MOX–Report No. 53/2013

A continuum variational approach based on optimal control to adaptive moving mesh methods

Micheletti, S.

MOX, Dipartimento di Matematica “F. Brioschi”
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

A continuum variational approach based on optimal control to adaptive moving mesh methods

Stefano Micheletti

5 novembre 2013

♯ MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
Piazza L. da Vinci 32, 20133 Milano, Italy
stefano.micheletti@polimi.it

Sommario

We cast mesh adaptation based on point relocation in a continuum mechanics analogy. The movement of the mesh points is thus interpreted as a displacement of points of the continuum. We describe our approach on the Dirichlet problem for the Poisson equation in 2D. It is well known that, for a fixed mesh, the best approximation in the energy norm, $|||\cdot|||$, to the exact solution, u, is the Galerkin approximation, u_h, in a finite element space, and that u_h minimizes also a suitable energy functional. The best error, however, still depends on the mesh. The energy functional is then rewritten in terms of the displacement through its displacement-gradient tensor. Thus finding the optimal mesh, where $|||u - u_h|||$ is a minimum, among a family of possible meshes, amounts to computing the displacement field which minimizes the energy functional. This is carried out via the optimal control approach, after enforcing the constraint that the displacement satisfies a diffusion equation with the control functions in the role of a variable diffusivity. This in turn yields the optimal movement of the mesh nodes. An algorithm based on a gradient flow delivers the actual adapted mesh.

Keywords: Mesh adaptation, Node relocation, Continuum mechanics, Optimal control

AMS: 65N30, 65N50, 49J20, 35Q74, 37C10

1 Introduction

In the context of finite element analysis of partial differential equations (PDEs), adaptive procedures try to automatically refine, coarsen, or relocate a mesh to achieve a solution having a specified accuracy or to compute the most accurate solution for a given number of degrees of freedom. Common procedures studied to date include either of the following strategies [33]:

- h-adaptivity: find an optimal distribution of the mesh elements, possibly varying the size, shape and orientation of the triangles (see, e.g., [26, 27]);
- p-adaptivity: find an optimal distribution of the polynomial degree, possibly employing a different formulation which allows to associate different degrees with the elements (see, e.g., [17]);
- r-adaptivity: find an optimal distribution of the mesh points, for a fixed topology of the mesh and a given polynomial degree.

Of course, all of these techniques can be applied singly or combined in parallel to get a better efficiency, sacrificing simplicity. Our approach fits the third issue, namely, mesh point relocation. This undoubtedly enjoys some advantages, since the topology of the mesh does not change over the adaptation iterations, and hence the number of mesh elements can be established a priori, with the desirable benefit that one knows up front the size of the memory required to store the data. On the other hand, it is clear that one cannot hope for the same effectiveness in reducing the error, as the one obtained via, e.g., the h- or the combined hp-adaptivity.

In [2], the two classical ways of relocating nodes, i.e., equidistribution and direct minimization are discussed. In the case of equidistribution, which is a standard device for achieving grid relocation in one dimension, a monitor function is introduced and the (irregularly spaced) grid points in the physical space are related to the (regularly spaced) grid points in the computational space by discrete values of a suitable continuous variable. For example, a very well known monitor function is the so-called arc-length, which ensures that the arc length itself be equidistributed. The alternative approach based on direct minimization consists of minimizing a measure of the error directly with respect to nodal positions as well as to the coefficients of the approximation. This yields the so-called Moving Finite Element equations [3]. These equations are nonlinear and their solution, therefore, requires some form of iteration.

As an example of a complex application, in [22], grid movement and adaptation for viscous flow simulations are controlled by a monitor function which may depend on velocity gradient or other flow variables, such as density or pressure. It is shown that the use of an adaptive mesh improves considerably the efficiency and accuracy of the method in comparison with methods with static mesh points, since the grid points can be easily moved and concentrated towards the regions with large velocity and density gradients, such as boundary layers and multi-material interfaces.

Another research area has been devoted to the generation of adaptive grids through harmonic maps between Riemannian manifolds, see, e.g., [14, 15, 16, 24]. These are essentially works based on differential geometry. The warping of one manifold onto another is realized via an appropriate choice of the metric tensor. In [14] the method is introduced and its reliability is corroborated by providing conditions under which existence and uniqueness for one-to-one maps between multidimensional multiconnected domains hold. The problem of finding a harmonic map between noncompact manifold is dealt with in [16] where, given some sufficient conditions on the domain, the target and the initial map, proof of the existence of a harmonic map that deforms the given map is furnished. The extension of the moving mesh methods based on harmonic maps to deal with mesh adaptation in three space dimensions is carried out in [24]. In obtaining the variational mesh, an optimization problem with some appropriate constraints is solved. The key idea of this approach is to update the interior and boundary grids simultaneously, rather than considering them separately. With any smooth mapping of one Riemannian manifold into another it is possible to associate a variety of invariantly defined functionals. Each such functional determines a class of extremal mappings, in the sense of the calculus of variations, and those extremals play an important role in a number of differential-geometric theories. The contribution of [15] is devoted to a rather general study of a functional of geometrical and physical interest, analogous to an energy, and the central problem is that of deforming a given mapping into an extremal of this functional.

There are many formulations of adaptive meshes and of the objective functions that are used to drive them. There is, however, agreement that an adaptive mesh should be continuous and differentiable with nonvanishing Jacobian. One reliable way to generate such a mesh is by solving elliptic equations to generate body-fitted coordinates [32], which transform the irregular physical mesh to a square computational domain. In practice, one tries to mould a general curvilinear system of coordinates into Cartesian coordinates, or vice versa. Orthogonality versus skewness of curvilinear coordinates is an issue of major concern in these cases, along with an appropriate mesh spacing. In [5], a suitable variational formulation based on a variable diffusion method is combined with a directional control functional which allows one to align mesh lines to a prescribed vector field while adapting the spacing to resolve the data. Other numerical grid generation techniques based on parabolic and hyperbolic systems are also reviewed in [32].

Other approaches, which deal with applications of some impact, exploit an equivalence between node movement and deformation of an elastic body. For example, in [6], to model the structural deformation of an aircraft, the domain must be modified at each time step to be compatible with the changing aircraft shape. The grid is represented as an elastic continuum, introducing a natural mechanisms to prevent node-face collisions of tetrahedra. In particular, to obtain good quality grids, an inverse power law with respect to the the minimal vertex distance is used to model the elementwise Young moduli. A technique combining global node repositioning and mesh optimization in order to perform arbitrary large deformations is proposed in [10] . The work [11] presents a robust, adaptive method for animating dynamic visco-elastic deformable objects that provides a guaranteed frame rate. The approach uses an automatic space and time adaptive level of detail technique, in combination with a large-displacement (Green) strain tensor formulation. In [19] the authors describe an incompressible Unified Continuum model in Euler (laboratory) coordinates with a moving mesh for tracking the fluid-structure interface as part of the discretization, allowing simple and general formulation and efficient computation. They target realistic 3D turbulent fluid-structure interaction applications, and introduce and compensate for mesh motion by defining a local Arbitrary Euler-Lagrange map on each space-time slab as part of the discretization, allowing a sharp phase interface on cell facets. In [29], mass-spring systems are employed to graphically model and animate the realistic behavior of deformable tissue in surgical simulations. A variationally consistent mesh adaptation method for triangular elements in explicit Lagrangian dynamics involving local mesh changes for triangular meshes is presented in [23]. Moving from the fact that time-integration methods developed from a variational principle as that of Hamilton's stationary action, necessarily conserve linear and angular momentum, topological changes for mesh adaptation are developed from Hamilton's principle and space-time discretization, leading to variational mesh adaptation which conserves the total momentum (linear and angular) of the discrete system. A multiscale method in surface processing is presented in [9], which combines the image processing methodology based on nonlinear diffusion equations and the theory of geometric evolution problems. Its aim is to smooth discretized surfaces while simultaneously enhancing geometric features such as edges and corners. This is obtained by an anisotropic curvature evolution, where time is the multiscale parameter. The diffusion tensor depends on the shape operator of the evolving surface.

We finally cite the recent monograph [21] which thoroughly reviews the state-of-the-art adaptive moving mesh methods.

The approach that we pursue in this paper is somewhat related to the works [30, 31, 12]. In all these works, the concern is with computing the derivative of some quantity with respect to node coordinates. In particular, in [12], the shape derivative calculus is employed to compute the derivative of the energy functional associated with the Poisson equation. In [30, 31] a mesh adaptation strategy which allows suitable anisotropy within the mesh is introduced. The approach draws upon methods from numerical optimization in order to modify the node positions of a given (isotropic) mesh such that an a posteriori error estimate is minimized. The discrete adjoint technique [18] is utilized to efficiently evaluate the gradient of the a posteriori error estimate. The Dual Weighed Residual error estimate [4] for the error in a quantity of interest is employed to allow goal driven adaptivity. Thus, all of these last three approaches may be considered "discrete" since they use directly the coordinates of the mesh nodes as degrees of freedom which must be determined in an optimal way in order to minimize a given functional. In our approach, we pursue the same objective as in [12], i.e., to minimize the energy norm of the difference between the exact solution and the numerical approximation to the Poisson equation, as described in section 3. We work mainly, however, on the continuous level, since we rely on the displacement field as the main independent variable, see section 2. The actual position of the mesh points is computed as a by-product of the numerical discretization of the optimality equations, since the mesh nodes are just particular points of the continuum. Moreover, in contrast to [30, 31, 12], we are not only dealing with a variational setting, but we are also resorting to a full optimal control approach. In fact, the displacement field is subjected to a diffusion equation, where the diffusion tensor plays the role of the control function, see section 4. Thus, it is this diffusivity which actually drives the mesh adaptation, modifying the displacement field which becomes the state variable. This diffusion equation, like many elliptic problems, satisfies a maximum principle, which in turn ensures that the mapping between the computational and the physical domains be one-to-one, (see, e.g., [1], where sufficient conditions for injectivity of such mappings are provided). In general, however, this is no longer true upon discretization of the diffusion equation [20], so that a certain care has to be exerted in order to guarantee that this property hold also on the discrete level. For this purpose, we resort to a suitable numerical algorithm to solve the optimality system. This is based on the discretization of a suitable gradient flow, introduced in section 4.1, via a numerical time stepping procedure. The advantages of this approach are that, on the one hand, the gradient flow ensures that the target error be monotonically decreasing as a function of time and that, on the other hand, the approach to the equilibrium state be gradual, so that it is less likely that invalid mesh elements are generated, i.e., elements overlapped one with the other. To further reduce the occurrence of this phenomenon, a Tikhonov regularization is added to the main energy functional to help restraining node movement. The numerical assessment is carried out in sections 5.1-5.3 and conclusions and a look ahead are discussed in section 6.

2 The continuum mechanics background

We identify the computational domain, $\Omega \subset \mathbb{R}^2$, as a generic two dimensional polygonal body whose motion under loading is allowed [7]. The reference configuration, $\widehat{\Omega} \subset \mathbb{R}^2$, corresponding to the configuration of the body at rest is assumed to coincide with Ω. The material coordinates, $\widehat{\mathbf{x}} = (\xi_1, \xi_2)$, are used to label the particles of the body. At any arbitrary time, t, the position of particle $\widehat{\mathbf{x}}$ is given by the coordinate, $\mathbf{x} = (x_1, x_2)$, and in general, the motion of the body is described by a *deformation mapping*,

$$\mathbf{x} = \mathbf{X}(\widehat{\mathbf{x}}, t), \tag{1}$$

where it is assumed that $\mathbf{X}$ is one-to-one with respect to the pair $\widehat{\mathbf{x}} - \mathbf{x}$, and that

$$\mathbf{X}(\widehat{\Omega}, t) \equiv \Omega, \quad t \geq 0, \tag{2}$$

that is, the domain appears to be unchanged as a whole, although the material particles may assume different positions inside the domain. Associated with $\mathbf{X}$, we introduce the *displacement vector*

$$\mathbf{U}(\widehat{\mathbf{x}}, t) = \mathbf{x} - \widehat{\mathbf{x}} = \mathbf{X}(\widehat{\mathbf{x}}, t) - \widehat{\mathbf{x}}, \tag{3}$$

which represents the displacement of a point from the initial configuration to the final configuration. Its gradient, $\nabla_{\widehat{\mathbf{x}}}\mathbf{U}$, the so-called *displacement-gradient tensor*, ca be related to the *deformation gradient tensor*

$$M(\widehat{\mathbf{x}}, t) = \nabla_{\widehat{\mathbf{x}}}\mathbf{X}(\widehat{\mathbf{x}}, t), \tag{4}$$

through the relation

$$\nabla_{\widehat{\mathbf{x}}}\mathbf{U}(\widehat{\mathbf{x}}, t) = \nabla_{\widehat{\mathbf{x}}}(\mathbf{X}(\widehat{\mathbf{x}}, t) - \widehat{\mathbf{x}}) = \nabla_{\widehat{\mathbf{x}}}\mathbf{X}(\widehat{\mathbf{x}}, t) - \boldsymbol{I} = M(\widehat{\mathbf{x}}, t) - \boldsymbol{I}, \tag{5}$$

where $\boldsymbol{I}$ is the identity tensor, and the derivatives are defined with respect to the material coordinate $\widehat{\mathbf{x}}$. In practice, we associate the reference configuration, $\widehat{\Omega}$, with the computational domain covered with an isotropic mesh, and the generic configuration, Ω, with a deformation of $\widehat{\Omega}$, which yields, in general, an anisotropic mesh. Thus, if we introduce the computational mesh, $\widehat{\mathcal{T}}_h$, of $\widehat{\Omega}$, then

$$\mathcal{T}_h(t) = \mathbf{X}(\widehat{\mathcal{T}}_h, t) \tag{6}$$

represents the computational mesh embedded in Ω. We recall that a triangular computational mesh, $\mathcal{T}_h$, is a conformal, shape-regular partition of $\overline{\Omega}$ into simplices $\{K\}$, where h denotes the maximum diameter of the triangles [8].

Actually, we are looking at an arbitrarily adapted, and in general anisotropic mesh, as the image of a fixed isotropic mesh under the mapping $\mathbf{X}$.

Remark 2.1 *The present point of view is just a useful physical analogy which helps understanding our mathematical approach with a more intuitive language. One need not identify the computational domain with an actual elastic material.*

3 The variational setting

To illustrate our approach, we consider the model Poisson problem:

$$\begin{cases} -\Delta u &= & f & \text{in } \Omega, \\ u &= & 0 & \text{on } \partial\Omega, \end{cases} \tag{7}$$

where Δ is the Laplacian operator and $f \in L^2(\Omega)$ is a given function. The weak formulation associated with (7) is: Find $u \in H_0^1(\Omega)$ such that

$$\int_\Omega \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega f(\mathbf{x}) v(\mathbf{x})\,\mathrm{d}\mathbf{x} \quad \forall v \in H_0^1(\Omega), \tag{8}$$

where $H_0^1(\Omega)$ is the subspace of the Sobolev space $H^1(\Omega)$, consisting of functions that together with their first weak derivatives are Lebesgue integrable, which have zero trace on the boundary, $\partial\Omega$, [25]. Notice that hereafter, the domain, Ω, is to be understood as the deformed configuration of the reference domain, $\widehat{\Omega}$, and that all the differential operators without any subscript act on the body coordinate, $\mathbf{x}$. It is also well known that the weak formulation (8) can be obtained as the Euler-Lagrange equation associated with the minimization of the energy functional (the Dirichlet integral), $J : H_0^1(\Omega) \to \mathbb{R}$, given by

$$J(v) = \frac{1}{2}\int_\Omega |\nabla v(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x} - \int_\Omega f(\mathbf{x})v(\mathbf{x}) \, \mathrm{d}\mathbf{x} \quad \forall v \in H_0^1(\Omega), \tag{9}$$

namely,

$$J(u) = \min_{v \in H_0^1(\Omega)} J(v). \tag{10}$$

The Galerkin approximation to (8) based on finite elements (FEM) is obtained in a straightforward way, after introducing the FEM space $V_h^r = \{v_h \in C^0(\overline{\Omega}) : v_h|_K \in \mathbb{P}_r(K), \forall K \in \mathcal{T}_h\}$, where $\mathbb{P}_r(K)$ is the space of polynomials of maximum degree r over K [8]. We then let $V_{h,0}^r = V_h^r \cap H_0^1(\Omega)$. Thus the discrete formulation reads: Find $u_h \in V_{h,0}^r$ such that

$$\int_\Omega \nabla u_h(\mathbf{x}) \cdot \nabla v_h(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \int_\Omega f(\mathbf{x})v_h(\mathbf{x}) \, \mathrm{d}\mathbf{x} \quad \forall v_h \in V_{h,0}^r. \tag{11}$$

It is also straightforward to check that the discrete solution, u_h, can be characterized as the minimum of the energy functional (9) over the finite dimensional space $V_{h,0}^r$, i.e.,

$$J(u_h) = \min_{v_h \in V_{h,0}^r} J(v_h). \tag{12}$$

From (10) and (12), we have that

$$\begin{aligned} J(u_h) - J(u) &= \frac{1}{2}\int_\Omega (|\nabla u_h|^2 - |\nabla u|^2) \, \mathrm{d}\mathbf{x} - \int_\Omega f(u_h - u) \, \mathrm{d}\mathbf{x} \\ &= \frac{1}{2}\int_\Omega |\nabla(u_h - u)|^2 \, \mathrm{d}\mathbf{x} + \underbrace{\int_\Omega \nabla u \cdot \nabla(u_h - u) \, \mathrm{d}\mathbf{x} - \int_\Omega f\,(u_h - u) \, \mathrm{d}\mathbf{x}}_{(a)} \\ &= \frac{1}{2}\int_\Omega |\nabla(u_h - u)|^2 \, \mathrm{d}\mathbf{x}, \end{aligned} \tag{13}$$

where we have omitted the dependence on $\mathbf{x}$ for simplicity, and we have exploited the weak form (8) with $v = u_h - u$ to infer that the term (a) in (13) is zero. Thus we have obtained the result that

$$J(u_h) = J(u) + \frac{1}{2}\int_\Omega |\nabla(u - u_h)|^2 \, \mathrm{d}\mathbf{x}, \tag{14}$$

which provides a link between $J(u_h) - J(u)$ with the so-called *energy norm*, $|||u - u_h|||$, of the discretization error, i.e.,

$$|||u - u_h||| = \Big(\int_\Omega |\nabla(u - u_h)|^2 \, \mathrm{d}\mathbf{x} \Big)^{1/2}. \tag{15}$$

On the other hand, using the inclusion, $V_{h,0}^r \subset H_0^1(\Omega)$, along with (10) and (12), we have immediately that $J(u_h) \geq J(u)$, but relation (14) provides a more quantitative information, and shows also that $J(u_h) = J(u)$ if and only if $u = u_h$. Actually, relations (12) together with (14)-(15) show that

$$u_h = \arg \min_{v_h \in V_{h,0}^r} J(v_h) \equiv \arg \min_{v_h \in V_{h,0}^r} |||u - v_h|||, \tag{16}$$

that is, minimizing the functional J over $V_{h,0}^r$ is equivalent to computing the best approximation to u in the energy norm, out of the finite element space.

Thanks to this property, we thus know that one cannot do better (in the energy norm) than minimizing the functional J over the discrete space, $V_{h,0}^r$, namely, than computing the Galerkin approximation, u_h, solution to (11). However, the resulting error still depends on the finite element space, i.e., on the polynomial degree, r, and on the computational mesh, $\mathcal{T}_h$, defined as in (6), through the deformation mapping, $\mathbf{X}$, in (1). Sticking to r-adaptivity, the key point of our approach is that the problem of relocating the mesh points in an optimal way can be thus equivalently reformulated as the task of computing the deformation mapping which provides the mesh $\mathcal{T}_h$, among all the other possible meshes with the same topology, where the Galerkin approximation features the least error in the energy norm. Formally, we would like to solve

$$\min_{\mathbf{X} \in X} |||u - u_h(\mathcal{T}_h)||| \quad : \quad \mathcal{T}_h = \mathbf{X}(\widehat{\mathcal{T}}_h), \quad u_h = u_h(\mathcal{T}_h) \text{ solution to (11)}, \tag{17}$$

where we have dropped the time dependence, we have agreed that u_h is the Galerkin approximation to u, depending only the actual mesh, $\mathcal{T}_h$, and X is some function space, yet to be defined, collecting all the possible deformation mappings. In principle, this space should consist of all of the possible smooth, one-to-one, mapping from $\widehat{\Omega}$ onto Ω. Of course, this is not practical to deal with. We now show how to define the actual space of trial deformation mappings.

The first step is to observe that, from (3), the displacement vector, $\mathbf{U}$, is thoroughly equivalent to the deformation mapping, $\mathbf{X}$, i.e., to assign a given deformation mapping is equivalent to enforcing a related displacement. Thus, from now on, we focus on the displacement vector as the new independent unknown, instead of the deformation mapping. In order to satisfy (2), suitable boundary conditions should be enforced on $\mathbf{U}$. In particular, a necessary condition is that $\mathbf{U} \cdot \mathbf{n} = 0$, where $\mathbf{n}$ is the unit outward normal vector to $\partial\widehat{\Omega}$.

As a second step, by relating the body coordinate with the material coordinate, it is possible to recast the functional J in (9) as a functional defined on $H_0^1(\widehat{\Omega})$. We introduce the pullback of an arbitrary function $v \in H_0^1(\Omega)$ as

$\widehat{v}(\widehat{\mathbf{x}}) = v(\mathbf{x}) = v(\widehat{\mathbf{x}} + \mathbf{U}(\widehat{\mathbf{x}}))$, and recall that, via the differentiation chain rule, $\nabla v(\mathbf{x}) = M^{-T}(\widehat{\mathbf{x}})\nabla_{\widehat{\mathbf{x}}}\widehat{v}(\widehat{\mathbf{x}})$, M^{-T} denoting the transposed inverse of M, and the differential transform $\mathrm{d}\mathbf{x} = |\mathrm{det}M(\widehat{\mathbf{x}})|\,\mathrm{d}\widehat{\mathbf{x}}$, where $\mathbf{U}(\widehat{\mathbf{x}})$ and $M(\widehat{\mathbf{x}}) = \boldsymbol{I} + \nabla_{\widehat{\mathbf{x}}}\mathbf{U}(\widehat{\mathbf{x}})$ are defined in (3) and (4), respectively.

As a third step, we can express the functional J in (9) as the new functional, $\widehat{J}(\widehat{v}, \mathbf{U})$, with

$$\begin{aligned}\widehat{J}(\widehat{v}, \mathbf{U}) &= \frac{1}{2}\int_{\widehat{\Omega}} |M^{-T}(\mathbf{U}(\widehat{\mathbf{x}}))\nabla_{\widehat{\mathbf{x}}}\widehat{v}(\widehat{\mathbf{x}})|^2\,|\mathrm{det}M(\mathbf{U}(\widehat{\mathbf{x}}))|\,\mathrm{d}\widehat{\mathbf{x}} \\ &- \int_{\widehat{\Omega}} f(\widehat{\mathbf{x}} + \mathbf{U}(\widehat{\mathbf{x}}))\,\widehat{v}(\widehat{\mathbf{x}})\,|\mathrm{det}M(\mathbf{U}(\widehat{\mathbf{x}}))|\,\mathrm{d}\widehat{\mathbf{x}} \quad \forall \widehat{v} \in H_0^1(\widehat{\Omega}). \qquad (18)\end{aligned}$$

It is important to notice that, whereas the source function, f, has to be evaluated in the body coordinate, where it is naturally defined, the arbitrariness of $v \in H_0^1(\Omega)$ reflects in the arbitrariness of $\widehat{v} \in H_0^1(\widehat{\Omega})$, so that it is no longer required to keep track of the implicit dependence of $\widehat{v}$ on $\mathbf{U}$. Moreover, we have emphasized the dependence of M on $\mathbf{U}$ due to the relation $M = \boldsymbol{I} + \nabla_{\widehat{\mathbf{x}}}\mathbf{U}$. We anticipate that in order for this functional to be well defined, we require that the deformation mapping be a C^1-diffeomorphism, i.e., a continuously differentiable bijection, with a continuously differentiable inverse, and that the orientation-preserving constraint, $\mathrm{det}M(\widehat{\mathbf{x}}) > 0$ for a.e. $\widehat{\mathbf{x}} \in \widehat{\Omega}$, hold. With a view to the numerical approximation, this regularity can be relaxed, i.e., it suffices that the deformation mapping be a piecewise C^1-homeomorphism.

The Gâteaux derivative of (18) with respect to $\mathbf{U}$ is

$$\begin{aligned}&\frac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{v}, \mathbf{U})\mathbf{W} = \\ &\int_{\widehat{\Omega}} \Big(\frac{\partial M^{-T}}{\partial \mathbf{U}}(\mathbf{U})\mathbf{W}\Big)(\widehat{\mathbf{x}})\,\nabla_{\widehat{\mathbf{x}}}\widehat{v}(\widehat{\mathbf{x}}) \cdot M^{-T}(\mathbf{U}(\widehat{\mathbf{x}}))\nabla_{\widehat{\mathbf{x}}}\widehat{v}(\widehat{\mathbf{x}})\,|\mathrm{det}M(\mathbf{U}(\widehat{\mathbf{x}}))|\,\mathrm{d}\widehat{\mathbf{x}} \\ +\ &\frac{1}{2}\int_{\widehat{\Omega}} |M^{-T}(\mathbf{U}(\widehat{\mathbf{x}}))\nabla_{\widehat{\mathbf{x}}}\widehat{v}(\widehat{\mathbf{x}})|^2\,\Big(\frac{\partial|\mathrm{det}M|}{\partial \mathbf{U}}(\mathbf{U})\mathbf{W}\Big)(\widehat{\mathbf{x}})\,\mathrm{d}\widehat{\mathbf{x}} \\ -\ &\int_{\widehat{\Omega}} f(\widehat{\mathbf{x}} + \mathbf{U}(\widehat{\mathbf{x}}))\,\widehat{v}(\widehat{\mathbf{x}})\,\Big(\frac{\partial|\mathrm{det}M|}{\partial \mathbf{U}}(\mathbf{U})\mathbf{W}\Big)(\widehat{\mathbf{x}})\,\mathrm{d}\widehat{\mathbf{x}} \\ -\ &\int_{\widehat{\Omega}} \nabla f(\widehat{\mathbf{x}} + \mathbf{U}(\widehat{\mathbf{x}})) \cdot \mathbf{W}(\widehat{\mathbf{x}})\,\widehat{v}(\widehat{\mathbf{x}})\,|\mathrm{det}M(\mathbf{U}(\widehat{\mathbf{x}}))|\,\mathrm{d}\widehat{\mathbf{x}}, \qquad (19)\end{aligned}$$

where the differential terms can be computed using the auxiliary results provided in the Appendix. Moving from the expression (19), one can already devise a minimization algorithm for computing the optimal pair $(\widehat{u}, \mathbf{U})$. One possibility is to resort to a steepest descent method. Alternatively, one can adopt the strategy based on the following gradient flow. For this purpose, let us now introduce the time dependence of the displacement vector, i.e., we assume that $\mathbf{U} = \mathbf{U}(\widehat{\mathbf{x}}, t)$

and, for every $\widehat{\mathbf{x}} \in \widehat{\Omega}$, we introduce the ordinary differential problem:

$$\begin{cases} \dfrac{\partial \mathbf{U}}{\partial t}(\widehat{\mathbf{x}}, t) = -\dfrac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{u}(\widehat{\mathbf{x}}, t), \mathbf{U}(\widehat{\mathbf{x}}, t)), \quad t \in (0, +\infty) \\ \mathbf{U}(\widehat{\mathbf{x}}, 0) = \mathbf{U}_0(\widehat{\mathbf{x}}), \end{cases} \tag{20}$$

where $\frac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{u}, \mathbf{U})$ is the gradient of $\widehat{J}$ with respect to $\mathbf{U}$, which can be recovered from the directional derivatives in (19), and $\mathbf{U}_0$ is a given initial displacement. The function $\widehat{u}(\widehat{\mathbf{x}}, t)$ solves the Euler-Lagrange relation

$$\frac{\partial \widehat{J}}{\partial \widehat{v}}(\widehat{u}, \mathbf{U})\phi = 0 \quad \forall \phi \in H_0^1(\widehat{\Omega}), \tag{21}$$

where the left-hand side is the Gâteaux derivative of (18), i.e.,

$$\frac{\partial \widehat{J}}{\partial \widehat{v}}(\widehat{u}, \mathbf{U})\phi = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\big(\widehat{J}(\widehat{u} + \varepsilon\phi, \mathbf{U}) - \widehat{J}(\widehat{u}, \mathbf{U})\big), \tag{22}$$

given by

$$\begin{aligned} \frac{\partial \widehat{J}}{\partial \widehat{v}}(\widehat{u}, \mathbf{U})\phi &= \int_{\widehat{\Omega}} M^{-T}(\mathbf{U}(\widehat{\mathbf{x}}))\nabla_{\widehat{\mathbf{x}}}\widehat{u}(\widehat{\mathbf{x}}) \cdot M^{-T}(\mathbf{U}(\widehat{\mathbf{x}}))\nabla_{\widehat{\mathbf{x}}}\phi(\widehat{\mathbf{x}}) \, |\mathrm{det} M(\mathbf{U}(\widehat{\mathbf{x}}))| \, d\widehat{\mathbf{x}} \\ &- \int_{\widehat{\Omega}} f(\widehat{\mathbf{x}} + \mathbf{U}(\widehat{\mathbf{x}})) \, \phi(\widehat{\mathbf{x}}) \, |\mathrm{det} M(\mathbf{U}(\widehat{\mathbf{x}}))| \, d\widehat{\mathbf{x}} \quad \forall \phi \in H_0^1(\widehat{\Omega}). \end{aligned} \tag{23}$$

The gradient flow problem (20) represents the continuous-in-time form of a steepest descent algorithm. Actually, to first order in the time step Δt, from (20) we have that

$$\mathbf{U}(\widehat{\mathbf{x}}, t + \Delta t) \simeq \mathbf{U}(\widehat{\mathbf{x}}, t) - \Delta t \, \frac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{u}(\widehat{\mathbf{x}}, t), \mathbf{U}(\widehat{\mathbf{x}}, t)), \tag{24}$$

which can be interpreted as a step of a steepest descent method with step length Δt. Thus, solving (20), will drive us towards the equilibrium solution where $\frac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{u}, \mathbf{U}) = 0$. In fact, the reduced functional, $\widetilde{J}(t) = \widehat{J}(\widehat{u}(\cdot, t), \mathbf{U}(\cdot, t))$, from (20), satisfies

$$\frac{d\widetilde{J}}{dt}(t) = \frac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{u}(\cdot, t), \mathbf{U}(\cdot, t)) \cdot \frac{\partial \mathbf{U}}{\partial t}(\cdot, t) = -\left\| \frac{\partial \widehat{J}}{\partial \mathbf{U}}(\widehat{u}(\cdot, t), \mathbf{U}(\cdot, t)) \right\|^2 \leq 0, \text{ in } \widehat{\Omega}, t \geq 0, \tag{25}$$

that is, $\widetilde{J}(t)$ is a non-increasing function of time over all the trajectory. In practice, it is not possible to solve (20) exactly, and a suitable time-marching numerical scheme has to be adopted, which might fail to satisfy (25) on the discrete level. Moreover, we would like to guarantee that, over all the evolution, no entanglement of the elements occurs on passing from $\widehat{\mathcal{T}}_h$ to $\mathcal{T}_h(t)$, i.e., we want that the image of the points that constitute $\widehat{\mathcal{T}}_h$ define a valid mesh, $\mathcal{T}_h(t)$, according to (6), with $\mathbf{X}$ determined from $\mathbf{U}$ through (3).

4 An approach based on optimal control

In order to try and satisfy these requirements, we resort to an alternative approach, hinging on optimal control. In particular, we require that all the trial deformation mappings, $\mathbf{X} = (x_1, x_2)$, satisfy a diffusion equation driven by an anisotropic diffusion tensor. Namely, we demand that

$$\left\{ \begin{array}{rcll} -\nabla_{\widehat{\mathbf{x}}} \cdot \big((I + A^2)\nabla_{\widehat{\mathbf{x}}}\, x_i\big) & = & 0 & \text{in } \widehat{\Omega}, \quad i = 1, 2 \\ \mathbf{x} \cdot \mathbf{n} & = & \widehat{\mathbf{x}} \cdot \mathbf{n} & \text{on } \partial\widehat{\Omega}, \end{array} \right. \tag{26}$$

where we recall that $\mathbf{n}$ is the unit outward normal vector to $\partial\widehat{\Omega}$, and the boundary condition is equivalent to requiring that $\mathbf{U} \cdot \mathbf{n} = (\mathbf{x} - \widehat{\mathbf{x}}) \cdot \mathbf{n} = 0$. The diffusion coefficient depends on the identity matrix, $I \in \mathbb{R}^{2\times 2}$, and on the symmetric tensor,

$$A = \left[\begin{array}{cc} q_1 & q_2 \\ q_2 & q_3 \end{array} \right], \tag{27}$$

where $q_i, i = 1, 2, 3$, play the role of control variables. In general, $I + A^2$, is a symmetric and positive definite non-homogeneous tensor. Thus, we are subjecting the deformation mapping to satisfy a diffusion equation, where the amount of diffusion is adjusted by tuning the control parameters. When $q_i = \text{const}, i = 1, 2, 3$, it is easy to check that the solution to (26) is $\mathbf{x} = \widehat{\mathbf{x}}$, that is, there is no deformation whatsoever. For a general polygonal domain, due to the boundary conditions, (26) represents a coupled system of partial differential equations for the pair (x_1, x_2). However, when Ω consists only of horizontal and vertical sides, (26) splits into two independent problems. For example, when the domain is the square $(-1, 1)^2$, the two sub-problems become the second-order elliptic problems completed with mixed boundary conditions, given by

$$\left\{ \begin{array}{rcll} -\nabla_{\widehat{\mathbf{x}}} \cdot \big((I + A^2)\nabla_{\widehat{\mathbf{x}}}\, x_1\big) & = & 0 & \text{in } \widehat{\Omega} \\ x_1 & = & \pm 1 & \text{on } \{\xi_1 = \pm 1 \,\&\, -1 < \xi_2 < 1\} \\ (I + A^2)\nabla_{\widehat{\mathbf{x}}} x_1 \cdot \mathbf{n} & = & 0 & \text{on } \{\xi_2 = \pm 1 \,\&\, -1 < \xi_1 < 1\} \end{array} \right. \tag{28}$$

and

$$\left\{ \begin{array}{rcll} -\nabla_{\widehat{\mathbf{x}}} \cdot \big((I + A^2)\nabla_{\widehat{\mathbf{x}}}\, x_2\big) & = & 0 & \text{in } \widehat{\Omega} \\ x_2 & = & \pm 1 & \text{on } \{\xi_2 = \pm 1 \,\&\, -1 < \xi_1 < 1\} \\ (I + A^2)\nabla_{\widehat{\mathbf{x}}} x_2 \cdot \mathbf{n} & = & 0 & \text{on } \{\xi_1 = \pm 1 \,\&\, -1 < \xi_2 < 1\}. \end{array} \right. \tag{29}$$

Notice that the Neumann boundary conditions are somewhat arbitrary, since they are not directly implied by the constraint $\mathbf{U} \cdot \mathbf{n} = 0$. The ones in (28)-(29) are handy, and enforce the vanishing of the conormal derivative associated with the anisotropic diffusion operator. Alternatively, they can be replaced by the standard Neumann conditions $\frac{\partial x_1}{\partial \xi_2} = 0$ and $\frac{\partial x_2}{\partial \xi_1} = 0$, in (28) and (29), respectively. As explicit functions of the control variables, the components of the diffusion tensor, $I + A^2$, are give by:

$$(I+A^2)_{11} = 1+q_1^2+q_2^2, \quad (I+A^2)_{12} = q_1 q_2+q_2 q_3, \quad (I+A^2)_{22} = 1+q_2^2+q_3^2. \tag{30}$$

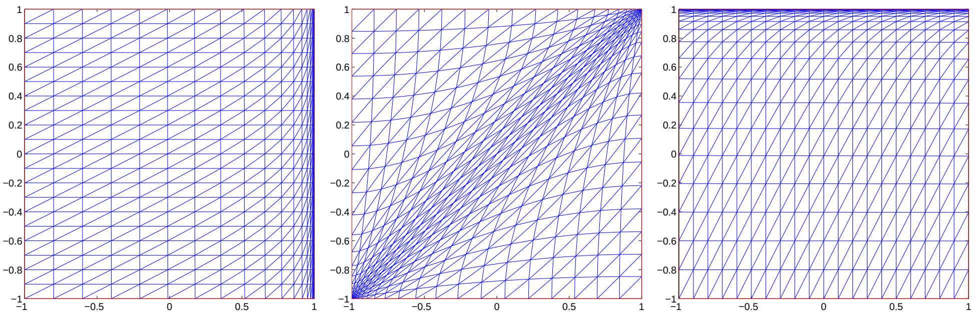

Figura 1: Deformed meshes associated with different control functions: q_1 (left), q_2 (center), and q_3 (right).

To get a flavor about the different role played by the three control functions, we solve (28)-(29) with the following three sets of data: $\{q_1 = \exp(3\,\xi_1), q_2 = q_3 = 0\}$; $\{q_1 = 0, q_2 = \exp(3\,\xi_1\,\xi_2), q_3 = 0\}$; $\{q_1 = q_2 = 0, q_3 = \exp(3\,\xi_2)\}$. More precisely, the solution to these equations has been computed by a FEM approximation of degree 1 on a 20 × 20 structured mesh, $\widehat{\mathcal{T}}_h$, of the reference domain $\widehat{\Omega} = (-1,1)^2$. The corresponding deformed meshes, given by (6) (independently of time), are displayed in Figure 1. Although there is a nonlinear coupling among the three variables, as indicated by (30), which prevents using the superposition principle, we can, however, draw some partial conclusions. Strong values of q_1 induce a refinement of the mesh in the ξ_1 direction, high values of q_3 yield an analogous refinement in the ξ_2 direction, whereas a more involved effect is associated with q_2. In each case, no entanglement of the elements occurs, despite the very steep gradients exhibited by the meshes. For example, in Figure (1) (left), the size of the mesh elements along the right-hand side of the domain is on the order of 10^{-4}.

It is convenient to rewrite equations (28)-(29) in terms of the displacement components, using the transformation (3). This yields

$$\left\{\begin{array}{rcll} -\nabla_{\widehat{\mathbf{x}}}\cdot\big((I+A^2)\nabla_{\widehat{\mathbf{x}}}\,u_1\big) & = & \nabla_{\widehat{\mathbf{x}}}\cdot\big((I+A^2)\,\mathbf{e}_1\big) & \text{in } \widehat{\Omega} \\ u_1 & = & 0 & \text{on } \{\xi_1 = \pm 1\,\&\, -1<\xi_2<1\} \\ (I+A^2)\nabla_{\widehat{\mathbf{x}}}u_1\cdot\mathbf{n} & = & -(I+A^2)\mathbf{e}_1\cdot\mathbf{n} & \text{on } \{\xi_2 = \pm 1\,\&\, -1<\xi_1<1\} \end{array}\right. \tag{31}$$

and

$$\left\{\begin{array}{rcll} -\nabla_{\widehat{\mathbf{x}}}\cdot\big((I+A^2)\nabla_{\widehat{\mathbf{x}}}\,u_2\big) & = & \nabla_{\widehat{\mathbf{x}}}\cdot\big((I+A^2)\,\mathbf{e}_2\big) & \text{in } \widehat{\Omega} \\ u_2 & = & 0 & \text{on } \{\xi_2 = \pm 1\,\&\, -1<\xi_1<1\} \\ (I+A^2)\nabla_{\widehat{\mathbf{x}}}u_2\cdot\mathbf{n} & = & -(I+A^2)\mathbf{e}_2\cdot\mathbf{n} & \text{on } \{\xi_1 = \pm 1\,\&\, -1<\xi_2<1\}, \end{array}\right. \tag{32}$$

where $\mathbf{e}_1 = (1,0)^T$ and $\mathbf{e}_2 = (0,1)^T$ denote the two unit vectors associated with the Cartesian coordinates. With a view to the optimal control problem, these two equations represent the state equations, i.e., the parameter-to-solution map: $\{q_i\}_{i=1,2,3} \to \{u_i\}_{i=1,2}$. Their weak form can be obtained in the usual way.

Define the two boundary subsets,

$$\Gamma_1 = \{\xi_1 = \pm 1 \,\& -1 < \xi_2 < 1\}, \quad \Gamma_2 = \{\xi_2 = \pm 1 \,\& -1 < \xi_1 < 1\}, \tag{33}$$

and the associated Hilbert spaces, $H^1_{\Gamma_i}(\widehat{\Omega}) = \{v \in H^1(\widehat{\Omega}) : v|_{\Gamma_i} = 0\}$, $i = 1, 2$. Then, the weak form associated with (31) is: Find $u_1 \in H^1_{\Gamma_1}(\widehat{\Omega})$, such that

$$\int_{\widehat{\Omega}} (I + A^2)\nabla_{\widehat{\mathbf{x}}}\, u_1 \cdot \nabla_{\widehat{\mathbf{x}}}\, \varphi_1 \,\mathrm{d}\widehat{\mathbf{x}} = -\int_{\widehat{\Omega}} (I + A^2)\, \mathbf{e}_1 \cdot \nabla_{\widehat{\mathbf{x}}}\, \varphi_1 \,\mathrm{d}\widehat{\mathbf{x}} \quad \forall \varphi_1 \in H^1_{\Gamma_1}(\widehat{\Omega}), \tag{34}$$

while the weak form of (32) is: Find $u_2 \in H^1_{\Gamma_2}(\widehat{\Omega})$, such that

$$\int_{\widehat{\Omega}} (I + A^2)\nabla_{\widehat{\mathbf{x}}}\, u_2 \cdot \nabla_{\widehat{\mathbf{x}}}\, \varphi_2 \,\mathrm{d}\widehat{\mathbf{x}} = -\int_{\widehat{\Omega}} (I + A^2)\, \mathbf{e}_2 \cdot \nabla_{\widehat{\mathbf{x}}}\, \varphi_2 \,\mathrm{d}\widehat{\mathbf{x}} \quad \forall \varphi_2 \in H^1_{\Gamma_2}(\widehat{\Omega}). \tag{35}$$

We are now in a position to introduce the optimal control problem. This consists of finding the critical points of the Lagrangian, $\mathcal{L} : V \times Q \times \Lambda \to \mathbb{R}$, given by

$$\begin{aligned}
&\mathcal{L}(\{\widehat{u}, \mathbf{U}\}, \mathbf{q}, \mathbf{\Lambda}) = \\
&\widehat{J}(\widehat{u}, \mathbf{U}) + \sum_{i=1,2} \int_{\widehat{\Omega}} (I + A(\mathbf{q})^2)\nabla_{\widehat{\mathbf{x}}}\, (u_i + \xi_i) \cdot \nabla_{\widehat{\mathbf{x}}} \lambda_i \,\mathrm{d}\widehat{\mathbf{x}} \\
&+ \frac{\alpha}{2} \sum_{j=1}^{3} \int_{\widehat{\Omega}} q_j(\widehat{\mathbf{x}})^2 \,\mathrm{d}\widehat{\mathbf{x}},
\end{aligned} \tag{36}$$

which collects the functional (18) with the constraints provided by the weak form of the state equations, (34)-(35), by adjoining them via Lagrange multipliers, and we have also included a possible Tikhonov regularization term depending on the regularization parameter $\alpha \geq 0$, which controls the tradeoff between goodness of fit to the data and stability. In practice, this terms should help preventing the generation of invalid elements, i.e., elements which overlap with other elements. In particular, the pair $\{\widehat{u}, \mathbf{U}\} = \{\widehat{u}, u_1, u_2\} \in V$ represents the state variable, and we have introduced the compact notation $\mathbf{q} = \{q_i\}_{i=1,2,3} \in Q$ to denote the control variable, and $\mathbf{\Lambda} = \{\lambda_i\}_{i=1,2} \in \Lambda$ which plays the role of the adjoint variable. Moreover, the relation $\nabla_{\widehat{\mathbf{x}}} \xi_i = \mathbf{e}_i$ has been used to unify notation.

Remark 4.1 *Although the study of the well-posedness of the optimal control problem is beyond the scope of this manuscript, we just observe that the Lagrangian $\mathcal{L}$ in* (36) *is well defined, i.e., it is a continuous functional on $V \times Q \times \Lambda$, for the following choices of function spaces: $V = H^1_0(\widehat{\Omega}) \times U$, where $U = \{\mathbf{U} : \widehat{\Omega} \to \mathbb{R}^2 | \mathbf{X}(\widehat{\mathbf{x}}) = \widehat{\mathbf{x}} + \mathbf{U}(\widehat{\mathbf{x}}) \in H^1(\widehat{\Omega})^2 \cap \mathrm{Hom}^{1,+}(\widehat{\Omega})\}$, where $\mathrm{Hom}^{1,+}(\widehat{\Omega})$ designates the set of all piecewise C^1-homeomorphisms from $\widehat{\Omega}$ onto itself, with the constraint that $\det M > 0$ a.e. in $\widehat{\Omega}$; $Q = L^\infty(\widehat{\Omega})^3$, and $\Lambda = H^1_{\Gamma_1}(\widehat{\Omega}) \times H^1_{\Gamma_2}(\widehat{\Omega})$.*

The critical points of the Lagrangian satisfy the well known Karush-Kuhn-Tucker (KKT) system [28]

$$\left\{\begin{array}{lcl}
\dfrac{\partial\mathcal{L}}{\partial\boldsymbol{\Lambda}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})\boldsymbol{\Phi} & = & 0 \quad \forall\boldsymbol{\Phi}\in\Lambda\\
\dfrac{\partial\mathcal{L}}{\partial\widehat{u}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})\,\phi & = & 0 \quad \forall\phi\in H_0^1(\widehat{\Omega})\\
\dfrac{\partial\mathcal{L}}{\partial\mathbf{U}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})\mathbf{W} & = & 0 \quad \forall\mathbf{W}\in U\\
\dfrac{\partial\mathcal{L}}{\partial\mathbf{q}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})\,\mathbf{r} & = & 0 \quad \forall\mathbf{r}\in Q.
\end{array}\right. \tag{37}$$

Equation $(37)_1$ yields the state equations (34)-(35); the adjoint equation $(37)_2$ corresponds to (21) while the adjoint equation $(37)_3$ yields

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial\mathbf{U}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})\mathbf{W} = \frac{\partial\widehat{J}}{\partial\mathbf{U}}(\widehat{u},\mathbf{U})\mathbf{W}\\
&+\sum_{i=1,2}\int_{\widehat{\Omega}}(I+A(\mathbf{q})^2)\nabla_{\widehat{\mathbf{x}}}\,w_i\cdot\nabla_{\widehat{\mathbf{x}}}\lambda_i\,\mathrm{d}\widehat{\mathbf{x}} = 0 \quad \forall\mathbf{W}=(w_1,w_2)\in U,
\end{aligned} \tag{38}$$

where $\frac{\partial\widehat{J}}{\partial\mathbf{U}}(\widehat{u},\mathbf{U})\mathbf{W}$ is given in (19). The optimality condition $(37)_4$, the so-called gradient equation, becomes

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial\mathbf{q}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})\,\mathbf{r} = \sum_{i=1,2}\int_{\widehat{\Omega}}B(\mathbf{q},\mathbf{r})\nabla_{\widehat{\mathbf{x}}}\,(u_i+\xi_i)\cdot\nabla_{\widehat{\mathbf{x}}}\lambda_i\,\mathrm{d}\widehat{\mathbf{x}}\\
&+\alpha\sum_{j=1}^{3}\int_{\widehat{\Omega}}q_j(\widehat{\mathbf{x}})r_j(\widehat{\mathbf{x}})\,\mathrm{d}\widehat{\mathbf{x}} = 0 \quad \forall\mathbf{r}=(r_1,r_2,r_3)\in Q,
\end{aligned} \tag{39}$$

where the tensor $B(\mathbf{q},\mathbf{r}) = \frac{\partial A}{\partial\mathbf{q}}(\mathbf{q})\mathbf{r}$ can be derived from (30) and is

$$B = \begin{bmatrix} 2(q_1r_1+q_2r_2) & r_1q_2+q_1r_2+r_2q_3+q_2r_3\\ r_1q_2+q_1r_2+r_2q_3+q_2r_3 & 2(q_2r_2+q_3r_3)\end{bmatrix}. \tag{40}$$

4.1 The gradient flow

In order to compute the solution to (37), we again resort to a gradient flow, analogously to (20). For this purpose, we first recall that, as is well known in the optimal control framework, the quantity $\frac{\partial\mathcal{L}}{\partial\mathbf{q}}(\{\widehat{u},\mathbf{U}\},\mathbf{q},\boldsymbol{\Lambda})$ appearing on the left-hand side of (39), can be thought of as the derivative with respect to $\mathbf{q}$ of the reduced functional, appearing in the Lagrangian, (36),

$$\mathcal{J}(\mathbf{q}) = \widehat{J}(\widehat{u}(\mathbf{q}),\mathbf{U}(\mathbf{q})) + \frac{\alpha}{2}\sum_{j=1}^{3}\int_{\widehat{\Omega}}q_j(\widehat{\mathbf{x}})^2\,\mathrm{d}\widehat{\mathbf{x}}, \tag{41}$$

where it is understood that the state variable, $\{\widehat{u}, \mathbf{U}\}$, as well as the adjoint variable, $\mathbf{\Lambda}$ (entering $\partial\mathcal{L}/\partial\mathbf{q}$), are functions of the control variable, $\mathbf{q}$, through the state equation $(37)_1$ and the adjoint equations $(37)_{2-3}$, respectively. Then, we consider the problem, for any $\widehat{\mathbf{x}} \in \widehat{\Omega}$,

$$\begin{cases} \dfrac{\partial \mathbf{q}}{\partial t}(\widehat{\mathbf{x}}, t) = -\dfrac{\partial \mathcal{J}}{\partial \mathbf{q}}(\mathbf{q}(\widehat{\mathbf{x}}, t)) & t \in (0, +\infty) \\ \mathbf{q}(\widehat{\mathbf{x}}, 0) = \mathbf{q}_0(\widehat{\mathbf{x}}), \end{cases} \tag{42}$$

where $\mathbf{q}_0$ is a given initial guess for the control variable. Thus, the solution to (37) can be recovered as the asymptotic solution to (42), when $\frac{\partial\mathcal{J}}{\partial\mathbf{q}}(\mathbf{q}) = 0$. Indeed, along the trajectory of this dynamical system, the functional

$$\widetilde{\mathcal{J}}(t) = \mathcal{J}(\mathbf{q}(\cdot, t)) \tag{43}$$

is nonincreasing, since

$$\frac{d\widetilde{\mathcal{J}}}{dt}(t) = -\frac{\partial \mathcal{J}}{\partial \mathbf{q}}(\mathbf{q}(\cdot, t)) \cdot \frac{\partial \mathbf{q}}{\partial t}(\cdot, t) = -\left\| \frac{\partial \mathcal{J}}{\partial \mathbf{q}}(\mathbf{q}(\cdot, t)) \right\|^2 \leq 0, \quad t \geq 0.$$

Of course, also in this case, we shall compute an approximate solution to (42) via a suitable time marching procedure.

5 Numerical assessment

We carry out some experiments aimed at validating the proposed adaptive procedure. For this purpose, we resort to suitable approximations of the function spaces required in (37). In particular, both the finite element solution, u_h, and the components of the displacement, say $u_{1,h}, u_{2,h}$, are picked in $V_h^1 \cap H^1_{\Gamma_1}(\widehat{\Omega}), V_h^1 \cap H^1_{\Gamma_2}(\widehat{\Omega})$, i.e., continuous piecewise linear vanishing on Γ_1, Γ_2, respectively; a consistent choice is made for the components, λ_i, $i = 1, 2$ of the adjoint variable, while the control variables, $\{q_i\}_{i=1,2,3}$ are approximated by piecewise constant functions over the elements, i.e., $q_i \in \{q \in L^2(\widehat{\Omega}) : q|_K = \text{const}, \ \forall K \in \widehat{\mathcal{T}}_h\}$. In all cases, the computational domain for (7) coincides with the square $\widehat{\Omega} = (-1, 1)^2$, and the reference mesh, $\widehat{\mathcal{T}}_h$, shown in Figure 2 (top left), consists of 312 triangles. We have also adopted the time marching procedure, `ode23`, provided by the Matlab ode suite. This handy function employs a Runge-Kutta pair of explicit methods and incorporates time step adaptivity as well. To enhance precision, the optional parameter, `RelTol`, is set to 10^{-6}. The initial guess $\mathbf{q}_0$ in (42) is picked as a piecewise random function. The final time, say T, is a problem dependent parameter that has to be set for each test case. However, the actual stopping criterion exploits the `events` options of the ode suite to enforce termination on the occurrence of the condition

$$\left\| \frac{\partial \mathcal{J}}{\partial \mathbf{q}}(\mathbf{q}(\cdot, t)) \right\|_\infty < \texttt{TOL}, \tag{44}$$

for a given tolerance `TOL`.

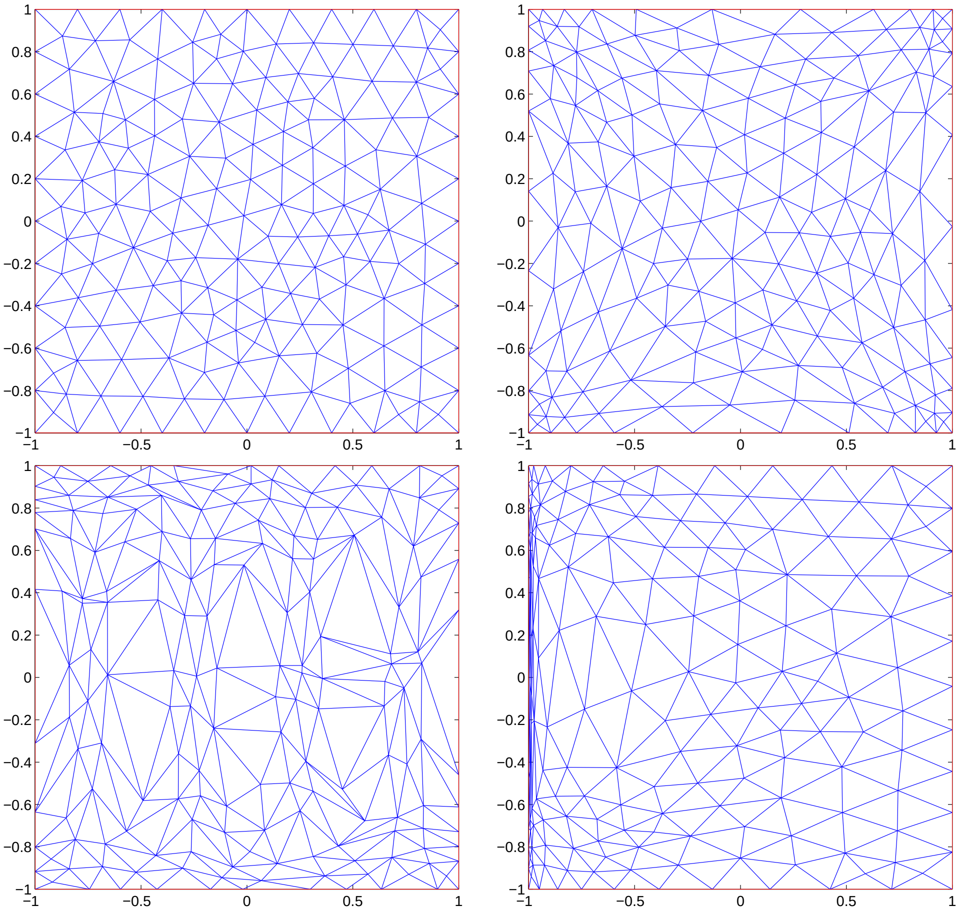

Figura 2: Deformed meshes: reference computational grid $\widehat{\mathcal{T}}_h$ (top left); deformed mesh for test case 1 (top right); deformed mesh for test case 2 (bottom left); deformed mesh for test case 3 (bottom right).

5.1 Test case 1

We choose the source function $f = 1$ in (7). This gives a "hill" like solution, and we expect only a moderate degree of deformation to occur. The final time of the simulation is set to $T = 10^5$. The tolerance `TOL` in (44) is set to 10^{-5}. The Tikhonov regularization parameter is not used, i.e., $\alpha = 0$. The ode suite function carries out 145 time steps, and the deformed mesh, associated with the final value $T = 8.1340 \cdot 10^4$, is displayed in Figure 2 (top right). The behavior of the functional $\mathcal{J}$ in (43) as a function of time is plotted in Figure 3 (left). Notice that the trend is monotonous throughout all the iterations, as expected.

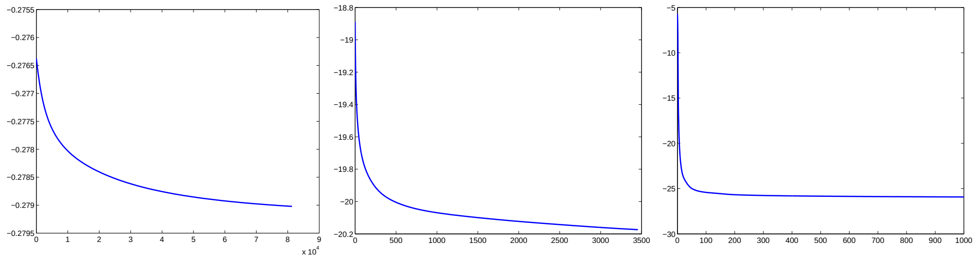

Figura 3: Trend of the functional $\mathcal{J}$ as a function of time: test case 1 (left), test case 2 (center), and test case 3 (right).

5.2 Test case 2

The source f is manufactured such as the exact solution be

$$u = \sin(2\pi x_1)\cos(\pi/2\, x_2).$$

It is characterized by a wavy behavior along the x_1 direction, where two full waveforms occur, whereas only a hill is featured in the orthogonal direction. The final time is fixed to $T = 10^4$, while $\texttt{TOL} = 10^{-3}$. The Tikhonov regularization parameter is set to $\alpha = 0$. The simulation takes 791 time steps to complete, converging at the actual value $T = 3.4543 \cdot 10^3$. The final deformed mesh is shown in Figure 2 (bottom left). The trend of the functional $\mathcal{J}$ in (43) as a function of time is captured in Figure 3 (center).

5.3 Test case 3

We choose the source f in order that the exact solution be

$$u = (x_2 - 1)(x_2 + 1)\big((x_1 - 1)/2 + \exp\left(-100(x_1 + 1)\right)\big),$$

which satisfies only approximately the boundary conditions on the vertical right-hand side, $\{\xi_1 = 1\,\& \; -1 < \xi_2 < 1\}$. The actual value, however, on the order of 10^{-87}, seems really an acceptable approximation to 0. This solution exhibits a very steep gradient across the vertical left-hand side, $\{\xi_1 = -1\,\& \; -1 < \xi_2 < 1\}$. We thus expect very large deformations to occur near to this side. The final time takes on the value $T = 10^3$, and the stopping tolerance is set to $\texttt{TOL} = 2.5 \cdot 10^{-3}$. Due to the strong boundary layer, the Tikhonov regularization parameter is set to $\alpha = 10^{-1}$, in order to prevent element flipping. The simulation takes 236 time steps to converge, yielding $T = 9.2456 \cdot 10^2$. The deformed mesh at the final time is displayed in Figure 2 (bottom right). Figure 3 (right) shows the evolution of the functional $\mathcal{J}$ in (43) as a function of time.

Appendix

We recall some results which are required in the body of the paper. Let $A = A(s)$ be a square regular matrix whose entries depend on the real parameter s. Then the following results can be proved.

Lemma 5.1 (Derivative of a matrix inverse) *The derivative of the inverse of A with respect to s satisfies*

$$\frac{d}{ds}A^{-1}(s) = -A^{-1}(s)\frac{d}{ds}A(s)\, A^{-1}(s).$$

Proof. Differentiating the relation, $A(s)A^{-1}(s) = I$, where I is the identity matrix, we get, $\frac{d}{ds}A(s)A^{-1}(s) + A(s)\frac{d}{ds}A^{-1}(s) = 0$, which yields the desired result after rearranging the terms. □

From this Lemma, it is also easy to prove the following statement.

Corollary 5.1 (Derivative of a transposed inverse matrix)

$$\frac{d}{ds}A^{-T}(s) = -A^{-T}(s)\Big(\frac{d}{ds}A(s)\Big)^T A^{-T}(s).$$

It also hods

Lemma 5.2 (Jacobi's formula: Derivative of a matrix determinant)

$$\frac{d}{ds}\mathrm{det}A(s) = \mathrm{det}A(s)\,\mathrm{trace}\Big(A^{-1}(s)\frac{d}{ds}A(s)\Big) = \mathrm{det}A(s)\,\mathrm{trace}\Big(\frac{d}{ds}A(s)A^{-1}(s)\Big).$$

Proof. See, e.g., [13]. □

Lemma 5.3 (Deformation gradient tensor)

$$M(\widehat{\mathbf{x}}) = \boldsymbol{I} + \nabla_{\widehat{\mathbf{x}}}\mathbf{U}(\widehat{\mathbf{x}}) = \begin{pmatrix} 1 + \dfrac{\partial u_1}{\partial \xi_1} & \dfrac{\partial u_1}{\partial \xi_2} \\ \dfrac{\partial u_2}{\partial \xi_1} & 1 + \dfrac{\partial u_2}{\partial \xi_2} \end{pmatrix}$$

Lemma 5.4 (Determinant of the deformation gradient tensor)

$$\mathrm{det}M = 1 + \nabla_{\widehat{\mathbf{x}}} \cdot \mathbf{U} + \mathrm{det}\nabla\mathbf{U} = 1 + \frac{\partial u_1}{\partial \xi_1} + \frac{\partial u_2}{\partial \xi_2} + \frac{\partial u_1}{\partial \xi_1}\frac{\partial u_2}{\partial \xi_2} - \frac{\partial u_1}{\partial \xi_2}\frac{\partial u_2}{\partial \xi_1}$$

Lemma 5.5 (Inverse of the deformation gradient tensor)

$$M^{-1} = \frac{1}{\mathrm{det}M}\begin{pmatrix} 1 + \dfrac{\partial u_2}{\partial \xi_2} & -\dfrac{\partial u_1}{\partial \xi_2} \\ -\dfrac{\partial u_2}{\partial \xi_1} & 1 + \dfrac{\partial u_1}{\partial \xi_1} \end{pmatrix}$$

Definition 5.1 (Gâteaux derivative) *The Gâteaux derivative of either a scalar, vector, or tensor quantity, $q = q(\mathbf{U})$, at $\mathbf{U}$ in the direction $\mathbf{W}$, is the scalar, vector, or tensor given by*

$$\frac{\partial q}{\partial \mathbf{U}}(\mathbf{U})\mathbf{W} = \lim_{s\to 0}\frac{1}{s}\big(q(\mathbf{U}+s\,\mathbf{W}) - q(\mathbf{U})\big) = \frac{d}{ds}q(\mathbf{U}+s\,\mathbf{W})\Big|_{s=0}. \tag{45}$$

Applying Definition 5.1 to the deformation gradient tensor in Lemma 5.3, we have

Corollary 5.2 (Gâteaux derivative of the deformation gradient tensor)

$$\frac{\partial M}{\partial \mathbf{U}}\mathbf{W} = \frac{\partial M}{\partial u_1}w_1 + \frac{\partial M}{\partial u_2}w_2,$$

with

$$\frac{\partial M}{\partial u_1}w_1 = \begin{pmatrix} \dfrac{\partial w_1}{\partial \xi_1} & \dfrac{\partial w_1}{\partial \xi_2} \\ 0 & 0 \end{pmatrix}$$

and

$$\frac{\partial M}{\partial u_2}w_2 = \begin{pmatrix} 0 & 0 \\ \dfrac{\partial w_2}{\partial \xi_1} & \dfrac{\partial w_2}{\partial \xi_2} \end{pmatrix}.$$

6 Conclusions

The moving mesh method that we have proposed may be interpreted as a sort of elliptic mesh generator, in the sense of [32], where a diffusion equation with a variable diffusivity tensor is solved in the reference/computational domain, $\widehat{\Omega}$, to obtain the deformed mesh in the physical space, Ω. This exploits an analogy by which the movement of the mesh nodes is described in continuum mechanics terminology, resorting to the concept of displacement field. Thus the mesh nodes of a given isotropic mesh, $\widehat{\mathcal{T}_h}$, in the computational domain move as a part of a continuum body. The novelty is that the diffusion tensor acts as a control variable through which one can exert control over the displacement. In particular, the optimal control variables are determined in such a way that a suitable energy functional, measuring the error between exact and approximate solution to the Poisson problem, and dependent on the displacement, be minimized. This eventually yields the optimal mesh, $\mathcal{T}_h$, as the image of $\widehat{\mathcal{T}_h}$ under the resulting deformation mapping. The actual movement of the nodes is realized by introducing a suitable gradient flow, i.e., a time-dependent differential equation, which drives the evolution of the hypothetical dynamical system associated with the body along a trajectory where the energy functional is nonincreasing, and hopefully decreasing, until the equilibrium state is reached where the energy, and thus the error, is at a minimum. In practice, the optimality equations are discretized by finite element approximations on $\widehat{\mathcal{T}}_h$, and the gradient flow is solved through a suitable time marching procedure.

So far, we have applied our approach to the Poisson equation, but the extension to other problems is straightforward, whenever the exact solution minimizes an energy functional, e.g., when the differential operator is self-adjoint and coercive. The theoretical analysis of the well-posedness of the optimal control problem is still an open issue.

Riferimenti bibliografici

[1] G. Alessandrini and V. Nesi, *Univalent σ-harmonc mappings*, Arch. Rational Mech. Anal., 158 (2001), pp. 155–171.

[2] M.J. Baines, *Grid adaptation via node movement*, Appl. Numer. Math., 26 (1998), pp. 77–96.

[3] M.J. Baines, *Moving Finite Elements*, Oxford University Press, Oxford, 1994.

[4] W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser Verlag, 2003.

[5] J.U. Brackbill, *An adaptive grid with directional control*, J. Comput. Phys., 108 (1993), pp. 38–50.

[6] L. Cavagna, G. Quaranta and P. Mantegazza, *Application of Navier–Stokes simulations for aeroelastic stability assessment in transonic regime*, Comput. & Structures, 85 (2007), pp. 818–832.

[7] D.S. Chandrasekharaiah and L. Debnath, *Continuum Mechanics*, Academic Press, Inc., San Diego, 1994.

[8] Ph. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland Publishing Company, Amsterdam, 1978.

[9] U. Clarenz, U. Diewald and M. Rumpf, *Anisotropic geometric diffusion in surface processing*, Proceeedings, Visualization 2000, (2000), pp. 397–405.

[10] G. Compère, J-F. Remacle, J. Jansson and J. Hoffman, *A mesh adaptation framework for dealing with large deforming meshes*, Int. J. Numer. Meth. Engng, 82 (2010), pp. 843–867.

[11] G. Debunne, M. Desbrun, M-P. Cani and A.H. Barr, *Dynamic real-time deformations using space & time adaptive sampling*, ACM SIGGRAPH 2001, Los Angeles, CA, USA, 2001.

[12] M. Delfour, G. Payre and J.-P. Zolésio, *An optimal triangulation for second order elliptic problems*, Rapports de Recherche, 420, INRIA, 1985.

[13] P.J. Dhrymes, *Mathematics for Econometrics*, 3rd edition, Springer-Verlag, New York, 2000.

[14] A.S. Dvinsky, *Adaptive grid generation from harmonic maps on Riemannian manifolds*, J. Comput. Phys., 95 (1991), pp. 450–476.

[15] J. Eells, Jr and J.H. Sampson, *Harmonic mappings of Riemannian manifolds*, Amer. J. Math., 86(1) (1964), pp. 109–160.

[16] A. Fotiadis, *Harmonic maps between noncompact manifolds*, J. Nonlinear Math. Phys., 15(3) (2008), pp. 176–184.

[17] S. Giani and P. Houston, *Anisotropic hp-adaptive discontinuous Galerkin finite element methods for compressible fluid flows*, Int. J Numer. Anal. Model., 9(4) (2012), pp. 928–949.

[18] M.B. Giles and N.A. Pierce, *An introduction to the adjoint approach to design*, Flow Turbul. Combust., 65(3-4) (2000), pp. 393–415.

[19] J. Hoffman, J. Jansson and M. Stöckli, *Unified continuum modeling of fluid-structure interaction*, Math. Models Methods Appl. Sci., 21(3) (2011), pp. 491–513.

[20] W. Huang, *Discrete maximum principle and a Delaunay-type mesh condition for linear finite element approximations of two-dimensional anisotropic diffusion problems*, Numer. Mat. Theory Methods Appl., 4(3) (2011), pp. 319–334.

[21] W. Huang and R.D. Russell, *Adaptive Moving Mesh Methods*, Applied Mathematical Sciences, Springer, New York, 2011.

[22] C. Jin and K. Xu, *An adaptive grid method for two-dimensional viscous flows*, J. Comput. Phys., 218 (2006), pp. 68–81.

[23] S.K. Lahiri, J. Bonet and J. Peraire, *A variationally consistent mesh adaptation method for triangular elements in explicit Lagrangian dynamics*, Int. J. Numer. Meth. Engng, 82(9) (2008), pp. 1073–1113.

[24] R. Li, T. Tang and P.Zhang, *A moving mesh finite element algorithm for singular problems in two and three space dimensions*, J. Comput. Phys., 177 (2002), pp. 365–393.

[25] J.L. Lions and E. Magenes, *Non-Homogeneous Boundary Value Problems and Applications*, vol. I. Springer-Verlag, Berlin, 1972.

[26] S. Micheletti and S. Perotto, *Space-time adaptation for purely diffusive problems in an anisotropic framework*, Int. J Numer. Anal. Model., 7(1) (2010), pp. 125–155.

[27] S. Micheletti and S. Perotto, *The effect pf anisotropic mesh adaptation on PDE-constrained optimal control problems*, SIAM J. Control Optim., 49(4) (2011), pp. 1793–1828.

[28] J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res. Financial Eng., Springer, New York, 2006.

[29] C. Paloc, A. Faraci and F. Bello, *Online remeshing for soft tissue simulation in surgical training*, IEEE Comput. Graph. Appl., (2006), pp. 24–34.

[30] R. Schneider and P.K. Jimack, *Anisotropic mesh adaption based on a posteriori estimates and optimisation of node positions*, Int. Conference on Boundary and Interior Layers, BAIL 2006, G. Lube, G. Rapin (Eds), 2006.

[31] R. Schneider and P.K. Jimack, *On the evaluation of finite element sensitivities to nodal coordinates*, Electron. Trans. Numer. Anal., 32 (2008), pp. 134–144.

[32] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, *Numerical Grid Generation: Foundations and Applications*, North-Holland Publishing Co., New York, 1985.

[33] O.C. Zienkiewicz, R.L. Taylor and J.Z. Zhu, *The Finite Element Method: Its Basis & Fundamentals*, 6th Edition, Elsevier Butterworth-Heinemann, 2006.

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

53/2013 Micheletti, S.
A continuum variational approach based on optimal control to adaptive moving mesh methods

52/2013 Chen, P.; Quarteroni, A.; Rozza, G.
Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by Stokes equations

51/2013 Chen, P.; Quarteroni, A.
Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraint

47/2013 Chkifa, A.; Cohen, A.; Migliorati, G.; Nobile, F.; Tempone, R.
Discrete least squares polynomial approximation with random evaluations - application to parametric and stochastic elliptic PDEs

50/2013 Antonietti, P.F.; Verani, M.; Zikatanov, L.
A two-level method for Mimetic Finite Difference discretizations of elliptic problems

49/2013 Micheletti, S.
Fast simulations in Matlab for Scientific Computing

48/2013 Simone Palamara, Christian Vergara, Elena Faggiano, Fabio Nobile
An effective algorithm for the generation of patient-specific Purkinje networks in computational electrocardiology

45/2013 Sangalli, L.M.; Secchi, P.; Vantini, S.
Analysis of AneuRisk65 data: K-mean Alignment

46/2013 Marron, J.S.; Ramsay, J.O.; Sangalli, L.M.; Srivastava, A.
Statistics of Time Warpings and Phase Variations

44/2013 Sangalli, L.M.; Secchi, P.; Vantini, S.
AneuRisk65: a dataset of three-dimensional cerebral vascular geometries