V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

AN IMMERSED-BOUNDARY METHOD FOR SOLVING CONJUGATE HEAT TRANSFER PROBLEMS IN TURBOMACHINERY

M. D. de Tullio, S. S. Latorre, P. De Palma, M. Napolitano, G. Pascazio

Politecnico di Bari, DIMeG and CEMeC
via Re David 200, 70125 Bari, Italy
e-mail: m.detullio@poliba.it, latorre@imedado.poliba.it
depalma@poliba.it, napolita@poliba.it, pascazio@poliba.it

Key words: compressible flows, RANS equations, complex geometry

Abstract. *This paper provides an immersed boundary method using a flexible local grid refinement technique for solving conjugate-heat-transfer problems. The proposed method is used to solve the flow between two heated cylinders together with the temperature field within the inner hollow cylinder and then to predict turbomachinery blade cooling.*

1 INTRODUCTION

In recent years the immersed boundary (IB) method is emerging as a very appealing approach for solving flows past very complex geometries, like those occurring in most industrial applications. Its main, very significant, feature is the use of a Cartesian grid embodying the complex boundaries of the flow domain, which allows one to generate the computational mesh within few minutes instead of the hours or even the days required to generate very complicated body-fitted grids. The IB technique was originally developed for incompressible flows [1, 2, 3, 4, 5] using non-uniform Cartesian grids to take advantage of simple numerical algorithms. Some of the authors have extended the IB method to the preconditioned compressible Reynolds-averaged Navier–Stokes (RANS) equations in order to solve complex flows for a wide range of the Mach number [6] and equipped it with a local mesh refinement procedure to resolve boundary layers and regions with high flow gradients (e.g. shocks) [7].

In this work, the IB method is extended to the solution of conjugate heat-transfer problems. The Fourier heat-conduction equation is solved inside the immersed body coupled together with the RANS equations closed by the $k-\omega$ turbulence model.

In the following sections, after a brief description of the method and of the boundary conditions at the immersed boundary, results are obtained for a well documented test-case as well as for the blade cooling of a turbine cascade.

2 GOVERNING EQUATIONS AND NUMERICAL METHOD

The Reynolds Averaged Navier–Stokes (RANS) equations, written in terms of Favre mass-averaged quantities and using the $k-\omega$ turbulence model, can be written as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho\, u_j) = 0, \tag{1}$$

$$\frac{\partial(\rho\, u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho\, u_j\, u_i) = -\frac{\partial p_t}{\partial x_i} + \frac{\partial \hat{\tau}_{ji}}{\partial x_j}, \tag{2}$$

$$\frac{\partial(\rho\, \tilde{H} - p_t)}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho\, u_j\, \tilde{H}\right) = \frac{\partial}{\partial x_j}\left[u_i\, \hat{\tau}_{ij} + (\mu + \sigma^* \mu_t)\frac{\partial k}{\partial x_j} - q_j\right], \tag{3}$$

$$\frac{\partial(\rho\, k)}{\partial t} + \frac{\partial}{\partial x_j}(\rho\, u_j\, k) = \tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta^* \rho\, \omega\, k + \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \mu_t)\frac{\partial k}{\partial x_j}\right], \tag{4}$$

$$\frac{\partial(\rho\, \omega)}{\partial t} + \frac{\partial}{\partial x_j}(\rho\, u_j\, \omega) = \frac{\gamma\, \omega}{k}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta\, \rho\, \omega^2 + \frac{\partial}{\partial x_j}\left[(\mu + \sigma \mu_t)\frac{\partial \omega}{\partial x_j}\right]. \tag{5}$$

In the equations above, $\tilde{H}$ and p_t are the total enthalpy and the pressure comprehensive of the turbulent kinetic energy, k,

$$\tilde{H} = h + \frac{1}{2} \mid \mathbf{u} \mid^2 + \frac{5}{3} k, \qquad p_t = p + \frac{2}{3}\rho\, k; \tag{6}$$

the eddy viscosity, μ_t, is defined in terms of k and of the specific dissipation rate, ω, according to the low-Reynolds-number $k-\omega$ turbulence model of Wilcox [8], namely:

$$\mu_t = \gamma^* \frac{\rho\, k}{\omega}. \tag{7}$$

Moreover, $\hat{\tau}_{ij}$ indicate the sum of the molecular and Reynolds stress tensor components. According to the Boussinesq approximation, one has:

$$\hat{\tau}_{ij} = (\mu + \mu_t)\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] - \frac{2}{3}\rho\, k\, \delta_{ij}. \tag{8}$$

Finally, the heat flux vector components, q_j, are given as:

$$q_j = -\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial h}{\partial x_j}, \tag{9}$$

where $Pr = 0.71$ and $Pr_t = 1$ are the laminar and turbulent Prandtl numbers, respectively. The Sutherland law is used to compute the molecular viscosity coefficient.

It is useful to write the RANS equations (1)-(5) in compact form:

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} - \frac{\partial E_v}{\partial x} - \frac{\partial F_v}{\partial y} - \frac{\partial G_v}{\partial z} = D, \tag{10}$$

where Q is the conservative variable vector, E, F, G and E_v, F_v, G_v indicate the inviscid and viscous fluxes, respectively, and D is the vector of the source terms.
A pseudo-time derivative is added to the left-hand-side of equation (10) in order to use a time marching approach for both steady state and unsteady problems and the preconditioning matrix, $\mathbf{\Gamma}$, proposed in [9, 10] is used to premultiply the pseudo-time derivative in order to improve efficiency. The final system reads:

$$\mathbf{\Gamma}\frac{\partial Q_v}{\partial \tau} + \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} - \frac{\partial E_v}{\partial x} - \frac{\partial F_v}{\partial y} - \frac{\partial G_v}{\partial z} = D, \tag{11}$$

where $Q_v = (p_t, u, v, w, T, k, \omega)^T$ is the primitive variable vector, which is related to Q by the Jacobian matrix $\mathbf{P} = \partial Q / \partial Q_v$. Equation (11) is rewritten in delta form discretizing the pseudo-time derivative by an Euler implicit scheme and approximating, for unsteady-flow problems, the physical-time derivative by a second-order-accurate three-point backward difference. After some modifications to improve the efficiency of the method, which do not affect the residual [7], applying the diagonalization procedure of Pulliam and Chausee [11], followed by a factorization, one obtains the following system:

$$\begin{gathered}\mathbf{S}\mathbf{M}_x\left[\mathbf{I} + \Delta\tau\frac{\partial}{\partial x}\left(\mathbf{\Lambda}_x - R_x\mathbf{I}\frac{\partial}{\partial x}\right)\right]\mathbf{M}_x^{-1}\mathbf{M}_y\left[\mathbf{I} + \Delta\tau\frac{\partial}{\partial y}\left(\mathbf{\Lambda}_y - R_y\mathbf{I}\frac{\partial}{\partial y}\right)\right]\mathbf{M}_y^{-1} \\ \mathbf{M}_z\left[\mathbf{I} + \Delta\tau\frac{\partial}{\partial z}\left(\mathbf{\Lambda}_z - R_z\mathbf{I}\frac{\partial}{\partial z}\right)\right]\mathbf{M}_z^{-1}\Delta Q_v = -\Delta\tau\left[\frac{3Q^r - 4Q^n + Q^{n-1}}{2\Delta t} + \mathcal{R}^r\right],\end{gathered} \tag{12}$$

where $\mathcal{R}^r$ represents the residual at the previous iteration and the matrices $\mathbf{S}$, $\mathbf{M}$, $\mathbf{R}$, and $\mathbf{\Lambda}$ are given in [7].

Equation (12) is discretized in space using a cell-centred finite volume approach. The convective terms at the RHS are discretized using a second-order-accurate upwind flux-difference-splitting scheme. The viscous terms are discretized by second-order-accurate centred differences. The LHS convective term is always discretized using a first-order upwind scheme, according to a deferred-correction approach, in order to ensure convergence of the iterative solver.

The resulting discrete system is solved direction-by-direction using a BiCGStab [12] approach, the boundary conditions being treated explicitly.

A data structure is employed which allows an efficient local grid refinement (LGR) for clustering cells near the immersed boundary and at other high-flow-gradient regions. For each face, the contributions of the neighbour cells are collected to build the corresponding convective and diffusive operators for the cell, the maximum number of neighbours being limited to two for the present 2D computations (see [7], for details).

3 IMMERSED BOUNDARY METHOD

The IB technique used in this work is based on that proposed in [3, 4]. In a preliminary step, the geometry under consideration, which is described by a closed polygon in two dimensions (a closed surface in three dimensions), is overlapped onto a Cartesian (non uniform) grid. Using the ray tracing technique based on the geometrical algorithms reported in [13], the computational cells occupied entirely by the flow are tagged as *fluid cells*; those whose centres lie within the immersed body are tagged as *solid cells*. Furthermore, the *fluid cells* neighbouring with *solid* ones are tagged as *interface fluid cells*, and the *solid cells* neighbouring with *fluid* ones are tagged as *interface solid cells*. Interface cells are used to enforce the boundary conditions between the solid body and the fluid. In the present implementation, starting from an auxiliary grid with uniform mesh size, a structured grid is generated by recursively halving the mesh size at the immersed boundary region, until an assigned target value is reached. This automatic refinement is based on the following strategy. A tag function, generated using the ray tracing technique, is used to mark the cells inside and outside the immersed body: an integer value ±1 is assigned to “fluid” and “solid” cells, respectively. The gradient of this function is different from zero only at the immersed boundary and depends on the local grid size. The components of this gradient in the x and y directions are used to select the rows of cells to be refined. The grid is refined until a user specified resolution is achieved at the boundary. A smoothing function can be applied on the ±1 tagging function to obtain a smeared interface that will allow a smoother transition between the coarse and the refined regions. By this procedure, a uniform grid is obtained. Then, starting from such a grid, it is possible to coarse the cells in the regions far from the boundaries until a maximum prescribed cell-size is achieved. Finally, one can define other regions of the computational domains to be refined, selecting the local resolution of the mesh, like the wake or wall regions and, finally, it is possible to refine on *void* surfaces, namely, surfaces without solid or interface points, like the bow-shock regions.

4 CONJUGATE HEAT TRANSFER

The RANS equations, solved at the “fluid cells” are coupled with the following equation for the thermal conduction which is solved at all points inside the solid body

$$\frac{\partial T}{\partial t} = \frac{k_s}{\rho_s c_s} \nabla^2 T, \tag{13}$$

where T is the temperature and k_s, ρ_s, and c_s are the solid thermal conductivity, density, and specific heat, respectively. Two boundary conditions are needed at the surface of the body to enforce the equality of the solid (s) and fluid (f) temperatures and heat fluxes at their interface:

$$T_f = T_s, \tag{14}$$

$$k_f \nabla T_f = k_s \nabla T_s, \tag{15}$$

k_f being the thermal conductivity of the fluid. The two conditions above are implemented according to the following procedure. The temperature gradient in the fluid at the surface, ∇T_f, is firstly computed; then, the temperature gradient in the solid, ∇T_s, is evaluated from equation (15). For each interface cell inside the solid, using such a gradient and the computed values of the temperature at the solid points surrounding the interface cell, $T_{i,s}$, $i = 1, ...N_{nbr}$, the temperature value at the closest point on the wall, $T_{w,s}$, is determined. The following interpolation formula is employed for this purpose:

$$T_s = \frac{\sum_i^{N_{nbr}} T_{i,s}\alpha_i}{\sum_i^{N_{nbr}} \alpha_i} + \left(\frac{1}{\beta} + \frac{1}{\sum_i^{N_{nbr}} \alpha_i} \right) \nabla T_s, \tag{16}$$

where α_i and β are the inverse distances between the surrounding cell centers and the interface cell center and between the wall point and the interface cell center, respectively. Finally, equation 14 is employed to compute T_f. In conclusion, when going from the fluid to the solid, the value of ∇T_f is enforced, whereas, when going from the solid to the fluid, the value of T_s is enforced.

5 RESULTS

5.1 Rotating flow inside a tube

The heat transfer problem studied in [14] has been considered at first to validate the present approach. As shown in figure 1, the fluid is comprised between a stationary hollow inner cylinder and a rotating outer one. The outer cylinder has a radius $R_o = 1.8\ m$, moves at a speed $U_0 = 5\ m/s$ and is kept at $T_o = 700\ K$. The inner hollow cylinder has radii equal to $R_i = 0.45\ m$ and $R_m = 0.9\ m$, respectively, the inner surface is kept at $T_i = 500\ K$

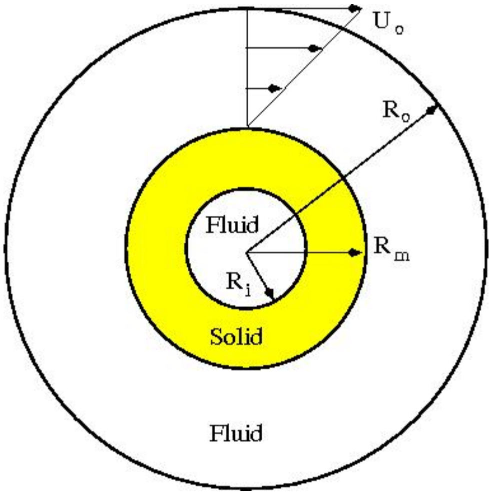


Figure 1: Scheme of the rotating flow test case.

and $k_s/k_f = 9$. An analytical solution is available for the two-dimensional Navier–Stokes equations with the above boundary conditions [14]. The velocity distribution is given as

$$u_r = 0; \quad u_\theta(r) = \begin{cases} 0, & for\, R_i < r < R_m\,(solid) \\ -\dfrac{R_o R_m^2 U_o}{(R_o^2 - R_m^2)r} + \dfrac{R_o U_o}{(R_o^2 - R_m^2)} r, & for\, R_m < r < R_o\,(fluid) \end{cases} \tag{17}$$

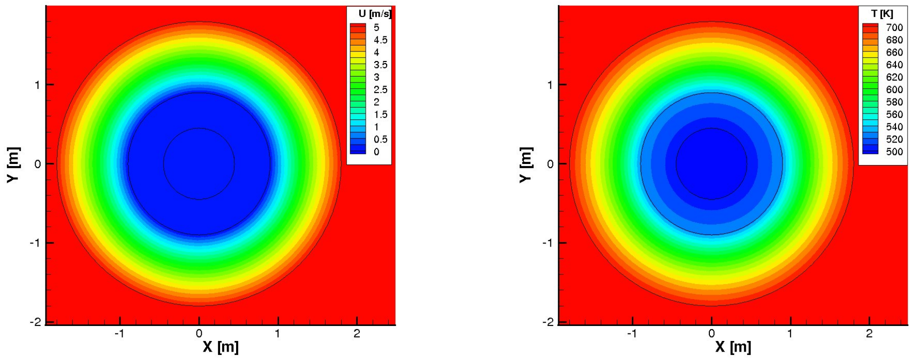
whereas the temperature distribution is given as

$$T(r) = \begin{cases} T_i + \dfrac{T_o - T_i}{log(R_m/R_i) + (k_s/k_f)\, log(R_o/R_m)}\, log\left(\dfrac{r}{R_i}\right), & for\, R_i < r < R_m\,(solid) \\ T_o - \dfrac{T_o - T_i}{log(R_o/R_m) + (k_f/k_s)\, log(R_m/R_i)}\, log\left(\dfrac{R_o}{r}\right), & for\, R_m < r < R_o\,(fluid). \end{cases} \tag{18}$$

Computations have been performed using a Cartesian grid having about 131000 cells with $\Delta x = \Delta y = 0.025$. The solution provides the flow field as well as the temperature field within the hollow cylinder, see Figure 2 which provides the contours of the velocity magnitude (a) and of the temperature (b). Figure 3 provides the comparison between the scatter plots of the velocity (a) and temperature (b) radial distributions and the corresponding analytical solutions. A satisfactory agreement is obtained.

5.2 Flow past a cooled turbine cascade

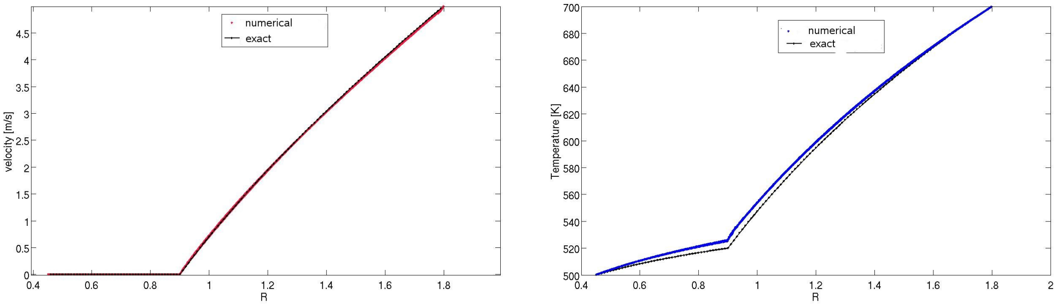
This second and last test case concerns the simulation of a highly-loaded cooled two-dimensional turbine cascade. The geometry of the blade is provided in [16] and it is known as the T106 turbine cascade which has been modified in order to add three cooling channels. The flow is subsonic, with isentropic exit Mach number equal to 0.3, inlet flow



(a)

(b)

Figure 2: Rotating flow: velocity contours (a); temperature contours (b).

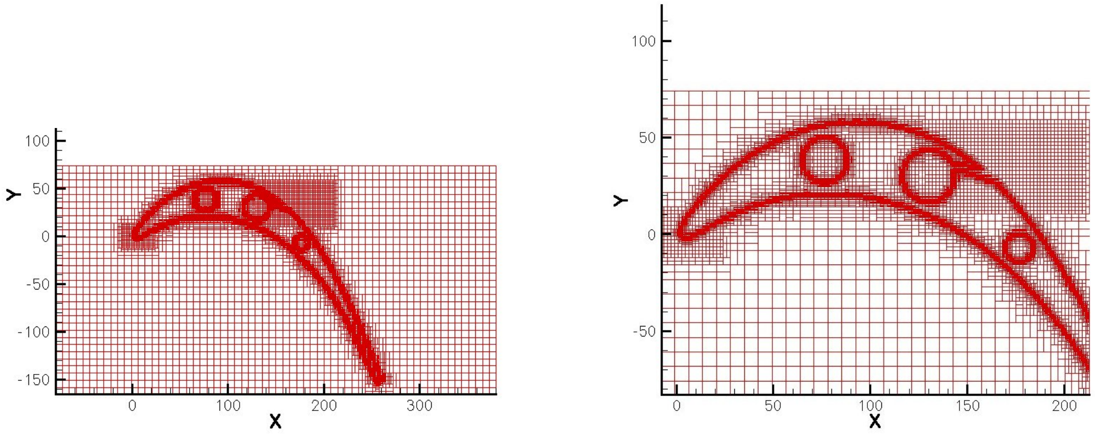


(a)

(b)

Figure 3: Rotating flow: radial velocity distribution (a); radial temperature distribution (b).

angle equal to 37.7, and Reynolds number, based on the chord length and on the exit conditions, equal to 3×10^5. Air and stainless steel are considered for the fluid and for the solid, respectively. At the inlet boundary points, the total pressure and temperature are assigned, together with the flow direction, whereas only the static pressure is prescribed at the outlet points. Three cooling holes are added to the original geometry. Two of them have assigned wall temperature, equal to $T_c = 200\ K$, whereas, cooling air with inlet temperature $T_c = 200\ K$ and inlet velocity $v_c = 5\ m/s$ flows through the main central hole. Such a cooling air issues from a secondary channel into the main flow forming a film along the suction side. In the span-wise direction, the total pressure and temperature at the cells corresponding to the inlet of the main cooling channel are imposed, together with the direction of the velocity, normal to the endwalls. Thanks to the versatility of the present IB approach, the complete geometry of the blade can be discretized easily and



(a)

(b)

Figure 4: T106 cascade: locally refined grid (a); local view of the grid (b).

efficiently. The computational grid, using about 66000 cells (33700 in the solid region), shown in figure 4(a), is refined at the leading edge of the blade, at the region of maximum curvature, and near the cooling holes, see figure 4(b). Figures 5(a) and (b) provide the computed temperature countours in the solid and in the fluid, and the velocity-vector field in and around the main central cooling channel. This test case demonstrates the capability of the present method to solve conjugate-heat-transfer problems of industrial interest.

6 CONCLUSIONS

An immersed boundary method for computing compressible viscous flows using a flexible local grid refinement technique has been extended to solve conjugate-heat-transfer problems. The proposed method has been tested versus the flow between two heated cylinders and then applied with success to predict turbomachinery blade cooling.

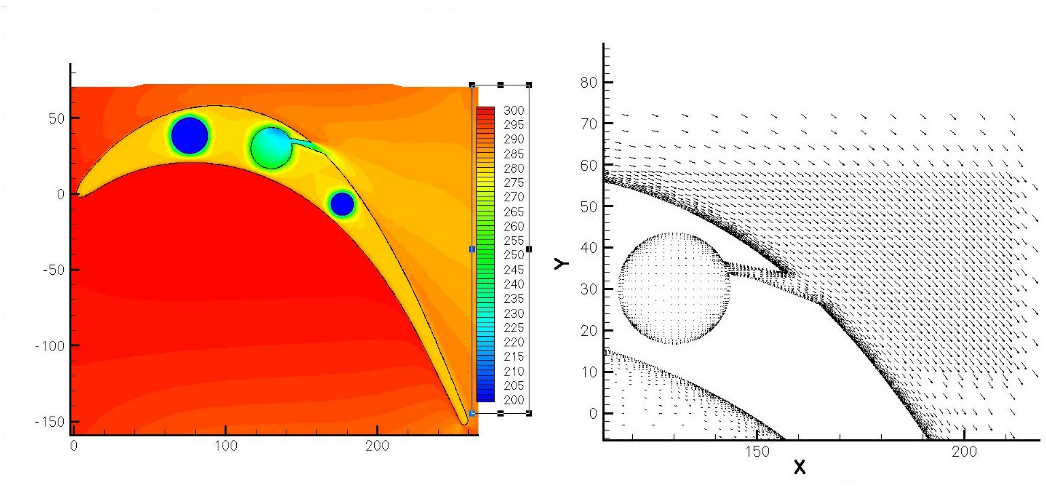
7 ACKNOWLEDGEMENTS

This work has been supported by MiUR and Politecnico di Bari, Cofinlab 2000 and PRIN-2007 grants.

REFERENCES

[1] C. S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.*, **10**, 252, (1972).

[2] J. Mohd-Yusof *Combined Immersed Boundaries/B-Splines Methods for Simulations*



(a)

(b)

Figure 5: T106 cascade: temperature contours (a); velocity vectors (b).

of Flows in Complex Geometries. CTR Annual Research Briefs, NASA Ames / Stanford University (1997).

[3] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yosuf, Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *J. Comp. Phys.*, **161**, 35 (2000).

[4] G. Iaccarino, R. Verzicco, Immersed Boundary Technique for Turbulent Flow Simulations, *Appl. Mech. Rev.*, **56**, 331-347, 2003.

[5] R. Mittal, G. Iaccarino, Immersed Boundary Methods. *Annu. Rev. Fluid Mech.*, **37**, 239 (2005).

[6] P. De Palma, M. D. de Tullio, G. Pascazio, M. Napolitano, An immersed boundary method for compressible viscous flows, *Comput. Fluids*, **35**, 693, (2006).

[7] M. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, *J. of Comp. Phys.*, **225**, 2098-2117, (2007).

[8] D.C. Wilcox, Turbulence Models for CFD, 2nd Edition, *DCW Industries, Inc.* (1998).

[9] S. Venkateswaran, C. L. Merkle. Dual time stepping and preconditioning for unsteady computations, AIAA Paper 95-0078, (1995).

[10] C. L. Merkle, Preconditioning methods for viscous ow calculations, *In Computational Fluid Dynamics Review 1995*, M. Hafez, K. Oshima Eds., Wiley, New York,419436 (1995).

[11] T. H. Pulliam and D. S. Chaussee, A diagonal form of an implicit factorization algorithm, *J. of Comp. Phys.*, **39**, 347-363, (1981).

[12] H. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Statist. Comput.*, **13**, 361 (1992).

[13] J. ORourke, Computational Geometry in C, *Cambridge University Press*, Cambridge (1998).

[14] S. Kang, An improved immersed boundary method for computation of turbulent flow with heat transfer, *Ph. D. Thesis*, Stanford University (2008).

[15] G. M. Laskowski, S. P. Kearney, G. Evans, R. Greif, Mixed convection heat transfer to and from a horizontal cylinder in crossflow with heating from below, *Int. J. of Heat and Fluid Flow*, **28**, 454-468 (2007).

[16] H. Hoheisel, R. Kiock, H. J. Lichtfuss, L. Fottner, Influence of free-stream turbulence and blade pressure gradient on boundary layer and loss behaviour of turbine cascades. *ASME J. Turbomach.*, **109**, 210219 (1987).