Adjoint-based shape optimization at isoconnectivity through robust mesh deformation

Rigid Motion Mesh Morpher (R3M)

[1,2]George Eleftheriou
[1]Guillaume Pierrot
{george.eleftheriou,guillaume.pierrot}@esi-group.com

[1]ESI-Group
[2]National Technical University of Athens

WCCM-ECCM-ECFD Barcelona, 22 July 2014

get it right®

ABOUTflow

Motivation

Re-meshing

Trashing of the mesh at the end of each optimization cycle and generation of a new one. Time-consuming, gradient consistency lost from one cycle to the other. In some cases manual intervention during mesh generation.

Morphing

Deformation of the existing mesh. Aim: adjoint-based optimization at iso-connectivity. Challenges: avoiding twisted/heavily distorted cells, robustness issues (mesh anisotropy, mesh rotation).

The basic idea

The internal nodes of the mesh should gracefully follow the movement of boundary nodes, as indicated by the optimization algorithm.

Motivation

Re-meshing

Trashing of the mesh at the end of each optimization cycle and generation of a new one. Time-consuming, gradient consistency lost from one cycle to the other. In some cases manual intervention during mesh generation.

Morphing

Deformation of the existing mesh. Aim: adjoint-based optimization at iso-connectivity. Challenges: avoiding twisted/heavily distorted cells, robustness issues (mesh anisotropy, mesh rotation).

The basic idea

The internal nodes of the mesh should gracefully follow the movement of boundary nodes, as indicated by the optimization algorithm.

Motivation

Re-meshing

Trashing of the mesh at the end of each optimization cycle and generation of a new one. Time-consuming, gradient consistency lost from one cycle to the other. In some cases manual intervention during mesh generation.

Morphing

Deformation of the existing mesh. Aim: adjoint-based optimization at iso-connectivity. Challenges: avoiding twisted/heavily distorted cells, robustness issues (mesh anisotropy, mesh rotation).

The basic idea

The internal nodes of the mesh should gracefully follow the movement of boundary nodes, as indicated by the optimization algorithm.

Contents

Contents

Existing mesh morphing methods

Method	**Shortcomings**
Spring analogy	Not robust
Laplacian smoothening	More robust. No rotation. No mesh anisotropy.
Linear elasticity	More robust but mesh anisotropy?
Radial Basis Functions	Dense matrices, Limitations in mesh size, trade-off between computational cost & implementation simplicity

Contents

Brief introduction to R3M

Why “Rigid Motion” ?

Technically speaking it’s not “rigid”. It’s “as-rigid-as-possible”. And it’s not meant for the entire mesh (how could it be?). It’s meant for groups of nodes called **stencils**.

Stencils?

Example: A node plus its neighbouring nodes (sharing one or more cells or edges with it).

Brief introduction to R3M

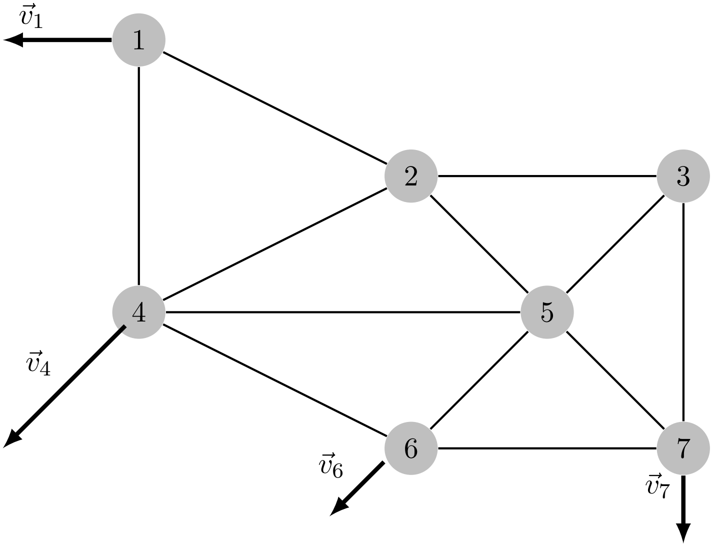
Why "Rigid Motion" ?

Technically speaking it's not "rigid". It's "as-rigid-as-possible". And it's not meant for the entire mesh (how could it be?). It's meant for groups of nodes called **stencils**.

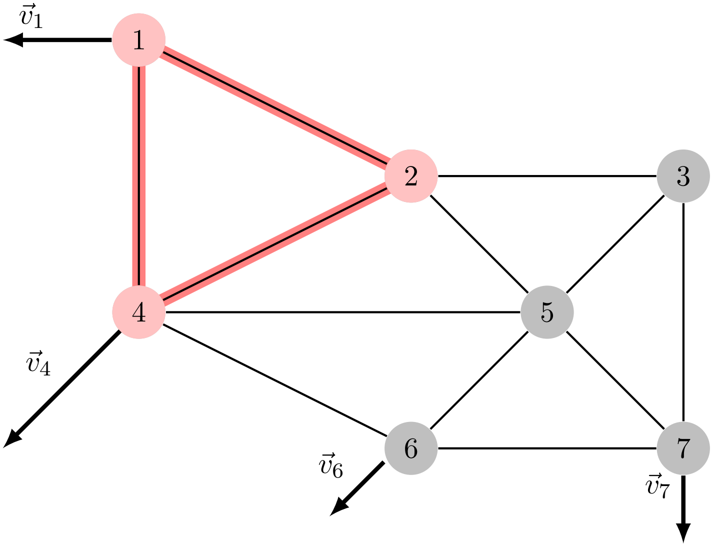
Stencils?

Example: A node plus its neighbouring nodes (sharing one or more cells or edges with it).

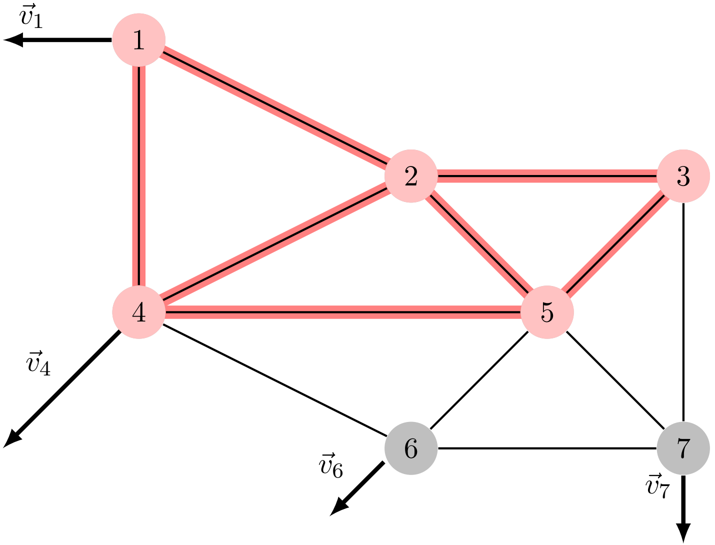
Brief introduction to R3M



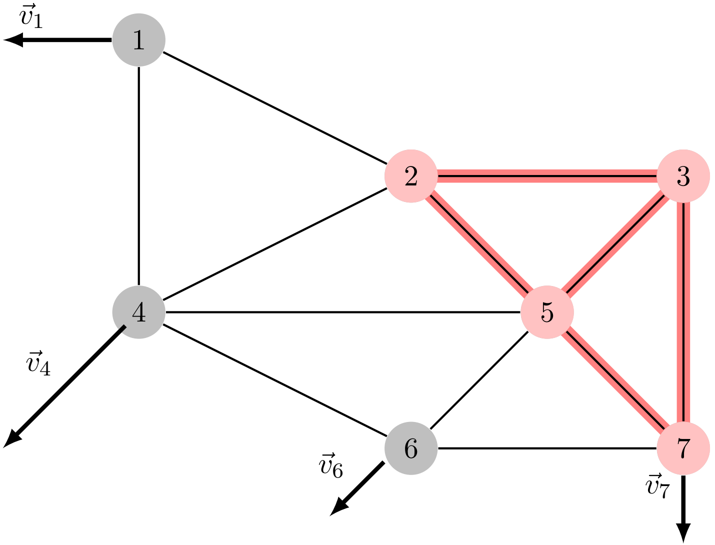
Brief introduction to R3M



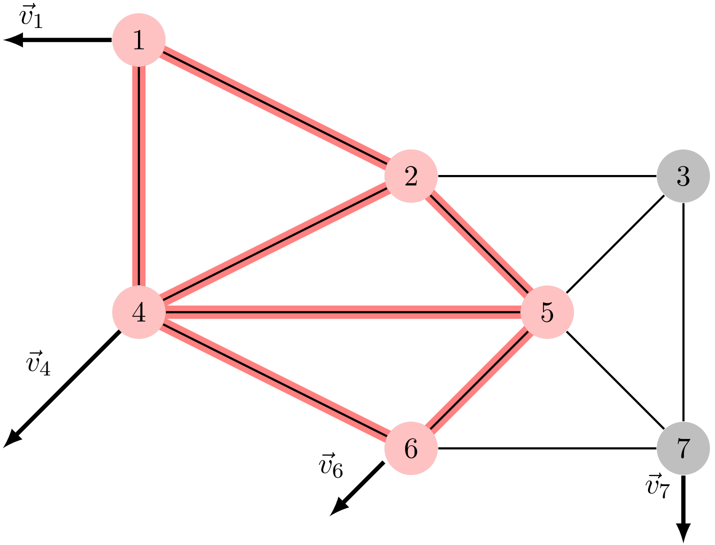
Brief introduction to R3M



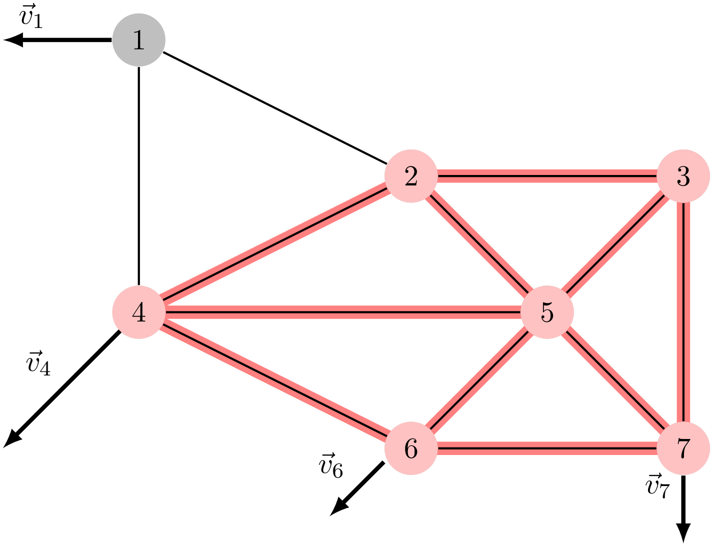
Brief introduction to R3M



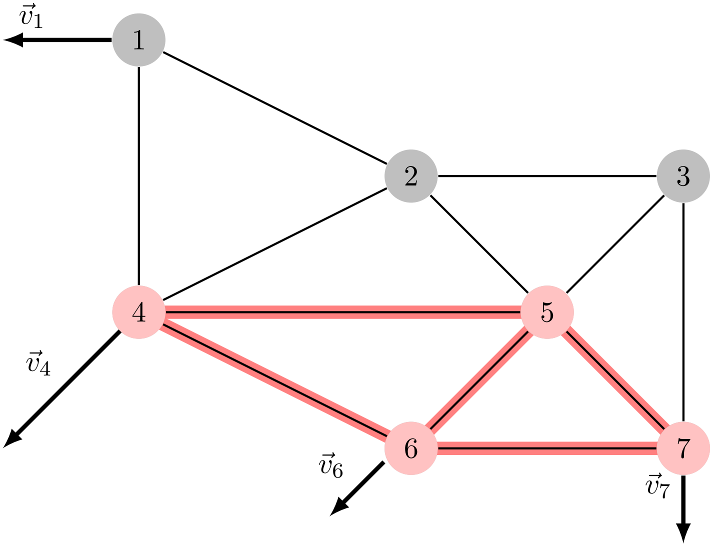
Brief introduction to R3M



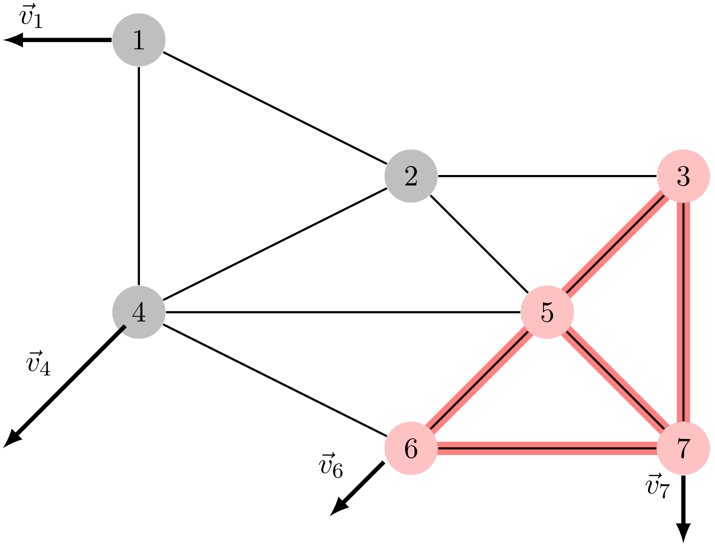
Brief introduction to R3M



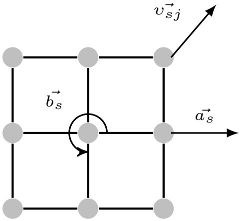
Brief introduction to R3M



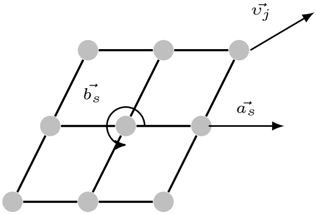
Brief introduction to R3M



Brief introduction to R3M



$$\vec{v_{sj}} = \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

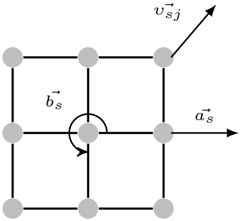


$$\vec{v_j} \neq \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

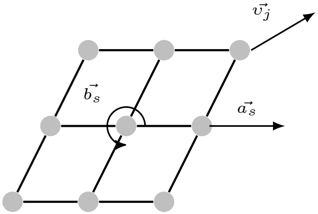
$$\vec{e_{sj}} = \vec{v_j} - \vec{v_{sj}} = \vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

$$\vec{e_{sj}} = \sqrt{w_s \mu_{sj}} (\vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s}))$$

Brief introduction to R3M



$$\vec{v_{sj}} = \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

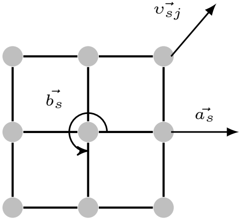


$$\vec{v_j} \neq \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

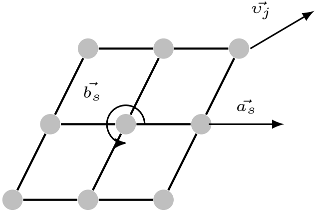
$$\vec{e_{sj}} = \vec{v_j} - \vec{v_{sj}} = \vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

$$\vec{e_{sj}} = \sqrt{w_s \mu_{sj}} (\vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s}))$$

Brief introduction to R3M



$$\vec{v_{sj}} = \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$



$$\vec{v_j} \neq \vec{a_s} + \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

$$\vec{e_{sj}} = \vec{v_j} - \vec{v_{sj}} = \vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s})$$

$$\vec{e_{sj}} = \sqrt{w_s \mu_{sj}}(\vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s}))$$

Brief introduction to R3M

The total distortion energy of the mesh will be

$$E = \sum_s \sum_{j \in s} \vec{e_{sj}}^T \cdot \vec{e_{sj}} = \sum_s \sum_{j \in s} \|\vec{e_{sj}}\|^2$$

It serves as the distortion metric which needs to be minimized.
Hence the classification of the method as "optimization based".

$$\frac{\partial E}{\partial v_j} = \frac{\partial E}{\partial a_s} = \frac{\partial E}{\partial \beta_s} = 0$$

Final system of equations

The quadratic minimization problem of the total distortion energy of the mesh, as shown above, brings us to the following symmetric positive definite system

$$\begin{bmatrix} \mathbb{A}_{uu} & \mathbb{A}_{u(a|b)} \\ \mathbb{A}_{(a|b)u} & \mathbb{A}_{(a|b)(a|b)} \end{bmatrix} \cdot \begin{bmatrix} u \\ (a|b) \end{bmatrix} = \begin{bmatrix} P_u \\ P_{(a|b)} \end{bmatrix}$$

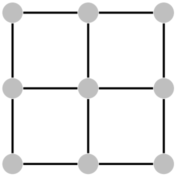
where the RHS consists of the boundary conditions, namely the prescribed nodes' velocities. Attempting to solve it using the Schur complement leads to two different cases of elimination:

Either $\quad u = f(a|b) \quad$ or

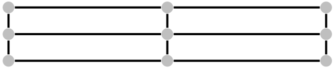
$$(a|b) = g(u)$$

Contents

μ_{sj} explained



Isotropic stencil

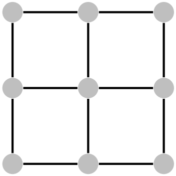


Squeezed/anisotropic stencil (we favour rigidity in the direction of squeeze)

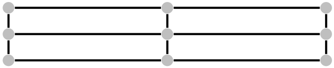
Reminder: $\vec{e_{sj}} = \sqrt{w_s \mu_{sj}}(\vec{v_j} - \vec{a_s} - \vec{b_s} \wedge (\vec{x_j} - \vec{c_s}))$

$$h_{sj} = \|\vec{x_j} - \vec{c_s}\| \rightarrow \quad h_s = \frac{\sum_{j \in S} h_{sj}}{n_s} \text{ or } h_s = \max(h_{sj})$$

μ_{sj} explained



Isotropic stencil



Squeezed/anisotropic stencil (we favour rigidity in the direction of squeeze)

$$\mu''_{sj} = \exp\left(-\frac{\|\vec{x_j} - \vec{c_s}\|^2}{h_s^2}\right) \qquad \mu'_{sj} = \frac{\mu''_{sj}}{\sum_{j \in S} \mu''_{sj}})$$

$$\mu_{js} = \frac{\mu'_{js}}{\|\vec{x_j} - \vec{c_s}\|^2 + h_s^2}$$

Relative weight stencil ⇒ node (scalar coefficient)

w_s explained (or not?)

Besides the definition of μ_{sj}, there is also need for a per-stencil coefficient, w_s in cases where we desire to “freeze” deformation of some stencils but let some other stencils evolve.

From a theoretic point of view this may seem quite arbitrary, however this necessity will become obvious after the first demonstration of results (so bear with me for now).

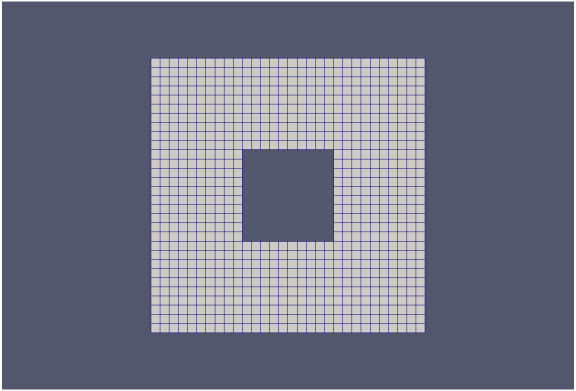
w_s explained (or not?)

Besides the definition of μ_{sj}, there is also need for a per-stencil coefficient, w_s in cases where we desire to "freeze" deformation of some stencils but let some other stencils evolve.

From a theoretic point of view this may seem quite arbitrary, however this necessity will become obvious after the first demonstration of results (so bear with me for now).

Contents

Results (45 deg)



Remarks

- Notice how the first layer of cells (read stencils) just around the critical area suffers heavy distortion...
- ... while the next layers are unharmed.
- We need to come up with a "way" of better propagating the distortion to the outer layers. A way of regulating stencils deformation depending on some "quantity".
- This "way" is the w_s coefficient. In the just demonstrated case it was $w_s = 1.0$.
- And the "quantity" it could depend on, is the accumulated stencil distortion energy ($E_s = \sum_{j \in S} \|\vec{e_{sj}}\|^2$) over time.
- Hence $w_s = f(\int_t E_s dt)$.

Remarks

- Notice how the first layer of cells (read stencils) just around the critical area suffers heavy distortion...
- ... while the next layers are unharmed.
- We need to come up with a "way" of better propagating the distortion to the outer layers. A way of regulating stencils deformation depending on some "quantity".
- This "way" is the w_s coefficient. In the just demonstrated case it was $w_s = 1.0$.
- And the "quantity" it could depend on, is the accumulated stencil distortion energy ($E_s = \sum_{j \in S} \|\vec{e_{sj}}\|^2$) over time.
- Hence $w_s = f(\int_t E_s dt)$.

Remarks

- Notice how the first layer of cells (read stencils) just around the critical area suffers heavy distortion...
- ... while the next layers are unharmed.
- We need to come up with a "way" of better propagating the distortion to the outer layers. A way of regulating stencils deformation depending on some "quantity".
- This "way" is the w_s coefficient. In the just demonstrated case it was $w_s = 1.0$.
- And the "quantity" it could depend on, is the accumulated stencil distortion energy ($E_s = \sum_{j \in S} \|\vec{e_{sj}}\|^2$) over time.
- Hence $w_s = f(\int_t E_s dt)$.

Remarks

- Notice how the first layer of cells (read stencils) just around the critical area suffers heavy distortion...
- ... while the next layers are unharmed.
- We need to come up with a "way" of better propagating the distortion to the outer layers. A way of regulating stencils deformation depending on some "quantity".
- This "way" is the w_s coefficient. In the just demonstrated case it was $w_s = 1.0$.
- And the "quantity" it could depend on, is the accumulated stencil distortion energy ($E_s = \sum_{j \in S} \|\vec{e_{sj}}\|^2$) over time.
- Hence $w_s = f(\int_t E_s dt)$.

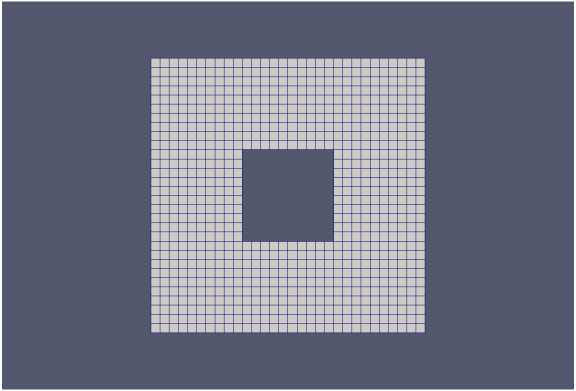
Remarks

- Notice how the first layer of cells (read stencils) just around the critical area suffers heavy distortion...
- ... while the next layers are unharmed.
- We need to come up with a "way" of better propagating the distortion to the outer layers. A way of regulating stencils deformation depending on some "quantity".
- This "way" is the w_s coefficient. In the just demonstrated case it was $w_s = 1.0$.
- And the "quantity" it could depend on, is the accumulated stencil distortion energy ($E_s = \sum_{j \in S} \|\vec{e_{sj}}\|^2$) over time.
- Hence $w_s = f(\int_t E_s dt)$.

Remarks

- Notice how the first layer of cells (read stencils) just around the critical area suffers heavy distortion...
- ... while the next layers are unharmed.
- We need to come up with a "way" of better propagating the distortion to the outer layers. A way of regulating stencils deformation depending on some "quantity".
- This "way" is the w_s coefficient. In the just demonstrated case it was $w_s = 1.0$.
- And the "quantity" it could depend on, is the accumulated stencil distortion energy ($E_s = \sum_{j \in S} \|\vec{e_{sj}}\|^2$) over time.
- Hence $w_s = f(\int_t E_s dt)$.

Results (70 deg)



Remarks

- $w_s = \frac{1}{n_s} \exp(\frac{I_s}{\sum_S I_s})$
- All stencils are affected. From inner to outer layers, all stencils (read cells) gradually saturate and freeze. Which is good...
- ...but: what if we decide to reverse the rotation? Can we get the mesh back to its "sane" initial state?
- We can't. The deformations caused are irreversible. Problem: the weight w_s depends on the accumulated stencil deformation energy which is an ever increasing (as a sum of square norms) quantity, thus every deformation (even towards the direction of mesh "improvement") is perceived as negative and deserving to be penalized.
- Therefore the stencils become saturated and rigid...

Remarks

- $w_s = \frac{1}{n_s} \exp(\frac{I_s}{\sum_S I_s})$
- All stencils are affected. From inner to outer layers, all stencils (read cells) gradually saturate and freeze. Which is good...
- ...but: what if we decide to reverse the rotation? Can we get the mesh back to its "sane" initial state?
- We can't. The deformations caused are irreversible. Problem: the weight w_s depends on the accumulated stencil deformation energy which is an ever increasing (as a sum of square norms) quantity, thus every deformation (even towards the direction of mesh "improvement") is perceived as negative and deserving to be penalized.
- Therefore the stencils become saturated and rigid...

Remarks

- $w_s = \frac{1}{n_s}\exp(\frac{I_s}{\sum_S I_s})$
- All stencils are affected. From inner to outer layers, all stencils (read cells) gradually saturate and freeze. Which is good...
- ...but: what if we decide to reverse the rotation? Can we get the mesh back to its "sane" initial state?
- We can't. The deformations caused are irreversible. Problem: the weight w_s depends on the accumulated stencil deformation energy which is an ever increasing (as a sum of square norms) quantity, thus every deformation (even towards the direction of mesh "improvement") is perceived as negative and deserving to be penalized.
- Therefore the stencils become saturated and rigid...

Remarks

- $w_s = \frac{1}{n_s} \exp(\frac{I_s}{\sum_S I_s})$
- All stencils are affected. From inner to outer layers, all stencils (read cells) gradually saturate and freeze. Which is good...
- ...but: what if we decide to reverse the rotation? Can we get the mesh back to its "sane" initial state?
- We can't. The deformations caused are irreversible. Problem: the weight w_s depends on the accumulated stencil deformation energy which is an ever increasing (as a sum of square norms) quantity, thus every deformation (even towards the direction of mesh "improvement") is perceived as negative and deserving to be penalized.
- Therefore the stencils become saturated and rigid...

Remarks

- $w_s = \frac{1}{n_s} \exp(\frac{I_s}{\sum_S I_s})$
- All stencils are affected. From inner to outer layers, all stencils (read cells) gradually saturate and freeze. Which is good...
- ...but: what if we decide to reverse the rotation? Can we get the mesh back to its “sane” initial state?
- We can’t. The deformations caused are irreversible. Problem: the weight w_s depends on the accumulated stencil deformation energy which is an ever increasing (as a sum of square norms) quantity, thus every deformation (even towards the direction of mesh “improvement”) is perceived as negative and deserving to be penalized.
- Therefore the stencils become saturated and rigid...

Towards even better weighting...

Taking into account only the mesh quality parameters (such as non-orthogonality/skewness numbers) into the (dynamic) calculation of w_s. E.g $w_s = f(\text{N-O}, \text{skewness})$. This will allow for more delicate, targeted and certainly reversible deformations. (CAUTION! The point IS NOT to improve the mesh. It is just to avoid turning it into garbage!)

Contents

Conclusions / Next Steps

R3M is/does:

- Essentially mesh-less (only needs nodes and not cell or inertial data)
- Manage intrisically mesh anisotropy and rotation.

Next steps:

- Coupling with ESI's i-adjoint solver for automated optimization loops.
- Preservation of feature lines.

Acknowledgements

This work has been conducted within the **About Flow** project on "Adjoint-based optimisation of industrial and unsteady flows".

`http://aboutflow.sems.qmul.ac.uk`

About Flow has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under Grant Agreement No. 317006.

National Technical University of Athens