Interactive editing of 3D geological structures and tectonic history sketching via a rigid element method

Gautier Laurent, Guillaume Caumon, Mark Jessell

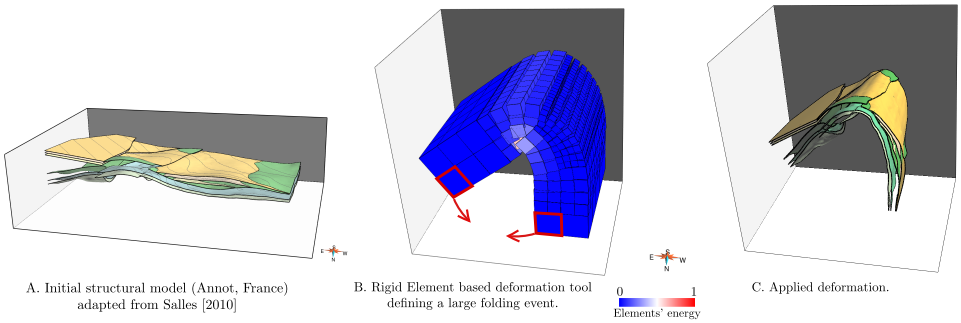
Author's Preprint – Published in Computers & Geosciences
10.1016/j.cageo.2014.10.011

Interactive Editing of 3D Geological Structures and Tectonic History Sketching via a Rigid Element Method

Gautier Laurent[1,2,*], Guillaume Caumon[1], and Mark Jessell[3,4]

[1]*GeoRessources (UMR 7359), Université de Lorraine-ENSG, CNRS, CREGU Campus Brabois, TSA 70605, 54518 Vandoeuvre-lès-Nancy Cedex, France*
[2]*School of Earth, Atmosphere and Environment, Monash University PO Box 28E, Victoria, Australia.*
[3]*University of Western Australia, Centre for Exploration Targeting, M006, 35 Stirling Highway, Crawley WA 6009 Australia.*
[4]*IRD, UR 234, GET, OMP, Université Toulouse III, 14 Avenue Edouard Belin, 31400 Toulouse, France.*
[*]*Corresponding author: gautier.laurent@univ-orleans.fr*

Graphical Abstract



A. Initial structural model (Annot, France) adapted from Salles [2010]

B. Rigid Element based deformation tool defining a large folding event.

C. Applied deformation.

Contents

Author's Preprint – Published in Computers & Geosciences
10.1016/j.cageo.2014.10.011

Interactive Editing of 3D Geological Structures and Tectonic History Sketching via a Rigid Element Method

Gautier Laurent[1,2,*], Guillaume Caumon[1], and Mark Jessell[3,4]

[1]*GeoRessources (UMR 7359), Université de Lorraine-ENSG, CNRS, CREGU Campus Brabois, TSA 70605, 54518 Vandoeuvre-lès-Nancy Cedex, France*
[2]*School of Earth, Atmosphere and Environment, Monash University PO Box 28E, Victoria, Australia.*
[3]*University of Western Australia, Centre for Exploration Targeting, M006, 35 Stirling Highway, Crawley WA 6009 Australia.*
[4]*IRD, UR 234, GET, OMP, Université Toulouse III, 14 Avenue Edouard Belin, 31400 Toulouse, France.*
[*]*Corresponding author: gautier.laurent@univ-orleans.fr*

Highlights

- ▷ We propose specifications for deformation tools for 3D geological structure modelling.
- ▷ We select a deformation algorithm based on rigid elements and deformation energy.
- ▷ We propose a process for modelling structures with several folding events.
- ▷ We implement boundary conditions for geological structure restoration.
- ▷ We implement anisotropy and local volume conservation rules.

Abstract Numerical models of geological structures are generally built with a geometrical approach, which lacks an explicit representation of the deformation history and may lead to incompatible structures. We advocate that the deformation history should be investigated and represented from the very first steps of the modelling process, provided that a series of rapid, interactive or automated, deformation tools are available for local editing, forward modelling and restoration. In this paper, we define the specifications of such tools and emphasise the need for rapidity and robustness. We briefly review the different applications of deformation tools in geomodelling and the existing deformation algorithms. We select a deformation algorithm based on rigid elements, first presented in the Computer Graphics community, which we refer to as *Reed*. It is able to rapidly deform any kind of geometrical object, including points, lines or volumes, with an approximated mechanical behaviour. The objects to be deformed are embedded in rigid cells whose displacement is optimised by minimising a global cost function with respect to displacement boundary conditions. This cost function measures the difference in displacement between neighbouring elements. The embedded objects are then deformed based on their original position with respect to the rigid elements. We present the basis of our implementation of this algorithm and highlight its ability to fulfil the specifications we defined. Its application to geomodelling specific problems is illustrated through the construction of a synthetic structural model of multiply deformed layers with a forward modelling approach. A special boundary condition adapted to restore large folds is also presented and applied to the large anticline of Han-sur-Lesse, Belgium, which demonstrates the ability of this method to efficiently perform a volumetric restoration without global projections.

Keywords
3D Modelling
Fold
Interactive Deformations
Rigid Elements
Forward Modelling
Restoration

1 Introduction

3D models of geological structures are a fundamental ingredient of natural resource characterisation. Indeed, accurate modelling of geological structures is crucial for the proper characterisation of the geometry and the connectivity of rock volumes. Such structural models are generally obtained by a static and deterministic procedure, which relies on geometrical compatibility with geophysical data and well interpretations [Mallet, 2002; Caumon et al., 2009]. Kinematical and mechanical consistency of the model can be achieved when dense, high quality data are present and through appropriate decisions during modelling, but is not enforced by the modelling algorithms themselves.

Restoration of the geological model is a way to check these aspects *a posteriori*, and also to provide critical insights into the tectonic history, but computation time and modelling effort may be prohibitive [Moretti, 2008; Durand-Riard et al., 2010, 2013]. When inconsistencies are detected, the model may have to be edited or rebuilt, which may lead to significant additional modelling efforts. Taking the deformation history into account at an early stage of the 3D structural modelling appears to be a natural way to produce geologically consistent models, which should reduce the need for quality control.

When building a model, several tectonic scenarios may have to be rapidly tested, which calls for tools that are simple, efficient and robust enough to be used in an automated process, while allowing for rapid user feedback and editing. In this paper, we introduce new specifications for a general dynamic deformation tool to let geologists sketch and rapidly test deformation scenarios on 3D geomodels, for example to build synthetic 3D geometries based on a succes-

sion of events (as in Jessell and Valenta, 1996), restore a 3D geomodel or edit a 3D structural interpretation (Section 2). These specifications are based on two key ideas:

1. The scale at which the deformation has to be modelled is generally larger than the scale of the model. Depending on the application, the scale and resolution of a geological model may depend on lithological and stratigraphic heterogeneities, and on local geometrical peculiarities (*e.g.* thin stratigraphic layers, acute angular contacts, local features such as persistence of paleo-relief). The deformations that are taken into account in the modelling generally impact the global shape of the geological model while preserving these detailed features. Therefore the wavelength of the deformation may be considerably larger than the size of the mesh, which is used to describe the geometry of the model.
2. Because the full deformation path is generally poorly constrained, the interactivity of the deformation tool may be more important than its physical accuracy. Certainly, accurate simulation of rock deformations are required in certain applications (*e.g.* in studies aiming at improving theoretical understanding of tectonic principles), however when few data are available while building quantitative structural models, the benefit of an accurate simulation would be limited by the associated uncertainties. On the contrary, in an exploration context, rapidly testing several deformation scenarios with approximate simulations could be a useful step before performing more precise, but more expensive, simulations.

To meet these specifications, we propose to use a dynamical deformation model (Section 3), first presented in the Computer Graphics community [Botsch et al., 2007]. It consists of a spatial discretisation with rigid polyhedrons that are linked by a non-linear cost function similar to the elastic mechanical energy. This cost function measures the difference in displacement between neighbouring elements. Boundary conditions are set on certain elements in order to honour typical subsurface data. The final deformation is obtained by optimising the displacements with respect to the cost function. The displacement field is finally applied to the geological objects embedded into the rigid elements.

In Sections 4 and 5, we show a few examples of the capabilities of this deformation algorithm to build, edit and restore geological structures. We also illustrate how this approach can be applied to a complete forward modelling exercise. With this approach, 3D structural models would then be produced by successively applying the effects of kinematic events on an assumed initial geometry until the observations of the current structures are honoured [Jessell and Valenta, 1996]. The underlying tectonic history of resulting models is thus explicitly controlled by the interpreter and can be used to study structural uncertainties.

2 Deformation algorithms in geomodelling

In this section, we analyse the different uses of deformation algorithms for modelling geological structures and point out the need for a new range of algorithms which reinforce robustness and interactivity compared to physical accuracy. We derive new specifications from these observations and briefly review existing algorithms.

2.1 The uses of deformation computation for structural modelling

Deformation tools are, or may be, used at different stages throughout the construction of a 3D structural model:

Restoration: this process aims at assessing the compatibility of a given structural model by unfolding and unfaulting the geological structures, which also provides precious insights into their possible paleo-geometry. Historically, the first approaches to restoration aimed at restoring cross-sections based on geometrical criteria [Chamberlin, 1910; Dahlstrom, 1969]. Since then, this method has been extensively developed and adapted to 3D geological structures through a variety of approaches: continuum theory [Mallet, 2002; Massot, 2002], surface unfolding [Gratier et al., 1991; Williams et al., 1997; Egan et al., 1997; Rouby et al., 2000; Griffiths et al., 2002], mechanical methods [Maerten and Maerten, 2006; Moretti et al., 2006; Muron, 2005]. The application of 3D kinematic methods is relatively easy but relies on assumptions about the deformation mode (*e.g.* simple shear or flexural slip) and relies on global projections to approximate volumetric deformations. In contrast, geomechanical restoration is much more flexible but its application is more difficult due to preliminary conformal meshing and the possibly large number of mechanical parameters to be defined. In this paper, we will show how a light volumetric deformation tool can be used to fill the gap between these methods (Section 5.3).

Forward History Modelling: deformation algorithms may also be used to go forward in time. Starting from a hypothetical initial geometry, several kinematic deformation events can be progressively applied to pre-existing geometries. This method allows us to build very complex 3D structures with historical and geological control [Jessell and Valenta, 1996; Caumon et al., 2013]. Advanced methods for physically-based forward modelling of large deformations also exist, for instance Arbitrary Lagrangian-Eulerian schemes [Allken et al., 2011; Loiselet et al., 2010].

Local Editing: deformation may be applied at a more local scale to modify the geometry of a model, for example for integrating new data or interpretations. It generally involves locally rebuilding a part of the model after modifying some data or constraints [Grosse, 2002; Caumon et al., 2004; Tertois and Mallet, 2007].

Uncertainty Space Exploration: the deformation applied to a model may be used to explore the related uncertainties, for example by adding faults and consistently deforming surrounding objects [Georgsen et al., 2012; Cherpeau et al., 2012; Laurent et al., 2013]. A history matching process, ensuring the consistency with production data, may also be applied to reduce the geological uncertainties, which implies editing the model in order to minimise the production misfit. This process generally focuses on the rock properties, but it can also include structural aspects [Cherpeau et al., 2012; Seiler et al., 2010]. These validation steps come at a high computational cost and require several iterations of the modelling procedure. As a result of the uncertainties associated with the available data, their interpretation and the geological processes, several possible models should be produced to properly explore the domain of uncertainties and the relevant deformation scenarios. De-

formation algorithms can then be used to ease the construction of new possible models by modifying existing ones.

2.2 Specifications for a light geological deformation algorithm

Deformation algorithms are principally used in geology to run precise geomechanical simulations, either for restoration [Durand-Riard et al., 2010; Maerten and Maerten, 2006; Moretti et al., 2006] or forward modelling [Burbidge and Braun, 2002; Schmid et al., 2008; Loiselet et al., 2010; Allken et al., 2011]. These simulations rely on information about the mechanical material and detailed geometry, which may be hypothetical or unavailable at the early stage of structural model construction. In fact, for most of the applications listed in the previous section, the physical accuracy may not be the most important aspect due to the lack of observations.

This observation leads us to define the following specifications for a light generic deformation tool for geological editing and deformation scenario testing, with a relaxed constraint on physical accuracy. The following criteria are presented in order of importance:

Interactivity: the algorithm has to be fast enough to allow interactive deformation. For example, it should provide a result in at most a couple of seconds when directly manipulated by the user for editing a geological model. This way the user could interactively modify the boundary conditions while the deformation is recomputed. For more complex deformation, it has to allow us to explore different deformation scenarios in an acceptable time by iterative algorithms such as response surfaces and stochastic approaches. This would require us to compute hundreds of deformation scenario in a few minutes.

Adapted Scale: the deformation has to be computed at a scale relevant of its dimension. The deformation algorithm should not be affected by small-scale heterogeneities and geometrical details unless they lead to localisation phenomena.

Robustness: the deformation computation has to stay interactive and avoid breaking down even under extreme deformations. In particular, constitutive elements used for computation (typically triangles or tetrahedrons) should never degenerate during the process, *i.e.* their geometry and neighbourhood relationships should be maintained allowing the optimisation process to continue.

Topological Self-Consistency: the deformation has to affect all the structural objects contained in the model without perturbing their structural relationships. For discontinuities like faults and unconformities, the requirement to preserve these characteristics depends on the application context.

Parsimony: the 3D model has to be manipulated through a small set of parameters, both to simplify user interaction and to limit the problem of empty space for inverse approaches.

Physical Consistency: the deformations produced have to be kinematically consistent, *i.e.* honour possible natural displacement fields for both small and large deformations, while minimising deformations, avoiding self-intersections and local collapses or volumetric inversions. Ideally, the implied mechanical behaviour should also be consistent with natural geological materials, a condition that may be relaxed because of the large uncertainties about initial geometries and materials involved in the deformation. A first order compatibility may be enough, in particular because relaxing this constraint allows us to potentially gain in interactivity and robustness.

Extensibility: the algorithm's formalism has to remain adaptable to geological discontinuities, such as faults, and the specific mechanical behaviour of rocks (*e.g.* transverse isotropic).

Reversibility: the algorithm should be able to reverse the deformations it produces. This does not imply that the deformation has to be physically reversible, but the algorithm should be able to go backward as long as all the parameters of the deformation are known. This is necessary to interactively add and remove deformation components.

2.3 Overview of physically-based deformation algorithms

A large variety of deformation algorithms exists in the literature, mainly because of the diversity of applications and the variability of deformation behaviour. Here, we focus on mechanical models that are based on some physical considerations allowing us to control the kinematical and mechanical aspects of the deformation.

In the context of geomodelling, the most common deformation methods are the *Finite Element Method* (FEM), mainly for restoration purposes [Durand-Riard et al., 2010; Maerten and Maerten, 2006; Moretti et al., 2006; Muron, 2005; Schmid et al., 2008; Siavelis, 2011; Vidal-Royo et al., 2012], the *Discrete Element Method* (DEM), mainly for studying rock deformation and tectonics [Burbidge and Braun, 2002; Scholtès and Donzé, 2013; Vidal-Royo et al., 2011], and *Arbitrary Lagrangian-Eulerian Methods* (ALE). They generally aim at producing an accurate simulation of physical rock behaviour.

In a different context, the Computer Graphics community proposes a very different view of deformation computation: it has to be accurate enough to be realistic but fast enough to be interactive. In other words, the requirement of interactivity and robustness may come before the physical accuracy of the model. This philosophy is very close to the requirements for a light geological deformation algorithm presented in this paper.

A good review of the physically-based deformation algorithms is presented in Nealen et al. [2006]. They can be split into three main categories:

Continuum Mechanics Based Methods: these methods use different partitions of the space to solve the physical equations governing the deformation phenomenon described by continuum mechanics. The principal methods in this category are the Finite Difference Method (FDM), the Finite Element Method (FEM) and the Boundary Element Method (BEM). As these methods rely on actual physical laws, they generally converge towards the actual solution when the computation support is sufficiently fine.

Simplified Medium Methods: they approximate the deformed space with elements that deform with a simplified behaviour. For example, Mass-Spring Systems use a set of masses linked together by springs whose behaviour are described by very simple equations. Discrete Element Methods (DEM) also use this kind of simplification as they approximate the model by a set of particles interacting with

each other by contact forces [Burbidge and Braun, 2002; Scholtès and Donzé, 2013; Vidal-Royo et al., 2011]. As these methods use an approximation of the physical laws, there is no guarantee they converge towards the physical solution.

Space Deformation Methods: these methods decouple the problem of computation of the deformation from the representation of the object to be deformed. The first space deformation methods were mainly driven by user manipulation [Gain and Bechmann, 2008], without constraints on the physical accuracy, *e.g.* Free Form Deformation (FFD) [Barr, 1984; Sederberg and Parry, 1986] and Simple Constrained Deformation (SCODEF) [Borrel and Rappoport, 1994]. Subsequently, additional behaviours have been added to the space deformation approaches, *e.g.* by controlling the volume variations [Hirota et al., 2000; Teschner et al., 2004; Von Funck et al., 2006].

Two characteristics appear to be very important in the context of this paper: the space deformation, which meets the *Space Consistency* and *Adapted Scale* requirements, and the physical approximation, which meets the *Interactivity over Physical Accuracy* requirements. Botsch et al. [2007] present an algorithm which shows these characteristics together with very interesting robustness capabilities. As it is based on rigid elements and deforms the structures that are embedded into its elements, we refer to this algorithm as *Reed* (Rigid Elements Embedded Deformation). This acronym is a metaphor for the robustness of this model (it bends, but does not break).

3 Presentation of the deformation model Reed

Reed is a deformation algorithm based on the optimisation of the position of rigid elements with respect to a cost function E, similar to linear mechanical energy. The main steps of this algorithm are (Fig. 1):

1. Encapsulating the objects to be deformed into rigid elements (Fig. 1B).
2. Applying boundary conditions on some elements (Fig. 1C).
3. Iteratively optimising the rigid motion of the elements by minimising E (Fig. 1D).
4. Applying the interpolated displacements to the embedded objects (Fig. 1E).

Most of the principles and equations presented in this section are adapted from Botsch et al. [2007]. However, the given details correspond to the developments of our implementation, whose formulation slightly differs from Botsch et al. [2007]. The principal difference relates to the description of the rigid element displacements. We consider the rotation components as a local rotation around the centre of the elements instead of a global rotation around the origin. This better separates translation and rotation increments and simplifies the process of computing rigid motion from the optimal linearised rigid displacements (see Section 3.3).

3.1 The encapsulation of deformed objects in rigid elements

The objects to be deformed are embedded in the rigid elements of the deformation model (Fig. 1B). All the displacement computations are made on these elements and the resulting interpolated displacement field is applied to the embedded objects at the end. This scheme simplifies the computation and guarantees the robustness of the approach for large deformations. The interpolation process is detailed in A.

When several objects are deformed, they are embedded in the same set of elements. Therefore, they are all deformed by the same displacement field, which makes it possible to preserve the relationships between the different deformed objects (provided that they are conformably discretised and that the discretisation is fine enough). The scale of *Reed* objects can be adapted to the scale of the deformation, independently of the scale of the geometrical details that are present in the structures to be deformed (Fig. 1), which reduces the number of elements required for the computation and allows fast computation.

The elements are displaced with a rigid motion (Fig. 2A), consisting of a rotation R around its centre c and a translation T:

$$M : \begin{array}{ccc} \mathbb{R}^3 & \longrightarrow & \mathbb{R}^3 \\ x & \longmapsto & c + \mathsf{R}(x - c) + \mathsf{T} \end{array} \tag{1}$$

For convenience, all the elements used in this paper are cuboid, however *Reed* can also be applied to arbitrary polyhedral elements made to conform to geological features such as faults. The shape of the elements is only shown to represent the volume of initial space covered by each element for the energy definition (Section 3.2). They have no "space-filling" property.

Most of the computations performed on these elements require information about neighbour relationships between elements. The connectivity between the elements is given explicitly and is kept during the deformation process, which avoids the need to re-evaluate each element's neighbourhood. However, *Reed* elements do not have an explicit neighbourhood pattern. The number of neighbours per face is not fixed. This allows for locally refining the discretisation by splitting some elements, which is useful for locally adapting the elements to the embedded objects.

3.2 The cost function coupling the elements

The rigid elements are linked together by a cost function (Fig. 3). To stay close to the notations of Botsch et al. [2007], this cost function is referred to as the coupling energy E_{ij}, even if it does not have the characteristics of energy in the strict sense.

E_{ij} is defined as the integral, over the pair of elements (C_i, C_j), of the difference between the displacements M_i and M_j, weighted by w_{ij} [Botsch et al., 2007]:

$$E_{ij} = \frac{1}{2}\, w_{ij} \int_{C_i \cup C_j} \left\| M_i(x) - M_j(x) \right\|^2 \mathrm{d}x \tag{2}$$

The weight w_{ij} for two adjacent elements depends on their respective volumes, V_i and V_j, on the contact area S_{ij} and on the distances h_i and h_j between each centre and S_{ij}. It also allows us to introduce stiffness variations by weighting the contribution of certain elements or connections to the global energy. This is obtained by multiplying the geometrical factors by a dimensionless stiffness weighting w^{st}:

$$w_{ij} = w^{st}_{ij} \frac{\mathrm{S}_{ij}}{\left(\mathrm{h}_i + \mathrm{h}_j\right)\left(\mathrm{V}_i + \mathrm{V}_j\right)} \tag{3}$$

where h_i and h_j are the distances between the contact surface and the element's barycentre. The total energy E

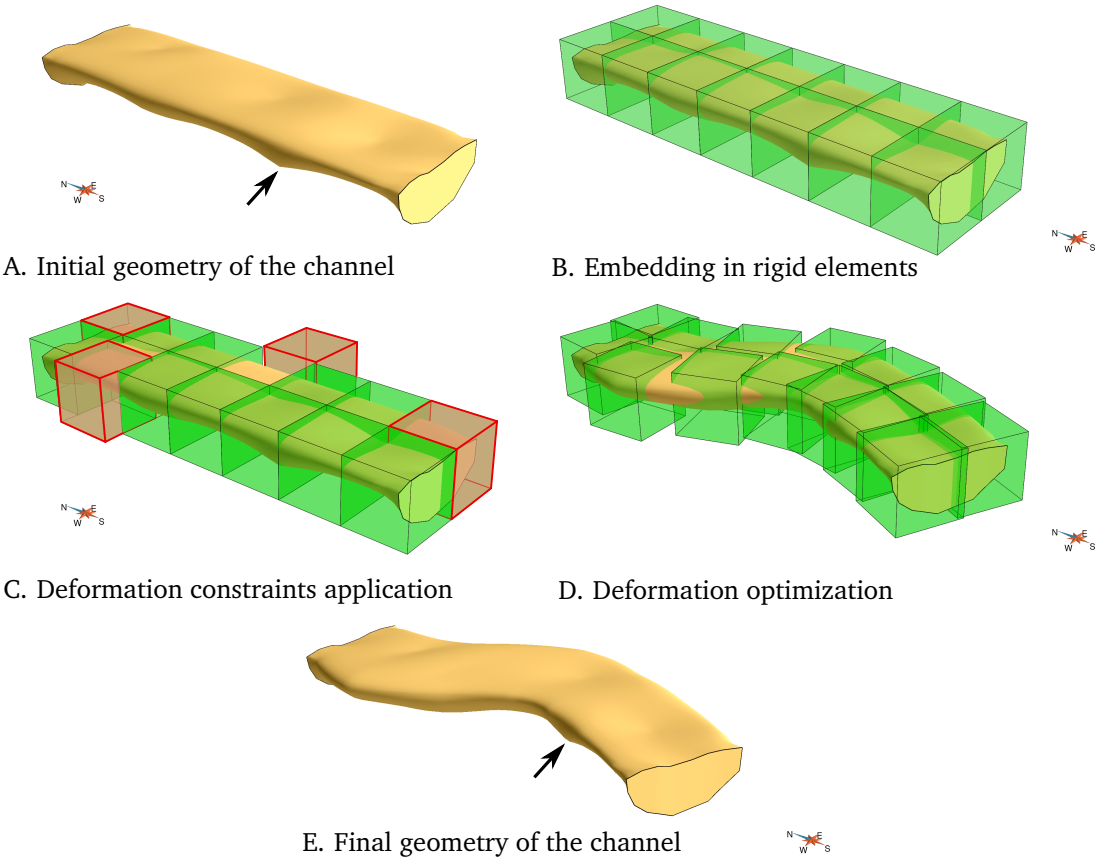


Figure 1 Rigid element embedding deformation approach. The approach is illustrated for the editing of the geometry of a fluvial channel. (A) The initial geometry shows a channel with a straight path. (B) The channel is embedded inside a *Reed* object made of 14 cubic rigid elements. (C) Some constraints (red) are applied to fix the position of certain elements. Only the translations are fixed here, the elements are still allowed to rotate. (D) The translation and rotation of the elements are optimised by globally minimising the energy of the *Reed* object (Fig. 4). (E) The displacement field is applied to the channel giving its final geometry. Note how the initial high-frequency details (*e.g.* the one pointed by the black arrow) of the geometry have been preserved.

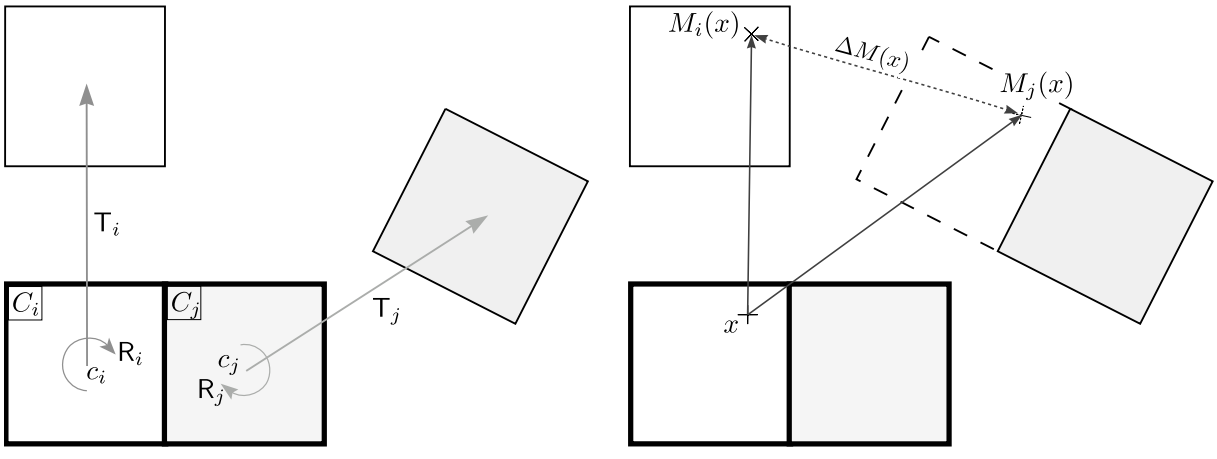


Figure 2 Rigid element displacement. (A) Two neighbour elements C_i and C_j (bold squares) are displaced (thin squares). Each element is moved by a rigid motion M, made of a rotation R around its center c and a translation T. (B) The displacement of both elements are applied to the point x yielding the final positions $M_i(x)$ and $M_j(x)$. The distance $\Delta M(x)$ between $M_i(x)$ and $M_j(x)$ represents the difference of displacement between C_i and C_j at point x. This quantity is the basis of the formulation of the coupling energy between neighbour elements (Eq. 2). The dashed square represents the position of C_i if it follows the displacement of C_j.

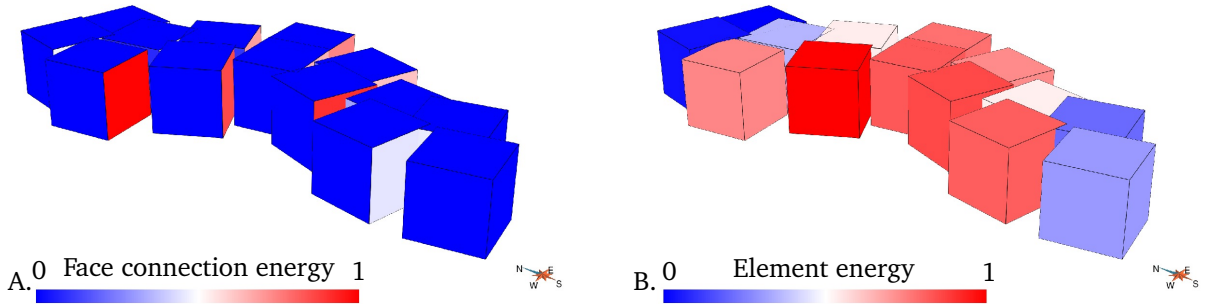


Figure 3 The energy linking the rigid elements. This illustration corresponds to the *Reed* object used to deform the synthetic channel presented in Fig. 1. (A) The coupling energy rendered per element faces. (B) The coupling energy rendered per element.

of the deformation model is obtained by summing all the coupling terms E_{ij}.

The *Reed* energy formulation shares some interesting common points with the restoration algorithm based on rigid triangles proposed by Gratier et al. [1991]. The optimal position of the triangles are obtained by minimising the difference of displacement between the vertices of each triangle and those of the hole left by its neighbours. *Reed*'s criterion slightly differs as it directly minimises the difference of rigid motion between each element and its neighbour.

3.3 The optimisation of the rigid element displacements

The deformation is driven by Dirichlet boundary conditions. These conditions are implemented by displacing elements and partly or fully fixing their displacement parameters. Similarly, as in Botsch et al. [2007], the displacement of the other elements are then optimised by a Newton-like iterative process to minimise the global energy E (Fig. 4). At each step k of this process, two kinds of displacements are considered for each element: the current rigid motion M^k and a displacement increment A^k. The descent direction is obtained by finding A^k such that E is minimised.

To overcome the non-linearity of E (Eq. 2), a linearised form of the displacement increment A is used instead of an actual rigid motion. The incremented displacements of an element C_i applied to a point x is thus approximated by $A_i\big(M_i(x)\big)$ (Fig. 4), where A_i is a combination of a translation vector v and a rotation vector ω, defining a rotation around the current centre of the element (*i.e.* after it has been affected by M):

$$A_i^k : \begin{array}{ccc} \mathbb{R}^3 & \longrightarrow & \mathbb{R}^3 \\ x & \longmapsto & x + v_i + \omega_i \times \big(x - M_i(c)\big) \end{array} \tag{4}$$

where $\times$ refers to the cross product, when applied to vectors.

The translation vector v_i and rotation vector ω_i are the unknown of the descent direction estimation process. For each element, they are gathered into an unknown vector U_i, which is of dimension 6×1. A global displacement unknown vector U is constituted by assembling the U_i of all the elements.

These approximated displacements are substituted to M in Eq. (2) yielding the linearised coupling energy E^*_{ij}:

$$E^*_{ij} = \frac{1}{2}\, w_{ij} \int_{C_i \cup C_j} \Big\| A_i\big(M_i(x)\big) - A_j\big(M_j(x)\big) \Big\|^2 \,\mathrm{dx} \tag{5}$$

Applying this equation to all the elements yields a second order polynomial of the global displacement vector U (detailed in B):

$$E^* = U^T Q\, U + R^T U + C \tag{6}$$

where Q, R and C are the polynomial coefficients of E^*. The optimal incremented displacement is then obtained by finding the minimum of E^*, by cancelling its first order derivative with respect to U:

$$QU + \frac{1}{2}R = 0 \tag{7}$$

Without appropriate boundary conditions, this linear system is not well posed as an infinite number of solutions exists. They correspond to all the rigid transformations of the initial position of the elements. A minimum of 6 degrees of freedom have to be constrained to produce a unique solution. More constraints can also be applied in order to impose a deformation to the *Reed* object. These constraints are set by fixing some degrees of freedom, *i.e.* setting some values of U to 0 depending on the boundary conditions to be applied. It allows us to remove the corresponding rows and columns of Q and R in Eq. (6). This yields the free terms Q_f and R_f, which are respectively of dimensions $n_f \times n_f$ and $n_f \times 1$, where n_f is the number of non-constrained degrees of freedom.

To compute the rigid motions of the next step M^{k+1}, the linearised displacement increment A^k is re-transformed into a rigid motion by computing a rigid rotation increment from ω: the rotation axis is the unit vector parallel to ω with the same orientation and the rotation angle is given by $\arctan\big(\|\omega\|\big)$. M^k is then updated by adding this rigid rotation increment to its rotation part and v to its translation part.

4 Testing specifications of Reed

This section inspects the capabilities of *Reed* to meet the specifications given in Section 2.2.

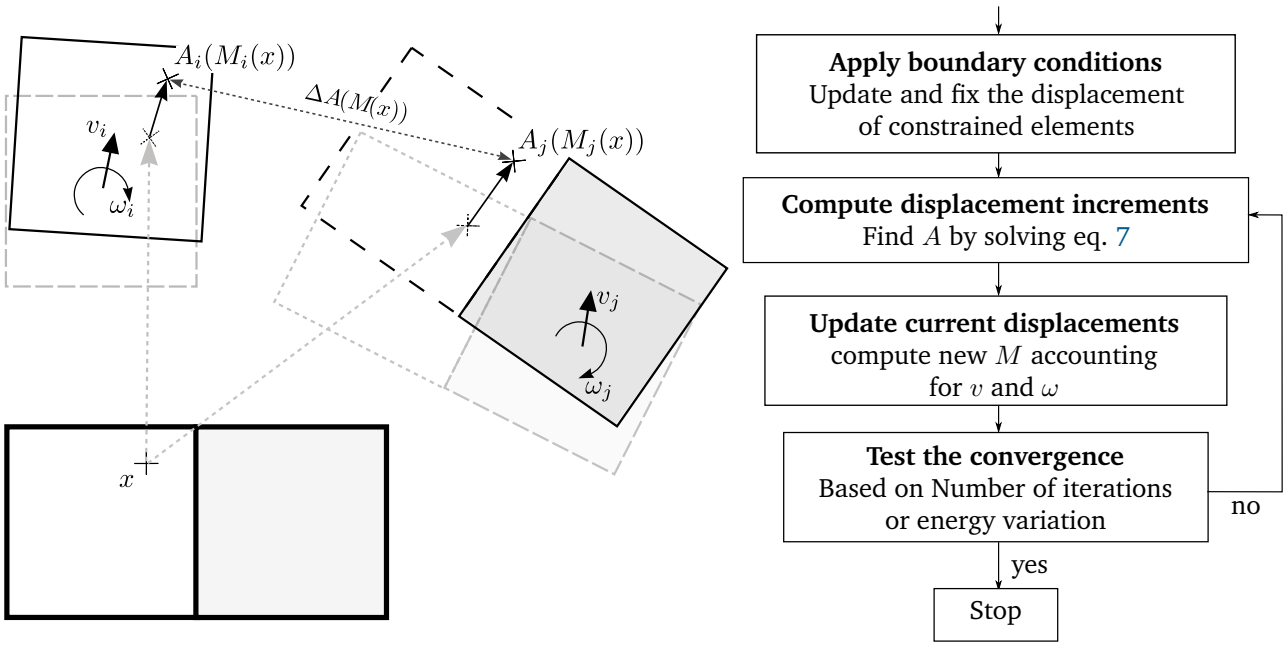


Figure 4 Optimisation process. The optimisation process consists in optimising the elements' position (see Fig. 2) by progressively applying small linear displacement increments A, made of a translation v and a rotation ω. During the optimisation process, the cost function (Eq. 5) penalizes the difference of incremented displacement $\Delta A\big(M(x)\big)$. The whole optimisation process is presented in the flow diagram (right). The rigid body motion M of each element is updated with optimal displacement increment until a convergence criterion is reached.

4.1 Robustness to extreme deformation

As a first application example, we deformed a model of the Annot Formation (SE France), adapted from Salles [2010]. It consists of a series of structural horizons with several faults and an unconformity, all represented by triangulated surfaces. *Reed* is applied to these structures with an increasing deformation intensity (Fig. 5). The results illustrate(1) the robustness to extreme deformation, (2) the conservation of the relationships between the different structural elements. The complete optimisation runs in about 6 seconds on a personal computer (2×2.66 GHz, 4 Go RAM) with about 1200 rigid elements.

This example also illustrates the algorithm's ability to locally refine the discretisation (Fig. 5D) to ensure a better deformation optimisation and interpolation. This refinement can be automated, for example by refining a cell when it has accumulated too much energy [Botsch et al., 2007].

In a second application we deformed a cross-section containing six layers having different stiffness w^{st} and with one refined layer (Fig. 6). They are laterally shortened by fixing the position of the elements located on the left extremity and by translating the elements of the right extremity. In their final state, the layers have been highly shortened to a degree that elements overlap. This configuration could cause the elements of other deformation algorithms to degenerate, which would make it impossible to continue computing deformations. With *Reed*, the elements are rigid and therefore cannot degenerate, which allows for additional iterations, which eventually results in a more reasonable deformation.

Also, even if the resulting configuration of the elements looks degenerate, there are no self-intersections of the deformed horizons (Fig. 6B). In extreme cases, the displacements obtained can produce inconsistent deformation of the deformed objects. This is the case if two neighbour elements invert their position or if non-neighbour elements collide. However, such a behaviour is highly penalised by the deformation energy. Such degenerate cases should then appear only when the applied constraints are inconsistent relatively to each other or inconsistent with the mechanical capabilities of the deformed objects. For example, natural extreme deformation may induce faulting, but our implementation is presently unable to dynamically introduce faults during the deformation.

The configuration observed on the final state of the deformation may look degenerate since many elements are penetrating their neighbour's volume (Fig. 6A). However, this configuration is not actually degenerate in the sense that neighbour elements do not invert their positions and non-neighbour elements do not collide. Consequently, no intersections occur in the deformed layers (Fig. 6B).

Another demonstration of *Reed*'s numerical robustness is shown by removing the constraints applied to the displaced elements (Fig. 6C). The rigid elements return rapidly to their initial position. This illustrates the capability of the system to continue computing even with extreme deformation and to rapidly adapt to boundary condition changes introduced by the user, which is crucial for its robustness and interactivity and for providing visual feedback to the user when boundary conditions change.

4.2 Kinematical and mechanical consistency

First-order kinematical and mechanical consistency is another crucial specification. It is investigated by comparing the resulting deformation with the results of a finite element method. It is important to note that these two methods are not based on the same equations, so they may not converge to the same result. FEM is used as a reference because it re-

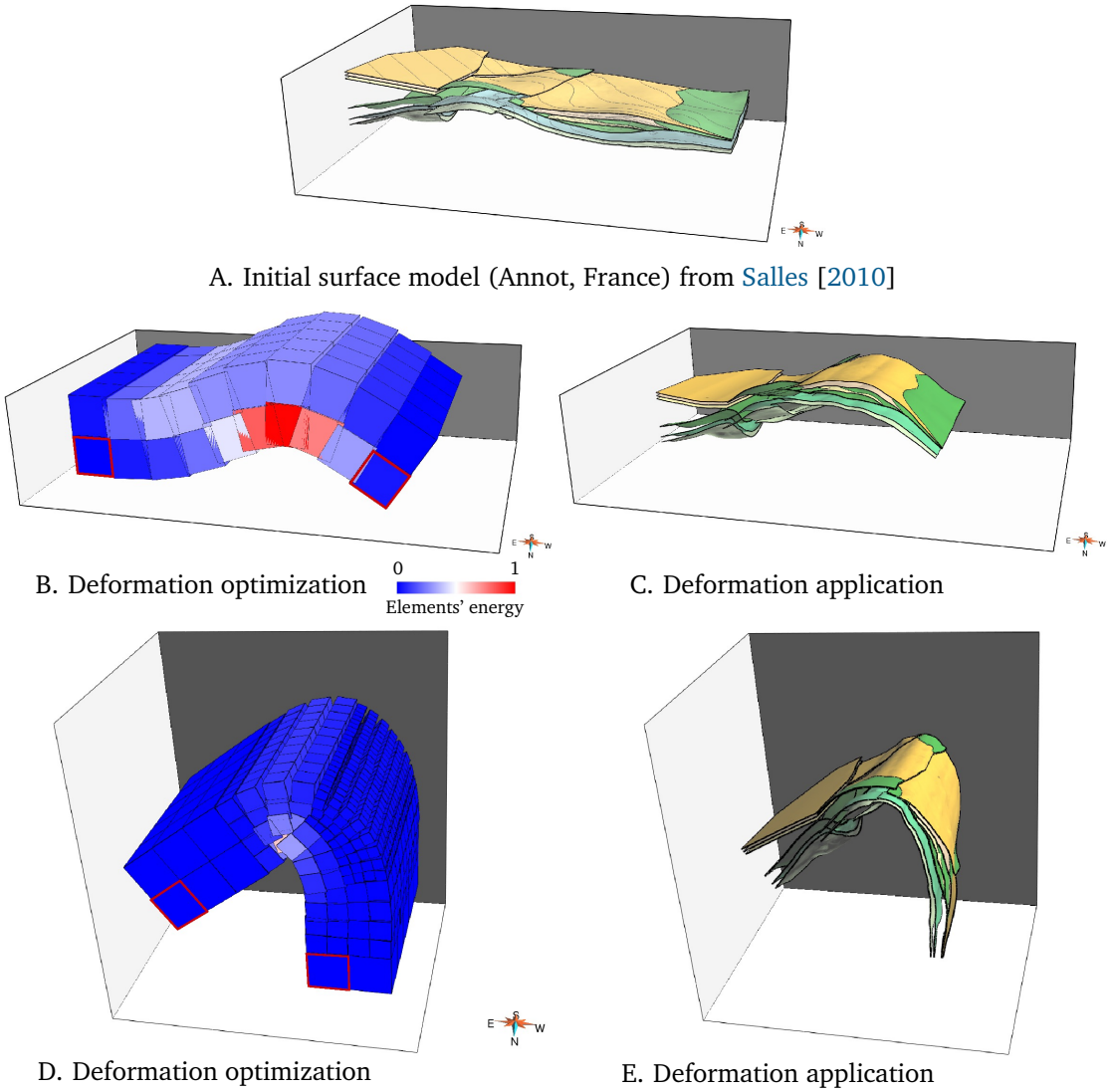


Figure 5 Folding geological structures. (A) The initial structural model (Annot Formation, SE France) [Salles, 2010]. Each surface represents the interface between successive layers. (B and D) The final position of the rigid elements (constrained elements highlighted by red borders). (D) Certain elements have been refined in the region of the hinge of the fold. (C and E) The deformation is applied to the structural model while maintaining the relationships between the different structural elements, even with extreme deformation (D and E).

lies on mechanics and thus theoretically converges towards an acceptable physical solution. It is interesting to observe that very similar results are obtained with *Reed* on a simple case (Fig. 7).

The deformed object is a sandstone beam with dimensions $30\times3\times3$m and a linear elastic model (with non-linear Green strain tensor). It is deformed with FEM and *Reed* having similar constraints, *i.e.* the left wall is fixed and the top of the right wall is constrained to go down 10m.

The finite element computations were performed with the Gocad plugin RestorationLab [Durand-Riard et al., 2010; Muron, 2005]. The beam is discretised by a tetrahedral mesh containing 1146 nodes. The FEM calculations are performed directly on this mesh whereas *Reed*'s computations are performed on a coarser mesh made of 288 cubic elements. Different numbers of elements have been tested to illustrate the evolution of the computation time when increasing the number of degrees of freedom (Table 1). This table indicates that *Reed* is more than 10 times faster on this example even for the same number of degrees of freedom (both implementations being scalar). Also, Fig. 7 shows that a few elements are sufficient to obtain a good approximation of physical deformations.

The displacements obtained by the two methods are very similar and they are both consistent with the theoretical mechanical behaviour of beams. For example, Fig. 7B shows that the lower and upper third of the beam accommodate the most work during the deformation. The object local dilatations are also similar, even if *Reed* tends to produce dilatations that are slightly higher than FEM in this example. This can be explained by the fact that the volume variation is not explicitly penalised in *Reed*. Also, in the present form, *Reed* cannot reproduce certain physical behaviour such as the Poisson effect (*e.g.* thinning of a beam being stretched). Finally, the differences between the displacements are relatively low, except in the left part where it is of the same order of magnitude as the displacement, because the norms of the displacement and the differences are both nearly 0.

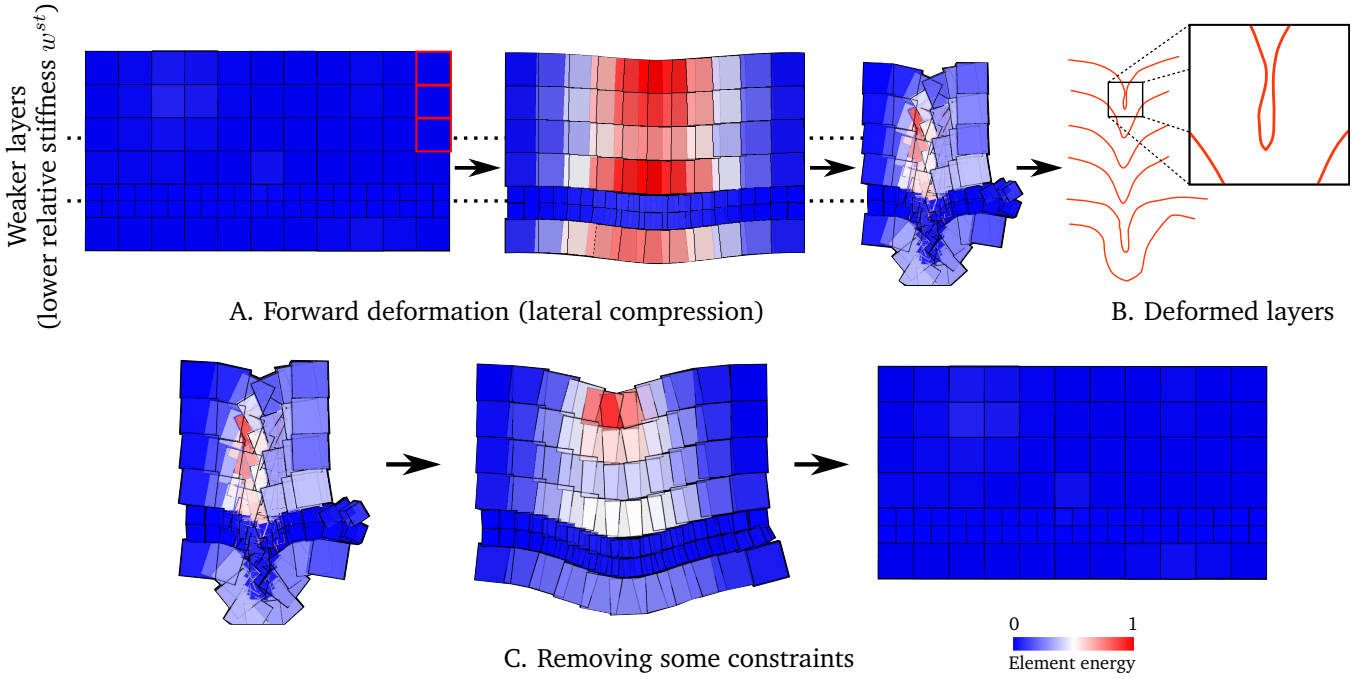


Figure 6 Robustness during extreme deformation. (A) A cross-section is laterally compressed by displacing the upper 3 right border elements (red) to the left until an extreme deformation is obtained. Some layers have a lower stiffness (stippled line) or are more refined which produces different strain localisation between the layers. (B) Six horizon traces, originally flat and centred on each layer, are deformed by the *Reed* object in (A). (C) When removing certain constraints, the system returns to its original configuration, which illustrates its robustness.

Method	Number of Elements	Number of degrees of freedom	time (sec)
FEM	1146	3438	5.3
Reed	11	66	0.03
Reed	288	1328	0.22
Reed	672	4032	0.33

Table 1 Comparison of the execution time of *Reed* and FEM. The time in seconds correspond to the time observed for *Reed* and FEM to converge for the bending of a beam (Fig. 7). The FEM computation uses the Gocad plugin RestorationLab Durand-Riard et al. [2010]; Muron [2005]. *Reed*'s result are obtained with our implementation in a Gocad plugin. All the times was obtained on a personal computer (2x2.66 GHz, 4 Go RAM).

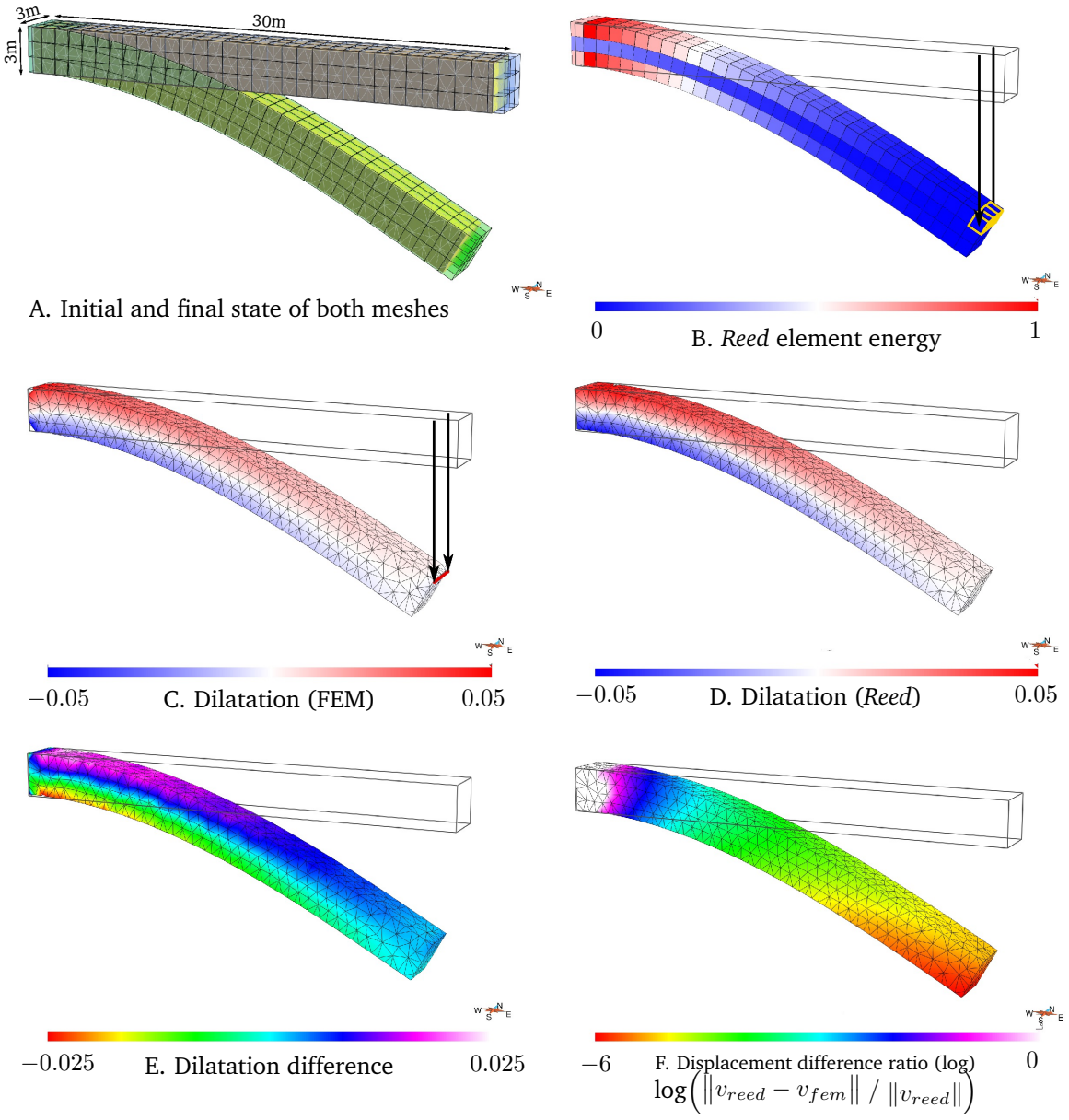


Figure 7 First order mechanical validity of *Reed*, comparison with FEM. (A) The displacements obtained by FEM (yellow solid with tetrahedral mesh) and those obtained with *Reed* (green transparent cubes). The embedding cubes are positioned over the left and right walls of the solid to make these walls correspond to the center of the elements. (B) The normalised energy of the rigid elements in their optimal position. (C and D) The dilatation of the object obtained by *Reed* and by FEM. (E) The difference between the two dilatations (positive values mean the dilatation is larger in *Reed* than in FEM and vice versa). (F) The logarithm of the ratio of the relative norm of the displacement difference highlights the relation between the difference of displacement and the displacement.

5 Building, editing and restoring geological models with Reed

This section illustrates three examples of situations where *Reed* can be successfully applied. We first present *Reed* as an editing tool for globally deforming an existing model. Then we demonstrate how it can be used to progressively build a complex synthetic model with several geological events. Finally, *Reed* is also adapted to the purpose of restoration.

5.1 Structural model editing

The editing of geological objects represents a first kind of application in which *Reed* can be used either for helping to generate an adapted initial geometry or to update geomodel geometry when new data becomes available or to reflect subsurface uncertainty. This can be helpful while constructing the model, for example to add small or large scale folds that are poorly constrained by data. A common approach to modelling structures is to optimise the geometry of triangulated surfaces with respect to data constraints [Caumon et al., 2009; Mallet, 2002]. To this end, having a good initial geometry is generally a key point, particularly when the structures are highly folded (*e.g.* Fig. 8 in Mallet [1997]). *Reed* can help in building these initial surfaces by rapidly folding simple geometries making the construction of structural surfaces easier.

Another application consists of locally perturbing the geometry of a structural model in order to explore the uncertainties about the structures, for instance to reflect velocity uncertainty in time-to-depth conversions (Fig. 8). In this context, keeping consistent geometrical relationships between the different structural objects can be tedious. Furthermore, the editing has to be relatively fast and involve few parameters to keep the dimensionality of the problem relatively low. This can be done by restricting the deformation parameters to the displacement of a few *Reed* elements. For example, in Fig. 8, the elements on borders are fixed while a vertical displacement is specified for a central element. Such deformation thus involves only one parameter which is the vertical offset. The deformation in this example takes 0.6s, allowing us to rapidly explore the space of uncertainties, for example by sampling the possible different models while varying the values of the structural parameters. The topological relationship between the different layers are maintained (faults and unconformities).

5.2 Forward modelling

Reed can also be used to build forward models (Fig. 9) similar to what was presented by Jessell and Valenta [1996]. Using a pseudo-mechanical deformation tool helps us to obtain more flexible deformations than the geometric operators of Jessell and Valenta [1996]. Indeed, with a kinematic approach, the displacement field has to be specified for the whole object to be deformed whereas a dynamic approach allows us to specify only the displacement of some key elements, the complete displacement field being then a result of the deformation model.

We generated a model with two successive sequences: two sedimentation events leading to horizontal layers, each followed by a folding event. The resulting rather complex model, with its associated deformation history, was built within a few minutes.

Our approach is promising for modelling basins and deformed belts with complex polyphase tectonics, by separating the different episodes. To this aim, an inverse modelling approach could be used with such a tool to explore the space of possible deformation scenarios and to help modelling geological structures with respect to these possible scenarios as in Cardozo [2005]; Cardozo and Aanonsen [2009].

5.3 Restoration of large deformations

Compared to most restoration methods, *Reed* focuses on robustness and interactivity rather than on simulation accuracy. It allows us to run approximate restorations very quickly, without necessarily having to generate a volumetric 3D mesh. This is useful for instance when only a few structural surfaces have been built and one wants to rapidly generate a petrophysical model in restored space. It can also help to characterise possible tectonic scenarios in the early stages of the modelling process.

This requires us to adapt *Reed*'s boundary conditions to the purposes of restoration (Fig. 10), the classical boundary condition in restoration being the flattening of the uppermost stratigraphic horizon of the structural model. With *Reed*, deformed objects, including stratigraphic horizons, are not directly involved in the deformation computation. As a consequence, this condition has to be transferred to the rigid elements embedding the stratigraphic horizon. This is similar to the restoration of implicit surfaces [Durand-Riard et al., 2010].

The successive tasks involved in the implementation of this constraint in *Reed* are as follows:

1. The elements which intersect the stratigraphic horizon are detected.
2. For each intersected element, the intersection is approximated by the mid-plane of the points contained by the element (Fig. 10B).
3. The z value of the center of each intersected element is computed by adding the signed-distance to the mid-plane to derived elevation of the target horizon.
4. The rotation of each intersected element is computed such that the direction going from the center of the element to its projection on the mid-plane is vertical.

The boundary conditions computed in this way are progressively applied to the model while optimizing the position of the free elements until the surface is restored.

Our experience shows that the conditions in z and in rotation may conflict when the curvature of the surface is too large relative to the size of the elements. Indeed, some approximations are introduced in the computation of the boundary conditions because of the use of local mid-planes instead of the complete surface. The mid-planes represent a simplification of the surface as they cannot take into account its local curvature. This effect can be limited by locally adapting the size of the elements to the geometry of the stratigraphic horizon. In the future these conditions could also be relaxed by introducing them as soft constraints into the model rather than locking the corresponding degrees of freedom.

The restoration with *Reed* has been tested on the Han-sur-Lesse anticline, Belgium [Bonniver, 2011; Henrion et al., 2009]. The faults contained in the model are not taken into account because they represent small displacements and occurred after the anticline formation. Only the lower horizons have been restored because the other ones have

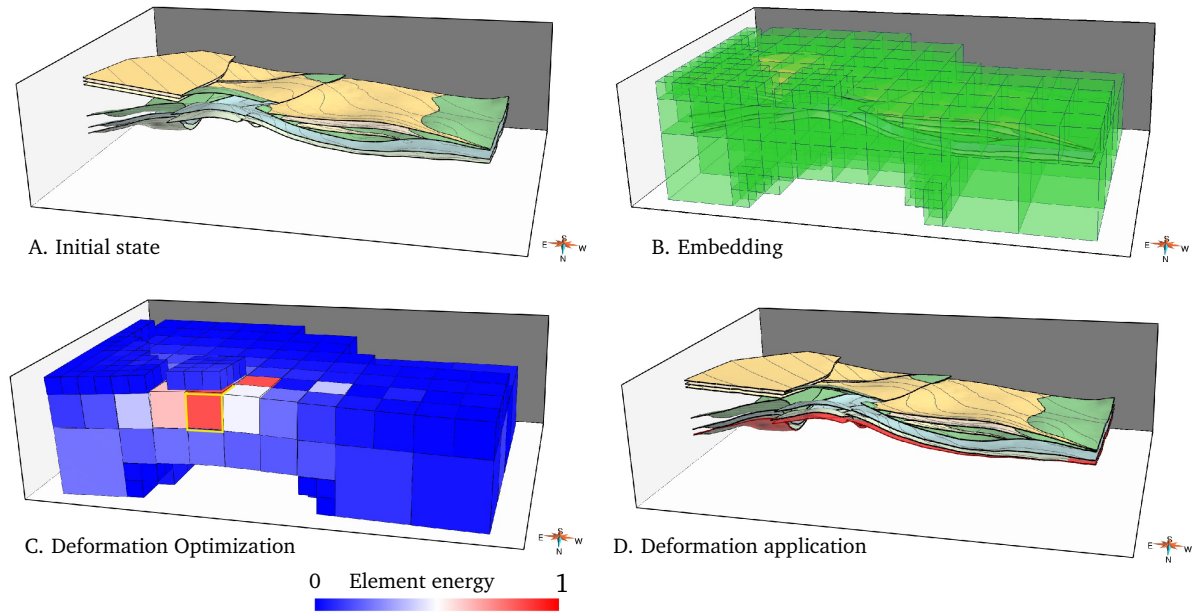


Figure 8 Structural model editing. (A) A model of the Annot Formation (SE France), from Salles [2010] consisting of a series of structural horizons with faults and an unconformity. (B) The structural objects are embedded in rigid elements. The elements are refined to fit the structures. (C) The deformation is computed with boundary conditions fixing the position of the bottom elements on the left and right borders and fixing a given vertical displacement for a central element (yellow borders). (D) Result of the deformation applied onto the initial structural model. The red surface represents the initial position of the base horizon.

very closed limbs, which complicates the boundary condition computation as several surface parts located on opposite limbs may intersect the same rigid elements. Refining the elements would be a solution to avoid this problem.

Fig. 11 shows the horizons in their deformed position and their restoration with three different *Reed* objects with an increasing number of elements. The horizon flattening error is color-coded on the restored surface and the histogram of this error is also presented. For each *Reed* object, 7 iterations have been performed to ensure the convergence of the restoration. Calculation times are indicated given the number of iterations and the time per iteration. The total times are 1.52s for 360 elements, 3.08s for 594 elements and 29.4s for 2700 elements. The decreasing restoration error as the number of elements increases illustrates the impact of a better intersection computation when setting the restoration boundary conditions. However, the computation time grows rapidly, which limits the benefits of refining the elements. Refining the elements only locally, around the restored surface, would be a solution to avoid this problem while minimizing the restoration error.

6 Conclusion and perspectives

In this paper we define original specifications for a light geological deformation tool, which represent the main contribution of this paper. This kind of deformation algorithm should help geologists to better integrate geological deformation histories at the early stages of the modelling process. Namely, they could help the geomodellers to interactively deform, forward model and restore the geological structures, as illustrated in Section 5. These specifications are oriented towards interactivity and robustness during interpretive tasks. They place the physical accuracy on a second level, which introduces some interesting perspectives for the range of deformation tools for geomodelling. We propose *Reed* as a potential answer to these specifications, but other algorithms may also be investigated.

The fact that *Reed* relies on rigid elements brings an efficiency to this algorithm and a robustness to large deformations. The deformed objects are embedded into the rigid elements, which means that the deformation is computed at the scale of the rigid elements and transferred afterwards to the embedded objects. This allows us to decouple large-scale deformations and small-scale geometric details. Adaptive refinement is another strength of this algorithm which allows us to adapt the size of the elements to the details of the embedded objects or to the degree of deformation. These capabilities make this method applicable even in the case of extreme flexural deformation (Fig. 5 and Fig. 6).

The examples provided in this paper indicate that *Reed* matches the desired criteria in terms of interactivity, robustness and, to some degree, physical accuracy (Section 4.2). In the case of 3D restoration, our experiments suggest that this method is a good complement to existing kinematic methods and elastic geomechanical methods. Indeed, it introduces a new volumetric way to unfold layers affected by possibly large deformations.

However, some limitations remain in the use of *Reed* for geological deformations. The mechanical behaviour of the rigid elements, induced by the coupling energy minimisation, promotes a pure bending deformation. In natural stratigraphic layers, the orientation of the layering results in an anisotropic mechanical response [Durand-Riard et al., 2013]. Another prospect for future development of *Reed* is therefore to adapt the energy formulation to anisotropic behaviour.

A third limitation comes from the fixed connectivity of the elements. In large displacement contexts, faults may

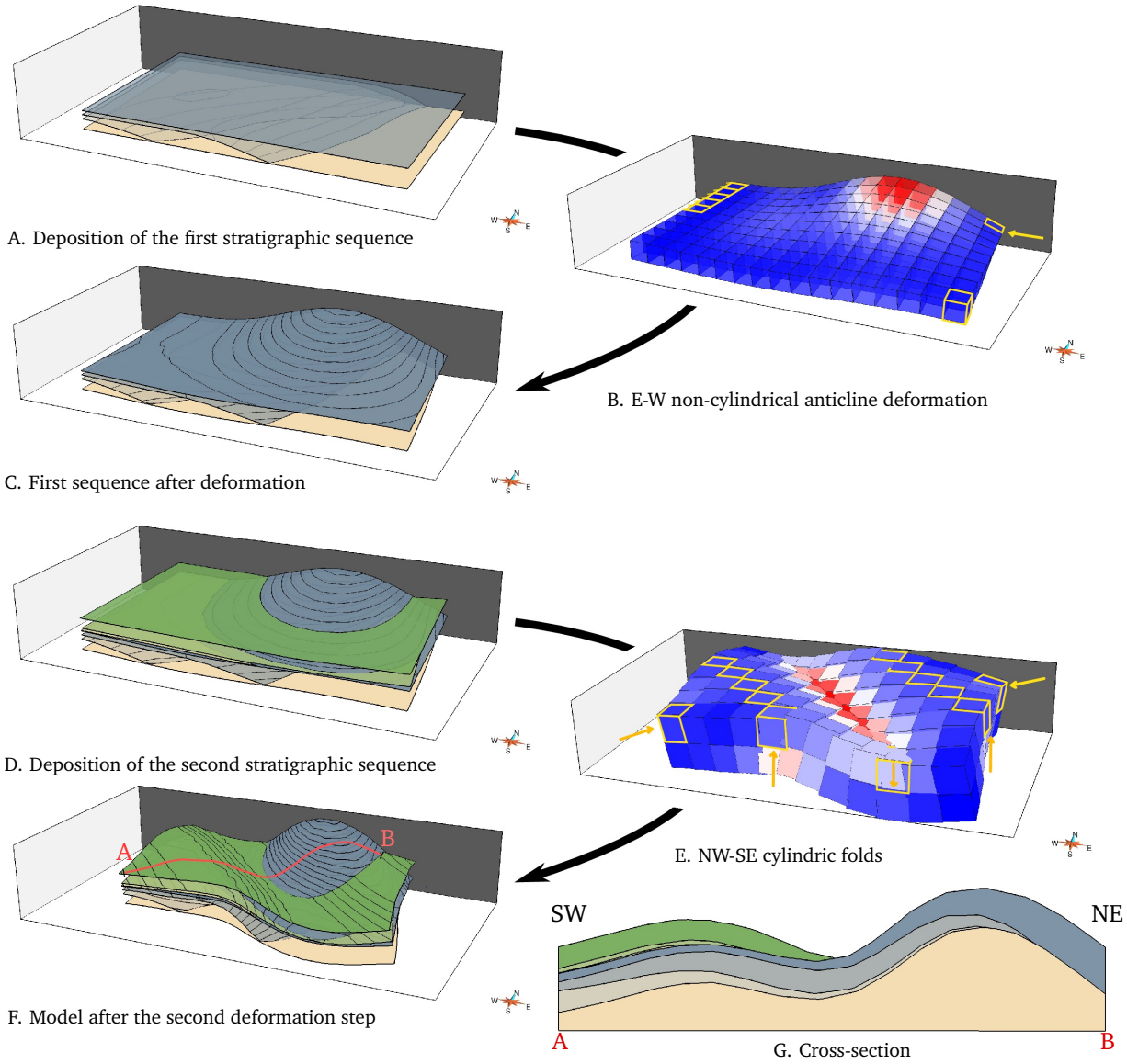


Figure 9 Forward modelling with *Reed*. The deformation algorithm is used to represent the tectonics involved in a scenario with several successive phases. (A) Modelling of the deposition of a first stratigraphic sequence in onlap configuration. (B) A first deformation event is constrained by the boundary conditions (yellow) which fix the NW and SE corners while displacing the NE corner to the W. (C) The deformation is interpolated onto the first stratigraphic sequence and results in a non-cylindrical anticline localised in the north part of the model. (D) A second stratigraphic sequence is deposited. (E) A second deformation event is computed with this time the NE and SW corners that are displaced toward each other and NW-SE rows of elements that are constrained vertically to force the appearance of two parallel cylindrical anticlines. (F) This deformation is applied to both sequences which superimposes the two deformation events. (G) A SW-NE cross-section shows the final geometry. The cross-section location is shown on (F).

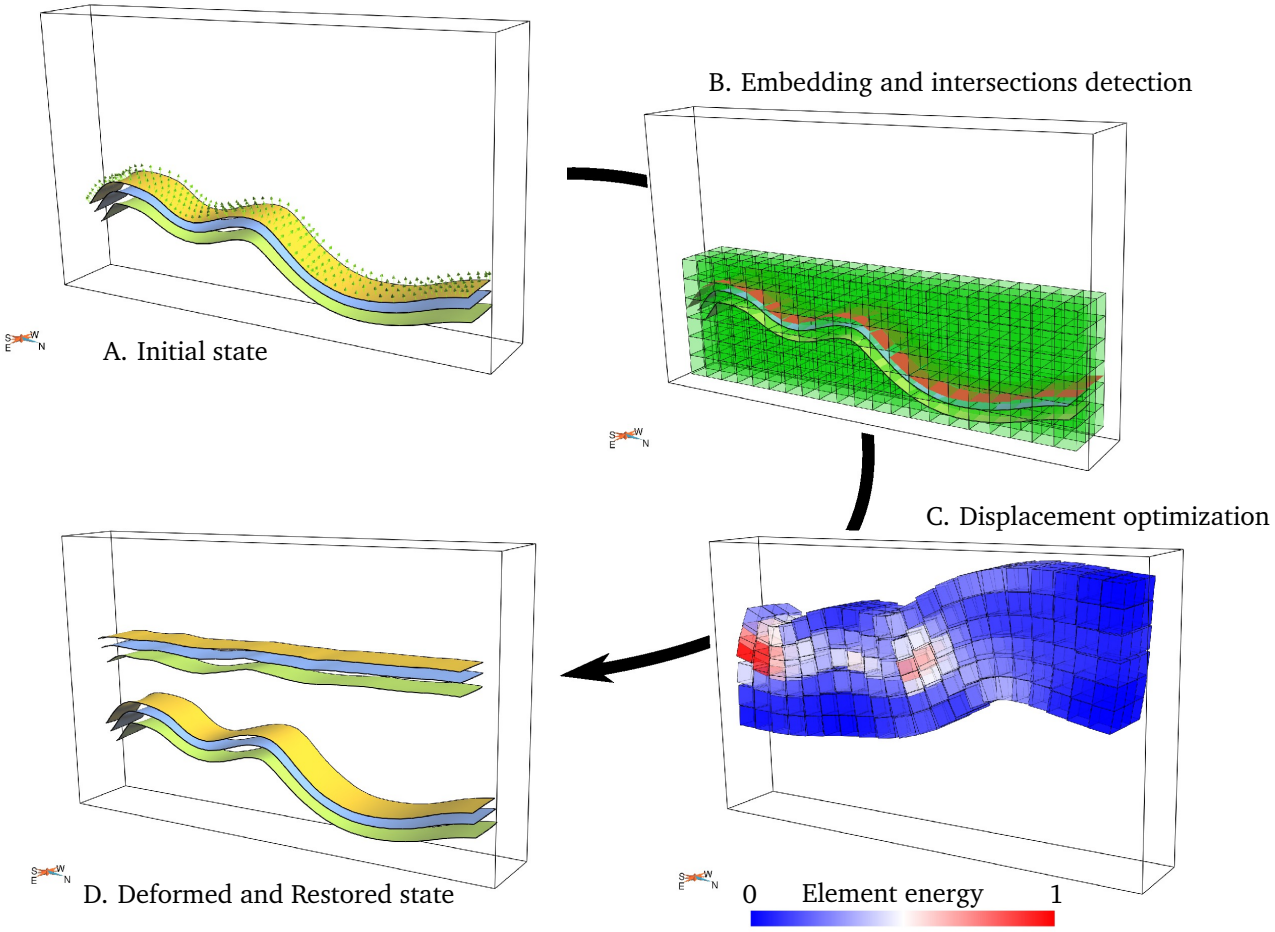


Figure 10 Restoration of synthetic horizons with *Reed*. (A) The initial stratigraphic horizons. The normal vectors are also represented (green arrows). (B) The intersections between the rigid elements (green) and the horizon to be restored are detected, and each part of the surface is represented by its mean plane inside each element (red). (C) The boundary conditions are applied and the position of the cubes optimised with respect to the global energy of the model. Additional boundary conditions are implemented to ensure a unique solution by fixing the position of the elements intersected by the left extremity of the horizon. (D) The deformation is finally applied to the stratigraphic horizons.

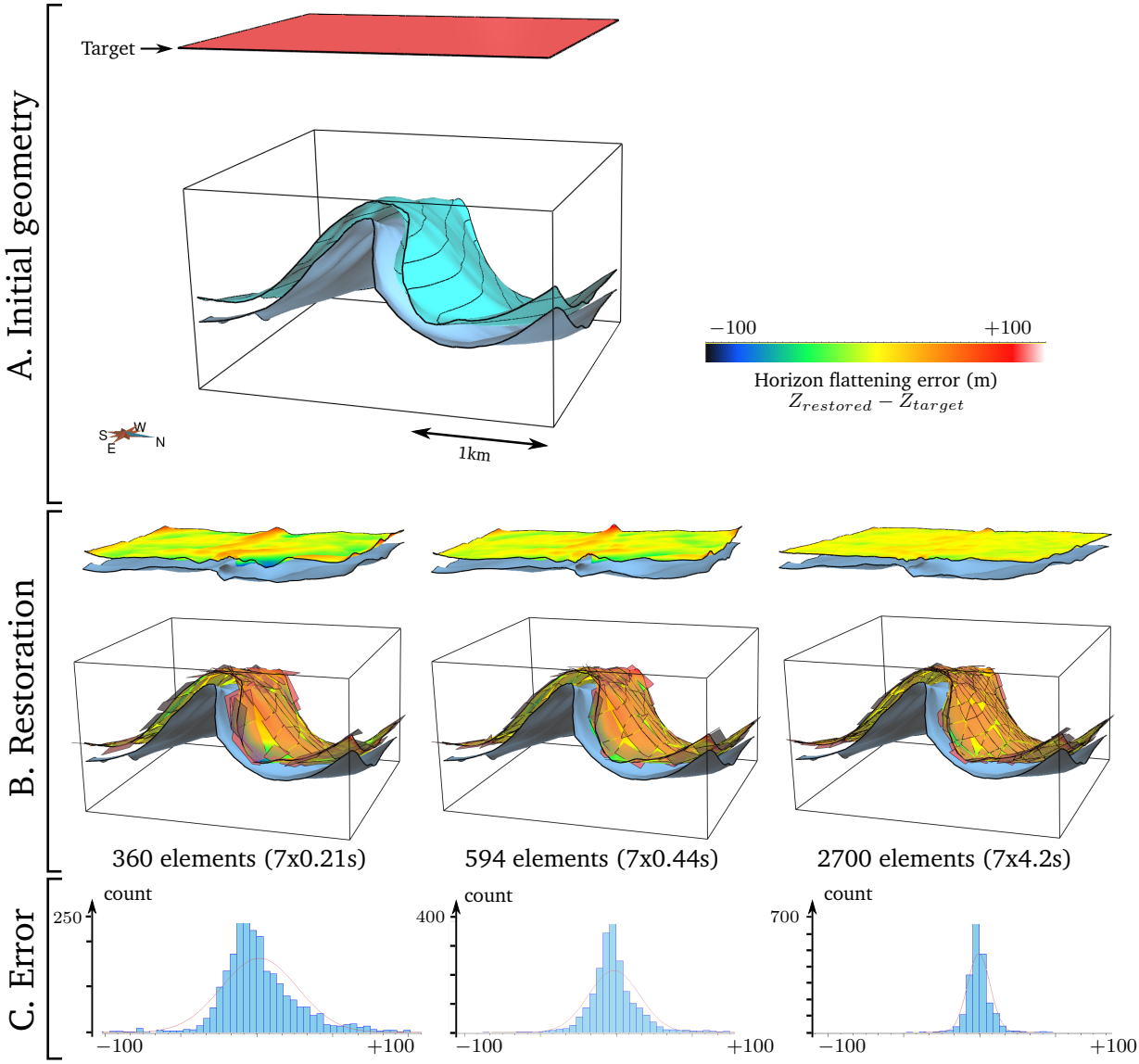


Figure 11 Restoration of the Han-sur-Lesse anticline (Belgium) with *Reed*. Two horizons of the Han-sur-lesse anticline [Bonniver, 2011; Henrion et al., 2009] are restored. (A) The initial geometry (blue) and the target position of the top horizon (red). The horizontal and vertical scales are identical. (B) Restoration of the top horizon with *Reed*. The three different images correspond to an increasing number of elements, which allows us to improve the restoration results. This is due to the increasing quality of the surface approximation in the rigid elements (red polygons). The error corresponding to the difference between the target depth and the actually obtained depth is painted on the top surface in its restored and initial state. (C) The distribution of the restoration error (each histogram is rendered with the same limits to ease comparison).

be required to let the relationship between elements evolve during the deformation. Taking into account faults in the deformation model is also important for future work. This last limitation is certainly more challenging as it would require us to adapt the geometry of the elements cut by a fault, to introduce the fault as an internal object in the rigid element description and to define a cost function that depends on the displacement of elements that are not initially neighbours.

A last limitation is that, for the sake of performance, the algorithm does not check for self-collision. However, in the context of very large deformations, for example with recumbent folds, one fold limb can come in contact with another. This kind of situation could be detected in order to avoid the two limbs intersecting.

Despite these limitations, the examples presented in this paper illustrate the capabilities of *Reed* and its utility for large deformation and 3D geomodel editing. The application to restoration in a large deformation context is promising and needs to be further examined. Other kinds of applications can also be explored, for example, building a structural model together with its associated deformation could help the determination of fracture location or stress evaluation for well planning in an industrial context.

Reed may not be the only method to fulfil the specifications we defined. However, it highlights the interest of using methods which balance space deformation and physically realistic behaviour. As suggested by Natali et al. [2013], *Reed* has provided a working example highlighting the application of Computer Graphics methods for interactively modelling geological structures.

Acknowledgements

The authors are grateful to the academic and industrial sponsors of the Gocad Consortium managed by ASGA (Association Scientifique pour la Géologie et ses Applications, www.gocad.org) for funding this work, and to Paradigm Geophysical for providing the Gocad software and API. This work was performed in the framework of the « Investissements d'avenir » Labex RESSOURCES21 (ANR-10-LABX-21).

References

V. Allken, R. S. Huismans, and C. Thieulot. Three-dimensional numerical modeling of upper crustal extensional systems. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 116(B10), 2011. doi: doi:10.1029/2011JB008319. *(Cited pages 3 and 4)*

A. H. Barr. Global and local deformations of solid primitives. In *ACM Siggraph Computer Graphics*, volume 18, pages 21–30, 1984. *(Cited page 5)*

I. Bonniver. *Étude hydogéologique et dimensionnement par modélisation du « Système-Traçage » du réseau karstique de Han-sur-Lesse (Massif de Boine - Belgique)*. PhD thesis, Faculté des Sciences, FUNDP, Namur, Belgique, 2011. *(Cited pages 12 and 16)*

P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics (TOG)*, 13(2):137–155, 1994. *(Cited page 5)*

M. Botsch, M. Pauly, M. Wicke, and M. Gross. Adaptive space deformations based on rigid cells. *Computer Graphics Forum*, 26(3):339–347, 2007. ISSN 1467-8659. *(Cited pages 3, 5, 7, 8, and 18)*

D. Burbidge and J. Braun. Numerical models of the evolution of accretionary wedges and fold-and-thrust belts using the distinct-element method. *Geophysical Journal International*, 148(3):542–561, 2002. *(Cited pages 4 and 5)*

N. Cardozo. Trishear modeling of fold bedding data along a topographic profile. *Journal of Structural Geology*, 27(3):495–502, 2005. ISSN 0191-8141. doi: 10.1016/j.jsg.2004.10.004. URL http://www.sciencedirect.com/science/article/B6V9D-4FC3RR9-1/2/72fe26e01f3a1ca8d6ffe3562085245f. *(Cited page 12)*

N. Cardozo and S. Aanonsen. Optimized trishear inverse modeling. *Journal of Structural Geology*, 31(6):546–560, 2009. ISSN 0191-8141. doi: 10.1016/j.jsg.2009.03.003. URL http://www.sciencedirect.com/science/article/B6V9D-4VTVPRM-1/2/373cd7dcbd621004211c4603dcc42359. *(Cited page 12)*

G. Caumon, F. Lepage, C. H. Sword, and J.-L. Mallet. Building and editing a Sealed Geological Model. *Mathematical Geology*, 36(4):405–424, 2004. doi: 10.1023/B:MATG.0000029297.18098.8a. *(Cited page 3)*

G. Caumon, P. Collon-Drouaillet, C. Le Carlier de Veslud, S. Viseur, and J. Sausse. Surface-based 3D modeling of geological structures. *Mathematical Geology*, 41(8):927–945, 2009. *(Cited pages 2 and 12)*

G. Caumon, G. Laurent, J. Pellerin, N. Cherpeau, F. Lallier, R. Merland, and F. Bonneau. Current bottlenecks in geomodeling workflows and ways forward. In D. Garner, D. Thenin, and C. V. Deutsch, editors, *Closing the Gap: Advances in Applied Geomodeling for Hydrocarbon Reservoirs*, volume 20 of *CSPG Memoir*, chapter 4, pages 43–52. Canadian Society of Petroleum Geologists, 2013. *(Cited page 3)*

R. T. Chamberlin. The Appalachian folds of Central Pennsylvania. *The Journal of Geology*, 18:228–251, 1910. *(Cited page 3)*

N. Cherpeau, G. Caumon, J. K. Caers, and B. Lévy. Method for stochastic inverse modeling of fault geometry and connectivity using flow data. *Mathematical Geology*, 44(2):147–168, 2012. ISSN 1874-8961. doi: 10.1007/s11004-012-9389-2. *(Cited page 3)*

C. Dahlstrom. Balanced cross sections. *Canadian Journal of Earth Sciences*, 6:743–757, 1969. *(Cited page 3)*

P. Durand-Riard, G. Caumon, and P. Muron. Balanced restoration of geological volumes with relaxed meshing constraints. *Computers & Geosciences*, 36(4):441–452, 2010. doi: 10.1016/j.cageo.2009.07.007. *(Cited pages 2, 4, 9, 10, and 12)*

P. Durand-Riard, C. A. Guzofski, G. Caumon, and M.-O. Titeux. Handling natural complexity in 3D geomechanical restoration, with application to the recent evolution of the outer fold-and-thrust belt, deepwater Niger Delta. *AAPG Bulletin*, 97(1):87–102, 2013. doi: 10.1306/06121211136. *(Cited pages 2 and 13)*

S. S. Egan, T. S. Buddin, S. J. Kane, and G. D. Williams. Three-dimensional modelling and visualisation in structural geology: New techniques for the restoration and balancing of volumes. In *Proceedings of the 1996 Geoscience Information Group Conference on Geological Visualisation, Electronic Geology*, volume 1, pages 67–82, 1997. *(Cited page 3)*

J. Gain and D. Bechmann. A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics (TOG)*, 27(4): Article 107, 2008. doi: 10.1145/1409625.1409629. *(Cited page 5)*

F. Georgsen, P. Røe, A. R. Syversveen, and O. Lia. Fault displacement modelling using 3D vector fields. *Computers & Geosciences*, 16:247–259, 2012. *(Cited page 3)*

J.-P. Gratier, B. Guillier, A. Delorme, and F. Odonne. Restoration and balance of a folded and faulted surface by best-fitting of finite elements: principle and applications. *Journal of Structural Geology*, 13(1):111–115, 1991. ISSN 0191-8141. doi: 10.1016/0191-8141(91)90107-T. *(Cited pages 3 and 7)*

P. Griffiths, S. Jones, N. Salter, F. Schaefer, R. Osfield, and H. Reiser. A new technique for 3-D flexural-slip restoration. *Journal of Structural Geology*, 24(4):773–782, 2002. ISSN 0191-8141. doi: 10.1016/S0191-8141(01)00124-9. *(Cited page 3)*

O. Grosse. *Mise en cohérence automatique d'un modèle géologique 3D*. PhD thesis, INPL, 2002. *(Cited page 3)*

V. Henrion, L. Abasq, I. Bonniver, and G. Caumon. Integrated characterization and modeling of cave network: application to the karstic aquifer of Han-sur-Lesse (Belgium). In *Proc. of the 29^{th} Gocad Meeting, Nancy, France*, 2009. *(Cited pages 12 and 16)*

G. Hirota, R. Maheshwari, and M. C. Lin. Fast volume-preserving free-form deformation using multi-level optimization. *Computer-Aided Design*, 32(8):499–512, 2000. *(Cited page 5)*

M. Jessell and R. Valenta. Structural geophysics: integrated structural and geophysical modelling. *Computer Methods in the Geosciences*, 15: 303–324, 1996. *(Cited pages 3 and 12)*

G. Laurent, G. Caumon, A. Bouziat, and M. Jessell. A parametric method to model 3D displacements around faults with volumetric vector fields. *Tectonophysics*, 590:83–93, 2013. ISSN 0040-1951. doi: 10.1016/j.tecto.2013.01.015. URL http://www.sciencedirect.com/science/article/pii/S0040195113000498. *(Cited page 3)*

C. Loiselet, J. Braun, L. Husson, C. Le Carlier de Veslud, C. Thieulot, P. Yamato, and D. Grujic. Subducting slabs: Jellyfishes in the Earth's mantle. *Geochemistry, Geophysics, Geosystems*, 11(8), 2010. doi: doi:10.1029/2010GC003172. *(Cited pages 3 and 4)*

L. Maerten and F. Maerten. Chronologic modeling of faulted and fractured reservoirs using geomechanically based restoration: Technique and industry applications. *AAPG Bulletin*, 90(8):1201–1226, 2006. *(Cited pages 3 and 4)*

J. Mallet. Discrete modeling for natural objects. *Mathematical Geology*, 29 (2):199–219, 1997. *(Cited page 12)*

J.-L. Mallet. *Geomodeling*. Oxford University Press, New York, NY, USA, 2002. *(Cited pages 2, 3, and 12)*

J. Massot. *Implémentation de méthodes de restauration équilibrée 3D*. PhD thesis, Centre de Recherche Pétrographique et Géochimique, 2002. *(Cited page 3)*

I. Moretti. Working in complex areas: New restoration workflow based on quality control, 2D and 3D restorations. *Marine and Petroleum Geology*, 25:205, 2008. *(Cited page 2)*

I. Moretti, F. Lepage, and M. Guiton. Kine3D: A new 3D restoration method based on a mixed approach linking geometry and geomechanics. *Oil & Gas Science and Technology*, 61(2):277–289, 2006. *(Cited pages 3 and 4)*

P. Muron. *Méthodes numériques 3D de restauration des structures géologiques faillées*. PhD thesis, Institut National Polytechnique de Lorraine, 2005. *(Cited pages 3, 4, 9, and 10)*

M. Natali, E. M. Lidal, I. Viola, and D. Patel. Modeling terrains and subsurface geology. *Proc. of EuroGraphics 2013 State of the Art Reports (STARs)*, pages 155–173, 2013. *(Cited page 17)*

A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006. *(Cited page 4)*

D. Rouby, H. Xiao, and J. Suppe. 3-D restoration of complexly folded and faulted surfaces using multiple unfolding mechanism. *AAPG Bulletin*, 84(6):805–829, 2000. *(Cited page 3)*

L. Salles. *Contrôles structuraux en 3 dimensions de la sédimentation turbiditique dans les chaînes plissées: exemple des Grès d'Annot (Sud-Est de la France)*. PhD thesis, Institut National Polytechnique de Lorraine, 2010. *(Cited pages 8, 9, and 13)*

D. Schmid, M. Dabrowski, and M. Krotkiewski. Evolution of large amplitude 3D fold patterns: A FEM study. *Physics of the Earth and Planetary Interiors*, 171(1-4):400–408, 2008. *(Cited page 4)*

L. Scholtès and F.-V. Donzé. A DEM model for soft and hard rocks: Role of grain interlocking on strength. *Journal of the Mechanics and Physics of Solids*, 61(2):352–369, 2013. ISSN 0022-5096. doi: 10.1016/j.jmps.2012.10.005. URL http://www.sciencedirect.com/science/article/pii/S0022509612002268. *(Cited pages 4 and 5)*

T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *ACM Siggraph Computer Graphics*, 20(4):151–160, 1986. *(Cited page 5)*

A. Seiler, S. Aanonsen, G. Evensen, and J. Rivenaes. Structural surface uncertainty modeling and updating using the ensemble Kalman filter. *Society of Petroleum Engineers Journal*, 15(4):1062–1076, 2010. *(Cited page 3)*

M. Siavelis. *Modélisation numérique X-FEM de grands glissements avec frottement le long d'un réseau de discontinuités.* PhD thesis, École Central de Nantes, 2011. *(Cited page 4)*

A.-L. Tertois and J.-L. Mallet. Editing faults within tetrahedral volume models in real time. In S. J. Jolley, D. Barr, J. J. Walsh, and R. J. Knipe, editors, *Structurally Complex Reservoirs*, volume 292 of *Geol. Society Spec. Publ.*, pages 89–101, 2007. doi: 10.1144/sp292.5. *(Cited page 3)*

M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International 2004 Proceedings*, pages 312–319. IEEE, 2004. *(Cited page 5)*

O. Vidal-Royo, S. Hardy, and J. A. Muñoz. The roles of complex mechanical stratigraphy and syn-kinematic sedimentation in fold development: insights from discrete-element modelling and application to the Pico del Águila anticline (External Sierras, Southern Pyrenees). *Geological Society, London, Special Publications*, 349(1):45–60, 2011. *(Cited pages 4 and 5)*

O. Vidal-Royo, N. Cardozo, J. A. Muñoz, S. Hardy, and L. Maerten. Multiple mechanisms driving detachment folding as deduced from 3D reconstruction and geomechanical restoration: the Pico del Águila anticline (External Sierras, Southern Pyrénées). *Basin Research*, 24(3):295–313, 2012. ISSN 0950-091X. doi: 10.1111/j.1365-2117.2011.00525.x. *(Cited page 4)*

W. Von Funck, H. Theisel, and H. Seidel. Vector field based shape deformations. *ACM Transactions on Graphics (TOG)*, 25(3):1118–1125, 2006. *(Cited page 5)*

G. Williams, S. Kane, T. Buddin, and A. Richards. Restoration and balance of complex folded and faulted rock volumes: flexural flattening, jigsaw fitting and decompaction in three dimensions. *Tectonophysics*, 273(3–4): 203–218, 1997. ISSN 0040-1951. doi: 10.1016/S0040-1951(96)00282-X. *(Cited page 3)*

A Displacement interpolation process

In Botsch et al. [2007], the embedded objects are deformed by interpolating the translation of the center of the elements, with inverse distance weighting or with a sum of Radial Basis Functions (RBF). Here, we propose to use the whole rigid motion of each neighbour element by taking the rotation of the elements into account.

Fig. 12 illustrates this process on a 2D example with a bilinear interpolation:

1. The point to be displaced x is located with respect to the element where it is embedded and its neighbours. This is described by the local coordinates u and v, as shown in Fig. 12A.
2. The contribution of each neighbour is computed by applying their final rigid motion M_i to the point x.
3. The interpolated final position $\overline{M(x)}$ is computed as a weighted sum of the neighbours contributions:

$$\begin{aligned}\overline{M(x)} = (1-u)(1-v)\; M_0(x) + \\ u\;(1-v)\; M_1(x) + (1-u)\; v\; M_2(x) + u\; v\; M_3(x)\end{aligned}$$

For the final rendering of the deformed object, higher degree interpolation (*e.g.* cubic interpolation or tri-harmonic RBFs, as proposed by Botsch et al. [2007]) can be used to ensure smoother results.

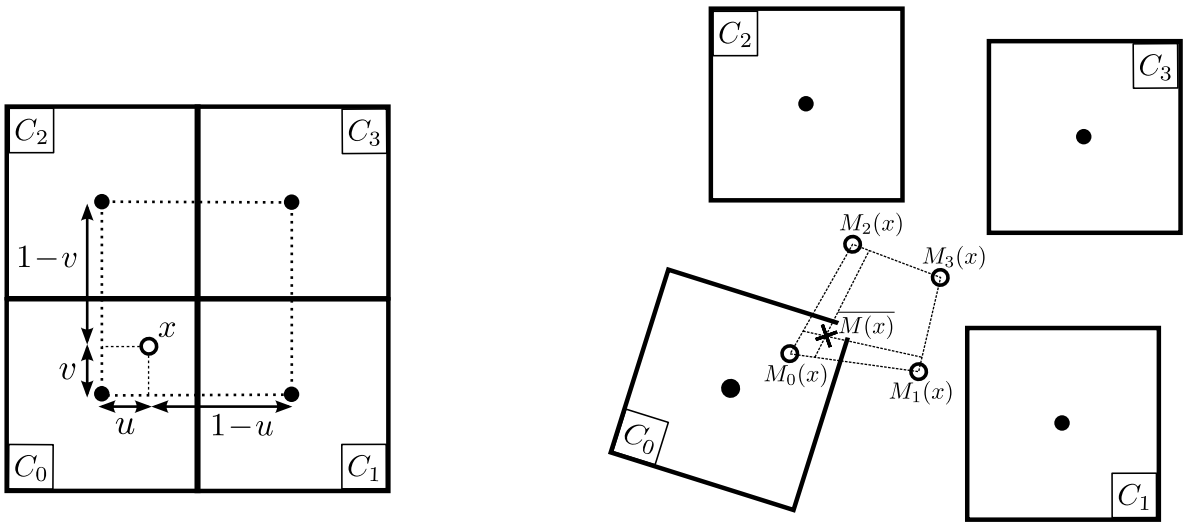
B Linear system computation details

This appendix details the computation of the polynomial coefficients Q, R and C, in the expression of the linearised energy E^* (Eq. 6). These coefficients are respectively a matrix, a vector and a constant. They are computed by applying the linearised increment of displacement (Eq. 4) to the coupling energy equation (Eq. 2). This equation is derived in its matricial form by applying the following operations.

First, the linearised increment of displacement is expressed with respect to the unknown, namely the linear velocity v and the angular velocity ω:

$$\begin{aligned}A_i\big(M_i(x)\big) &= M_i(x) + v_i + \omega_i \times \big(M_i(x) - M_i(c_i)\big) \\ &= M_i(x) + v_i + \\ &\quad \omega_i \times \big(\mathsf{R}_i(x-c_i) + \mathsf{T}_i + c_i - \mathsf{R}_i(c_i - c_i) - \mathsf{T}_i - c_i\big) \\ &= M_i(x) + v_i + \omega_i \times \big(\mathsf{R}_i(x-c_i)\big) \\ &= M_i(x) + P_i(x)\, U_i\end{aligned}$$

$$\text{with}\quad P_i(x) = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & \rho_i(x)_2 & -\rho_i(x)_1 \\ 0 & 1 & 0 & -\rho_i(x)_2 & 0 & \rho_i(x)_0 \\ 0 & 0 & 1 & \rho_i(x)_1 & -\rho_i(x)_0 & 0 \end{array}\right]$$



A. Position of the point within the elements in initial state

B. Interpolated position after displacement

Figure 12 Displacement interpolation within rigid elements. (A) The point x to be displaced is located within the elements C_0 to C_3 in their original position with local coordinates u and v. (B) $M_0(x)$ to $M_3(x)$ represent the deformed positions according to each surrounding element. The final position $\overline{M(x)}$ is obtained by a bilinear interpolation of these deformed positions.

$$\text{and} \quad \rho_i(x) = \mathsf{R}_i(x - c_i) = \boxed{\begin{matrix} \rho_i(x)_0 \\ \rho_i(x)_1 \\ \rho_i(x)_2 \end{matrix}}$$

where $P_i(x)$ is a 3×6 matrix and $\rho_i(x)$ a 3×1 vector.

The difference between the linearised displacement of two neighbour elements is expressed as:

$$\begin{aligned} & A_i\big(M_i(x)\big) - A_j\big(M_j(x)\big) \\ &= M_i(x) - M_j(x) + P_i(x)\,U_i - P_j(x)\,U_j \\ &= M_{ij}(x) + P_{ij}(x)\,U_{ij} \end{aligned}$$

$$\text{with} \quad M_{ij}(x) = \boxed{M_i(x) - M_j(x)},$$

$$P_{ij}(x) = \left[\begin{array}{c|c} P_i(x) & -P_j(x) \end{array}\right] \quad \text{and} \quad U_{ij} = \boxed{\begin{matrix} U_i \\ U_j \end{matrix}}$$

where $M_{ij}(x)$ is a 1×3 vector, $P_{ij}(x)$ a 3×12 matrix and U_{ij} a 12×1 matrix.

The next step is to compute the squared norm of this term. The notations of the terms depending on x are simplified in the following equations to make it easier to read:

$$\begin{aligned} & \Big\| A_i\big(M_i(x)\big) - A_j\big(M_j(x)\big) \Big\|^2 \\ &= \big(M_{ij} + P_{ij}U_{ij}\big)^T \big(M_{ij} + P_{ij}U_{ij}\big) \\ &= M_{ij}^T M_{ij} + M_{ij}^T P_{ij} U_{ij} + \big(P_{ij}U_{ij}\big)^T M_{ij} + \big(P_{ij}U_{ij}\big)^T P_{ij} U_{ij} \\ &= M_{ij}^T M_{ij} + M_{ij}^T P_{ij} U_{ij} + U_{ij}^T P_{ij}^T M_{ij} + U_{ij}^T P_{ij}^T P_{ij} U_{ij} \end{aligned}$$

In this last equation, the term $U_{ij}^T P_{ij}^T M_{ij}$ is a scalar. As a consequence, it can be replaced by its transposed form, which yields:

$$\begin{aligned} & \Big\| A_i\big(M_i(x)\big) - A_j\big(M_j(x)\big) \Big\|^2 \\ &= M_{ij}^T M_{ij} + 2M_{ij}^T P_{ij} U_{ij} + U_{ij}^T P_{ij}^T P_{ij} U_{ij} \\ &= C_{ij} + R_{ij} U_{ij} + U_{ij}^T Q_{ij} U_{ij} \end{aligned}$$

with

C_{ij}	$=$	$M_{ij}^T M_{ij}$	a scalar term,
R_{ij}	$=$	$2M_{ij}^T P_{ij}$	a 12×1 vector,
Q_{ij}	$=$	$P_{ij}^T P_{ij}$	a 12×12 matrix.

E^* is integrated over the volume of the two neighbours by a Gauss-Legendre integration scheme, which transforms the integral operator over each element into a sum over a given number of integration points. As the integrated term is a second order polynomial, it is integrated exactly with only 8 integration points for cubic elements. As compared to the analytical solution, this approach should yield exactly the same result and is easier to implement.

Finally, the integration process assembles the contribution of each pair of connected elements by gathering all the terms Q_{ij} and R_{ij}, which respectively forms Q and R, and yields Eq. (6).