*Progress in Propulsion Physics* **1** (2009) 305-318
DOI: 10.1051/eucass/200901305


# VARIABLE TRANSFER METHODS FOR FLUID–STRUCTURE INTERACTION COMPUTATIONS WITH STAGGERED SOLVERS

**J. M. Vaassen, I. Klapka, B. Leonard, and C. Hirsch**

This paper intends to study methods that have been tested to transfer variables from one skin mesh to another (the two meshes being non-conform) in order to compute fluid–structure interaction (FSI) problems with staggered solvers. The methods are a contact elements method developed by Stam, and different radial basis functions methods. The structure code is OOFELIE® developed at Open-Engineering (Belgium) and the fluid code is FINE™/Hexa developed at Numeca International (Belgium). The paper presents the performances of the methods on a simple variable transfer, and testcases that have been performed with the solver developed by the two companies.

## 1 INTRODUCTION

In FSI problems, a fluid flow imposes forces (due to pressure and viscous stresses) on a structure that is able to be deformed. This deformation modifies the fluid domain and the boundary conditions at the FSI, which modifies the fluid flow itself. A FSI problem is composed of three main equations

$$\begin{aligned} \frac{\partial}{\partial t}\,\vec{u}_F &= \overrightarrow{\mathrm{RHS}}_F(\vec{u}_F, \vec{u}_S, \vec{u}_G)\,; \\ M\frac{\partial^2}{dt^2}\,\vec{u}_S + C\frac{\partial}{\partial t}\,\vec{u}_S &= \overrightarrow{\mathrm{RHS}}_S(\vec{u}_F, \vec{u}_S)\,; \\ 0 &= \overrightarrow{\mathrm{RHS}}_G(\vec{u}_S, \vec{u}_G) \end{aligned}$$

The first equation computes the fluid solution $u_F$, depending on the structure solution $u_S$ and the fluid mesh deformation $u_G$. The second one computes the structure solution $u_S$, depending on the fluid solution $u_F$. The last one computes the mesh deformation $u_G$, depending on the structure solution $u_S$.

To compute FSI problems, two approaches are available:

(1) the monolithic approach: unique software is developed to solve the global problem (aggregated equations). All the problems are thus solved simultaneously; and

(2) the staggered (partitioned) approach: different codes are used for each of the problems (segregated equations). These solvers exchange information at synchronization points only.

The approach that is used in this paper is the second one. Two industrial codes have been coupled to create a global solution able to compute FSI problems. These codes are:

– OOFELIE® developed at Open-Engineering. This software is a unified object-oriented development environment, a multiphysics, multifield, and multielement computer modeling tool for strong coupling based on finite elements, boundary elements, and others.

– FINE™/Hexa developed at Numeca International. It is a full automated and integrated software package able to compute three-dimensional (3D) fluid flows in various regimes (from incompressible to hypersonic) with turbulence models and deforming meshes. It uses a cell-centered finite-volume solver [1].

The main issue is such an approach is the transfer of information from one code to the other. This imposes:

– the transfer of the positions, velocities, and temperature from the structure skin mesh to the fluid one to impose the equality of these values between the two meshes (kinematic boundary conditions); and

– the transfer of the forces and heat flux from the fluid skin mesh to the structure one, to transform the aerodynamic loads into respective work equivalent forces on the structure mesh (compatibility conditions).

These transfers are performed in two steps:

(1) the transfer from one code to another, which is done by Message Passing Interface (MPI) communication functions; and

(2) the transfer from one mesh to another.

This last point is crucial. Indeed, as a fluid problem generally needs much more nodes than a structure one (to capture vortices or shock waves), the two skin meshes are chosen to be nonconform. The transfer of the variables from one skin mesh to another is thus a real issue. Several transfer methods are studied in this paper and compared on FSI problems.

The next section is dedicated to the presentation of the available transfer methods. Section 3 presents results for the transfer of imposed functions on a cylindrical mesh. Section 4 presents results obtained when using the FSI code to compute the deformation of a beam due to the overpressure behind a shock wave. Section 5 presents results for a subsonic flow around a wind turbine blade.

## 2 VARIABLE TRANSFER THROUGH NONCONFORM SKIN MESHES

The interaction of a fluid with a structure is driven by the exchange of mass, momentum, and energy at their common interface (what is lost on one side is gained on the other side). The correct representation of this exchange is crucial, but is very difficult to do when the problem is discretized. In fact, it can be demonstrated that if the conservation is ensured for the discrete problem, the accuracy of the global scheme is several orders of magnitude higher than if it is not. Moreover, a violation of the conservation properties can result in instabilities in the numerical solution.

### 2.1 Conservation Law

The kinematic boundary condition imposes the equality of a variable $s$ (position, velocity, or temperature) on each side of the interface. A discrete form of this condition is written as

$$\vec{s}_F = H\vec{s}_S$$

where indices $F$ and $S$ are the fluid and structure side of the interface, respectively; and $H$ is the transfer matrix which will be computed by one of the methods described later. To ensure the equality of the virtual works of the fluid and the structure acting on the interface, it can be demonstrated [2] that the following equation has to be solved:

$$\vec{F}_S = H^T\vec{F}_F$$

The challenging problem is thus to compute a matrix that interpolates values from the structure skin mesh to the fluid skin one, the second being generally much more refined than the first one. Two methods are presented hereunder.

### 2.2 Contact Dynamic Method

Suppose that the structure skin mesh is composed of a set of triangles. Any point inside such a triangle can be characterized by two curvilinear coordinates: $v \in [0, 1]$ and $w \in [0, v]$.

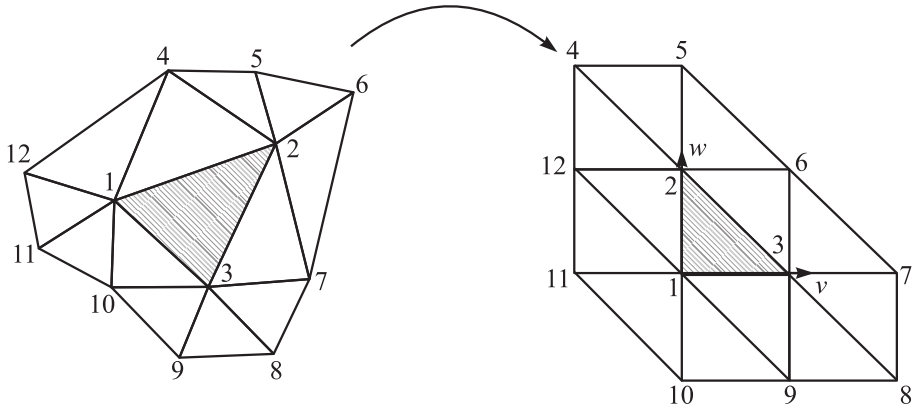


**Figure 1** Stencil reported in $v$ and $w$ coordinates to compute the weights

Around every single triangle, a stencil is created, grouping a list of $N$ neighboring nodes (all the nodes forming the triangles adjacent to each of the nodes of the considered triangle). It is then possible, for each of the triangles, to compute weights $b_i(v, w)$ such that the position of any point on the triangle is expressed as a linear combination of the positions of the stencil nodes:

$$\mathbf{x}(v, w) = \sum_{i=1}^{N} b_i(v, w)\mathbf{x}_i \,.$$

If the three nodes of the triangle have exactly 6 adjacent triangles (Fig. 1), the computation of these weights is straightforward [3].

If it is not the case, Stam [3] proposed a technique of subdivision of the triangles, so that $b_i(v, w)$ can be evaluated.

The fluid skin mesh is considered as a list of nodes. The connectivity is not considered to compute $H$. By projecting each fluid node on the structure interface, it is possible to determine in which triangle the fluid node is located and the coordinates $v$ and $w$ of this projection in the triangle. When the structure nodes $\mathbf{x}_i$ are moving, the fluid nodes are moving according to the previous formula. The interpolation of any scalar quantity (to ensure the kinematic boundary conditions) can then be made by using the same weights.

### 2.3 Radial Basis Functions Methods

Suppose that a scalar quantity $u$ is given at the structure nodes

$$X = \{x_1, \ldots, x_N\}$$

and that this information is to be transferred to the fluid nodes

$$Y = \{y_1, \ldots, y_M\}\,.$$

A Radial Basis Function (RBF) interpolant is a function of the form

$$s(x) = \sum_{j=1}^{N} \alpha_j \phi\left(\| x - x_j \|_2\right) + p(x)$$

where $\phi$ is a conditionally positive definite function of a certain order $m$ and $p$ is a polynomial of degree $m-1$ [4, 5]. As is made by many authors, only $m = 2$ is considered in this work. The conditions that have to be imposed are

$$s(x_j) = u_j\,; \qquad \sum_{j=1}^{N} \alpha_j q(x_j) = 0 \quad \forall q \text{ polynomial of order } m-1\,.$$

If $\{p_1, \ldots, p_Q\}$ is a basis of the space of polynomial of order $m-1$, any polynomial $p$ or $q$ can be written as

$$p(x) = \sum_{k=1}^{Q} \beta_k p_k(x)$$

and the second condition is then reads

$$\sum_{j=1}^{N} \alpha_j p_k(x_j) = 0\,, \qquad k = 1, \ldots, Q\,.$$

It is then possible to define the following matrices:

$$A = \phi\left(\| x_k - x_j \|_2\right) \in \mathcal{R}^{N\times N}\,; \qquad P = p_k(x_j) \in \mathcal{R}^{N\times Q}\,;$$
$$\tilde{A} = \phi\left(\| y_i - x_j \|_2\right) \in \mathcal{R}^{M\times N}\,; \qquad \tilde{P} = p_k(x_i) \in \mathcal{R}^{M\times Q}$$

so that the two conditions are written in a matrix form:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} u \\ 0 \end{pmatrix}\,.$$

The values on the fluid skin mesh are finally computed by the expression (giving directly the transfer matrix $H$):

$$s|Y = \begin{pmatrix} \tilde{A} & \tilde{P} \end{pmatrix} \begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} u \\ 0 \end{pmatrix} = C \begin{pmatrix} u \\ 0 \end{pmatrix}\,.$$

The following functions have been tested in this work:

– Thin-Plate Splines (TPS)

$$\phi\left(\|\mathbf{X}\|\right) = \|\mathbf{x}\|^2 \log\left(\|\mathbf{x}\|\right) ;$$

– Euclide Hat (EU)

$$\phi\left(\|\mathbf{x}\|\right) = \left(1 - \frac{\|\mathbf{x}\|}{r}\right)_+^2 \left(\frac{1}{2}\,\frac{\|\mathbf{x}\|}{r} + 1\right) ;$$

– Beckert–Wendland (BW)

$$\phi\left(\|\mathbf{x}\|\right) = \left(1 - \frac{\|\mathbf{x}\|}{r}\right)_+^4 \left(4\frac{\|\mathbf{x}\|}{r} + 1\right) .$$

In the last two functions, $r$ is a parameter specified by the user, and $+$ means that if what is inside the parenthesis is negative, the function is equal to zero. These two functions are said to be compactly supported functions, because they are only nonzero in a neighborhood of a node (specified by the value of $r$).

As presented here, the RBF method demands the inversion of a square matrix that has a size equal to the double of the number of structure skin nodes. For large industrial problems, this inversion can be very costly, and the method cannot be used. An improvement of this method is thus to do. At the beginning of the computation, every interface is divided into a certain number of patches, whose size is prescribed by the user (size $N$). Every fluid skin node is first localized in the patch, and the global transfer is solved by part. An RBF function is computed on every patch, and the variables at the fluid skin nodes are interpolated through the corresponding function. The larger the number of patches, the more precise is the interpolation, but also the higher the computational costs.

## 3 COMPARISON OF THE METHODS ON A SIMPLE TRANSFER

In order to check the accuracy of the transfer methods and the choice of the parameters, a simple study is first considered. This consists in transferring a prescribed function from a structure skin mesh to a fluid one, finer. The domain that has been chosen is a disk with a radius equal to 1. The problem is thus two-dimensional (2D) and the meshes are nonconform (structure skin mesh: 30 nodes / fluid skin mesh: 120 nodes).

A predefined function has been imposed on the structure skin nodes and transferred to the fluid ones. In the last methods, two parameters can be set:

(1) the radius $r$ (only defined in the EU and BW functions); and

(2) the number $N$ of cells forming a patch. The larger $N$, the bigger the matrix to invert.

The study is focused on two particular functions:

(1) a sinusoid with a small period (0.2), so that the function is badly discretized on the structure skin mesh; and

(2) a double step function, to study the transfer of a field presenting discontinuities similar to those typical of the temperature in supersonic flows.

Figure 2 illustrates the interpolation of the sinusoid with the contact method (Fig. 2*a*) and the double step with the contact method and RBF–EU (Fig. 2*b*). The solid line is the exact function, the crosses are the values at the structure skin nodes, and the circles are the interpolated values on the fluid skin nodes.

It can be seen in these two examples that the interpolation is good, despite the very poor quality of the structure mesh. To provide a better comparison of the different methods, a mean error between the exact and interpolated solutions has been computed for each method, with different values of $N$ for the RBF methods. This error is plotted against parameter $r$. For the contact method and RBF–TPS, this error is obviously constant (Fig. 3).

The conclusions are:

– It is necessary to take radius $r$ that is at least equal to the typical dimension of the problem;

– Increasing the size $N$ of the patches provides smaller errors, but the computational cost is higher. It can be demonstrated that the cost is proportional to $N^2$. Some tests have shown that taking a value higher than 15 (even in 3D) is prohibitive because it does not decrease the errors but is much more costly; and

– For smooth functions that are badly discretized (like the sine function), the RBF–BW seems to provide the best results. However, if the function to interpolate presents discontinuities, it is the worst method (too dispersive), while the RBF–EU method seems to be the best one (the only one to be sufficiently dissipative to not introduce oscillations in the interpolated function). Radial basic function – thin-plate splines and contact methods are good intermediate, if the user has no idea on the behavior of the function to interpolate. For very smooth functions, all the methods are equivalent.

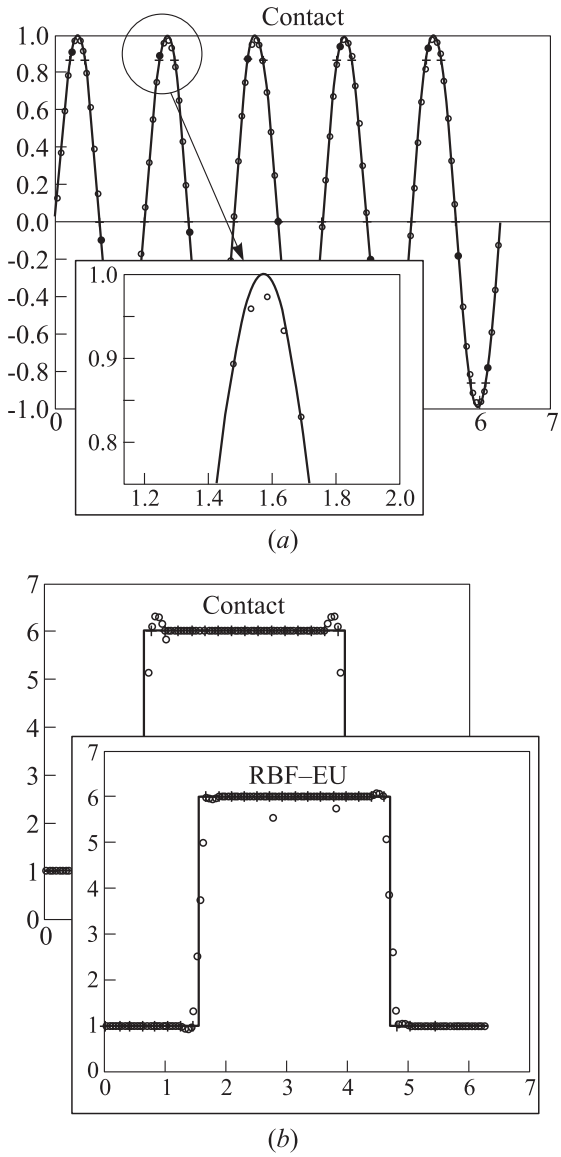


**Figure 2** Interpolation of a sinusoid ($a$) and a double ($b$) step

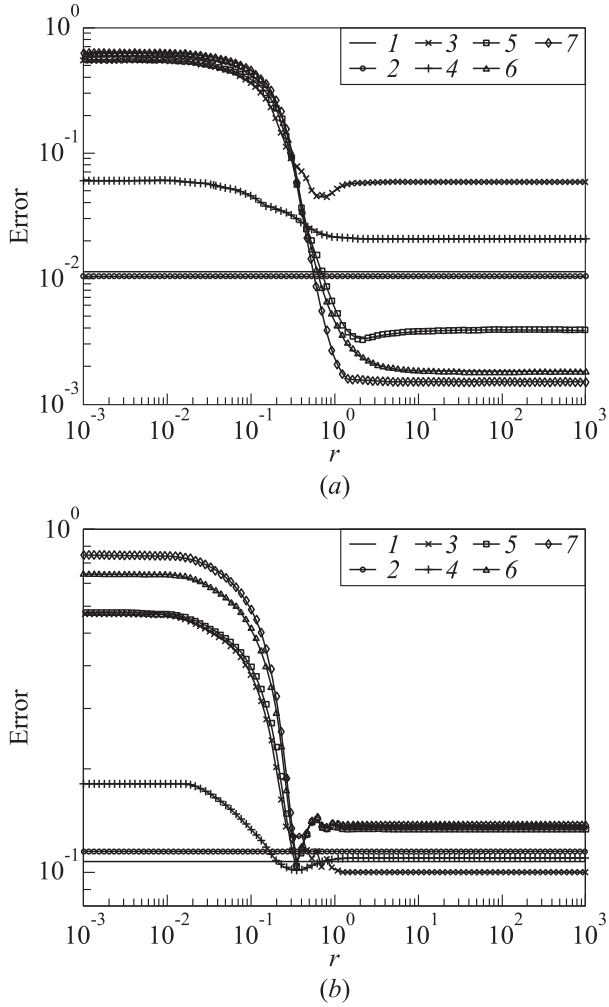


**Figure 3** Mean interpolation error for a sinusoid (*a*) and a double (*b*) step: *1* — contact; *2* — RBF–TPS, $N = 8$; *3* — RBF–EU, $N = 8$; *4* — RBF–BW, $N = 2$; *5* — RBF–BW, $N = 8$; *6* — RBF–BW, $N = 20$; and *7* — RBF–BW, $N = 60$

## 4 HYPERSONIC COMPRESSION BEAM

This problem is interesting because it has an almost analytical solution. An inviscid hypersonic flow (inlet Mach number = 6) hits a ramp (0.1 m high) with an angle of 15° (Fig. 4). A shock wave is developed and creates an important com-

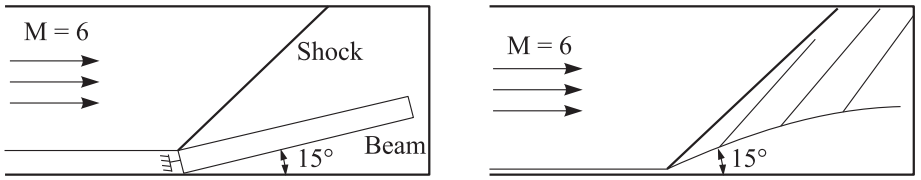


**Figure 4** Representation of the compression beam problem

pression behind it. The ramp, considered as a beam, is then deformed according to the equation

$$d\Phi = -\frac{M}{EI}\, ds$$

where $\Phi$ is the infinitesimal rotation of the beam, $M$ is the momentum, $E$ is the Young modulus, $I$ is the inertia of the structure, and $s$ is the abscissa along the ramp. Because of the curvature, an expansion wave appears in the flow and progressively decreases the pressure, which decreases the force applied to the beam. The flow and the deformation of the structure are thus coupled. This theory, however, neglects the reflections of the expansion waves with the shock wave. These interactions can affect the shock wave (that is then weaker) and thus reduce the pressure behind it.

The Rankine–Hugoniot and Prandtl–Meyer theories can compute the value of the pressure (and thus the momentum) after the shock and after an infinitesimal expansion wave $d\Phi \Rightarrow$ on each point of the ramp. This results in an ordinary differential equation, giving the Mach number on the ramp, from which one can compute the pressure and the rotation of the beam. The solution of the converged problem is given in Fig. 5 in terms of the pressure distribution in the domain, and the pressure forces acting on the fluid structure interface.

The problem has been solved with the four methods described above. Figure 6*a* shows that no significant differences are noticed between the four methods when *good* parameters ($r$ and $N$) are used. This figure represents the angle of rotation of the beam with respect to a horizontal line and compares the predictions with the analytical solution. All the methods give a deformation which is too small.

At the end of the beam, the computed angle of rotation is 1.1685°, while the analytical solution gives the value of 1.21045°, which leads to a relative error of 3.4%. A possible explanation is that the analytical problem neglects the reflection of the expansion waves on the shock, which weakens this shock. The shock is thus weaker in the numerical solution and the pressure behind it is less than the analytical one, providing a smaller deformation. Another explanation is that the shock wave discretization is not sufficiently accurate.

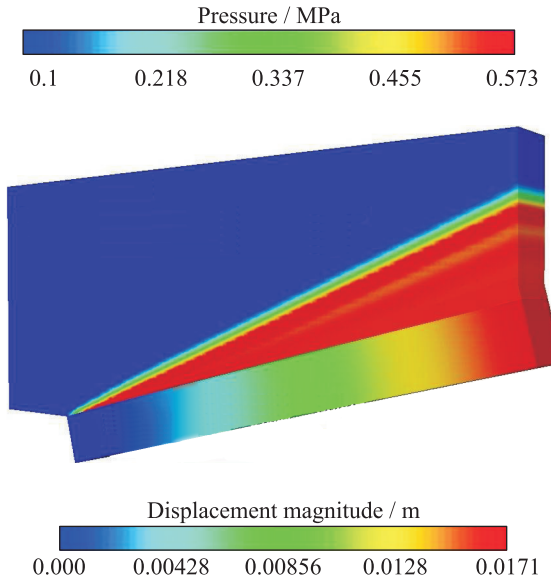


**Figure 5** Solution of the compression beam problem (pressure distribution). (Refer Vaassen *et al.*, p. 315.)

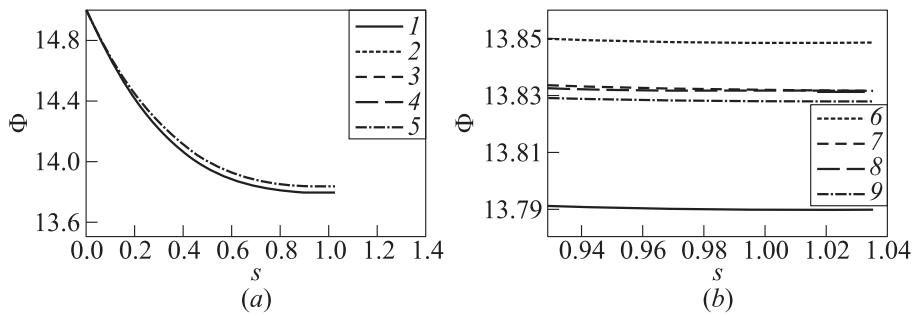


**Figure 6** Comparison between different transfer methods for the compression beam: *1* — analytical solution; *2* — control; *3* — RBF–TPS; *4* — RBF–EU; *5* — RBF–TBW; *6* — BW, $r = 10^{-2}$; *7* — BW, $r = 10^{0}$; *8* — BW, $r = 10^{1}$; and *9* — BW, $r = 10^{3}$

Figure 6*b* illustrates the same angle when different parameters $r$ have been chosen (here, for the RBF–BW method). Only the extremities of the curves are presented. A small value of $r$ provides less good solutions, as already mentioned previously.

# 5 SUBSONIC FLOW AROUND A WIND TURBINE BLADE

The second example considered in this work is a steady subsonic flow around a wind turbine blade. The fluid is viscous and a laminar flow is supposed. An inlet flow with a velocity equal to 30 m/s is imposed at the entry.

Figure 7 illustrates the flow around the blade in terms of three horizontal cuts and arrows that are proportional to the velocity magnitude. Two vortices appear behind the blade. To illustrate them, streamlines were plotted, starting from the extremity of the blade. These streamlines clearly draw spirals that go down the blade, and finally leave it to follow the mean flow. The blade itself is also gray-scaled with its displacement (the greater the displacement, the darker the blade).

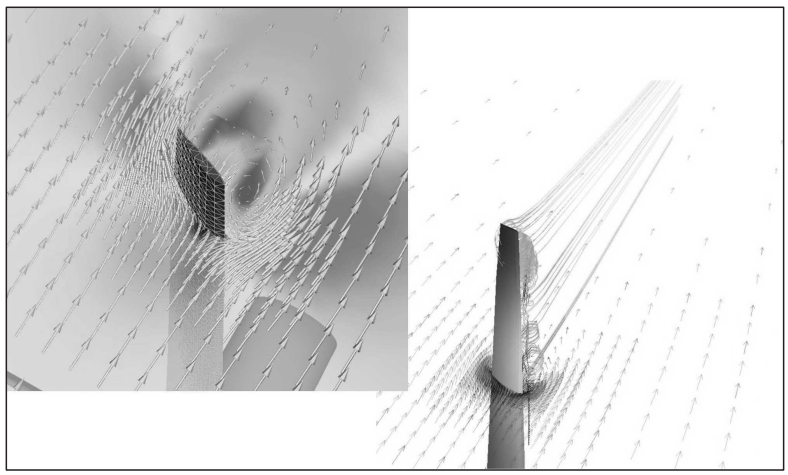

**Figure 7** Flow around a wind turbine blade

Figure 7 was plotted using the contact method. The same computation was made using other methods. In the RBF methods, the patches were composed of about 20 cells ($N = 20$) and the radius was equal to 5 (Table 1).

The predicted deformations were thus almost the same with the different methods. This is not surprising as the functions that are transferred do not present any discontinuities and are relatively well discretized on the structure mesh. The previous considerations had demonstrated that only discontinuities or badly discretized functions induced differences between the transfer methods that were envisaged in this work.

**Table 1** Deformation obtained with different transfer methods

| | Contact | RBF–TPS | RBF–EU | RBF–BW |
|---|---|---|---|---|
| Maximum deformation (m) | 0.013861 | 0.013823 | 0.013831 | 0.013818 |

## 6 CONCLUDING REMARKS

The aim of this paper was to present different transfer methods that can be used to transfer variables from one skin mesh to another (the two meshes being nonconform) for FSI problems using staggered solvers. After presenting the necessity to use such methods, two different methods have been introduced:

($i$) a method based on the contact between two meshes, and

($ii$) the RBF method.

These methods have been first analyzed on the transfer of a sharp imposed function from one coarse mesh to another which was more refined. The conclusions were that all the methods are equivalent for smooth functions, but that RBF-BW seems to be the best method if a smooth function badly discretized on the structure mesh has to be transferred. However, if the function presents discontinuities, RBF-EU is the only method that dissipates oscillations in the neighborhood of the discontinuity.

Examples were treated to compute real FSI problems. The accuracy of the solution was fairly good, even if the functions to transfer in these examples were too smooth to really distinguish the different methods.

In the future, an improved analysis of the space and time accuracy of the global solver is to be made, and more complicated test cases will be envisaged to continue the study of the transfer methods.

## REFERENCES

1. Hirsch, C. 1998. *Numerical computation of internal and external flows*. Vol. 1. John Wiley & Sons.
2. De Boer, A., H. Bijl, and A. van Zuijlen. 2005. Comparing different methods for the coupling of non-matching meshes in Fluid-Structure Interaction computations. *AIAA CFD Conference*. Toronto.
3. Stam, J. 1998. Exact evaluation of Catmull–Clark Subdivision Surfaces at arbitrary parameters values. In: *Computer Graphics Proceedings Annual Conference Series*. 395–404.

4. Ahrem, R., A. Beckert, and H. Wendland. 2001. A new multivariate interpolation method for large-scale spatial coupling problems in aeroelasticity. Technical Report. Institüt für Numerische und Angewandte Mathematik, Universität Göttingen, Germany.
5. Beckert, A., and H. Wendland. 2001. Multivariate interpolation for fluid–structure interaction problems using Radial Basis Functions. Aerospace Science Technologies. Elsevier Scientific and Medical Editions. S1270-9638 (00) 01087-7/FLA.