

FEDSM-ICNMM2010-30* - (

AN IMMERSED BOUNDARY METHOD FOR CONJUGATE HEAT TRANSFER PROBLEMS

M. D. de Tullio*

Politecnico di Bari, DIMeG and CEMeC
via Re David 200, 70125 Bari, Italy
Email: m.detullio@poliba.it

**S. S. Latorre, P. De Palma
M. Napolitano, G. Pascazio**

Politecnico di Bari, DIMeG and CEMeC
via Re David 200, 70125 Bari, Italy
Email: latorre@imedado.poliba.it, depalma@poliba.it,
pascazio@poliba.it, napolita@poliba.it

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 past a heated hollow cylinder inside a channel together with the temperature field within the cylinder and then to predict turbomachinery blade cooling.

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 a few minutes, whereas a very complicated body fitted grid may require several hours or even days. The IB technique was originally developed for incompressible flows [1–5] 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 at any value 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.

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.

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:

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

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

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

*Address all correspondence to this author.

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

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

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

$$\tilde{H} = h + \frac{1}{2} |\mathbf{u}|^2 + \frac{5}{3} k, \quad p_t = p + \frac{2}{3} \rho k; \quad (6)$$

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]:

$$\mu_t = \gamma^* \frac{\rho k}{\omega}. \quad (7)$$

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

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

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

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

where Pr and Pr_t are the laminar and turbulent Prandtl numbers equal to 0.71 and 1, 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:

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

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 Eqn. (10) in order to use a time marching approach for both steady and unsteady problems; the preconditioning matrix, Γ ,

proposed in [9, 10] is used to premultiply the pseudo-time derivative in order to improve efficiency. The final system reads:

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

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 the physical-time derivative by a second-order-accurate three-point backward difference. After some simplifications of the LHS operator which improve the efficiency of the method without affecting the residual [7], the diagonalization procedure of Pulliam and Chausee [11], followed by a factorization, provides the following final system:

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

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

Equation (12) is discretized in space using a cell-centred finite volume approach. The convective terms in 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).

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 *solid* ones are tagged as *interface fluid cells*, and the *solid cells* neighbouring *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 region. By this procedure, a uniform grid is obtained. Then, starting from such a grid, it is possible to coarsen 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 domain 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 bow-shock regions.

CONJUGATE HEAT TRANSFER

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

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

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 fluid-solid interface to enforce the equality of the solid (s) and fluid (f) temperatures and

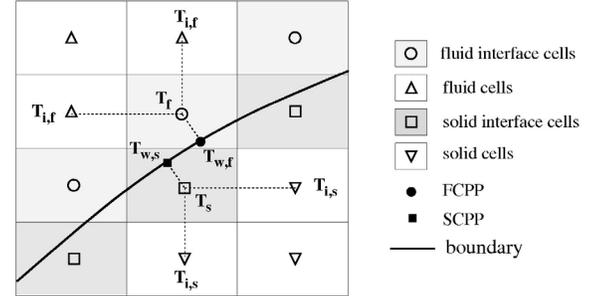


Figure 1. SCHEME FOR THE INTERFACE BETWEEN FLUID AND SOLID.

heat fluxes:

$$T_{w,f} = T_{w,s}, \quad (14)$$

$$k_f \nabla T_f \cdot \mathbf{n}_w = k_s \nabla T_s \cdot \mathbf{n}_w, \quad (15)$$

where k_f is the thermal conductivity of the fluid and \mathbf{n}_w is the unit vector normal to the wall pointing from the solid to the fluid. Eqn. (13) is discretized in space by a second-order-accurate centred difference, whereas, the physical-time derivative is approximated by a second-order-accurate three-point backward difference. The dual time stepping approach is employed also at solid points. The two boundary conditions above are implemented according to the following procedure. As proposed in [15, 16], “fluid” and “solid” interface cell centres are projected onto the body surface along the normal direction to the surface itself, so that one has “solid” cell projection points (SCPP) and “fluid” cell projection points (FCPP) which in general do not coincide, see figure (1). The heat fluxes are approximated using first-order-accurate one-sided differences,

$$\nabla T_f \cdot \mathbf{n}_w = (T_f - T_{w,f}) \beta_f, \quad (16)$$

$$\nabla T_s \cdot \mathbf{n}_w = (T_s - T_{w,s}) \beta_s, \quad (17)$$

where β_f and β_s are the inverse distances between the wall points (FCPP, SCPP) and the corresponding interface-cell centres. Then, the values of the temperatures at the “fluid/solid” interface points are approximated using the following weighted-average formulas

$$T_f = \frac{\sum_i^{N_{nbr}} \alpha_i T_{i,f} + T_{w,f} \beta_f}{\sum_i^{N_{nbr}} \alpha_i + \beta_f}, \quad (18)$$

$$T_s = \frac{\sum_i^{N_{nbr}} \alpha_i T_{i,s} + T_{w,s} \beta_s}{\sum_i^{N_{nbr}} \alpha_i + \beta_s}. \quad (19)$$

where $T_{i,f}$, $i = 1, \dots, N_{nbr}$ and $T_{i,s}$, $i = 1, \dots, N_{nbr}$ are the values of the temperature at the fluid and solid points surrounding the interface cells and α_i are the inverse distances between the surrounding cell centres and the interface cell centres. Equations. (16), (18) and (17), (19) are combined to eliminate T_f and T_s to give

$$\nabla T_f \cdot \mathbf{n}_w = \frac{\sum_i^{N_{nbr}} \alpha_i T_{i,f}}{\frac{\sum_i^{N_{nbr}} \alpha_i}{\beta} + 1} - \frac{T_{w,f}}{\frac{1}{\beta} + \frac{1}{\sum_i^{N_{nbr}} \alpha_i}}, \quad (20)$$

$$T_{w,s} = \frac{\sum_i^{N_{nbr}} T_{i,s} \alpha_i}{\sum_i^{N_{nbr}} \alpha_i} - \left(\frac{1}{\beta} + \frac{1}{\sum_i^{N_{nbr}} \alpha_i} \right) \nabla T_s \cdot \mathbf{n}_w. \quad (21)$$

Notice that equations (20) and (21) are written in a different form, for convenience. If the values $\nabla T_f \cdot \mathbf{n}_w$ and $\nabla T_s \cdot \mathbf{n}_w$ and $T_{w,f}$ and $T_{w,s}$ were computed at the same points (which are not, see figure (1)), the two boundary conditions (14) and (15), together with Eqns. (20) and (21) would provide a closed system. Here, the boundary conditions are enforced explicitly. Therefore, we compute $\nabla T_f \cdot \mathbf{n}_w$ at all FCPP points using the old values for $T_{w,f}$. We then interpolate $\nabla T_f \cdot \mathbf{n}_w$ at all SCPP points and use boundary condition (15) to evaluate $\nabla T_s \cdot \mathbf{n}_w$ at all such points. We then compute all $T_{w,s}$ values at such points from Eqn. (21). Finally, all T_f and T_s values are evaluated from Eqns. (18) and (19). It is noteworthy that such an explicit procedure could be reversed providing the same results within the accuracy of the interpolation procedure. In the first case, we interpolate the $\nabla T_f \cdot \mathbf{n}_w$ values from the FCPP points to the SCPP ones. In the second case, we would interpolate the $\nabla T_s \cdot \mathbf{n}_w$ values from the SCPP points to the FCPP ones. The first procedure is chosen because it is anticipated to be more stable since in most practical applications $k_s \gg k_f$ so that $\nabla T_f \cdot \mathbf{n}_w \gg \nabla T_s \cdot \mathbf{n}_w$. Finally, it is noteworthy that Eqns. (16) and (17) are only first-order-accurate, but these conditions involving a line/surface in 2D/3D applications, the overall second-order-accuracy of the method is maintained.

RESULTS

Flow past a heated cylinder in a channel

This test case is based on the numerical and experimental investigation of Laskowski *et al.* [14], also studied in [15, 16]. It consists in the simulation of an unsteady flow, involving natural convection, transition to turbulence and conjugate heat transfer. The test case aims at a deeper understanding of the base mechanisms of channel flow heat transfer for the design of

more efficient and compact heat exchangers, also in the field of micro-fluidic applications [17]. The experimental configuration is shown in figure 2: water flows inside a channel and a circular tube, with two coaxial surfaces, is immersed in the flow. The

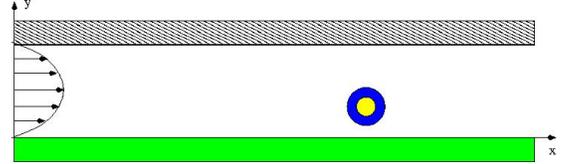


Figure 2. SCHEME OF THE FLOW PAST A HEATED CYLINDER IN A CHANNEL.

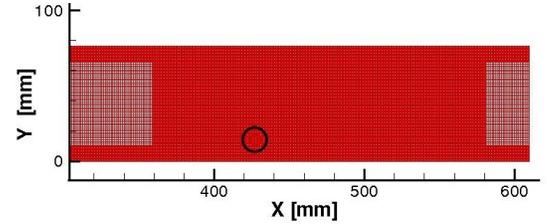


Figure 3. LOCAL VIEW OF THE GRID.

bottom wall of the channel is heated, so that the outer surface of the cylinder touches the thermal boundary layer developing along the bottom wall. Inside the tube, hot water flows heating the tube. The Reynolds number, based on the mean streamwise velocity and the channel half-width is about equal to 414. In this flow conditions, transition to turbulence is due to thermal plumes generated by the buoyancy force. A two-dimensional computation has been performed in a rectangular domain with length equal to 61 cm and height equal to 7.62 cm. The inner and outer radii of the tube are equal to 0.635 cm and 1.5875 cm, respectively, its centre having coordinates 42.7 cm and 1.43 cm in the streamwise and wall-normal directions. The inlet velocity profile is provided in [14] and the inlet temperature of the water is equal to 284 K. The bottom wall of the channel is maintained at 318 K whereas the top wall is adiabatic. Finally, the temperature of the inner surface of the tube is set to 320.5 K. At walls, the no-slip condition is enforced and at outlet points the convective outflow condition is employed. Material properties are the ones for water and stainless steel. The acceleration of gravity in the wall normal direction has been considered equal to 9.81 m/s^2 . A Cartesian mesh with about 19100 (15000 for discretizing the tube) cells has been employed. The mesh is locally refined near the wall and the tube, as shown by the local view provided in figure 3. A snapshot of the temperature contours obtained using the present approach are given in figure 4(a), whereas figure 4(b) provides the

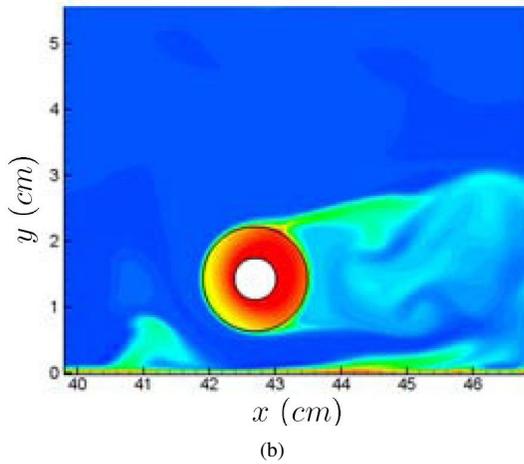
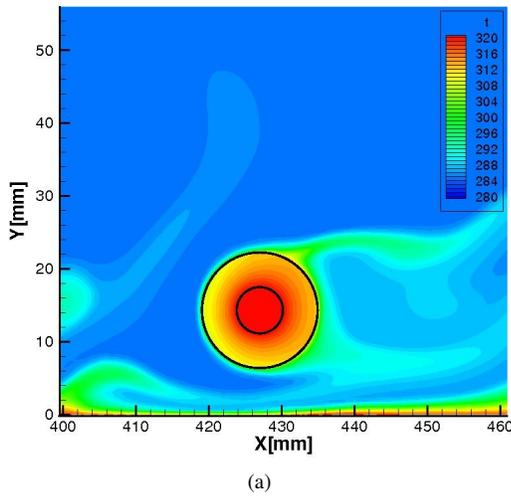


Figure 4. LOCAL VIEW OF THE TEMPERATURE CONTOURS: PRESENT RANS SIMULATION (a); LES FROM REFERENCE [15] (b).

results of the large eddy simulation performed in [15]. The qualitative comparison between the two results demonstrates that the present unsteady RANS simulation can capture the main structures developing in the flow. In particular, the effect of the thermal plumes, generated by the heated wall, on the temperature distribution over the cylinder surface is predicted with reasonable accuracy. Figure 5 provides the time averaged heat flux over the outer surface of the tube. The angle θ is measured from the trailing edge in the counter-clockwise direction. The numerical results obtained by the present method are compared with the experimental data of [14]. Also in this case, the comparison indicates that the numerical prediction is in good agreement with the real dynamics of the flow.

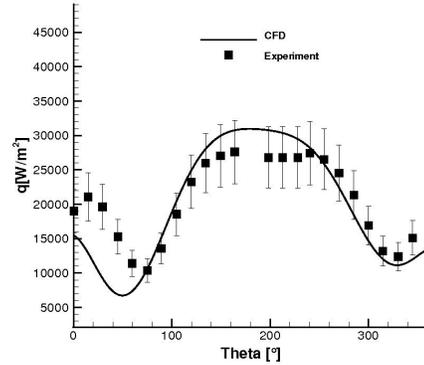
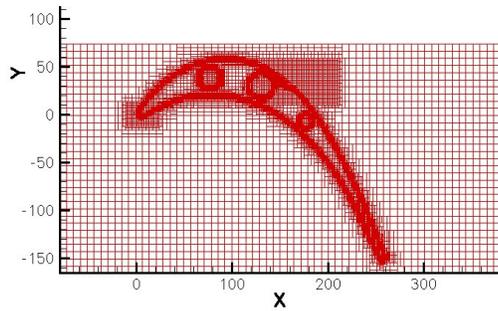


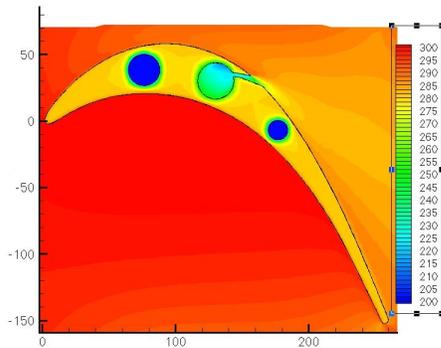
Figure 5. HEAT FLUX DISTRIBUTION OVER THE OUTER SURFACE OF THE TUBE.

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 [18] 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 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 $T_c = 200\text{ K}$, whereas, cooling air with inlet temperature $T_c = 200\text{ K}$ and inlet velocity $v_c = 5\text{ m/s}$ 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 efficiently. The computational grid, using about 66000 cells (33700 in the solid region), shown in figure ??(a), is refined at the leading edge of the blade, at the region of maximum curvature, and near the cooling holes, see figure ??(b). Figures ??(a) and (b) provide the computed temperature contours 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.



(a)



(b)

Figure 6. COOLED T106 CASCADE: LOCALLY REFINED GRID (a); TEMPERATURE CONTOURS (b).

CONCLUSIONS

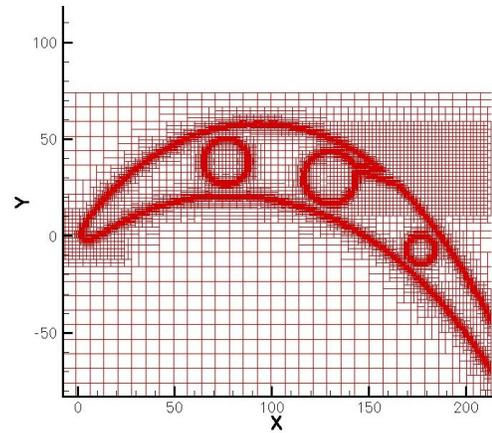
An immersed boundary method for computing compressible viscous flows has been improved and applied to solve complex turbomachinery flows. The method has been extended to solve conjugate-heat-transfer problems with a flexible local grid refinement technique. The proposed method has been applied with success to solve complex flow fields, even at high values of the Reynolds number, as well as to predict turbomachinery blade cooling.

ACKNOWLEDGMENT

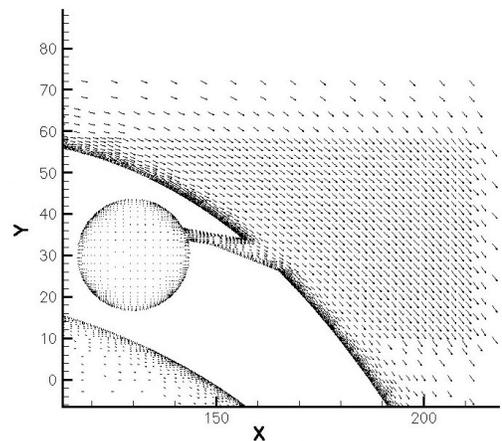
This work has been supported by the MiUR and the Politecnico di Bari, Cofinlab 2000 and by PRIN-2007 grants.

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 of Flows in Complex Geometries*.



(a)



(b)

Figure 7. T106 CASCADE: LOCAL VIEW OF THE GRID (a); VELOCITY VECTORS (b).

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. Comput. 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. Comput. 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 flow 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. Comput. 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. O'Rourke, Computational Geometry in C, *Cambridge University Press*, Cambridge (1998).
- [14] 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).
- [15] S. Kang, An improved immersed boundary method for computation of turbulent flow with heat transfer, *Ph. D. Thesis*, Stanford University (2008).
- [16] S. Kang, G. Iaccarino, F. Ham, DNS of buoyancy-dominated turbulent flows on a bluff body using the immersed boundary method, *J. Comput. Phys.*, **228**, 3189 (2009).
- [17] M. Meis, F. Varas, A. Velázquez, J. M. Vega, Heat transfer enhancement in micro-channels caused by vortex promoters, *Int. J. of Heat and Mass Transfer*, **53**, 2940 (2010).
- [18] 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).