Boundary Adaptation in Grid Generation for CFD Analysis of Screw Compressors

Ahmed Kovacevic

Centre for Positive Displacement Compressor Technology
City University London, EC1V 0HB, UK

Tel: +44 20 7040 8780; Fax: +44 20 7040 8566

a.kovacevic@city.ac.uk, www.city-compressors.co.uk

ABSTRACT

This paper describes some aspects of an advanced grid generation method used, with Computational Fluid Dynamics (CFD) procedures, to model three-dimensional flow through screw compressors. The increased accuracy of the flow predictions thus derived, enable such machines to be designed with improved performance and for lower development costs.

To achieve this, a wholly original boundary adaptation procedure has been developed, in order to allow for convenient mapping of the internal grid points of a screw compressor, which is sufficiently flexible to fit any arbitrary rotor profile. The procedure includes a practical transformation method, which adapts the computationally transformed region to produce a regular boundary distribution on the mesh boundaries. It also allows for subsequent generation of an algebraic grid, which enables the three-dimensional domain of a screw compressor to be mapped regularly even in regions where the flow patterns are complex and the geometrical aspect ratio is high.

This procedure enables more efficient use of a CFD solver for the estimation of the flow parameters within both oil free and oil injected screw compressors, with either ideal fluids or real fluids, with of without change of phase.

Keywords: Boundary Adaptation, Algebraic Grid Generation, Screw Compressor, Numerical simulation

NOTATION

A	- area of the cell surface
a, b, c	- distances in the physical domain
d_i	- distance in transformed coordinate system
$\mathbf{d}_1$, $\mathbf{d}_2$	- vectors of diagonal
e_1, e_2	- cell edges maximal values in coordinate directions
$f^i(s)$	- adaptation variable
f^i_k	- weight function
$F^i(s)$	- integrated adaptation variable
k_P	- point counter
K	- number of points
$\mathbf{r}$	- radius vector
R_i	- grid point ratio
s	- transformed coordinate
w	- weight factor
W	- weight function
X_ξ	- grid spacing
x, y, z	- physical coordinates
X, Y, Z	– points on physical boundaries

x_p, y_p	- coordinates of the calculation point
y_p', y_p''	- first and second derivatives in vicinity of the point P
a_p, r_p, d_p	
E_p, C_{sin}, C_{cos}	- adaptation variables
C_1, C_2	– cell centres
L_m	- Lagrangian product of transformed coordinates
π	- 3.14
σ	- tension spline parameter
ξ, η	- computational coordinates

Subscripts

s	- grid values
min	- minimum
max	- maximum
p	- point

INTRODUCTION

The finite volume method is a powerful CFD numerical technique, which allows fast and accurate solution of the governing differential equations for fluid flow within complex geometries. However, an acceptable grid system, which describes shapes accurately, must be available in order for the method to be used. Inability to generate such a numerical mesh for screw compressors was the main reason why such machines have not been previously analysed by this means. Apart from the authors' publications [5], [6], [7], [12], there is hardly any reported activity in the use of CFD for screw compressor studies. This is mainly because the existing grid generators and the majority of solvers are still too weak to cope with the problems associated with the screw compressor geometry, Figure 1, and physics of the compressor process. Since a screw compressor comprises both moving rotors and a stationary housing, any numerical grid applied must move, slide and deform. Moreover, if flow is to be calculated through the compressor clearances, the geometric length scale ratio of the working chamber and clearance value may rise up to 1000. Despite this, the grid aspect ratio should be kept low. This cannot be done with the majority of existing CFD grid generators. Compressor flow, even in its simplest form, is further complicated by sharp pressure changes and high accelerations, which may drastically affect the flow structure. If, in addition, the working fluid is a real gas or a two-phase fluid or it contains particles, then there is hardly any CFD solver which can produce a straightforward solution. Therefore, special care is needed to blend the grid generation procedure with an adequate numerical solver to obtain a useful numerical solution of screw compressor processes.

The prerequisite for efficient use of the finite volume method is regular space discretisation. The number and topology of control volumes used to discretise a computational domain depends on the geometry and physics of the system and on the required accuracy. Types and structures of numerical grids are described in [1]. For three-dimensional problems, the use of hexahedral cells in structured grid systems gives the most conservative finite volume discretisation and is therefore the preferred grid type [2].

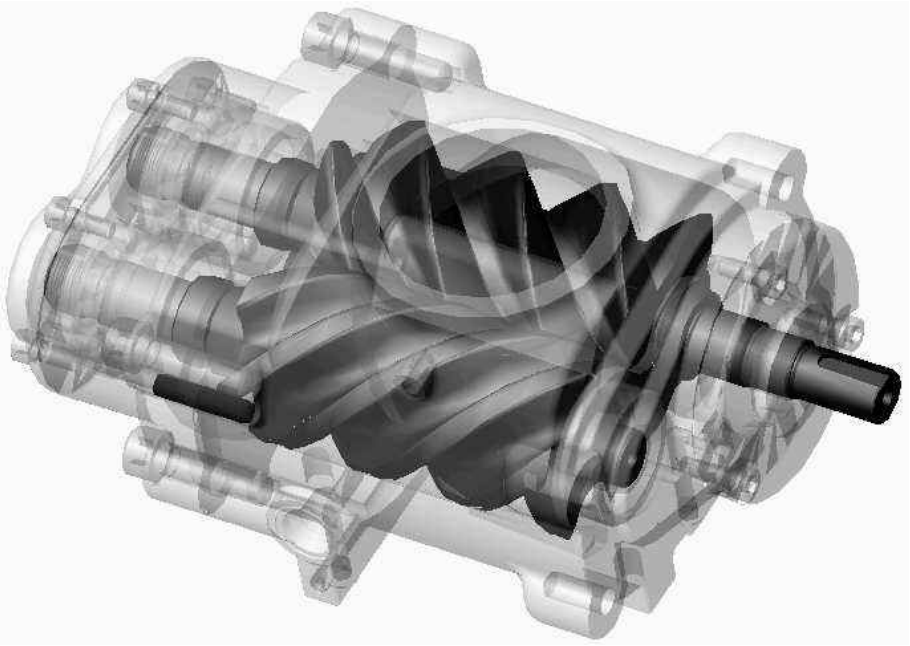


Figure 1 Screw Compressor Working Chamber

Boundary-fitted or boundary-conforming grids made of hexahedral cells can be produced by algebraic, differential or variational methods [3]. Algebraic methods calculate interior points of a grid through formulae of transfinite interpolation (TFI). They are relatively simple and enable rapid grid generation [4] and are therefore applied in the method described in this paper [6]. However, in regions where the shapes are complicated, cell faces generated by algebraic methods can degenerate severely so that the cells overlap or cross their boundaries. It is shown in this paper that an analytical grid generation procedure can generate a screw compressor numerical grid of the desired quality to avoid these problems if used in conjunction with boundary adaptation and orthogonalization.

Grid Quality Measures

The grid quality is identified through a number of factors, three of which are the most important, namely: the cell aspect ratio, the angle of non-orthogonality and the warp angle, as shown in [5].

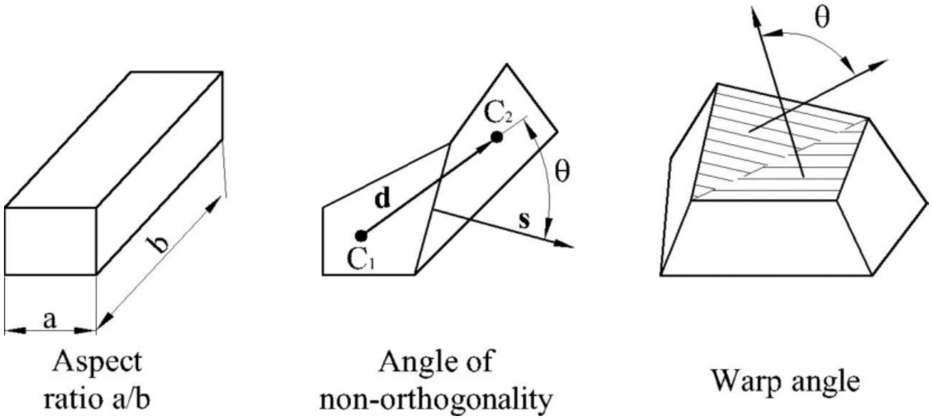


Figure 2 Measures of grid quality

Geometric consistency is a property of the numerical mesh that is difficult to measure but that influences the result significantly. It has two important aspects. The first criterion is that the numerical mesh must have a sufficient number of points in the interior of the computational domain to describe the physical domain accurately. Also, when the number of points increases to infinity, their distance apart should reduce to zero. This is an important requirement for the numerical mesh of a screw compressor, where the domain size changes by several orders of magnitude between the clearance and the main domain while the number of points in the domain is kept constant. The other requirement is that a sufficient number of points must be specified on the boundary to represent it accurately. This requires the number of points to be increased or the boundary to be adapted according to the surface geometry. If both requirements are fulfilled, then the numerical mesh is called a boundary fitting or boundary confirming mesh.

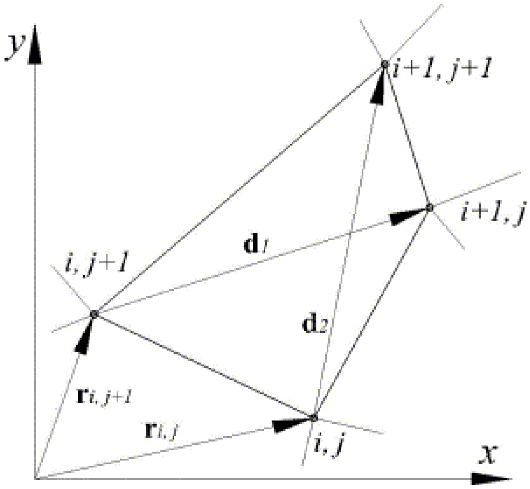


Figure 3 Vector definition of a two-dimensional cell

Methods of assessment are needed to compare and confirm the quality of different meshes. In most cases, these can be evaluated by two parameters, namely the position of the intersection point of the cell diagonals and the skewness value. They can be calculated later to give a measure of the quality not only for each particular cell but also for the mesh as a whole.

If a two-dimensional cell is considered, the skewness value σ_s can be defined as the cell area divided by two quantities represented by the maximum lengths of the cell edges in two curvilinear coordinate directions, as shown in Figure 3. Thus:

$$\sigma_s = \frac{A}{e_1 e_2}, \tag{1}$$

where

$$e_1 = \max\left\{\left|\mathbf{r}_{i+1,j} - \mathbf{r}_{i,j}\right|, \left|\mathbf{r}_{i+1,j+1} - \mathbf{r}_{i,j+1}\right|\right\}; \quad e_2 = \max\left\{\left|\mathbf{r}_{i,j+1} - \mathbf{r}_{i,j}\right|, \left|\mathbf{r}_{i+1,j+1} - \mathbf{r}_{i+1,j}\right|\right\}, \tag{2}$$

while the cell area is calculated as half the vector product of the diagonals:

$$A = \left|\frac{1}{2}\mathbf{d}_1 \times \mathbf{d}_2\right|. \tag{3}$$

If the intersection point of diagonals $\mathbf{d}_1$ and $\mathbf{d}_2$ in Figure 3 falls between the cell vertices, a numerical cell is considered to be regular. Otherwise, the cell is inverted or twisted. At the same time, the skewness value indicates whether the cell is orthogonal or not. For an orthogonal cell the skewness value σ_s=1. The value of the skewness factor tends to zero as non-orthogonality increases. The skewness factor is calculated here for a two-dimensional cell because a numerical mesh for a screw compressor is generated from two-dimensional calculations in cross sections, which are later conveniently connected in the third dimension. For three-dimensional cells the skewness can be calculated as the ratio of the cell volume to three quantities defining the maximum lengths of the edges in three curvilinear coordinate directions. Because of their properties, the grid quality measures are calculated for every cell in a computational domain to check the quality of each of them. The mesh can be accepted as regular only if all cells have a skewness value greater then zero.

Grid Topology

Block structured grids permit the convenient grid generation of complex geometries. However, although the grid generation process is simplified when the whole domain is subdivided into a number of simpler blocks, it is not always easy to select a suitable grid topology within a block. Also, although simpler than the whole domain, a sub-domain is not necessarily simple enough for efficient grid generation. The aim of algebraic grid generation is to find a transformation function $x(\xi)$ which transforms a computational domain Ξ^n to the physical domain $\mathbf{X}^n$ or vice versa, as shown in Figure 4. Four basic topology types are commonly used to specify a numerical grid within the block. These are polyhedral, H, O and C grids. Only polyhedral and O grids are used in the screw compressor grid generation process:

a) A block grid is represented as a polyhedron which retains the schematic form of a block domain. This type is mainly used for single-block grids. A numerical grid has all the properties of the computational block and fulfils all requirements of a physical domain through boundary fitting to the computational domain. In the generation of screw compressor geometry, it is used to produce grids of the inlet and outlet ports as well as of other regions that retain a polyhedral block shape. Both the physical and the numerical domains retain their block shape, as shown in Figure 4a. The numerical domain is transformed in a hexahedral block while the physical domain remains fitted within the specified boundaries.

b) A numerical grid of the O type is generated as a solid cube or rectangle in the computational domain and transformed from the physical domain, as presented in Figure 4b. This type of numerical grid has only two boundary faces in a two-dimensional domain and four boundaries in a 3D domain. The remaining two boundary faces are connected implicitly point to point.

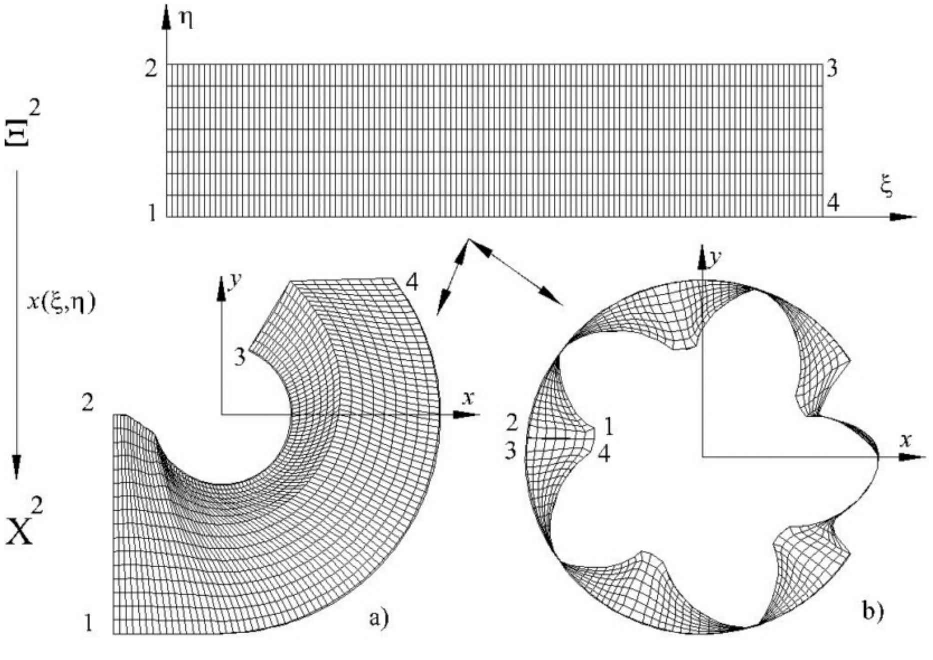


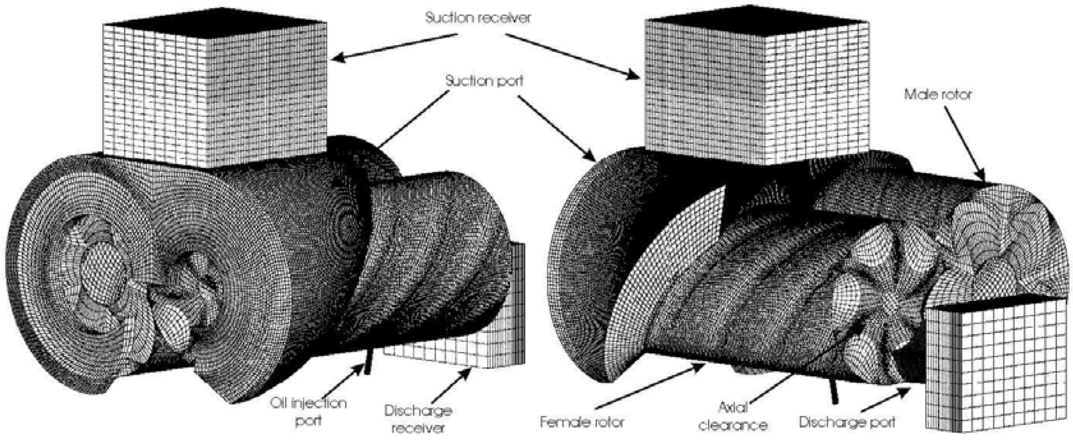
Figure 4 Patterns of grid topology in a screw compressor

a) Block type grid of the suction port *b) O-type grid of the rotor*

Decomposition of a screw compressor working domain

A composite grid, made of several structured grid blocks patched together and based on a single boundary fitted co-ordinate system is used to transform the geometry of a screw compressor into discrete volumes, as shown in Figure 5. The numerical mesh consists of several sub-domains, two of which are critical and difficult to generate. These represent the fluid domains around the male and female compressor rotors. Not only is a main working domain contained in them but also all the clearances and leakage paths, such as the radial, axial and interlobe leakage gaps and the blow-hole area. The suction and discharge ports, suction and discharge receivers and other openings, such as the oil injection port, are each represented by an additional block. The grid blocks are then connected by a sliding interface over the defined regions on their boundaries which coincide with the other parts of the entire numerical mesh.

The rotors of a screw compressor are helical type elements generated by the simultaneous revolution of the rotor profile around the rotor axis and its translation along the axis. The entire screw compressor geometry can thereby be generated within a two-dimensional definition of the rotors by the calculation of points in cross sections and connecting them with appropriate vertices in other cross sections. Screw compressor rotors are generated from 'slices', each of which is calculated separately following the procedure described further.

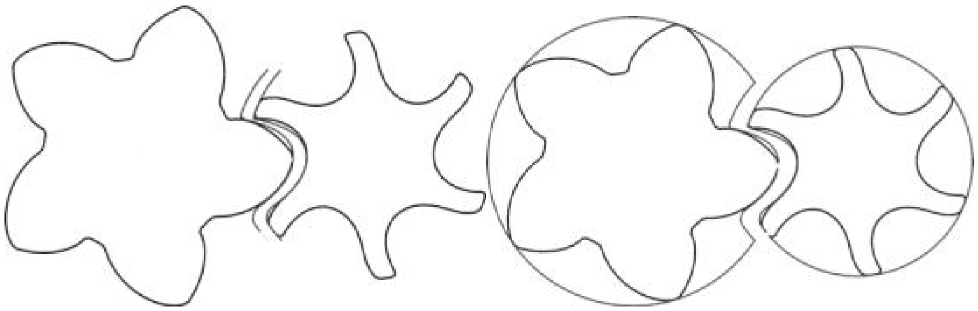


a) Suction side *b) Discharge side*

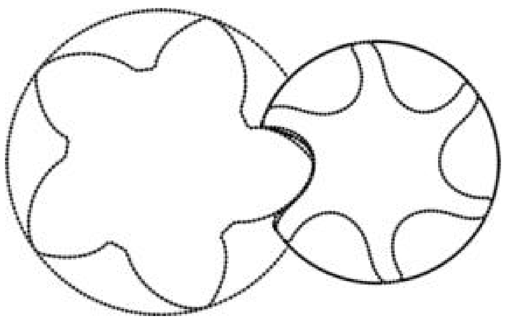
Figure 5 Numerical mesh of a screw compressor

Grid generation of screw compressor rotors starts from the definition of their spatial domains determined by the rotor profile coordinates and their derivatives. These are obtained by means of a rack generation procedure described in [8]. A distribution of characteristic curves is specified on the rack, which is a rotor of infinite radius that has the shortest possible lobe length. Interlobe clearances are accounted for by the geometry and added to the rack at this point. The envelope meshing method is then applied for generation of both the male and female rotor profiles as shown in Figure 6a. By means of this procedure, the inner boundary of the future "O" mesh is specified. The next step is to obtain an outer boundary of the "O" mesh. To do that, the rack is connected to the outer circle to form a closed line as shown in Figure 6b. The circle, to which the rack is connected, represents the rotor housing.

Figure 6c shows both parts of the mesh together. Grid generation continues with the discretisation of these two boundaries. The easiest way to distribute the points on the boundary is to put constant distances between the points along the profile. However, in such a case, some details of a particular rotor can be lost and the simulation process can lead to an inaccurate result. Therefore, some other procedure has to be performed. Preferably, the boundaries can be discretised according to the rotor geometry and flow characteristics by adaptation of the point distribution on the boundaries. The adaptation is first applied to the inner boundary which is specified in the parametric form of the rotor coordinates. The point distribution on the other boundary is modified, based on rotor discretisation in order to form a boundary definition which later enables the regular generation of grid inner points as well as fast and accurate numerical solution on that grid. That procedure is described in detail in this paper.



a) Generation of rotor profiles *b) Outer circle*



c) Boundary discretisation

Figure 6 Phases in the generation of screw compressor rotor boundaries

GENERATION AND ADAPTATION OF DOMAIN BOUNDARIES

Methods for grid adaptation have been discussed extensively by many authors [9], [4]. Generally, they can be classified as differential or algebraic methods. Differential methods are based on the solution of the Euler-Lagrange differential equations, usually solved simultaneously with other equations of fluid flow when the mesh is adapted, together with the development of a solution for fluid flow. Algebraic methods are based on a direct equidistribution technique which does not require solution of the differential equations. Consequently, they require less computational effort.

Numerical grid adaptation can be performed both dynamically and statically. Dynamic adaptation is mainly used for simulation of fluid flows with excessive local gradients of the dependent variables. It is then applied, together with the calculation of the flow properties. On the other hand, static grid adaptation is usually used to improve a numerical grid in advance, before obtaining a solution. Adaptation is based on the existing boundary geometry and on the flow characteristics expected in the domain.

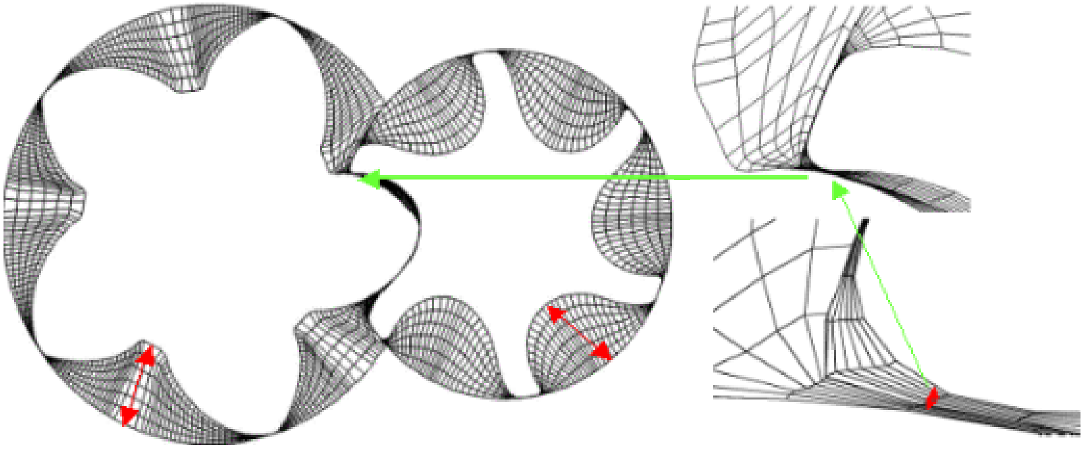


Figure 7 Comparison of cell sizes in the working chamber and clearances

A static analytical adaptation is applied in this case to the rotor boundaries of the screw compressor. There are two reasons for this; i) the cell irregularity and ii) accurate representation of the boundaries. Firstly, excessively deformed cells appear in the clearance regions as a consequence of the reduction of the cell face in the radial direction, as shown in Figure 7. Since the number of cells in the radial direction is the same for the clearances and the working cavity, a large aspect ratio appears which can be as high as 1000. Therefore, the cell shapes easily become unacceptable or the cell becomes too deformed for accurate calculation. Secondly, if a low number of boundary points is applied to a very curved boundary, for example on the top of the female rotor, some of the geometry features which greatly affect flow, could be lost. This situation can be overcome by introducing more cells along the boundary or alternatively by local mesh refinement. Both approaches lead to an increase in the number of control volumes, which affects both the efficiency and speed of calculation adversely.

The approach adopted here, therefore distributes more numerical points where required as, for example, in the clearances, and less at other places. The numerical mesh in Figure 7 is generated from the adapted 2-D rotor boundaries where the ratio of the working chamber to clearance size was 400.

Adaptation function

The analytical equidistribution technique is based on minimization of the distribution error by redistributing points along the curve to keep the product of the 'weight function' and the grid spacing constant, i.e.

$$X_{\xi} \cdot W = const. \tag{4}$$

where X_{ξ} represents the grid spacing and W is the weight function.

This equation can be treated as the Euler-Lagrange equation which, when integrated with respect to a computational coordinate ξ, becomes:

$$X(\xi) = \int_0^{\xi} \frac{d\xi}{w(\xi)} \bigg/ \int_0^{1} \frac{d\xi}{w(\xi)} \ . \tag{5}$$

In equation (5), both the grid spacing and the weight function depend on the computational coordinate ξ. This means that the equation is implicit and must be solved iteratively.

However, if equation (4) is integrated with respect to the physical coordinate x, it then gives the explicit form which can be solved directly. Its explicit form in the function of the physical coordinate is [10]:

$$\xi(x) = \int_{X_{\min}}^{x} W(x)dx \bigg/ \int_{X_{\min}}^{X_{\max}} W(x)dx . \tag{6}$$

Finally, if the starting equation is integrated with respect to the natural coordinate or the arc-length, which follows the curve, it gives:

$$\xi(s) = \int_0^{s} W(s)ds \bigg/ \int_0^{S\max} W(s)ds . \tag{7}$$

The last form of the equation appears to be the most appropriate for adaptation of complex curves which represent the boundaries of screw compressors. Its one-dimensional nature allows adaptation along one single grid line only, or along the set of grid lines which all follow a single natural direction. Multidimensional adaptation can be achieved by successive repetition of this procedure along all the sets of grid lines.

The way a grid line is adapted depends only on the selection of the weight function. It is suggested in [10] to use of a weight function in the form of:

$$W(s) = 1 + \sum_{i=1}^{I} b^i f^i(s) , \tag{8}$$

The value 1, added to equation (8), is there to ensure uniformity of distribution. The adaptation function, which appears in the same equation, is integrated along the length of the grid:

$$F^i(s) = \int_0^S f^i(s)ds\,. \qquad (9)$$

Equations (7), (8) and (9) together give:

$$\xi(s) = \frac{s + \sum_{i=1}^{I} b^i F^i(s)}{S_{\max} + \sum_{i=1}^{I} b^i F^i(S_{\max})}\,. \qquad (10)$$

Equation (10) can be used to move the grid points along a fixed curve. Parameters b^i and F^i have positive values to ensure that $\xi(s)$ is monotonic. However, if the line, which is adapted, changes in time, or if the adaptation has to be applied to a group of grid lines in a two or three dimensional grid, b^i has to be updated to keep the same emphasis of the point concentration. In this more general case it is useful to define a grid point ratio R^j assigned to each particular function f^j as:

$$R^j = b^j F^j(S_{\max}) \Big/ \left\{ S_{\max} + \sum_{i=1}^{I} b^i F^i(S_{\max}) \right\}, \qquad j = 1, 2, \ldots, I \qquad (11)$$

Now, if the grid point ratio (11) is calculated, continuous updating of b^i is avoided and the emphasis of the point concentration is kept constant. After implementing (11) in equation(10), the final form of the adaptation function becomes:

$$\xi(s) = \frac{s}{S_{\max}} \left\{ 1 - \sum_{i=1}^{I} R^i \right\} + \sum_{i=1}^{I} \left\{ R^i \frac{F^i(s)}{F^i(S_{\max})} \right\} \qquad (12)$$

Adaptation variables

In order to map the boundaries of a computational domain using the previous equation, adaptation variables, which form a weight function, must be selected. The boundary adaptation should be conducted in conjunction with the geometrical characteristics of the boundary and flow characteristics in the nearby region. In this case, the simultaneous use of any combination of two variables is presented.

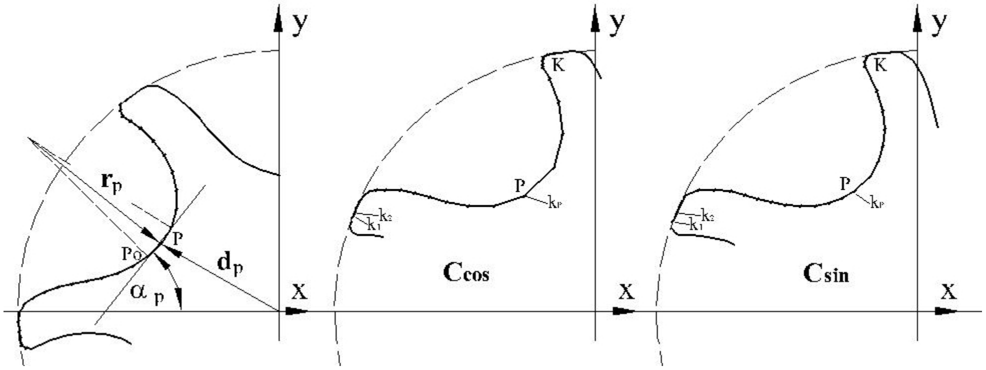


Figure 8 Graphical representation of adaptation variables

Six adaptation variables with different effects are presented here, any of which can be selected Figure 8:

The tangent angle on the curve at the calculation point $\alpha_P = \arctan\left(\frac{\partial x_p}{\partial y_p}\right)$,

The radius of curvature in the vicinity of the calculation point $r_P = \frac{1+{y'_P}^2}{y''_P}\cdot\frac{3}{2}$,

The distance of the calculation point from the rotor centre $d_P = \sqrt{x_P^2 + y_P^2}$, (13)

The curve flatness around the calculation point $E_P = \frac{\sqrt{\left(x_P - x_{P_o}\right)^2 + \left(y_P - y_{P_o}\right)^2}}{d_P - d_{P_o}}$,

Sinusoidal distribution of points $C_{\sin} = \sin(\pi \cdot k_P / K)$,

Cosinusoidal distribution of points $C_{\cos} = \left|\cos(\pi \cdot k_P / K)\right|$.

All the variables selected above are geometrical characteristics of the boundary which, when represented through their gradients along the boundary curve, give various distributions in the characteristic region. For example, if a radius of curvature changes rapidly with the natural coordinate, then the application of r_P as the adaptation variable will give more points in this region than in other places. A similar situation happens if other adaptation variables are applied.

Although the computational method introduced here allows adaptation by two selected adaptation variables it is still general, which means that any number of adaptation variables can be introduced if required.

Adaptation based on two variables

As mentioned previously, any geometrical or flow characteristic can be used as a boundary adaptation. To illustrate the method, the gradients of a tangent angle and radius of curvature have been used as weight functions f_k^1 and f_k^2 respectively in the first and second adaptation criterion:

$$f_k^1 = \left|\frac{\partial \alpha_P}{\partial s}\right|_k = \left|\frac{\alpha_{Pk+1} - \alpha_{Pk-1}}{s_{k+1} - s_{k-1}}\right|, \qquad f_k^2 = \left|\frac{\partial r_P}{\partial s}\right|_k = \left|\frac{r_{Pk+1} - r_{Pk-1}}{s_{k+1} - s_{k-1}}\right| \qquad (14)$$

In the previous equation, index k counts the numerical points along the curve which is adapted. Here, weight functions are given both differentially and in the form of finite differences. Derivatives with respect to the arc-length, i.e. the physical coordinate, are the most convenient for the static adaptation. However, in the case of a general dynamic adaptation, the derivatives with respect to the computational coordinate ξ can be more suitable. There are many ways to represent them. The most usual is central differencing on the computational mesh with equidistant distribution of the points, which gives:

$$f_k^1 = \left|\frac{\partial \alpha_P}{\partial \xi}\right|_k = \left|\frac{\alpha_{Pk+1} - \alpha_{Pk-1}}{2}\right|, \qquad f_k^2 = \left|\frac{\partial r_P}{\partial \xi}\right|_k = \left|\frac{r_{Pk+1} - r_{Pk-1}}{2}\right|. \tag{15}$$

The second step to modify the boundaries is to evaluate the integral adaptation variable (9) which is later used to calculate the point distribution from equation (10). Here a trapezoidal rule for non-uniform spacing of the physical coordinate s_k is used for calculation of the integrated adaptation variable in each numerical point. The discretised formula has a standard form for all applied weight functions:

$$F^i(S_1) = 0, \qquad F^i(s_k) = F^i(s_{k-1}) + \frac{\left[f_k^i + f_{k-1}^i\right]\cdot\left[s_k - s_{k-1}\right]}{2}, \tag{16}$$

where $k=2,3,\ldots,K$ is the number of points along the curve, while $i=1,2,\ldots, I$ is the number of variables used for adaptation.

The third step is the calculation of a new computational coordinate along the curve as a function of the previous physical coordinate. This step is performed through equation (12), which requires the grid ratio from equation (11) to be known. As a result, a new coordinate distribution is given as a function of the previous distribution:

$$\xi(k) = \xi(S_k). \tag{17}$$

In a more convenient form, this equation, which gives a new distribution of points, can be written as:

$$\xi(k) = \frac{S_k}{S_K}\cdot\left(1 - \sum_i R^i\right) + \sum_i \frac{F^i(S_k)}{F^i(S_K)}\cdot R^i, \tag{18}$$

where R^i is the weight coefficient for each adaptation variable when applying a new distribution. The sum of factors R for all adaptation variables i should be less then or equal to 1.

Finally, the fourth step is to find the inverse function of (17) which gives a new physical coordinate $\overline{S}_k$ in the new coordinate system $\overline{\xi}_k$. This can be done by an interpolation procedure which aims to find a new arc-length in the form of

$$\overline{S}_k(\overline{\xi}_k) = \sum_{m=1}^{K} L_m(\overline{\xi}_k)\cdot S_k(\xi_k), \tag{19}$$

where the bar values represent new values of the physical and computational coordinates and the others are previous values of the same coordinates. A new transformed coordinate and both the starting and ending value n_{min} and n_{max} of the Lagrangian product of transformed coordinates:

$$\begin{aligned} \overline{\xi}_k &= (k-1)/(K-1) \\ L_m\left(\overline{\xi}\right) &= \prod_{\substack{n=n_{\min} \\ n\neq m}}^{n=n_{\max}} \left(\overline{\xi} - \xi_m\right) \Big/ \left(\xi_n - \xi_m\right) \end{aligned} \tag{20}$$

must satisfy the conditions:

$$\xi_{n_{\min}-1} \leq \overline{\xi}_k \leq \xi_{n_{\min}}; \qquad 1 \leq n_{\min} \; and \; n_{\max} \leq K \;. \tag{21}$$

An example of the use of a geometrical weight function to adapt the rotor boundary of a screw comrpessor is shown in Figure 9. On the left hand figure, a uniform distribution is shown along the pair of rotors with 4/6 lobes on the male/female rotors. The male rotor diameter is 142.25 mm and the distance between the axes is 108.4 mm. The number of cells generated by such a distribution of boundary points on the both rotors is the same.

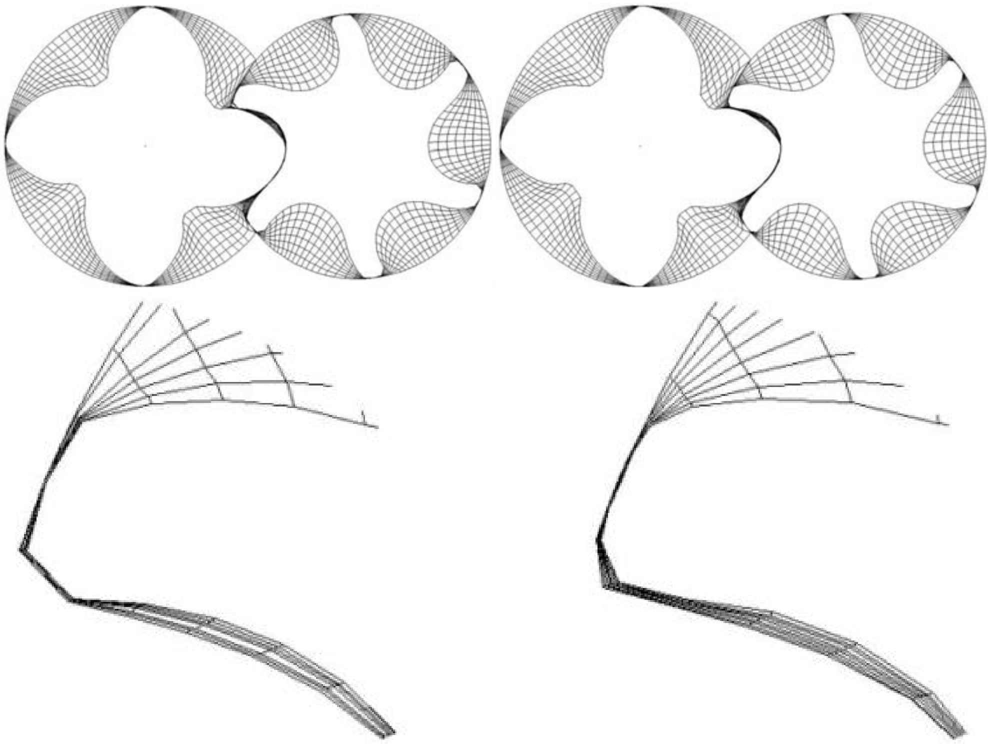


Figure 9 Comparison between the original (left) and adapted rotors (right)

The boundary distribution on the rotors on the right of Figure 8 is modified with the equal value of 0.2 of two weighting functions, namely the tangent angle and the radius of curvature. This produces a much better arrangement of both the boundary points and the entire mesh, which is particularly visible in the gaps and rotor inter-connections. However, in the parts of the numerical mesh where the cells are not so deformed and where the pattern of flow variables is expected to be nearly uniform, the cell size is not modified at all.

The distribution of boundary points plays a significant role in the grid generation process. A more appropriate boundary distribution allows easier generation of inner mesh nodes.

Mapping of the outer boundary

As explained earlier, the desired distribution on the rotor inner boundary of a 2-D “O” mesh is obtained by adapted mapping. In Figure 10, points are given in index notation with respect to the physical coordinate system:

$$\mathbf{r}_{i,j=0} = \mathbf{r}_{i,j=0}\left(x, y\right) \tag{22}$$

The outer boundary needs to be mapped with the same number of points as the inner boundary. This usually means that the cells in the structured grid generated from nodes on the boundaries are regular. The easiest way to obtain the same number of points on both boundaries is to apply the same arc-length distribution of points to the outer boundary as to the inner one. This distributes the boundary nodes for generation of the inner points successfully only if the mesh is simple and its aspect ratio does not change rapidly. However, for complex meshes and high aspect ratios, this method usually does not produce a distribution which results in a regular mesh. Therefore, a new arc-length on the outer boundary has to be specified before the generation of the inner points can be performed. The procedure starts with the transformation of the outer boundary to a straight line, then its adaptation to a new coordinate system and subsequently it continues with reverse transformation to the physical boundary in parametric form. To apply the procedure, the outer circle is first mapped with the same number of points and the same arc-length distribution as the rotor itself:

$$\mathbf{r}_{i,j'=1} = \mathbf{r}_{i,j'=1}\left(x,y\right) = \mathbf{r}_{i,j'=1}\left(\frac{s_i}{s_I}\right) \tag{23}$$

where s_i is a natural coordinate of the rotor i.e. this is a distance from the starting point to the point i along the rotor curve, while s_I is the length of the rotor boundary.

Figure 10 shows how the initial distribution of the points on the outer circle is achieved by the same arc-length as on the inner boundary represented by the screw compressor female rotor. The distribution of points obtained by this means is not always satisfactory, especially if the points are extrapolated to the rack.

Lengths a, b and c are shown for an arbitrary point i in the physical domain as shown on the left of Figure 11.

These are calculated as:

$$\begin{aligned} a &= \sqrt{\left(x_i - x_{i-1}\right)^2_{j=0} + \left(y_i - y_{i-1}\right)^2_{j=0}}, \\ b &= \sqrt{\left(x_{i-1,j=0} - x_{i-1,j'=1}\right)^2 + \left(y_{i-1,j=0} - y_{i-1,j'=1}\right)^2}. \\ c &= \sqrt{\left(x_{i,j=0} - x_{i-1,j'=1}\right)^2 + \left(y_{i,j=0} - y_{i-1,j'=1}\right)^2} \end{aligned} \tag{24}$$

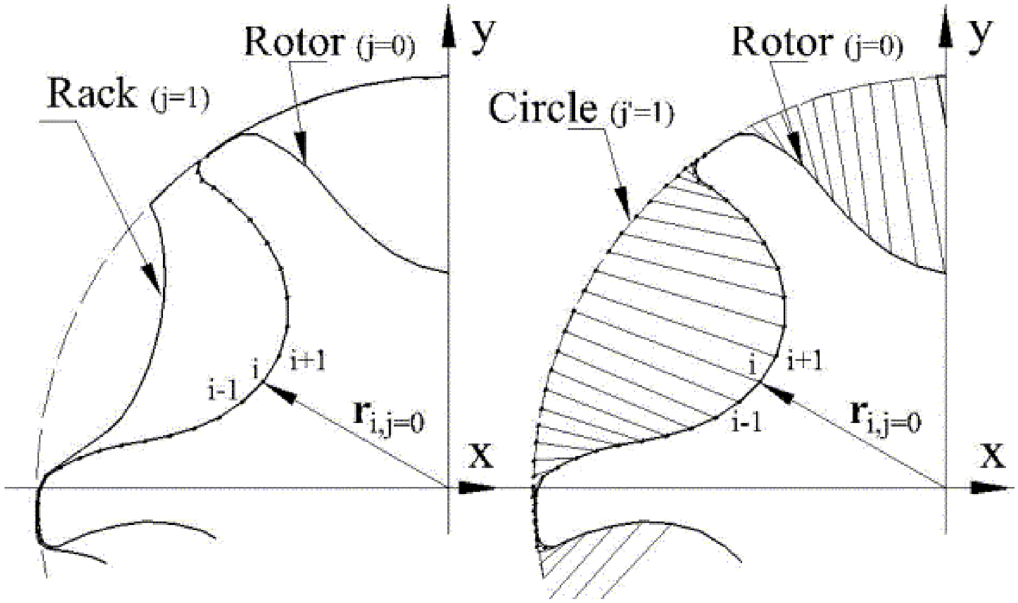


Figure 10 Point distribution for rotor and circle with equal arc-length

The angle between *a* and *c* is determined by the cosine theorem:

$$\cos\alpha_{ac} = \frac{a^2 + b^2 - c^2}{2ab} \tag{25}$$

The computational coordinate system ξ-η is shown in the right diagram of Figure 11 with the rotor and circle coordinates transformed. The outer circle in the physical domain is transformed to the straight line along the ξ axis in the computational domain. The rotor profile is transformed from the physical to the computational domain so that each computational point on the rotor has the same value on the ξ coordinate as its corresponding point on the circle.

The point coordinates in the computational domain are given by the following expressions:

$$\begin{aligned} &\xi_{i,j=0} = \xi_{i-1,j=0} + c\,\cos\alpha_{ac}, && \xi_{i,j'=1} = \xi_{i,j=0} \\ &\eta_{i,j=0} = c\sin\alpha_{ac}, && \eta_{i,j'=1} = 0 \end{aligned} \tag{26}$$

The computational cells are formed between the points in the computational domain, as shown in Figure 11. As a consequence of the transformation, both the right and left cell boundaries are produced as vertical lines. Unfortunately, some of the cells generated by this procedure are either inverted or twisted. However, these can become regular if the points on the straight edge in the computational domain can be rearranged to form a monotonically increasing or decreasing sequence. In that case, the reverse transformation from the computational domain would also give a regular point distribution on boundaries in the physical domain.

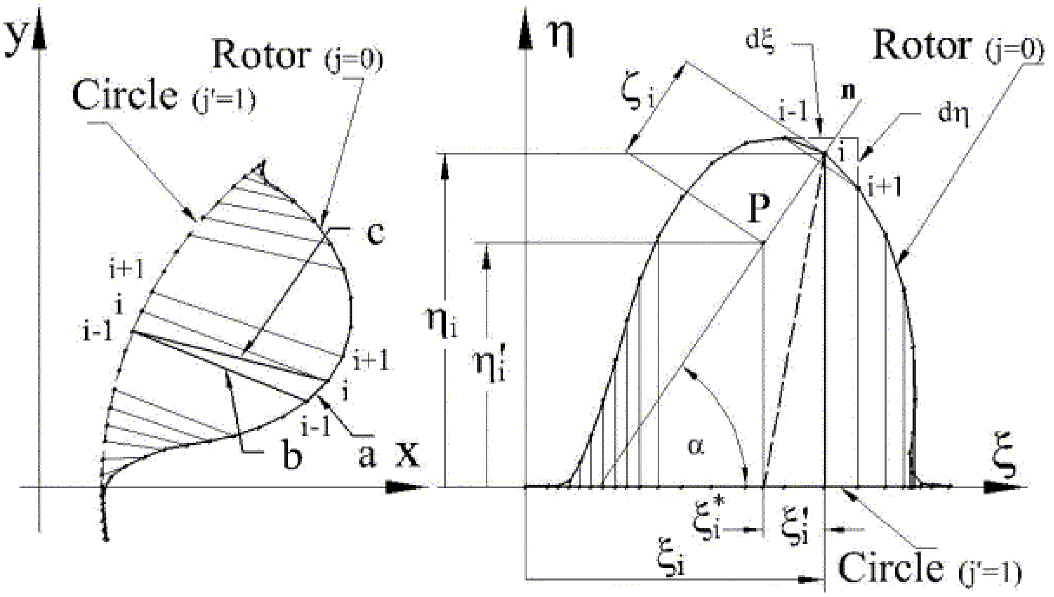


Figure 11 Transformation from physical (left) to computational domain (right)

Consider point *i* in the computational domain on the rotor boundary as shown in the right diagram of Figure 11. The line normal to the rotor profile in that point is defined by the angle between that line and the ξ axis as:

$$\tan\alpha = \frac{d\xi}{d\eta} \qquad (27)$$

Assume that point P lies on that normal line and its projection to the ξ axis gives the desired point distribution on the horizontal boundary, as shown in the diagram by the dashed line. The position of point *P* is defined by its vertical coordinate η'_i. The distance between points P and i is given by ζ_i, which is a function of the vertical coordinate of point P_i. In that case, the horizontal projection of the point P_i can be calculated as:

$$\xi_i^* = \xi_i - \xi'_i, \qquad (28)$$

The vertical projection is specified as:

$$\eta'_i = \eta_i - \zeta_i \cdot \sin\alpha. \qquad (29)$$

The ratio between the distance ζ_i and the vertical coordinate of point P_i is given by:

$$k_i = \frac{\eta'_i}{\zeta_i}, \qquad (30)$$

where the coefficient k_i can be any number greater then or equal to zero. If k_i is equal to zero then the point P_i is positioned on the ξ axis. However, if $k \to \infty$ then ζ_i becomes zero and point P_i corresponds to point *i* on the rotor. In that case, the distribution of the points is unchanged. By inserting (30) into equations (28) and (29), the new point projection on the ξ axis becomes:

$$\xi_i^* = \xi_i - \eta_i \cdot \frac{\cos\alpha}{k_i + \sin\alpha}. \tag{31}$$

If the coefficient k in equation (31) has a constant value $k=1$, the new point distribution in the computational domain is always regular, as shown in the left diagram of Figure 12.

It then remains to make the inverse transformation from the line in the computational domain to the circle in the physical domain with respect to the new arc-length:

$$\mathbf{r}^*_{i,j'=1} = \mathbf{r}^*_{i,j'=1}(x,y) = \mathbf{r}_{I,j'=1}\left(\frac{\xi_i^*}{\xi_I^*}\right). \tag{32}$$

The result of that procedure is shown on the right diagram in Figure 12. Linear interpolation or extrapolation between corresponding points on the circle *j'=1* and rotor j=0 gives the point on the rack, as shown in the right drawing of Figure 12.

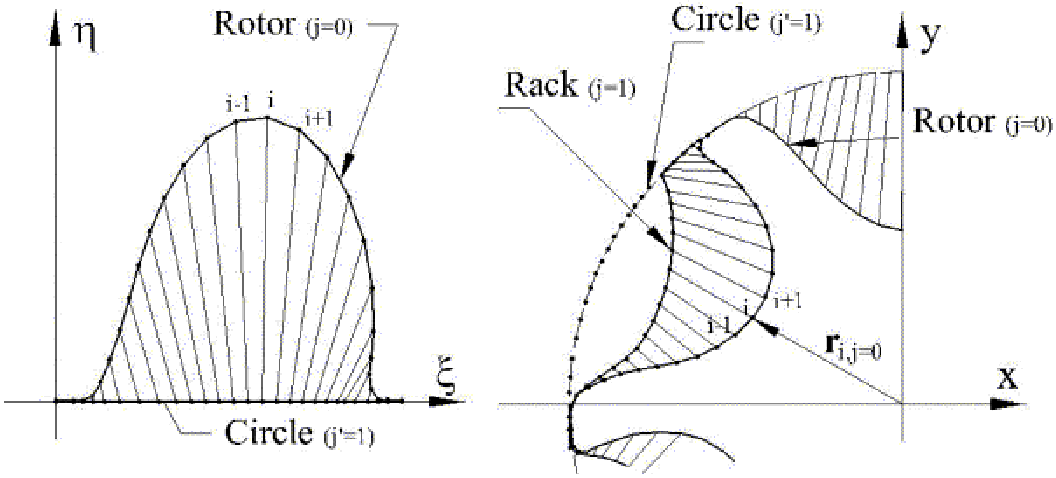


Figure 12 Final point distribution in computational (left) and physical (right) domain

The method presented here ensures the satisfactory distribution of boundary points in the 2D cross section of the screw compressor rotor domains. The level of redistribution can be controlled by the factor k_i, which may be constant for all points through the domain or can be changed for each point by use of some characteristic parameter. For the screw compressor rotors, it appears that a constant value *k=1* always gives a regular distribution of points on the boundaries, which is an essential prerequisite for successful generation of the internal points.

GENERATION OF VOLUME GRIDS

Algebraic grid generation methods are often based on a transfinite interpolation. This is defined as a multivariate interpolation procedure or a Boolean sum of univariate interpolations along each computational coordinate [4]. The method is commonly used for grid generation in domains with smooth boundaries that are not highly deformed, or as an initial approximation for the iterative process of an elliptic grid solver. However, in conjunction with the boundary adaptation and orthogonalization, transfinite interpolation can be used successfully for domains with complex boundaries which is a convenient method for screw compressor numerical grid generation.

After the boundaries of a physical domain have been calculated and the boundary points adapted to the geometry conditions, they have to be mapped to a computational domain in which the calculation of the inner nodes of a computational mesh will be performed. The coordinates of the 2D physical domain are given in an *x-y* coordinate system while the computational coordinates are ξ-η as shown in Figure 13. Both, the block grid (bottom) and the “O” grid (top) are mapped to a similar computational grid. These domains are highlighted in the figure.

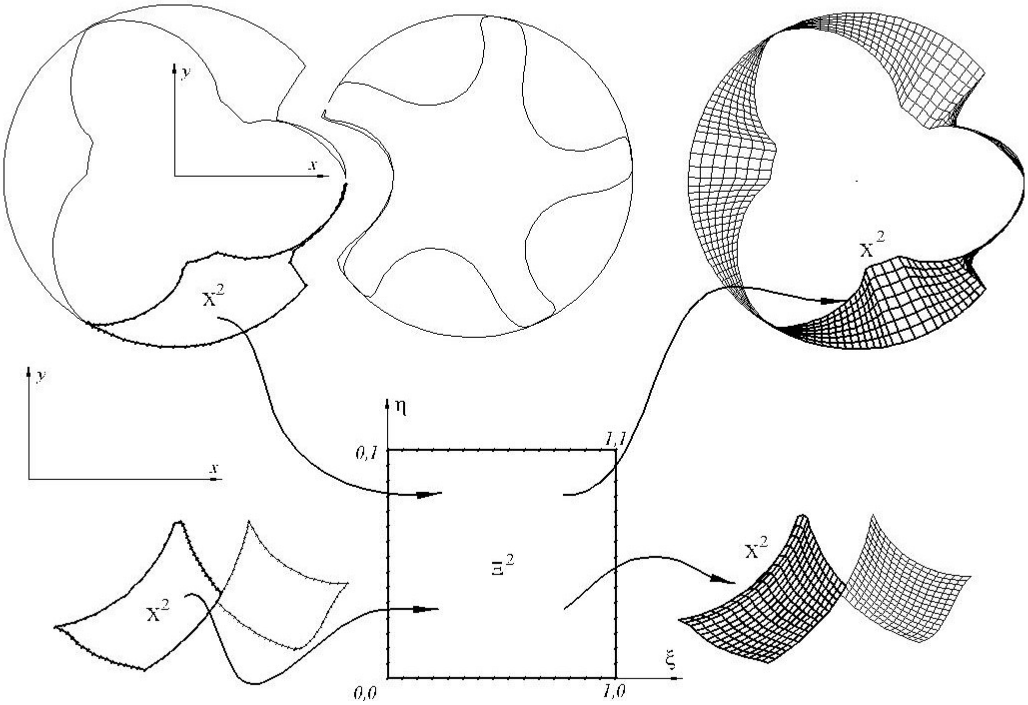


Figure 13 Conformal mapping of a physical domain X^2 to computational domain Ξ^2

The algebraic grid generation method has been developed by employing both well known principles of transfinite interpolation [3], [4] and novel features such as multidimensional stretching functions [11], grid orthogonalisation and smoothing [12]. This method is used for calculation of the inner point coordinates for screw compressor domains. More details on it can be found in [6] and [7].

APPLICATION OF THE METHOD TO CFD ANALYSIS OF SCREW COMPRESSORS

Dry compressors are commonly used to produce pressurised air, free of any oil. A typical example of such a machine is given in Figure 13. This is a single stage compressor with 4 male and 6 female rotor lobes. The male and female rotor outer diameters are 142.380 mm and 135.820 mm respectively, while their centre lines are 108.4 mm apart. The rotor length to main diameter ratio *l/d*=1.77. Thus, the rotor length is 252.0 mm. The male rotor, with a wrap angle α_w=248.4^0 is driven at a speed of 6000 rpm by an electric motor through a gearbox. The male and female rotors are synchronised through timing gears with the same ratio as that of the compressor rotor lobes i.e. 1.5:1. The female rotor speed is therefore 4000 rpm. The male rotor tip speed is then 44.7 m/s, which is moderate for dry air compressors. The working chamber is sealed from its bearings by a combination of lip and labyrinth seals.

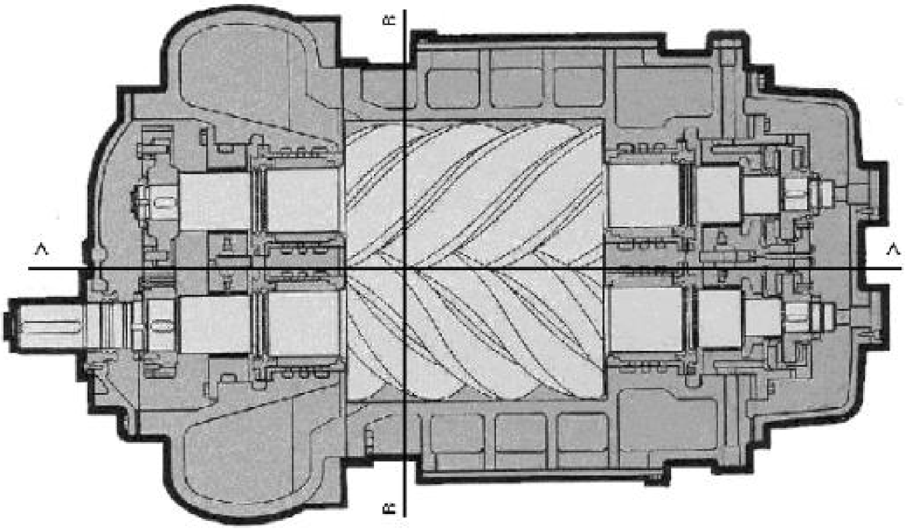


Figure 14 Cross section of a dry screw compressor

The compressor takes in air from the atmosphere and discharges it to a receiver at a constant output pressure of 3 bar. Although the pressure rise is moderate, leakage through radial gaps of 150 μm are substantial. In many studies and modelling procedures, volumetric losses are assumed to be a linear function of the cross sectional area and the pressure difference, assuming that the interlobe clearance is kept more or less constant by the synchronising gears. The leakage through the clearances is then proportional to the clearance gap and the length of the leakage line. However, a large clearance gap is needed to prevent contact with the housing caused by rotor deformation due to the pressure and temperature changes within the working chamber. Hence, the only way to reduce leakage is to minimise the length of the sealing line. This can be achieved by careful design of the screw rotor profile. Although minimising leakage is an important means of improving screw compressor efficiency, it is not the only one. Another is to increase the area between the lobes and thereby increase the compressor flow capacity, thereby reducing the relative effect of leakage. Modern profile generation methods take these various effects into account by means of optimisation procedures which lead to enlargement of the male rotor interlobes and reduction in the female rotor lobes. The female rotor lobes are thereby strengthened and their deformation thus reduced.

In order to demonstrate the improvements possible from such rotor profile optimisation, a three dimensional flow analysis has been carried out for two different rotor profiles within the same compressor casing, as shown in Figure 15, using the foregoing numerical method for the adaptation of the boundary point distribution. Both rotors are of the “N” type and are rack generated [8].

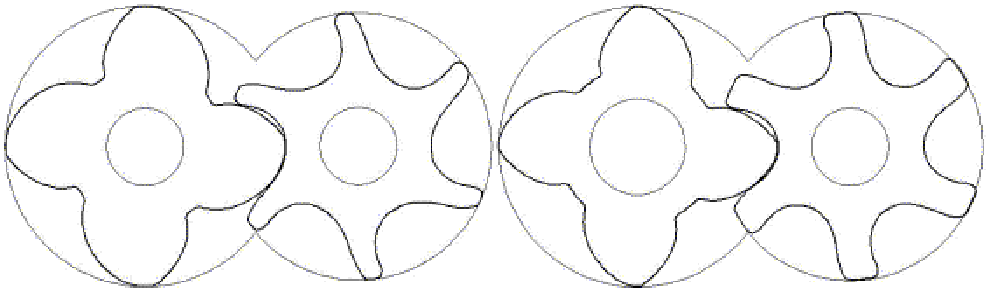


Figure 15 ‘N’ Rotors, Case-1 left, Case-2 right

Case 1 is an older design, similar in shape to SRM “D” rotors. Its features imply that there is a large torque on the female rotor, the sealing line is relatively long and the female lobes are relatively weak.

Case 2, shown on the right of Figure 15, has rotors optimised for operating on dry air. The female rotor is stronger and the male rotor is weaker. This results in higher delivery, a shorter sealing line and less torque on the female rotor. All these features help to improve screw compressor performance.

The results of these two analyses are presented in the form of velocity distributions in the planes defined by cross-sections A-A and B-B, shown in Figure 15.

Grid generation

In Case-1, the rotors are mapped with 52 numerical cells along the interlobe on the male rotor and 36 cells along each interlobe on the female rotor in the circumferential direction. This gives 208 and 216 numerical cells respectively in the circumferential direction for the male and female rotors. A total of 6 cells in the radial direction and 97 cells in the axial direction is specified for both rotors. This arrangement results in a numerical mesh with 327090 cells for the entire compressor. The cross section for the Case-1 rotors is shown in Figure 16. The female rotor is relatively thin and has a large radius on the lobe tip. Therefore, it is more easily mapped then Case-2 which has a smaller tip radius, as shown in Figure 17.

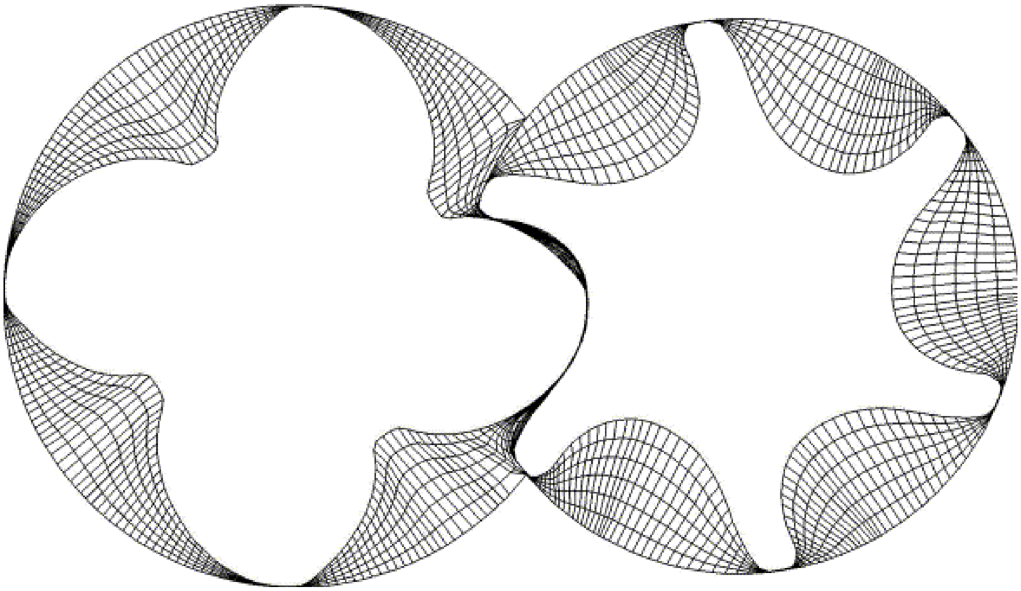


Figure 16 Cross section through the numerical mesh for Case-1 rotors

The rotors in Case 2 are mapped with 60 cells along the male rotor lobe and 40 cells along the female lobe, which gives 240 cells along both rotors in the circumferential direction. In the radial direction, the rotors are mapped with 6 cells while 111 cells are selected for mapping along the rotor axis. Thus, entire working chamber for this compressor has 406570 cells. In this case, different mesh sizes are applied and different criteria are chosen for the boundary adaptation of these rotors.

The main adaptation criterion selected for the rotors is the local radius curvature with a grid point ratio of 0.3 to obtain the desired quality of distribution along the rotor boundaries. By this means, the more curved rotors are mapped with only a slight increase in the grid size to obtain a reasonable value of the grid aspect ratio. To obtain a similar grid aspect ratio without adaptation, 85 cells would have been required instead of 60 along one interlobe on the female rotor. This would give 510 cells in the circumferential direction on each of rotors. If the number of cells in the radial direction is also increased to be 8 instead of 6 but the number of cells along axis is kept constant, the entire grid would contain more then a million cells which would, in turn, result in a significantly longer calculation time and increased requirements of computer memory.

In order to explore this situation more closely, three combinations of adaptation criteria for the rotor tip detail are presented in Figure 18. The applied number of grid points is kept the same for all three cases.

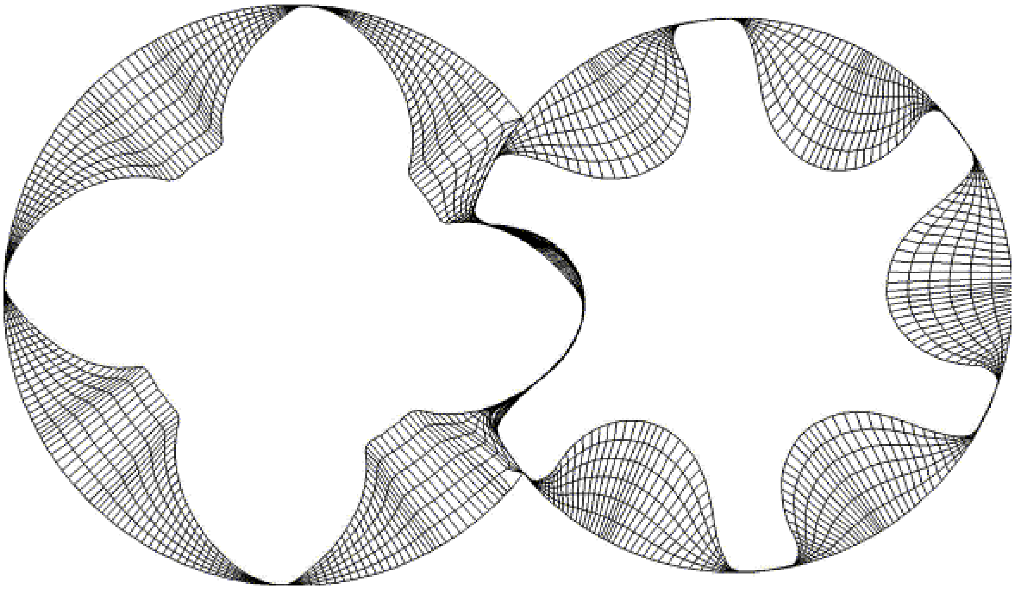
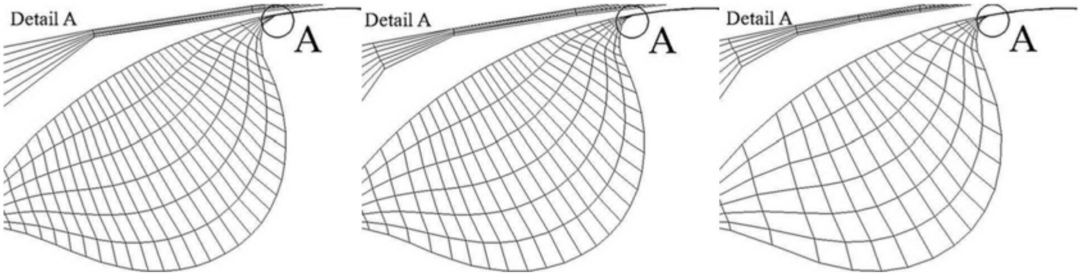


Figure 17 Cross section through the numerical mesh for Case-2 rotors

In case i), the far left diagram is obtained with equidistant distribution of boundary points. This gave an unacceptably high cell ratio and very poor capture of the small radius on the tip of the rotor. This tip comes into contact with the other rotor, and form's an important leakage path known as the 'blow-hole' area.

In case ii) two adaptation functions were used for the distribution of boundary points in the middle diagram, namely the local radius curvature with the grid point ratio R_1=0.3 and the tangent angle with R_2=0.1. This gave much better capturing of small details on the rotors. The cell ratio in the long clearance path between the rotor and the housing was still high but acceptable for obtaining solution.

In case iii) the best distribution of both the boundary and the internal points is obtained with the local radius curvature weighted 0.5 and the 'flatness' in the vicinity of point with grid point ratio 0.1. This numerical mesh is the one most easy to use for a CFD solution since the cell ratio is kept low in all domains while all other important features of the rotors are captured properly.



Uniform distribution $F_1=\mathbf{r}_p$, $R_1=0.3$; $F_2=\alpha_p$, $R_2=0.1$ $F_1=\mathbf{r}_p$, $R_1=0.5$; $F_2=E_p$, $R_2=0.1$

Figure 18 Application of different adaptation criteria

Mathematical model and solver

The mathematical model used is based on the momentum, energy and mass conservation equations as given in [5] and [6]. The equation for space law conservation is calculated in the model in order to obtain cell face velocities caused by the mesh movement. The system of equations is closed by Stoke's, Fourier's and Fick's laws and the equation of state for an ideal gas. In the case of calculation with real fluids, the equation of state is based on the reality factor with linearised coefficients over the desired region as described in [5]. The Euler-Lagrange method is used for calculation of two-phase flows where a concentration equation is calculated for each dispersed phase.

Field Code Changed

The numerical solver used in this paper for calculation of fluid flow in screw compressors is based on finite volume discretisation and a conjugate gradient (CG) or CGSTAB method. This allows simultaneous calculation of the fluid flow and solid structure over both structured and unstructured numerical meshes of arbitrary shape with sliding and stretching boundaries. Details are available in [5].

To establish a full range of working conditions and to obtain an increase of pressure from 1 to 3 bars between the compressor suction and discharge, 15 time steps were required. A further 25 time steps were then needed to complete the full compressor cycle. Each time step needed about 15 minutes running time on a 2000 MHz AMD Athlon processor. The solution of the presented case required about 400 MB of RAM.

Comparison of the screw compressor flow calculations for the two different rotor profiles

Figure 19 shows a comparison of the velocity vectors for Case-1 and Case-2 at the cross sections B-B (Figure 14). The left diagram is given for Case-1 rotors and the right one for Case-2. As may be seen, the Case 2 rotors result in a smoother velocity distribution than the Case 1 rotors. This may have some advantage and could increase the compressor adiabatic efficiency by reduction in flow drag losses.

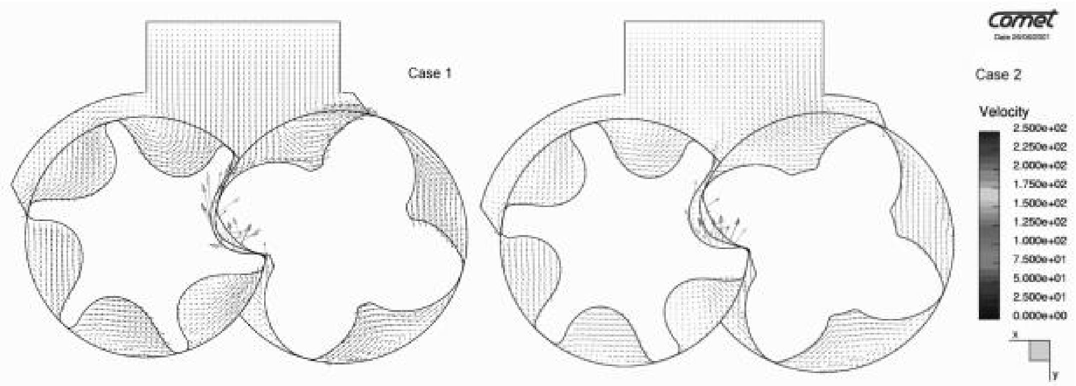


Figure 19 Velocity field in the compressor cross section for Case1 and Case2 rotors

The rise in pressure is similar across the compressor working domains in both cases as presented in Figure 20. However, leakages in the second case are smaller due to the shorter sealing line and consequently the pressure rise is slightly steeper. This results in lower power consumption for Case 2 than for Case 1, despite the higher compressor delivery. These benefits are also visible in the diagram of pressure against angle of rotation where the two cases are compared.

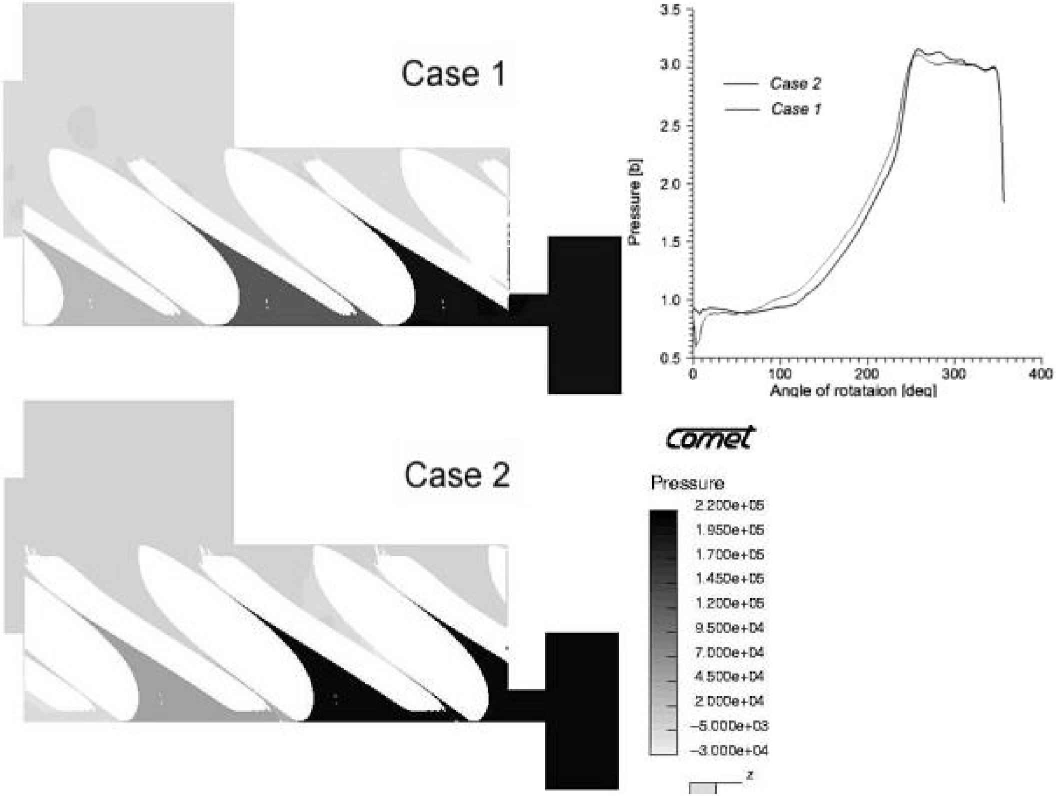


Figure 20 Pressure field in the cross section and comparison of indication diagrams

The pressure differences within the screw compressor working domain create forces on the bearings. The forces on both, the male and the female radial bearings are lower for Case 2 then in Case 1, while the bearings on the discharge side are loaded with almost equal force in both cases. The torque is a function of the compressor suction and discharge pressure, which is according to Figure 20 almost the same. However, the profile shape is significantly different for these two cases and consequently the distribution of torque on the male and female rotors is not the same. In Case 2 the torque on the male rotor is higher then in the Case 1 but the torque transmitted to the female rotor is significantly lower, as shown in Figure 21. Thus the torque on the female rotor is reduced from 10 [Nm] in the Case 1 to practically zero in the Case 2, while at the same time the female rotor is thicker. This gives the Case 2 rotors many advantages. The most important of them are: the lower female rotor deformation caused by the pressure rise, greater flow cross sectional area and a shorter sealing line.

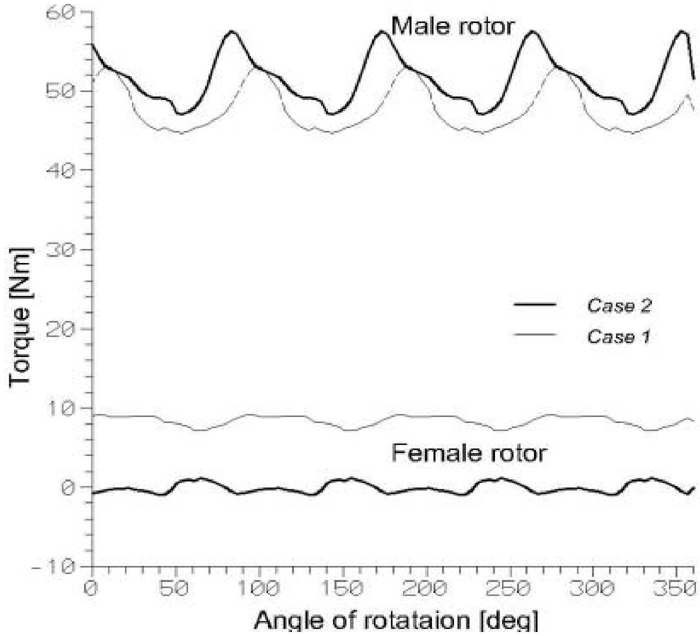


Figure 21 Comparison of the compressor torque in Case 1 and Case 2

All the features listed above, are confirmed by the diagrams, obtained from the 3-D CFD calculations and show that the modern rotors in Case 2 have significant advantages over the more traditional rotors of Case 1. This is also confirmed by comparison of the integral parameters given in Table 1. Here it can be seen that the specific power of the compressor with Case 2 rotors is 12% lower then in the other case. This indicates that both, the consumed power is lower and the delivered flow rate is greater and is confirmed by a 12% higher volumetric efficiency η_v and a 10% higher thermodynamic efficiency η_i for the compressor with the Case 2 rotor profile.

Table 1 Comparison of the integral parameters for the two cases

	$\dot{V}$ [m^3/min]	P [kW]	P_{spec} [kW/m^{3}min]	ηv [%]	ηi [%]
Case 1	**17.285**	**56.757**	**3.283**	**74.371**	**60.960**
Case 2	**19.674**	**55.920**	**2.842**	**83.940**	**67.450**

CONCLUSIONS

A grid generation procedure for a screw compressor working domain has been described in the paper. It is based on the use of pre-specified rack coordinates and a few other parameters to generate a fully structured hexahedral, block oriented numerical mesh for all parts of screw compressor domain. The prerequisite for successful grid generation is the regular initial distribution of points along the boundaries. This is achieved by a two-parameter static adaptation method on one of the boundaries derived specially for screw compressor rotors. The second boundary is mapped from the first one by a transformation methodology, which is also specially derived for the generation of a screw compressor. It allows a regular distribution of boundary nodes in complex domains. These, together with the successful novel application of the combined transfinite interpolation, orthogonalization and smoothing methods allowed fast calculation within a CFD code. An example of its application and the benefits on screw compressor performance, which it can confer, is presented in the paper.

REFERENCES

[1] P.R. Eiseman, J. Hauser, J.F. Thompson, N.P. Waterhill, (Ed.) Numerical Grid Generation in Computational Field Simulation and Related Fields, Proceedings of the 4th International Conference, Pineridge Press, Swansea, Wales, UK (1994)

[2] J.H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, Springer, Berlin (1996)

[3] V.D. Liseikin, Grid generation Methods, Springer-Verlag (1999)

[4] J.E. Thompson, B. Soni, N.P. Weatherrill, Handbook of Grid generation, CRC Press (1999)

[5] A. Kovacevic, Three-Dimensional Numerical Analysis for Flow Prediction in Positive Displacement Screw Machines, Ph.D. Thesis, School of Engineering ad Mathematical Sciences, City University London, UK (2002)

[6] A. Kovacevic, N. Stosic, I.K. Smith, 3-D Numerical Analysis of Screw Compressor Performance, Journal of Computational Methods in Sciences and Engineering, Vol.3, No.2 (2003) 259-284

[7] A. Kovacevic A, N. Stosic, I.K. Smith, Analysis of Screw Compressor Performance by means of 3-Dimensional Numerical Modelling, in: R.L. Elder at all, ed.: Advances of CFD in Fluid Machinery Design, IMechE Proceedings, London, UK (2001) 171-197

[8] N. Stosic N, On Gearing of Helical Screw Compressor Rotors, Journal of Mechanical Engineering Science, Vol.212 (1998) 587

[9] V.D. Liseikin, Algebraic Adaptation Based on Stretching Functions, Russian Journal for Numerical and Analytical Mathematical Modelling, Vol.13, No.4 (1998) 307-324

[10] J.Samareh-Abolhassani, R.E. Smith, A Practical Approach to Algebraic Grid Adaptation, Computers Mathematical Applications, Vol.24, No.5/6 (1992) 69-81

[11] Steinthorsson E, Shih T. I. P, Roelke R. J, Enhancing Control of Grid Distribution In Algebraic Grid Generation, International Journal for Numerical Methods in Fluids, Vol. 15 (1992) 297-311

[12] Kovacevic A, Stosic N, Smith I. K, Grid Aspects of Screw Compressor Flow Calculations, Proceedings of the ASME Advanced Energy Systems Division, Vol. 40 (2000) 83