Unstructured Grid Smoothing for Turbomachinery Applications

Mehdi Falsafioon¹, Sina Arabi¹, Ricardo Camarero¹ and François Guibault²

¹Department of Mechanical Engineering, École Polytechnique de Montréal
CP 6079, succ. Centre-ville, Montréal, QC, H3C 3A7, Canada, mehdi.falsafioon@polymtl.ca, sina.arabi@polymtl.ca, ricardo.camarero@polymtl.ca
²Department of Computer Engineering, École Polytechnique de Montréal
CP 6079, succ. Centre-ville, Montréal, QC, H3C 3A7, Canada, francois.guibault@polymtl.ca

Abstract

In the present study, two mesh smoothing techniques, Laplace and Winslow smoothing techniques, for unstructured grids on turbomachinery application are investigated. These operators are based on the solution of elliptic equations. In the first case, Laplace’s equations are solved using a barycentric averaging procedure. Solution of Winslow’s equations has been a challenging work for unstructured grids because of existence of cross derivative terms in the equations. This issue is addressed devising a local control volume. Both methods are compared using different grid quality criteria. Finally, these operators have been applied to turbomachinery configurations and the advantages and disadvantages are discussed.

Keywords: Grid Smoothing, Unstructured Grid, Winslow’s Equation, Grid quality

1. Introduction

Grid smoothing is a post-processing procedure designed to improve the mesh quality. This can be done with various techniques where the nodal coordinates are modified using an operator such as Laplace, Winslow smoothing [1], or the use of functionals [2]. These have been successfully used for structured meshes. However, elliptic smoothing has been challenging for unstructured grid generation due to the non-conservative form of these equations, as well as the lack of an implied unique computational domain. Knupp [3] applied unstructured Winslow mesh smoothing on unstructured quadrilateral meshes using a locally defined computational domain. Karman [4] and Sahasrabudhe [5] successfully applied unstructured smoothing by introducing a local computational domain called virtual control volumes. Masters [6] extended this to stretched viscous grids. Arabi et al. [7] addressed two major problems concerning the application of elliptic smoothing methods to unstructured meshes. The first is the non-conservative aspect, and the second addresses the mapping procedure by employing an explicit mapping. However, the method is not unique because the physical space is mapped to a square which can be arbitrary for a general unstructured mesh. The solution of the Winslow’s equation on an unstructured mesh was achieved by introducing a 9-point finite difference stencil around each node of the mesh to correctly handle the mixed derivatives.

Following the work of Winslow [1] and Sahasrabudhe [5], in the current study, local computational space along with a finite volume scheme introduced by [7] is used to handle cross derivatives using an averaging process instead of augmented cells. The present article, is an extension to the study presented at the 27th IAHR Symposium on Hydraulic Machinery and Systems [8].

2. Elliptic Smoothing

Elliptic smoothing consists in solving a partial differential equations, where the dependent variables are the coordinates in physical space, \((x,y)\), in terms of independent variables in computational space \((\xi,\eta)\). Two such techniques will be investigated and compared. Laplacian smoothing is the simplest and consists in solving a Laplace operator for \((x,y)\)

\[
\nabla^2 x = 0 \\
\nabla^2 y = 0
\]

(1)

In this work, this will be carried out simply by a barycentric averaging procedure of the coordinates connected to each node of the mesh.
where \( n \) is the number of nodes around the specific node \( i \) of the mesh. This yields smooth meshes for most regular geometries, but invalid meshes can result for configurations with concave corners where large changes of curvature occur. The smoothing procedure can be improved with Winslow’s equations (eq. (1)) in computational space \((\zeta, \eta)\) given as,

\[
L(x) = g_{11} x_{\zeta\zeta} - 2 g_{12} x_{\zeta\eta} + g_{11} x_{\eta\eta} = 0 \\
L(y) = g_{11} y_{\zeta\zeta} - 2 g_{12} y_{\zeta\eta} + g_{11} y_{\eta\eta} = 0
\]  

(2)

where

\[
g_{11} = x_{\zeta}^2 + y_{\eta}^2 \\
g_{12} = x_{\zeta} x_{\eta} + y_{\zeta} y_{\eta} \\
g_{22} = x_{\eta}^2 + y_{\zeta}^2
\]  

(3)

While Winslow’s operator guarantees continuum global mapping [3, 9], truncation errors can lead to folded meshes. In such instances, additional control is needed to adapt the mesh around the boundaries to insure the validity of the results, especially around discontinuous parts of the physical boundary.

A method based on Taylor series expansion to solve this operator on equilateral triangles, where all the angles are equal to \( \pi/3 \), has been proposed in [3]. Another method is presented in [4], based on generating a virtual control volume in the physical domain locally around each node, as a local computational space with the same number of neighboring nodes as in physical space. Arabi et al. [7] compared two techniques for the solution of the functional operator [2] using finite volume and finite difference methods. The latter based on inserting a 9-point finite difference stencil presented several sharp corners. In the current work, Winslow’s equation is solved using a local computational space [4] combined with the averaging method presented by [7] for evaluating of cross-derivatives.

2.1 Computational Space

Elliptic equations enforce a smoothness condition in physical space \((x, y)\) through the use of a regular computational domain. In structured meshes, the computational domain has the same topology as the physical space, and ideal spacing i.e \( \delta x = \delta y \), thus effectively enforcing an equal spacing around each node. For an unstructured mesh, there is no such a universal computational domain that matches every unstructured topology; therefore, for each unstructured mesh a computational domain must be devised to match the topology of the physical mesh. A solution to this problem was proposed by Sahasrabudhe with the introduction of computational domains that are only defined locally [10]. For a stencil of elements surrounding a single node, a regular mesh can be defined by equally spacing the connected points on a unit circle. These stencils, called virtual control volumes, are locally defined and can be used to drive the solution of the smoothing equations. Thus formulated, smoothing by the use of elliptic operators consists in solving a distinct boundary value problem with dirichlet boundary conditions for each node.

2.2 Finite Volume Scheme

The Winslow’s operator is in non-conservative form and the three coefficients, \( g_{11}, g_{12} \) and \( g_{22} \) in eq. 2 are functions of the gradients of the dependent variables in the computational space. The coefficients are frozen during each iteration. Using a linearization procedure, the various terms of eq. 3 can be integrated separately over a control volume defined around each point of the mesh in computational space. The integration path for the application of Green’s theorem is formed by joining the centroid of each triangular element to the midpoints of its sides, as shown by the dashed lines in Fig. 1. The cell edges divide each triangular element into three equal areas which form non-overlapping contiguous control volumes associated with a node in the mesh. The hashed region in Fig. 1 indicates a control volume with a centroid node which is the storage location of all dependent variables. Using integral form of eq. 2 and linearizing this equation results in the following form
Fig. 1 Local mapping from physical to virtual computational space

(a)

(b)

\[ g_{11} \int x_{x\zeta} d\Gamma - 2 g_{12} \int x_{x\eta} d\Gamma + g_{22} \int x_{y\eta} d\Gamma = 0 \]

\[ g_{11} \int y_{x\zeta} d\Gamma - 2 g_{12} \int y_{x\eta} d\Gamma + g_{22} \int y_{y\eta} d\Gamma = 0 \]

Applying the divergence theorem to the second order derivative terms, \( x_{x\zeta} \) and \( x_{y\eta} \), gives

\[ \int \int x_{x\zeta} d\Gamma = \int \int \nabla F d\Gamma \]

where the components of function \( F(x_\zeta, 0) \). Similarly, for the cross derivative terms, for \( x_{y\eta} \) we have

\[ \int \int x_{y\eta} d\Gamma = \int \int \nabla Q d\Gamma \]

While mathematically \( x_{x\zeta} = x_{x\zeta} \) for continuous functions, numerically these integration procedures yield different results depending on the order of the integration. In [4] these terms were calculated using a set of augmented cells attached to the control volumes. Therefore, the shape of virtual control volumes is different for the cross derivative terms. In [7], it was found that taking an average by splitting the cross derivatives in two components and applying the Green’s theorem in both directions avoids the generation of folded cells in the physical domain. In addition, this averaging procedure maintains the symmetry of the final mesh when the deformation of the physical boundary is symmetric. Ref. [7] concluded that for the cross derivative terms, applying Green’s theorem only for one component on each control volume side around node \( (\zeta_i, \eta_i) \) yields a degenerated final mesh in most cases of geometries with severe boundary curvature variations. In other words, for arbitrary deformations in the \((x, y)\) plane, the values of the calculated fluxes in \((\zeta_i, \eta_i)\) are dominated by the values from the cross derivatives terms. Moreover, taking only one component of the cross derivative term after applying the Green’s theorem, deflects the final mesh in one direction.

Based on this, the term \( Q \) is computed in two manners. Using \( Q_1 = (0, x_\eta) \) and \( Q_2 = (x_\zeta, 0) \), gives \( Q \) as an average of these two

\[ Q = \frac{1}{2} (Q_1 + Q_2) \]

Integrating over the control volume and applying the divergence theorem for each dependent variable, for example \( \zeta \), gives

\[ \int \int \nabla Q d\Gamma = \oint F \cdot \hat{n} dS \]

(5)

The term on the RHS integral represents the net flux through the surface of the volume and, for the Winslow’s operator, can be evaluated as

\[ g_{11} \int x_{x\zeta} n_x dS - 2 g_{12} \int 1/2 (\int x_{x\eta} n_x dS + \int x_{y\eta} n_y dS) + g_{22} \int x_{y\eta} n_y dS = 0 \]

(6)

These line integrals are approximated by a summation over the sides of a polygon forming a co-volume around each node. For the first term, this gives,
\[ g_{11} \int x_{\xi} n_{x} dS = g_{11} \sum_{j} \int x_{\xi} n_{x} dS \approx g_{11} \sum_{j} (x_{\xi} n_{x} dS)_{sidej} \]

where \( x_{\xi} \) and \( n_{x} \) are evaluated at the center of \( sidej \) with length \( S \). Replacing the line integrals in eqs. 6 by this summation, yields,

\[ (g_{11} - g_{12}) \sum_{j} (x_{\xi} n_{x} dS)_{sidej} + (g_{22} - g_{12}) \sum_{j} (y_{\xi} n_{y} dS)_{sidej} = 0 \]

\[ (g_{11} - g_{12}) \sum_{j} (y_{\xi} n_{y} dS)_{sidej} + (g_{22} - g_{12}) \sum_{j} (y_{\xi} n_{y} dS)_{sidej} = 0 \]

(7)

The tangent vector along \( sidej \) is \((\Delta \xi, \Delta \eta)\) and the normal vector is \((\Delta \eta, -\Delta \xi)\), then the unit normal vector is

\[ \vec{n} = (n_\xi, n_\eta) = (\Delta \eta, -\Delta \xi) / S \]

Where \( S = \sqrt{\Delta \xi^2 + \Delta \eta^2} \) is the length of the \( sidej \). Substituting into eqs. 7 gives,

\[ (g_{11} - g_{12}) \sum_{j} (x_\xi \Delta \eta) |_{sidej} - (g_{22} - g_{12}) \sum_{j} (y_\xi \Delta \xi) |_{sidej} = 0 \]

\[ (g_{11} - g_{12}) \sum_{j} (y_\xi \Delta \eta) |_{sidej} - (g_{22} - g_{12}) \sum_{j} (y_\xi \Delta \xi) |_{sidej} = 0 \]

(8)

2.3 Discretization of Equations

In the present work, a node centered method is employed for the finite volume scheme. For each element, moving in a counter-clockwise direction, nodes are labeled 1, 2 and 3. Values of the dependent variable \( x \) and \( y \) are calculated and stored at these points.

Evaluation of the terms in eq. 8 requires \( \nabla x \), \( \nabla y \) and \( n \), the normal vector to the \( sidej \), as shown in Fig. 1(b). In this way, values at an arbitrary point within the element can be approximated with piecewise linear interpolation

\[ \phi(\xi, \eta) \approx a\xi + b\eta + c, \quad \phi = x, y \]

(9)

where the constant coefficients \( a, b \) and \( c \) satisfy the nodal relationships

\[ \phi_i(\xi, \eta) \approx a\xi_i + b\eta_i + c, \quad i = 1, 2, 3 \quad \text{and} \quad \phi = x, y \]

(10)

such that, over the element, the continuous unknown field can be expressed as the linear combination of the values at nodes \( i = 1, 2, 3 \)

\[ \phi_i(\xi, \eta) \approx \sum_{i=1} F_i(\xi_i, \eta_i) \phi_i, \quad \phi = x, y \]

(11)

where \( F1, F2 \) and \( F3 \) are the shape factors of each certain element and are respectively \( a, b \) and \( c \) in eq. 11, then

\[ \nabla \phi(\xi, \eta) = (\phi_x, \phi_y) = (a, b) \]

(12)

The next step is converting the integral form of the governing equations (eqs.4) into an algebraic discrete form. Terms of node distribution throughout the physical space are as follow

1. The control volumes and all the neighbour nodes and edges attached to each node \( i \) are identified; this requires an appropriate data structure which is explained in the next section;
2. Using numerical integration and the shape function approximations of the values \( (x \) and \( y \) \) in each element sharing node \( j \), eq. 4 is rewritten in terms of the nodal values of \( x \) and \( y \).
3. After accumulating terms, the resulting equation for node \( i \) can be expanded in the general discrete form

\[ a_i \phi_i = \sum_{nb} a_{nb} \phi_{nb}, \quad \phi = x, y \]

(13)

where \( a_i \) and \( a_{nb} \) are coefficients for the unknown nodal values of \( \phi \). This equation provides an algebraic relationship between the values of \( \phi \) at the node \( i \) and the neighboring nodes in its vicinity.

3. Data Structure

In order to automate the process through the use of a computer code, one needs to provide a data structure that can identify and store the mesh components associated with every single point of the domain. In addition to this, a data structure is required to update certain properties of the grid such as edge and node properties and boundary conditions. Based on this, the data structure covers used in the present study, covers following information

1. Labeling of the nodes (\( i = 1, 2, \ldots, n \)) where \( n \) is the number of nodes in the domain;
2. Vector of the nodal coordinates \((x \) and \( y \));
3. Numbering of the triangles in the domain and the node of the vertices of these triangles in a counter-clockwise order;
4. List of nodes, edges and elements in the vicinity of each node in a counter-clockwise order;
5. List of the nodes on the boundaries of the domain.
The data structure is edge-based with four components: nodes, edges, elements and boundaries that are structured around an edge as shown in Fig. 2. Each edge is an oriented topological segment from node $N_1$ to $N_2$ shared by the left and right elements, $E_L$ and $E_R$, and left and right nodes, $N_L$ and $N_R$. The link to the geometry is through a reference of $N_i$ to $x(N_i)$ and $y(N_i)$, the physical location. The link to the computational domain is a reference to an internal segment or a boundary segment.

![Fig. 2 A typical configuration of the inlet of a guide vane](image)

4. Mesh Quality and Mesh Smoothness

Several criteria can be used to measure the quality of a mesh such as minimum angle, maximum angle, minimum Jacobian and shape factor. We have used minimum angle and shape factor as mesh quality criteria. The minimum angle criterion means that elements with small angles are considered to be of a worse quality than ones with larger angles. The shape factor criterion measures the likeness of an element to a reference equilateral triangle as

$$SQ_i = \frac{4\sqrt{3}A_i}{\sum_{j=1}^{3}l_i^2}$$

where $A_i$ is the area of the triangle, and $l_i$ ($i = 1, 2, 3$) are the lengths of the triangle’s edges. However, in order to devise a mesh mapping appropriate for arbitrary boundary shapes in the physical domain, in addition to positive Jacobian for all cells, other measurable criteria must also be considered. In contrast to the traditional measure of mesh quality, which usually considers individual criteria of each element, smoothness of a mesh has a global definition. Thus, these two distinct measures of mesh quality and mesh smoothness may be contradictory for some cases. Indeed, a smoother mesh does not necessarily imply better mesh quality as we are going to show in this study.

The smoothness criterion was introduced by [7] and will be used to compare meshes resulting from the two smoothing techniques investigated in this study. The mesh smoothness is quantified for each cell as

$$SR_i = \frac{A_i}{\max(A_n)}$$

where $SR_i$ represents the smoothness ratio, $A_i$ is the area of cell $i$ and the denominator represents the maximum area of its adjacent cells. An ideal value for $SR_i$ is as close as possible to one.

The Smoothness Factor ($SF$) of a mesh, was defined in [7] as follows,

$$SF = \frac{1}{N_e} \sum_{i=1}^{N} \min(SR_i, \frac{1}{SR_i})$$

where $N_e$ is the total number of elements in the mesh. The range of values for this factor is $0 < SF \leq 1$, and hence, the greater $SF$, the smoother the mesh.

5. Results and Discussion

The two smoothing techniques were implemented and investigated in the context of the preceding framework for their effectiveness. Numerous test cases for different geometries and grid sizes were performed. Two representative examples, a sharp spike and a slit inside a circle, will be used to illustrate the results. The effect of both smoothing techniques on an initial raw mesh, generated using a frontal unstructured grid procedure, is shown in Fig. 3. These qualitative results are quantified in Figs. 4 to 6 for the shape measure, minimal angle and smoothness ratio, respectively.

As can be seen from Fig. 4, the barycentric method gives a more satisfactory distribution for the shape factor. This is also the case for the minimum angle criterion (Fig. 5), as the smallest angle for the barycentric method is around 40°, while for Winslow’s method this value is around 25°. However, from smoothness point of view, it is clear that the Winslow’s operator gives better results than the barycentric method. Table 1 shows that the smoothness factor for Winslow’s method is higher than that resulting from the barycentric method.
Another test was carried out to investigate the behavior of each method starting from an invalid mesh shown in Fig. 7(a). Applying the two smoothing techniques shows that both are capable to yield a valid mesh. However, the grid obtained by Winslow’s method is not identical to that obtained from an initial valid mesh, indicating the dependence of Winslow’s method on the initial mesh. Another critical configuration is a slit with several sharp corners inside a circle (Fig. 8). Results show better mesh quality for the barycentric method but smoothness remains better for the Winslow’s method. However, for a configuration with sharp curvature(s) barycentric fails to give a valid grid after smoothing process (Fig. 9). This is the greatest disadvantage of the barycentric method versus Winslow’s operator in a grid smoothing procedure.

In order to show the grid smoothing ability of these procedures for turbomachinery applications, a grid is generated for geometry similar to the inlet of typical guide vanes shown in Fig. 10. The grid is generated using a Delaunay triangulation and contains 18895 nodes for a quarter of the entire geometry. The barycentric method is much faster than Winslow’s equation giving respectively 0.3955s and 2.0014s. Figure 11 shows the results for both methods compared with the raw grid. As the quantitative comparisons show (Fig. 12), the Winslow’s operator has a better effect on the smoothing process either from minimum angle point of view or smoothness. Although the barycentric method has more elements with angles close to π/3 (the ideal element angle) but Winslow’s operator globally gives better result regarding to a global minimum angle criterion.

![Fig. 3 Comparison of two smoothing methods with the raw mesh](image-url)

![Fig. 4 Comparison of mesh quality using shape factor criterion](image-url)

### Table 1 Comparison of global Smoothness Factor based on different mapping operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Spike (coarse grid)</th>
<th>Slit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Mesh</td>
<td>0.860</td>
<td>0.865</td>
</tr>
<tr>
<td>Barycentric</td>
<td>0.921</td>
<td>0.913</td>
</tr>
<tr>
<td>Winslow</td>
<td>0.955</td>
<td>0.948</td>
</tr>
</tbody>
</table>
Fig. 5 Comparison of mesh quality using minimum angle criterion

6. Conclusion

Two elliptic mesh smoothing methods, barycentric and Winslow, have been compared for 2D unstructured grids. An averaging method along with the local computational space method was employed to introduce a new discretization scheme for cross derivative terms of Winslow’s equation. Both methods show reasonable results concerning grid smoothness and grid quality criteria. From mesh quality point of view, the barycentric method almost always gives equal or better results than the Winslow’s equation. However, Winslow’s operator always shows better results for smoothness. Our investigation shows that final results based on the Winslow’s operator are dependent on the initial mesh. The barycentric technique can be employed as a pre-smoother and/or initial guess for the Winslow’s smoothing method which always gives valid grid even for configurations with sharp concave curvatures. Moreover, since the Winslow’s operator respects the positive Jacobian for all the cells in the physical domain, they can be used for many applications such as geometry optimization or fluid-structure interaction problems as well. These two smoothing methods can be extended to three dimensional cases.

Fig. 6 Comparison of mesh smoothness (SR)

Fig. 7 Investigation of the behavior of two smoothing methods for an invalid mesh
Fig. 8 Investigation of the behavior of two smoothing methods for an invalid mesh. Comparison of mesh quality and mesh smoothness for a slit inside a circle.
**Fig. 9** The barycentric method fails to smooth a configuration with sharp curvature(s)

**Fig. 10** A typical configuration of the inlet of a guide vane
Fig. 11 Smoothed grid for the inlet of a 2D guide vanes configuration
References


