Mesh movement strategy based on octree decomposition

M. Gómez, M. Cordero-Gracia, J. Ponsin, E. Valero
E.T.S.I. Aeronáuticos, Madrid, Spain
Problem: Transfer deformations between meshes

BodyTransfer
- Transfer deformations from a structural mesh to an aerodynamic or surface mesh.

MeshMove
- Transfer deformations from an aerodynamic mesh to a volumetric mesh.
Transfer deformations from a structural mesh (CSM grid) to an aerodynamic mesh (CFD grid)
Transfer deformations from an aerodynamic mesh (CFM grid) to a volumetric mesh (CFD grid)
Objetives

- Transfer deformations from the boundary mesh to the volumetric mesh for a wide range of perturbations.

- Efficiency in computational resources (time and memory).

- Mesh quality preservation.

- Applicability to any 3D data set (any kind of 3D meshes: structured, multiblock structured, unstructured and hybrid).
**Methodology:**
- Interpolation of deformations with *Radial Basis Functions*.

**Strategy:**
- Definition of interpolation domains that cover the whole mesh.
- Generation and Management of domains with *Octree data structure*
- Definition of an optimal ordering to the domain deformation sequence.
Interpolation
Given $N_s$ centers \( \{x_1^s, \ldots, x_{N_s}^s\} \) and their displacements \( \{h_1^s, \ldots, h_{N_s}^s\} \), and $N_a$ evaluation nodes \( \{x_1^a, \ldots, x_{N_a}^a\} \).

The problem consists on obtaining the displacements \( \{h_1^a, \ldots, h_{N_a}^a\} \) via interpolation methods, in a smooth and regular way.
Reconstruct a continuous spatial distribution $h(\bar{x})$ using the discrete values $\bar{x}_i^s$

$$h(\bar{x}) = \sum_{i=1}^{N_s} w_i \Phi(||\bar{x} - \bar{x}_i^s||) + \Pi(\bar{x})$$

where

- $w_i$ are the coefficients.
- $\Phi$ is a fixed basis function which is radial with respect to the Euclidean distance (*Radial Basis Function*)
- $\Pi$ is a $m$ degree polynomial that depends on the $\Phi$ function.
**Interpolation condition**

\[ h_i^s \equiv h(\bar{x}_i^s) \]

**Zero condition**

\[ \sum_{i=1}^{N_s} w_i q(\bar{x}_i) = 0 \]

for all polynomials \( q \) with a degree \( \deg(q) \leq \deg(\Pi) \)

To avoid transfer of fictitious displacements, zero degree polynomials are required

\[ \Pi(\bar{x}) = \gamma_0 \implies \sum_{i=1}^{N_s} w_i = 0 \]
RBF Interpolation

Coefficients computation

\[ h^s_i = h(\bar{x}^s_i) \quad i = 1, \ldots, N_s \]
\[ \sum w_i = 0 \]

\( \left( \begin{array}{c}
0 \\
h^s_1 \\
h^s_2 \\
\vdots \\
h^s_{N_s}
\end{array} \right) \quad = \quad \left( \begin{array}{ccccc}
0 & 1 & 1 & \cdots & 1 \\
1 & \Phi_{s_1s_1} & \Phi_{s_1s_2} & \cdots & \Phi_{s_1s_{N_s}} \\
1 & \Phi_{s_2s_1} & \Phi_{s_2s_2} & \cdots & \Phi_{s_2s_{N_s}} \\
\vdots & \cdots & \cdots & \cdots & \cdots \\
1 & \Phi_{s_{N_s}s_1} & \Phi_{s_{N_s}s_2} & \cdots & \Phi_{s_{N_s}s_{N_s}}
\end{array} \right) \quad \left( \begin{array}{c}
\gamma_0 \\
w_1 \\
w_2 \\
\vdots \\
w_{N_s}
\end{array} \right) \]

\[ \bar{h}^s = C_{ss} \bar{\omega} \]
RBF Interpolation

Applying to evaluation nodes

\[ h^a_i = h(\bar{x}^a_i) = \sum_{k=1}^{N_s} w_k \Phi(||\bar{x}^a_i - \bar{x}^s_k||) + \gamma_0 \quad i = 1, \ldots, N_a \]

\[
\begin{pmatrix}
  h^a_1 \\
  h^a_2 \\
  \vdots \\
  h^a_{N_a}
\end{pmatrix}
\begin{pmatrix}
  1 & \Phi_{a_1s_1} & \Phi_{a_1s_2} & \cdots & \Phi_{a_1s_{N_s}} \\
  1 & \Phi_{a_2s_1} & \Phi_{a_2s_2} & \cdots & \Phi_{a_2s_{N_s}} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  1 & \Phi_{a_{N_a}s_1} & \Phi_{a_{N_a}s_2} & \cdots & \Phi_{a_{N_a}s_{N_s}}
\end{pmatrix}
\begin{pmatrix}
  \gamma_0 \\
  w_1 \\
  w_2 \\
  \vdots \\
  w_{N_s}
\end{pmatrix}
\]

\[ \bar{h}^a = A_{as} \bar{\omega} = A_{as} C_{ss}^{-1} \bar{h}^s \]
## Function Definition

<table>
<thead>
<tr>
<th>Function</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume Spline</td>
<td>$</td>
</tr>
<tr>
<td>Wendland $C^0$</td>
<td>$(1 -</td>
</tr>
<tr>
<td>Wendland $C^2$</td>
<td>$(1 -</td>
</tr>
<tr>
<td>Wendland $C^4$</td>
<td>$(1 -</td>
</tr>
</tbody>
</table>
RBF interpolation method

**Advantages**

- Valid for any dimensional problem.
- Translation and rotation invariant.
- Smooth representation.
- Possibility to manage any kind of 3D data (only depend on the distance between nodes).

**Disadvantages**

- Time and memory consumption.
Mesh movement methodology
MeshMove tool

Problems

Numerical viability
- Mesh size

Accuracy
- Distance between centers and evaluation nodes

Strategy

Generate local interpolation domains
- Enough centers to guarantee a reliable interpolation,
- Not too many to ensure a manageable $C_{ss}$ matrix size.
Non-disjoint sets of centers and evaluation nodes, linked by their proximity.
Non-disjoint sets of centers and evaluation nodes, linked by their proximity.

Octree data structure
Interpolation domains

- Non-disjoint sets of centers and evaluation nodes, linked by their proximity.

- The whole mesh is recovered by the union of all domains.

Octree data structure
Each domain represents a full interpolation problem.

A domain is characterized by

- The centers and evaluation nodes that form its own $C_{ss}$ and $A_{as}$ matrices.
- Its position index in the ordered sequence of domains.
An interpolation domain is made up of:

- one kernel cube containing the evaluation nodes
- and all neighbour cubes
An interpolation domain is made up of:

- one kernel cube containing the evaluation nodes
- and all neighbour cubes

As the interpolation is going on the domain’s sequence

- the evaluation nodes of the neighbour cubes already deformed, act as centers.
**Advancing front strategy**

Information travels through concentric layers surrounding the boundary.

**From the surface to the farfield**

Domains whose kernel contains surface nodes make up the first layer.

Once the first domain has been chosen, it is possible to cover the whole boundary travelling across neighbouring kernels.
Interpolation sequence
Mesh movement strategy based on octree decomposition—p. 20
Interpolation sequence
Interpolation sequence

Mesh movement strategy based on octree decomposition—p. 20
Computational strategy

**Computational bottlenecks**
- Number of interpolation domains.
- Maximum size of the interpolation matrix in each domain.

**Solution**
- Two user-provided parameters.
Computational strategy

**Computational bottlenecks**

- Number of interpolation domains.
- Maximum size of the interpolation matrix in each domain.

**Solution**

- Two user-provided parameters.

  **Max. number of nodes within an octree cube.**

  - It controls the number of interpolation domains.
  - It is related with the preprocessing time.
Computational strategy

**Computational bottlenecks**

- Number of interpolation domains.
- Maximum size of the interpolation matrix in each domain.

**Solution**

- Two user-provided parameters.
- **Max. number of centers within an interpolation domain.**
  - It controls the size of the interpolation matrix $C_{ss}$
  - It is related with the evaluation time
Mesh quality metrics
Two quality algebraic metrics have been incorporated to measure the quality of the deformed mesh in order to:

- Prove the validity of the methodology.
- Stop the computation cycle: Maximum allowed deformation when any of the quality parameters go below a prescribed threshold (degenerated mesh).
Mesh quality metrics

**Relative size metric**

\[ f_{\text{size}} = \min \{ \tau, \frac{1}{\tau} \} \]

Degenerated deformed element \( \implies f_{\text{size}} < 0 \)

**Shape metric**

\( f_{\text{shape}} \) combination of skew metric and element edge-length ratios

Three edges at one vertex coplanar \( \iff f_{\text{shape}} = 0 \)
Numerical results
34,000 surface mesh nodes
180,000 volumetric mesh nodes
Types of deformations

**Torsion**

\[ \varphi(y) = \frac{y}{L} \varphi_L \]

\[ \varphi_L \equiv \text{twist at wing tip} \]

**Bending**

\[ \eta(y) = \frac{y^2(6L^2 - 4Ly + y^2)}{3L^4} \]

\[ \eta \equiv \text{vertical displacement} \]
Robustness tests

- Maximum deformation running the algorithm once.
- Maximum deformation running it iteratively.

<table>
<thead>
<tr>
<th>Test</th>
<th>spline</th>
<th>Wend. $C^0$</th>
<th>Wend. $C^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_{L_{max}}$ one step</td>
<td>51°</td>
<td>37°</td>
<td>34°</td>
</tr>
<tr>
<td>$\varphi_{L_{max}}$ multiple steps</td>
<td>100°</td>
<td>65°</td>
<td>50°</td>
</tr>
<tr>
<td>$\eta_{max}$ one step</td>
<td>54 % $L$</td>
<td>38 % $L$</td>
<td>34 % $L$</td>
</tr>
<tr>
<td>$\eta_{max}$ multiple steps</td>
<td>100 % $L$</td>
<td>100 % $L$</td>
<td>100 % $L$</td>
</tr>
</tbody>
</table>

Mesh quality condition: \[ f_{\text{size}} > 0 \quad \text{and} \quad f_{\text{shape}} > 0 \] Both close to 1

Mesh movement strategy based on octree decomposition—p. 28
70% twist at wing tip

Close to wing tip (80% span)  Leading edge

Mean values of quality parameters:

\[ f_{size} = 0.996 \quad f_{shape} = 0.9996 \]
Viscous mesh. ONERA M6

Original mesh

Deformed mesh

(Wing rotation of 10°)

- 43,200 surface mesh nodes
- 1,500,000 volumetric mesh nodes

Mesh movement strategy based on octree decomposition— p. 30
Two flow computations have been carried on using DLR_TAU code (Mach=0.2, Re = $11.2 \times 10^6$)

1. Over the original mesh with $\alpha = 10^o$

2. Over the deformed mesh with $\alpha = 0^o$

Mean values of quality parameters:

$$f_{size} = 0.984 \quad f_{shape} = 0.995$$
Pressure coefficient distributions at sections of wing

Mesh movement strategy based on octree decomposition—p. 32
Conclusions

- An general interpolation tool based on RBFs together with an advancing front strategy for moving 3D meshes has been developed.

- It can be applied to any kind of meshes (structured or unstructured).

- It is robust, efficient and preserves the quality of the original mesh for very large deformation.

- It can be parallelized because of the inherent domain decomposition strategy.
Thank you