An Embedded Boundary Algorithm for Large Eddy Simulations in Complex Geometries

Andreea Cristina PETCU*,[1], Ionut PORUMBEL[1], Cristian CARLANESCU[1]

*Corresponding author
*,[1] National Research and Development Institute for Gas Turbines "COMOTI"
B-dul Iuliu Maniu 220D, Bucharest 061126, Romania
andreea.petcu@comoti.ro

DOI: 10.13111/2066-8201.2012.4.2.10

Abstract: *The paper presents a novel implementation of the embedded boundary algorithm in a Large Eddy Simulation CFD algorithm. The embedded boundary method aims at allowing the simulation of complex geometry flows on Cartesian computational grids, increasing the numerical accuracy of the method, while maintaining acceptable computational requirements and ease of implementation. The novelty of the present implementation consists in a new method for determining the interpolation points required for computing the state vector at the location required for a correct embedded boundary condition application. The paper also includes method validation results for two flow cases: a laminar Poiseuille flow and laminar and turbulent flows past circular cylinders. The simulation results are in good agreement with the analytical solution (where available) and with experimental measurements in the literature.*

Key Words: *CFD, LES, embedded Boundary, boundary conditions*

1. INTRODUCTION

As most engineering problems involve complex geometries, the ability to handle these in an accurate and computationally efficient way is one of the challenges for the today's computational fluid dynamics. The advent of parallel computing and the ever increasing availability of memory space and CPU time for both research laboratories and industry stimulated the usage of LES in modelling turbulent flows. The main advantage of the method over the old RANS techniques is its ability to capture some of the small scales involved in a turbulent flow and, accordingly, some of the turbulent structures. The LES approach is particularly of interest in reactive flows, such as the turbulent combustion inside a gas turbine combustor, as the chemical reactions are known to take place at very small (molecular) scales.

Compared to the RANS approach, the LES is more computationally expensive. When a complex geometry is governing the flow, the difficulties of constructing a grid that follows the geometry may drive the computational expenses to unaffordable values. As discussed in a recent review [1], current research is using three general types of approaches for the state - of - the - art LES codes: the unstructured mesh approach, the immersed boundary method or its variant, the embedded method approach.

The generation of an unstructured is considerably less time consuming then a block structure one for complex geometries, and new non - dissipative schemes were developed for use on unstructured grids [2], [3]. However, the unstructured grid approach is difficult to implement and has high memory requirements.

An easier implemented alternative is the immersed boundary method. Essentially, the method consists of substituting a body immersed in a flow by a force field affecting the

Navier - Stokes governing equations of the fluid flow. The original immersed boundary method was developed by Peskin et al. [4], [5], [6], [7], [8] to simulate blood flow in heart valves. The Navier - Stokes equations governing the fluid flow can be expressed, in the Einstein summation convention [9], as:

$$\frac{\partial(\rho u_i)}{\partial t}+\frac{\partial(\rho u_i u_j)}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\mu\frac{\partial^2 u_i}{\partial x_i \partial x_j}+f_i \tag{1}$$

while the continuity equation is:

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u_i u_j)}{\partial x_i}=0 \tag{2}$$

where ρ is the fluid density, u_i is the *i*-th component of the velocity vector, *p* is the fluid pressure, x_i is the *i*-th spatial direction, *t* is the time, and μ is the fluid viscosity. The last term, f_i represents, in general, the external body force density acting on the fluid element and, in the conditions of this paper, deserves special attention. In the Embedded Boundary approach, this body force will be chosen in such a way as to simulate the presence of the body boundary in the flow. Thus, the grid does not have to "know" about the existence of the body and instead the effects of the body its existence will be included in the governing equation as a momentum source, according to equation (1).

To achieve this, as shown in earlier papers [10], [11], the body force needs to be computed as:

$$f_i=\int_0^{L_u}\left\{\int_0^{L_v}F_i(s_u,s_v,t)\delta\left[x_i-X_i(s_u,s_v,t)\right]\mathrm{d}s_v\right\}\mathrm{d}s_u \tag{3}$$

where F_i is the boundary force density:

$$F_i=S_i\left[X_i(s_u,s_v,t)\right] \tag{4}$$

Several symbols in equations (3) and (4) require further explanations: Thus, $X_i(s_u,s_v,t)$ is the parametric description of the surface of the immersed body, L_u, respectively L_v are the parameters limits of variation, δ is the three dimensional Dirac delta function [12]. Equation (3) describes the force density applied to the fluid by the immersed boundary, while equation (4) relates the boundary force density at a given position on the surface, and at a given time *t* to the boundary configuration. The time dependency of the equation allows for the possibility of a moving boundary.

Further developments of the method [13], [14], consisted in using a feedback scheme in which the velocity was used to iteratively determine the desired force. The downside of the method was its tendency to induce oscillations, thus forcing the usage of a very small CFL number (of order 10^{-2}). A different approach was, later, suggested, [15], proposing a discrete - time derivation of the forcing values and being thus able to remove the constraints on the CFL number.

Up to the present, the method was successfully used by a significant number of researchers in various fields involving fluid flows in complex geometries, as, for example, its application for a finite difference scheme using a slightly different algorithm [16], where the momentum forcing was applied inside the flow field, as opposed to the previous approach, which was applying the forcing inside the body and also including, for the first time, a subgrid scale turbulence model for a three dimensional flow, a finite volume approach, including a correction for mass conservation [17], or a study of multiphase flows using the immersed boundary method [18]. Large eddy simulations of high Reynolds number turbulent flows in complex geometries were also simulated with good results using the immersed boundaries technique [19].

A third solution to the complex geometry problem proposed in the literature [20] is a variant of the immersed boundary method, called the embedded boundary method. This new version consisted of a pointwise implementation of the momentum forcing principle. Under this method, the supplementary forcing terms are not explicitly computed and, instead, the values of the state vector inside the solid body are thus set that the fluxes through the solid surface are zero when computed. If, for instance, Δu_i is the difference between the value of the *i*-th velocity component and the forced velocity value, then equation (1) can be re-written, in a discretized form, as:

$$f_i = \frac{\partial\left(\rho u_i u_j\right)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \mu \frac{\partial^2 u_i}{\partial x_i \partial x_j} + \rho \frac{\Delta u_i}{\Delta t} \tag{5}$$

Thus, the momentum forcing vector can be expressed as a function of the current velocity values and the velocity values imposed, as boundary conditions, inside the solid body. Any method can be derived from the other and therefore they are mathematical equivalent. The advantage of the embedded boundary method is that it is more computationally efficient and easier to implement. The current study proposes the application of the embedded boundary method in the previously described formulation [20] for turbulent reactive flows. To the knowledge of the authors, no work combining the embedded boundary method with the Large Eddy Simulation (LES) of turbulence and considering reactive flows has been published yet.

The method presents several advantages. The main one is its ability to manage complex geometries. In the case of a gas turbine combustor, since the geometry is circular in nature, a traditional approach will require a cylindrical grid, which will yield acceptably accurate results except for the centreline region where the gradients will tend to infinity as the radius tends to zero. An alternate method, the unstructured grid approach, will be able to handle the centerline problem, but will have overwhelming computational requirements, both in terms of CPU time, and of memory. Since the final goal is the LES of a reactive flow, which very computationally demanding by itself, the unstructured grid computational expenses becomes prohibitive. Another advantage of the method is the fact that it allows the numerical scheme to employ a Cartesian grid irrespective of the complexity of the geometry involved. Firstly, this allows for orthogonal cells, which diminishes the numerical error and improves the convergence. Also, the aspect ratio of the cells is much easier to control. Lastly, since the immersed boundary algorithm can be formulated as a subroutine apart from the main solver, widely different geometries can be easily set up using the same code, with only the minor change of replacing the function describing the body surface.

2. METHOD DESCRIPTION

As discussed earlier, the embedded boundary method is based on setting the values of the state vector at particular locations inside the solid body in such a way that the solid wall boundary conditions are satisfied at the solid wall.

The first issue is, thus, to find the locations inside the body where the state vector values need to be set.

The Navier - Stokes solver employed herein, and described elsewhere [21], uses a second order central discretization scheme. The computational stencil used by the numerical scheme is shown in Fig. 1a.

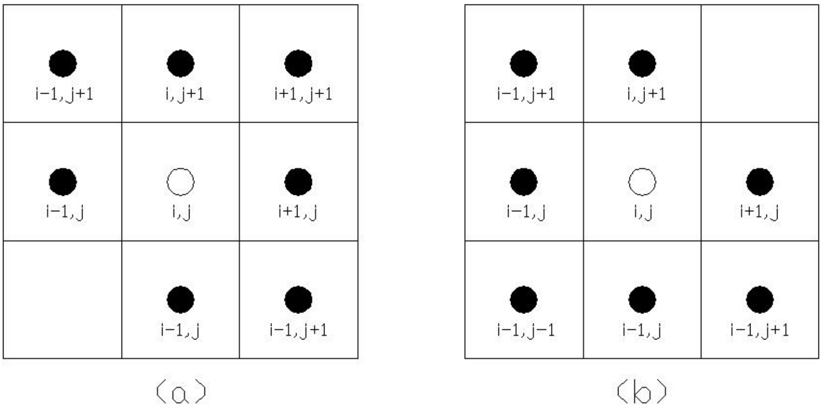


Fig. 1 – Computational stencil (left) and utilization of current point information (right)

It results that the information from a given point (*i, j*) is used as shown in Fig. 1b. Therefore, to enforce the solid wall boundary condition the state vectors must be specified at points inside the solid body which convey information to points in the flow (called hereafter "boundary points"). The algorithm achieving this is simple and very inexpensive since it only needs to be performed once, when the code is initialized, and the information can be afterwards stored and used to enforce the boundary conditions without re-computing the boundary cells locations.

If equation (6) describes in a general form a surface inside the computational domain, then a given point *P(x,y,z)* is on the same side as the coordinate system origin with respect to the surface if $\psi < 0$ and on different sides if $\psi > 0$.

$$\psi(x, y, z) = 0 \tag{6}$$

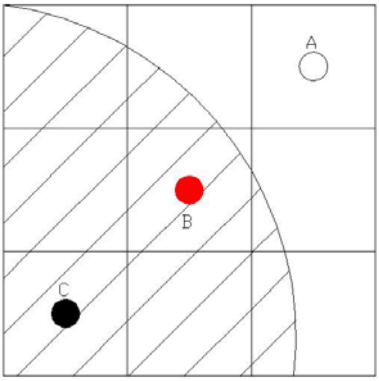


Fig. 2 – Point types in the embedded boundary method. Point „*A*” is a flow point, point „*B*” is a boundary point, and point „*C*” is a dead point.

Knowing this, the algorithm used to determine the boundary points computes the value of the function ψ at each cell center. With the location of the coordinate system origin known with respect to the solid body embedded in the flow, it is easy to decide whether the point is internal or external, based on the value of ψ in the current point. Next, if the point is outside the flow, the values of ψ are evaluated at each of the locations marked with black bullets in Fig. 1b. If any of these points results to be inside the flow, then the current point is flagged as a boundary point. Else, it is flagged as "dead" point (inside the solid body and not affecting the flow) and no computation will be carried on for it. Obviously, if the point is in the flow it is marked as such and the task of computing its state vector at every time step is left to the core LES solver. The three point types are represented in Fig. 2.

Once the locations of the boundary points are known, the next step is to determine the state vector values to be set. As ψ is a solid wall, the following boundary conditions apply:

$$\begin{cases} v_t = 0 \\ v_n = 0 \\ \dfrac{\partial p}{\partial n} = 0 \\ \dfrac{\partial T}{\partial n} = 0 \\ \dfrac{\partial Y_k}{\partial n} = 0 \quad k = \overline{1, M} \end{cases} \tag{7}$$

where v_n is the velocity component normal to the wall, v_t is the velocity in the plane tangent to the wall, p is the pressure, T is the temperature, n the direction of the outward normal to the embedded surface, Y_k are the mass fractions of the chemical species tracked in the problem, and M is the total number of chemical species tracked in the problem. In the equation system (7), the first represents the no - slip condition, the second expresses the wall impermeability, the third describes the wall pressure condition for a viscous flow and results from applying the first two equations in system (7) in the Navier - Stokes equation (1), the fourth equation describes an adiabatic wall and, depending on the type of problem at hand may be replaced by an equation describing an isothermal wall of known temperature, T_w:

$$T = T_w \tag{8}$$

Finally, the last equation in system (7) describes the wall quenching effect, which is the inhibition of chemical reaction near a solid wall, due to heat losses and to radical recombination at the wall [22].

The outward normal direction, n, can be computed using:

$$\vec{\boldsymbol{n}} = \frac{\nabla \psi}{|\nabla \psi|} \tag{9}$$

For a circle or a cylinder, as it is the situation in the cases presented herein, equation (9) reduces to:

$$\vec{\boldsymbol{n}} = (\cos\theta; \sin\theta) \tag{10}$$

where, if x_c and y_c are the circle center coordinates:

$$\theta = \arctan\frac{y - y_c}{x - x_c} \tag{11}$$

It is noteworthy that in all cases studied herein:

$$\frac{\partial \psi}{\partial z} = 0 \tag{12}$$

This fact greatly simplifies the problem allowing a bi – dimensional treatment. The same applies if ψ is in itself a bi – dimensional curve.

By combining the first two equations in system (7), the total velocity at the wall can be written as:

$$\vec{\boldsymbol{U}} = 0 \tag{13}$$

which translates to:

$$u_i = 0 \tag{14}$$

for any spatial direction i. Equation (14), together with the third and fourth equations of system (7), forms a complete set of boundary conditions for the supergrid equations governing the flow. To set a boundary condition for the turbulence model used by the LES algorithm, supplementary embedded boundary conditions are needed. The turbulence model used in this paper uses a subgrid kinetic energy transport equation [23], therefore, the supplementary equation must define the turbulent kinetic energy, *k*, at the embedded boundary. For the work presented here, the assumption used was:

$$k = 0 \tag{15}$$

based on equation (14). However, turbulent kinetic energy is known to peak in the wall boundary layer [24], which may, or may not be resolved by the computational LES grid. Therefore, condition (15) may need further study in the future.

In order to implement the above equations in the numerical algorithm, they need to be discretized. For this, first, the normal direction is determined for each boundary point, as per equation (10), or its simplified version (11). Next, the distance from the boundary point to the surface is computed along the normal direction and a new point (called hereafter "mirrored point") is determined inside the flow and at the same distance from the surface as the boundary point, as shown in Fig. 3.

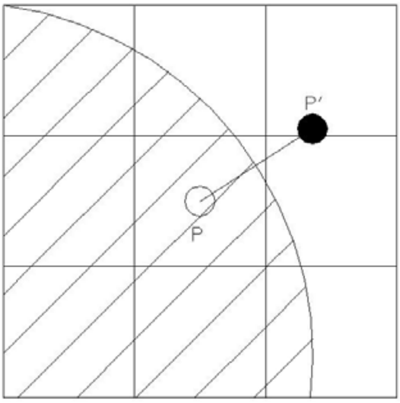


Fig. 3 – Construction of a mirrored point. The hatched region is the solid body, *P* is a boundary point and *P'* the mirrored point.

Using the state vector values at the mirrored point *P'*, the boundary conditions can be discretized, using a second order central finite difference scheme as:

$$\begin{cases} u_P = -u_{P'} \\ v_P = -v_{P'} \\ w_P = -w_{P'} \\ p_P = p_{P'} \\ T_P = T_{P'} \\ Y_{k,P} = Y_{k,P'} \quad k = \overline{1,M} \\ k_P = -k_{P'} \end{cases} \tag{16}$$

It is important to note that the temperature equation in system (16) may be replaced, in the case of an isothermal wall of known temperature, according to equation (8), by:

$$T_P = -T_{P'} + 2T_w \tag{17}$$

However, very seldom the mirrored point happens to coincide to the center of a cell, where the state vector values are known and, therefore, to set the boundary conditions one needs to interpolate the known values in the centers of the cells surrounding the mirrored point to get the value of the state vector at *P'*. Several methods are proposed in the literature for this. Most of them employ a bilinear interpolation scheme (equation (18)) which has the advantage of simplicity and of using unbiased information from both directions.

Of course, the employment of equation (18) assumes that the problem can be reduced to a bi - dimensional one near the immersed boundary, as information is collected only from two of the three spatial directions.

$$f(x, y) = ax + by + cxy + d \tag{18}$$

In equation (18) *f* can be any component of the state vector (density, velocity components, pressure, temperature, subgrid kinetic energy or mass fractions of the chemical species), with the coefficients *a, b, c, d* determined by forcing the function to assume the known values at the centers of the neighbouring cells.

As discussed previously, since the normal to the solid embedded boundary is independent of the third spatial direction (equation (12)), the nature of the problems investigated so far and presented herein complies with this requirement, so the bilinear interpolation proved useful. However, more complicated geometries may require a three - dimensional interpolation. An extension of equation (18) to three dimensions may be used:

$$f(x, y) = ax + by + cxy + d \tag{19}$$

However, the proper selection of an interpolation function for a three – dimensional geometry requires more in-depth consideration and testing.

$$f(x, y, z) = ax + by + cz + dxy + exz + fyz + gxyz + h \tag{20}$$

The coefficients *a, b, c,* and *d* in equation (19) are determined by solving the linear equation system below:

$$\begin{cases} F_1 = ax_1 + by_1 + cx_1y_1 + d \\ F_2 = ax_2 + by_2 + cx_2y_2 + d \\ F_3 = ax_3 + by_3 + cx_3y_3 + d \\ F_4 = ax_4 + by_4 + cx_4y_4 + d \end{cases} \tag{21}$$

with the notations in Fig. 4, where F_i stands for any of the state vector components at the location *i*.

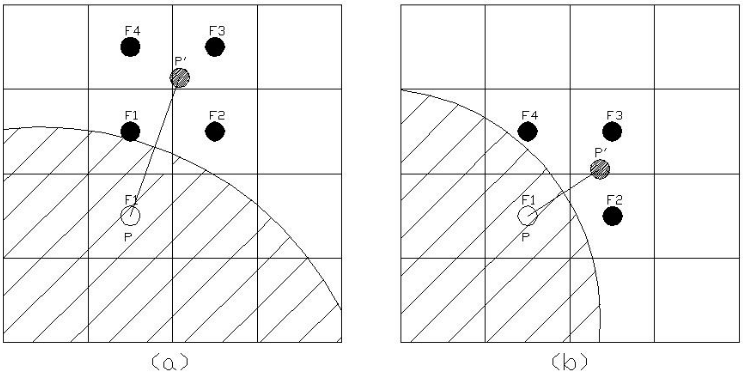


Fig. 4 - Interpolation method. a) All points required for the interpolation are inside the flow field. b) Some of the points required for the interpolation are outside the flow filed.

If the mirrored point is such that all its neighbouring cells have centers located inside the flow field (as shown in Fig. 4a) the equation system (21) can be solved explicitly and boundary conditions at the solid wall can be, hence, set. If the situation in Fig. 4b occurs, the system (21) becomes implicit and its solution becomes more difficult and more computationally expensive. To avoid this, a new interpolation method was developed for this study, a method that uses the information concerning the behaviour of the state vector variables at the solid wall, instead of their unknown values inside the body.

Depending on the situation, as shown in Fig. 5, the unknown interpolation points are replaced with points on the boundary, where the solid wall boundary conditions (equations (16)), apply.

Thus, the equation corresponding to point *P* will be replaced by the equation corresponding to point *Q*:

$$0 = ax_Q + by_Q + cx_Q y_Q + d \tag{22}$$

for the velocity components and the subgrid kinetic energy or, for a circular surface:

$$0 = a + b\tan\theta + c\left(y_Q + x_Q\tan\theta\right) \tag{23}$$

with *θ* given by equation (11), for pressure, temperature and chemical species mass fractions.

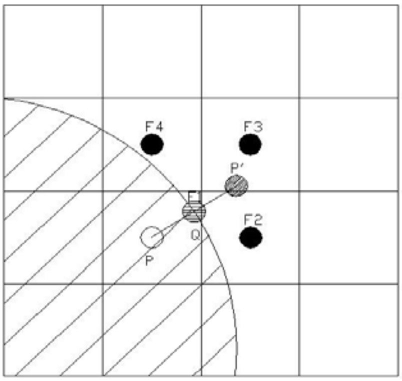


Fig. 5 - Interpolation method modification to avoid an implicit system. Point *Q* will be used for interpolation instead of point *P*.

3. NUMERICAL RESULTS

The embedded boundary method numerical algorithm presented earlier was integrated in an LES solver developed at INCDT – COMOTI [21] and used for two numerical studies, in order to validate the embedded boundary approach. The results, compared against numerical, analytical and experimental data existing in the literature, are presented in the following.

Computational Grid

Before actually presenting the numerical results, a few general issues concerning the computational grid are worth mentioning. Firstly, as discussed before, the computational grid is Cartesian and, thus, easier to generate and handle and yielding higher numerical accuracy. Secondly, the grid spacing used for this study was minimal near the embedded boundary to increase accuracy and was gradually decreasing away from the embedded boundary with a stretching factor of no more than *5*%. The grid resolution near the embedded wall determines the ability of the numerical scheme to capture the flow scales in the near wall region and is, thus, essential to the accuracy of the results [15].

Case 1: Fully Developed Laminar Flow in a Pipe

The first validation case, and the simplest of the three, concerns a fully developed laminar flow within a circular pipe (a.k.a. the Poiseuille flow). The computational grid was a Cartesian uniform grid of 100 x 100 x 100 points embedding the pipe. The simulated pipe diameter was 0.034 m and its length 0.1 m. Periodic boundary conditions were used along the pipe axis. The flow was initialized with a velocity field in accordance to the analytical solution for laminar flow within a circular pipe (equation (24)) and the computations were carried on for 5 flow - through times. The Reynolds number of the flow was 40 to ensure a perfectly laminar flow.

$$u(r)=\frac{R^2-r^2}{4\mu}\frac{\partial p}{\partial x} \tag{24}$$

where R is the pipe radius and x is the axial direction.

The velocity field computed after *5* flow - through times is shown in Fig. 6.

It can be seen that the flow field maintained the parabolic distribution given by equation (24). Also, the circular shape of the velocity contour lines presented in Fig. 6 is very accurate, signifying that the grid resolution used to describe the embedded walls was accurate enough for the problem.

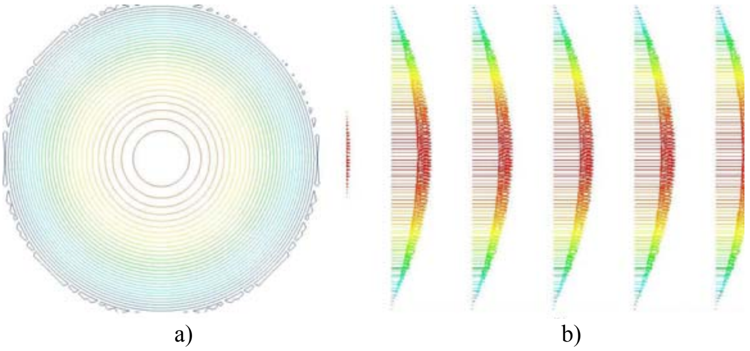

a) b)

Fig. 6 - Axial velocity in a Poiseuille flow. a) Velocity field view normal to the pipe axis. b) Velocity field view along the pipe axis.

A more clear verification of the accuracy of the method results from Fig. 7, showing the computed radial velocity profile and the analytical one. It can be seen that the match between the two profiles is excellent.

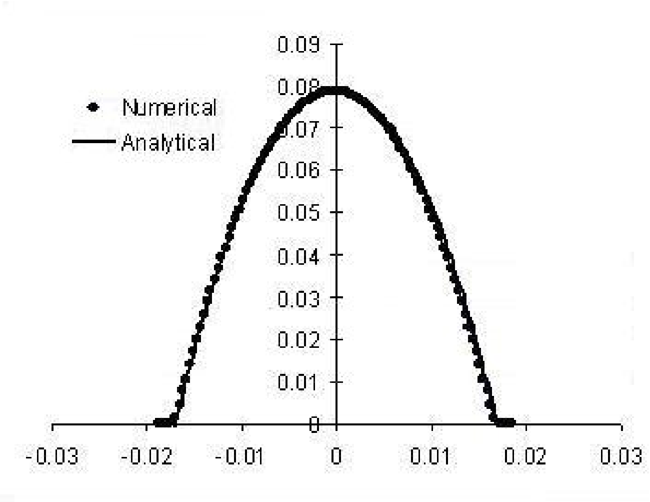


Fig. 7 - Radial profile of the axial velocity. The line indicates the analytical solution and the dots represent the numerical results.

Case 2: Flow Past a Circular Cylinder

The next case analysed in order to validate the code was the flow past a circular cylinder.

The computational grid was a Cartesian grid of *331* x *193* x *5* points embedding a circular cylinder of *0.3 m* diameter and extending throughout the computational domain in the *z* direction.

The computational domain is *70* cylinder diameters long, *100* diameters wide and 1 diameter deep.

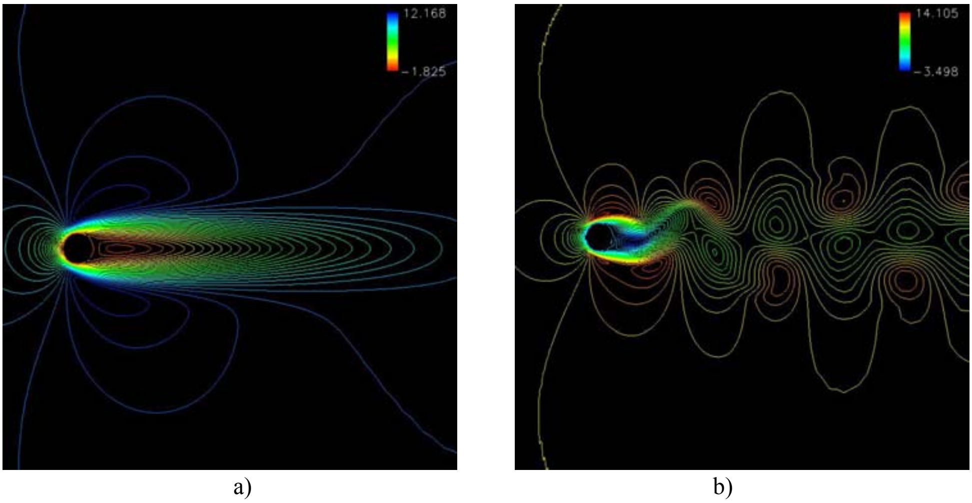
The geometrical characteristics were chosen in such a way as to match those in papers treating the same problem and using an immersed body technique [11], [17], since the validation will be done against these results.

Characteristic inflow - outflow boundary conditions [25] were used in the *x* direction, slip wall boundary conditions in the *y* direction and periodic boundary conditions in the *z* direction (where the z direction is aligned with the cylinder axis). The initial flow field was set to zero.

The computations were carried on for only *1* flow - through time, since the domain is much bigger than the solid body perturbing the flow.

Computations were performed at different Reynolds numbers: *40*, *100*, and *150*, and in order to reduce the computation time, the viscosity was artificially increased to allow a velocity of *10* m/s at the fixed Reynolds number.

Fig. 8 shows the contour lines of streamwise velocity around the cylinder in for Re = 40 and for Re = 150.

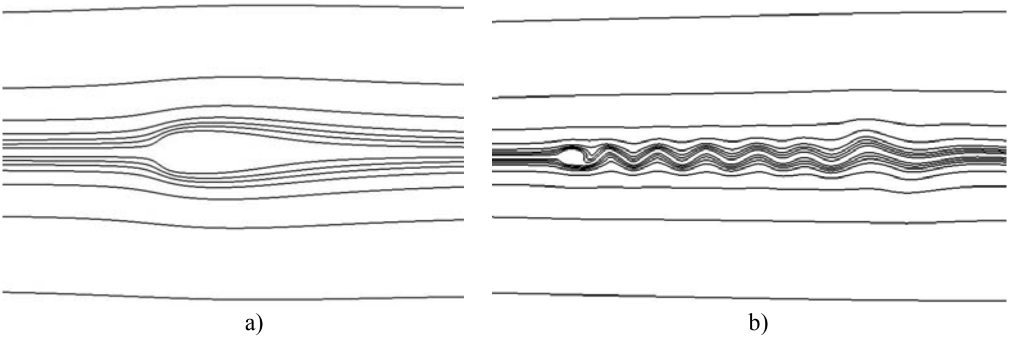


a) b)

Fig. 8 - Stremwise velocity contour lines. a) Re = 40; b) Re = 150.

The symmetric contours shown by the low Reynolds number flow Fig. 8a are changed as the turbulence develops with the increasing of the Reynolds number towards a fluctuating pattern in Fig. 8b.

The characteristic counter-rotating von Karman vortex street that appears as the turbulence sets in with the increase of the Reynolds number is even more obvious when looking at the flow streamlines, as shown in Fig. 9.



a) b)

Fig. 9 - Stremwise velocity contour lines. a) Re = 40; b) Re = 150.

Fig. 9 also indicates that in the higher Reynolds number case separation has occurred, while for the low Re case the flow is still attached and the streamlines follow the body surface.

The vortical structures that form in the flow behind the cylinder can be better seen in Fig. 10 showing the spanwise vorticity field.

As Fig. 10 shows, the qualitative agreement between the results of this study and those of previous similar simulations [11], [17] is reasonably good.

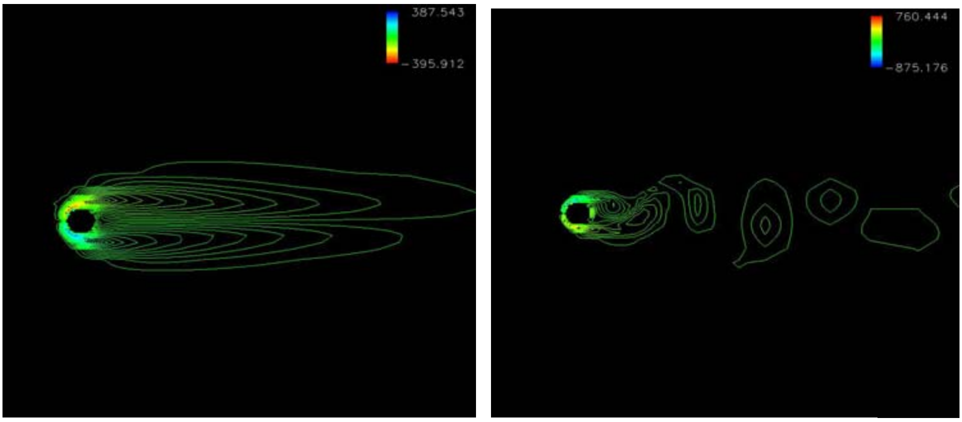
The older studies results are not included here, due to lack of author’s permission, but can be found in the quoted references.

To obtain quantitative information and to better appraise the capability of our numerical code more elaborate comparisons were carried on.

For instance, the pressure coefficient of the cylinder, given by:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2} \tag{25}$$

where the subscript ∞ refers to the upstream conditions can be compared with the numerical results in [17] and [26], as shown in Fig. 11 for Re = 40 and Re = 150:



a) b)

Fig. 10 - Vorticity field. a) Re = 40, current study; b) Re = 150.

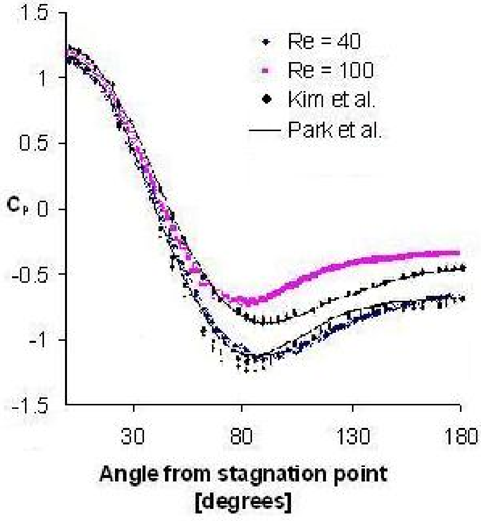


Fig. 11 - Pressure coefficient. The purple dots show the present study results at Re = 40, the blue dots show the present study results at Re = 100, the black dots the results of Kim at al. [17] and the black line the experimental data of Park et al. [26].

As Fig. 11 indicates, the agreement is very good for the low Reynolds number case, but less perfect for angles larger than 90°, i.e. in the wake behind the cylinder for the high Reynolds number case. This may be the effect of the artificially increased viscosity used for our computation. As the viscosity increases the effect of the momentum transfer across the wake is increased, hence the velocity deficit in the wake decreases faster than in the low viscosity case and, therefore, the pressure coefficient in the wake is higher in the present case.

The lift coefficient, defined as:

$$C_L = \frac{-S\oint\limits_{\Omega} p \cdot \mathrm{d}s}{\frac{1}{2}\rho U_\infty^2 D} \tag{26}$$

where *Ω* is the cylinder contour, *S* is the cylinder span and *D* its diameter was computed next. Its variation in time for Re = 150 is presented in Fig. 12. On the *x* - axis in Fig. 12 a non-dimensional time scale is used, defined as:

$$T = \frac{U_\infty t}{\frac{1}{2}D} \tag{27}$$

where *t* is the time in seconds.

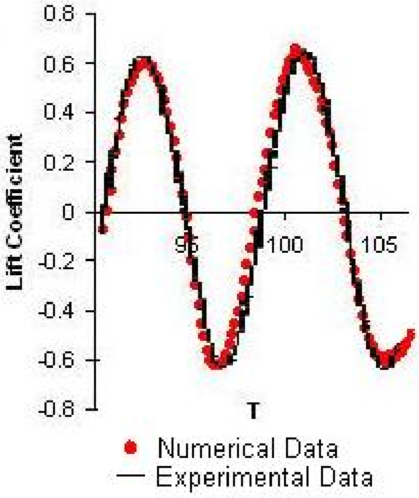


Fig. 12 - Lift coefficient. The red dots show the present study results and the black line the numerical data of Lai and Peskin [11].

The result compares well with the numerical data provided by Lai and Peskin [11]. An excellent agreement with the data in the literature can be noticed for the Strouhal number at a Reynolds number of 150, as presented in Table 1, where the Strouhal number was computed as:

$$St = \frac{2}{\Delta T_p} \tag{28}$$

Table 1 - Parameter values for steady state

Source	Strouhal number (St)
Present study	0.185
Reference [11]	0.184
Reference [27] (Williamson)	0.183
Reference [27] (Rosko)	0.182

4. CONCLUSION AND FUTURE WORK

A new implementation of the embedded grid method is developed and implemented in an LES CFD algorithm. The embedded grid method allows the simulation of complex geometry flows on Cartesian computational grids, increasing the numerical accuracy of the method, while maintaining acceptable computational requirements and ease of implementation. To the date, this is the first successful implementation of the method in a reactive LES numerical solver. The novelty of the approach also consists in the development of a new method for determining the interpolation points required for the computation of the state vector where the embedded boundary conditions are applied. The new method maintains an explicit numerical scheme, avoiding additional computational costs.

The algorithm was tested with good results for laminar and turbulent isothermal flows. The embedded boundary algorithm described here was designed and implemented also for reactive flows, but it has not yet been tested, as it requires a significant computational effort.

Future work will have to consist, obviously, in achieving the algorithm validation for reactive flows. Another research issue remains the turbulent kinetic energy value at the wall, as discussed earlier. Yet unsolved remains, also, the problem of the three - dimensional interpolation algorithm. Further research efforts may also be needed for improving the interpolation accuracy to properly simulate the turbulent effects in shear layers, since the interaction between turbulence and combustion is critical for a correct numerical simulation of reactive flows [21].

REFERENCES

[1] P. Moin, Advances in Large Eddy Simulation Methodology for Complex Flows, *International Journal of Heat and Fluid Flow*, vol. **23**, pp. 710 – 720, 2002.

[2] K. Mahesh, G. Constantinescu, S. Apte, G. Iaccarino, P. Moin, Large - Eddy Simulation of Gas - Turbine Combustors, *Annual Research Briefs*, Center for Turbulent Research, Stanford University, 2001.

[3] K. Mahesh, G. Constantinescu, P. Moin, Large - Eddy Simulation of Gas - Turbine Combustors, *Annual Research Briefs*, Center for Turbulent Research, Stanford University, 2000.

[4] C.S. Peskin, *Flow Patterns around heart valves: A Digital Computer Method for Solving the Equations of Motion*, Ph. D. thesis, Albert Einstein College of Medicine. Yeshiva University, New York NY, U.S.A., 1972.

[5] C. S. Peskin, D.M. McQueen, *Fluid Dynamics of the Heart and its Valves*, in H.G. Othmer, F.R. Adler, M. A. Lewis, and J.C. Dallon (Eds.) *Case Studies in Mathematical Modeling: Ecology, Physiology and Cell Biology*, Englewood Cliffs, N.J.: Prentice-Hall, pp 309 – 337, 1996.

[6] D.M. McQueen, C. S. Peskin, Shared - Memory Parallel Vector Implementation of the Immersed Boudary Method for the Computation of Blood Flow in the Beating Mammalian Heart, *Journal of Supercomputing*, vol. **11**, no 3, pp. 213 – 236, 1997.

[7] D.M. McQueen, C. S. Peskin, *Heart Simulation by an Immersed Boundary Method with Formal Second Order Accuracy and Reduced Numerical Viscosity* in H. Aref, and J. W. Phillips (Eds.) *Mechanics for A New Millenium*, Proceedings of the International Conference on Theoretical and Applied Mechanics (ICTAM), Kluwer Academic Publishers, 2001.

[8] C. S. Peskin, The Immersed Boundary Method, *Acta Numerica*, vol. **11**, pp. 479 – 517, 2002.

[9] A. Einstein, "The Foundation of the General Theory of Relativity". Annalen der Physik. 1916. Available at http://www.alberteinstein.info/gallery/gtext3.html.

[10] C. S. Peskin, B.F. Printz, Improved Volume Conservation in the Computation of Flows with Immersed Elastic Boundaries, *Journal of Computational Physics*, vol. **105**, pp. 33, 1993.

[11] M. K. Lai, C. S. Peskin, Immersed - Boundary Method with Formal Second Order Accuracy and Reduced Viscosity, *Journal of Computational Physics*, vol. **160**, 2000, pp. 705 – 719, 2000.

[12] R. Bracewell, The Fourier Transform & Its Applications, *McGraw-Hill Science/Engineering/Math*; 3rd Edition, pp. 94, 1999.

[13] D. Goldstein, R. Handler, L. Sirovich, Modeling a No - Slip Flow Boundary with an External Force Field, *Journal of Computational Physics*, vol. **105**, pp. 354, 1993.

[14] E. M. Saiki, S. Biringen, Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method, *Journal of Computational Physics*, vol. **123**, pp. 450, 1996.

[15] J. Mohd - Yusof, *Combined Immersed - Boundary / B - Spline Methods for Simulations of Flow in Complex Geometries*, Annual Research Briefs, Center for Turbulence Research, NASA, Ames and Stanord University, pp. 317, 1997.

[16] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd - Yusof, Combined Immersed - Boundary Finite - Difference Methods Methods for Three - Dimensional Complex Fluid Geometries, *Journal of Computational Physics*, vol. **161**, pp. 35, 2000.

[17] J. Kim, D. Kim, H. Choi, An Immersed - Boudary Finite - Volume Method for Simulation of Flow in Complex Geometries, *Journal of Computational Physics*, vol. **171**, pp. 132 – 150, 2001.

[18] H. S. Udaykumar, H. C. Kan, W. Shyy, R. Tran - Son - Tay, Multiphase Dynamics in Arbitrary Geometries on Fixed Cartesian Grids, *Journal of Computational Physics*, vol. **137**, no 2, pp. 366 – 405, 1997.

[19] R. Verzicco, J. Mohd - Yusof, P. Orlandi, D. Haworth, Large Eddy Simulation in Complex Geometric Configurations Using Boundary Body Forces, *AIAA Journal*, vol. **38**, no 3, pp. 427 – 433, 2000.

[20] S. Xu, T. Aslam, D Scott Stewart, *High Resolution Numerical Simulation of Ideal and Non - ideal compressible Reacting Flows with Embedded Internal Boundaries*, The McGraw - Hill Companies, 1999.

[21] I. Porumbel, *LES of Bluff Body Stabilized Premixed and Partially Premixed Combustion*, VDM Verlag Dr. Muller, Saarbrucken, Germany, 2008.

[22] I. Glassman, *Combustion*, Academic Press, New York, NY, USA, 1977.

[23] I. Porumbel, C. Cârlănescu, F.G. Florean, C.E. Hrițcu, LES Algorithm for Turbulent Reactive Flows Simulation. Part I. Algorithm Description, *ACC'10 Proceedings of the 2010 international conference on Applied computing conference*, pp. 71-86, 2010

[24] S. B. Pope, *Turbulent Flows*, University Press, Cambridg, UK, 2000.

[25] T. J. Poinsot, S. K. Lele, Boundary Conditions for Direct Simulations of Compressible Viscous Flows, *Journal of Computational Physics*, vol. **101**, pp. 104 – 129, 1992

[26] J. Park, K. Kwon, H. Choi, Numerical Solutions of Flow Past a Circular Cylinder ar Reynolds Numbers up to 160, *KSMA International Journal*, vol. **12**, pp. 1200, 1998.

[27] C. Liu, X. Zheng, C. H. Sung, Preconditioned Multigrid Methods for Unsteady Incompressible Flows, *Journal of Computational Physics*, vol. **139**, pp. 35, 1998.