvilinear edge swap (see Figure 4). Assume two curvilinear triangles $T_{c1}(\mathbf{x}_{c1}, \mathbf{x}_{c2}, \mathbf{x}_{c3})$ and $T_{c2}(\mathbf{x}_{c1}, \mathbf{x}_{c4}, \mathbf{x}_{c2})$ sharing a common edge e_c . Let e'_c be the geodesic between \mathbf{x}_{c3} and \mathbf{x}_{c4} . The curvilinear edge swap operator evaluates the opportunity of replacing edge e_c by edge e'_c . Two indicators decide whether the edge swap should be performed:

1. The new curvilinear triangles $T'_{c1}(\mathbf{x}_{c1}, \mathbf{x}_{c4}, \mathbf{x}_{c3})$ and $T'_{c2}(\mathbf{x}_{c4}, \mathbf{x}_{c2}, \mathbf{x}_{c3})$ have to be both valid, ac-

ording to the criterion based on robust estimations developed in [35].

2. The quality of the mesh has to be improved by the edge swap:

$$\begin{aligned} & \min(q_{ct'_{c1}}(\mathcal{M}(\mathbf{x})), q_{ct'_{c2}}(\mathcal{M}(\mathbf{x}))) \\ & > \min(q_{ct_{c1}}(\mathcal{M}(\mathbf{x})), q_{ct_{c2}}(\mathcal{M}(\mathbf{x}))) \end{aligned}$$

where $q_{ct}(\mathcal{M}(\mathbf{x}))$ is the curvilinear quality measure of triangle T_c with respect to metric field $\mathcal{M}(\mathbf{x})$.

The quality measure used here is a straightforward extension of the standard quality measure defined in [36]. We define:

$$q_{ct}(\mathcal{M}(\mathbf{x})) = \frac{4\sqrt{3}}{\mathcal{L}(\mathcal{M}(\mathbf{x}))} \int_{t_c} \sqrt{|\mathcal{M}(\mathbf{x})|} d\mathbf{x} \quad (20)$$

with

$$\mathcal{L}(\mathcal{M}(\mathbf{x})) = \mathcal{L}_{e_{c1}}^2(\mathcal{M}(\mathbf{x})) + \mathcal{L}_{e_{c2}}^2(\mathcal{M}(\mathbf{x})) + \mathcal{L}_{e_{c3}}^2(\mathcal{M}(\mathbf{x})),$$

where e_{c1} , e_{c2} and e_{c3} are the curvilinear edges of T_c , and $\mathcal{L}_{e_{c1}}$, $\mathcal{L}_{e_{c2}}$ and $\mathcal{L}_{e_{c3}}$ are their geodesic lengths according to the metric field $\mathcal{M}(\mathbf{x})$.

Note that triangle inequality is not necessary verified in Riemannian metrics, i.e., the inequality $\mathcal{L}_{e_{c1}} < \mathcal{L}_{e_{c2}} + \mathcal{L}_{e_{c3}}$ does not always hold. In consequence, the quality measure $q_{ct}(\mathcal{M}(\mathbf{x}))$ may be larger than one. Edges are swapped until a stable configuration is found. Here, we generalize Delaunay triangulation to Geodesic Delaunay triangulation [33].

5.4 Curvilinear Small Polygon Reconnection

The mesh curving procedure explained in the previous section 5.3 is very similar to the one proposed in [28].

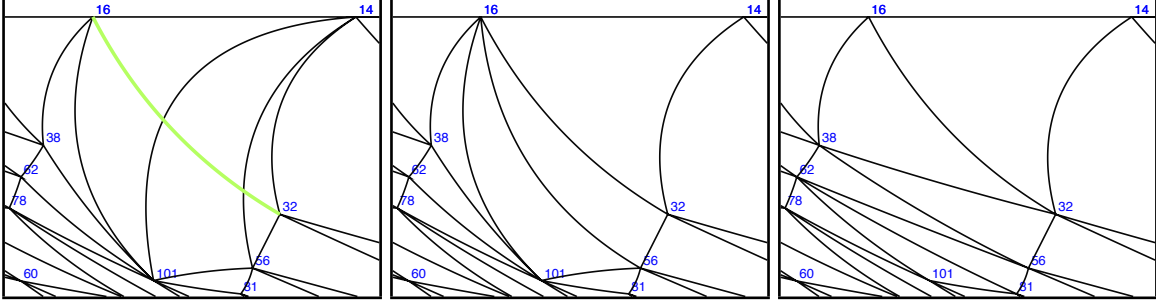


Figure 6: Mesh quality improved by CSPPR(Curvilinear Small Polygon Reconnection).

In this paper, we pointed out that this operation never produces short edges by construction. Figure 1 shows the curvilinear mesh generated through basic curved swaps, and one can indeed check that no short edges has been generated. The shortest edge has an adimensional length of $0.837 > 0.7$. Yet long edges exist, with a maximum edge length of $1.82 > 1.4$. Long edges may remain in the mesh, due to the inability of the basic edges swap process to properly connect points that are close with respect to the metric \mathcal{M} without generating invalid P^2 triangles.

This issue is fixed in this paper by introducing a new local mesh optimisation operator, called Curvilinear Small Polygon Reconnection (CSPPR), that allows overriding local quality maxima to further enhance the overall quality of the P^2 mesh.

The CSPPR is the curvilinear version of the small polygon reconnection (SPR) technique, a local mesh modification operator initially proposed by [37]. Considering a n -cavity, i.e., a set of n contiguous triangles with no internal vertex and with $n + 2$ boundary vertices, the SPR algorithm finds the best triangulation of the n -cavity among all possible triangulations. Catalan numbers $T_n = \frac{1}{n+1} \binom{2n}{n}$ give the number of possible triangulations of a n -cavity, which can be all found using a branch and bound algorithm [38]. In our work, all triangulations of n -cavities with $n < 10$ have been tabulated, so as to avoid any on-the-fly combinatorial computations of triangulations. Figure 5 shows all triangulations of a 4-cavity and the corresponding curved mesh with respect to the running test case metric.

The CSPPR requires cavities, which are obtained in two ways:

1. Along unit geodesics: consider all unit geodesic connecting the points of the mesh. Whenever those geodesics are not in the mesh, form cavity with all the triangles that intersect that geodesic.
2. Choose a long edge ($\mathcal{L} > 1.4$) and form cavity

with its two adjacent curvilinear triangles. If there are other long edges in those triangles, repeat the process.

In the process, if an internal edge has the same two points as before, we will keep the midpoint as before, otherwise, we compute a new midpoint; for all boundary edges, we always keep the same midpoint as before. For each possible swapped configuration, if the worst quality of all the elements is improved, the configuration is kept and will be in the new mesh unless another swapped configuration provides a better quality improvement. It works as:

- i. The new curvilinear triangles $t'_{c_1}(\mathbf{x}_{c_1}, \mathbf{x}_{c_2}, \mathbf{x}_{c_3})$, $t'_{c_2}(\mathbf{x}_{c_2}, \mathbf{x}_{c_3}, \mathbf{x}_{c_4})$, \dots , $t'_{c_n}(\mathbf{x}_{c_n}, \mathbf{x}_{c_{n+1}}, \mathbf{x}_{c_{n+2}})$ have to be all valid. The validity criterion that is used is the same one for curvilinear edges swap in Section 5.3.
- ii. The quality of the mesh has to be improved by the swapping:
$$\min(q_{ct'_{c_1}}(\mathcal{M}(\mathbf{x})), q_{ct'_{c_2}}(\mathcal{M}(\mathbf{x})), \dots, q_{ct'_{c_n}}(\mathcal{M}(\mathbf{x}))) > \min(q_{ct_{c_1}}(\mathcal{M}(\mathbf{x})), q_{ct_{c_2}}(\mathcal{M}(\mathbf{x})), \dots, q_{ct_{c_n}}(\mathcal{M}(\mathbf{x})))$$

where $q_{ct}(\mathcal{M}(\mathbf{x}))$ is the curvilinear quality measure of triangle t_c with respect to metric field $\mathcal{M}(\mathbf{x})$ - the same one for curvilinear edges swap in Section 5.3.

Figure 6 illustrates the CSPPR applied to the running example. A first 3-cavity is constructed with all elements crossing the green geodesic between points 16 and 32. This cavity is remeshed using CSPPR, producing a mesh that is better, but still not optimal. Indeed, a shorter curvilinear edge still exists between points 38 and 32 that is not in the mesh. A 5-cavity is then constructed around this edge, and remeshed to eventually produce an optimal mesh.

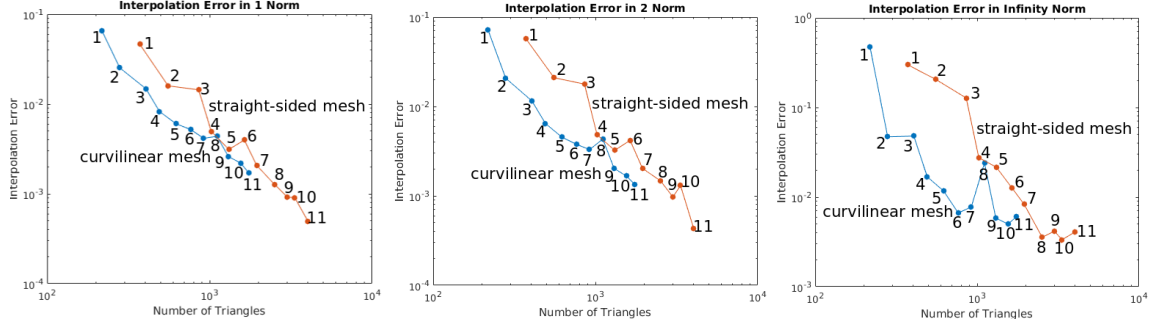


Figure 7: 1-norm, 2-norm, ∞ -norm interpolation error of analytic function $f(x)$ (19), the points 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11 respectively correspond to mesh size uniform scaling factors $a = 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2, 1.3, 1.4, 1.5$. The mesh size \mathcal{L}_a with a is $\mathcal{L}_a = \mathcal{L}/a^2$ and \mathcal{L} be the mesh size computed by (9).

6. INTERPOLATION ERROR

We complete our analysis by running our algorithm using the same function (19), but now defined on the geometrically non-trivial domain depicted in Figure 8, and by analyzing whether or not the curvilinear mesh adaptation reduces the interpolation error.

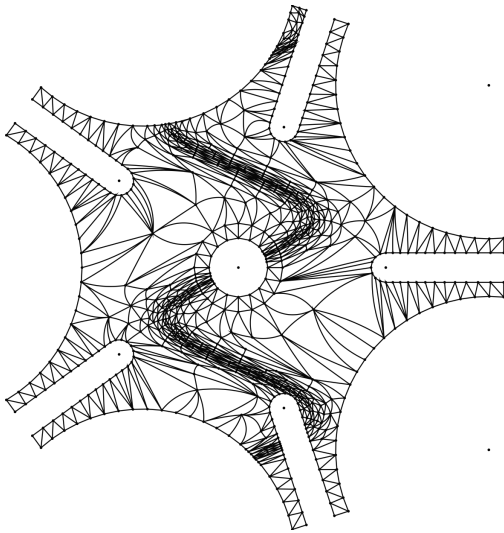


Figure 8: Curvilinear mesh adaptation based on function (19) on a mechanical part.

In order to compare the level of accuracy reached by meshes generated with our method or by meshes generated with a conventional straight-sided anisotropic mesh adaptation, the ∞ -norm, the 1-norm and the 2-norm of the quadratic interpolation error are computed with both meshes, and plotted against the number of triangles of the mesh. Eleven different meshes were generated, by varying a global mesh size uniform scaling parameter a between 0.4 and 1.5. The conver-

gence curves are shown in Figure 7. The numerical results show that our adapted curvilinear meshes allow a significant error reduction of about 50% with respect to straight-sided P^2 meshes of the same size. This mainly attributes to the numerical solutions of middle points of curvilinear meshes is more exact than numerical solutions of middle points of straight-sided P^2 meshes. Figure 9 shows that the case $a = 1.2$ (8th point) is an outlier. As shown in Figure 9, this is due to the persistence of long edges that the CSPR has not been able to remove.

7. CONCLUSIONS

Building on the paper [28], which provided an embryo of solution to the problem of curvilinear mesh adaptation, we have extended our work to the true P^2 mesh adaptation, i.e., the adaptation of a P^2 mesh to a given function $f(x, y)$. A methodology for building a specific metric field suited for this P^2 mesh adaptation has been explained, and a new local mesh modification operator called Curvilinear Small Polygon Reconnection (CSPR) has been developed to build the optimal curvilinear triangulation of a given curvilinear polygon/cavity. This paper focuses, for didactical reasons, on a single running example function (19), but the methodology has been applied successfully to other functions, with similar conclusions. The method is however not yet able to build a unit mesh in all cases. Figure 9 shows that, long edges may sometime fail to be eliminated by the CSPR, leaving an eventually interpolation error that is not better than a P^1 meshes. Yet, in most of cases, a clear reduction of the interpolation error is observed when curving the elements.

For the failure of the method to build a unit mesh in all cases, it maybe attribute to: the adimensional length used to insert points is 1.0 and the adimensional length used to check if two points are too close is 0.7, this may result the geodesic between two points

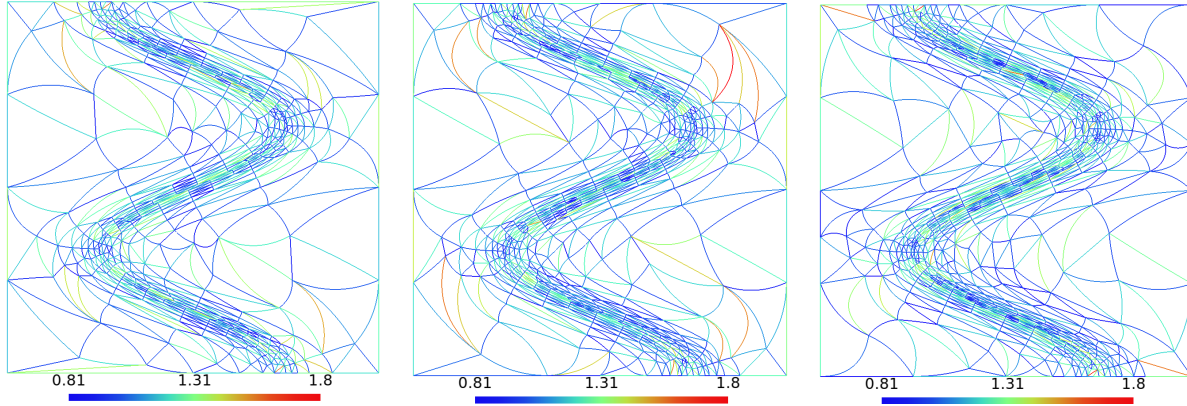


Figure 9: Adapted curvilinear meshes for computing interpolation error 1-norm, 2-norm, ∞ -norm at points 7-left, 8-middle, and 9-right (the same points 7, 8, and 9 in Figure 7 of curvilinear mesh).

is about 1.7, and this maybe be fixed by change the adimensional length used to insert points to between 1.0 and 0.7.

Our next move will be to replace the analytical functions with high-order finite element solutions. We foresee new issues there, like the accurate computation of the third order derivatives of finite element solutions. For 3D, the approach is straightforward by constructing a 3D metric field and computing 3D geodesics, it maybe time cost but still in control.

This research is supported by the European Research Council (project HEXTREME, ERC-2015-AdG-694020) and by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS).

References

- [1] Bernard P.E., Remacle J.F., Legat V. “Boundary discretization for high-order discontinuous Galerkin computations of tidal flows around shallow water islands.” *International Journal for Numerical Methods in Fluids*, vol. 59, no. 5, 535–557, 2009
- [2] Fortunato M., Persson P.O. “High-order unstructured curved mesh generation using the Winslow equations.” *Journal of Computational Physics*, vol. 307, 1–14, 2016
- [3] Hartmann R., Leicht T. “Generation of unstructured curvilinear grids and high-order discontinuous Galerkin discretization applied to a 3D high-lift configuration.” *International Journal for Numerical Methods in Fluids*, vol. 82, no. 6, 316–333, 2016
- [4] Moxey D., Ekelschot D., Keskin Ü., Sherwin S.J., Peiró J. “High-order curvilinear meshing using a thermo-elastic analogy.” *Computer-Aided Design*, vol. 72, 130–139, 2016
- [5] Karman S.L., Erwin J.T., Glasby R.S., Stefanski D. “High-order mesh curving using WCN mesh optimization.” *46th AIAA Fluid Dynamics Conference*, p. 3178. 2016
- [6] Ruiz-Gironés E., Roca X., Sarrate J. “High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation.” *Computer-Aided Design*, vol. 72, 52–64, 2016
- [7] Toulorge T., Geuzaine C., Remacle J.F., Lambrechts J. “Robust untangling of curvilinear meshes.” *Journal of Computational Physics*, vol. 254, 8–26, 2013
- [8] Remacle J.F., Toulorge T., Lambrechts J. “Robust untangling of curvilinear meshes.” *Proceedings of the 21st International meshing roundtable*, pp. 71–83. Springer, 2013
- [9] Kroll N. “ADIGMA: A European project on the development of adaptive higher order variational methods for aerospace applications.” *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, p. 176. 2006
- [10] Kroll N., Hirsch C., Bassi F., Johnston C., Hillewaert K. *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach: Results of a Collaborative Research Project Funded by the European Union, 2010-2014*, vol. 128. Springer, 2015
- [11] Almeida R.C., Feijóo R.A., Galeao A.C., Padra C., Silva R.S. “Adaptive finite element computational fluid dynamics using an anisotropic error

- estimator.” *Computer Methods in Applied Mechanics and Engineering*, vol. 182, no. 3-4, 379–400, 2000
- [12] Buscaglia G.C., Dari E.A. “Anisotropic mesh optimization and its application in adaptivity.” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 22, 4119–4136, 1997
- [13] Castro-Díaz M., Hecht F., Mohammadi B., Pironneau O. “Anisotropic unstructured mesh adaptation for flow simulations.” *International Journal for Numerical Methods in Fluids*, vol. 25, no. 4, 475–491, 1997
- [14] Formaggia L., Micheletti S., Perotto S. “Anisotropic mesh adaptation in computational fluid dynamics: application to the advection–diffusion–reaction and the Stokes problems.” *Applied Numerical Mathematics*, vol. 51, no. 4, 511–533, 2004
- [15] Dompierre J., Vallet M.G., Fortin M., Bourgault Y., Habashi W. “Anisotropic mesh adaptation-Towards a solver and user independent CFD.” *35th Aerospace Sciences Meeting and Exhibit*, p. 861. 1997
- [16] Huang W. “Metric tensors for anisotropic mesh generation.” *Journal of computational physics*, vol. 204, no. 2, 633–665, 2005
- [17] Frey P.J., Alauzet F. “Anisotropic mesh adaptation for CFD computations.” *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, 5068–5082, 2005
- [18] Gruau C., Coupez T. “3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric.” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 48-49, 4951–4976, 2005
- [19] Li X., Shephard M.S., Beall M.W. “3D anisotropic mesh adaptation by mesh modification.” *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, 4915–4950, 2005
- [20] Tam A., Ait-Ali-Yahia D., Robichaud M., Moore M., Kozel V., Habashi W. “Anisotropic mesh adaptation for 3D flows on structured and unstructured grids.” *Computer Methods in Applied Mechanics and Engineering*, vol. 189, no. 4, 1205–1230, 2000
- [21] Pain C., Umpheby A., De Oliveira C., Goddard A. “Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations.” *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 29-30, 3771–3796, 2001
- [22] Schall E., Leservoisier D., Dervieux A., Koobus B. “Mesh adaptation as a tool for certified computational aerodynamics.” *International journal for numerical methods in fluids*, vol. 45, no. 2, 179–196, 2004
- [23] Courty F., Leservoisier D., George P.L., Dervieux A. “Continuous metrics and mesh adaptation.” *Applied Numerical Mathematics*, vol. 56, no. 2, 117–145, 2006
- [24] Chen L., Sun P., Xu J. “Optimal Anisotropic Meshes for Minimizing Interpolation Errors in L^p -Norm.” *Mathematics of Computation*, vol. 76, no. 257, 179–204, 2007. URL <http://www.jstor.org/stable/40234373>
- [25] Alauzet F., Loseille A., Dervieux A., Frey P. “Multi-dimensional continuous metric for mesh adaptation.” *Proceedings of the 15th International Meshing Roundtable*, pp. 191–214. Springer, 2006
- [26] Loseille A., Alauzet F. “Continuous mesh framework part I: well-posed continuous interpolation error.” *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, 38–60, 2011
- [27] Loseille A., Alauzet F. “Continuous mesh framework part II: validations and applications.” *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, 61–86, 2011
- [28] Zhang R., Johnen A., Remacle J.F. *Curvilinear Mesh Adaptation*, pp. 57–69. Springer International Publishing, Cham, 2019. URL https://doi.org/10.1007/978-3-030-13992-6_4
- [29] Botti L. “Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces.” *Journal of Scientific Computing*, vol. 52, no. 3, 675–703, 2012
- [30] Radzevich S.P. *Geometry of Surfaces: A practical Guide for Mechanical Engineers*. Wiley, 2013
- [31] Beaufort P.A., Lambrechts J., Henrotte F., Geuzaine C., Remacle J.F. “Computing cross fields A PDE approach based on the Ginzburg-Landau theory.” *Procedia engineering*, vol. 203, 219–231, 2017
- [32] Baudouin T.C., Remacle J.F., Marchandise E., Henrotte F., Geuzaine C. “A frontal approach to hex-dominant mesh generation.” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 1, no. 1, 8, Feb 2014. URL <https://doi.org/10.1186/2213-7467-1-8>

- [33] Gu X.D., Yau S.T. *Computational conformal geometry*, vol. 3 of *Advanced Lectures in Mathematics (ALM)*. International Press, Somerville, MA; Higher Education Press, Beijing, 2008. With 1 CD-ROM (Windows, Macintosh and Linux)
- [34] Dobrzynski C. *MMG3D: User guide*. Ph.D. thesis, INRIA, 2012
- [35] Johnen A., Remacle J.F., Geuzaine C. “Geometrical validity of curvilinear finite elements.” *Journal of Computational Physics*, vol. 233, 359–372, 2013
- [36] Shewchuk J.R. “What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures (Preprint).” 2002
- [37] Liu Jianfei Sun Shuli W.D. “Optimal Tetrahedralization for Small Polyhedron: A New Local Transformation Strategy for 3-D Mesh Generation and Mesh Improvement.” *Computer Modeling in Engineering & Sciences*, vol. 14, no. 1, 31–44, 2006. URL <http://www.techscience.com/CMES/v14n1/26667>
- [38] Marot C., Verhetsel K., Remacle J.F. “Revisiting the Search for Optimal Tetrahedralizations.” *Proceedings of the 28th International Meshing Roundtable*. Zenodo, Buffalo, New York, USA, Feb. 2020. URL <https://doi.org/10.5281/zenodo.3653420>