



City Research Online

City St George's, University of London

Citation: Bianchi, G., Kovacevic, A. & Cipollone, R. (2017). Deforming grid generation for numerical simulations of fluid dynamics in sliding vane rotary machines. *Advances in Engineering Software*, 112, pp. 180-191. doi: 10.1016/j.advengsoft.2017.05.010

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://city-ref-test.eprints-hosting.org/id/eprint/17531/>

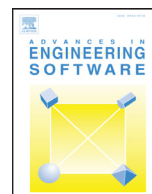
Link to published version: <https://doi.org/10.1016/j.advengsoft.2017.05.010>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).



Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

Deforming grid generation for numerical simulations of fluid dynamics in sliding vane rotary machines

Giuseppe Bianchi^{a,b,*}, Sham Rane^a, Ahmed Kovacevic^a, Roberto Cipollone^c^a Centre for Compressor Technology, City, University of London, Northampton Square, London, EC1V 0HB, United Kingdom^b Centre for Sustainable Energy Use in Food Chains, Brunel University London, Kingston Lane, Uxbridge, UB8 3PH, United Kingdom^c Department of Industrial and Information Engineering and Economics, University of L'Aquila, via G. Gronchi 18, L'Aquila, 67100, Italy

ARTICLE INFO

Article history:

Received 9 November 2016

Revised 1 May 2017

Accepted 14 May 2017

Available online xxx

Keywords:

Computational fluid dynamics

Algebraic grid generation

Deforming grid

Sliding vane compressor

Positive displacement compressor

ABSTRACT

The limiting factor for the employment of advanced 3D CFD tools in the analysis and design of rotary vane machines is the unavailability of methods for generation of a computational grid suitable for fast and reliable numerical analysis. The paper addresses this issue through an analytical grid generation based on the user defined nodal displacement which discretizes the moving and deforming fluid domain of the sliding vane machine and ensures conservation of intrinsic quantities by maintaining the cell connectivity and structure. Mesh boundaries are defined as parametric curves generated using trigonometrical modelling of the axial cross section of the machine while the distribution of computational nodes is performed using algebraic algorithms with transfinite interpolation, post orthogonalisation and smoothing. Algebraic control functions are introduced for distribution of nodes on the rotor and casing boundaries in order to achieve good grid quality in terms of cell size and expansion. For testing of generated grids, single phase simulations of an industrial air rotary vane compressor are solved by use of commercial CFD solvers FLUENT and CFX. This paper presents implementation of the mesh motion algorithm, stability and robustness experienced with the solvers when working with highly deforming grids and the obtained flow results.

© 2017 Published by Elsevier Ltd.

1. Introduction

Positive displacement machines perform thermodynamic transformations through volume variations of their working chambers. Depending on the technology considered, the reference closed volume capacity for fluid transformations can be the one enclosed between piston and cylinder, a pair of screw rotors or fixed and orbiting scrolls. In sliding vane devices, working chambers are instead defined by a given number of blades that slide along the stator surface while they rotate driven by the rotor. The accurate representation of these complex, deforming and moving computational domains has been the main constraint that limited the application of advanced numerical simulation tools to support the design of positive displacement machinery [1].

Among the available approaches for grid deformation, the most common ones, which can be nowadays found in commercial solvers using Finite Volume Methods (FVM) for solving the partial differential equations are diffusion equation mesh smoothing,

spring analogy mesh smoothing, tetrahedral re-meshing, hexahedral layering, key-frame re-meshing and User Defined Nodal Displacement (UDND). Table 1 presents a comparison of the most common approaches for grid deformation in FVM used by commercial solvers like ANSYS CFX and FLUENT, and STAR-CCM+.

In the diffusion smoothing, at the beginning of each time step, the specified displacement of the boundary nodes is propagated to the interior nodes by solving a diffusion equation. Spring analogy uses a balance of strain forces to move interior nodes. In tetrahedral re-meshing, the cells that deform too much such that their quality degrades beyond limit are selected along with some adjacent cells and the entire local volume is re-meshed inside the solver. On similar lines, hexahedral cell layers are added or removed in the layering algorithm. In key-frame re-meshing, upon the discovery of irregular cells generated after a preliminary diffusion process, the domain selected for re-meshing is replaced by a pre-generated mesh at that boundary position. During re-meshing operations, the boundary geometry is retained but with a change in the number of cells and their connectivity. At the same time, the flow solution is interpolated between two time steps and this results with inaccuracies. On the other hand, in user defined nodal displacement, customised grid generators can be used to create a set of grids representing nodal locations for each time step exter-

* Corresponding author.

E-mail address: giuseppe.bianchi@brunel.ac.uk (G. Bianchi).

Nomenclature

a_{st}	Stator x semi axis
b_{st}	Stator y semi axis
e	eccentricity
t_{bl}	Blade thickness
C_{gap}	Clearance gap
R_{rot}	Rotor radius
R_{tip}	Blade tip radius
τ, δ	Auxiliary geometrical parameters
S	length along the blade profile
R	length of the rotor profile
S'	length of the stator profile
I	Number of nodes in the circumferential direction
J	Number of nodes in the radial direction
σ	Stretching parameter
$r_{ij}(x, y)$	Nodal radius vector of node with index i, j and point coordinates x, y
$\mathbf{P}$	Projector operator
$\oplus$	Boolean sum
x, y	Cartesian coordinates
ρ, ϑ	Polar coordinates
η, ξ	Transformed coordinates in computational space
FN	Fluent Node
TN ⁰	Mesh node at initial time step

nally, prior to the numerical flow solution in the calculation domain. Therefore, such approach guarantees conservation of space and equations without any artificial sources [2].

User defined nodal displacement methodology was successfully applied to generate grids for screw machines and resulted in the development of two different strategies for grid generation. An algebraic approach was initiated by the research group at the City University London while a hybrid differential approach was later considered at the University of Ghent. In the first one [3], an analytical conjugate of the rotor profiles called “the rack” is used to

split the working domain into two domains with O-grid topologies each belonging to one of the rotors. In each of these domains an algebraic grid is generated using transfinite interpolation, post smoothing and orthogonalisation. In the latter one [4], the twin rotor domain is split into two O-grid topologies using the solution of Laplace equation on a starting triangular grid in the rotor cross section. This is followed by distribution of grid points on isopotential curves of the Laplace solution. Both of these grid generation methods result with numerical mesh suitable for three-dimensional single and multi-phase simulations of screw compressors and expanders. The algebraic grid generation approach proved more suitable for extension of the method to variable lead and variable profile rotors, while the hybrid differential approach resulted in better grid smoothing and adaptability to variety of rotor profiles [5].

In addition to the aforementioned academic research groups, Hesse et al. developed a UDND methodology that has been further implemented in the commercial customised mesh generation tool TwinMesh. In this approach, the fluid domain boundaries of a positive displacement machine are firstly imported from external files. Afterwards, the meshing software generates 2D grids for each rotation angle of the rotor including a refined boundary layer resolution and layers towards rotor and housing walls. The meshes are smoothed with an explicit and iterative method to reach homogeneous node distribution and orthogonality. The set of three dimensional grids is generated from the 2D meshes and eventually supplied to the CFD solver. This methodology has been successfully applied to twin-screw [6] and scroll geometries [7] and, in general, it is suitable for positive displacement machines. However, to date, only single phase simulations with ideal gas fluid modelling were performed and exclusively available for calculation with one CFD solver ANSYS CFX.

Unlike screw machines, the literature studies on the modelling of sliding vane devices mostly considered zero dimensional formulations, also known as chamber models. According to these approaches, spatial variation of quantities inside cavities (e.g. vane compressor cells) is collapsed in a lumped parameter approxima-

Table 1
Comparison of common grid deformation techniques [1,2].

	Pre-processing effort	Magnitude of boundary deformation	Space conservation law and solution accuracy	Intermediate Interpolation of data
Diffusion smoothing	Low	Low – cannot exceed 50% of cell size per step	Valid, high accuracy	No
Spring smoothing	Low	Low – cannot exceed 50% of cell size per step	Valid, high accuracy	No
Tetrahedral remeshing	Low	Low – cannot exceed 75–90% of cell size per step	Invalid, inaccurate	Yes
Hexahedral layering	Mid	Mid – can absorb 1–2 cell layers per step	Invalid, inaccurate	Yes
Key-frame remeshing	High	Low – cannot exceed 50% of cell size per step	Invalid, inaccurate	Yes
UDND	Mid	High – external control	Valid, high accuracy	No
	Solver effort	Grid deformation time	Quality control and reliability of cells after deformation	Suitability for vane geometry
Diffusion smoothing	Additional diffusion equation (iterative)	Low	Low	No
Spring smoothing	Additional smoothing solution (iterative)	Low	Low	No
Tetrahedral remeshing	Remeshing after every step	High	Low	Yes
Hexahedral layering	Remeshing after every step	High	Medium	No
Key-frame remeshing	Additional smoothing solution (iterative) plus mesh replacement and solution interpolation	Low	Low, replaced mesh can be of improved quality	No
UDND	Node position update – low solver effort	Low	High – external control	Yes

tion whereas equation of state (if the working fluid is a gas) or compressibility relationships (if the working fluid is a liquid) are considered in addition to the conservation equations to solve the numerical problem [8–13]. On the other hand, mass and energy exchange between cells with suction and discharge pipes can be modelled using a one-dimensional formulation of the conservation equations such that pressure pulsations at the intake and exhaust ducts can be respectively appreciated.

Research works on CFD analyses in sliding vane machines are highly limited. Most of them relied on commercial meshing tools to discretize the fluid domain enclosed in the working chambers. In particular, Montenegro et al. [14], investigated a sliding vane expander for energy recovery applications. The machine geometry had an elliptic stator while the CFD study was performed using the OpenFOAM solver. Due to the lack of an automatic grid generator, the vane mesh was created through the built-in Cartesian mesh tool in OpenFOAM. To speed up the computations, the momentum equation was modified by the introduction of a Darcy-Forchheimer type source term, which takes into account both the viscous and inertial effects occurring in a tiny gap; this resistance source term was locally applied only on the cells located at the flat shaped vane tip. Kolasiński and Błasiak [15] performed experimental and numerical studies on a sliding vane expander with circular stator and radial inlet and outlet ports. In their modelling activity, to avoid negative volumes in the mesh, the expander geometry was modified reducing vane thickness and increasing the vane tip clearance but in turn this latter adjustment yielded to a modified eccentricity. Movement and deformation of the vane grid were accomplished using an embedded tool of ANSYS suite that relocated the internal vane nodes according to a displacement diffusion model. Zhang and Xu [16] investigated the role of vane tip radius on local cavitation phenomena in a vane type oil pump. In their study, the static grids were generated with the software ICFM-CFD while the rotating and deforming ones resulted from a customised grid generator that, using an algebraic method, was able to control the node coordinates. The set of grids was eventually coupled with the Star-CD solver. As concerns the turbulence models, $k-\varepsilon$ was used in the free stream regions while $k-\omega$ SST in the regions of adverse pressure gradients and separating flow. To account for cavitation phenomena, Rayleigh model and Volume Of Fluid multiphase method were eventually considered [16]. Nonetheless, because of this grid deformation strategy, to avoid generation of artificial mass sources during the numerical solution, the space conservation law proposed by Demirdzic and Peric [17] had to be simultaneously satisfied with the other conservation equations.

In order to overcome all the issues encountered in the reported research, a general grid generation methodology for sliding vane machines was developed and tested in the current study. The algebraic grid generation algorithm used for screw machines [3,18], developed at City University and implemented in the software SCORG (Screw Compressor Rotor Grid Generator), is employed to distribute nodes inside the fluid domain of the sliding vane devices, whose grid boundaries were retrieved through an analytical model. The numerical mesh is used with two commercial 3D CFD solvers to produce the performance prediction of a single phase compressors. These two solvers are both finite volume solvers but are based on different solution strategies; FLUENT uses cell centred numeric while CFX is based on the vertex centred numeric. Main physical quantities obtained by two solvers such as pressures and velocities are eventually compared in cross sections as well as in the clearance gap between stator and blade tip where most of the leakage occurs.

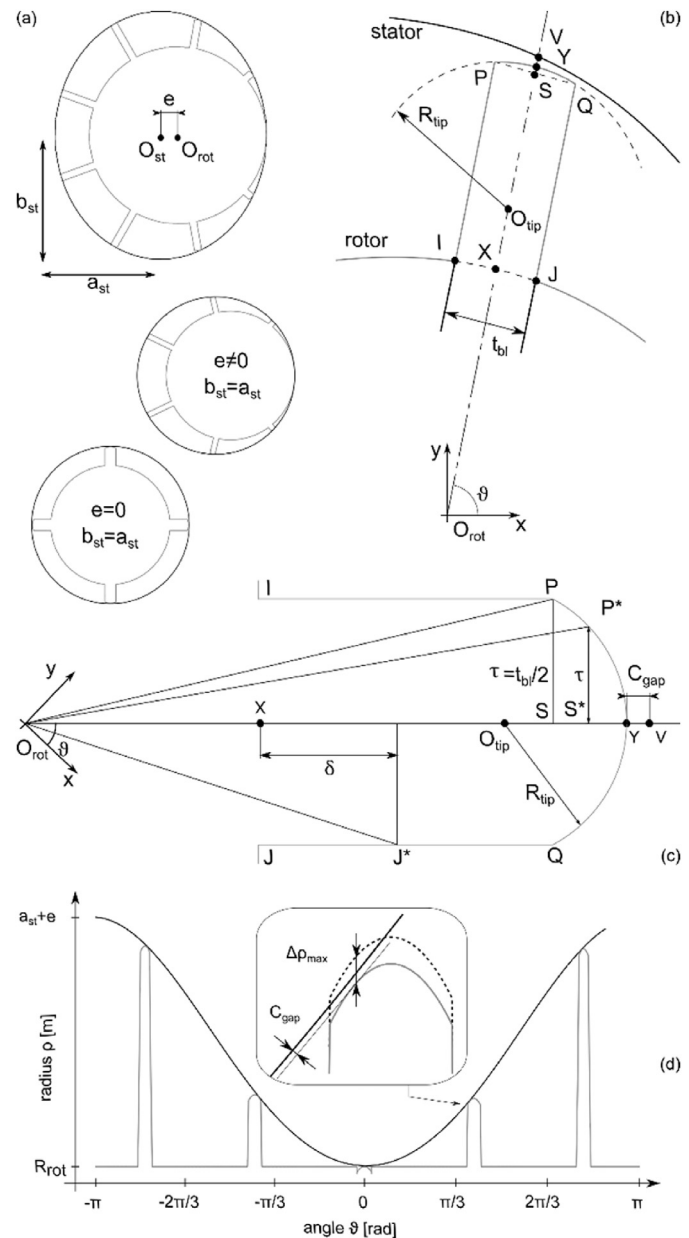


Fig. 1. Boundary generation procedure for sliding vane machines: input quantities for different machine geometries (a), overall calculation scheme (b) with detail on the blade tip modelling (c), numerical adjustment to ensure the minimum tip clearance gap (d).

2. Boundary generation

A sliding vane machine is a positive displacement device essentially composed of a fixed cylinder, named stator, and a rotating cylinder, called rotor, that rotates inside the stator and contains multiple slots in which blades called vanes slide and generate working chambers. If stator and rotor are eccentrically mounted, the chamber volume will change with rotation of the rotor. Hence, depending on the angular position of the intake and exhaust ports, the fluid trapped in the chamber will compress or expand.

In geometrical terms, the two cylinders are mounted with a given eccentricity and can be tangent along a generatrix, as shown in Fig. 1a. Moreover, while the stator may even have an elliptical cross section, the rotor is always a circular cylinder to prevent additional vibration phenomena during operation. Since the

Table 2

Trajectories of boundary generation key points displayed in Fig. 1b expressed in polar coordinates with respect to rotor origin: equations “a” refer to the radius ρ while “b” to the angle ϑ .

X	Rotor	R_{rot}	(1.a)
		ϑ	(1.b)
V	Stator	$\frac{-b_{st}(e - b_{st} \cos \vartheta + a_{st} \sqrt{(a_{st} \sin \vartheta)^2 + (b_{st} \cos \vartheta)^2 - (e \sin \vartheta)^2})}{(a_{st} \sin \vartheta)^2 + (b_{st} \cos \vartheta)^2}$	(2.a)
		ϑ	(2.b)
Y	Blade tip	$\rho_V - C_{gap}$	(3.a)
		ϑ	(3.b)
O _{tip}	Centre of circular tip profile	$\rho_V - R_{tip}$	(4.a)
		ϑ	(4.b)
S	Middle point at flat tip	$\rho_{Otip} + \sqrt{R_{tip}^2 - \tau^2} \quad 0 < \tau \leq t_{bl}/2$	(5.a)
		ϑ	(5.b)
P, Q	Blade tip profile (+ for P, - for Q)	$\sqrt{\rho_S^2 + \tau^2} \quad 0 < \tau \leq t_{bl}/2$	(6.a)
		$\vartheta \pm \tan^{-1}(\tau/\rho_S) \quad 0 < \tau \leq t_{bl}/2$	(6.b)
I, J	Blade walls (+ for I, - for J)	$\sqrt{(R_{rot} + \delta)^2 + t_{bl}^2/4} \quad 0 \leq \delta < \rho_S - R_{rot}$	(7.a)
		$\vartheta \pm \tan^{-1}(t_{bl}/(2(R_{rot} + \delta)))$	(7.b)

core of a sliding vane machine has a constant cross section profile with respect to its rotation axis, the boundaries of the fluid volume enclosed in the core were parameterized using an analytical bi-dimensional (2D) modelling approach that tracks the trajectories of stator, rotor and blade profiles outside the rotor slots. The motion of these components was modelled considering the trajectories of the points displayed in Fig. 1b and using analytical correlations with reference to the rotor centre as origin of the coordinate system.

A major assumption in the 2D modelling of the sliding vane machine and, in turn, in the resulting rotor grid was to consider a constant minimum clearance between the blade tip and the stator along a full vane revolution. The minimum tip clearance gap is shown in Fig. 1b and c as C_{gap} . This is a trade-off between the ideal geometrical tangency and real operating conditions, whereas the blade tip slides along a fluid layer that prevents a dry contact with the stator. It allows the use of a single grid block and avoids the calculation of the instantaneous minimum liquid film thickness between blade tip and stator which would have required either the usage of simplified hydrodynamic approaches [12,13] or the complex fluid-structure computation. In real operation, when a sliding machine runs as a compressor, the fluid layer is represented by the oil injected inside the vanes that, thanks to centrifugal forces, reaches the stator surface. On the other hand, when the machine acts as an expander, the fluid layer can result both by a partial condensation of the working fluid (if it was introduced as a vapour close to saturation state) or by a small oil injection inside the gas. The fluid layer in any case insures an effective sealing between adjacent cells. Additionally, the leakage losses through the clearance gaps between the rotor slots and blades as well as the leakage across the end wall plates were neglected.

In order to reconstruct the boundary of the rotor mesh, i.e. the computational grid related to the fluid domain enclosed between stator, rotor and blades, geometrical features of the machines need to be specified as input data. These dimensions include: rotor radius, stator semi-axes (or radius if circular), eccentricity, blade number, thickness and tip profile radius. After the grid generation procedure, the axial length of the stator is additionally required to convert the 2D mesh in a 3D one. With reference to Fig. 1b and c, at a given crank angle ϑ , polar coordinates of boundary mesh key points can be calculated with reference to Equations 1–7 grouped in Table 2.

In particular, the shape of the blade tip profile is fixed from the radius of curvature R_{tip} and the blade thickness t_{bl} . The left and right sides of the profile can be respectively reconstructed using P-type and Q-type points, which are defined through Eqs. 5 and 6. Depending on the number of points N chosen to discretise the

profile, the auxiliary thickness τ varies from the half of the blade thickness to zero. In the first limiting case, P and Q are located at the edges of the blade as in Fig. 1b. On the other hand, when τ is equal to zero points P, Q, S would collapse in point Y. This configuration refers to a blade without thickness and therefore it is discarded in the boundary generation. Hence, the blade tip profile is defined by $2N + 1$ points (N P-type, N Q-type and Y). Similarly, the blade walls can be reconstructed with reference to points I and J through Eq. (7) and the auxiliary length δ .

Further to the analytical boundary generation, a numerical adjustment is required to account for the finite blade thickness such that minimum gap between blade tip and stator is occurring only once and it is equal to the tip clearance gap value C_{gap} provided as input of the calculation. This correction implies an outwards translation of the blade along its slot (i.e. radial direction) if the minimum gap between blade tip and stator is greater than the input tip clearance gap. On the other hand, as it is displayed in Fig. 1d, an inwards translation occurs if the minimum gap at the tip is lower than the input value or if the blade intersects with the stator. For a given blade tip profile, the magnitude of the radial offset in the point of maximum deviance can be calculated according to Eq. (8), where ρ_{PQ} can refer to the radius of either a P-type or to a Q-type point depending on the side of the tip profile where the maximum deviance occurs. For instance, with reference to Fig. 1d, one can see that for positive angles the radial offset is calculated using P-type points while for negative angles Q-type points have to be used. This offset is then applied to the whole tip profile to preserve the original shape.

$$|\Delta \rho_{max}| = |\rho_V - \rho_{PQ}| - C_{gap} \quad (8)$$

The resulting 2D boundary of the fluid volume enclosed in the core of a sliding vane machine is eventually composed of the stator profile and of the rotor profile with all the blades outside the rotor slots. Fig. 1a additionally shows the boundary generation task for multiple geometries: a sliding vane machine (compressor or expander) with circular stator, a machine with elliptic stator and a device with zero eccentricity that might refer to a sliding vane pump. In all the images, the inner boundary is displayed in grey while the outer one in black.

3. Boundary discretisation

Prior to their distribution inside the computational domain, nodes are placed along the grid boundaries to produce an “O” topology. An “O” structure for the current rotor mesh was purposely selected because it avoids any inaccuracies that can be introduced due to a non-matching mesh connection between the core and the leakage regions. Furthermore, in order to control the

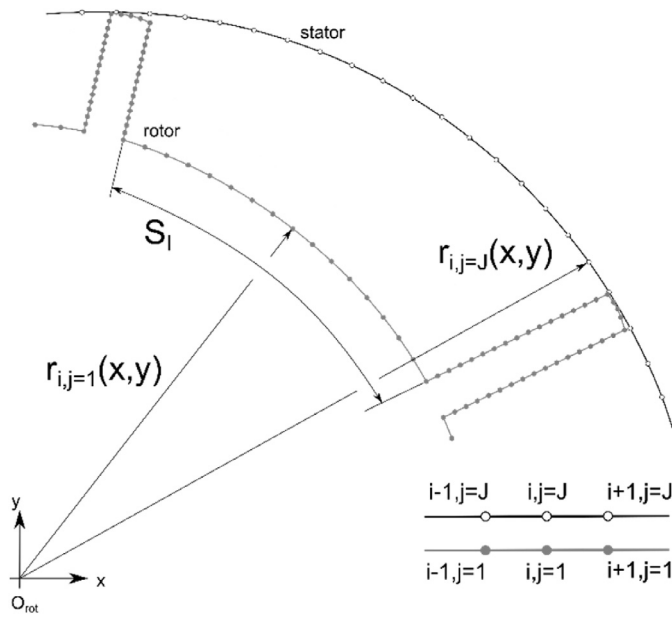


Fig. 2. Reference scheme for nodal radius vector.

sudden transition from the leakage gaps to the core region, stretching functions were introduced in the discretisation algorithm to gradually flare the radial mesh lines from leakage gap into the core.

The index notation for the boundary discretisation procedure refers to the scheme presented in Fig. 2. Let $r_{i,j}(x, y)$ represent the points distributed on the boundaries with respect to the physical coordinate system. Here r is the nodal radius vector, index i is along the circumferential direction of the boundary profile (composed of the vane side walls and vane tip together with the rotor) and index j is along the mesh line in radial direction. A value of $i=1$ indicates the first node on the boundary and $i=I$ is the last node. I is the total number of nodes that are used to discretise a particular boundary curve. Since the grid structure is an 'O' grid, in the final mesh nodes with index $i=I$ are merged with index $i=1$ for all values of j . Index $j=1$ indicates the rotor boundary and $j=J$ indicates the stator boundary. J is the total number of nodes in the radial direction.

3.1. Rotor and blade tip profiles

The nodes are generated with a uniform distribution based on arc length Eq. (9) and they appear as in Fig. 2.

$$|r_{i,j=1}(x, y)| = |r_{i-1,j=1}(x, y)| + S_i i \quad (9)$$

With reference to the blade tip profile, S_i is the length of the profile and I is the number of nodes specified, while S_i is the increment in spacing per node.

$$S_i = \sum_{i=2}^I |r_{i,j=1}(x, y) - r_{i-1,j=1}(x, y)| \quad (10)$$

$$S_i = S_i / I \quad (11)$$

In order to achieve a more accurate control of the discretisation process, an independent specification of the number of divisions I is allowed on the different regions of the grid boundaries. Hence, the parameters S_i and S_l are always defined according to Eqs. (10) and (11) but, from a geometrical point of view, they refer to different quantities. For instance, Fig. 3a and b represent S_i and S_l related to the blade walls.

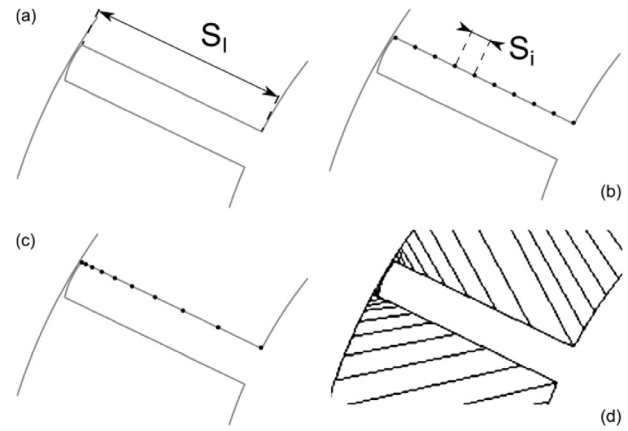


Fig. 3. Vane side wall boundary discretization: wall boundary geometry (a), equidistance distribution (b), stretching function applied distribution (c), node distribution projected on stator boundary (d).

3.2. Blade walls

The vane walls are parallel surfaces that form a convex connection with the vane tips. This convexity is the major challenge in generation of vane rotor grid because the aspect ratio between the gap and the core changes by many orders of magnitude. In order to get a valid mesh and control on the grid expansion ratio a stretching function is introduced on the vane walls. This function concentrates the nodes close to the connection with the tip and increases node spacing at the connection with the rotor. Initial equidistance distribution $r_{i,j=1}(x, y)$ is according to Eqs. (9)–(11) (Fig. 3b) and then it is modified to $r'_{i,j=1}(x, y)$ using stretching parameter σ_1 according to Eq. (12).

$$|r'_{i,j=1}(x, y)| = |r_{i-1,j=1}(x, y)| + \sigma_1 \left[\frac{\sum_{i=2}^I |r_{i,j=1}(x, y) - r_{i-1,j=1}(x, y)|}{\sum_{i=2}^I i} \right], \quad (12)$$

$i = 2, 3, 4, \dots, (I-1)$ and $0.01 \leq \sigma_1 \leq 1.0$

Resultant node distribution with 10 nodes on the vane wall and $\sigma_1 = 0.2$ is highlighted in Fig. 3c.

3.3. Stator

In order to converge the nodes at approximately mid-way in the compression chamber core, a stretching function σ_3 is introduced on the stator distribution. Initially the stator nodes are distributed as radial projections from the vane and rotor nodes with rotor centre (0, 0) as the origin i.e. distribution $r_{i,j=J}(x, y)$, Fig. 4a. This distribution is then updated using Eq. (13)–(16) to $r'_{i,j=J}(x, y)$.

$$r'_{i,j=J}(x, y) = r'_{i-1,j=J}(x, y) + \sigma_3 r_{i,j=J}(x, y) \quad (13)$$

$$\sigma_3 = \sigma_2 \frac{S'_i}{R_i} \frac{|r_{i-1,j=J}(x, y)|}{R_{rot}} \quad (14)$$

$$S'_i = \sum_{i=2}^I |r_{i,j=J}(x, y) - r_{i-1,j=J}(x, y)| \quad (15)$$

$$R_i = \sum_{i=2}^I |r_{i,j=01}(x, y) - r_{i-1,j=01}(x, y)| \quad (16)$$

$i = 2, 3, 4, \dots, (I-1)$, and $1.0 \leq \sigma_2 \leq 2.0$

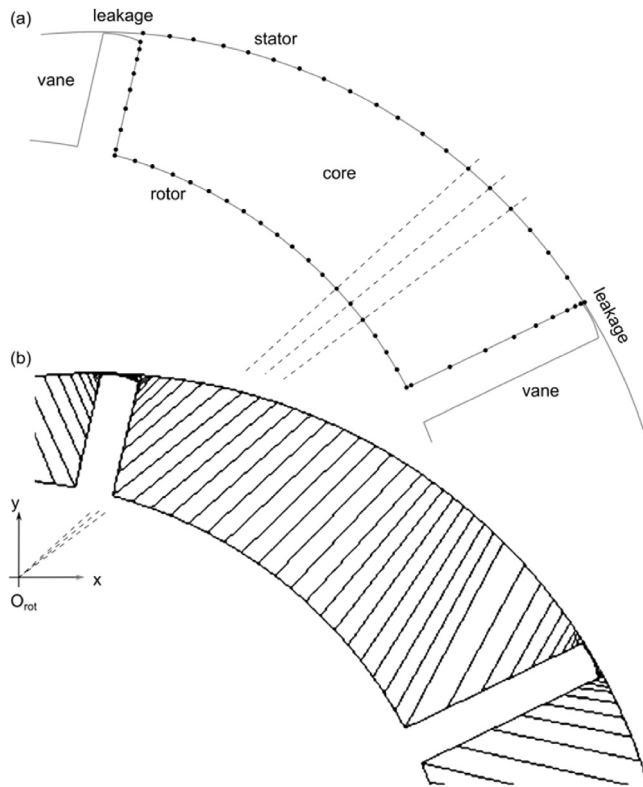


Fig. 4. Stator wall boundary discretization: vane stator distribution radially projected on stator boundary (a), stretching function applied stator distribution (b).

S'_i is the length of the stator profile from $i=1$ to i and R_i is the length of the rotor profile from $i=1$ to i . Resultant node distribution with 20 nodes on the rotor, 10 nodes on the vane wall and $\sigma_2 = 1.5$ is highlighted in Fig. 4b.

The scaling factor $\frac{|r_{i=1,j=j}(x,y)|}{R_{rot}}$ specified in Eq. (14) blends the stretching function σ_3 based on the local difference between the rotor and the stator radius. This is necessary because in the minimum gap zone the inner and outer boundaries are very close to each other and it is desirable to have distribution close to equidistance distribution. At the other extremity, on negative x-axis, the radial distance between the rotor and stator is maximum, approximately twice the axis offset and hence it is desirable to activate full scale of the stretching function.

At the position where the core region topology does not remain rectangular, as it happens when the vane slides inside the rotor, a simplification has been introduced to convert the vane side walls into short segments on the rotor surface. This is seen in Fig. 5 and it is anticipated that the leakage flow will not be influenced by this simplification as the gap clearance does not change. Nevertheless, this will have an influence on the solver stability due to sudden change in topology and could demand higher solver relaxation parameters.

4. Interior node distribution and 3D mesh assembly

The second stage in mesh generation is the distribution of interior nodes. In order to obtain a good quality hexahedral “O” grid, transfinite interpolation, orthogonalisation and smoothing have been applied. Mathematical formulation of these numerical algorithms is reported in detail in reference [3] and here briefly recalled.

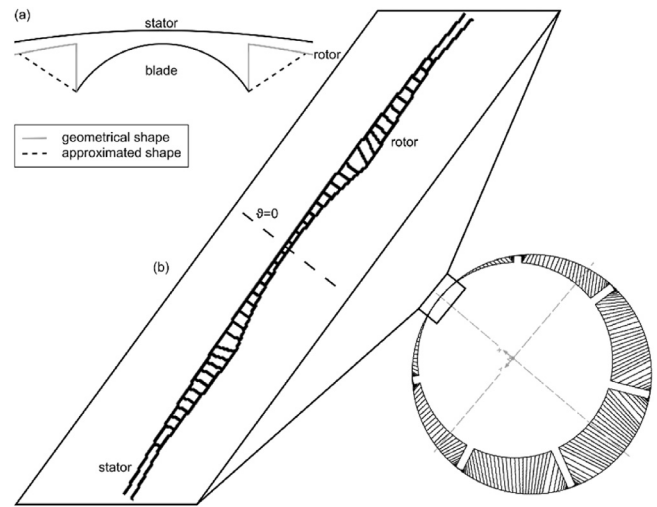


Fig. 5. Simplification at the tangency region between stator and rotor: conceptual scheme (a) and detailed view obtained during the 2D grid generation (b).

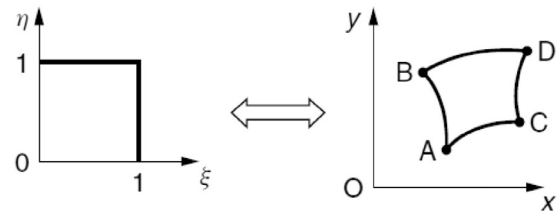


Fig. 6. Mapping of a unit square onto a curved four sided region.

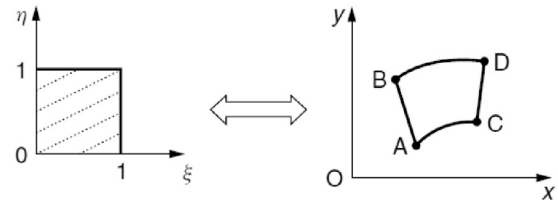


Fig. 7. Mapping of a unit square by projector P_η .

Consider a transformation $x=x(\xi, \eta)$, $y=y(\xi, \eta)$ which maps a unit square $0 \leq \xi \leq 1$, $0 \leq \eta \leq 1$, onto the interior of the region ABDC in x-y plane as shown in Fig. 6.

The edges $\xi = 0$ and 1 map to boundaries AB and CD respectively which can be formulated as position vectors $\mathbf{r}(0, \eta)$ and $\mathbf{r}(1, \eta)$. Similarly boundaries AC and BD can be formulated as position vectors $\mathbf{r}(\xi, 0)$ and $\mathbf{r}(\xi, 1)$.

A projector P_η is a new transformation that maps points in computational space to points (position vectors) in the physical space considering linear Lagrange basis functions.

$$P_\eta(\xi, \eta) = (1 - \eta) \mathbf{r}(\xi, 0) + (\eta) \mathbf{r}(\xi, 1) \quad (17)$$

This transformation maps the unit square in the plane (ξ, η) onto the region ABDC in which the boundaries AB, DC are replaced by straight lines as shown in Fig. 7. Furthermore coordinates of constant ξ are mapped into straight lines rather than coordinate curves in the physical region.

Similarly projector P_ξ can be defined as the one that maps the unit square onto a region which preserves the boundaries AB, CD but replaces the boundaries AC, BD with straight lines.

$$P_\xi(\xi, \eta) = (1 - \xi) \mathbf{r}(0, \eta) + (\xi) \mathbf{r}(1, \eta) \quad (18)$$

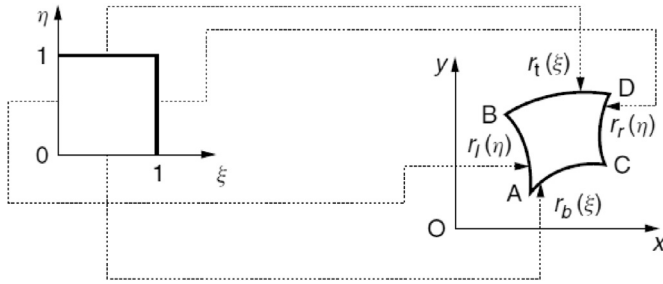


Fig. 8. - TFI mapping of a unit square by composite projector $P_{\xi} \oplus P_{\eta}$.

A composite mapping of P_{ξ} and P_{η} can be defined such that,

$$\begin{aligned} P_{\xi}(P_{\eta}(\xi, \eta)) &= P_{\xi}[(1 - \eta) \mathbf{r}(\xi, 0) + (\eta) \mathbf{r}(\xi, 1)] \\ &= (1 - \xi) [(1 - \eta) \mathbf{r}(0, 0) + (\eta) \mathbf{r}(0, 1)] \\ &\quad + (\xi) [(1 - \eta) \mathbf{r}(1, 0) + (\eta) \mathbf{r}(1, 1)] \\ &= (1 - \xi) (1 - \eta) \mathbf{r}(0, 0) + (1 - \xi) (\eta) \mathbf{r}(0, 1) \\ &\quad + (\xi) (1 - \eta) \mathbf{r}(1, 0) + (\xi) (\eta) \mathbf{r}(1, 1) \end{aligned} \quad (19)$$

This bilinear transformation has the property that the four vertices A, B, C and D are preserved, but the boundaries are all replaced by straight lines. i.e. the unit square is mapped onto a quadrilateral ABDC. Another composite mapping $P_{\xi}(P_{\eta}(\xi, \eta))$ gives $P_{\xi} P_{\eta} = P_{\xi}$.

But the composite mapping $P_{\xi} + P_{\eta} - P_{\xi} P_{\eta}$ is a transformation which maps the entire boundary of the unit square onto the entire curved boundary ABDC. This map is the Boolean Sum of the transformation P_{ξ} and P_{η} and is denoted by $P_{\xi} \oplus P_{\eta}$. The mapping has been represented in Fig. 8.

$$\begin{aligned} P_{\xi} \oplus P_{\eta} &= P_{\xi} + P_{\eta} - P_{\xi} P_{\eta} \\ (P_{\xi} \oplus P_{\eta})(\xi, \eta) &= P_{\xi}(\xi, \eta) + P_{\eta}(\xi, \eta) - P_{\xi}(P_{\eta}(\xi, \eta)) \\ &= (1 - \xi) \mathbf{r}(0, \eta) + (\xi) \mathbf{r}(1, \eta) \\ &\quad + (1 - \eta) \mathbf{r}(\xi, 0) + (\eta) \mathbf{r}(\xi, 1) \\ &\quad - (1 - \xi) (1 - \eta) \mathbf{r}(0, 0) - (1 - \xi) (\eta) \mathbf{r}(0, 1) \\ &\quad - (\xi) (1 - \eta) \mathbf{r}(1, 0) - (\xi) (\eta) \mathbf{r}(1, 1) \end{aligned} \quad (20)$$

This transformation is the basis of Transfinite Interpolation (TFI) in two dimensions. A grid will be generated by taking discrete values of ξ and η with

$$\begin{aligned} 0 \leq \xi_i &= \frac{i-1}{I-1} \leq 1, \quad i = 1, 2, 3, \dots, I \\ 0 \leq \eta_j &= \frac{j-1}{J-1} \leq 1, \quad j = 1, 2, 3, \dots, J \end{aligned} \quad (21)$$

TFI is the most commonly used approach in algebraic grid generation and, in reference [19], it has successfully applied for boundary adaptation on screw machine grid generation in the rotor domain as well as for theoretical suction and discharge ports.

Once the internal node distribution inside a given cell volume is complete, successive rotor positions are rigid body rotations for the rotor while for the fluid volume it is a deforming region. This is a complex boundary motion where rotor rotates, stator is stationary and each vane undergoes a general body motion. For simplification, variable clearance gaps have not been accounted in the presented analysis, but the meshing algorithm is flexible to further introduce a gap variation function.

Fig. 9 summarizes all the numerical steps: once the 2D mesh displayed in Fig. 9b is generated for all rotor positions, the mesh is assembled into 3D volume in the required format of the solver (Fig. 9c). In the current implementation there is possibility to export this mesh to variety of commercial CFD solvers. The rotor domain is eventually connected to suction and discharge ports via non-conformal interfaces, as explained in Section 6.

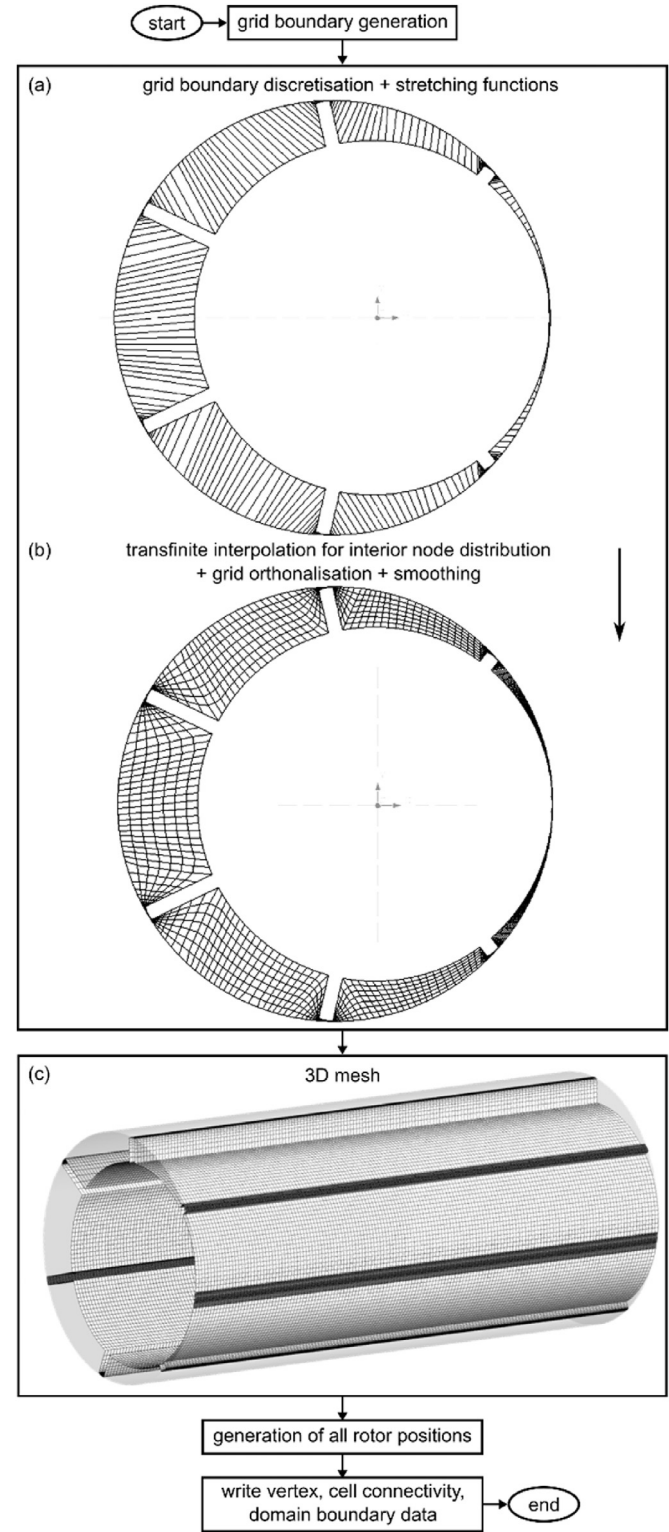


Fig. 9. Grid generation procedure: boundary discretisation (a), internal node distribution (b), final three-dimensional mesh (c).

5. Motion of customized grids in commercial CFD solvers

Grid generation procedures based on the user defined nodal displacement yield a mesh database containing all node coordinates at given reference positions, namely at defined crank angles for rotary machines. These information are stored in a series of

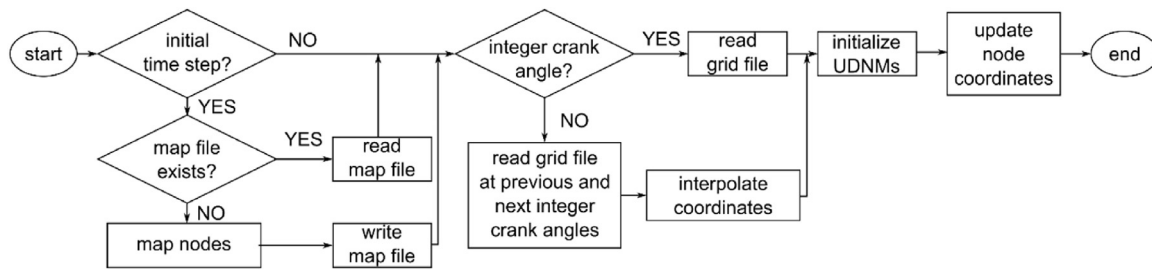


Fig. 10. Flow chart of the User Defined Function to interface the customized grids with ANSYS FLUENT.

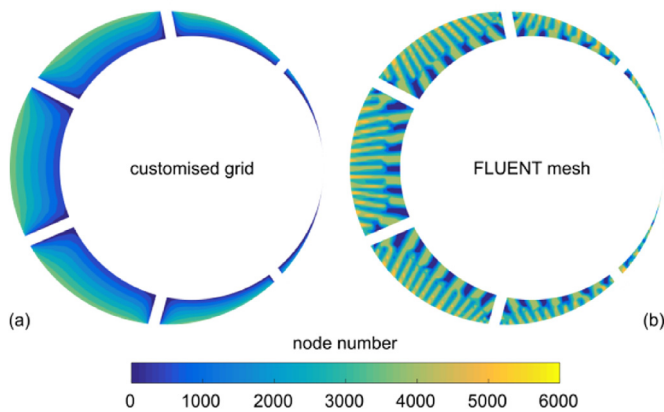


Fig. 11. Node number mismatching between the mesh loaded in FLUENT and the customised grid stored in a data file: cross sections refer to $z=0$ and initial time step.

data files that are read by the CFD solver when an update of dynamic mesh is required. If the solver is unstructured, i.e. the cell nodes do not follow a systematic (i, j, k) indexing, node numbering in data files and the mesh loaded in the CFD solver differ; in turn, prior to the node coordinates update, a preliminary mapping of the nodes also is required.

Nowadays, most of the commercial CFD solvers have developed their own tools that successfully allow to interface the solver and customised grids. Among them, the commercial CFD solvers that are compatible with the grids generated using the methodology presented in the previous sections are: STAR-CCM+, Simics PumpLinx, ANSYS CFX. ANSYS CFX uses a Fortran interface called “junction box routine” to exchange external meshes with the solver. Similarly STAR-CCM+ has a C++ library that works with the user defined vertex motion module to pass node locations to the solver at each time step. PumpLinx has a mesh deformation function that reads the external node file to update node positions in the solver. All these implementations are available in parallel architecture and need some level of code compilation and formatting depending on the solver version and operating system used.

On the other hand, to interface the unstructured cell centred solver ANSYS FLUENT with customised grids, special routines called User Defined Functions (UDFs) are required. These functions, written in C language, should be able to read node coordinates stored in data files and update the computational grid at each time step of the simulation. In particular, Fig. 10 reports a flowchart of the tasks accomplished by the UDF purposely developed for vane rotor grids in the presented work but also compatible with any grid database such as twin screw compressors.

At initial time step, node numbering between input data files and mesh loaded in FLUENT solver are different as Fig. 11 demonstrates with respect to the first cross section of the vane rotor grid ($z=0$). In the data file, node numbering start from the inner

boundary and increases with radial coordinate up to 4312 (total amount of nodes per cross section). On the other hand, in FLUENT the same coordinate pairs correspond to node numbers that exceeds 4312. Hence, node numbering spread is three-dimensional. This issue is even more significant in parallel calculations, whereas the computational domain is partitioned and leads to duplication of the nodes belonging to adjacent subdomains for data exchange.

Hence, to properly update node coordinates at the following time steps, at the initial one mesh nodes are mapped, i.e. a unique correspondence is found between nodes in the FLUENT grid (marked as FN in Eq. (22)) and node coordinates of the data file related to the mesh at the initial time step (TN^0). The mapping criterion is based on the minimum of distance between nodes: the j-th node of input file at initial time step corresponds to the ith node of the initial FLUENT mesh Υ if Eq. (22) is satisfied.

$$TN_j^0 \equiv FN_i \Leftrightarrow \text{dist}(TN_j^0, FN_i) = \dots \dots \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} < \text{dist}(TN_j^0, FN_k) \forall k \neq i \quad (22)$$

Due to the serial nature of the UDF coding, the mapping task is highly demanding in terms of computational resources. For instance, a rotor grid of 0.3 million nodes takes three days to be mapped. Nevertheless, this activity is performed just once per mesh and not at the beginning of every simulation. Indeed, as the first mapping proceeds, a file containing the mapping indexes is exported and replaces the mapping task from second simulation onwards.

Node coordinates update occurs using User Defined Node Memories (UDNMs), which are memory locations defined for every node that need to be initialized before hooking the UDF [20]. Once nodes are mapped, a first UDNM is initialized with the mapping index corresponding to the each node. Afterwards, at each time step, the corresponding grid data file is read and unmapped node coordinates are stored in auxiliary matrixes. Three additional UDNMs are used to store mapped X, Y, Z coordinates of each node respectively. A cell loop eventually assigns UDNMs values to node coordinates. If a solver exit operation is performed, the next restart automatically picks up the right node file calculated using the flow time in the data file and calculations can continue without disruption.

Depending on the angular step set in the grid generation, the number of mesh files per revolution changes. As a trade-off value between grid accuracy and disk cost of the grid database, 1° crank angle step was adopted (i.e. 360 files per revolution). To further lower the motion of the rotor grid especially at the simulation start-up, linear interpolation of node coordinates was also implemented in the UDF. When interpolation is enabled, grid files at previous and next integer crank angle steps are read. If crank angle is an integer number, interpolation is not required and only grid file at current time step will be used to initialize UDNMs. Otherwise, coordinates of the two grid files are read, interpolation is

Table 3
Geometrical features of the simulation test case.

Quantity	Dimension	Unit
Rotor radius	97	mm
Stator radius	120	mm
Number of blades	7	–
Blade thickness	8	mm
Blade tip profile radius	20	mm
Tip clearance gap	200	μm

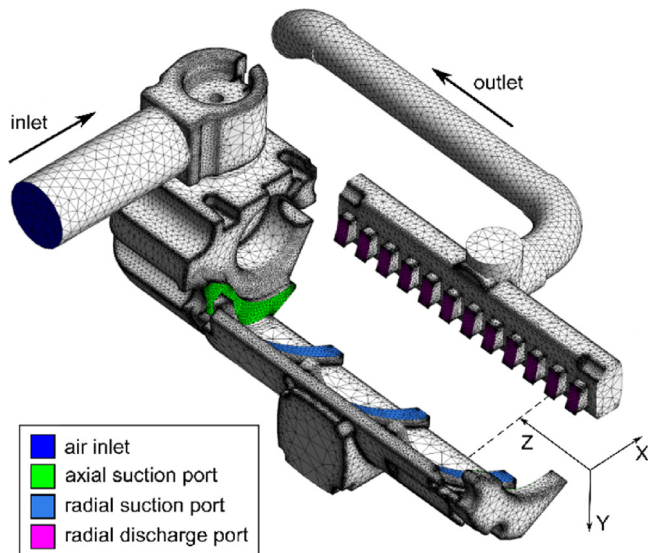


Fig. 12. Port grid blocks with interfaces (dashed line refers to the axial cross section considered for results shown in Figs. 14–17).

performed and, eventually, UDNMs are initialized with calculated node coordinates.

6. Numerical setup

The simulation test case for the numerical procedure is an industrial sliding vane compressor whose main dimensions are reported in Table 3 while its computational domain consists of three grid blocks related to suction line, compressor core and discharge line. The compressor core, i.e. the stator, rotor and blades assembly was meshed using the presented novel grid generation approach using 20 nodes to discretise the blade tip profile and 10 nodes for the walls of the blades in each chamber. These values gave a total circumferential node count of 392 along the vane rotor profile. A specification of 8 divisions in the radial direction and 150 axial cross sections resulted in a total 3D cell count of 411,600 and vertex count of 473,536. Finally, a clearance gap (C_{gap}) of 200 μm was considered. This value, slightly larger than the actual mounting tolerances between stator and rotor, allowed to limit the rotor mesh skewness to 0.2.

Apart of the fluid domain enclosed between the stator, rotor and blades, all other fluid zones were retrieved using CAD models of the industrial machine and further discretized using tetrahedral static numerical mesh produced by a commercial grid generator. As shown in Fig. 12, in the sliding vane machine the suction process occurs both through axial and radial ports: the first ones are located on the end wall plates of the compressor while the radial suction port is composed of three slanted slots along the stator. On the other hand, the discharge process takes place through a series of radial slots located on the stator. Suction and discharge grid blocks also include intake valve and discharge pipe respectively. In order to reduce the mesh count, inflation layers were neglected.

Table 4
Features of the stationary grid blocks.

	Suction	Core	Discharge
Elements	tetrahedral	hexahedral	tetrahedral
Millions of cells	0.976	0.412	0.218
Volume average skewness	0.357	0.202	0.365
Maximum aspect ratio	1210	435	56
Minimum orthogonality	0.012	0.186	0.082

Table 5
Operating conditions for the testing of the numerical methodology.

Quantity	Magnitude	Unit
Inlet pressure	1.01	bar_a
Inlet temperature	25.4	$^{\circ}\text{C}$
Outlet pressure	8.7	bar_a
Outlet temperature	88.7	$^{\circ}\text{C}$
Revolution speed	980	RPM

This choice is additionally motivated by the interest in the main flow rather than the one in the boundary layer region. Furthermore, in this way, a better expansion factor could be achieved. Additional features of the grids are reported in Table 4.

The exchange of mass and energy between the fixed and rotating grids occurs through non-conformal sliding interfaces circumferentially located along the stator surface (e.g. suction and discharge ports) or on the end wall plates for the axial suction ports. Setup of all mesh interfaces is highlighted in Fig. 12.

Air as ideal gas is used as working fluid in this study while the pressure boundary conditions were used at the inlet and the outlet of the compressor. The specification of pressure at the discharge boundary is the major influencing factor for the numerical stability of the solver. A time dependent function was used to linearly ramp up the boundary pressure to the full value of 8.7 bar_a in the first three revolutions of the simulation. On the other hand, the speed or rotations was set to the full value of 980 RPM from the simulations start. Additional experimental data used to set the boundary conditions of the simulations are reported in Table 5.

These common settings were applied to two different CFD solvers, namely ANSYS FLUENT and ANSYS CFX. ANSYS FLUENT is a Green-Gauss finite volume method with a cell centred formulation; as shown in Fig. 13a. The reference control volumes for the CFD solver are same as these created in the computational grid. On the other hand, ANSYS CFX uses the finite volume method based on elements with a cell vertex formulation. In this case, the control volume is assembled around each node, as shown in Fig. 13b. In the current study, both CFD solvers were based on the pressure-based coupled algorithm. This approach solves a coupled system of equations comprising the momentum equations and the pressure-based continuity equation. The remaining equations are solved in a decoupled way, as shown in Fig. 13c. Despite a larger memory requirement, the simultaneous resolution of the momentum and continuity equations significantly improves the rate of solution convergence [20]. Table 6 eventually summarises and compares settings and parameters used within the two selected solvers.

7. Results and discussion

The FLUENT solver proved to require more iterations to achieve converged solution with this highly deforming computational grids. This is shown by the number of sub-iterations per time step shown in Table 6, the use of relaxation factors and the advection schemes.

The simulation results obtained from CFX and FLUENT are presented in Figs. 14–17. In each figure, the comparison is made with

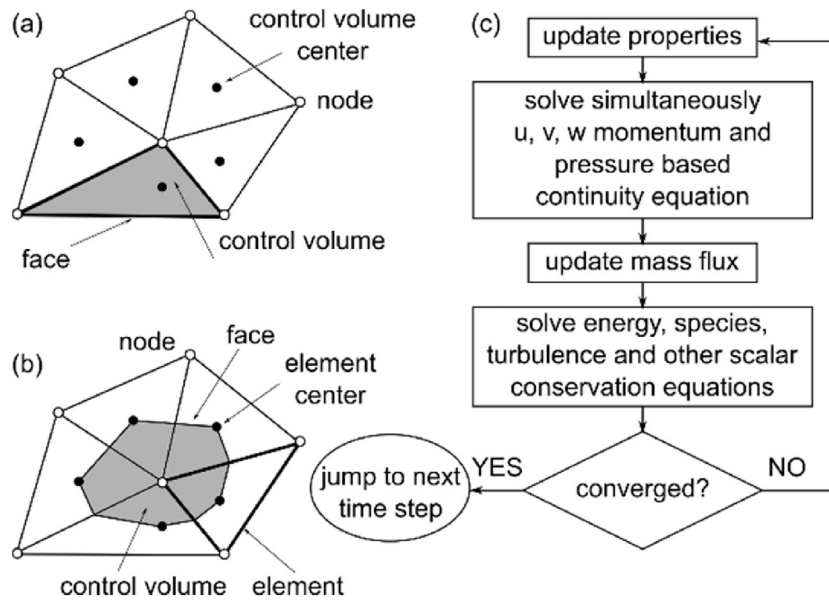


Fig. 13. Reference schemes for cell centred (a) and vertex centred (b) CFD solvers, and Pressure-Based Coupled Algorithm (c).

Table 6

Numerical setup used for the two solvers and conditional remarks.

Criteria	ANSYS CFX	Remark	ANSYS FLUENT	Remark
Mesh deformation	User defined nodal displacement	Via junction box routines in FORTRAN	User defined nodal displacement	UDF subroutines in C
Mesh in ports	Tetrahedral with boundary layer refinements	Generated by ANSYS pre-processor	Same as for CFX	
Turbulence model	SST – k Omega	Standard Wall Functions	Realizable k- epsilon	Standard Wall Functions
Inlet boundary condition	Opening	Specified total pressure and temperature	Pressure Inlet	Specified total pressure and temperature
Outlet boundary condition	Opening	Static pressure For backflow; total pressure and temperature	Pressure Outlet	Static pressure No backflow
Control volume gradients	Gauss divergence theorem	Shape functions used to interpolate ϕ	Least square cell based scheme	
Advection scheme	Upwind	High resolution, Blend of 1st and 2nd order	First order Upwind	For stability
Pressure-Velocity coupling	Co-located layout	Rhie and Chow 4th order	Staggered layout	Pressure – Standard Interpolation
Transient scheme	Second order	Fully implicit Backward Euler	First order	Implicit Backward Euler
Transient inner loop coefficients	Up to 20 iterations per time step		Up to 50 iterations per time step	
Convergence criteria	1e-03	r.m.s residual level	1e-03	r.m.s residual level
Relaxation parameters	Solver relaxation factor	0.1	Flow Courant Number – 200 Momentum and Pressure explicit relaxation – 0.05	For stability. Density, Energy and Turbulence –0.25

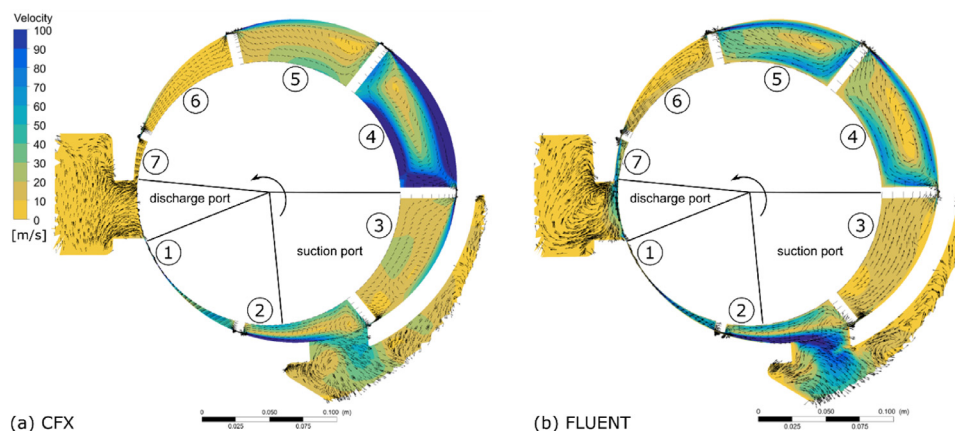


Fig. 14. Velocity magnitude with superimposed velocity vectors.

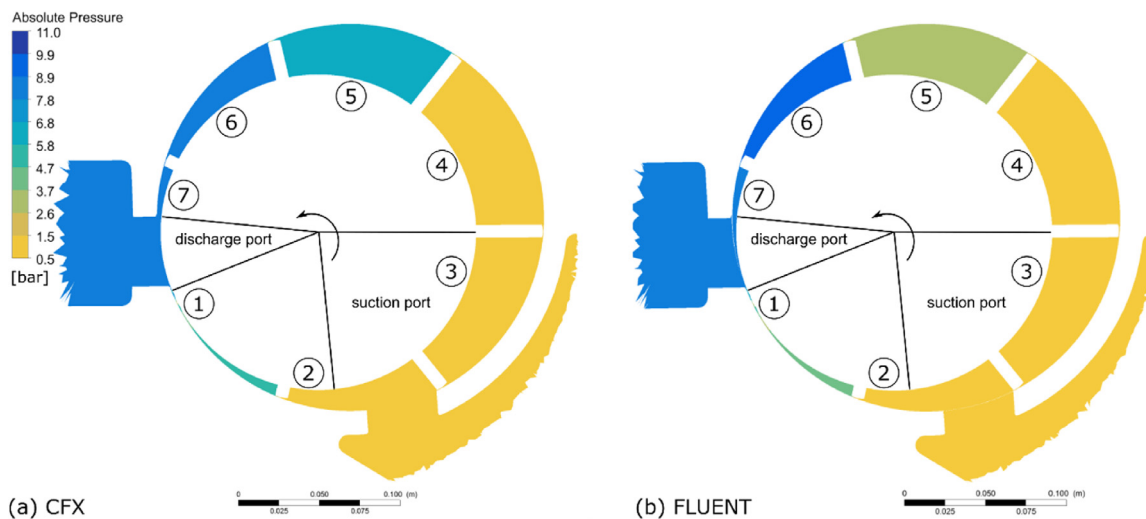


Fig. 15. Absolute pressure contours.

reference to the same cross section (15% of total axial length, as shown by the dashed line in Fig. 12) and post-processing settings. The selected crank angle position represents the beginning of the compression process for cell #4, i.e. when the blade that separates the compressor cells #3 and #4 closes the suction port. The compressor rotates counter clockwise.

Fig. 14 shows the velocity magnitude with superimposed velocity vectors while Fig. 15 presents the absolute pressure contours. Both solvers show the same overall flow topology. The velocity fields aligned with the direction of the rotor rotation and mixing of the outlet flows coming from the series of discharge grooves in the exhaust manifold are displayed in Fig. 12. Compressor cells #2 and #3 are at suction pressure as well as cell #4 which is about to start the compression process. On the other hand, cell #7 is at the discharge pressure. The volumetric compression ratio of the machine is slightly higher than the ratio between discharge and suction atmospheric pressure. For this reason, pressure in cell #6 is slightly greater than the one in cell #7.

Despite similarities, the results provided by the two CFD solvers are highly different in the regions affected by leakage flows. For instance, absolute pressure at cells #1 and #5 were higher in CFX (Fig. 15a) than in FLUENT (Fig. 15b). In addition to that, details on the flow field are presented in Figs. 16 and 17. In particular, Fig. 16 presents the velocity streamlines with superimposed velocity vectors for cells #4 and #5 while Fig. 17 is a magnified view of Fig. 14 with reference to the leakage flow between cells #5 and #6.

In both solvers, tip leakage flows cause recirculation flows which rotate in the opposite direction of the main rotation. The velocity magnitude is greater in cell #4 than #5. Moreover, velocity streamlines in CFX results (Fig. 16a) tend to follow the outer surfaces of the compressor cell: they are almost parallel to the stator and rotor surfaces as well as to the blade walls. On the other hand, in FLUENT results (Fig. 16b) the streamlines tend to have a direction that gradually converges towards the leakage gap. As shown in Fig. 17, the different streamline orientation slightly affects the velocity direction and magnitude in the region upstream the tip gap. However, the trends predicted by the two solvers completely change downstream the throat of the nozzle created by blade tip and stator. In particular, between cell #6 and #5, FLUENT results experience a greater flow velocity in the tip region that further propagates in the core region of the cell, as per Fig. 16b.

The different treatment of the leakage flow velocity in the diverging region of the tip gap is not only sensitive to the pressure

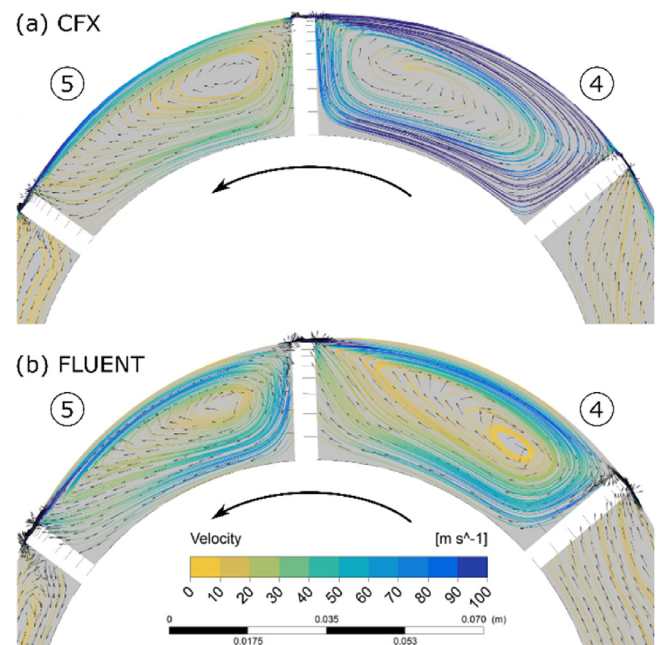


Fig. 16. Streamlines coloured with velocity magnitude and superimposed velocity vectors: detail on compressor cells #4 and #5.

difference across the cells that a given blade separates but also to the shape of the nozzle, which in turn depends on the mutual curvature between blade tip and stator. For these reasons, unlike leakage flow in Fig. 17, the one between cells #5 and #4 is greater in CFX than FLUENT.

8. Conclusions

Positive displacement compressors of the vane type can be modelled using 3D CFD in order to visualize the flow field, estimate the performance and optimize the design of a compressor. A limiting factor in application of these numerical models is the availability or generation procedures for appropriate deforming numerical grids that can be employed for calculations.

The presented work addresses this difficulty in numerical modelling by proposing a novel method for boundary generation and discretisation. Mesh boundaries are in this case described by para-

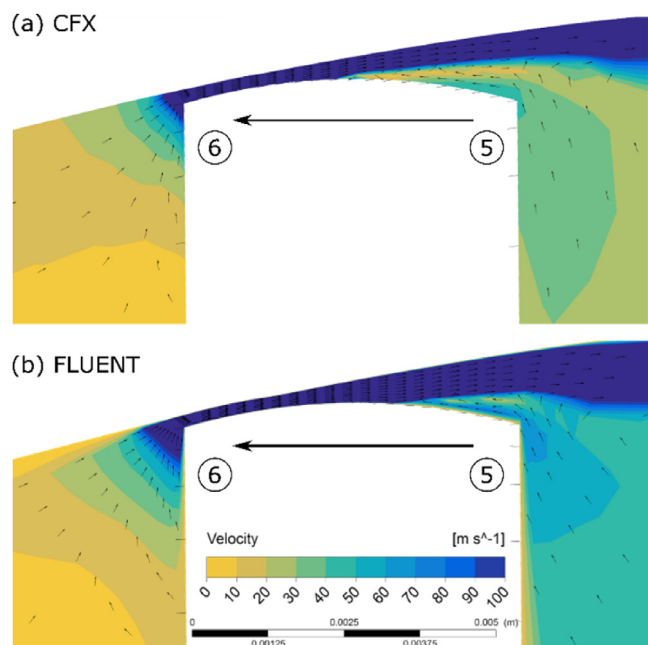


Fig. 17. Velocity contours with superimposed velocity vectors: detail on leakage flow across compressor cells #5 and #6.

metric curves generated using trigonometrical modelling of the axial cross section of the machine. The distribution of the computational nodes is performed using algebraic control functions for adaptation of boundary distribution to get control on grid quality in terms of size and expansion ratio of the cells. The distribution of internal nodes is performed using algebraic transfinite interpolation and iterative orthogonalisation and smoothing. This led to the development of a 'O' grid topology, fully hexahedral cell structure for the compressor vane chamber and also maintained the seamless connectivity of the leakage gaps with the core chambers. Such a mesh could provide accuracy in flow calculation and is suitable for parallel solver scalability by avoiding non-matching interfaces between the gap and core zones.

For testing of the generated grids, single phase simulations on an industrial air compressor were performed using commercial CFD solvers FLUENT and CFX. Despite the same computational grid and numerical schemes, the two solvers produced different numerical results. For instance, streamlines are parallel to the blade walls in CFX simulations while in FLUENT ones these are not. Furthermore, recirculation flows triggered by the leakage flow at the tip clearance gaps have different magnitude. At the beginning of compression, velocity magnitude was higher in CFX than in FLUENT; while later in compression it is reversed. Calculations obtained with FLUENT and CFX confirmed that the grid generated by the proposed algorithm was of a good quality and topology for numerical calculation using variety of CFD solvers.

Future developments of the numerical methodology will aim at the validation of the results of numerical simulation with respect to the integral performance and internal pressure history measurements. In order to achieve this goal, grid independency studies and the impact of different tip clearance gaps on the simulation results need to be investigated which will identify the optimal grid size. Another improvement to be made at the mesh level is the inclusion of the computational domains where the end wall leakage flows occur. In this way, the effects of different turbulence models at these tight gaps and at the blade tip ones could be compared. Furthermore, the sliding vane compressor taken as the reference test case and many others are not able to run efficiently without

oil injection. However, to reproduce this real conditions at numerical level, multi-phase simulations are required. In particular, while the simulation of vane compressors requires the modelling of the interaction between a gaseous fluid and a liquid fluid, other vane machines such as pumps and expanders for organic Rankine cycle applications will likely work with multiple phases of the same fluid.

Acknowledgements

The work documented in this paper has been funded by the 2015 Scholarship of the Knowledge Centre on Organic Rankine Cycle technology (www.kcorc.org), the organization formed by the members of the ORC Power Systems committee of the ASME International Gas Turbine Institute (IGTI).

The Authors are additionally grateful to Dr. Giulio Contaldi and Dr. Stefano Murgia of Ing. Enea Mattei S.p.A. (Italy) for the CAD geometry of the sliding vane compressor.

References

- [1] Prasad BGS, CFD for Positive Displacement Compressors. International compressor engineering conference at Purdue University, West-Lafayette (IN), United States; 2004. docs.lib.purdue.edu/icec/1689.
- [2] Rane S, et al. Grid deformation strategies for CFD analysis of screw compressors. *Int J Refrig* 2013;36(7):1883–93. ISSN 0140-7007. doi:10.1016/j.ijrefrig.2013.04.008.
- [3] Kovacevic A, et al. *Screw compressors—three dimensional computational fluid dynamics and solid fluid interaction*. Springer; 2006. ISBN: 978-3-540-36302-6.
- [4] Vande Voorde J, et al. A Grid Generator for Flow Calculations in Rotary Volumetric Compressors. Fourth European congress on computational methods in applied sciences and engineering; 2004. URL hdl.handle.net/1854/LU-298015.
- [5] Rane S. *Grid Generation and CFD analysis of Variable Geometry Screw Machines PhD Thesis*. City University London; 2015.
- [6] Spille-Kohoff A, Hesse J, Shorbagy AEL. CFD simulation of a screw compressor including leakage flows and rotor heating. *IOP Conf Ser* 2015;90(1) IOP Publishing. doi:10.1088/1757-899X/90/1/012009.
- [7] Hesse J, Rainer A. CFD Simulation of a Dry Scroll Vacuum Pump Including Leakage Flows. International compressor engineering conference; 2016. Paper 2498. URL docs.lib.purdue.edu/icec/2498.
- [8] Badr O, et al. Multi-vane expanders: geometry and vane kinematics. *Appl Energy* 1985;19(3):159–82. ISSN 0306-2619. doi:10.1016/0306-2619(85)90006-6.
- [9] Badr O, et al. Multi-vane expanders: vane dynamics and friction losses. *Appl Energy* 1985;20(4):253–85. ISSN 0306-2619. doi:10.1016/0306-2619(85)90018-2.
- [10] Tramschek AB, Mkumbwa MH. Mathematical modelling of radial and non-radial rotary sliding vane compressors. International Compressor Engineering Conference at Purdue University, West-Lafayette (IN), United States; 1996. URL: docs.lib.purdue.edu/icec/1151.
- [11] Al-Hawaj O. Theoretical modeling of sliding vane compressor with leakage. *Int J Refrig* 2009;32(7):1555–62. ISSN 0140-7007. doi:10.1016/j.ijrefrig.2009.07.005.
- [12] Bianchi G, Cipollone R. Theoretical modeling and experimental investigations for the improvement of the mechanical efficiency in sliding vane rotary compressors. *Appl Energy* 2015;142:95–107. ISSN 0306-2619. doi:10.1016/j.apenergy.2014.12.055.
- [13] Bianchi G, Cipollone R. Friction power modeling and measurements in sliding vane rotary compressors. *Appl Therm Eng* 2015;84:276–85. ISSN 1359-4311. doi:10.1016/j.applthermaleng.2015.01.080.
- [14] Montenegro G, et al. Evaluating the performance of a rotary vane expander for small scale organic rankine cycles using CFD tools. *Energy Procedia* 2014;45:1136–45. ISSN 1876-6102. doi:10.1016/j.egypro.2014.01.119.
- [15] Kolasinski P, Blasiak P. Numerical and experimental investigation on the rotary vane expander operation in micro ORC system. 3rd international seminar on orc power systems, October 12–14 Brussels, Belgium; 2015. ISBN: 978-2-9600059-2-9.
- [16] Zhang Q, Xu XY. Numerical simulation on cavitation in a vane pump with moving mesh. 5th international conference on computational methods Cambridge, England; 2014. URL www.sci-en-tech.com/ICCM2014/PDFs/401-835-1-PB.pdf.
- [17] Demirdiç I, Perić M. Space conservation law in finite volume calculations of fluid flow. *Int J Numer Methods Fluids* 1988;8:1037–50. doi:10.1002/flid.1650080906.
- [18] Kovacevic A. Boundary adaptation in grid generation for CFD analysis of screw compressors. *Int J Numer Methods Eng* 2005;64:401–26. doi:10.1002/nme.1376.
- [19] Kovacevic A, et al. Numerical simulation of fluid flow and solid structure in screw compressors. In: *Proceedings of ASME congress, New Orleans, Paper IMECE2002-33367*; 2002. p. 409–16. doi:10.1115/IMECE2002-33367.
- [20] ANSYS FLUENT 13 theory guide. (2011), ANSYS Inc., USA.