Journal of
Applied & Computational Mathematics
ISSN: 2168-9679

Karabelas, J Appl Computat Math 2015, 4:4
DOI: 10.4172/2168-9679.1000234

Research Article Open Access

Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method

Karabelas SJ*

Institute of Aerospace Engineering, Mechanical Engineering, Brno University of Technology, Czech Republic

Abstract

A novel Cartesian grid discretization method is developed to simulate two and three dimensional problems, governed by partial differential equations. In the present approach, the grid points lie exactly onto the surface of an immersed object or of the domain's boundaries, allowing for accurate imposition of the surface boundary conditions. This is well demonstrated for symmetric objects, but it can be also extended for non-symmetric shapes. The method intrinsically possesses higher accuracy than the conventional body fitted or Immersed Boundary Methods, since the implemented grid is universally orthogonal and the boundary conditions are imposed precisely onto the surface, without any interpolation or tuning of the governing equations. Conformal Cartesian Grids bridge the topology of Cartesian grid methods with the treatment of the surface boundary conditions, which is adopted in conventional body-fitted grid approaches. Emphasis is given in a two dimensional fluid dynamics problem to demonstrate this approach. A finite difference code has been developed, which encompasses the present methodology. Space discretization is performed via the second order accurate central difference scheme and time discretization by the fourth order accurate Runge-Kutta method. The flow past a cylinder at low Reynolds number is resolved to validate the accuracy and performance of the method. Two different flow regimes are thoroughly investigated at Re numbers varying from 10 up to 100 based on the cylinder's diameter. Computed results agree well with the available measurements and numerical computations in literature. Three dimensional results are also briefly presented mainly for revealing the applicability of the method.

Keywords: Cartesian grid methods; Immersed boundary methods; Cylinder flow; Conformal; Finite-difference method

Introduction

Conventional strategies developed so far to simulate physical problems evolving differential equations, could be summarized in two main categories based on the grid topology and conformity with the domain boundaries: The first considers grids, which conform onto the boundaries of the computational domain (immersed objects, surrounding walls and others). Those methods could implement structured or unstructured grids of different topology, thus encompassing tetrahedral, hexahedral, prisms, polyhedral or other elements. The second category considers computational grids, which have Cartesian topology through the whole domain, but the grid does not necessarily conform to the boundaries of the domain. Those are well known as "Cartesian Grid Methods" or "Immersed Boundary methods" and they have been developed mainly the last three decades [1].

Cartesian grid methods started to develop by Peskin [2], who mainly attributed the term "Immersed Boundary methods" to this methodology. Later on, a considerable number of approaches followed, which were mainly designed for simulation of in viscid flows past complex solid boundaries [3]. A highlighted study of Euler flow simulations by using those methods is addressed in the article of Berger and Aftosmis [4]. The latter methods enjoyed universal acknowledgement and they are still developed by many researchers. Nevertheless, viscous flows resolved by Immersed Boundary Methods arise many questions regarding the accuracy and the consistency of the numerical approach. The basic complications are obviously related to the imposition of the boundary condition in the non-conformal grid. The main basic approaches include continuous forcing, discrete forcing, cut-cell formulations and general interpolation algorithms [1,5]. Those methodologies are developed and improved continuously. Some recent modifications with validated results are the studies of Lima E Silva et al. [1] and Rajani et al. [6]. Complex and moving boundaries could be also resolved with distinct advantages over the conventional methods [1,7].

Despite the general effort on reducing the uncertainties regarding the imposition of the boundary condition on the surface of complex geometries, it appears that none method could guarantee validity through the whole range of Reynolds numbers and for all types of flow regimes. Until the time, a universal method will be developed; always uncertainties will arise regarding every individual approach.

The preceding statements possibly motivated researchers to develop more elaborated methods, which could retain the orthogonality of the mesh as much as possible and impose precisely the boundary condition onto the immersed surface. This concept is well described in the paper of Fujimoto et al. [8], where the term "Body Fitted Cartesian Method" is referenced. In this research effort, an orthogonal Cartesian grid was constructed from the external boundaries of the domain up to an area close to the immersed surface. The Cartesian grid contains hanging nodes to adapt accordingly to the contour of the immersed object. Then a prismatic layer grid is generated to resolve the boundary layer up to the object's walls. The interface between the orthogonal grid and the prisms is non-conformal in this study (Figure 1). In a more recent study of Fujimoto et al. [9], the discontinuous interface is removed by deploying a Cartesian grid front projection. By implementing this grid

***Corresponding author:** Karabelas SJ, Institute of Aerospace Engineering, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2, 616 69 Brno, Czech Republic, Tel: 42 0773667938; E-mail: stkarabelas@gmail.com

Received July 02, 2015; **Accepted** July 13, 2015; **Published** July 20, 2015

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

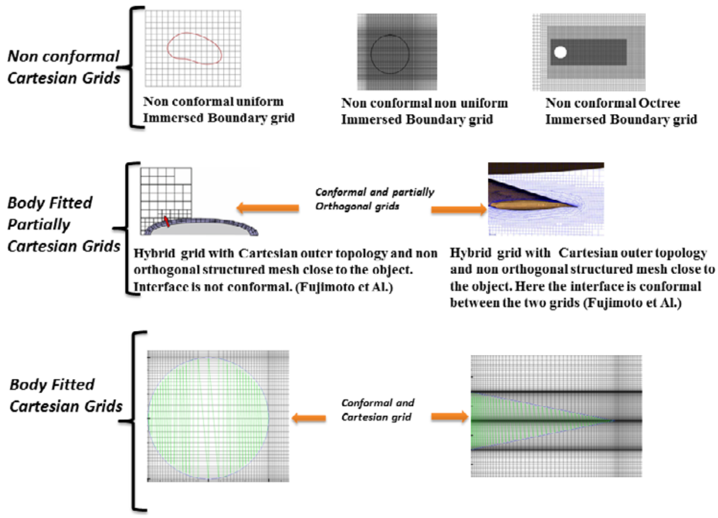

Figure 1: State of the art of the main Cartesian grid methods developed so far. The present method is depicted at the bottom of the picture.

projection, conformity is preserved between the boundary layer mesh and the orthogonal outer mesh (Figure 1). The boundary layer mesh consists of hexahedral non-orthogonal elements.

The latter research surveys could be viewed as interesting variations of the Cartesian Grid Methods, which could be more accurate than the Immersed Boundary Methods, specifically through the boundary layer area. However, those grids are not universally orthogonal. They cease to be orthogonal from the interface up to the immersed surfaces. Also they contain some "transitional" cells of trapezoidal or tetrahedral topology, which could further deteriorate the accuracy of the method. Those methods based on the author's consideration should not be entitled as "Body Fitted Cartesian", because the grid is definitely not Cartesian up to the surface. In order to distinguish the approach presented here with the latter grid methods, the term "Conformal Cartesian Grids" is utilized.

In the present research effort a novel method is introduced, which comes to fill the gap between the two main aforementioned strategies. The method of Conformal Cartesian Grids retains the Cartesian topology and the orthogonality of the grid in the whole computational domain up to the immersed surface, while in the same time allows the boundary condition to be imposed precisely onto the object's boundaries, without any forcing or interpolation. Those meshes are body fitted and clearly inherit properties from the conformal body-fitted grids and the Cartesian grid methods. All the preceding methods and the present one are demonstrated in a simplified evolution graph in Figure 1 to reveal the state of the art in conformal and orthogonal grids.

The current paper will describe in detail the grid generation algorithm of the present approach. A thorough discussion follows regarding the accuracy and performance, by addressing advantages and deficiencies of the method. Focus is given on the numerical validation by solving a well-known fluid dynamics benchmarking problem. The numerical implementation for resolving the physical problem is thoroughly discussed. Results are compared to available numerical and experimental data.

Description of the Method

The concept of conformal cartesian grid generation for symmetric objects

Before demonstrating numerically how the Conformal Cartesian Grids (CCG) method performs, a clear description of the grid generation process is presented. The basic motivation for implementing those methods in PDE problems is extracted from the observation, that it is possible and simple to construct a body fitted Cartesian grid past a two dimensional symmetric body. The symmetry axis could be the axial, vertical or both directions. To present the simplicity of the approach, consider a curve implicitly defined as follows:

$$f(x, y) = 0 \tag{1}$$

with the following properties:

1. Being a closed curve

2. Being a Jordan curve and

3. Possessing symmetry along the x or y or both axes simultaneously, thus $f(-x, y) = 0$ or $f(x, -y) = 0$ or $f(x, -y) = f(-x, y) = f(-x, -y) = 0$ correspondingly. Then, it is always possible to construct a Conformal Cartesian Grid.

The proof of this statement could be easily deduced by observing graphically how the Cartesian grid could be fitted exactly to the surface of the symmetric body. In the left top part of Figure 2 the example of an object which possesses symmetry along x axis is demonstrated. Starting arbitrarily from a point marked as A, it is projected via a vertical line to the lower symmetric surface generating point B. Then, a second projection of the latter point follows by extending a *y*=constant line creating point C. Then C is projected similarly to D. The latter point is connected to the origin point A, finalizing one cycle of the process. If this procedure repeats for other origin points onto the surface of the current shape, the result will be the right top grid plotted in Figure 2. It is evidently shown that the grid is fitted exactly onto the surface and then it is only needed extension of the grid lines towards the boundaries of the computational domain to complete the generation of the grid.

The final extended grid is shown in the bottom area of Figure 2,

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

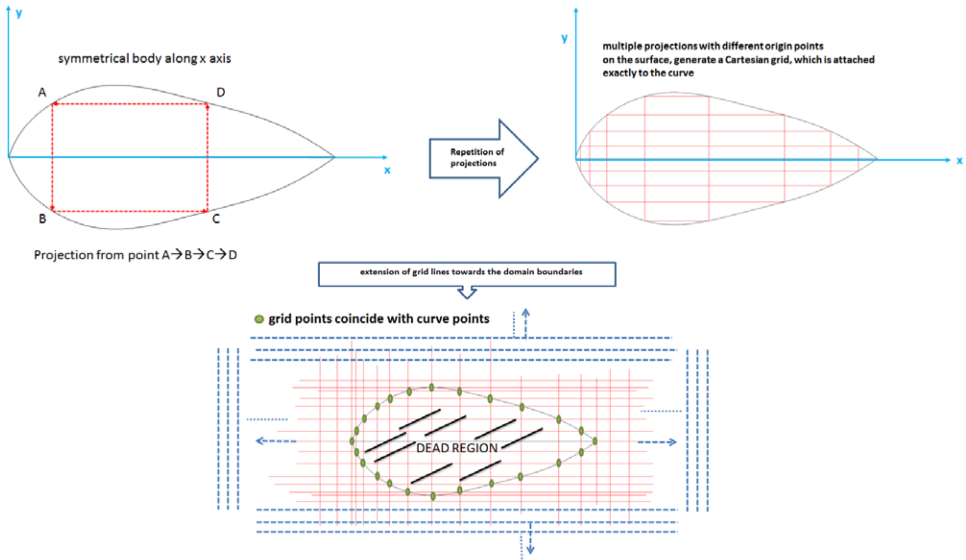

Figure 2: Conceptual description of the grid generation process. At the top left, one cycle of the process is completed. At the top right, a number of cycles have been finished and at the bottom area, the final extended mesh is shown.

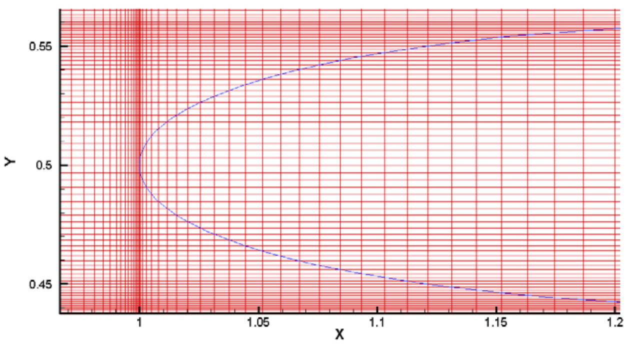

Figure 3: Conformal Cartesian grid in the vicinity of a NACA 0012 airfoil. The points of the grid coincide with the airfoil's profile.

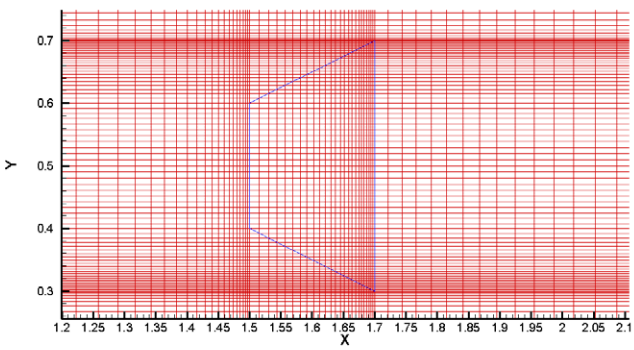

Figure 4: Conformal Cartesian grid in the vicinity of a trapezoid.

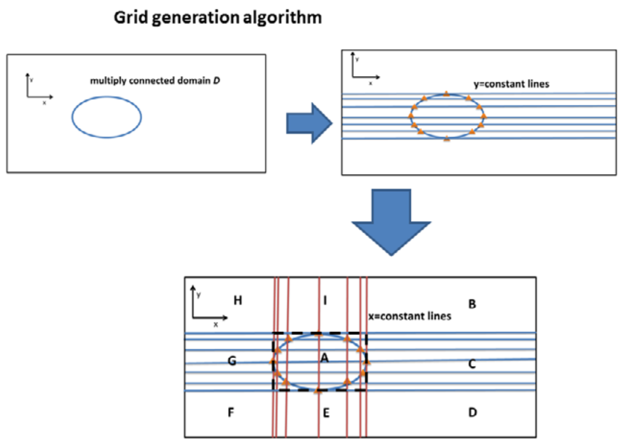

Figure 5: Steps of the grid generation algorithm, as programmed in the present study.

where also the grid points on the surface are marked. The dash lines represent the extension of grid lines. Dead zone is the area of grid points, which do not interfere in the numerical solution. This is an example of a body-fitted Cartesian grid. Other examples of a NACA 0012 airfoil and a trapezoid grid are plotted in Figures 3 and 4 respectively. In areas, where non uniform refined grids should be generated, two algorithms are implemented, the $\tanh$ function and geometric progression [10]. Refinement is observed close to the area of the immersed objects.

Description of a grid generation algorithm

The above grid generation procedure gives the perception of how a Cartesian grid could conform precisely onto the boundaries. In the present section, the algorithm of grid generation is described for a two dimensional problem. The author followed the process, which is depicted in Figure 5 and it is believed that is quite simple and straightforward.

Given a multiply connected domain (top left side of Figure 5), we would like to generate a conformal Cartesian grid past the immersed object. In the beginning of the procedure, horizontal lines are drawn, which intersect the object and the domain's boundaries. The distribution of those lines is better to be symmetrical (to be equivalently distributed below and upper of the midline of the object), as it is shown at the top

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

right area of Figure 5. After the intersections of those lines with the object, the coordinates of the knots are calculated and stored (orange points). Then by starting from each knot, vertical lines are drawn. These lines are depicted at the bottom of the same figure and they are marked as red. At this stage the essential procedure has finished. The only remaining action is to extend the mesh left, right, below and upper of the object. In sub domains B, D, F and H, it is possible to generate a mesh with arbitrary distribution of the grid lines at x and y direction. In sub domains E and I, the x distribution is given by the previous steps of the algorithm, therefore only at y direction the mesh could be freely generated. In sub domains G and C, the y distribution is already prescribed, therefore only at x direction the mesh could be freely generated. Finally in the sub domain A, the circumscribed rectangular (marked with the dash line), the discretization has been already completed and no further action could be applied. This is exactly one of the distinguishing features of the method, which is not found in the other grid discretization methods: The mesh inside the circumscribed rectangular of the immersed body is generated based only on the distribution of the object points.

Grid generation for 2-D arbitrary shaped objects

The rationale presented so far, can be also extended for non-symmetric bodies. However, this could be the main topic for future research effort in CCG methods. The present paper, deals only with a brief example. In non-symmetric bodies, the projection of the points from one part of the surface to another is not straightforward as in symmetric cases. The formulation of the mathematical problem could be summarized in the graph shown in Figure 6. The black line separates the object into two parts. Then the upper and lower curves are assigned to the functions f(x(s) and g(x(s)) respectively, where s denotes the index of the vertical grid lines. The question now is how we can fulfill simultaneously in both sub domains with given x(s) the following condition:

$$\left\{\begin{matrix} y = f(x(s)) : y \in D1 \\ y = g(x(s)) : y \in D2 \end{matrix}\right\}$$

This is a difficult and novel mathematical problem and any further analysis extends beyond the scope of the present article.

The author devised a simple graphical method, which could give a preliminary feedback whether it is possible or not to construct a Cartesian grid, which is body fitted to 2-d non-symmetric shapes.

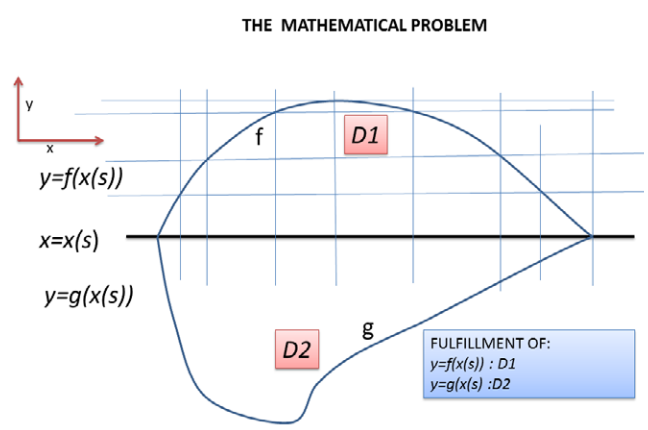

Figure 6: Mathematical formulation of the 2-d non symmetric problem.

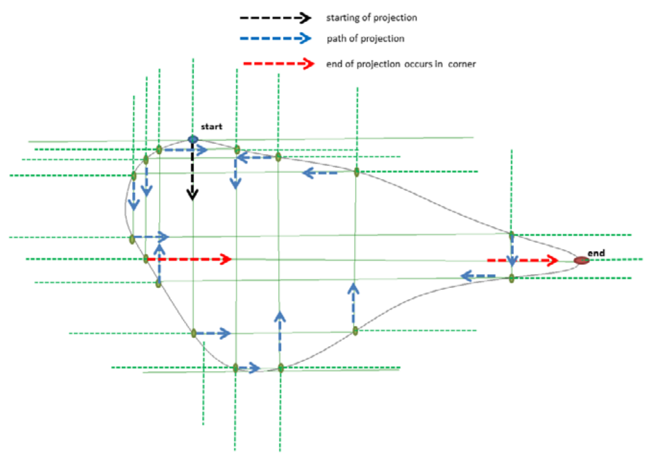

Figure 7: Illustrative description of an example of grid generation process for a non-symmetric object.

It appears that if the shape fulfills the three main properties already defined for the symmetric problem above, then it is possible to construct a grid, but not certain, since this statement needs a concrete mathematical proof. In Figure 7 it is shown a non-symmetric object and one complete step of the projection-intersection process. Starting arbitrarily from the blue marked point (starting of projection), a projection is done to the lower surface via ax = const. line. Then the new point is projected via a y=const. line towards the right part of the surface. By continuing projections of the points to the left and right parts of the surface (see blue arrows), multiple points are marked on the surface, which fulfill the criteria of a body-fitted Cartesian grid. However, the process yet has not been completed. It was observed by repeating efforts, that the process is possible to be finalized if the projection will end to a maximum, minimum or saddle point. This is the case in Figure 7. The finalization occurs at a local extreme point (marked as red) and there is no need to project any more that point to the surface. By starting from different origin points, it is possible to construct a conformal Cartesian grid of high resolution.

The present graphical method for non-symmetric bodies provides nor any information regarding the grid resolution neither whether this process can be finalized each time. It basically motivates for further research and gives a preliminary idea of grid construction feasibility.

Grid generation for three dimensional symmetric objects

The generation of conformal Cartesian grids extends to three dimensional problems. The simplest one could be just a straight extrusion of a 2-D object towards the normal direction. This mesh can be easily generated and one solved example is shown in section 5. The more complex case for 3-d symmetric objects is considered when the object is not generated by straight extrusion of a 2-D symmetrical object, but by a non rectilinear profile. This case could be the shape of a sphere (Figure 8).

In any three dimensional Cartesian grid, each slice of the mesh is exactly the same with all the other slices. This property is important to be evaluated during all the steps of the grid generation. By viewing a sphere, it is easily observed that the sections at different heights (red curves) are not the same. Therefore it is expected that the mesh slices will not be the same at those heights. Indeed, if someone follows the 2-d method presented earlier to construct the mesh at each slice, then the mesh slices will be completely different and a 3-d construction appears not to be feasible.

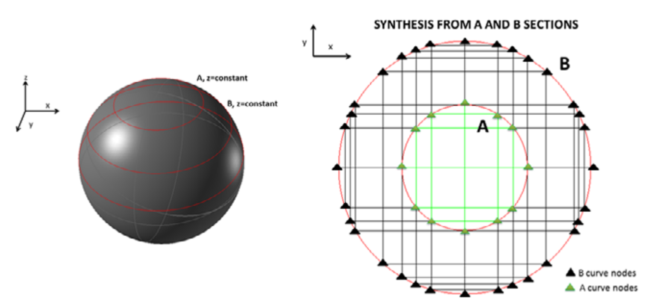

Figure 8: Sections of a sphere illustrated with red lines (left) and synthesis of the two upper sections (right).

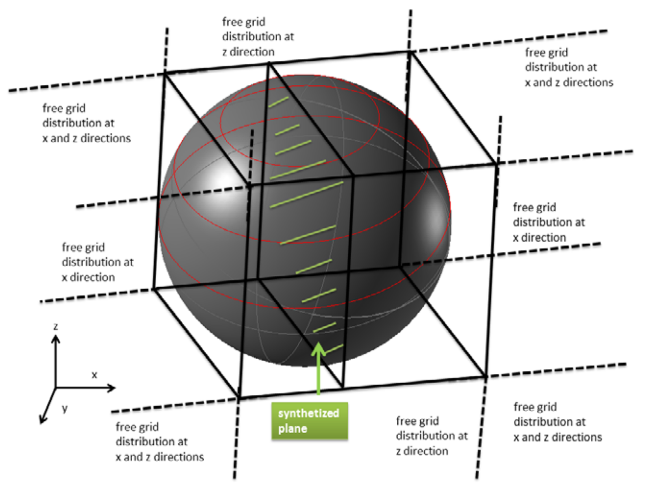

Figure 9: Synthesized plane depicted inside the circumscribed cube and some written restrictions regarding the distribution of the grid lines in the areas outside the cube.

A different path could be followed to construct the 3-d mesh. If the slices of an object are different, then all the grid lines from each slice should be projected onto the other slices. When the projection finishes and the intersections of all sections are completed then the slice mesh is derived. In the present case to be comprehensive, only two different planes are considered. In Figure 8, synthesis of those two planes, produce the final slice mesh. It is shown that the synthesized slice intersects accurately both planes and the projections have finished. This process continues until all sections of the object are considered (the number of the sections depends upon the initial choice). After the synthesis of all projections, a final slice is produced (a complex 2-d grid). This slice is the unique section of the 3-d mesh inside the circumscribed cube (Figure 9) and this area is meshed only based on the object points distribution (as in the 2-d case with the domain inside the circumscribed rectangular). Out of the cube, the mesh can be generated with some restrictions, as in the 2-d case. In Figure 9, some of those restrictions are shown.

Added value of the present method

The present method combines the effectiveness of the Cartesian methods and the accuracy of the body-fitted conventional grids. Below, this will become clear by stating the advantages and disadvantages of the approach. CCG methods retain most of the advantages sourced from the Cartesian grid topology. Here, they will be briefly quoted, but the reader could be directed to the literature resources [1,5] for more insight. One advantage is the relative ease of generating the grid past an object with an axis of symmetry, but still slightly more complicated than generating Immersed Boundary grids (Figure 2). Compared to the body-fitted structured and unstructured mesh topologies, the conformal Cartesian grids are generated much faster.

In terms of memory, Cartesian topology allows for efficient indexing of all node variables structured in an I x J x K form. This structure enhances the convergence of the iterative algorithms, since it generates n-diagonal matrices. When the computer power is considered, then it is clear that a Cartesian grid can significantly reduce the per-grid point computing time, because of the lower number of discretized terms in the governing equations [11]. Cartesian grids are also amenable to efficient computing algorithms. Line-iterative and multi-grid algorithms can be always used to reduce the CPU time and increases convergence. This is not easy to be implemented in unstructured grids of any topology (polyhedral, tetrahedral, hexahedral and others).

Accuracy and performance of the method becomes obvious when the imposition of the boundary conditions is considered. In the present approach, the boundary conditions are imposed precisely onto the object's surface as in the conventional body fitted grids. This property evidently increases the accuracy and enhances the simplicity of programming. On the contrary, the developed Cartesian grid methods deploy interpolation or continuous forcing close to the immersed boundaries (Figure 10a), which could deteriorate the accuracy of the solution, especially at high *Re* numbers. This could be explained by the fact that forcing introduces additive truncation error in the equations and specifically in sensitive flow areas. The alternative approach of interpolation, utilizes ghost points or cut-cell techniques, which raise criticism regarding feasibility of discretization, when high order schemes are considered. This is well verified in literature, since the latter methods have been used extensively for Euler flows, but still there

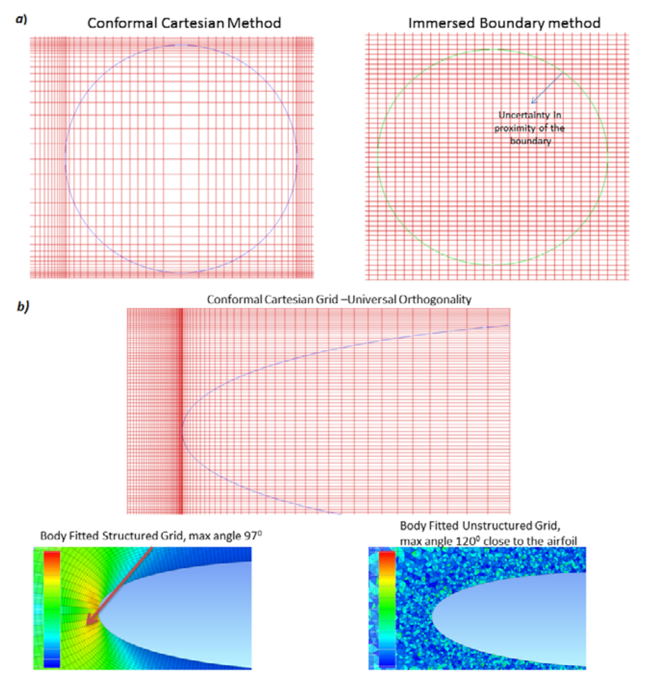

Figure 10: a) A Conformal Cartesian and an Immersed Boundary grid in the proximity of a circle. b) Orthogonality computed based on the maximum cell's angle for an O-type grid and an unstructured grid around an airfoil.

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

is a big challenge to be effective in viscous flow regimes [1,4,5] and in turbulent regimes [12].

Conformal Cartesian Grids are orthogonal in the whole computational domain, while in all the other mesh topologies, this is not feasible (Figure 10b). This intrinsic property enhances the accuracy of the method. Indeed, to compute the convective fluxes with the highest possible accuracy you need to connect with a line the two neighboring cell centers through the mid of the face. The same holds for the calculation of the diffusive fluxes with the requirement that the line should be orthogonal to the cells face [11]. In a Cartesian grid, those requirements are fulfilled. The above statements could become clear mathematically, when it is considered the generic conservation equation for momentum transport of a variable φ written in generalized coordinates ξ_i :

$$J\frac{\partial(\rho\varphi)}{\partial t}+\frac{\partial}{\partial\xi_i}[\rho U_i\varphi-\frac{1}{J}(\frac{\partial\varphi}{\partial\xi_k}B^{ki})]=0 \text{ and } i=1,2,3 \qquad (2)$$

$J=\det(\frac{\partial x_i}{\partial\xi_\xi})$ is the Jacobian of the transformation from the Cartesian coordinates x_i to the generalized ones ξ_i. U_i denotes the velocities normal to the coordinate surface ξ_i = constant. The remaining coefficients B^{ki} are expressions of the cofactors of the Jacobian.

These coefficients are zero in orthogonal grids when $k\neq i$. In eq. (2) when the grid is non-orthogonal, six extra terms arise in the 3-D Navier-Stokes equations and four extra terms in the two dimensional case. These terms constitute the truncation error (*T.E.*) due to the mesh non-orthogonality:

$$E=T.E.(A^{21}\frac{\partial^2\varphi}{\partial\xi_2\partial\xi_1}+A^{31}\frac{\partial^2\varphi}{\partial\xi_3\partial\xi_1}+A^{12}\frac{\partial^2\varphi}{\partial\xi_1\partial\xi_2}+A^{32}\frac{\partial^2\varphi}{\partial\xi_3\partial\xi_2}+A^{13}\frac{\partial^2\varphi}{\partial\xi_1\partial\xi_3}+A^{23}\frac{\partial^2\varphi}{\partial\xi_2\partial\xi_3}) \qquad (3)$$

The same conclusion could be reached by adopting the finite volume approach. Evaluating the fluxes in a 2-d non-orthogonal grid, leads to two components of the surface normal vector n_i at face *i*, while in Cartesian grids there is only one component. The extra components when multiplied by gradients of the solving variable (diffusive fluxes) increase further the truncation error.

The penalty for possessing universal orthogonality and imposing the boundary conditions precisely onto the immersed surface is the rigorous dependency of the grid upon the object's shape (contour). This is evident in Figure 11, which plots the left side of a circular cylinder. In conventional grid structured methods, it is possible to control the points in the vicinity of the object independently for each marching direction (ξ, η). This might be also the case for the IB methods, where the grid could be independent of the surface curvature. On the contrary, for the present case, the distribution of the points in *x* direction is correlated always to the distribution of the points in *y* direction close to the object. This is imperative in order to attain coincidence of the grid points to the surface. For example in the latter plot, close to the left corner, where curvature reduces, it is needed to generate a lot of points along *x* direction to follow the steep difference along *y* direction.

This drawback brings other deficiencies. Shape dependency enforces for generation of highly bunched grids, which sometimes could have influence on the convergence and accuracy of the method. It is observed, by employing such grids that at surfaces almost horizontal, they fail to resolve adequately the contour of the surface, unless quite high aspect ratio cells are generated. Therefore mentioned observations are well reproduced also in Figure 3 independency of the discretization process between the two marching directions is attained

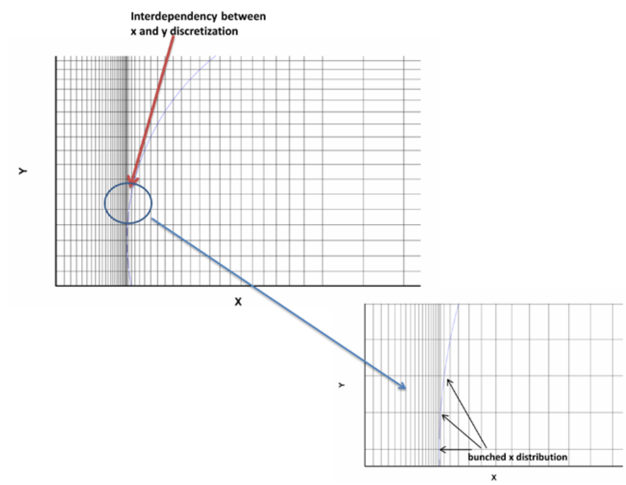

Figure 11: The mesh close to the left side of a cylinder reveals the interdependency between the discretization along x and y directions.

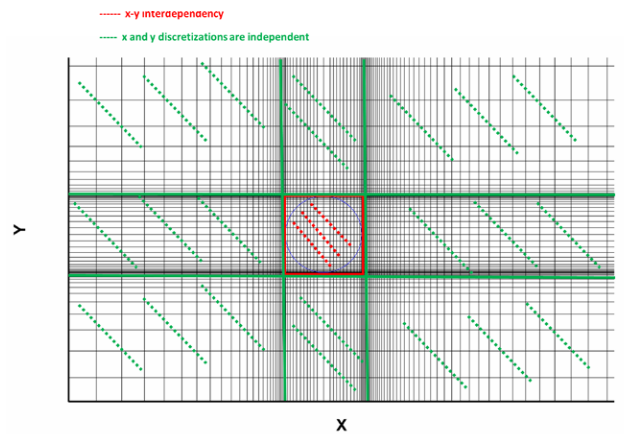

Figure 12: Borders showing where the distribution of grid points along one direction, is dependent upon the distribution of grid points along the other direction.

far from the object, as it is depicted in Figure 12. The red border lines embrace the "dependent discretization" area and the green lines mark the "independent discretization" regions. This drawback could be mitigated by implementing Adaptive Mesh Refinement strategies. One strategy, which is planned to be encompassed in forthcoming improvements of the CCG method, is described in the next section.

Adaptive mesh refinement in conformal cartesian grids

The methods for adaptive mesh refinement (AMR), which are implemented in general Cartesian grids, could be also adopted in Conformal Cartesian Grids [11]. However, special care should be taken into regions close to the surface. To avoid the overly refined grid lines, which are mainly generated from nearly horizontal or vertical surfaces, they should be split close to the object, in order to keep the aspect ratio of cells in acceptable levels. In Figure 13, those horizontal lines are marked with light blue color and they begin from the top area of the surface and they are split a few cells away. This splitting does not refine the grid, yet it generates a qualitative mesh with low aspect ratio cells by introducing hanging nodes within the stencils. In second level, refinement close to the surface could be performed by adding surface nodes and extending the grid lines close to the surface. These grid lines are marked as green in Figure 13. Although, the error could be reduced by such AMR strategies, various problems arise, such as the ordering of the hanging nodes and the solution of the Navier-Stokes equations at the interface (red border in Figure 13). Finite-Difference methods should be combined with Finite- Volume methods at the interface area. Conservation of mass, momentum and energy could be assured only by implementation of Finite-Volume methods at irregular cells with many sub-faces (marked also in the same figure with magenta lines). Finite-difference methods could still apply to cells with normal structured stencil.

Numerical Formulation for Solving Two Dimensional Flows

To implement numerically the method in applied physics, it is chosen the investigation and validation of the two dimensional flow past a cylinder. This chapter discusses analytically the governing equations and the mathematical formulation of the PDE problem.

Governing equations and algorithms

A classic benchmarking problem for Cartesian grids is the flow past a circular cylinder. This is a challenging case, since the cylinder always possesses curvature which amplifies the difficulty for resolving the flow with Cartesian grid methods. In the present study the two dimensional flow past a circular cylinder is investigated. The Reynolds number based on the cylinder diameter ranges from 10 up to 100. In this range two flow regimes are observed. Up to *Re*=47.4 based on Norberg [13], the flow remains steady and separated from the cylinder's surface. At higher *Re* number, vortex shedding develops and the flow becomes unsteady. Three dimensional effects are observed at higher than the examined *Re* numbers.

Viscous and incompressible flow past the circular cylinder is governed by the two dimensional Navier- Stokes equations. Those equations are solved in a non-primitive formulation by adopting the vorticity stream-function approach [10,11]. By combining the conservation equations for continuity and momentum, the following two equations are derived:

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=-\omega \quad (4)$$

$$\rho\frac{\partial\omega}{\partial t}+\rho u\frac{\partial\omega}{\partial x}+\rho v\frac{\partial\omega}{\partial y}=\mu\left(\frac{\partial^2\omega}{\partial x^2}+\frac{\partial^2\omega}{\partial y^2}\right) \quad (5)$$

where it holds $u=\frac{\partial\psi}{\partial y}$, $v=-\frac{\partial\psi}{\partial x}$ and $\omega=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}$. The variable ψ denotes the stream function, u the axial velocity, v the vertical velocity, $\omega$the vorticity around z axis, $\mu$the molecular viscosity and ρ the density of the fluid. When the steady-state version of the code is implemented (*Re*<47.4) the time derivative term is dropped from eq. (5). In this version under-relaxation of vorticity has been embedded within the solving procedure. In the unsteady full version of the code three time marching algorithms are available, namely Euler- explicit scheme, Runge-Kutta and Gauss-Seidel implicit iterative solver. From those algorithms, it is chosen the Runge-Kutta method, because it is fourth order accurate in time $O(\Delta t^4)$. Below there is a short mathematical description of the method:

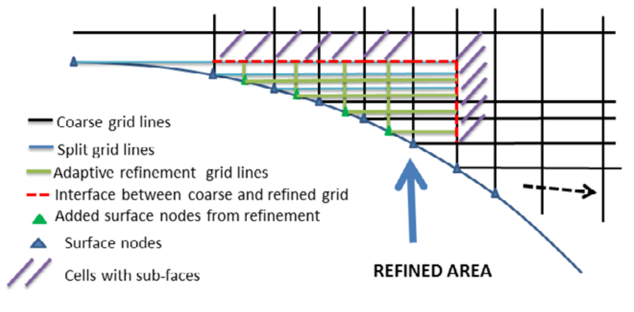

Figure 13: Adaptive refinement of a conformal Cartesian grid in the vicinity of an object.

$$\dot{\omega}=R(\mathrm{t},\omega) \quad (6)$$

$$R(t,\omega)=-\rho u^t\frac{\partial\omega^t}{\partial x}-\rho v^t\frac{\partial\omega^t}{\partial y}+\mu\left(\frac{\partial^2\omega^t}{\partial x^2}+\frac{\partial^2\omega^t}{\partial y^2}\right) \quad (7)$$

$$\omega^{t+dt}=\omega^t+\frac{1}{6}dt(\alpha+2b+2c+d) \quad (8)$$

The slopes *a, b, c and d are given by the expressions:*

$$\begin{aligned} a&=R(t,\omega)\\ b&=R(t+0.5dt,\omega+0.5dta)\\ \mathrm{c}&=R(t+0.5dt,\omega+0.5dtb)\\ \mathrm{d}&=R(t+dt,\omega+dtc)\end{aligned} \quad (9)$$

The terms in the governing equations are discretized in space based on the central differencing scheme of second order accuracy.

Derivatives discretization

The first derivative terms could be discretized with second order accuracy schemes independent of the gird points distribution (grid bunching). This is possible, by incorporating the value of the unknown function at the current node *i*. Then one could derive for the central differencing scheme, the following leading truncation error:

$$e=\frac{dxdx'}{6}\frac{\partial^3 f}{\partial x^3} \quad (10)$$

where $dx=x_{i+1}-x_i$ and $dx'=x_i-x_{i-1}$

and the final expression for the first derivative is:

$$\frac{\partial f}{\partial x}=\frac{f_{i+1}dx'^2-f_{i-1}dx^2+f_i(dx^2-dx'^2)}{dxdx'(dx+dx')} \quad (11)$$

which is always of second order.

Upwind and downwind differencing schemes could be treated in the same way and could be of second order spatial accuracy. The latter schemes provide a solution, when high order accuracy is desired onto the boundary, since they are one-sided and penetration towards the dead region is avoided.

Second order derivatives are difficult to be independent of the grid bunching, when second order accuracy is considered, because in Taylor expansions first and third order derivatives should be eliminated simultaneously. In the implemented code, the truncation error of central scheme depends upon the stretch ratio *a*, defined here as $a=\frac{dx}{dx'}$ and it is equal to $e=\frac{(1-a)dx'}{3(a+1)}\frac{\partial^3 f}{\partial x^3}$, thus when the grid discretization is uniform (a =1), $e=0$ and the method becomes second order accurate. Upwind and downwind schemes are also influenced by

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

the grid bunching, when second order accuracy is desired.

Spatial convergence rate of the algorithm

To verify mathematically that the spatial accuracy of the method is of second order, repetitive simulations have to be be performed with different levels of global mesh refinement and then the results should be compared to an analytical solution. In the present case, it was chosen the laminar flow inside two parallel plates, which possesses an analytical solution [10]. In this physical problem, with chosen *Re=10*, the flow develops fast and then a constant parabolic profile of the axial velocity is preserved through the whole channel. Four different meshes are tested with different resolution: 20 × 20, 50 × 50, 100 × 100 and 200 × 200 cells at *x* and *y* direction. Then the L1 and L2 norms of the global error are evaluated by considering the difference between the numerical and the analytical value of velocity. The results from those tests are plotted in Figure 14, revealing the convergence of the velocity profile as the refinement increases. The global error is plotted logarithmically in Figure 15, which obviously reveals that the method is second-order accurate.

Initial and boundary conditions

Before proceeding to the numerical results, it would be useful to describe in detail the numerical treatment of the boundary conditions, which are implemented into the mathematical problem. The domain and the applied conditions onto the boundaries are depicted in Figure 16. At the inlet of the domain a constant velocity profile is imposed and at the outlet a zero stream-wise velocity gradient condition. A zero cross-flow velocity gradient condition is applied into the upper and lower boundaries to further mitigate any blockage effects. Onto the

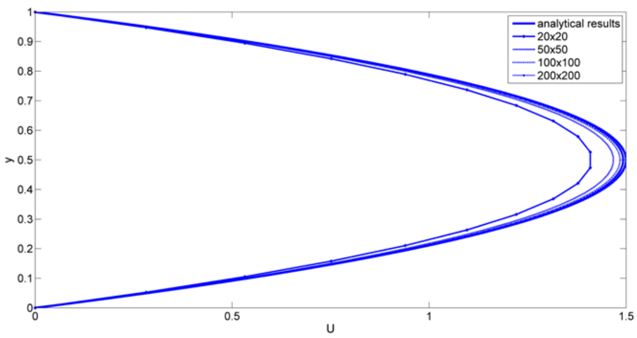

Figure 14: Axial velocity profiles derived from the mesh tests and compared to the analytical solution.

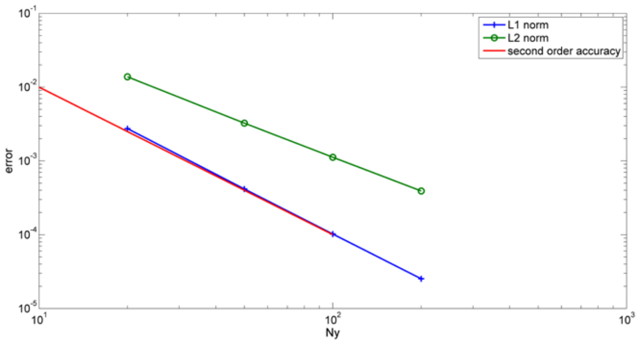

Figure 15: L-1 and L-2 error norms, plotted logarithmically against the number of cells at y direction.

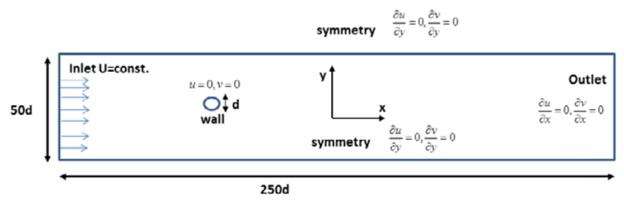

Figure 16: Domain's boundaries and dimensions. The type of boundary conditions is also shown.

cylinder the no-slip condition is imposed. Although those conditions are clear when using a pressure-velocity coupling procedure, in the vorticity stream-function approach, needs to be expressed in terms of the solving variables.

The initial conditions, when no interpolation or no other initial solution is incorporated, are the following ones:

t=0

$$\left.\begin{aligned}\omega &= 0\\ \psi &= 0\end{aligned}\right\} x, y \in \Omega \tag{12}$$

$$\left.\begin{aligned}\omega &= f(\psi)\\ \psi &= \psi(\mathrm{x}, \mathrm{y})\end{aligned}\right\} x, y \in B \tag{13}$$

where Ω is the internal computational domain and *B* the boundaries of the domain (inlet, outlet, symmetry and wall conditions). At those boundaries, the vorticity is expressed as a function of the stream function to satisfy the corresponding velocity condition.

The inlet condition for a constant axial velocity profile in time and space u, could be imposed by the following expressions for vorticity and stream-function at the node (i, j):

$$\psi = \int_{inlet} u dy \tag{14}$$

$$\omega(i,j) = 2(\psi(i,j) - \psi(i+1,j))(\Delta x)^{-2} + \frac{2(\Delta x)}{3!}\frac{\partial^3 \psi}{\partial x^3} + \ldots \tag{15}$$

The outlet condition similarly is expressed as:

$$\frac{\partial^2 \psi}{\partial x^2} = 0 \tag{16}$$

$$\frac{\partial \omega}{\partial x} = 0 \tag{17}$$

The symmetry or zero cross-flow gradient condition is simply expressed as:

$$\left.\begin{aligned}\psi &= const\\ \omega &= 0\end{aligned}\right\} symmetry \tag{18}$$

At object's walls, the conditions become more complicated and here they will be described in a more general content. Along the surface of the object it holds $\frac{\partial \psi}{\partial x} = 0$ and $\frac{\partial \psi}{\partial y} = 0$, thus $\psi = const$. For deriving wall's vorticity we define two vectors $\vec{x}$ and $\vec{x}_k$. The former points to a position inside the flow n and the latter to the object's surface. The origin of the vectors is always the reference frame base point (Figure 17):

$$\begin{aligned}\vec{x} &= x_i i + y_i j\\ \vec{x}_k &= x_k i + y_k j\end{aligned} \tag{19}$$

The symbols x_i, y_i denote the coordinates of a flow point and x_k, y_k the coordinates of the node onto the object's curve.

It is also defined the L2 norm of the vectors difference as:

$\Delta s = \left|\vec{x} - \vec{x}_k\right|$, where the sign depends upon the convention for the positiveness of the s direction.

By expanding Taylor series for the stream function, the following equation holds:

$$\psi(\mathrm{x}_i, \mathrm{y}_i) = \psi(\mathrm{x}_k, \mathrm{y}_k) + (\Delta s)\frac{\partial \psi(\mathrm{x}_k, \mathrm{y}_k)}{\partial s} + \frac{(\Delta s)^2}{2!}\frac{\partial^2 \psi(\mathrm{x}_k, \mathrm{y}_k)}{\partial s^2} + \frac{(\Delta s)^3}{3!}\frac{\partial^3 \psi(\mathrm{x}_k, \mathrm{y}_k)}{\partial s^3} + \ldots \quad (20)$$

In Conformal Cartesian Grids Δs is always Δx or Δy, thus the third term in the right hand side is equal to $-\frac{(\Delta s)^2}{2!}\omega$. Based on the definition of vorticity and the no slip condition, which should apply onto the wall's surface, the following expression results:

$$\omega(\mathrm{x}_k, \mathrm{y}_k) = \frac{2(\psi(\mathrm{x}_k, \mathrm{y}_k) - \psi(\mathrm{x}_i, \mathrm{y}_i))}{(\Delta s)^2} + \frac{2(\Delta s)}{3!}\frac{\partial^3 \psi}{\partial s^3} + \ldots \quad (21)$$

The well-known first order expression for the wall's vorticity can be derived from the last equation:

$$\omega(\mathrm{x}_k, \mathrm{y}_k) = \frac{2(\psi(\mathrm{x}_k, \mathrm{y}_k) - \psi(\mathrm{x}_i, \mathrm{y}_i))}{(\Delta s)^2} + O(\Delta s) \quad (22)$$

The preceding boundary condition performs well for vertical or horizontal walls, where the vorticity onto the surface is influenced only from the next grid point at the vertical or the horizontal direction. However, this is not the case, when the wall has curvature and the vorticity onto the surface is influenced by two grid points, which lie at the x and y direction respectively. An alternative is to deploy a Taylor expansion at x and y direction separately, and in second step to add them. For a surface node, which is wetted from the top and right area it holds after the addition of the two Taylor expansions:

$$\omega(\mathrm{x}_k, \mathrm{y}_k) = \psi(\mathrm{x}_k, \mathrm{y}_k)\left(\frac{\Delta x^2 + \Delta y^2}{0.5\Delta x^2 \Delta y^2}\right) - (\Delta x^2(\psi(\mathrm{x}_k, \mathrm{y}_{k+1}) + \Delta y^2(\psi(\mathrm{x}_{k+1}, \mathrm{y}_k)) / (0.5\Delta x^2 \Delta y^2) + O(\Delta x, \Delta y) \quad (23)$$

The author found the latter mixed spatial boundary condition more accurate, than considering only one direction. For instance, solving the flow close to the leading edge of the cylinder and expanding the Taylor series across the vertical direction, produces significant error. Indeed, tests proved that when the pressure distribution is evaluated, the mixed boundary condition (23) produced excellent pressure profiles, while eq. (22) failed considerably close to the leading edge.

Pressure derivation

Static pressure can't be computed during the solving process, but it should be explicitly derived from the Poisson pressure-velocity equation:

$$\nabla^2 p = 2\rho\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right) \quad (24)$$

Equation (24) is solved in the same manner with eq. (4) by implementing a Gauss-Seidel solver.

The wall boundary conditions for pressure can be formulated in the general case as:

$$\frac{\partial p}{\partial \tau} = -\mu\frac{\partial z}{\partial n} \quad (25)$$

$$\frac{\partial z}{\partial n} = \nabla z \cdot \vec{n} \quad (26)$$

Where n is the direction normal to the surface, τ the direction tangent to the surface, $\vec{n}$ the normal unit vector acting along n direction and μ the molecular viscosity.

The final boundary condition is reformulated based on the two previous equations as:

$$\frac{\partial p}{\partial \tau} = -\mu\nabla z \cdot \vec{n} \quad (27)$$

Cartesian derivatives of z should be derived to give the final pressure derivative across the surface. This boundary condition allows finding all surface pressure values, unless one value is known onto the object.

Numerical Validation of the 2-D Method

Feature parameters

The CCG method is applied to resolve the flow past a cylinder at *Re* number from 10 up to 100. In this range of *Re* numbers, two main flow regimes develop. Up to approximately *Re*=47.4 based on Norberg[13], the flow exhibits a steady-state behavior and separates from the cylinder's surface. Beyond this first instability point, the flow becomes unsteady and oscillating, separating also from the cylinder's surface, revealing the well-known vortex street. For consistency and clarity of the numerical validation, those two regimes will be examined separately, demonstrating clearly the performance of the method. In the present analysis the following basic parameters evolve:

The *Re* number is defined as:

$$\mathrm{Re} = \frac{\rho U d}{\mu} \quad (28)$$

where d denotes the cylinder's diameter and U the free-stream speed.

The dimensionless time is defined as:

$$t^* = \frac{tU}{d} \quad (29)$$

The drag coefficient is expressed as:

$$C_d = \frac{\int_s P \cdot n_x ds}{\frac{1}{2}\rho U^2 d} \quad (30)$$

and the lift coefficient as:

$$= \frac{\int_s P \cdot n_y ds}{\frac{1}{2}\rho U^2 d} \quad (31)$$

Where P denotes the static pressure, n_x the unit vector acting towards the x direction, n_y the unit vector acting towards the y direction and S the object's surface.

Steady separated laminar flow past a cylinder

Grid independency: In this flow regime, computations are performed at *Re*=10, 20, 30 and 40. The computational domain has dimensions 250d × 50d (Figure 16). To reduce any blockage effects, the ratio of the cylinder's diameter to the domain's width, namely blockage ratio, it is chosen to be equal to B=0.02 [14]. Zero cross-flow velocity gradient boundary conditions apply to the top and bottom boundaries, further mitigating the blockage effects. Mesh independency tests are performed to obtain converged results and to be assured that the final

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

mesh produce reliable solutions. Four grids are tested with the same fraction of cells discretizing the cylinder (24% of the total cells in y direction and 12% in x direction) and the same minimum grid spacing offset of the cylinder's surface. The difference lies on the total number of cells generated along x and y direction. The grids have total number of cells starting from the coarser one: 201 × 101, 301 × 151, 401 × 201 and 501 × 251. The criterion for convergence adopted here is the magnitude of the drag coefficient, as commonly used in other studies [1,14]. Table 1 reports all values of the drag coefficient. At higher resolutions than 401 × 201 the results converge quite well. The latter mesh appears to be a good compromise between accuracy and computational cost and it is implemented at all *Re* number calculations. Figure 18 plots the final mesh close to the cylinder.

Results and flow topology: Figure 19 plots the flow pattern at each *Re* number in this regime. As the *Re* increases the bubble's length in the wake area increases and the flow separates earlier from the cylinder's surface. Those results are in general agreement with previous computations and experiments [1,6,15]. The pressure distribution versus angle θ is plotted in Figures 20 and 21 at *Re*=30 and *Re*=40 (left side). The convention for angle θ is illustrated in Figure 22. Results are in good agreement with the experiment of Grove et al. [16] and the computations of Park et al. [17]. The computations of Lima E Silva et al. [1] deviate from the cited results and the present ones along the pressure side of the cylinder. In the latter study, the Immersed Boundary Method is implemented with a uniform grid.

Integral loads are computed and compared with literature data (Figure 21). Lift coefficient is found to be zero at this regime, revealing a completely symmetric flow. Drag coefficient agrees very well with the numerical results of Sen et al. [14]. Both numerical data lie between the experimental drag curves of Triton [18,19].

Length of the bubble in the cylinder's wake is also compared to available literature data. This length is defined as the distance of two stagnation points behind the cylinder (*Lw* in Figure 23). In the present study, it is determined based on the change of the axial velocity's sign. Figure 16 compares the present results with the experiment of Taneda [15] and the numerical simulations of Lima E Silva et al. [1] and Park et al. [17]. The present results agree very well with the experiment of Taneda [15].

Figure 24 plots the separation angle as a function of the *Re* number. The angle is measured based on the convention adopted in Figure 22. The point, where the flow separates can be detected by the condition $\tau_w = 0$. After that point the wall shear stress τ_w changes sign and the flow recirculates. In this plot, the present results agree well with results of other authors [14,20]. All simulations differ slightly from the experiment of Countanceau and Bouard [21].

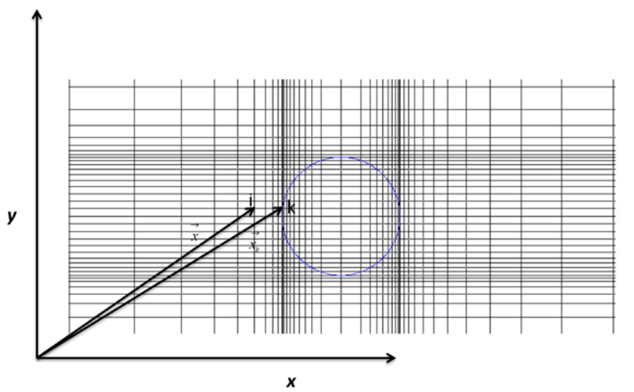

Figure 17: Vectors used for computing vorticity on the object's boundaries. The vector   ends to a random fluid point in this plot, but usually directs to a point next to the solid surface.

Re=10	
Non uniform grids with minimum dy=0.005d	
Grid size	**Cd**
201 × 101	2.6
301 × 151	2.76
401 × 201	2.85
501 × 251	2.88

Table 1: Mesh independency study.

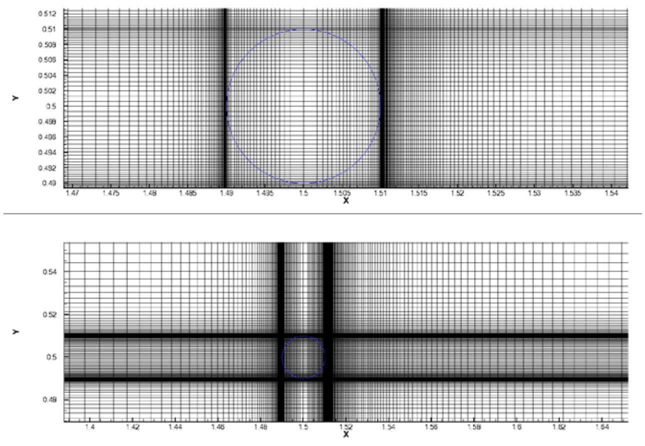

Figure 18: Conformal Cartesian mesh consists of 401 x 201 cells, resolving the flow past the cylinder. The top plot is a zoomed view and the bottom a far-field view.

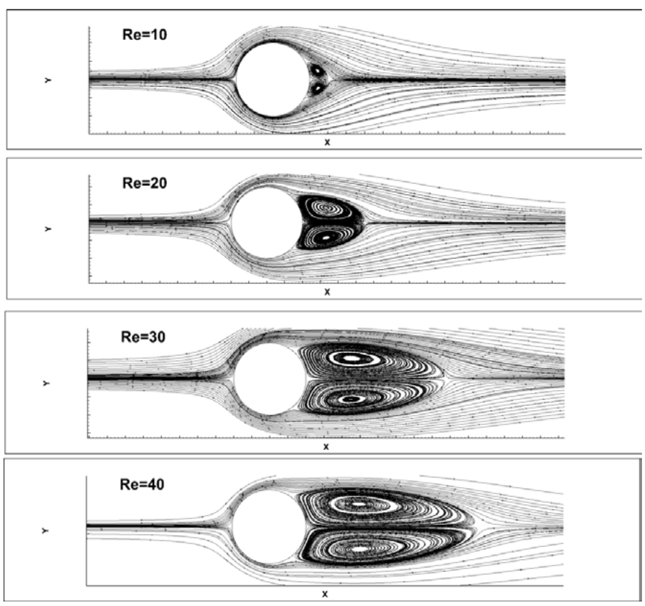

Figure 19: Flow patterns illustrated by streamlines at each examined Re number for the steady flow regime.

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

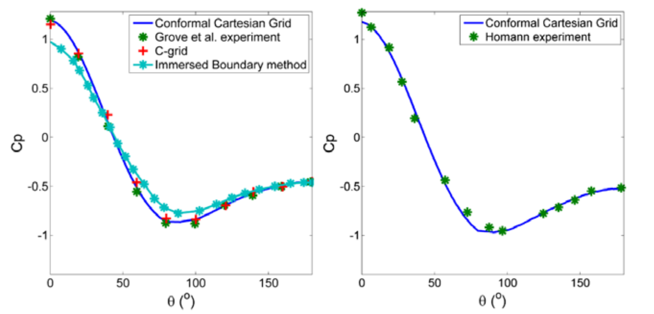

Figure 20: Pressure distribution along the cylinder's perimeter. At the right graph, Re=30 and at the left Re=40.

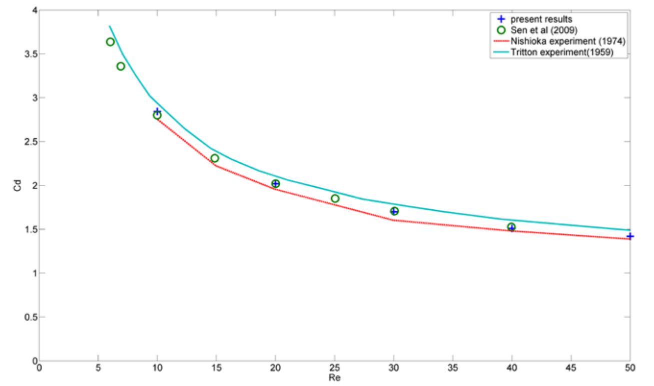

Figure 21: Drag coefficient versus Re number for the steady flow regime.

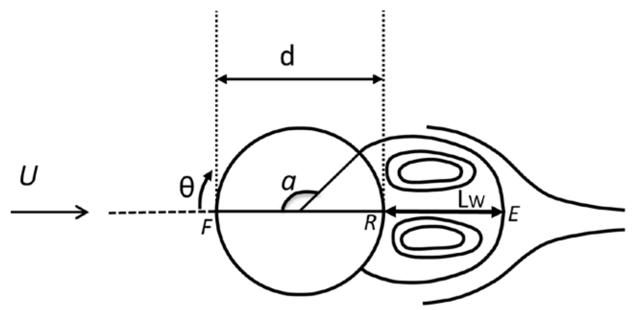

Figure 22: Schematic illustration of conventions for angle θ and bubble's length.

In Nishioka and Sato experiment [19], wake velocity measurements were performed close and further from the cylinder area. Figure 25 plots the dimensionless velocity versus $\frac{y}{d}$, at $\frac{x}{d}=3$ downstream of the cylinder's center. The present computational results generally coincide with the measurements. Small deviations of the order 3-5% are observed in the middle line and at $\frac{y}{d}\geq 3$.

Unsteady laminar flow past a cylinder

Unsteady flow is resolved at *Re*=50, 60, 80 and 100. In this range, the flow is still two dimensional and unsteady with wake instabilities, exhibiting downstream of the cylinder the Von Karman vortex street.

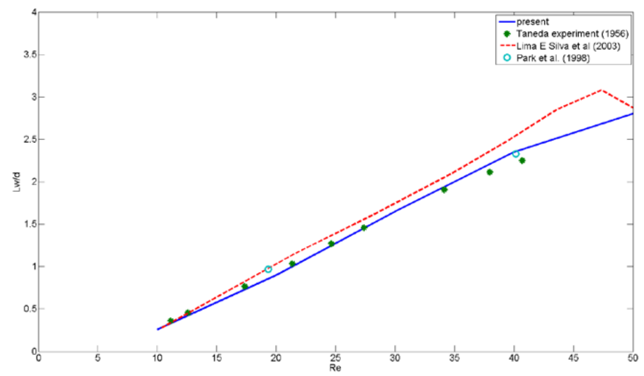

Figure 23: Wake bubble's length as a function of the Reynolds number for the steady flow regime.

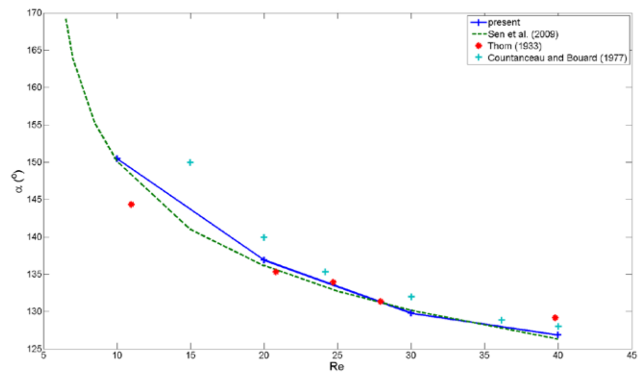

Figure 24: Separation angle as a function of the Reynolds number for the steady flow regime.

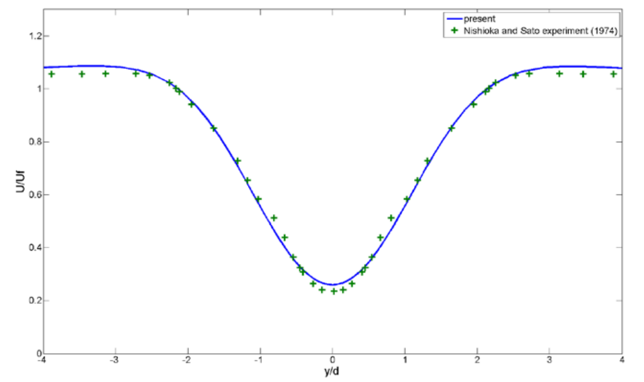

Figure 25: Wake velocity vertical profile at Re=10 compared to measurements at x=3d downstream of the cylinder's center. The velocity is scaled with the free-stream speed, denoted as Uf.

To resolve the unsteadiness of the flow, the governing Eqs. (2) and (3) are solved. Time advancement of the solving procedure proceeds with the 4th order accurate Runge-Kutta method (Eqs. (4)-(7)). The dimensionless time step was kept constant and is chosen to be $dt^{'}=7.5x10^{-5}$, which corresponds to physical time step $dt^{'}=1.5x10^{-6}$ *s*. At higher time steps it was detected inaccurate velocity distribution in the wake.

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

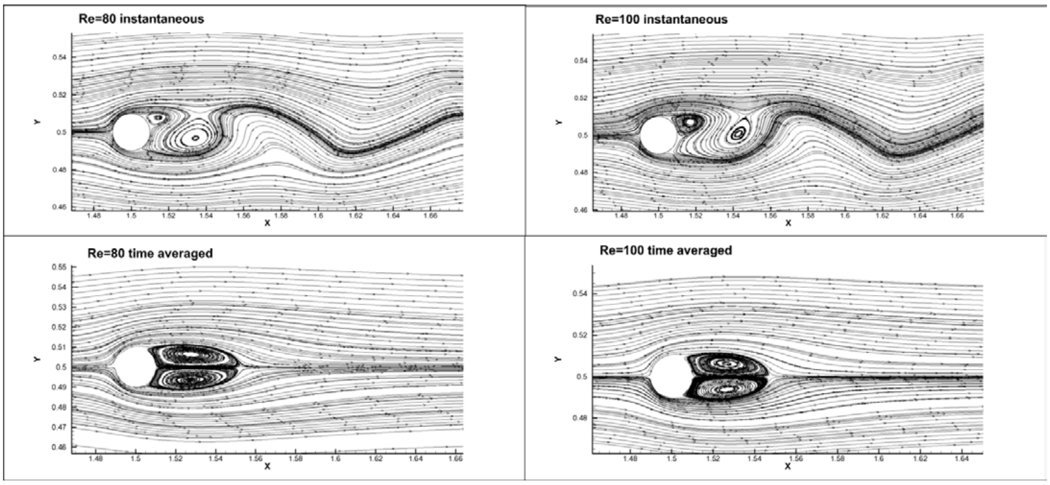

Figure 26: Flow patterns illustrated by streamlines at Re = 80 and Re = 100. Instantaneous fields are plotted at the top and time averaged fields at the bottom of the figure.

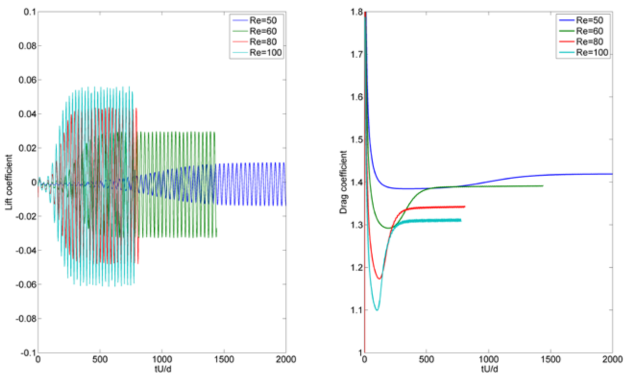

Figure 27: Time history of the load coefficients. They oscillate at constant amplitude after a certain time, which varies with Reynolds number.

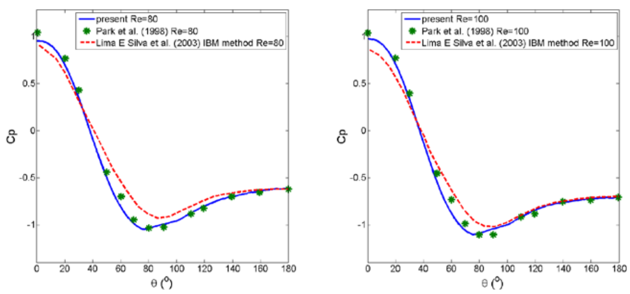

Figure 28: Mean pressure distribution along the cylinder's perimeter. At the right graph, Re=100 and at the left Re=80.

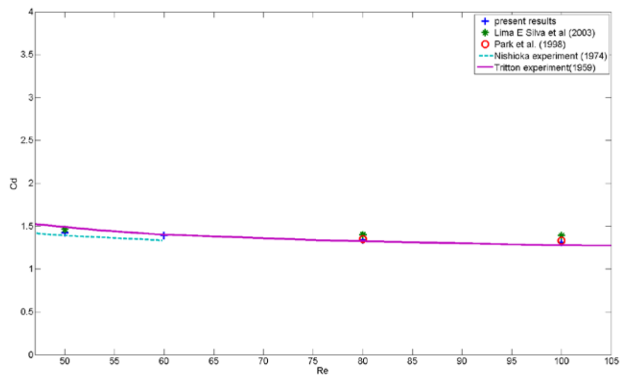

Figure 29: Mean drag coefficient versus Re number for the unsteady flow regime.

The topology of the flow is shown in Figure 26, where instantaneous streamlines are plotted to reveal the wavy pattern downstream of the cylinder at *Re*=80 and *Re*=100. Although the flow significantly oscillates, the mean velocity and pressure field is symmetric, producing zero lift force. The time averaged streamlines are also plotted in the latter figure. The loads are computed up to dimensionless time $t^{'} = 800 - 2000$ to ensure statistical convergence for each case. Force coefficients are plotted in Figure 27. It is evidently shown, that the force coefficients oscillate with constant amplitude after significant time since the beginning of the fluid motion. The latter time period extends with decrease of *Re* number. At *Re*=50 close to the wake instability point, it was needed to run up to 10 s physical time to achieve statistical convergence.

Pressure distribution is plotted in Figure 28. Agreement is reached, when comparing with the results of Park et al. [17]. Lima E Silva et al. [1] data deviate from both aforementioned results in the same way as in the steady flow regime (left diagram of Figure 13).

Unfortunately, no experimental pressure curves were found in this *Re* regime to support verification of the latter numerical results.

Drag coefficient is compared with the available data in Figure 29. In this range of *Re* number, very good agreement is reached with the other simulations and measurements.

Bubble's length could be also derived, but as a time averaged quantity. Indeed, during oscillations, eddies are developed in the near wake and shift in time the stagnation point downstream of the cylinder. However, if the flow is time averaged, then the structure is similar to those ones observed in the steady separated flow regime (Figure 19).

Figure 30 plots the time averaged length of the eddies computed in the present study and Lima E Silva et al. [1] and measured in the

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

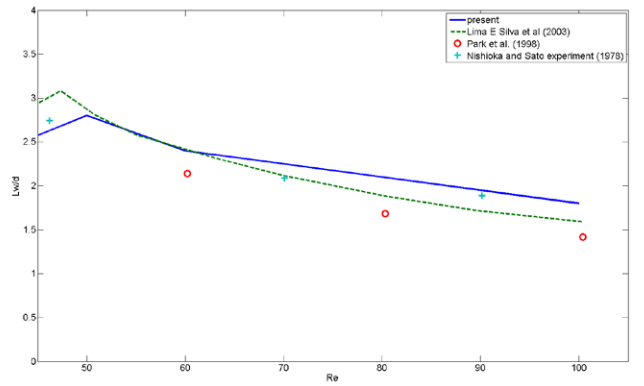

Figure 30: Wake bubble's time averaged length as a function of the Reynolds number for the unsteady flow regime.

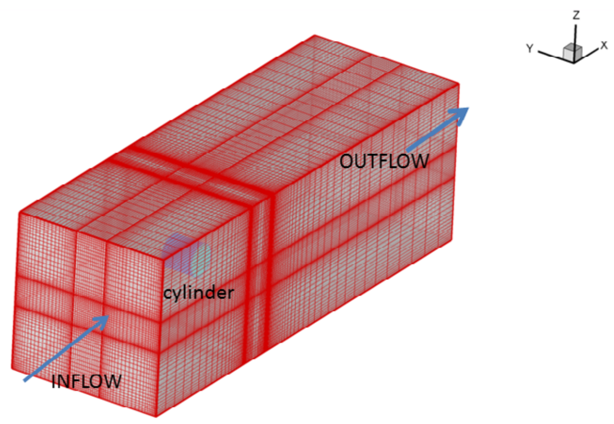

Figure 31: The computational domain is illustrated by the boundary mesh. The dimensions of the domain are 25d x 5d x 5d, where d denotes the cylinder's diameter.

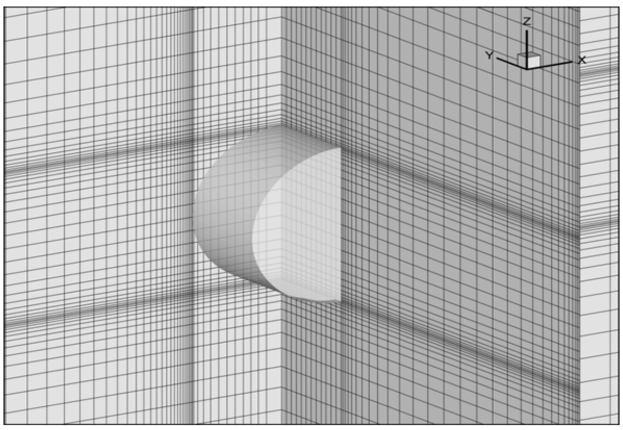

Figure 32: Three dimensional Conformal Cartesian mesh in the vicinity of the cylinder's surface, illustrated by two sections at y=d/2 and z=d/2, where d is the cylinder's diameter. Surface's transparency depicts the accurate intersection of the grid lines with the surface.

experiment of Nishioka and Sato cited in [1]. Results slightly over-predict the time averaged eddy measured in the experiment. The present

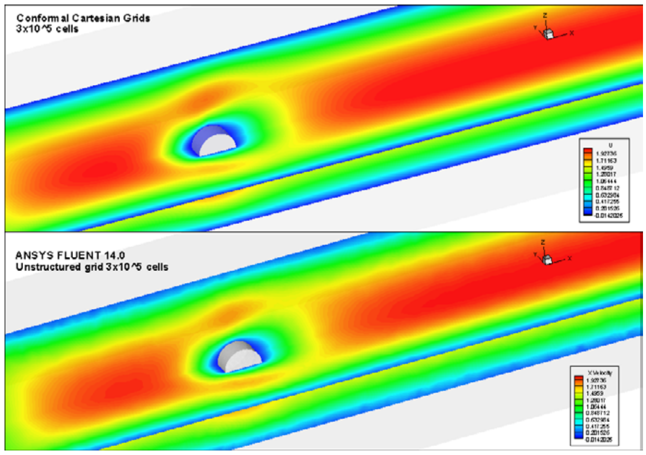

Figure 33: Double section at the middle of the cylinder (y = d/2 and z = d/2) illustrating the flow by axial velocity contours. The Reynolds number is 10 based on the diameter D of the channel and the dimension of the domain is 3D x 1D x 1D.

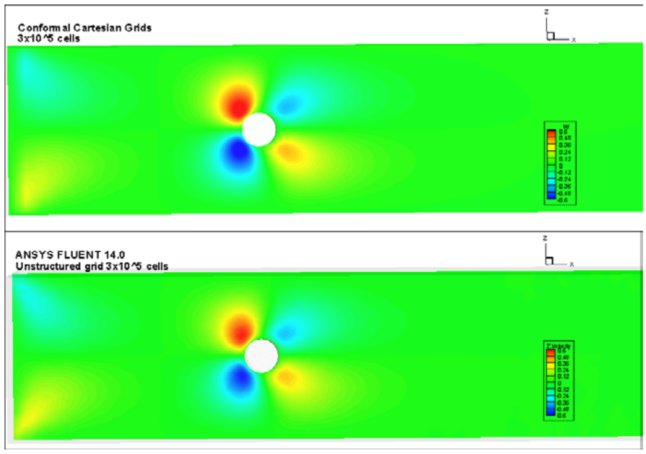

Figure 34: Section at the middle of the cylinder (y=d/2) illustrating the flow by vertical velocity contours.

computations are closer to Lima E Silva et al. [1] simulations. Park et al. [17] results deviate approximately 10-15% from the experiment, which is quite astonishing based on the fact that in all other comparisons, the computations were found to be in good agreement with the experiments.

Solution of a Three Dimensional Flow Past a Cylinder

The method is extensible to three dimensional symmetric problems. In the present section only a brief description of the method for such flow cases will follow, since it is planned those problems to be addressed and validated in a separate paper.

The author implements the well-known vorticity-velocity methods to solve the three dimensional Navier-Stokes equations (19), expressed in the following form:

$$\frac{\partial \vec{z}}{\partial t} + \nabla \times (\vec{z} \times \vec{V}) = \frac{1}{\mathrm{Re}} \nabla^2 \vec{z} \qquad (32)$$

Citation: Karabelas SJ (2015) Conformal Cartesian Grids for Symmetric Bodies: A Novel Boundary Fitted Grid Method. J Appl Computat Math 4: 234. doi:10.4172/2168-9679.1000234

$$\nabla^2 \vec{V} = -\nabla \times \vec{z} \tag{33}$$

where $\vec{V}$ denotes the velocity field.

By following the above formulation, it is known that the continuity equation is not necessarily satisfied. In the present version of the code, a Helmholtz projection of the velocity field is implemented to guarantee the fulfillment of the continuity equation:

$$\nabla^2 \varphi = \nabla \cdot \vec{V} \tag{34}$$

$$\vec{V} = \vec{V} - \nabla \varphi \tag{35}$$

where φ represents a scalar variable, which corrects the velocity field in order to be solenoidal.

In all 3-D simulations, the time advancement is performed via the Crank-Nicolson method, which is second order accurate in time. The second order accurate central difference scheme is chosen to discretize spatially the domain. Boundary conditions for vorticity are imposed as a function of the velocity conditions (applying the definition of vorticity). These boundary conditions are consistent if and only if the Helmholtz projection has been applied.

The laminar flow past a cylinder with end effects, which is bounded by side and vertical walls is chosen to demonstrate the applicability of the method in 3-D domains (Figure 31). The present problem is a three dimensional steady flow, which develops across the channel walls and the cylinder's surface. The Reynolds number based on the channel's diameter is 10, the aspect ratio of the cylinder 1.0 and the blockage ratio 0.04. The generated mesh is shown in Figure 32 and it consists of 3×10^5 cells.

The results are qualitatively compared to the solution of the well-established commercial software ANSYS Fluent 14.0 (Figures 33 and 34). Satisfactory agreement is reached regarding the global flow patterns of such wall-bounded flow. Thorough validation of the 3-D solver is planned to be completed in a future paper and this section only provides general results to demonstrate the applicability of the method for the 3-D symmetric objects.

Conclusions and Perspectives

A novel Cartesian grid method is developed, entitled as "Conformal Cartesian Grids". The Conformal Cartesian grids could possess full conformity with the domain's and object's boundaries and remain orthogonal everywhere. The method is easily applied to resolve problems related to 2-D and 3-D symmetric objects, but it is also possible to be extended for non-symmetric shapes. However, this extension requires a concrete mathematical proof for further applications, as it is explained in section 2.3.

Summarizing the general performance of the method, it could be stated that it retains almost all of the advantages of a Cartesian method, thus structured storage memory, low number of operations per grid point, low truncation error (since the mesh is fully orthogonal), simplicity in programming and robust grid generation. The fact that the boundary conditions could be imposed precisely onto the surface, gives a distinct advantage over the traditional Cartesian methods. Nevertheless, the penalty for this ideal combination of advantages, is that the discretization along one marching direction (e.g. axial) is dependent upon the discretization along the other (e.g. vertical) in order to retain coincidence of the grid points to the object's surface. Independency of the marching directions discretization could be attained far from the object.

Validation of the present method's fidelity is completed by resolving the flow past a cylinder at low *Re* numbers and comparing the results with extensive database of experiments and other numerical simulations. Results of the CCG method are found to be in very good agreement with measurements and previous computations.

Next steps of development will consider validation of the three dimensional problems and meshing around non-symmetric objects. Grids past arbitrary shapes constitute a great mathematical challenge to cope with. There are a lot of obstacles regarding the finalization of the mesh generation process in those problems. The extreme points condition already mentioned in section 2.3., could be a possible way to finalize the process, but it is not yet established the methodology, which should be followed to meet fast and efficiently this condition.

References

1. Lima E, Silva ALF, Silveira-Neto A, Damasceno JJR (2003) Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. Journal of Computational Physics 189: 351-370.
2. Tannehill JC, Anderson DA, Pletcher RH (1997) Computational fluid mechanics and heat transfer. (3rdEdn) Taylor and Francis Inc., London.
3. Peskin CS (1977) Numerical analysis of blood flow in the heart. J Comp Phys 25: 220-252.
4. Berger MJ, Aftosmis MJ (1998) Aspects (and aspect ratios) of Cartesian mesh methods. Proceedings of the 16th International Conference on Numerical Methods in Fluid Dynamics To appear in Lecture Notes in Physics, Germany.
5. Udaykumar HS, Shyy W, Rao MM (1996) A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries. Int J Numer Methods Fluids 22: 691-705.
6. 6.Rajani BN, Kandasamy A, Majumdar S (2009) Numerical simulation of laminar flow past a circular cylinder. Applied Mathematical Modeling 33: 1228-1247.
7. Thom A (1933) The flow past circular cylinders at low speeds. Proc R Soc Lond 141: 651-669.
8. Fujimoto K, Fujii K (2007) Study on the automated cfd analysis tools for conceptual design of space transportation vehicles. 5th Joint ASME/JSME Fluids Engineering Conference, California, USA.
9. Fujimoto K, Fujii K, Wang ZJ (2009) Improvements in the reliability and efficiency of body-fitted Cartesian grid. Method. 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, Orlando, Florida.
10. Sen S, Mittal S, Biswas G (2009) Steady separated flow past a circular cylinder at low Reynolds numbers. J Fluid Mech 620: 89-119.
11. Ferziger JH, Peric M (1996) Computational methods in fluid dynamics. Springer-Verlag, New York.
12. Iaccarino G, Kalitzin G, Moin P, Khalighi B (2004) Local grid refinement for an immersed boundary RANS solver, 42nd Aerospace Sciences meeting and Exhibit, Nevada.
13. Norberg C (1994) An experimental investigation of the flow around a circular cylinder: Influence of aspect ratio. J Fluid Mech 258: 287-316.
14. Nishioka M, Sato H (1974) Measurements of velocity distributions in the wake of a circular cylinder at low Reynolds numbers. J Fluid Mech 65: 97-112.
15. Taneda S (1956) Experimental investigation of the wakes behind cylinders and plates at low Reynolds numbers. J Phys Soc Japan 11: 302.
16. Grove AS, Shair FH, Petersene EE, Acrivos A (1964) An experimental investigation of the steady separated flow past a circular cylinder. J Fluid Mech 19: 60-80.
17. Park J, Kwon K, Choi H (1998) Numerical solutions of flow past a circular cylinder at Reynolds number up to 160. KSME Int J 12: 1200-1205.
18. Triton DJ (1959) Experiments on the flow past a circular cylinder at low Reynolds number. J Fluid Mech 6: 547-567.

19. Mittal R, Iaccarino G (2005) Immersed boundary methods. Annu Rev Fluid Mech 37: 239-261.

20. Peskin CS (1981) The fluid dynamics of heart valves: Experimental, theoretical and computational methods. Annu Rev Fluid Mech 14: 235-259.

21. Coutanceau M, Bouard R (1977) Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation part 1 steady flow. J Fluid Mech 79: 231-256.