Finite transformation rigid motion mesh morpher; A grid deformation approach

Athanasios G. Liatsikouras1,2,3 | Gabriel Fougeron4
George S. Eleftheriou4 | Guillaume Pierrot4,5

1ESI Software Germany GmbH, Essen, Germany
2Envirotherm GmbH, Essen, Germany
3National Technical University of Athens, School of Mechanical Engineering, Parallel CFD & Optimization Unit, Athens, Greece
4ESI - Group, Rungis, France
5ICON Technology & Process Consulting Ltd. Management, Paris, France

Correspondence
Athanasios G. Liatsikouras, ESI Software Germany GmbH, Kruppstr. 90, ETEC H4, 3.OG, Essen 45145, Germany.
Email: thanasis.liatsi@gmail.com

Summary
In any shape optimization framework and specifically in the context of computational fluid dynamics, a robust and reliable grid deformation tool is necessary to undertake the adaptation of the computational mesh to the updated boundaries at each optimization cycle. Grid deformation has its share of challenges, namely, to maintain high mesh quality (avoid distorted elements and tangles) even when dealing with extreme deformations. In this work a novel grid deformation algorithm, the finite transformation rigid motion mesh morpher (FT-R3M) is proposed. FT-R3M is essentially a mesh-free grid deformation approach, since it does not require any inertial quantities and it gracefully propagates the movement of the boundaries (surface mesh) to the internal nodes of the mesh (volume mesh), by keeping the motion of its parts (referred to as stencils) as-rigid-as-possible. It is an optimization-based method, which means that the interior nodes of the computational mesh are displaced to minimize a distortion metric related to the elastic deformation energy, by favoring rigidity in critical directions, thus being able to handle mesh anisotropies very efficiently. Results are presented for three test cases; a rotated airfoil with a mesh appropriate for viscous flow; a simulation of a low Reynolds duct case; a beam.

Keywords
as-rigid-as-possible, grid displacement, mesh anisotropy, mesh-free/mesh-less, nonlinear constraint, rigid motion, stencils

1 | INTRODUCTION

A variety of stochastic and gradient-based methods have been devised to solve aerodynamic shape optimization problems such as the design of ducts or manifolds for minimum power losses, cars with optimal combination of drag and lift,1 aircraft/wings for maximum lift2,3 or minimum drag,4 and so forth. During the optimization, a technique to deal with the necessary changes of the boundaries (surface mesh), namely, to adapt the computational mesh (volume mesh) to the updated geometry, is required.

A well-known method to handle the boundary changes during optimization, is remeshing, in which after every optimization cycle a computational mesh is generated anew. This is a time-consuming process and especially in complex

Abbreviation: FT-R3M, finite transformation rigid motion mesh morpher
geometries manual intervention might be needed, something which is not desired in a fully automated optimization process. In addition, in the case of a gradient-based optimization method, inconsistencies are produced in the gradient since the latter is computed at isoconnectivity, which makes this technique neither robust nor reliable.

On the other hand, a rival and a promising way to deal with the necessary changes is to use a grid deformation tool (a.k.a. mesh morpher) which propagates the movement of the boundaries to the interior computational mesh. The prerequisite is to generate a mesh only once, at the beginning of the optimization process and then, the mesh morpher will undertake the adaptation of the computational mesh to the updated boundary shapes at each optimization step without changing the mesh topology.

The challenge when a grid deformation tool is used, is to maintain a good mesh quality of the deformed mesh, namely, to avoid tangled or distorted elements, highly skewed cells, changes in the aspect ratio of the elements, and so forth, which may cause nonconvergence or even divergence of the computational fluid dynamics (CFD) software, for example, in CFD-based problems. In literature, there is a variety of grid deformation techniques. Most of them require a trade-off between the attained mesh quality and CPU cost. A popular grid deformation method is the spring analogy method. In this approach, each edge of the mesh is “replaced” by a tension spring with the spring stiffness as the inverse of the edge length. Unfortunately, this method does not prevent inverted elements and it fails when the local mesh motion exceeds significantly the local mesh size. Many improvements have been achieved to prevent element inversion by introducing torsional springs or semitorsional springs, but handling mesh anisotropies is still a challenge. Another simple and easy way to deform a mesh, in terms of implementation complexity, is by using the Laplacian smoothing. The computation of the displacement of the internal nodes of the computational mesh requires the solution of a linear system the dimension of which scales with the number of nodes of the mesh. This technique proves to be efficient but not very robust in case of significant rotations and complex transformations.

A competitive grid deformation tool is the linear elasticity morpher, in which the computational mesh is handled as an elastic solid body. In this method, mesh deformation is accomplished by solving the linear elasticity equations for each mesh point inside the geometric domain. Since the elasticity equations contain material properties (Young's modulus and Poisson's ratio), these should be related to the mesh characteristics. In the context of a finite volume CFD solver, implementing this method is not trivial, since it is usually solved with the finite element method. A very promising and well-established grid deformation technique is this one based on the fundamental work of de Boer et al., Rendall and Allen, and Jakobson and Amoignon on mesh morphing based on radial basis functions (RBF) interpolation. Mesh morphing based on RBF is a robust and reliable grid deformation technique but computationally very heavy, as the matrices involved in the computations are very dense. The cost of solving such a system with an iterative method scales with the number of surface points \( N_s \) that cause the deformation as \( O(N_s^2) \). Improvements have been achieved by using appropriate preconditioners and the fast multipole method for the acceleration of the matrix-vector multiplications that can reduce the cost to \( O(N_s \log N_s) \), or even less, but at the cost of increasing the complexity. The work of Rendall and Allen is based on data reduction algorithms for selecting the surface points in such an efficient way so that the cost of training the RBF network is reduced up to eight times in comparison to previous works and the cost of the mesh update up to 55 times. Their work has been followed by Michler, 29 where he is focused on the deflection of control surfaces of aircraft configurations where the meshes are very dense and big. In his work, locally confined RBF-based mesh deformation has been applied, the mesh deformation is restricted in the vicinity of the deflecting control surfaces, leading thus to a reduced computational cost of the method. A big challenge in this technique is the selection of nonuniform spacing of points and handling of mesh anisotropies.

The rigid motion mesh morpher has already been introduced. It has been shown that it can handle mesh rotation and mesh anisotropy very efficiently. An improvement upon this concept, is the development of the finite transformation rigid motion mesh morpher (FT-R3M), which is presented in this article that eliminates the need for subcycling and keeping track of the “rigid-motion” history of the stencils for all subcycles. FT-R3M is a nonlinear variant of the rigid motion mesh morpher, since, as it is shown in this article, a nonlinear constraint is imposed. It employs, as it will be explained below, the Gauss–Newton algorithm to solve the nonlinear least squares problem that occur and the Polar decomposition method in order to project the estimation of the rotation matrix that is computed, to the special orthogonal space \( SO(n) \) (rotation group), where \( n \) is the spatial dimensions of the problem.

The basic idea of FT-R3M is to adapt/deform the internal nodes of the computational mesh, the displacement of which is not prescribed, to a given displacement field of the prescribed nodes (in most cases the prescribed nodes correspond to the boundary nodes of the shape), by keeping parts/elements (a.k.a. stencils) of the mesh as-rigid-as-possible. To do so, the energy functional of the stencils between the initial and the final state of the shape, should be minimal.
2 | RIGID MOTION AS A BUILDING BLOCK TOWARDS MESH MORPHING

FT-R3M is a mesh-free minimization-based grid deformation technique, in which a target functional is defined, related to the elastic deformation energy, which needs to be minimized in order to propagate the movement of the boundaries to the internal nodes of the CFD mesh. In the subsections that follow, the rigid motion and the total energy functional to be minimized will be introduced.

2.1 Rigid motion

A rigid motion (or an isometry) consists of rotations, translations, reflections, and glide reflections (or a combination of them) such that the distance between every pair of points/vectors is preserved. More precisely, a rigid motion is defined as a map \( \phi : \mathbb{R}^n \rightarrow \mathbb{R}^n \) that conserves the following inner product

\[
\forall \ \vec{x}, \vec{y} \in \mathbb{R}^n, \quad \langle \phi(\vec{x}), \phi(\vec{y}) \rangle = \langle \vec{x}, \vec{y} \rangle,
\]

where \( \phi : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is an affine transformation with a rotation matrix.

The idea of the morpher, which is proposed in this article, is to try to mimic such kind of motion. This means that the internal nodes of the computational mesh, which are free to be displaced, are grouped into groups and are forced to follow a motion which is as-close-as-possible to its rigid motion. In the context of the FT-R3M, only in rotations and translations (and the combination of them) are allowed, since reflections and glide reflections produce inverted elements which is undesirable in most software dealing with the solution of partial differential equations. Thus, assuming a rotation matrix \( R \in \text{SO}(n) \) \( (R \cdot R^T = I_n, \text{where } I_n \text{ is the identity matrix}) \) and a translation vector \( \vec{t} \in \mathbb{R}^n \), a rigid motion when acting on any vector \( \vec{x} \), produces the transformed vector \( \phi(\vec{x}) \) of the form

\[
\phi(\vec{x}) = R \vec{x} + \vec{t}.
\]

2.2 The energy functional

Let us denote by \( \mathbb{N} \) the set nodes, \( \partial \mathbb{N} \) the boundary nodes, \( \tilde{\mathbb{N}} \) the internal nodes and \( \mathcal{M} \) the mesh. From now on, let \( \vec{X}_i \) be the initial position vector of node with index \( i \) and \( \vec{x}_i \) the final one. If the boundary nodes follow a rigid body motion (prescribed motion), we also want to a priori impose to the interior nodes of the mesh the same rigid body motion. The aforementioned can be expressed as

\[
\exists (R, \vec{t}) \in (\text{SO}(n) \times \mathbb{R}^n) \mid \forall i \in \partial \mathbb{N}, \ \vec{x}_i = R \vec{X}_i + \vec{t} \Rightarrow [\forall k \in \tilde{\mathbb{N}}, \ \vec{x}_k = R \vec{X}_k + \vec{t}].
\]

To find the solution in such a problem, the pair \( (R, \vec{t}) \in \text{SO}(n) \times \mathbb{R}^n \) and \( \vec{x}_i \in \tilde{\mathbb{N}} \) is needed. This leads to an optimization problem in which the following “energy functional” should be minimized

\[
E(R, \vec{t}, \vec{x}_1, \ldots, \vec{x}_n) = \sum_{i \in \tilde{\mathbb{N}}} \|\phi(\vec{X}_i) - \vec{x}_i\|^2 = \sum_{i \in \tilde{\mathbb{N}}} \|R \vec{X}_i + \vec{t} - \vec{x}_i\|^2.
\]

Solving an optimization problem like this in Equation (4) is “translated” as follows: Find the closest rigid motion between the initial and the final state. Without loss of generality, Equation (4) can be written on a pair of nodes that form an edge, not necessarily a geometrical edge, instead of the nodes themselves, as follows

\[
E(R, \vec{x}_1, \ldots, \vec{x}_n) = \sum_{(i,j) \in \mathcal{M}} \|R(\vec{X}_i - \vec{X}_j) - (\vec{x}_i - \vec{x}_j)\|^2.
\]
By minimizing Equation (5), the whole mesh is handled as one body, namely, there is no definition of locality. Therefore, in order to make the method more flexible and robust, nodes of the mesh are grouped into cloud of points, which in our context are called “stencils.” The role of the stencil is to introduce the locality on the proposed method and capture with more flexibility the deformation of different parts/areas of the computational mesh. Local transformation problem from the initial to the final state are defined and then, all those local problems are glued together by summing them up into a global problem. A stencil is practically a collection of nodes or in other words, cloud of points (graph). There is a freedom in the selection/creation of the stencils. The criteria of forming the stencils are distance-based, meaning that the topology of the mesh plays no role of their formation, thus the proposed method belongs to the family of the mesh-free (mesh-less) methods. An example of two stencils is shown in Figure 1. The overlapping of the stencils is necessary, which means that a node should belong to more than one stencil in order to achieve smooth deformations. The reason is that if the stencils are not overlapping, then the movement of the nodes of two neighboring stencils might be inconsistent, since the displacement of each node is computed by minimizing a global metric consisting from the summation of the local problems. The updated position of an internal node of the computational mesh is a compromise between the rigid motion of each stencil where this node belongs to. By defining the global energy functional as a weighted summation of this of each stencil belonging to the mesh, the final expression of the energy functional to be minimized can be written as

$$E = \sum_{s \in S} w_s \sum_{(i,j) \in s} \mu_{s,ij} \| R_s (\vec{x}_j - \vec{x}_i) - (\vec{x}_j - \vec{x}_i) \|^2. \quad (6)$$

In Equation (6), $S$ is the stencil set, $s$ is a stencil belonging to this, $(i, j) \in s$ denotes a a pair of nodes belonging to stencil $s$ and $R_s$ is the rotation matrix of stencil $s$; $w_s$ is a scalar weight per stencil that stresses the importance of some stencils as higher than that of some others’ (e.g., in case of a boundary layer), or reflects the volume size of the stencil to mimic a domain integral. Regarding $\mu_{s,ij}$, this is a scalar weight per stencil-edge that accounts for mesh anisotropy. It prevents the distortion of a stencil by favoring/penalizing most the edges that are compressed, handling thus mesh anisotropies. Furthermore, by denoting as

$$\tilde{e}_{s,ij} = \sqrt{w_s \mu_{s,ij} [R_s (\vec{x}_j - \vec{x}_i) - (\vec{x}_j - \vec{x}_i) ]} \quad (7)$$

the total energy functional in Equation (6) to be minimized can be rewritten as

$$E = \sum_{s \in S} \sum_{(i,j) \in s} \| \tilde{e}_{s,ij} \|^2. \quad (8)$$

In the final expression of the energy functional stated in Equation (6) and, consequently also in Equation (8), there is a “hidden” nonlinear constraint; namely, every matrix $R_s$ (one for each stencil of the mesh) should be a “real” rotation matrix that belongs in the special orthogonal $SO(n)$ group. This constraint can be expressed as

$$\forall s \in S, \quad R_s R_s^T = I_n. \quad (9)$$
In addition, if the constraint equation is denoted by

\[ \varphi(R_s) = R_s R_s^T - I_n = 0_n, \quad \forall s \in S \]  

(10)

then the problem to be minimized (including the constraint) so as to propagate the movement of the boundaries to the interior computational mesh, can be expressed as

\[
\begin{align*}
\min & \quad E = \sum_{s \in S} \sum_{(i,j) \in s} \| \tilde{e}_{s,ij} \|^2 \\
\text{s.t.} & \quad \varphi(R_s) = 0_n, \quad \forall s \in S.
\end{align*}
\]

(11)

Equation (11) express the energy functional which is required to be minimized in order to compute the final position of the internal nodes of the computational mesh, when a prescribed displacement field of the boundaries is given. The unknowns in this problem are the coefficients of each matrix \( R_s, \forall s \in S \) and each vector \( \tilde{x}_i, \forall i \in \tilde{N} \). In the chapter that follows, it is shown how the minimization of the nonlinear problem, stated in Equation (11), is treated and solved.

### 3 | TREATMENT OF THE NONLINEAR PROBLEM

Since the problem to be solved is nonlinear, due to the quadratic constraint of Equation (11), it has to be treated appropriately. In this case, the Gauss–Newton algorithm, which is appropriate for solving nonlinear least squares problems, is employed. According to the Gauss–Newton algorithm, the latter can be derived by linearly approximating the vector \( \tilde{e}_{s,ij} \).

Assuming that a current guess \((\tilde{x}_i^\kappa, R_s^\kappa)\) is available and by introducing the increments \((\delta \tilde{x}_i, \delta R_s)\), the energy expressed in Equation (11), using also Equation (7), can be written using the Taylor’s theorem as

\[
\begin{align*}
\min & \quad \sum_{s \in S} \sum_{(i,j) \in s} \left[ \tilde{e}_{s,ij} \right]_k^2 + \frac{\partial \tilde{e}_{s,ij}}{\partial R_s} : \delta R_s + \frac{\partial \tilde{e}_{s,ij}}{\partial \tilde{x}_i} \delta \tilde{x}_i + \frac{\partial \tilde{e}_{s,ij}}{\partial \tilde{x}_j} \delta \tilde{x}_j \\
\text{s.t.} & \quad \varphi(R_s) = 0_n, \quad \forall s \in S.
\end{align*}
\]

(12)

In our case, since \( \tilde{e}_{s,ij} \) is linear with respect to (w.r.t) its free variables, the linear approximation using Taylor’s theorem is exact. The term \( (\partial \tilde{e}_{s,ij} / \partial R_s : \delta R_s) \) in Equation (12) denotes the directional derivative of the vector valued function \( \tilde{e}_{s,ij} \) w.r.t. the second-order tensor \( R_s \) in the direction of \( \delta R_s \). Considering that at the current step, the matrix \( R_s^\kappa \) is a rotation matrix (to be explained later on how this constraint is ensured) it yields that

\[ \varphi(R_s) \big|_k = R_s^\kappa (R_s^\kappa)^T - I_d = 0_n. \]  

(13)

Taking into account Equation (13) and the fact that

\[
\begin{align*}
\frac{\partial \tilde{e}_{s,ij}}{\partial R_s} : \delta R_s &= \left[ \frac{d}{d \alpha} (\tilde{e}_{s,ij}(R_s + \alpha \cdot \delta R_s)) \right]_{\alpha=0} = \sqrt{w_{s,i,j}} \delta R_s (\tilde{X}_j - \tilde{X}_i) \\
\frac{\partial \tilde{e}_{s,ij}}{\partial \tilde{x}_i} \delta \tilde{x}_i &= \sqrt{w_{s,i,j}} \delta \tilde{x}_i \\
\frac{\partial \tilde{e}_{s,ij}}{\partial \tilde{x}_j} \delta \tilde{x}_j &= -\sqrt{w_{s,i,j}} \delta \tilde{x}_j \\
\frac{\partial \varphi(R_s)}{\partial R_s} : R_s &= \left[ \frac{d}{d \alpha} (\varphi(R_s + \alpha \cdot \delta R_s)) \right]_{\alpha=0} = \sqrt{w_{s,i,j}} \left[ \delta R_s (R_s^\kappa)^T + R_s^\kappa (\delta R_s^\kappa)^T \right]
\end{align*}
\]

and some mathematical manipulations, Equation (12) becomes
\[
\begin{aligned}
\min & \sum_{s \in S} \sum_{(i,j) \in S} \mu_{s,ij} [(R^s_i + \delta R_i)(\bar{x}_j - \tilde{x}_i) - (\delta \bar{x}_j - \delta \tilde{x}_i) - (\bar{x}_j - \tilde{x}_i)]^2 \\
\text{s.t.} & \quad \delta R_i(R^s_i)^T + R^s_i(\delta R_i)^T = 0_n, \quad \forall s \in S.
\end{aligned}
\] (14)

From the linearized set of equations stated in Equation (14) it is obvious that the matrix \( \Omega^{s+1} = \delta R_i(R^s_i)^T \) is skew-symmetric (antisymmetric), hence

\[
\begin{aligned}
\min & \sum_{s \in S} \sum_{(i,j) \in S} \mu_{s,ij} [(I_n + \Omega^{s+1})R^s_i(\bar{x}_j - \tilde{x}_i) - (\delta \bar{x}_j - \delta \tilde{x}_i) - (\bar{x}_j - \tilde{x}_i)]^2 \\
\text{s.t.} & \quad \Omega^{s+1} + (\Omega^{s+1})^T = 0_n, \quad \forall s \in S.
\end{aligned}
\] (15)

Finally, since \( \Omega^{s+1} \) is antisymmetric, it implies that the product \( \Omega^{s+1} \{ R^s_i(\bar{x}_j - \tilde{x}_i) \} \) can be replaced by the product \( \bar{b}_i \times \{ R^s_i(\bar{x}_j - \tilde{x}_i) \} \), where \( \bar{b}_i \) is a vector. By taking into consideration the latter and by rearranging some terms in Equation (15) we end up with

\[
\begin{aligned}
\min & \sum_{s \in S} \sum_{(i,j) \in S} \mu_{s,ij} \bar{b}_i \times \{ R^s_i(\bar{x}_j - \tilde{x}_i) \} - (\delta \bar{x}_j - \delta \tilde{x}_i) - (\bar{x}_j - \tilde{x}_i) + R^s_i(\bar{x}_j - \tilde{x}_i)]^2 \\
\text{s.t.} & \quad \Omega^{s+1} + (\Omega^{s+1})^T = 0_n, \quad \forall s \in S
\end{aligned}
\] (16)

or by denoting as

\[
e_{s,ij}^{\text{linear}} = \sqrt{w_s \mu_{s,ij}} \bar{b}_i \times \{ R^s_i(\bar{x}_j - \tilde{x}_i) \} - (\delta \bar{x}_j - \delta \tilde{x}_i) - (\bar{x}_j - \tilde{x}_i) + R^s_i(\bar{x}_j - \tilde{x}_i)]^2.
\] (17)

Equation (16) is written as

\[
\begin{aligned}
\min E_L & = \min \sum_{s \in S} \sum_{(i,j) \in S} \| e_{s,ij}^{\text{linear}} \|^2 \\
\text{s.t.} & \quad \Omega^{s+1} + (\Omega^{s+1})^T = 0_n, \quad \forall s \in S
\end{aligned}
\] (18)

Once the displacement field of the prescribed nodes (namely, the displacement of the nodes in the surface mesh) is known, the energy functional in Equation (18) is minimized in a least squares sense, by finding the stationary points w.r.t. the corresponding unknowns by satisfying

\[
\frac{\partial E_L}{\partial \delta \tilde{x}_i} = 0 \quad \text{and} \quad \frac{\partial E_L}{\partial \bar{b}_i} = 0.
\] (19)

The derivation of the linearized total energy functional w.r.t. the degrees of freedom, namely, \( \delta \tilde{x}_i \) and \( \bar{b}_i \), as stated at Equation (19), yields a symmetric positive definite system to be solved of the form

\[
\begin{bmatrix}
N & H \\
H^T & M
\end{bmatrix}
\begin{bmatrix}
\delta \tilde{x}_i \\
\bar{b}_i
\end{bmatrix} = \begin{bmatrix}
\bar{t}_{\delta \tilde{x}_i} \\
\bar{t}_{\bar{b}_i}
\end{bmatrix}
\] (20)

where \( N \) and \( H \) are the multipliers of \( \delta \tilde{x}_i \) and \( \bar{b}_i \), respectively, which derive from the first term of Equation (19), whereas \( H^T \) and \( M \) are the multipliers of \( \delta \tilde{x}_i \) and \( \bar{b}_i \), respectively, deriving from the second term of Equation (19). Thus, if the left-hand matrix is denoted as \( A \), we have

\[
A \tilde{u} = \bar{t},
\] (21)

where \( \tilde{u} \) is the vector that consists of the free variables. Using the condensation\(^{31} \) technique, also known as static condensation, we can eliminate \( \bar{b}_i \) in terms of \( \delta \tilde{x}_i \), so as to decrease the size of the system that is finally solved. To do so, the second
equation of the system (20) needs to be written in terms of \( \tilde{b}_s \), namely,

\[
\tilde{b}_s = M^{-1}(\tilde{I}_k - \tilde{H}^T \delta \tilde{x}_i) \tag{22}
\]

then, the first equation of the expression (20) becomes

\[
N \delta \tilde{x}_i + \tilde{H} M^{-1}(\tilde{I}_k - \tilde{H}^T \delta \tilde{x}_i) = \tilde{I}\delta \tilde{x}_i. \tag{23}
\]

After having solved the linear system and \( (\delta \tilde{x}_i, \delta R_s) \) are available, \( (\tilde{x}_i^{s+1}, R_s^{s+1}) \) are updated through

\[
\tilde{x}_i^{s+1} = \tilde{x}_i^s + \delta \tilde{x}_i \tag{24}
\]

\[
R_s^{s+1} = (I_n + \Omega_s^{s+1})R_s^s. \tag{25}
\]

The solution of the linear system (21) provides the displacement of every node of the mesh \( (\delta \tilde{x}_i) \) and the increment of the rotation matrix \( \delta R_s \). In this case, the quadratic constraint in Equation (9) is not satisfied, which means that \( R_s^{s+1} \) it is not a rotation matrix that belongs to the special orthogonal group \( \text{SO}(n) \) (more details in chapter 4; that is the reason for denoting the update as \( R_s^{s+1} \)).

### 4 PROJECTION ON THE ORTHOGONAL GROUP

Linearizing the quadratic constraint and then solving the linear system in Equation (21) does not satisfy the initial constraint, namely that \( R_s^{s+1} \) should belong to the special orthogonal space \( \text{SO}(n) \). The linearized constraint, meaning that the matrix \( \Omega_s^{s+1} \) is antisymmetric, ensures that the calculated matrix \( R_s^{s+1} \) lies on the space which is tangent at \( R_s^s \). Thus, to satisfy the initial constraint in Equation (9), the \( R_s^{s+1} \) needs to be projected, for every stencil, on the special orthogonal group.

A way to deal with such a problem, is to solve the so-called Wahba’s problem.\(^{32}\) The latter seeks for a rotation matrix between two coordinate systems from a set of weighted vector observations. Assuming that all \( \tilde{x}_i^{s+1} \) are known at the current step, which in our case are the nodal position vectors computed after the minimization of Equation (16), the rotation matrix \( R_s \) of any stencil \( s \in S \) is computed by minimizing the stencil energy

\[
\begin{aligned}
W(R) &= \sum_{(i,j) \in \mathcal{E}} \mu_{i,j} \|[R_s^{s+1} (\tilde{x}_j - \tilde{x}_i)] - ([\tilde{x}_j^{s+1} - \tilde{x}_i^{s+1}])\|^2 \\
\text{s.t.} \quad &R_s^{s+1}(R_s^{s+1})^T = I_n.
\end{aligned} \tag{26}
\]

According to Wahba’s problem, the minimization of Equation (26) is related to the orthogonal Procrustes problem, which is to find the orthogonal matrix \( R_s \) that is closest to \( B \), where \( B = \sum_i \mu_{i,j} ([\tilde{x}_j^{s+1} - \tilde{x}_i^{s+1}])^T (\tilde{x}_j - \tilde{x}_i) \), in the sense of Frobenius norm

\[
\|[R_s^{s+1} - B]^2 = \|[R_s^{s+1}]|^2 + \|[B]|^2 - 2\text{tr}(R_s^{s+1} B^T). \tag{27}
\]

Since \( R_s^{s+1} \) is a priori imposed to be an orthogonal matrix, with the proviso that the determinant is +1 (otherwise if the determinant is −1, it will still be orthogonal, but a reflection matrix), \( \text{tr}[R_s^{s+1}(R_s^{s+1})^T] = n \), where \( n \) is the dimension of the problem and matrix \( B \) is known. Hence, the term \( -2\text{tr}(R_s^{s+1} B) \) in Equation (27) should be minimized. In conclusion, \( R_s^{s+1} \) in nothing more than the polar factor of the matrix denoted as \( B \).

In the literature there are many ways to compute the polar factor of a matrix.\(^{33,34}\) A well-known method is the singular value decomposition (SVD), but is not recommended for performance reasons. After performing a wide set of simulations, it has been noticed that 5%–6% of the total runtime is consumed in computing the polar factor of the matrix \( B \). In our context, an iterative method is used based on Heron’s method\(^{35}\) for the square root of 1 (a.k.a. Newton method), which has been proved to be 2%–4% marginally faster CPU-wise than the SVD. These thoughts are also confirmed by Chao et al.\(^{36}\)
The Heron’s method does not ensure that the resulting orthogonal matrix will be of positive determinant. This means that, if the determinant is negative, the orthogonal matrix is a reflection matrix which is not desired in our case, since this leads to element inversion. Thus, SVD is used only when the determinant of the orthogonal matrix is negative so as to avoid such a case.

5 | ALGORITHM OF THE MORPHING FRAMEWORK

At this point, it is worth describing the algorithm to be followed in order to obtain the new/updated position of every internal node constituting the computational mesh. Since the problem expressed in Equation (11) is nonlinear, an iterative process needs to be followed to achieve convergence. Briefly, the necessary steps to be done in order to compute the final position of the internal nodes of the computational nodes are:

1. Create a graph which contains the stencils and the corresponding nodes belonging to them. A simple way to create the stencils is to define as many number of stencils as the number of the nodes and use distance-based criteria to find the neighbors. The number of neighbors that belong to the stencil is defined by a user defined radius. The bigger the radius it is, the more neighbors in the stencil are included. This step needs to be done only once, before starting the iterative process of computing the displacement of the nodes of the computational mesh.
2. Initialization of $\vec{x}_i^0 = \vec{X}_i, \forall i \in N$ and $R_s^0 = I_n, \forall s \in S$, where $I_n$ is the identity matrix.
3. Computation of the derivatives (Equation 19) required to build the system described in Equation (20). Solving the above-mentioned linear system (or the one that has been condensed, Equation (23)) with the use of least-squares method and computation of $(\vec{x}_i^{\nu+1}, R_s^{\nu+1})$ from $(\vec{x}_i^{\nu}, R_s^{\nu})$, namely, the nodal positions and an estimation of the rotation matrix, respectively. This step is the so-called prediction step.
4. Since this solution is obtained by linearizing the constraint in Equation (9), $R_s^{\nu+1}$ is not a rotation matrix that belongs to the orthogonal group SO($n$), but it belongs in the space which is tangent to this. Thus, it has to be projected back to SO($n$). In this step, the position vectors $\vec{x}_i^{\nu+1}$ are kept fixed and, the $R_s^\nu$ for every stencil are projected to the orthogonal group SO($n$) to obtain the closest rotation matrix $R_s^{\nu+1}$. This step is called correction step.
5. Check if steady state has been achieved, namely, if $\vec{X}_i, R_s$ have converged, namely, if the mesh is not changing anymore or in other words when $||\vec{x}_i^n - \vec{x}_i^{n-1}||_2 < \epsilon$, where $\epsilon$ is a very small threshold. Since the rotation matrix is analytically computed, it will also be converged if the vector $\vec{x}_i$ is also converged. If not, the iterative process continues returning to step 2 until convergence is achieved.

6 | APPLICATIONS

In this section, the FT-R3M proposed in this article, is being demonstrated and tested as a standalone tool to show its efficiency in some test cases. Metrics are used to quantify the grid quality of the initial and the final computational mesh. More specific, the value of the skewness, the nonorthogonality (those used by OpenFOAM) and the aspect ratio are used.

For the sake of completeness, it is worth mentioning that the nonorthogonality metric (to be denoted as N-O) measures the angle between the line connecting two cell centers and the normal of their common face (the lower the better). The skewness (to be denoted as SK) metric measures the distance between the intersection of the line connecting two cell centers with their common face and the center of that face (smaller is better). The nonorthogonality and skewness metrics are analyzed more detailed in the thesis of Jasak in pages 83 and 124, respectively. Last, for the aspect ratio (to be denoted as AR), the longest edge of an element is divided by the shortest one.

OpenFOAM is able to provide reliable simulations on meshes that the nonorthogonality of the cells of the computational mesh is smaller than 70. If the nonorthogonality is between 70 and 90, then special treatment is required, for example, nonorthogonal correctors. Regarding the skewness metric of the cells of the computational mesh, OpenFOAM provides solution with good/acceptable accuracy if the maximum value of the skewness metric in the mesh is smaller than 0.95. If the maximum value of the skewness in the computational mesh is bigger than the latter value, then difficulties in convergence may occur.
6.1 | Rotation of an isolated airfoil

The first case deals with the rotation of a two-dimensional (2D) isolated airfoil anticlockwise around its center. In this case, the far-field boundary of the airfoil is kept fixed, namely, the displacement field on that is 0. The computational mesh is appropriate for viscous flow and consists approximately of 32K nodes and 60K elements, with a boundary layer near the walls (quadrilateral elements) and triangular elements everywhere else.

The initial mesh and shape of the airfoil is demonstrated in Figure 2. The rotation of the airfoil takes place every 5° so as to keep track of the quality metrics for each position of the airfoil. However, the morphing procedure is performed between the initial shape and the desired position of the airfoil, since no subdivision of the displacement field is needed when the FT-R3M is used.

The airfoil is demonstrated in Figure 5 in various positions. A focus has been made in the leading and trailing edge on the resulting mesh in Figure 6, after 90° of rotation, for demonstration purposes and to show that the orthogonality of the boundary layer is preserved.

The history of the aforementioned grid quality metrics is shown in Figure 3. It is shown that FT-R3M preserves the grid quality after the deformation, even in extreme cases like that. In the latter, each one from the aforementioned metrics is normalized with corresponding one in the initial state of the airfoil. From Figure 3, we can observe that, the aspect ratio (mean value, maximum value, and SD) and the skewness are not changing significantly during the deformation. On the other side, the maximum and the mean value of the nonorthogonality of the elements is increasing (becoming worse) as the degrees of rotation are growing. This is reasonable since the deformation of 90° is extreme. However, even in such deformation there is no occurrence of any inverted element.
Indicatively, the cost of the morpher on this specific case it is compared with the cost that is needed to run the equivalent flow equations (EFS). If \( M \) is the cost of solving the EFS, then the cost of the morpher (e.g., for 10° of rotation) is 0.05\( M \). At this point, it worth mentioning that the cost of the EFS w.r.t. the cost of morpher is big since the flow is turbulent. Moreover, the cost of the morpher also depends on the nature of the displacement field. In other words, the bigger the rotation of the elements caused by the movement of the boundaries it is, the heavier (computationally) the system to be solved becomes.

Regarding the quality metrics and more specific the aspect ratio (normalized value) of the cells belonging to the boundary layer, these are tabulated on Figure 4. In this figure it is shown that the aspect ratio of the cells of the boundary layer is preserved during the deformation. The mean value, the SD and the maximum value of the aspect ratio of the cells of the boundary layer is changed approximately 0.0012% during the deformation of the airfoil in comparison to the initial one, whereas the minimum value of the aspect ratio has changed approximately 0.010% in comparison to the initial one (Figures 5 and 6).

### 6.2 Deformation of a duct

The second case is dealing with the deformation of an S-Bend duct (Figure 7) which is an air duct test case provided by VolksWagen AG. The displacement field has been provided by the NTUA group after performing an adjoint-based shape optimization for minimum total pressure losses across the duct. The flow is laminar, with inlet velocity \( U_{in} = 0.1 \text{m/s} \), normal to the boundary, leading to a Reynolds number of approximately \( Re = 400 \), based on the inlet hydraulic diameter. The mesh is comprised of approximately 470K hexahedrals and 480K nodes.

The final shape of the duct is the optimized one with approximately 60% less total pressure losses than the initial one. The challenge on this case is to deform the mesh from the initial to the optimized state and maintain a good quality. The final shape of the duct is shown in Figure 8 whereas in Figure 9, cut of the final shape is presented in order to demonstrate the displacement of a few of the internal nodes of the computational mesh.

In addition, quality metrics for the initial and the final mesh are tabulated in Table 1 to demonstrate the efficiency of FT-R3M. The CPU cost of the deformation of the duct is approximately 0.13\( M \). Furthermore, in Figure 10, the streamlines of the initial and the deformed/optimized duct are presented. In the initial geometry, there is a high recirculation area that takes place in the part of the duct which is near to the outlet. This has the biggest contribution to the total pressure losses, which is not the case in the optimized geometry, where the streamlines are smoother, hence the reduced total pressure losses.

### 6.3 Deformation of a beam

The next case, in which FT-R3M is used as the mesh morphing tool, deals with the deformation of a beam with a rectangular cross-section (Figure 11). The beam fixed at a wall from the one side and the purpose of this application is to apply...
FIGURE 5  Airfoil case: The resulting mesh in 25°, 50°, 75°, and 90° anticlockwise [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 6  Airfoil case: Focus on the leading (left) and trailing (right) edge of the rotated airfoil (90°). The orthogonality of the boundary layer in the rotated airfoil has been preserved which is very crucial for viscous flows [Colour figure can be viewed at wileyonlinelibrary.com]

extreme deformations and adapt the mesh to the updated boundaries (new shape of the beam) and check its grid quality. The unstructured mesh has been generated using CFD-GEOM® (ESI’s grid generator) and consists, approximately, of 195K elements (tetrahedrals) and 79K nodes.

The deformations applied in this test case are extreme and the reason is to show the efficacy of FT-R3M. Thus, a twisting, a bending deformation and a combination of them is applied to the right surface of the beam. Quality metrics that used in previous chapters are also used here and are quantifying the quality of the resulting grid. The aforementioned metrics are normalized with the corresponding one of the initial mesh and tabulated in Table 2. In Figures 12, 13, and
14, the beam is presented in various positions. The presence of the wall is just for demonstration reasons and it does not affect the morphing process. It indicates only which side of the beam is fixed at the wall.

7 | CONCLUSIONS AND DISCUSSION

The mesh morphing/movement tool for adapting the computational mesh to the updated boundaries was developed in this article and was shown to be capable of handling big deformation fields very efficiently. It was shown that, the computation of the final position of each node of the computational mesh requires the solution of a linear system, Equation (21)
and, a projection of the estimation of the rotation matrix of each stencil into the special orthogonal group SO(n) till convergence is achieved.

The energy functional, related to the deformation energy, between the initial state and the final one is minimized so that, at the same time a good quality of the final mesh is ensured, according to several used quality metrics. Moreover, since FT-R3M requires the formation of the so-called stencils, which are practically cloud of points and do not depended on the grid connectivity, it makes the grid displacement tool a mesh-free/mesh-less method and can naturally handle any kind of mesh, consisting even of nonconvex cells and nonplanar faces.

The cases shown in this article demonstrate the efficiency of the FT-R3M in big/extreme deformation fields. Mesh anisotropies are intrinsically handled, by favoring rigidity in critical directions and, the orthogonality of the boundary layers are very well preserved. Such a tool is essential, for example, in aerodynamic shape optimization since during the optimization, the computational mesh needs to be adapted to the updated boundaries and most important to be of good quality. For instance, maintaining the mesh quality near the body, for example, the boundary layers, is

---

**TABLE 1** Sbend case: Quality metrics before and after the deformation of the duct

| Normalized quality metric | State avg AR | std AR | max AR | min AR | avg N-O | std N-O | max N-O | avg SK | std SK | max SK |
|---------------------------|--------------|--------|--------|--------|---------|---------|--------|-------|-------|-------|
| Before                    | 1.000        | 1.000  | 1.000  | 1.000  | 1.000   | 1.000   | 1.000  | 1.000 | 1.000 | 1.000 |
| After                     | 0.914        | 1.080  | 1.000  | 1.020  | 1.024   | 0.998   | 0.896  | 1.182 | 1.409 | 1.833 |

*Note:* Each quality metric is normalized with the corresponding one of the initial mesh. Abbreviations: AR, aspect ratio; N-O, nonorthogonality metric; SK, skewness.

**TABLE 2** Quality metrics for each case of the beam

| Normalized quality metric | Case          | avg AR | std AR | max AR | min AR | avg N-O | std N-O | max N-O | avg SK | std SK | max SK |
|---------------------------|---------------|--------|--------|--------|--------|---------|---------|--------|-------|-------|-------|
|                           | Initial       | 1.000  | 1.000  | 1.000  | 1.000  | 1.000   | 1.000   | 1.000  | 1.000 | 1.000 | 1.000 |
|                           | Twisting      | 1.018  | 1.039  | 1.009  | 1.022  | 1.039   | 1.019   | 1.000  | 0.996 | 1.017 |       |
|                           | Bending       | 1.009  | 1.013  | 1.012  | 1.002  | 1.022   | 1.018   | 1.031  | 1.000 | 0.998 | 0.985 |
|                           | Combination   | 1.014  | 1.031  | 1.037  | 0.999  | 1.039   | 1.035   | 1.025  | 1.000 | 0.997 | 1.036 |

*Note:* Each quality metric is normalized with the corresponding one of the initial mesh. Abbreviations: AR, aspect ratio; N-O, nonorthogonality metric; SK, skewness.
very important in high-Reynolds number or in heat transfer problems in order to have accurate results of the analysis tool.

Coupling FT-R3M with a gradient-based optimization algorithm, for example, with the adjoint method, might show the efficacy of the proposed mesh movement tool inside the optimization loop. For instance, since it preserves the quality of the mesh, the optimization can go further without the need of remeshing, since it is very common that the optimization stops because of low quality of the resulting mesh.
FIGURE 14  Beam case: Combination of twisting and bending deformation of the beam [Colour figure can be viewed at wileyonlinelibrary.com]

ACKNOWLEDGMENTS
This work has been conducted within the IODA ITN on: “Industrial Optimal Design using Adjoint CFD.” The first author is an Early Stage Researcher (ESR) in this project which has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No. 642959.

ORCID
Athanasios G. Liatsikouras https://orcid.org/0000-0002-9362-7296

REFERENCES
1. Papoutsis-Kiachagias E, Porziani S, Groth C, Biancolini M, Costa E, Giannakoglou K. Aerodynamic optimization of car shapes using the continuous adjoint method and an RBF Morpher. Paper presented at: Proceedings of the EUROGEN 2015, 11th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems; September 14-16, 2015; Glasgow, UK.
2. Reuther J, Alonso J, Rimlinger M, Jameson A. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. Comput Fluids. 1999;28(4-5):675-700.
3. Parenteau M, Laurendeau E, Carrier G. Combined high-speed and high-lift wing aerodynamic optimization using a coupled VLM-2.5 D RANS approach. Aerosp Sci Technol. 2018;76:484-496.
4. Yu Y, Lyu Z, Xu Z, Martins JR. On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. Aerosp Sci Technol. 2018;75:183-199.
5. Knupp P. Remarks on Mesh Quality. Technical Report. Albuquerque, NM: Sandia National Laboratories (SNL-SAND); 2007.
6. Knupp P. Label-invariant mesh quality metrics. Paper presented at: Proceedings of the 18th International Meshing Roundtable; 2009:139-155; Springer, New York, NY.
7. Niu J, Lei J, He J. Radial basis function mesh deformation based on dynamic control points. Aerosp Sci Technol. 2017;64:122-132.
8. Liu X, Qin N, Xia H. Fast dynamic grid deformation based on Delaunay graph mapping. J Comput Phys. 2006;211(2):405-423.
9. Wang Y, Qin N, Zhao N. Delaunay graph and radial basis function for fast quality mesh deformation. J Comput Phys. 2015;294:149-172.
10. Zhou X, Li S. A novel three-dimensional mesh deformation method based on sphere relaxation. J Comput Phys. 2015;298:320-336.
11. Brown D, Nadarajah S, Yang H, et al. A linear elasticity mesh mover with adaptive mesh element stiffening for multi-element unstructured grids. Paper presented at: Proceedings of the 2018 AIAA Aerospace Sciences Meeting; 2018:0917, Springer, New York, NY.
12. Batina J. Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. AIAA J. 1991;29(3):327-333.
13. Bottasso C, Detomi D, Serra R. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. Comput Methods Appl Mech Eng. 2005;194(39-41):4244-4264.
14. Farhat C, Degant B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. Comput Methods Appl Mech Eng. 1998;163:231-245.
15. Blom F. Considerations on the spring analogy. Int J Numer Methods Fluids. 2000;32(6):647-668.
16. Hansbo P. Generalized Laplacian smoothing of unstructured grids. Int J Numer Methods Biomed Eng. 1995;11(5):455-464.
17. Su Z, Wang S, Yu C, Liu F, Shi X. A novel Laplacian based method for mesh deformation. J Inf Comput Sci. 2010;7(4):877-883.
18. Zhang S, Huang J, Metaxas D. Robust mesh editing using Laplacian coordinates. Graph Model. 2011;73(1):10-19.
APPENDIX A. STRUCTURE OF THE ORTHOGONAL GROUP AND ITS TANGENT SPACE

Before going to the linearization of the total energy functional, it is worth mentioning a few things about the structure of the orthogonal group and its tangent space that will also be helpful to better understand the implicit quadratic constraint in Equation (11). Let us denote as

$$SO(n) = \{ R \in \text{GL}(n, \mathbb{R}) | R^T R = R R^T = I_n \}. \quad (A1)$$

where $\text{GL}(n, \mathbb{R})$ is the general linear group of degree $n$ (the multiplication of two invertible matrices is also an invertible matrix). Since, $\varphi(R)$ in Equation (10) is continuous, $SO(n)$ is a closed set. Let $R \in SO(n)$ and $B \in \text{GL}(n, \mathbb{R})$ be given and
\( \epsilon \in \mathbb{R} \), then it yields
\[
\varphi(R + \epsilon B) = (R + \epsilon B)(R + \epsilon B)^\top - I_n
\]
\[
= RR^\top - I_n + \epsilon(BR^\top + RB^\top) + \epsilon^2 BB^\top
\]
\[
= \varphi(R) + \epsilon(BR^\top + RB^\top) + \epsilon^2 BB^\top.
\]

\( \text{(A2)} \)

Hence, the Gâteaux derivative of \( \varphi \) is
\[
d\varphi(R, B) = \lim_{\epsilon \to 0} \frac{\varphi(R + \epsilon B) - \varphi(R)}{\epsilon}
\]
\[
= \frac{d}{d\epsilon} \varphi(R + \epsilon B)\bigg|_{\epsilon=0}
\]
\[
= BR^\top + RB^\top.
\]

\( \text{(A3)} \)

Since \( d\varphi(R, B) \) is linear w.r.t. B and continuous, it follows that \( \text{SO}(n) \) is a differentiable manifold and that its tangent space at \( R \) \((T_R \text{SO}(n))\) is the null-space of \( d\varphi(R, B) \):
\[
T_R \text{SO}(n) = \{ d\varphi(R, B) = BR^\top + RB^\top = 0_n \}
\]

\( \text{(A4)} \)

which means that the space tangent to the orthogonal group \( \text{SO}(n) \) at \( R \) is the space of linear transformations \( B \) such that \( RB^\top \) is skew-symmetric.

**APPENDIX B. EIGENSTRUCTURE OF THE COMPUTED MATRIX \( R_s^* \)**

In order to explain the importance of the projection of the matrix \( R_s^{k+1} \) (let us talk for iteration \( k+1 \)), Equation (25), onto the orthogonal group \( \text{SO}(n) \), we have to investigate its eigenstructure, starting from the incremental transformation \( I_n + \Omega_s \) in which \( \Omega_s \) is skew-symmetric. Since \( \Omega_s \) is real, it is also skew-hermitian. If \( \lambda \in \mathbb{C} \) is a complex valued eigenvalue of \( \Omega_s \) then there exists a nonnull vector \( \vec{v} \in \mathbb{C}^n \) such that:
\[
\Omega_s \vec{v} = \lambda \vec{v}.
\]

\( \text{(B1)} \)

Then, we have:
\[
\lambda \| \vec{v} \|^2 = \lambda \vec{v}^\top \vec{v} = \vec{v}^\top(\lambda \vec{v}) = (\Omega_s \vec{v})^\top \vec{v} = -(\Omega_s \vec{v})^\top \vec{v}
\]
\[
= -(\lambda \vec{v})^\top \vec{v} = -\lambda \| \vec{v} \|^2
\]
\[
\Rightarrow \lambda \| \vec{v} \|^2 = -\lambda \| \vec{v} \|^2.
\]

\( \text{(B2)} \)

Hence, \( \lambda = -\overline{\lambda} \), which means that the eigenvalues \( \lambda \) of the matrix \( \Omega_s \) are pure imaginary. Furthermore, every nonreal eigenvalue of a real matrix comes in pair with its complex conjugate. Taking into account all the aforementioned and the fact that \( \det(R_s^k) = 1 \) (since it is a rotation matrix) then, the determinant of the matrix \( R_s^{k+1} \), Equation (25), is computed as follows:
\[
\det(R_s^{k+1}) = \det[(I_n + \Omega_s^{k+1})R_s^k]
\]
\[
= \prod_{\lambda \in \text{Sp}(\Omega_s^k)} 1 + \lambda
\]
\[
= \prod_{\alpha \in |\text{Sp}(\Omega_s)|} (1 + i\alpha)(1 - i\alpha)
\]
$$= \prod_{\alpha \in \text{Sp}(\Omega_s)} 1 + \alpha^2 \geq 1 > 0. \quad \text{(B3)}$$

Looking at the proof stated at (B3), it is shown that the determinant of the matrix $R_s^{k+1}$ is greater or equal to 1. This means that the aforementioned matrix needs projection to the special orthogonal group $\text{SO}(n)$, otherwise it is not a rotation matrix.

**APPENDIX C. RELATION BETWEEN FT-R3M AND DEFORMATION ENERGY**

In this section, the relation between the FT-R3M functional and the deformation energy is provided. For the sake of completeness we repeating the FT-R3M functional

$$E = \sum_{s \in S} \sum_{(i,j) \in s} \mu_{s,i,j} \|R_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i)\|^2 \quad \text{s.t.} \quad R_sR_s^T = I_n, \quad \forall s \in S. \quad \text{(C1)}$$

For a given stencil, being given the final position vector $\vec{x}_i$, $\forall i \in \mathbb{N}$, let us introduce the matrix $F_s$ as the constraint-free minimal of

$$E = \sum_{(i,j) \in s} \mu_{s,i,j} \|F_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i)\|^2. \quad \text{(C2)}$$

Assuming that the material points $\vec{x}_i$ are experiencing a continuous deformation $\vec{g}$, then $F_s$ is the best approximation, in the least squares sense, of the corresponding Deformation Gradient

$$\left( \frac{D\vec{g}}{D\vec{X}} \right) = \left[ \frac{\partial \vec{g}}{\partial \vec{X}} \right]. \quad \text{(C3)}$$

Now, introducing the formerly defined Deformation Gradient $F_s$, in the standard energy functional of FT-R3M in Equation (C1) it yields

$$E' = \sum_{s \in S} \sum_{(i,j) \in s} \mu_{s,i,j} \|(R_s - F_s + F_s)(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i)\|^2 \Rightarrow$$

$$E' = \sum_{s \in S} \sum_{(i,j) \in s} \mu_{s,i,j} \|(R_s - F_s)(\vec{X}_j - \vec{X}_i) + F_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i)\|^2 \Rightarrow$$

$$E' = \sum_{s \in S} \sum_{(i,j) \in s} \mu_{s,i,j} \|(R_s - F_s)(\vec{X}_j - \vec{X}_i)\|^2 + \sum_{s \in S} \sum_{(i,j) \in s} \mu_{s,i,j} \|F_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i)\|^2$$

$$+ 2 \sum_{s \in S} \sum_{(i,j) \in s} \mu_{s,i,j} \|\vec{X}_j - \vec{X}_i\|^T (R_s - F_s)^T [F_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i)]^2. \quad \text{(C4)}$$

In Equation (C4), the first term is the so-called deformation energy, which measures the distance between the deformation gradient and the nearest rotation and, the second one corresponds to second-order derivatives. The last term is zero due to the optimal condition of $F_s$ in Equation (C2).