Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 82 (2015) 209 – 214

Energy Procedia

ATI 2015 - 70th Conference of the ATI Engineering Association

A local, un-structured, re-meshing technique capable of handling large body-motion in rotating machinery.

A. Bonfiglioli[a,1], R. Paciorri[b]

[a]*Scuola di Ingegneria, Università degli Studi della Basilicata, V.le dell'Ateneo Lucano 10, Potenza, 85100, Italy*
[b]*Dip.to Ingegneria meccanica e aerospaziale, Università di Roma "La Sapienza" , Via Eudossiana 18, Rome, 00184, Italy*

Abstract

In this paper we propose a simple, unstructured mesh generation technique that is capable of handling the large and complex blade motion that is encountered in certain rotating machines, such as vertical axis wind turbines or cycloidal propellers. The technique is characterised by localised re-meshing and interpolation, so as to keep the mesh generation cost and interpolation error as low as possible.

Peer-review under responsibility of the Scientific Committee of ATI 2015

Keywords: Rotating machines, Unstructured mesh generation, Arbitrary Lagrangian Eulerian

1. Introduction

In certain rotating machinery, such as the Voith Cycloidal Rudder (VCR), the blades undergo a fairly complex relative motion. When the hydrodynamic behaviour of these devices is numerically simulated by means of CFD, the shape and volume of the domain occupied by the fluid change with time and the large displacement of the wetted surface might even lead to topological changes in the computational domain. Under such circumstances, mesh deformation algorithms [1] might be inapplicable and overlapping grids [2] or global re-meshing techniques should be used instead. The time-dependent re-meshing of the fluid domain can however amount to a significant portion of the total computational cost. Moreover, the transfer of information between the grids generated at subsequent time levels or between overlapping meshes is likely to lead to conservation violation and loss of resolution.

We propose a mesh generation technique that is characterised by localised re-meshing and interpolation, so as to keep the mesh generation cost and interpolation error as low as possible.

In the proposed technique, a body-fitted, unstructured grid is attached to each of the moving blades and each blade, along with the grid that is attached to it, is allowed to move over a fixed, background triangulation that covers the entire computational domain. At each time-step, the cells of the background mesh which are overlap by the body-fitted grids are temporarily removed and local re-meshing is applied only in the neighbourhood of the outer boundary of the body-fitted meshes.

[1]Corresponding author: Tel. +39-0971-205203. *Email Address:* aldo.bonfiglioli@unibas.it.

Peer-review under responsibility of the Scientific Committee of ATI 2015
doi:10.1016/j.egypro.2015.12.022

Nomenclature

α	angle of attack	I	identity matrix of order $d+2$
Δt	time-step length	k	reduced frequency
$\mathbf{w}$	nodal grid velocity	K_i^e	inflow parameter
$\mathbf{x}$	array of the Cartesian coordinates of the grid-points	M	Mach number
ω	angular frequency	p	static pressure
Φ	Inviscid flux balance	T	static temperature
Φ_i^e	signals sent to vertex i of cell e	T^e	triangle e
ρ	density	U	conserved variables
c	airfoil's chord	u	horizontal velocity component
C^i	median dual cell	U_∞	freestream velocity magnitude
d	dimension of the space	v	vertical velocity component
H	total enthalpy	Z	parameter vector = $\sqrt{\rho}\,(1, H, u, v)^T$

As far as the discretization of the governing PDEs is concerned, an Arbitrary Lagrangian Eulerian (ALE) scheme is required in order to account for the non-zero grid velocities that arise within the body-fitted meshes and along their interface with the background, stationary triangulation.

2. The numerical method

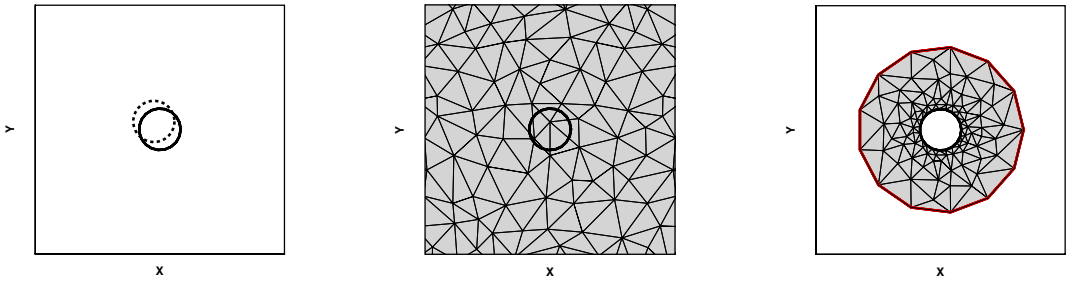
The numerical method we propose consists in the loose coupling between an unstructured grid generator that is used generate computational meshes around the moving bodies and within the fluid domain and an unstructured CFD solver that is used to discretise the governing PDEs using the boundary-conforming and non-overlapping meshes created by the mesh generator.

The two codes are loosely coupled in the sense that the grid generator invokes the CFD solver as a black box. This has obvious consequences in terms of algorithmic simplicity, since it allows to re-use any existing gas-dynamic code, as long as its discretization is vertex centred.

In the next two sections we will describe the unstructured solver and the mesh generator algorithm.

2.1. The `eulfs` *unstructured-grid solver*

The `eulfs` code is an in-house, unstructured CFD solver that has been developed over the last fifteen years: see [3] for a detailed description of its basic features and [4] for more recent developments. It relies on Fluctuation Splitting (FS), or Residual Distribution [5, 6, 7] schemes for the spatial discretisation. In the FS approach the dependent variables are stored at the vertices of the computational mesh which is made up of triangles in the 2D space, and tetrahedra in 3D and are assumed to vary linearly and continuously in space. The inviscid flux balance Φ^e (also referred to as the cell residual or cell fluctuation) is evaluated over each triangular/tetrahedral element e by means of a conservative linearisation [8] based on the parameter vector Z, and scattered to the element vertices using signals Φ_i^e. Within a cell e, the signals have to sum up to the net flux for conservation: $\sum_{i\in e} \Phi_i^e = \Phi^e$. The nodal residual is then assembled by collecting fractions Φ_i^e of the net fluxes Φ^e associated with all the elements by which the node i is surrounded. The various FS schemes proposed in the literature differ by the way cell residuals are split into signals. In this paper,



(a) The domain and the moving body. (b) The d-mesh. (c) The b-mesh.

Fig. 1. Domain, moving body and meshes.

we use the FS version [9, 10] of the popular Lax-Wendroff (LW) scheme, because this is the simplest two-time-levels, explicit scheme that is second-order-accurate both in space and time. Using the LW scheme, the signals sent to vertex i of cell e are:

$$\Phi_i^e = \left[\frac{1}{d+1} I + \frac{1}{2} \frac{\Delta t}{|T^e|} K_i^e \right] \Phi^e \tag{1}$$

where $|T^e|$ denotes the area of the triangle and K_i^e, the so-called inflow parameter, is a matrix that depends upon the cell-averaged Jacobian matrix of the inviscid fluxes and the normal to the edge opposite vertex i, see [3] for details. When the grid moves and/or deforms, matrix K_i^e also depends upon the cell-averaged grid velocity.

The following explicit update formula is obtained for the Arbitrary Lagrangian Eulerian (ALE) version of the LW scheme:

$$|C_i|^{n+1} U_i^{n+1} \quad = \quad |C_i|^n U_i^n + \Delta t \sum_{e \ni i} \Phi_i^e \tag{2}$$

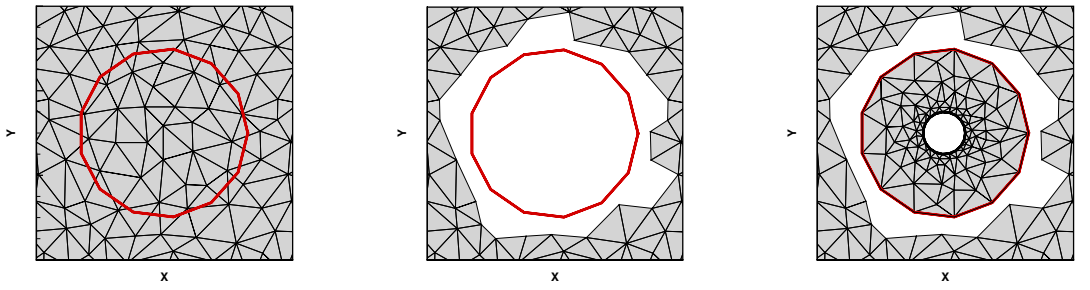
where C_i the median dual cell centred about grid-point i and U is the vector of the conserved variables which can be computed from parameter vector using the following identity:

$$U = \frac{1}{2} \left(\frac{\partial U}{\partial Z} \right) Z.$$

The Geometric Conservation Law is satisfied as long as the signals in Eq. (1) are evaluated on the mesh at time level $n + \frac{1}{2}$ and the telescoping property of the fluxes is guaranteed using the approach described in [11].

2.2. The mesh generator algorithm

In order to illustrate the algorithmic features of the proposed mesh generation algorithm, we consider the flow produced by a body, such as the circular cylinder shown in Fig. 1(a), that moves inside a fluid. At time t, the body is located in an arbitrary position, shown using the solid line in Fig. 1(a); over the time interval Δt, the body moves to a new location, shown by the dashed line in Fig. 1(a). As shown in Fig. 1(b), the flow domain is discretised using a triangular mesh that does not take into account the presence of the body. This mesh, which is fixed in space, will be hereafter denoted as the "d-mesh". Another mesh, that will be called the "b-mesh", is generated around the body, as shown in Fig. 1(c). This body-fitted mesh, which is entirely independent of the d-mesh, moves rigidly with the body. We assume that at time $t = n\,\Delta t$ the solution vector Z^n is known at all points of both the d-mesh (except for those nodes that are hidden by the body) and the b-mesh. The process that leads from the known solution Z^n at time t to the updated solution Z^{n+1} at time $t + \Delta t$ can be split into the three steps that will be described in detail in the following sub-sections.



(a) d-mesh with the boundary of b-mesh at time t.

(b) d-mesh after the cell removal process.

(c) Insertion of the b-mesh.

Fig. 2. Cell removal, followed by the insertion of the b-mesh.

2.2.1. Generation of the computational mesh

At time t, the b-mesh covers a certain region of the flow domain; this is shown in Fig. 2(a), where the outer boundary of the b-mesh has been highlighted in red. All cells of the d-mesh that have one of their vertices either: *i*) falling within the outer boundary of the b-mesh or *ii*) outside of it, but closer to it than a preset distance, are removed. The reason for removing the grid-points that meet the criterion *ii*) is that these nodes, if not removed, might be overcome by the b-mesh as it moves from its location at time t to the one at time $t + \Delta t$. All the nodes of the d-mesh that have been removed will be hereafter called: "phantom" nodes. Figure 2(b) shows the appearance of the "d-mesh" following the removal of the phantom nodes.

At this stage, the b-mesh can be inserted inside the hole dug into the d-mesh by the cell removal step; the two meshes are however still disjoint, as shown in Fig. 2(c). The two meshes are then mutually connected by means of a local constrained Delaunay triangulation (CDT) which does not introduce any additional grid-point. This CDT is constrained in the sense that the edges that belong to the boundary of the hole and those that make up the outer boundary of the b-mesh are forced to be part of the final triangulation, see Fig. 3(a). The new "computational" mesh that has been generated by the local CDT will be used to advance the solution from time t to time $t + \Delta t$. The solution vector Z^n that is stored in the grid-points of both the d- and b- meshes can be easily transferred to the vertices of the computational mesh because every node of the computational mesh belongs to either the d-mesh or the b-mesh.

2.2.2. Evolution of the solution on the computational mesh

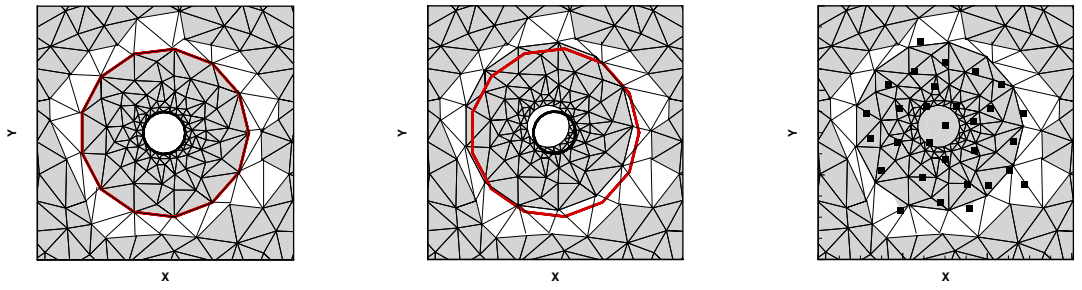
The solution Z^n and the computational mesh at time t, shown in Fig. 3(a), are provided as input to the `eulfs` code in order to advance the solution (and the computational mesh) from time t to time $t + \Delta t$. Since some of the nodes of the computational mesh, more precisely those that also belong to the b-mesh, move with the body, a nodal grid velocity must also be supplied to the CFD code. More precisely, for each node i of the computational mesh, the `eulfs` code needs the nodal grid velocity $\mathbf{w}_i$ computed at time $t + \Delta t/2$ using the finite difference formula:

$$\mathbf{w}_i^{n+1/2} = \frac{(\mathbf{x}_i^{n+1} - \mathbf{x}_i^n)}{\Delta t}.$$

The motion of the grid-points of the computational mesh causes a rigid motion of the triangles that also belong to the b-mesh and a deformation of the cells that have been generated by the local CDT, see Fig. 3(b). The output of the CFD code is the updated solution Z^{n+1} within all grid-points of the computational mesh at time $t + \Delta t$, shown in Fig. 3(b).

2.2.3. Projection of the solution available on the computational mesh on the d- and b-meshes

The computed solution at time $t+\Delta t$ has to be transferred from the computational mesh back to the d- and b-meshes. Concerning the b-mesh, the projection is trivial, because all grid-points of the b-mesh also belong to the computational mesh. By contrast, the projection of the solution Z^{n+1} on the d-mesh requires caution.



(a) Computational mesh at time t. (b) Computational mesh at time $t + \Delta t$. (c) Interpolation of phantom points.

Fig. 3.

This is because not all the grid-points belonging to the d-mesh also belong to the computational mesh: this is the case of the phantom nodes that have been identified and removed in step 2.2.1. Nonetheless the nodal state within the phantom nodes must also be updated to time $t + \Delta t$, since the set of the phantom nodes could change between subsequent time intervals. This is because, due to the motion of the body, some of the points of the d-mesh that were close the boundary of the b-mesh and therefore declared phantom when integrating from t to $t + \Delta t$, might re-appear during the subsequent time interval $[t + \Delta t, t + 2\Delta t]$. The interpolation of the phantom nodes of the d-mesh, which is made using the nodal values available on the computational mesh, allows to compute the state of all phantom nodes, except for those that are hidden by the body, see Fig. 3(c).

At this stage, the numerical solution has correctly been updated at time level $t + \Delta t$ on both the b- and d-meshes, so that the next time level can be computed re-starting from the first step 2.2.1 of the algorithm.

3. Numerical example

The verification of the algorithm described in Sect. 2 has been carried out by looking at the temporal evolution of the aerodynamic coefficients of a NACA 0012 airfoil that is pitching about its leading edge with the following law:

$$\alpha(t) = 0.016^\circ + 2.51^\circ \sin(\omega t). \tag{3}$$

The angular frequency ω that appears in Eq. (3) can be expressed as a function of the dimension-less reduced frequency:

$$k = \frac{\omega c}{2 U_\infty} = 0.0814.$$

Free-stream conditions are: $T_\infty = 273$ K, $p_\infty = 101325$ Pa, $M_\infty = 0.5$.

The choice of a law of body motion, such as Eq. (3), that gives rise to a moderate displacement of the body is motivated by the fact that, by doing so, it is also possible to perform the same calculation using a grid with fixed connectivity in which only the subset of nodes which is closest to the body undergoes a rigid motion (according to Eq. (3)), while the remaining grid-points remain fixed in space. This same approach would easily lead to a folding grid, if larger displacements were used. It is this calculation, which has been performed using the same `eulfs` code, that as been used as the reference calculation.

Figure 4 compares the temporal evolution of the lift coefficient computed using the newly developed mesh generation approach (MUMs) and the reference calculation; it is evident that the two approaches give identical results.

4. Conclusions

In this paper we a propose a simple mesh generation algorithm that can easily handle the large blade motion encountered in certain rotating machines. The proposed technique is characterised by localised re-

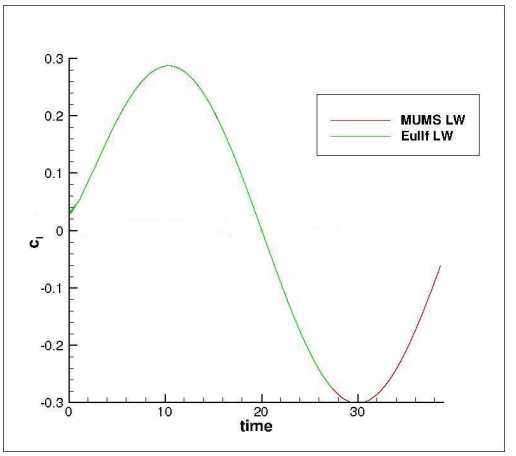
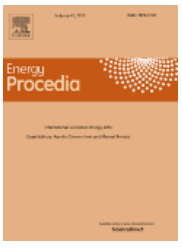


Fig. 4. Pitching airfoil: temporal evolution of the lift coefficient

meshing and interpolation, so as to keep the mesh generation cost and interpolation error as low as possible. The technique is also versatile, in the sense that it can be coupled with virtually any vertex-centred CFD code that is capable of solving the governing PDEs, written using an ALE formulation.

References

1. Jasak, H., Tuković, v.. Automatic mesh motion for the unstructured finite volume method. *Transactions of Famena* 2006;30(2):1–20.
2. Hadžić, I., Hennig, J., Perić, M., Xing-Kaeding, Y.. Computation of flow-induced motion of floating bodies. *Applied Mathematical Modelling* 2005;29(12):1196 – 1210. doi:http://dx.doi.org/10.1016/j.apm.2005.02.014.
3. Bonfiglioli, A.. Fluctuation splitting schemes for the compressible and incompressible Euler and Navier-Stokes equations. *International Journal of Computational Fluid Dynamics* 2000;14:21–39.
4. Bonfiglioli, A., Paciorri, R.. A mass-matrix formulation of unsteady fluctuation splitting schemes consistent with Roe's parameter vector. *International Journal of Computational Fluid Dynamics* 2013;27(4-5):210–227. doi:10.1080/10618562.2013.813491.
5. Deconinck, H., Paillère, H., Struijs, R., Roe, P.. Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Computational Mechanics* 1993;11(5/6):323–340.
6. van der Weide, E., Deconinck, H., Issman, E., Degrez, G.. A parallel, implicit, multi-dimensional upwind, residual distribution method for the Navier-Stokes equations on unstructured grids. *Computational Mechanics* 1999;23:199–208.
7. Abgrall, R.. Residual distribution schemes: Current status and future trends. *Computers and Fluids* 2006;35(7):641 – 669. doi:10.1016/j.compfluid.2005.01.007.
8. Deconinck, H., Roe, P., Struijs, R.. A Multi-dimensional Generalization of Roe's Flux Difference Splitter for the Euler Equations. *Computers and Fluids* 1993;22(2/3):215–222.
9. Struijs, R.. A multi-dimensional upwind discretization method for the Euler equations on unstructured grids. Ph.D. thesis; Technische Universiteit Delft, The Nederlands; 1994.
10. Rossiello, G., De Palma, P., Pascazio, G., Napolitano, M.. Second-order-accurate explicit fluctuation splitting schemes for unsteady problems. *Computers & Fluids* 2009;38(7):1384 – 1393. doi:DOI: 10.1016/j.compfluid.2008.01.021.
11. Michler, C., De Sterck, H., Deconinck, H.. An arbitrary lagrangian eulerian formulation for residual distribution schemes on moving grids. *Computers & Fluids* 2003;32(1):59 – 71. doi:10.1016/S0045-7930(01)00095-0.



Aldo Bonfiglioli graduated in 1992 summa cum laude in Mechanical Engineering from the TUBari, obtained the Diploma Course from the VKI and the PhD from the TUBari. Since 2007 he is Associate Professor of Fluid Machines at the University of Basilicata. His current research interests include the development of discretization schemes on unstructured meshes.