



Efficient Fluid Dynamic Design Optimization Using Cartesian Grids

A. Dadone
Politecnico di Bari, Italy

B. Grossman
National Institute of Aerospace, Hampton, Virginia

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- Email your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Telephone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/CR-2004-213036
NIA Report No. 2004-07



Efficient Fluid Dynamic Design Optimization Using Cartesian Grids

A. Dadone
Politecnico di Bari, Italy

B. Grossman
National Institute of Aerospace, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Cooperative Agreement NCC-1-02043

December 2004

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

Efficient Fluid Dynamic Design Optimization Using Cartesian Grids

A. Dadone*

Politecnico di Bari, 70125 Bari, Italy

and

B. Grossman†

National Institute of Aerospace, Hampton, VA 23666

ABSTRACT

This report is subdivided in three parts. The first one reviews a new approach to the computation of inviscid flows using Cartesian grid methods. The crux of the method is the curvature-corrected symmetry technique (CCST) developed by the present authors for body-fitted grids. The method introduces ghost cells near the boundaries whose values are developed from an assumed flow-field model in vicinity of the wall consisting of a vortex flow, which satisfies the normal momentum equation and the non-penetration condition. The CCST boundary condition was shown to be substantially more accurate than traditional boundary condition approaches. This improved boundary condition is adapted to a Cartesian mesh formulation, which we call the *Ghost Body-Cell Method* (GBCM). In this approach, all cell centers exterior to the body are computed with fluxes at the four surrounding cell edges. There is no need for special treatment corresponding to *cut cells* which complicate other Cartesian mesh methods. The merits of the procedure are indicated by the computation of the compressible flow about basic circular cylinders as well as of airfoil applications and detailed comparisons to body-fitted grid computations. The results show the surface non-penetration condition to be satisfied in the limit of vanishing cell size and the method to be second-order accurate in space. In addition, the results are independent of the position of the body with respect to the grid. The flexibility of the approach is illustrated with an application to the flow over a multiple-body configuration.

The original version of the Ghost Body-Cell Method, had the typical Cartesian grid requirement that any grid clustering near the body must be maintained to the far field boundary. This is not a significant penalty in 2D calculations, but it may become significant in 3D calculations. As a consequence, in the second part of the paper, we have introduced a far-field grid coarsening, based on the *ibanking approach*, which maintains the structured nature of the grid while computing only the active cell centers. The far-field grid coarsening has been applied to subsonic and transonic flows about an NACA 0012 airfoil, indicating that the results are practically unaffected by the coarsening. Meanwhile, the number of the computed cells is reduced to 40% – 50% and the CPU time to 34% – 47% of the original CPU time, depending on the particular test case.

The GBCM on far-field coarsened Cartesian grids appears to be particularly well-suited for design optimization problems. The third part of the paper is dedicated to this issue and presents an efficient formulation for the robust inverse design of compressible fluid flow problems on far-field coarsened Cartesian grids. The design approach has five essential features: First, the procedure utilizes the GBCM on coarsened Cartesian grids. Second, approximate but efficient design sensitivities are obtained using a discrete adjoint formulation, based on an artificially dissipative flow solver in order to obtain robust sensitivity derivatives in the presence of potentially noisy or non-smooth objective functions. The third feature is the implementation of quasi-time-dependent adjoint equations. Next, the procedure involves what we term *progressive optimization*, whereby a sequence of operations is performed containing a partially converged flow solution, followed by a partially converged adjoint solution followed by an optimization step. The final feature is the use of progressively finer grids for the progressive solution of the flow field, while only using a reduced domain for the solution of the adjoint equations.

The inverse design of airfoils in transonic flow conditions indicated that the method is accurate, robust and highly efficient, with converged design optimizations produced in no more than the amount of computational work to perform two flow analysis on the finest mesh. In addition, a test case involving inverse design of a NACA 0012 airfoil in subsonic flow conditions showed that the approach is able to bring the convergence of the optimization process to machine zero.

*Professor, Dipartimento di Ingegneria Meccanica e Gestionale, Sezione Macchine ed Energetica.

†Vice President, Education & Outreach. Also Prof. Aero. & Ocean Eng., Virginia Tech.

INTRODUCTION

The continuous growing power of computers is encouraging engineers to rely on computational fluid dynamics (CFD) for the design and testing of new technological solutions. Numerical simulations allow the analysis of the phenomena without resorting to expensive prototypes and complex experimental measurements. On the other hand, while simple geometries discretized by regular grids are efficiently handled by codes and hardware, flows with complex geometries requiring body-fitted grids are still challenging problems for today computers. The grid generation process, whether for composite block structured grids or unstructured grids is often laborious and is not always as automated as desired. The conceptual design process often requires detailed body-fitted grids over wide classes of different geometric topologies. Grid generation for complex topologies often takes weeks or even longer to develop and this becomes a major impediment to using CFD in design. Further complications may occur due to moving boundaries or problems which require continuing regeneration or deformation of grids.

One approach to develop a simplified alternative to complex grid generation involves the use of Cartesian grid methods. In these methods the body does not coincide with grid surfaces or grid volume edges. Instead, the body is immersed in the grid and the complication of generating a body-fitted grid is exchanged for complications associated with the implementation of the surface boundary conditions. Furthermore, since the body surfaces are not aligned with the grid it is nearly impossible to adequately resolve high Reynolds number boundary layers with Cartesian mesh methods. Nonetheless, there are many classes of flow problems which benefit from the efficiency of Cartesian mesh approaches.

Cartesian mesh methods have their origin in transonic potential flow computations^{1,2}. Early Cartesian Euler computations were performed in Refs. 3–5. More recent computations^{6–8} involved the use of grid adaptation and have produced very good results on some very complex configurations. Aftosmis and Berger⁹ have also shown that Cartesian mesh computations may be characterized by a truncation error which is approximately one order of magnitude smaller than truncation error of computations on triangular or quadrilateral body-fitted grids.

For completeness, we mention two other approaches which hold promise for avoiding grid generation difficulties. One is a Cartesian grid method which simulates viscous flows using vorticity confinement¹⁰ and the other is an approach in which a body force term is used to simulate moving boundaries^{11,12}.

Cartesian grid finite-difference schemes for computing fluid dynamic problems, *e.g.*, Ref. 5, have proven to be quite efficient, but require a complex topological discrimination of the cells located near the body profile. Cartesian grid finite-volume methods, *e.g.* Ref. 13, are more straightforward in that they allow to use composite elements (cut cells) determined by the intersection between the Cartesian grid and the body profile. On the other hand, these methodologies may suffer stability problems when such composite elements become very small.

The use of Cartesian grids without composite elements avoids this stability problem. On the other hand, it significantly increases the problem of enforcing the non-penetration condition at solid walls immersed in the Cartesian grid. For inviscid flows, the simple non-penetration condition can be easily enforced at solid walls if body-fitted grids are used, because the grids are aligned with the wall. For Cartesian grids, appropriate flow conditions must be enforced at the cell centers surrounding the body. The accuracy of the computed flow solution obviously depends on how accurately the flow conditions are enforced at these cell centers. As a consequence, an accurate model of the flow near the solid wall is required if accurate flow solutions on Cartesian grids are sought.

Recently, a model able to accurately represent inviscid flows near the solid walls has been introduced by the present authors¹⁴. It has been named CCST (*curvature-corrected symmetry technique*) and has been developed for body-fitted grids. The method considers solid walls as boundaries immersed in the flow field and enforces conditions at ghost cell centers located inside the body in a position close to the wall. The ghost cell values are developed from an assumed flow-field model in the vicinity of the wall consisting of a vortex flow, which satisfies the normal momentum equation and the non-penetration condition. This flow-field model enforces symmetry conditions for entropy and total enthalpy along a normal to the body surface. Then, the flow computations are performed using these ghost cell centers without taking into account the presence of the wall. Obviously, the fluid dynamic parameters are continuously updated at these ghost cell centers in order to enforce the wall impermeability condition at each stage of the computation.

The CCST boundary condition has been shown to be significantly more accurate for a variety of inviscid flows and has been shown to be at least one order of magnitude more accurate in inviscid two-

dimensional flows¹⁴ as well as in inviscid three-dimensional flows¹⁵. Other researchers have successfully implemented this boundary condition, *e.g.* Ref. 16 have reported excellent results using the CCST in a preconditioned Euler code.

The first part of the present paper is dedicated to the review of an innovative and accurate numerical technique to enforce non-penetration conditions at solid walls in Cartesian grid computations. This new approach utilizes the curvature corrected symmetry technique applied to Cartesian grids, which we call the *Ghost Body-Cell Method* (GBCM). This method, in some of our earlier papers, was called the *Immersed Body Methodology*^{17,18}. This approach uses only rectangular cells without having to define *cut cells* in the vicinity of the solid boundary. We first describe the governing equations and the classical boundary conditions to enforce the non-penetration condition at solid walls for body-fitted grids. Then the improved boundary condition called the curvature-corrected symmetry technique (CCST) is presented. Finally, the CCST is extended to model the flow near solid walls immersed in a Cartesian grid, GBCM. Details related to sharp corners and trailing edges are also described. The merits of the procedure are indicated by the computation of the compressible flow about basic circular cylinders as well as of airfoil applications and comparisons to body-fitted grid computations. The flexibility of the method is illustrated with an application to the flow over a multiple-body configuration.

In the aforementioned applications, the method is shown to produce accurate results for elementary two-dimensional geometries with relatively coarse grids. Unfortunately, in usual Cartesian grid formulations, any grid clustering near the body must be maintained to the far field boundary. This is not a significant penalty in 2D calculations, but it becomes significant in 3D calculations. As a consequence, the second part of the paper is dedicated to far-field grid coarsening to address this issue and improve the method's efficiency. Ref. 19 provides a basic approach to local refinement of structured grids. The suggested procedure is based on the idea of *ibanking*. The *ibanking* concept was introduced in Ref. 20 as a device to insert geometry into a structured grid by extending the grid inside the body and then blanking out the interior portion. In Ref. 19, the *ibanking* algorithm is extended to a coarsened structured grid, so that the blanked region is skipped, requiring neither storage of variables nor solution of equations. In such a way, incompressible Navier-Stokes equations in Ref. 19, were efficiently solved using a finite-difference method on a Cartesian grid. In the second part of the present paper, the *ibanking* approach is extended to the present finite-volume formulation. Results for airfoils will be presented with and without far-field coarsening and compared in terms of accuracy and efficiency.

The Ghost Body-Cell Method on coarsened Cartesian grids appears to be particularly well-suited for design optimization problems. The third part of the paper is dedicated to the presentation of the efficient design procedure that we utilize. This approach was originally developed for body-fitted grids²¹ and is adapted here for Cartesian grids. It has five essential ingredients. First, the procedure utilizes an accurate finite-volume method on coarsened Cartesian grids (GBCM) in order to accurately evaluate the objective function. Next, approximate design sensitivities are efficiently obtained using a discrete adjoint formulation, based on a simplified artificially dissipative flow solver in order to obtain robust sensitivity derivatives in the presence of noisy or non-smooth objective functions. This adjoint solver is only used for the solution of robust sensitivities. The third ingredient of the procedure is the use of quasi-time-dependent adjoint equations. The fourth ingredient of the procedure involves a *progressive optimization*, whereby a sequence of operations, containing a partially converged flow solution, followed by a partially converged adjoint solution followed by an optimization step is performed. Both the flow and the adjoint equation solutions are performed on Cartesian grids with body wall boundaries immersed in the Cartesian grid. Furthermore, the adjoint equations are solved on a reduced domain compared to the one used to compute the flow solution. The fifth ingredient is the use of progressively finer grids for the progressive solution of the flow field, while only using a reduced domain for the solution of the adjoint equations. In the third part of the paper the optimization procedure will be presented together with some computed results for the inverse design of airfoils.

1. GHOST BODY-CELL METHOD

1.1 Governing Equations

The vector form of the two-dimensional Euler equations in a Cartesian coordinate system can be written as:

$$\mathbf{q}_t + \mathbf{f}_x + \mathbf{g}_y = 0, \quad (1)$$

where the vector of conserved variables is given by:

$$\mathbf{q} = [\rho, \rho u, \rho v, \rho e_0]^T, \quad (2)$$

and the inviscid flux vectors are:

$$\begin{aligned} \mathbf{f} &= [\rho u, p + \rho u^2, \rho uv, \rho u h_0]^T, \\ \mathbf{g} &= [\rho v, \rho uv, p + \rho v^2, \rho v h_0]^T, \end{aligned} \quad (3)$$

where p , ρ , u , v are the pressure, the density and the two Cartesian components of the velocity vector, respectively, while e_0 and h_0 represent the total energy and the total enthalpy per unit mass, respectively.

In the work presented here, a quadrilateral, cell-centered finite-volume method has been implemented with a flux-difference-splitting algorithm²² as the flow solver. A MUSCL-type extrapolation, with a formal second-order accuracy in space, has been applied to extrapolate the physical variables onto the left and right sides of each cell edge. However, the methods developed in this paper is applicable to other flow solvers.

1.2 Wall Boundary Conditions

Classical non-penetration condition

We consider a cell-centered finite-volume representation of Eqs. (1) and focus our attention on a quadrilateral cell of the computational volume. The flux vector across a cell edge can be expressed as:

$$F = [\rho \tilde{v}, n_x p + \rho u \tilde{v}, n_y p + \rho v \tilde{v}, \rho h_0 \tilde{v}]^T, \quad (4)$$

where n_x and n_y are the direction cosines of the normal to the cell edge and \tilde{v} is the corresponding velocity component:

$$\tilde{v} = n_x u + n_y v. \quad (5)$$

Assume that this cell face is located on a solid boundary so that $\tilde{v} = 0$. We see from Eq. (4) that the only quantity required to compute the flux vector at this cell edge is the pressure at the wall. For a cell-centered scheme, there are several conventional ways to evaluate this pressure. The most common approach is to extrapolate from the interior of the flow field. A first-order pressure extrapolation is often used whereby the wall pressure is taken as the value associated with the nearest cell center. A second-order pressure extrapolation (P-II) is also sometimes used with the wall pressure linearly extrapolated from the two nearest cell centers:

$$p_w = \frac{3}{2}p_1 - \frac{1}{2}p_2, \quad (6)$$

where subscripts 1 and 2 refer to these cell centers (Fig. 1). In another approach, the pressure may be extrapolated to satisfy the normal momentum equation. Yet another boundary condition procedure utilizes a characteristic boundary condition. A description and comparison of these techniques together with their application to a shock reflection problem are reported in Ref. 14.

CCST: a more accurate non-penetration condition

The curvature corrected symmetry technique (CCST) described in Ref. 14 utilizes *ghost points* and a vortex model in the vicinity of the body surface. The procedure consists in first locating two rows of cell centers closest to the body surface inside the flow domain (points 1 and 2 in Fig. 1). Then two extra rows of cells are inserted (image cells or ghost cells) outside the computational flow field which are created by reflecting the internal cells 1 and 2 symmetrically with respect to the body surface (points -1 and -2 in Fig. 1). The fluid dynamic variables at the ghost cell centers are utilized to implement the non-penetration condition at the wall. The ghost cell values are developed from an assumed flow-field

model in vicinity of the wall, consisting of a vortex flow with locally symmetric distribution of entropy and total enthalpy along a surface normal. This flow model satisfies the normal momentum equation, the non-penetration boundary condition and enforces antisymmetric normal derivatives $\partial S/\partial n$ and $\partial h_0/\partial n$ along the normal to the body surface in the vicinity of the body. These entropy and total enthalpy distributions will produce zero normal derivatives when the flow is irrotational and will satisfy Crocco's relationship even when vorticity is present near the surface. The method has shown to produce superior accuracy¹⁴ compared to the classical surface boundary condition procedures discussed in the previous subsection.

The pressure at the image cell centers is obtained from the integration of the normal momentum equation (for a steady inviscid flow):

$$p_{-i} = p_i - \rho_i \frac{\tilde{u}_i^2}{R_i} \Delta n_i, \quad (7)$$

where the subscript i is equal to 1 or 2, Δn_i indicates the distance between the cell centers (i) and ($-i$), and \tilde{u}_i is the velocity component tangential to the body:

$$\tilde{u} = n_y u - n_x v. \quad (8)$$

Note that R_i is the signed local radius of curvature of the wall, positive if the center of curvature is on the same side of the body, negative if the center of curvature is on the side of the flow field.

The density and the velocity component tangential to the body are obtained from our vortex model:

$$\begin{aligned} \rho_{-i} &= \rho_i \left(\frac{p_{-i}}{p_i} \right)^{1/\gamma}, \\ \tilde{u}_{-i}^2 &= \tilde{u}_i^2 + \frac{2\gamma}{\gamma-1} \left(\frac{p_i}{\rho_i} - \frac{p_{-i}}{\rho_{-i}} \right) + \tilde{v}_i^2 - \tilde{v}_{-i}^2, \end{aligned} \quad (9)$$

where the subscript i is equal to 1 or 2 and γ is the constant ratio of specific heats.

The velocity component normal to the body is obtained from the non-penetration condition, following Ref. 14:

$$\begin{aligned} \tilde{v}_{-1} &= -\tilde{v}_1, \\ \tilde{v}_{-2} &= 3\tilde{v}_{-1} + \\ &\quad 2 [\tilde{v}_1 + (\tilde{v}_1 - \tilde{v}_2)/2] \sqrt{\frac{\rho_1 + (\rho_1 - \rho_2)/2}{\rho_{-1} + (\rho_{-1} - \rho_{-2})/2}}. \end{aligned} \quad (10)$$

Numerical experiments have also shown that \tilde{v}_{-2} can be approximated by a value antisymmetric to \tilde{v}_2 . Then Eqs. (10) can be simplified as follows:

$$\tilde{v}_{-i} = -\tilde{v}_i, \quad (11)$$

where the subscript i is equal to 1 or 2.

In Ref. 14 we show dramatic advantages of the CCST over conventional methods with regard to numerical entropy generation, total pressure loss and grid convergence.

1.3 Cartesian Grid Method

In this section, the curvature corrected symmetry technique (CCST) will be extended to non-body fitted Cartesian grids. The method is developed for two-dimensional flows, but may be extended to three dimensions. The new numerical technique, the *Ghost Body-Cell Method*, is described by the following steps:

1. The body is defined by a dense, ordered sequence of (x, y) coordinate pairs. The definition of the surface is completely independent of the mesh.
2. A Cartesian grid is selected and the computational flow field is subdivided into finite-volume cells.

3. The cell centers are discriminated between those located at the interior of the body and those located externally. The internal cell centers are flagged with $\lambda = 0$, while the external ones are flagged with $\lambda = 1$.
4. The row of cell centers closest to the body and located at the interior of the body is singled out. These cell centers are indicated by circles in Fig. 2 which presents a part of the cell center net and the curved body profile (the right-hand side of the solid line is interior to the body).
5. A second row of cell centers located at the interior of the body and adjacent to the first row is singled out. These cell centers are indicated by triangles in Fig. 2.
6. For each of the interior cell centers of the two rows described above (ghost cell centers), a corresponding symmetric point is determined at a location exterior to the body and *reflected symmetrically* with respect to the body surface. To locate these symmetric points, we first determine the straight line passing through the cell center and orthogonal to the body. On this line, the point symmetric of the interior cell center with respect to the body is determined. These symmetric points are indicated by dark squares in Fig. 2. As an example, the point B and the interior cell center A in Fig. 2 are symmetric with respect to the body.
7. The four cell centers surrounding each symmetric point are determined. As an example, points C, D, E, F are the cell centers surrounding point B in Fig. 2. Note that one or even two of these four cell centers may be located inside the body.
8. The value of the conserved variables \mathbf{q} at the point B are determined by a bilinear interpolation:

$$\mathbf{q}_B = \frac{k_1(x - x_C)(y - y_C) + k_2(x - x_C) + k_3(y - y_C) + k_4}{k_1(x - x_C)(y - y_C) + k_2(x - x_C) + k_3(y - y_C) + k_4} \quad (12)$$

where the constants k_i ($i = 1, \dots, 4$) are determined by the values of the conserved variables \mathbf{q} at the cell centers C, D, E, F. The bilinear interpolation is also used in situations when one or two of the cell centers surrounding point B are located inside the body surface. The internal values are determined from the CCST boundary condition (described next).

9. The values of the physical variables at the cell centers belonging to the previously defined two rows of internal cell centers are determined according to the CCST boundary condition, Eqs. (7), (9), (11). As an example, the physical variables at the cell center A are given by:

$$p_A = p_B - \rho_B \frac{\tilde{u}_B^2}{R} \Delta n \quad (13)$$

$$\rho_A = \rho_B \left(\frac{p_A}{p_B} \right)^{1/\gamma} \quad (14)$$

$$\tilde{u}_A^2 = \tilde{u}_B^2 + \frac{2\gamma}{\gamma - 1} \left(\frac{p_B}{\rho_B} - \frac{p_A}{\rho_A} \right) \quad (15)$$

$$\tilde{v}_A = -\tilde{v}_B \quad (16)$$

In Eqs. (13)–(16), Δn is the distance between points A and B, while R is the signed local radius of curvature of the body.

10. The time-dependent computation of all conserved variables at cell-centers located external to the body now proceeds without any other boundary condition. Each cell center utilizes the fluxes at surrounding cell edges without further approximation. Cell centers located near the boundary utilize the interior (ghost) cells.

The time dependent computation of the flow field starts from the undisturbed flow condition enforced at all the cell centers located outside of the body, while stagnation flow conditions are enforced inside the body. In addition, the computed residuals are multiplied by λ . These assumptions are irrelevant to the flow computation and have been used in order to avoid any computational problem at the cell centers located inside the body. In addition, the last assumption enables a check of the variations of the residuals, which would never converge if the residual of the cell centers located inside the body are taken into account.

A special procedure based on double-valued cell centers is required near unresolved thin surfaces and sharp corners, such as at the sharp trailing edge of an airfoil. At sharp corners, one cell center inside the geometry may be the ghost cell center for one side of the corner surface as well as for the other side. Also, a ghost cell center pertaining to one side of the corner surface may be located in the flow field on the other side of the sharp corner. One typical situation is shown in Fig. 3, where the first row of cell centers are again represented by circles and the second row by triangles. Note that point R has a symmetric point on the upper side of the trailing edge and a second symmetric point on the lower side. A similar situation holds for point U. In addition, point S belongs to the second row of ghost cell centers and has a symmetric point on the lower side of the trailing edge, while cell center T has a symmetric point located on the upper side. As a consequence, the cell centers R, S, T, U will be double-valued. The procedure we use in this region is described by the following steps:

1. The sharp corner region is defined by detecting the body region where at least one of the ghost cell centers pertaining to one side of the sharp corner does not differ from the ghost cell centers pertaining to the other side or is located inside the flow field. In Fig. 3, this region includes points R, S, T, U. Note that cell centers V, W, Q, Z do not belong to this region because the first two have symmetric points on the lower side and the other two cell centers have symmetric points on the upper side so that no cell center is double-valued.
2. The fluid dynamic properties at the ghost cell centers are computed by applying the previous procedure. The cell centers R and U, located inside the body, are characterized by two different sets of fluid dynamic properties evaluated by applying the previous procedure to both sides of the trailing edge. Also, the cell centers S and T, located in the flow field, are characterized by a first set of properties corresponding to the flow field and by a second set computed by applying the previous procedure to the lower and to the upper side of the trailing edge, respectively.
3. After computing the fluxes in a traditional manner, we recompute the fluxes involving the double values at cell centers R, S, T, U.
4. We recompute the fluxes downstream of the trailing edge, involving the points R and U, by averaging the two sets of values at these points.

In GBCM, all cell centers exterior to the body are computed with fluxes at the four surrounding cell edges. There is no need for special treatment corresponding to *cut cells* which complicate other Cartesian mesh methods. Special treatment for the evaluation of "ghost cell centers" as well as surface normals and radii of curvature are required. Since the body is represented pointwise and is independent of the mesh, it may be easily defined by a sufficient point density to allow straightforward accurate numerical calculation of the surface normals and radii of curvature. For the present test cases, the description of the body profile with 200 points, connected by segments, produces a sufficiently accurate calculation of Δn and R .

1.4 GBCM Results

First, the subsonic compressible inviscid flow over a circular cylinder has been analyzed to evaluate the merits of the Ghost Body-Cell Method for Cartesian grids. A free-stream Mach number of $M_\infty = 0.38$ has been prescribed. This flow condition produces a maximum Mach number of approximately 0.92 on the surface of the cylinder. The computations have been performed using three Cartesian grids: 110×110 , 56×56 , and 28×28 . The numbers of cell centers located in the first row surrounding the body on the internal side were 128, 64 and 32, respectively. The far-field boundary was located at approximately 20 diameters.

Three parameters have been used to evaluate the effectiveness of the Cartesian grid technique. These include the L_2 -norm of the normalized surface velocity component normal to the body, $\langle \tilde{v}_b \rangle$, the Mach number at the leading edge, M_{le} , and the Mach number at the trailing edge, M_{te} . These three parameters are plotted in Fig. 4 versus $1/N^2$, where N is the number of cell centers surrounding the body on the internal side. In order to generate the data for the curves in Fig. 4, the three parameters have been computed by interpolating the conserved variables at the body points from the four surrounding cell centers. Note that one or even two of the four cell centers may be located inside the body. Fig. 4 indicates that GBCM enforces the non-penetration condition in the limit of vanishing cell size, and appears to be second-order accurate in space. The errors in Fig. 4 are very small (of the order of 10^{-4}) when the finest grid is employed. Note that the free-stream velocities are scaled to 1.

The flow has been also computed using a body-fitted polar grids with 128×32 cells, with the larger number corresponding to the number of cells located in the circumferential direction. The wall non-penetration condition has been enforced with a classical second-order pressure extrapolation technique (P-II). The number of cell centers surrounding the body on the internal side is the same for polar grid and the corresponding Cartesian grid. The results with the Cartesian grids (using GBCM) and with the body-fitted polar grids have been compared. Figs. 5 and 6 present the contours of the normalized total pressure. The contours of total pressure found using CCST methodology with the Cartesian-grid technique are very regular, with a very small total pressure increase, corresponding to a maximum error equal to 0.2%, quite close to the maximum error found with the CCST on polar grids (see Ref. 14). In contrast, the P-II technique yields significant total pressure errors, with a maximum equal to 1.4%. Figs. 5 and 6, together with other results reported in Ref. 17, indicate that the use of the Ghost Body-Cell Method on Cartesian grids preserves the superior accuracy of the curvature-corrected symmetry technique on body-fitted polar grids and shows dramatic advantages with respect to the widely used second-order pressure extrapolation technique on body-fitted polar grids.

Next, more general two-dimensional airfoil geometries have been considered. Airfoils are of course an important class of geometries, but they also serve as a test of the method for sharp corners, such as near sharp trailing edges. For these cases, the double-valued ghost-cell-center procedure for sharp corners, outlined at the end of §1.3, has been applied. Results have been obtained for a NACA 0012 airfoil. Various Mach numbers and angles of attack, corresponding to subsonic as well as transonic flow conditions have been considered. Here we present one of the transonic results, corresponding to a free-stream Mach number $M_\infty = 0.8$ at an angle of attack equal to 1.25° . Fig. 7 presents the computed Mach number distribution on the airfoil surface (continuous lines) compared with the body-fitted grid Mach number distribution (symbols) obtained using the CCST technique. The two sets of results in Fig. 7 agree quite well. The computed lift and drag coefficients are: $C_L = 0.3535$, $C_D = 0.0232$. These values compare quite well with the results obtained by Salas and South (as discussed in Ref. 23, where their values $C_L = 0.3474$, $C_D = 0.0221$ are described as accurate.

The influence of the airfoil leading edge position with respect to the grid has also been investigated. A transonic test case with $M_\infty = 0.85$ at zero incidence has been considered. The flow has been first computed locating the leading edge precisely at a cell center, which will be considered as the reference test case in the following. Then four more computations have been performed by moving the leading edge by -50% , -25% , 25% , 50% of the local cell dimension in the x direction. Fig. 8 shows the Mach number distribution in the leading-edge region for three of these computations: the continuous line corresponds to the leading edge located coincident with the cell center, the circles indicate the results corresponding to the leading edge located 50% upstream, and finally the **X** symbols correspond to the leading edge located 50% downstream. Note that the abscissa is measured from the leading edge position. The position of the airfoil profile with respect to the cell center net is also shown in the small window in Fig. 8, using the same symbols already used for the Mach number distribution. Note the position change of many cell centers when changing the location of the leading edge with respect to the cell center net: many cell centers located inside the body are moved to the flow field and vice versa so that the geometrical configuration presents large changes. Nevertheless, the three sets of Mach number distributions coincide extremely well despite the effects of the curvature in this region and the changes in the geometrical configuration. This finding conclusively indicates that the method is practically independent of the airfoil profile location with respect to the grid. In order to reenforce this conclusion, Table 1 presents the mean square difference between the surface values corresponding to the different grid locations for some fluid-dynamic properties (pressure, density, Mach number, and the two velocity components). Also shown in Table 1 is the mean square error of the normalized total temperature on the body surface, $\langle T^\circ - 1 \rangle$, and of the normalized

velocity component normal to the body surface, $\langle \tilde{v} \rangle$, together with the lift, C_L , and drag coefficients, C_D , for the five prescribed locations of the leading edge. Note that Table 1 contains interpolated values which are used only for assessment purposes. Table 1 quantifies the effects of moving the leading edge with respect to the grid and establishes that the results are practically independent of the airfoil profile location with respect to the grid. This table also indicates that the normalized total temperature errors are very small (of the order of 10^{-3}) as well as the normalized velocity components normal to the body (of the order of 10^{-4}). This result is important in order to effectively utilize this method for application involving design optimization.

In order to show that the Ghost Body-Cell Method can be easily extended to multiple body configurations, the external/internal flow over the so called bi-NACA 0012 double airfoil is considered. The test case has been suggested for the GAMM workshop²⁴ on the numerical simulation of compressible Euler flows held at INRIA on June 1986. The geometry of this test case, plotted in Fig. 9, is made up of two NACA 0012 airfoils separated by a distance equal to 50% of the airfoil chord. Three flow conditions have been suggested in Ref. 24: a subsonic flow condition, a transonic flow with an internal shock, and a transonic flow with internal and external shocks. This last flow condition corresponds to an upstream Mach number $M_\infty = 0.55$ at an angle of attack of 6° and has the most interesting flow features. Indeed, on the upper side of both airfoils, there is a supersonic expansion due to the curvature of the leading edge, terminated by a sudden recompression. Also, the internal region between the two airfoils is shaped as a nozzle where a supersonic expansion terminated by a shock takes place. Five contributors of Ref. 24 presented results for this case in detail and one author provided only the corresponding lift and drag coefficients.

The computation has been performed using a Cartesian grid with 140×202 cells with 104 cell centers located in the first row surrounding the body on the internal side. The corresponding constant Mach number contours are plotted in Fig. 9, while the computed lift and drag coefficients are reported in Fig. 10 where they are compared to the values computed by the six contributors²⁴. Fig. 10 shows the lift coefficient versus the drag coefficient for the lower and the upper airfoils: the present results are indicated by dark symbols while the other results from Ref. 24 are indicated by light symbols. We see from Fig. 10 that the present results are in general agreement with those of Ref. 24 and that they are quite close to the results by Couallier *et al.* (right triangles), which were considered to be the most accurate²⁴.

2. FAR-FIELD COARSENING

2.1 Iblanking Approach

The iblanking approach¹⁹ is used here to preserve the structured nature of the Cartesian grid. All the cell centers are conserved, even in presence of grid coarsening, but only the active (non-iblaned) cell centers are computed. Consider Fig. 11, where the active cell centers are indicated by solid squares, while the iblanked cell centers are indicated by open squares. Before the grid coarsening, four cell centers located at j_1 , $j_1 + 1$, $j_1 + 2$ and $j_1 + 3$ are active at i_1 as well as at $i_1 + 1$. After the grid coarsening has been applied, four cell centers are still active at i_1 , while only two cell centers are active at $i_1 + 1$, because those located at $j_1 + 1$ and at $j_1 + 3$ have been iblanked, i.e., they are still present in the data structure, but the equations are not integrated at these cell centers. Note that the still active cell centers at $(i_1 + 1, j_1)$ and $(i_1 + 1, j_1 + 2)$ may be considered to have been geometrically moved to the positions indicated by dark circles. Obviously, the equations will be evaluated at these active cell centers and will be performed with the correct geometric data. In these computations, data will be taken from active cell centers, i.e., the computation at $(i_1 + 1, j_1)$ will take data from $(i_1 + 1, j_1 + 2)$, while skipping the data at $(i_1 + 1, j_1 + 1)$, because it is not active.

The criterion for iblanking the cell centers is the following: First, the computational flow field is subdivided into five regions. The first rectangular region is located at the center of the computational flow field and includes the body (refer to Fig. 12); no coarsening is performed in this region in order to preserve an adequate dense mesh close to the body. The other four regions are located upstream and downstream of the body in the x and y directions, respectively as indicated in Fig. 12. Consider the regions upstream and downstream of the body in the x -direction (regions U and D in Fig. 12). As you move away from the body, Δx_i increases due to grid stretching. In these regions, far upstream and downstream of the body, it is clear that Δx_i will be very much greater than Δy_j . A rational criterion for grid coarsening in these regions is that the grid will be coarsened when at a given x_i we find that

$\Delta y_j + \Delta y_{j+1} < \Delta x_i$. Similarly in the regions above and below the body (regions A and B in Fig. 12), the grid stretching has Δy_j increasing as you move away from the body. In these regions a rational criterion is to coarsen the grid when at a given y_j we find that $\Delta x_i + \Delta x_{i+1} < \Delta y_j$. In each of the regions, the coarsening is applied sweeping from the center of the flow field to the corresponding outer boundary.

At this point we illustrate the grid coarsening procedure. At the beginning, all the flow field cell centers are flagged with 1, then we start the grid coarsening in one of the four coarsening regions U, D, A, B of Fig. 12. Let us suppose to start the grid coarsening in the region downstream of the body (D) where the coarsening starts at the lowest i value. Referring to Fig. 11, we suppose that the grid coarsening begins with $i_1 + 1$. In Fig. 11, the cell centers after grid coarsening are indicated by solid circles. The computed flow quantities will be stored in the corresponding solid squares (below). The open squares will be treated by iblanking. The way that this is implemented is as follows: we sweep along the line $i = i_1 + 1$ and check if two nearby cells coalesce, (indicated by j_1 and $j_1 + 1$ in Fig. 11) and determine a new cell characterized by a $\Delta y < 0.9 \Delta x$. If this is true, the two cells coalesce to form a new larger cell at $i_1 + 1$ by merging the j_1 and $j_1 + 1$ cells. The merged cells are extended to the outer boundary for all $i > i_1 + 1$. Meanwhile all the cell centers $j_1 + 1$ are iblanked for $i \geq i_1 + 1$ and flagged with a 0. Obviously the cell centers j_1 for $i \geq i_1 + 1$ remain active and flagged with 1, but with updated geometric properties. Sweeping the whole $i_1 + 1$ line complete the grid coarsening at $i_1 + 1$. Then we move to $i_1 + 2$ and we repeat the previous process by considering only the cell centers flagged with 1, because those flagged with 0 have already been iblanked. A similar process of grid coarsening is applied up to the outer boundary and repeated in the other three regions, so that a far-field coarsened grid is finally obtained.

Next, in order to solve the equations at each active cell center, we need to determine the nearby active cells. This may be simply performed by sweeping all the i or j lines and determining the distances, in terms of i and j , between the active cell center and the nearest active cell centers. In such a way, the values of $\Delta i^+, \Delta i^-, \Delta j^+, \Delta j^-$ can be evaluated. Obviously, in absence of any coarsening, $\Delta i^+ = \Delta j^+ = 1$, $\Delta i^- = \Delta j^- = -1$, while the absolute values of these quantities may be larger in coarsened regions. Then, in order to solve the equations only at each active cell center, the previous sweep is repeated and the i and j coordinates of the active cell centers (flagged with 1) are inserted in two one-dimensional arrays and the computations are performed only at these coordinates. In addition, the cell centers located inside the body are flagged with a 0, so that these cell centers are not computed.

Some particular treatment is required at the *coarsening elements*. In Fig. 11, a coarsening element is the zone including the cells (i_1, j_1) , $(i_1, j_1 + 1)$, $(i_1 + 1, j_1)$. Typically, a coarsening element contains two small cells and one adjacent larger cell, obtained from the coalescence of two smaller cells. The fluxes at the vertically-oriented cell edges cannot be computed in a straightforward way because of the iblanked cell center $(i_1 + 1, j_1 + 1)$ and because of the change of the geometric definition of the active cell center $(i_1 + 1, j_1)$. A typical topological picture of the region surrounding a coarsening element in the coarsening region downstream of the body is reported in Fig. 13. In this figure, the dark circles indicate cell centers, some of which are designated by a capital letter. The **X** symbols and the corresponding numbers indicate cell edges and the + symbols and the corresponding numbers locate points used for extrapolation of the left and right states. Note that the scales of X and Y in Fig. 13 are different. As a consequence the cell dimensions are different from the representation in Fig. 11, *e.g.*, the cell P is almost a square while in Fig. 13 it appears to be a rectangle. The left and right states at the cell edges labelled with the numbers 1, 5, 9, 13, 14 can be evaluated by means of the traditional linear extrapolation relations, provided that the conserved variables at the points 3, 4, F , C , 7, 8, E , A , 10, H , 11, 12, R , S are known. The conserved variables at the cell centers labelled with letters are obviously known, while they must be interpolated at the points labelled with numbers. The point 3 is located between the cell centers H and I , the point 7 between G and H , the point 10 between E and F . A more uncertain situation holds for the remaining points 4, 8, 11 and 12. Indeed, depending on the coarsening, points 4, 8 and 11 may be located between cell centers M and L (present situation) or between cell centers L and N , while point 12 may be located between Q and P or between P and O . A straightforward linear interpolation based on the values at the surrounding cell centers provides the values of the conserved variables at the points 3, 7 and 10.

The uncertainty in the interpolation of these variables at point 4 can be overcome by using the following relation:

$$q_4 = q_L + \left[\frac{q_L - q_M}{y_L - y_M} l_{l_4} + \frac{q_N - q_L}{y_N - y_L} (1 - l_{l_4}) \right] (y_F - y_L), \quad (17)$$

where q indicates the conserved variables and y the ordinates, while the subscript letters correspond to the cell centers in Fig. 13. Furthermore, $ll_4 = 1$ if $y_F \leq y_L$, otherwise $ll_4 = 0$. Using Eq. (17), the conserved variables at point 4 are linearly interpolated using the values pertaining to the cell centers M and L if point 4 is located between these cell centers. Otherwise, the interpolation uses values at cell centers L and N . Similarly, at points 8, 11, 12, the following relations can be used:

$$q_8 = q_L + \left[\frac{q_L - q_M}{y_L - y_M} ll_8 + \frac{q_N - q_L}{y_N - y_L} (1 - ll_8) \right] (y_E - y_L), \quad (18)$$

$$q_{11} = q_L + \left[\frac{q_L - q_M}{y_L - y_M} ll_{11} + \frac{q_N - q_L}{y_N - y_L} (1 - ll_{11}) \right] (y_H - y_L), \quad (19)$$

$$q_{12} = q_P + \left[\frac{q_P - q_Q}{y_P - y_Q} ll_{12} + \frac{q_O - q_P}{y_O - y_P} (1 - ll_{12}) \right] (y_H - y_P). \quad (20)$$

In Eq. (18), $ll_8 = 1$ if $y_E \leq y_L$, otherwise $ll_8 = 0$. In Eq. (19), $ll_{11} = 1$ if $y_H \leq y_L$, otherwise $ll_{11} = 0$. In Eq. (20), $ll_{12} = 1$ if $y_H \leq y_P$, otherwise $ll_{12} = 0$.

Using the values at the appropriate cell centers and the interpolated values at the other points, the left and right states at the vertical cell edges 1, 5, 9, 13, 14 can be computed and the fluxes at these cell edges evaluated.

At this point, we see from Fig. 13 that the topology at the right side of cell H is general, while this is not true for the topology on the left side. Indeed, the cell center E can also belong to a coarsening element with smaller cells located at the left side of E as illustrated in Fig. 14, where even the cell F is in such a situation. This would bring in additional topological possibilities and complicate the evaluation of the fluxes. This in turn is not necessary if the coarsening element evaluation is performed sweeping from the highest i value (corresponding to the outer boundary) to the smallest one (corresponding to the left boundary of the coarsening region downstream of the body). In such a way, the possible errors in the flux evaluation at the cell edges of the cell E would be corrected when computing the coarsening element including the cells A , B , E . Finally, it must be emphasized that the cell H has 5 cell edges with different fluxes. This situation must be taken into account when computing the residuals at this cell. The previous procedure with minor modifications may be utilized for the evaluation of the fluxes at the coarsening element cell edges for the other three coarsening regions.

2.2 Far-Field Coarsening Results

We considered three test cases corresponding to subsonic and transonic flow conditions about a NACA 0012 airfoil. The first one is the transonic flow with an upstream Mach number $M_\infty = 0.80$ at an angle of attack of 1.25° . Figure 15 presents the Mach number distribution on the airfoil surface (continuous lines) computed by applying the grid coarsening approach. In this figure, this distribution is compared with the original grid Mach number distribution (symbols). The two sets of results in Fig. 15 coincide. The comparison of the aerodynamic parameters enforces this finding. The coarsened grid computation gives a lift coefficient $C_L = 0.3529$ and a drag coefficient $C_D = 0.0232$, practically coincident with the original grid computation results: $C_L = 0.3535$, $C_D = 0.0232$. In this test case, the far-field coarsening reduces the original 17272 cells to only 7601 cells. Only 44% of the original cells are computed resulting in a 34% reduction in CPU time. The cell center distribution is represented in Fig. 16 for the original grid, while the effect of the far-field coarsening is evident in the cell center distribution in Fig. 17. The clustering of the grid near the body is also evident in the far-field in Fig. 16, while it is completely eliminated in Fig. 17, where the far-field mesh presents an aspect ratio not far from one. For comparison, this test case has been recomputed using an uncoarsened grid of 8366 cells with 7747 cells, located in the flow field, to be computed, i.e., a cell number slightly higher than the number of cells considered in the previous coarsened grid computation. The computed aerodynamic parameters are $C_L = 0.3610$, $C_D = 0.0241$, which are significantly different from the reference results: $C_L = 0.3535$ and $C_D = 0.0232$.

The second test case is the subsonic flow with an upstream Mach number $M_\infty = 0.63$ at an angle of attack of 2° . Figure 18 shows the Mach number distribution on the airfoil surface computed by applying the grid coarsening approach (continuous lines), compared with the original grid Mach number distribution (symbols). Fig. 18 again shows the agreement of the two distributions. In this test case, the original grid lift and drag coefficients are $C_L = 0.3345$, $C_D = 0(10^{-4})$ while the coarsened grid

computation gives $C_L = 0.3341$ and $C_D = 0(10^{-4})$. Furthermore, only 40% of the original cells are computed and the CPU time is reduced to 47%.

Finally, Fig. 19 shows the Mach number distributions for the last test case, transonic flow with an upstream Mach number $M_\infty = 0.85$ at an angle of attack of 1° , and once again, indicates their agreement. The computed original grid aerodynamic parameters are $C_L = 0.3726$, $C_D = 0.0588$ while the coarsened grid results are $C_L = 0.3721$ and $C_D = 0.0589$. For this test case, only 50% of the original cells are computed and the CPU time is reduced to 38%. All the computed test cases prove that the far-field coarsening practically does not alter the computed results, while significantly reducing the computational time. One can easily imagine even more dramatic results for three-dimensional cases.

3. INVERSE DESIGN OF AIRFOILS

3.1 Objective Function

The inverse design of airfoils consists of finding the body geometric shape whose pressure distribution along the body surface, $p(s, \xi)$, matches a target pressure distribution, $\hat{p}(s)$, where s is the curvilinear coordinate measured along the surface. A discrete objective function, I , may be defined as:

$$I(\xi) = \frac{1}{2 N_b} \sum_{i=1}^{N_b} [p_i - \hat{p}_i]^2, \quad (21)$$

where the body length has been subdivided into N_b intervals, whose center locations are s_i ($i = 1, \dots, N_b$). The discrete interpolated pressures are $p_i = p(s_i, \xi)$ and the discrete target pressures are $\hat{p}_i = \hat{p}(s_i)$, where ξ represents the design parameters.

3.2 Adjoint Formulation

An auxiliary flow solver is used only for the computation of smooth sensitivity derivatives. It should be characterized by added artificial viscosity in order to smooth the shock. The simplest approach which will meet our goals is to use a central scheme with an adequate level of artificial viscosity, as described in Ref. 21. This auxiliary flow solver is used only in the discrete adjoint formulation described below.

The technique presented in Ref. 21 brings us to the following equation for the computation of the derivatives of the objective function:

$$\frac{\partial I}{\partial \xi_j} = \left(\frac{\partial I}{\partial \xi_j} \right)_{\mathbf{u}} - \mathbf{\Lambda}_k^T \left(\frac{\partial \mathbf{w}_k}{\partial \xi_j} \right)_{\mathbf{u}}, \quad (22)$$

where the adjoint variables $\mathbf{\Lambda}_k$ are evaluated according to the following conditions:

$$\frac{\partial \mathbf{\Lambda}_i}{\partial t} + \left(\frac{\partial \mathbf{w}_k}{\partial \mathbf{u}_i} \right)_{\xi}^T \mathbf{\Lambda}_k = \left(\frac{\partial I}{\partial \mathbf{u}_i} \right)_{\xi}^T, \quad (23)$$

for $i = 1, \dots, N \times M$ and $j = 1, \dots, n$, with n the number of design parameters, and repeated indices summed over their range. In the previous equations, \mathbf{u}_i corresponds to the vector of the physical variables at each cell center, i :

$$\mathbf{u}_i = [p, \rho, u, v]_i^T, \quad (24)$$

while $\mathbf{w}_k(\mathbf{u}, \xi) = 0$ represents the discretized system of governing partial differential equations corresponding to the auxiliary flow solver.

Equation (23) represents a set of $4 \times N \times M$ adjoint equations. In addition, the terms $(\partial \mathbf{w}_k / \partial \mathbf{u}_i)_{\xi}$ at the cell centers away from the body can be evaluated directly from the system of discretized flow equations, $\mathbf{w}_k = 0$, and depend on the solution of the flow field. At the cell centers close to the body, they are numerically evaluated following the procedure suggested in Ref. 25, which determines these values

by perturbing u_i at a cell center close to the body and by determining the corresponding variations of the residuals at the considered cell center and the neighboring cell centers. Then $(\partial \mathbf{w}_k / \partial \mathbf{u}_i)_\xi$ is approximated by the ratio between the variations of the residuals and the corresponding variations of u_i . In this computation, the corresponding changes of the boundary conditions are also taken into account. The terms $(\partial \mathbf{w}_k / \partial \xi_j)_\mathbf{u}$ are numerically evaluated in a similar way by perturbing the design parameters ξ_j , by evaluating the corresponding variations of the fluid dynamic properties at the row of cell centers surrounding the body on the interior and by determining the corresponding changes of the residuals at the cell centers surrounding the body and located in the flow field. The terms $(\partial I / \partial \mathbf{u}_i)_\xi$ can be easily evaluated from the objective function expression, while the terms $(\partial I / \partial \xi_j)_\mathbf{u}$ are evaluated by perturbing the design parameters and by evaluating the corresponding changes of the objective function.

The fluid dynamic variables computed with the accurate, basic flow solver are used to compute the terms $(\partial I / \partial \mathbf{u}_i)_\xi$ and $(\partial \mathbf{w}_k / \partial \mathbf{u}_i)_\xi$. Then, the adjoint equations are solved for the $4 \times N \times M$ adjoint variables, $\Lambda_\mathbf{k}$. During this computation, the far field boundaries are moved closer to the body with respect to their location for the flow field computation. This approximation allows to significantly reduce the CPU time required to evaluate the adjoint variables without significantly change their values at the cell centers located in the vicinity of the body. During the adjoint equation solution, the adjoint variables corresponding to the cell centers located inside the body are set to zero and never changed during the computation. Finally, Eq. (22) is used to evaluate the sensitivity derivatives of the objective function, $\partial I / \partial \xi_j$.

At this point we faced the choice of using the coarsened grid or resorting to an optimization process on the complete grid. We decided to adopt the second choice and to use the complete uncoarsened grid because of the following reasons: i) the domain employed to solve the adjoint equations is significantly reduced with respect to the one employed to solve the flow equations, so that the savings in cell center evaluation would have been very small; ii) the coarsening requires a special treatment at the de-refinement elements. The computational cost of this treatment may not be small, in comparison to the CPU saving due to the reduced number of cell centers to be computed, if few iblanked cell centers are involved. We concluded that the possible saving would have been very small, if any. Thus, before transferring the fluid dynamic data to the optimizer, we had to evaluate these data at all cell centers corresponding to the original uncoarsened grid. This problem was simply solved by interpolating the solution from the coarsened grid to the original one by a linear interpolation based on the values computed at the active cell centers.

3.3 Progressive Optimization

The goal of the progressive convergence technique is to determine the minimum of the objective function, while progressively reaching a converged solution for the flow field and the adjoint equations. The current implementation of the progressive algorithm is as follows:

1. Define the coarsest mesh and several finer levels. Each finer mesh is obtained from the corresponding coarser mesh by doubling the number of intervals in each direction.
2. Start with an initial set of design variables.
3. Start the flow computations on the coarsest grid.
4. Advance the flow solver for several iterations.
5. Evaluate the adjoint equations by advancing the adjoint solver for several iterations, with the condition that the residuals drop at least one order of magnitude with respect to the value it assumes at the beginning of the procedure.
6. Compute the objective function gradient, ∇I .
7. Update the design variables according to the relation:

$$\xi_j^{l+1} = \xi_j^l - a_j \frac{\partial I}{\partial \xi_j}, \quad (25)$$

where a_j are positive parameters.

8. Repeat steps 4 to 7 until the gradient of the objective function is sufficiently decreased (from half to one order of magnitude), with the condition that the residuals of the flow and of the adjoint equations drop progressively.
9. Move to the corresponding finer mesh and interpolate the computed flow solution to the finer grid.
10. Repeat steps 4 to 7 until the gradient of the objective function is decreased half to one order of magnitude more.
11. Repeat steps 9 and 10 until the finest grid is reached.
12. Repeat steps 4 to 7 until the objective function has sufficiently decreased.

The coefficient a_j in Eq. (25) is evaluated as $a_j = b c_j$. The coefficient b represents the minimum value of a_j and is given by:

$$b = \frac{|\Delta\xi_c|}{k |\nabla_j I|_{max}^0}, \quad (26)$$

where $\Delta\xi_c$ is a typical (order of magnitude) change of the design parameters, k is a constant ranging from 40 to 100 and depending on the test case, and $|\nabla_j I|_{max}^0$ is the largest absolute value assumed by the sensitivity derivatives of the objective function during the first global iterations through steps 2 to 6. The term c_j is an amplification factor with a minimum value equal to 1 and a maximum value c_M which depends on the convergence of the normalized gradient of the objective function. In particular, c_M is equal to 10 if the convergence is lower than one order of magnitude and is equal to 40 if it is higher than two orders, while it varies linearly between 10 and 40 for intermediate values of the convergence of the normalized gradient of the objective function. At the beginning of the computations, c_j is set to 1, then it is increased by 50% if the corresponding sensitivity derivative of the objective function maintains its sign, while it is decreased by 50% if the sign changes. This approach allows large changes for the design variables whose sensitivity derivatives maintain the same sign, while it assigns small changes to design variables whose sensitivity derivatives are changing their sign. We have devised this technique because of the very slow convergence of the method of steepest descent, corresponding to c_j equal to one. Our approach of using a smoothed adjoint solver based on an auxiliary flow solver leads to an approximate gradient evaluation, which caused convergence problems with classical optimizers. The use of the amplification factor c_j described above, significantly improves convergence, compared to the method of steepest descent.

In the computed results, we have used three grid levels and we have considered the objective function sufficiently decreased when the norm of its gradient has decreased 3 orders of magnitude with respect to the value it assumes after the first global iteration through steps 2 to 6.

3.4 Inverse-Design Results

The inverse design of airfoils in subsonic and transonic flow conditions has been considered. Following the procedure used in Ref. 26, the airfoil surface shape is described as:

$$Y(x/c) = \sum_{i=1}^{N_a} \xi_i Y_i(x/c), \quad (27)$$

where $Y = y/c$, with y being the airfoil ordinate and c the airfoil chord length. The shape functions $Y_i(x/c)$ are the ordinates of the N_a specified airfoils, which depend on the non-dimensional abscissa x/c . Finally, the weighting terms ξ_i are the design parameters.

First, we considered the subsonic flow about an NACA 0012 airfoil with an undisturbed Mach number $M_\infty = 0.63$ at an angle of attack of 2° , with the aim to check the ability of the method to bring the convergence of the optimization process to machine zero. The reference flow solution on the finest mesh converged to machine zero was used to define the pressure distribution on the airfoil surface. The airfoil optimization test was performed with only one design variable. The target airfoil is the NACA 0012 airfoil so that the target value of the design parameter is $\xi = 1$. We assumed an initial airfoil shape

corresponding to a value of the design parameter $\xi = 2.5$. The optimization process was performed using three mesh levels: 32×29 , 64×59 , 128×119 cells, with the larger number of cells being located in the x direction. A line of cell centers was located at $y = 0$ at all mesh levels. Fig. 20 shows the convergence history of the logarithm of the objective function, $\log_{10}(I)$, and of the logarithm of the magnitude of the normalized gradient of the objective function, $\log_{10} |\nabla I|$. The abscissa in Fig. 20 is the required work, with a unit of work considered as the computational time required to run a single analysis solution to convergence on the finest mesh. Fig. 20 shows that $\log_{10}(I)$ converged to 10^{-22} and $\log_{10} |\nabla I|$ converged to 10^{-14} . Meanwhile, the design parameter reached its target value, equal to 1, with a residual error of 10^{-11} . These results indicate that the optimization process reached the machine zero.

Then, we considered the transonic flow about an NACA 0012 airfoil with an undisturbed Mach number $M_\infty = 0.80$ at an angle of attack of 1.25° . The airfoil optimization test was performed with $N_a = 4$ and Y_1 – Y_4 as the shape of four specified airfoils: NACA 0012, NACA 64₁-412, NACA 64₂A215, NACA 65₂-415. The target airfoil is the NACA 0012 airfoil so that the target set of coefficients is $\xi = (1, 0, 0, 0)$. We assumed an initial airfoil shape corresponding to the set of coefficients $\xi = (1.5, 0.5, 0.5, 0.5)$. The optimization process was performed using the three mesh levels employed in the previous test case. Fig. 21 shows the convergence history of the logarithm of the objective function, $\log_{10}(I)$, and of the logarithm of the magnitude of the normalized gradient of the objective function, $\log_{10} |\nabla I|$. The results indicate that the work required to converge the gradient of the objective function by 3 orders of magnitude is equal to 1.07. In our progressive optimization strategy, large gains in efficiency are obtained because most of the geometrical changes occur at the coarsest mesh level at a very low computational cost. Fig. 22 shows the target pressure distribution on the airfoil surface (continuous line) and the converged pressure distribution (symbols). The two sets of results are practically coincident even in the very sensitive shock transition region, proving that the optimized solution is sufficiently converged. Fig. 23 presents the target airfoil shape (continuous line), the initial airfoil shape (dashed line), and the optimized airfoil shape (symbols). The figure shows that the optimized airfoil shape practically coincides with its target value although the optimization process starts from an airfoil shape very different from the optimized one.

Finally, we repeated the previous optimization test case by changing only the flow conditions. Here we considered the sometime difficult transonic flow about an NACA 0012 with an undisturbed Mach number $M_\infty = 0.85$ at an angle of attack of $1.^\circ$. Fig. 24 shows the convergence history of the logarithm of the objective function, $\log_{10}(I)$, and of the logarithm of the magnitude of the normalized gradient of the objective function, $\log_{10} |\nabla I|$. The results indicate that the work required to converge the gradient of the objective function by 3 orders of magnitude is equal to 2.33. Fig. 25 shows the target pressure distribution on the airfoil surface (continuous line) and the converged pressure distribution (symbols). The two sets of results are practically coincident even in the very sensitive shock transition region, proving that the optimized solution is sufficiently converged.

CONCLUSIONS

The first part of this report reviews a new approach to the computation of inviscid flows using Cartesian grid methods. The crux of the method is the curvature-corrected symmetry technique (CCST) developed by the present authors for body-fitted grids. The method introduces ghost cells near the boundaries whose values are developed from an assumed flow-field model in vicinity of the wall, consisting of a vortex flow with a symmetric distribution of entropy and total enthalpy along a local normal to the body surface. This model satisfies the normal momentum equation, the non-penetration boundary condition and Crocco's relation. The CCST boundary condition was shown to be substantially more accurate than traditional boundary condition approaches. This improved boundary condition has been adapted to a Cartesian mesh formulation, the *Ghost Body-Cell Method*. In this method, all cell centers exterior to the body are computed with fluxes at the four surrounding cell edges. There is no need for special treatment corresponding to *cut cells* which complicate other Cartesian mesh methods. The approach does require surface normals and radii of curvature. However, since the body is defined pointwise and is independent of the mesh, it may be easily defined by a sufficient point density to allow straightforward accurate numerical calculation of the surface normals and radii of curvature.

The investigation of the compressible inviscid flow about a circular cylinder has shown that the non-penetration condition is satisfied in the limit of vanishing cell size, with errors in the velocity component normal to the body of the order of 10^{-4} for the finest grid employed (with free-stream velocities scaled

to 1). The Cartesian grid results have then been compared with the results computed using the classical second-order pressure extrapolation technique and the curvature-corrected symmetry technique on body fitted polar grids. The computed contours of total pressure have been compared. This result proves that GBCM on Cartesian grids presents dramatic advantages with respect to the widely used second-order pressure extrapolation technique on body-fitted grids. A mesh refinement study has indicated that GBCM is second-order accurate in space.

Results for the transonic flow about an NACA 0012 airfoil have shown that GBCM can be easily extended to practical body shapes. The computed Mach number distributions compare quite well with body-fitted grid results obtained using the CCST technique. The computed lift and drag coefficients match published values considered to be accurate. Importantly, the results have been conclusively shown to be independent of the position of the leading edge with respect to the grid. The flexibility of the method has been finally illustrated with an application to the flow over a multiple-body configuration, the bi-NACA configuration. Comparison with published results has further confirmed the accuracy of the present results.

The original version of the GBCM, as with all Cartesian mesh formulations, has the property that any grid clustering near the body is maintained to the far field boundary. This is not a significant penalty in 2D calculations, but it may become significant in 3D calculations. In the second part of this report, we have introduced a far-field grid coarsening to address this issue. The technique we have adopted is based on iblanking of cell centers. The computational domain has been subdivided in five regions, the central one including the body and characterized by no coarsening and four external regions, located above, below, upstream and downstream of the body where the grid coarsening is applied. Only the coalescence of two smaller cells into a larger one is allowed, so that the coarsening is essentially anisotropic. The coarsening is applied only in the direction which is characterized by a smaller grid spacing and tends to determine near-square cells. The special treatment required at the connection between two smaller cells and a larger one determined by the coalescence of two smaller original cells is described in detail.

The far-field grid coarsening has been applied to subsonic and transonic flows about an NACA 0012 airfoil, indicating that the results are practically unaffected by the coarsening. Meanwhile, the number of the computed cells is reduced to 40% – 50% and the CPU time to 34% – 47% of the original CPU time, depending on the particular test case.

The Ghost Body-Cell Method on Cartesian far-field coarsened grids appears to be particularly well-suited for design optimization problems. The third part of the report addresses this issue and presents an efficient formulation for the robust inverse design of compressible fluid flow problems. The design approach has five essential features: First, the procedure utilizes the accurate GBCM flow solver on coarsened Cartesian grids. Second, approximate but efficient design sensitivities are obtained using a discrete adjoint formulation. The adjoint problem is based on an artificially dissipative flow solver in order to obtain robust sensitivity derivatives in the presence of noisy or non-smooth objective functions. This adjoint solver is only used for the solution of robust sensitivities. The next feature of the procedure is the use of quasi-time-dependent adjoint equations. The fourth feature involves what we term *progressive optimization*, whereby a sequence of operations, containing a partially converged flow solution, followed by a partially converged adjoint solution followed by an optimization step is performed. The last feature is the use of progressively finer grids for the progressive solution of the flow field, while only using a reduced domain for the solution of the adjoint equations.

The inverse design of airfoils in transonic flow conditions indicated that the methodology is accurate, robust and highly efficient, with converged design optimizations produced in no more than the amount of computational work to perform two flow analysis on the finest mesh. In addition, the inverse design of a NACA 0012 airfoil in subsonic flow conditions showed that the procedure is able to bring the convergence of the optimization process to machine zero.

Overall, we observe that the work to perform inverse design of airfoils on Cartesian grids using progressive optimization procedures is comparable to the work required by the design optimization procedure on body fitted grids, previously developed. Furthermore, the Cartesian grid design procedure only requires the generation of a simple Cartesian mesh which remains unchanged during the optimization iterations, while the body-fitted grid design procedure requires an ad hoc grid generation process and a continuous remeshing during the optimization process. This difference represents a significant advantage of the design procedure on Cartesian grids and makes this procedure much better suited for complex geometry optimization.

References

- [1] C. W. Boppe, “Transonic Flow Field Analysis for Wing-Fuselage Configurations,” NASA CR-3243, May 1980.
- [2] B. Wedan and J. C. South, Jr., “A Method for Solving the Full-Potential Equation for General Configurations,” *AIAA Paper 83-1889-CP*, 515–526 (1983).
- [3] R. L. Gaffney, H. A. Hassan and M. D. Salas, “Euler Calculations for Wings Using Cartesian Grids,” *AIAA Paper 87-0356* (1987).
- [4] S. K. Choi and B. Grossman, “A Flux-Vector Split, Finite-Volume Method for Euler’s Equations on Non-Mapped Grids,” *AIAA Paper 88-0227* (1988).
- [5] G. Moretti and A. Dadone, “Airfoil Calculations in Cartesian Grids,” *Notes on Numerical Fluid Mechanics*, **24**, 423–434 (1989).
- [6] D. De Zeeuw and K. G. Powell, “An Adaptively Refined Cartesian Mesh Solver for the Euler Equations,” *J. Comp. Phys.*, **104**, 56–68 (1993).
- [7] R. B. Pember, J. B. Bell, P. Collela, W. Y. Crutchfield and M. L. Welcome, “An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions,” *J. Comp. Phys.*, **120**, 278–304 (1995).
- [8] J. E. Melton, M. J. Berger, M. J. Aftosmis and M. D. Wong, “3D Applications of a Cartesian Grid Euler Method,” *AIAA Paper 95-0853* (1995).
- [9] M. J. Aftosmis and M. J. Berger, “Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries,” *AIAA Paper 2002-0863* (2002).
- [10] Y. Wenren, J. Steinhoff, L. Wang, L. and M. Fan, “Application of Vorticity Confinement to the Prediction of the Flow over Complex Bodies,” *AIAA Paper 2000-2621* (2000).
- [11] M. C. Lai and C. S. Peskin, “An Immersed Boundary Method with Formal Second Order Accuracy and Reduced Numerical Viscosity,” *Journal of Computational Physics*, **160**, 705–721 (2000).
- [12] R. Verzicco, J. Modh-Yusof, P. Orlandi and D. Haworth, “Large Eddy Simulation in Complex Geometric Configurations Using Boundary Body Forces,” *AIAA Journal*, **38**, 427–433 (2000).
- [13] M.J. Aftosmis, “Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries,” *VKI Lecture Series 1997-02* (1997).
- [14] A. Dadone and B. Grossman, “Surface Boundary Conditions for the Numerical Solution of the Euler Equations,” *AIAA Journal*, **32**, 285–293 (1994).
- [15] A. Dadone and B. Grossman, “Surface Boundary Conditions for the Numerical Solution of the Euler Equations in Three Dimensions,” *Lecture Notes in Physics*, **453**, 188–194 (1995).
- [16] A. G. Godfrey, R. W. Walters and B. Van Leer, “Preconditioning for the Navier-Stokes Equations with Finite-Rate Chemistry,” *AIAA Paper 93-0535* (1993).
- [17] A. Dadone and B. Grossman, “An Immersed Body Methodology for Inviscid Flows on Cartesian Grids”, *AIAA Paper 2002-1059* (2002).
- [18] A. Dadone, “Cartesian Grid Computation of Inviscid Flows about Multiple Bodies”, *AIAA Paper 2003-1121* (2003).
- [19] P. A. Durbin, G. Iaccarino, “An Approach to Local Refinement of Structured Grids”, *Journal of Computational Physics*, **181**, 639–653 (2002).
- [20] J. A. Benek, P. G. Buning, J. L. Steger, “A 3D Chimera Grid Embedding Technique”, *AIAA Paper 85-1523* (1985).

- [21] A. Dadone, B. Grossman, “Fast Convergence of Inviscid Fluid Dynamic Design Problems” , *Computer and Fluids*, **32**, 607–627, 2003.
- [22] P. L. Roe, “Characteristic Based Schemes for the Euler Equations,” *Ann. Rev. Fluid Mech.*, **18**, 337–365 (1986).
- [23] Agard Fluid Dynamics Panel WG07, “Test Cases for Inviscid Flow Field Methods,” AGARD-AR-211 (1985).
- [24] A. Dervieux, B. Van Leer, J. Periaux and A. Rizzi (Eds.), “Numerical Simulation of Compressible Euler Flows,” *Notes on Numerical Fluid Mechanics*, **26**, Vieweg, Braunschweig (1989).
- [25] L. A. Catalano, A. Dadone, “Progressive Optimization for the Efficient Design of 3D Cascades”, *AIAA Paper 2001-2578* (2001).
- [26] C. Y. Joh, B. Grossman, R. T. Haftka, “Design optimization of transonic airfoils”, *Engineering Optimization*, **21**, 1640–1647 (1993).

	−50%	−25%	0%	+25%	+50%
p	$2.78 \cdot 10^{-4}$	$2.48 \cdot 10^{-4}$		$4.12 \cdot 10^{-4}$	$5.95 \cdot 10^{-4}$
ρ	$2.41 \cdot 10^{-4}$	$2.02 \cdot 10^{-4}$		$3.14 \cdot 10^{-4}$	$4.94 \cdot 10^{-4}$
Mach	$4.95 \cdot 10^{-4}$	$3.74 \cdot 10^{-4}$		$5.07 \cdot 10^{-4}$	$9.84 \cdot 10^{-4}$
u	$4.63 \cdot 10^{-4}$	$3.96 \cdot 10^{-4}$		$4.44 \cdot 10^{-4}$	$8.90 \cdot 10^{-4}$
v	$3.82 \cdot 10^{-4}$	$2.38 \cdot 10^{-4}$		$3.63 \cdot 10^{-4}$	$5.08 \cdot 10^{-4}$
$\langle T^\circ - 1 \rangle$	$2.09 \cdot 10^{-3}$	$2.07 \cdot 10^{-3}$	$2.09 \cdot 10^{-3}$	$2.04 \cdot 10^{-3}$	$2.13 \cdot 10^{-3}$
$\langle \tilde{v} \rangle$	$3.28 \cdot 10^{-4}$	$3.68 \cdot 10^{-4}$	$4.46 \cdot 10^{-4}$	$3.98 \cdot 10^{-4}$	$3.18 \cdot 10^{-4}$
C_L	$O(10^{-8})$	$O(10^{-11})$	$O(10^{-10})$	$O(10^{-7})$	$O(10^{-8})$
C_D	0.04612	0.04615	0.04616	0.04609	0.04607

Table 1: Effects of the leading edge position.

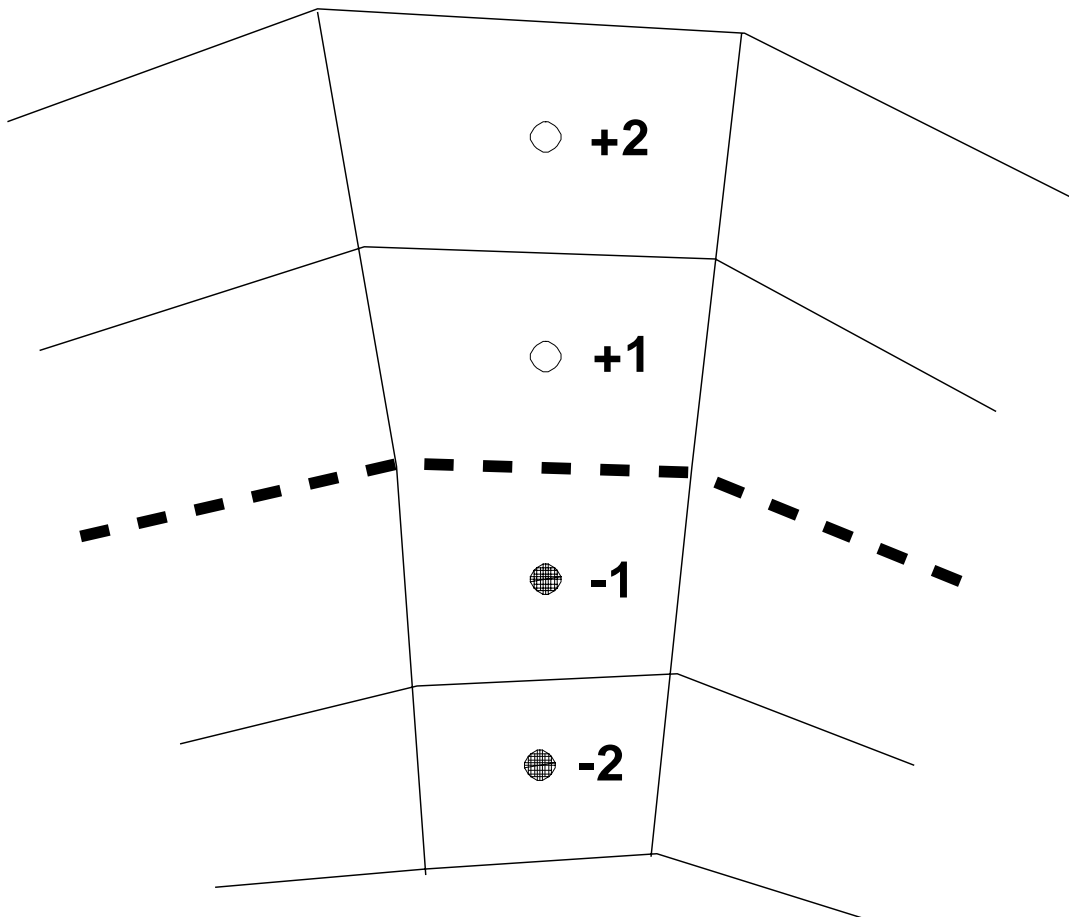


Figure 1: Image cells for body-fitted grid implementation of CCST boundary condition.

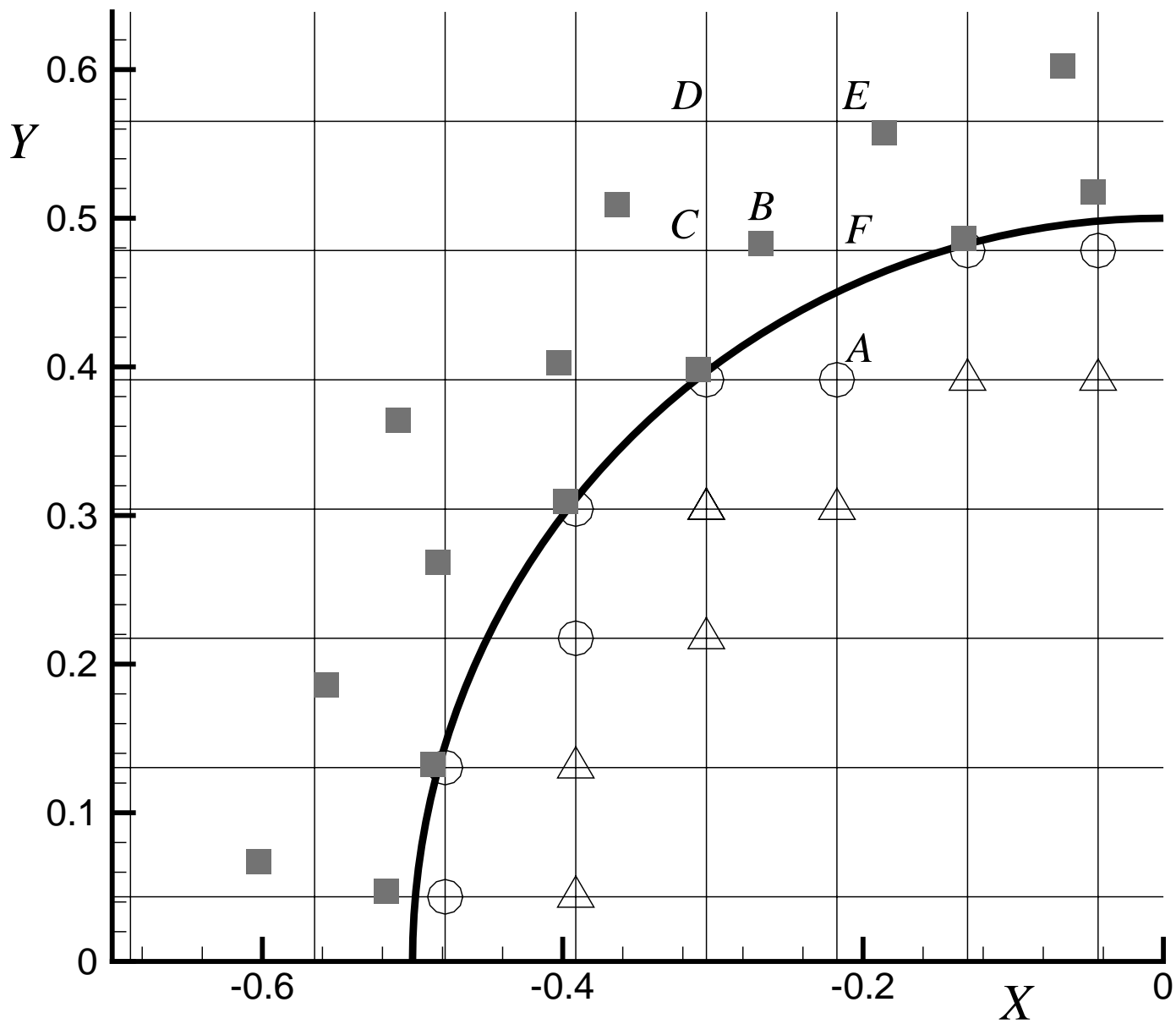


Figure 2: Cartesian grid: cell center net, interior cell centers close to the body (open symbols) and symmetrically reflected exterior cell centers (solid symbols).

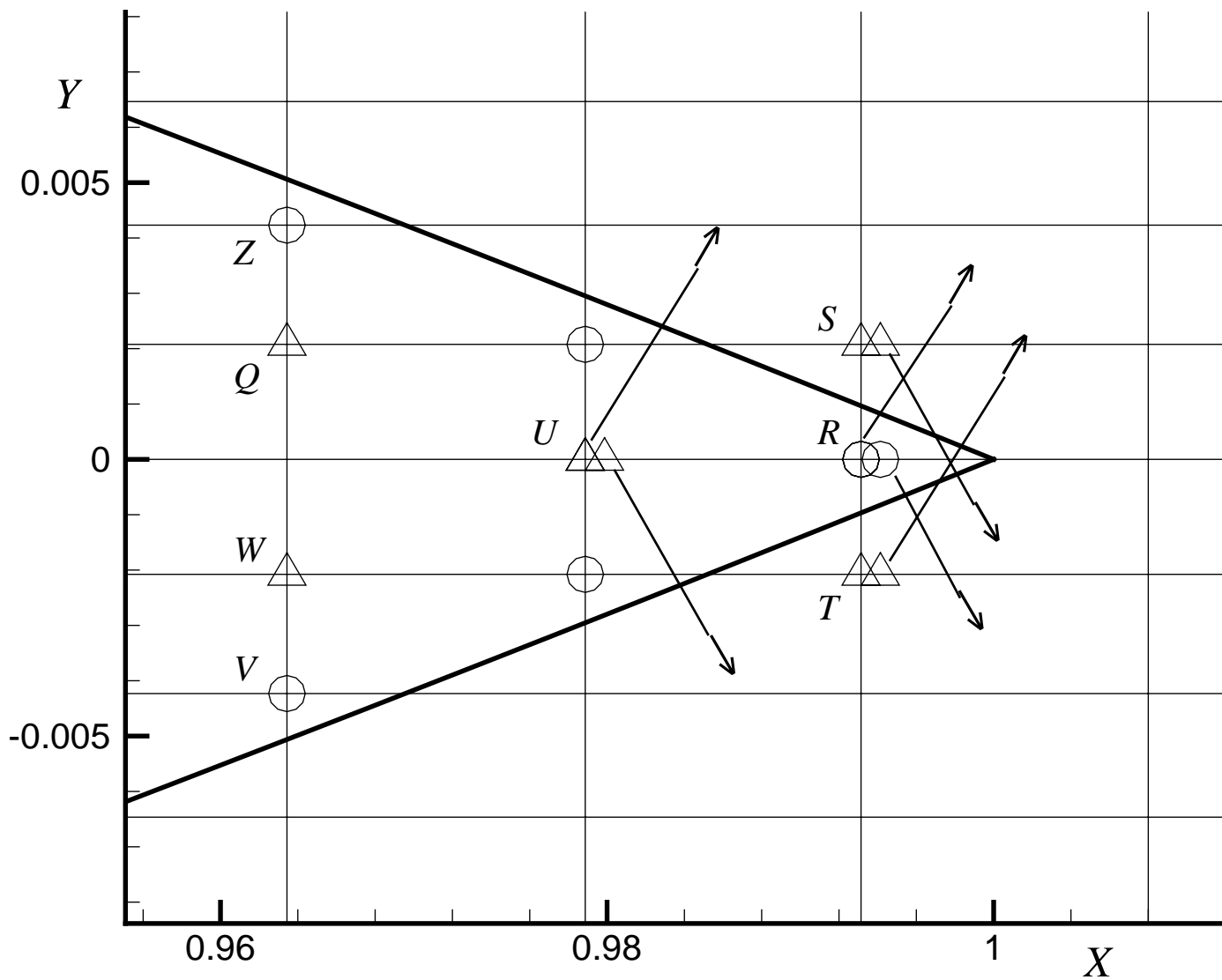


Figure 3: Double-valued ghost cell centers near a sharp trailing edge.

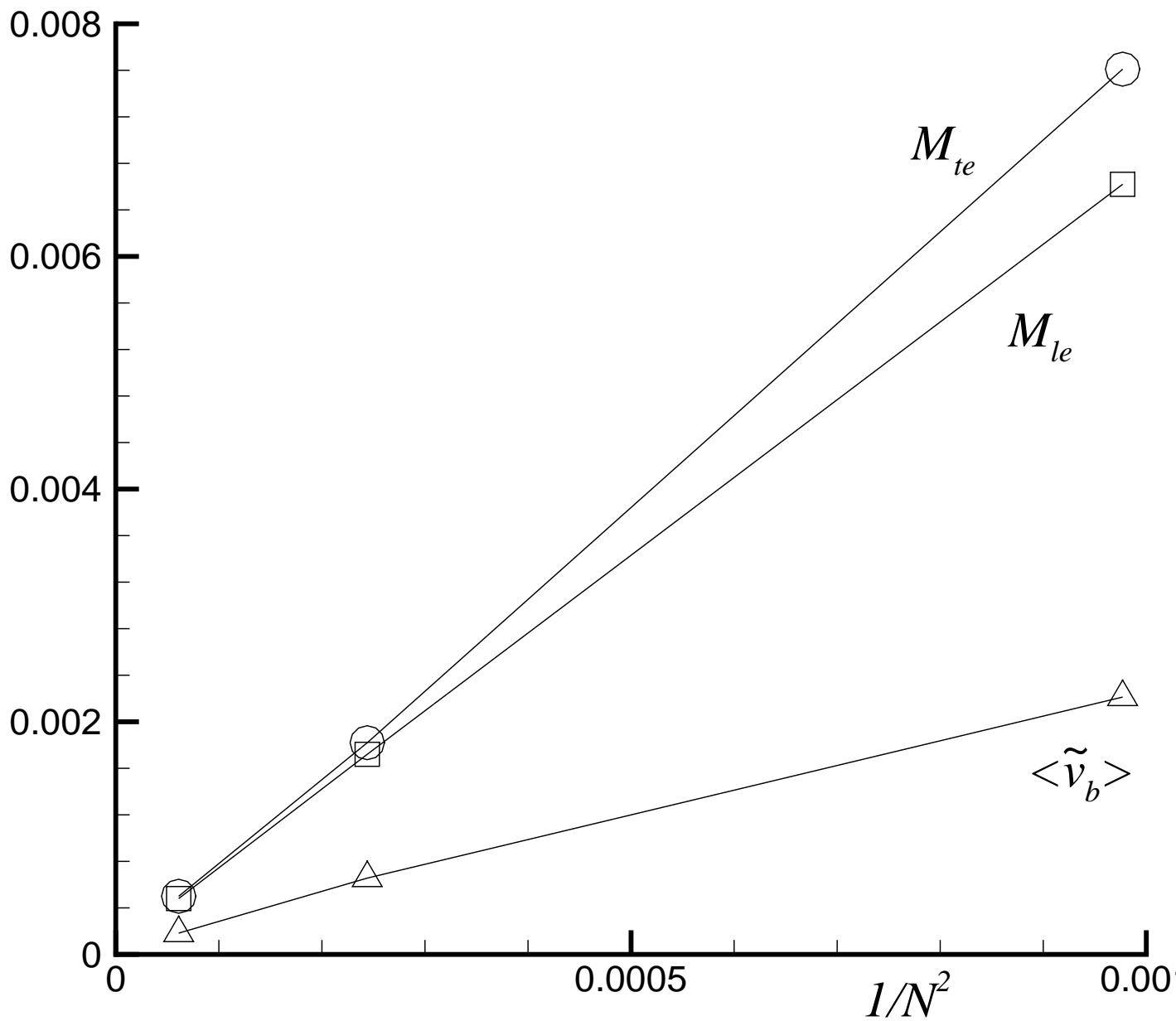


Figure 4: Grid convergence of flow over circular cylinder using Ghost Body-Cell Method. L_2 -norm of the surface normal velocity component and Mach number at the leading and trailing edges.

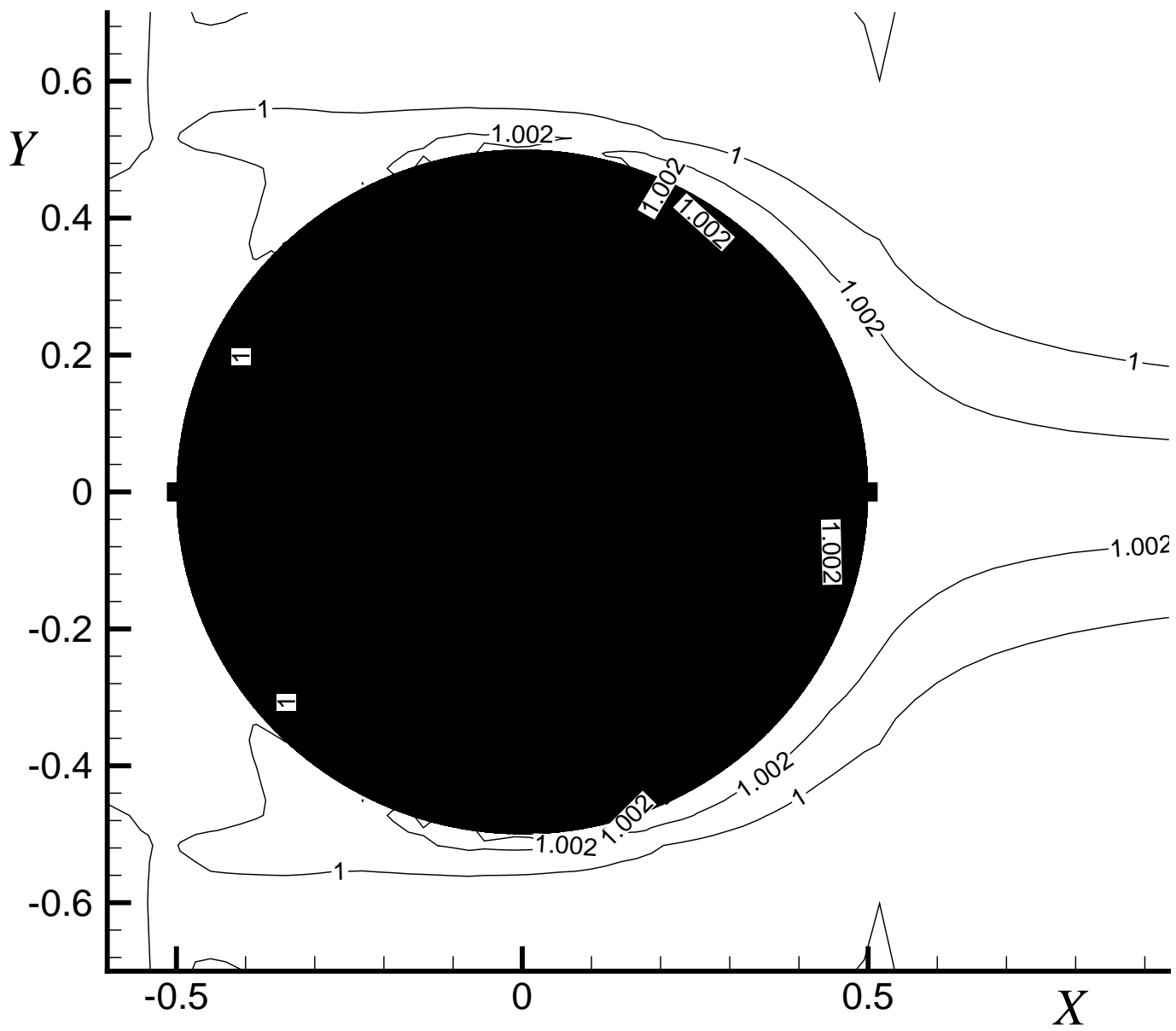


Figure 5: Isobars of total pressure. Ghost Body-Cell Method on Cartesian grid.

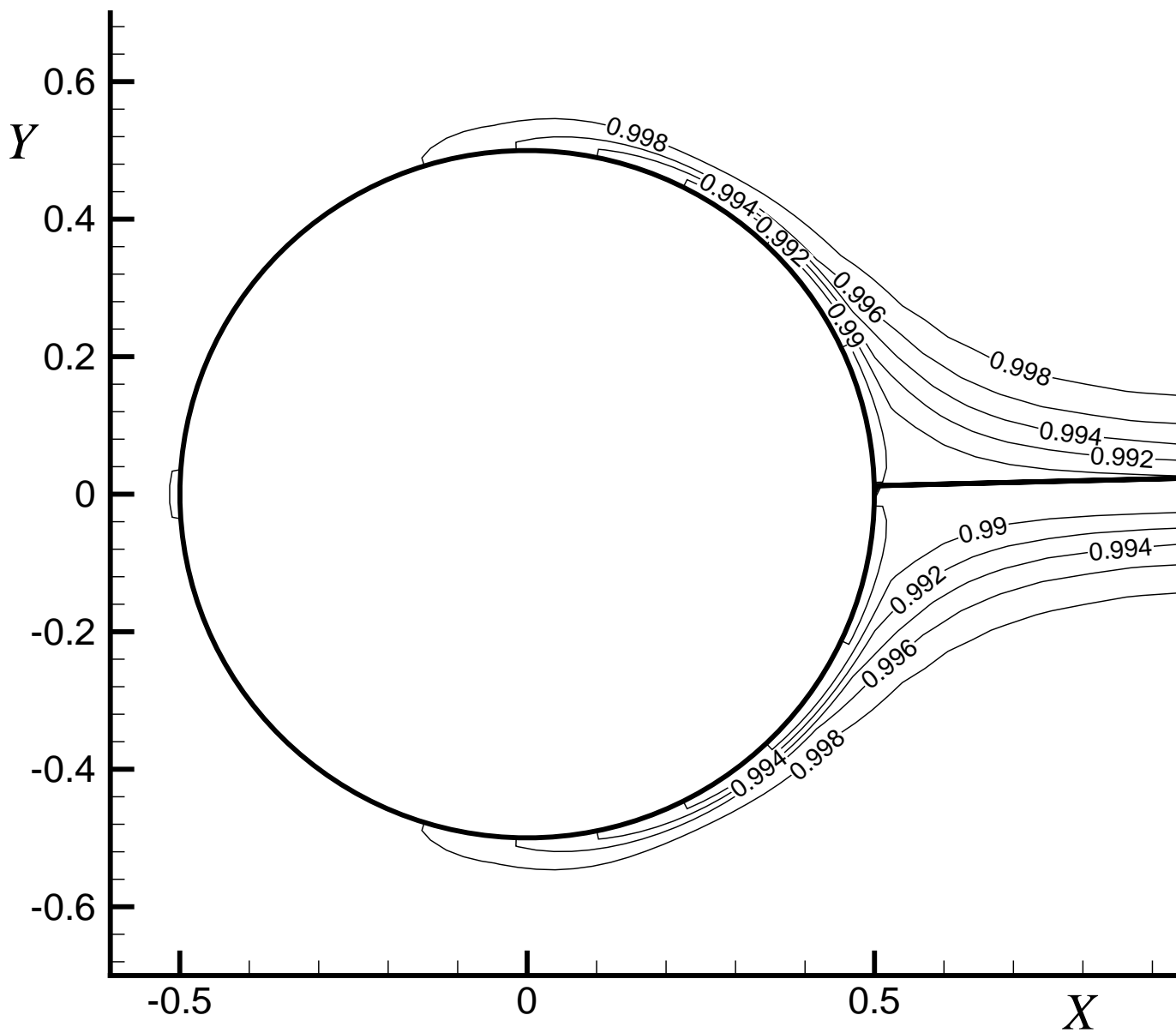


Figure 6: Isobars of total pressure. Second-order surface pressure extrapolation (P-II) on body-fitted polar grid.

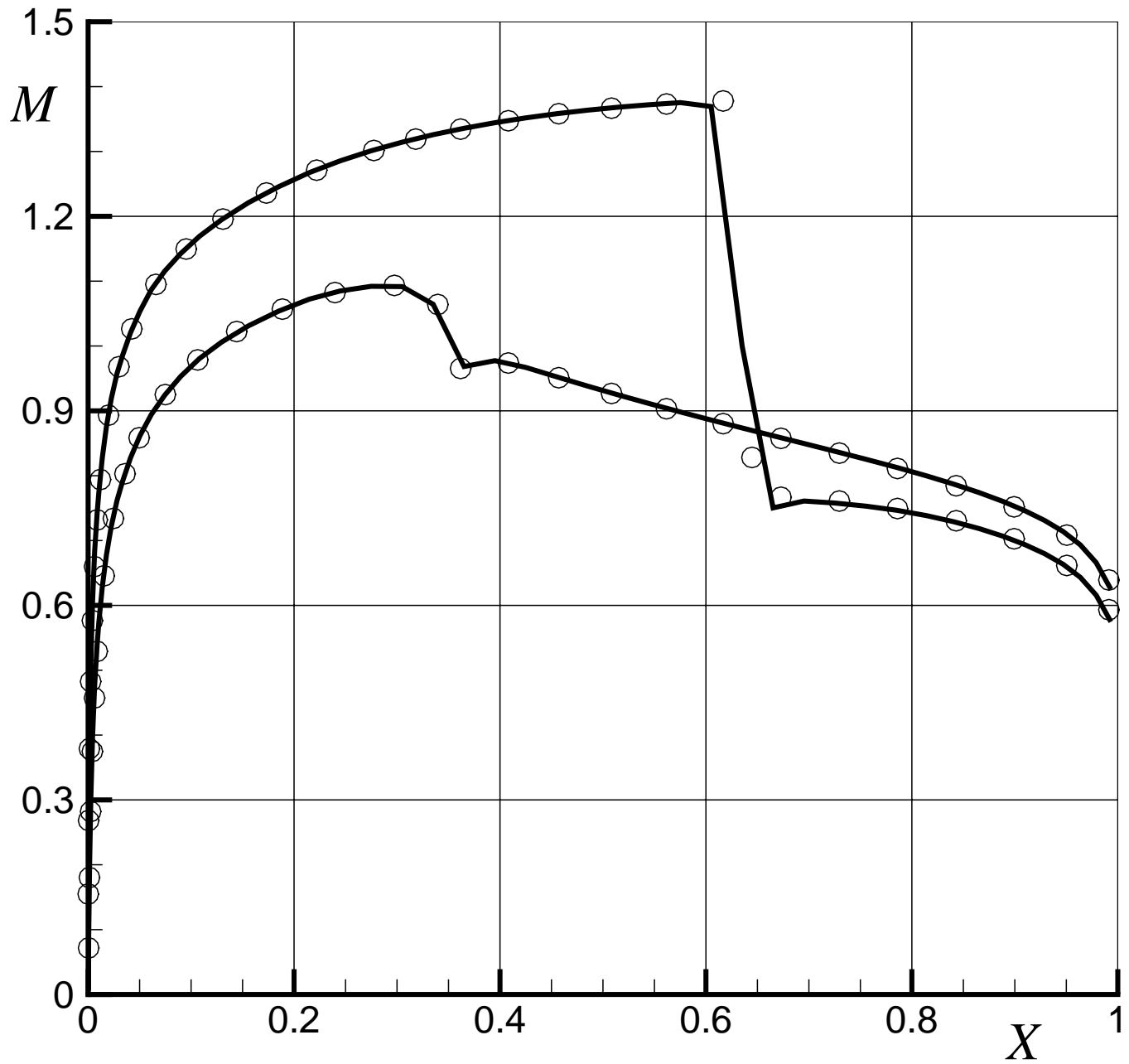


Figure 7: Surface Mach number distribution for NACA 0012 airfoil at $M_\infty = 0.80$, $\alpha = 1.25^\circ$. Cartesian grid GBCM (solid line) and body-fitted CCST (symbols).

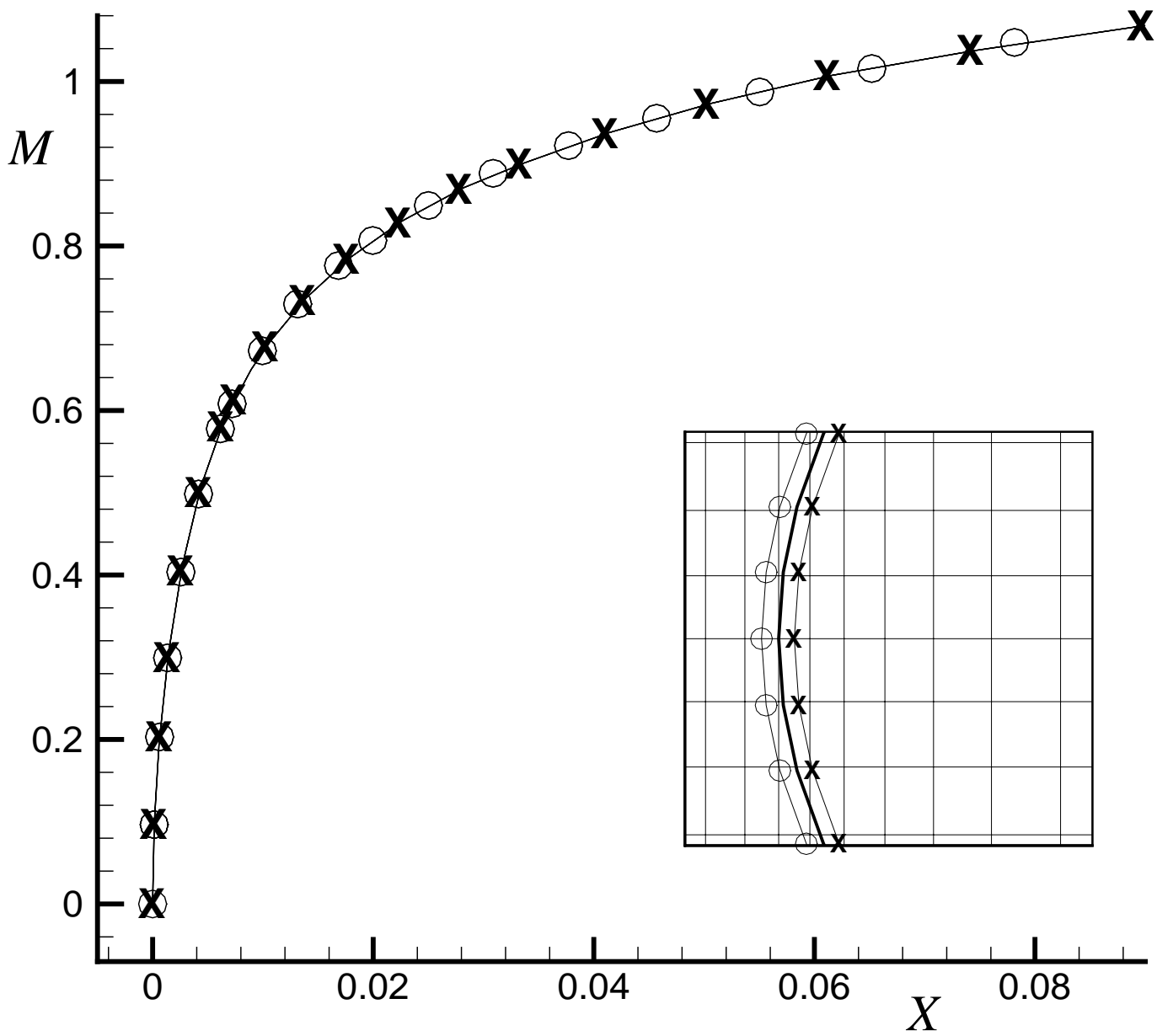


Figure 8: Cartesian grid GBCM calculation of NACA 0012 airfoil at $M_\infty = 0.85$, $\alpha = 0^\circ$. Surface Mach number distribution in the leading edge region. Calculations performed with various leading-edge locations with respect to the grid as indicated by the inset.

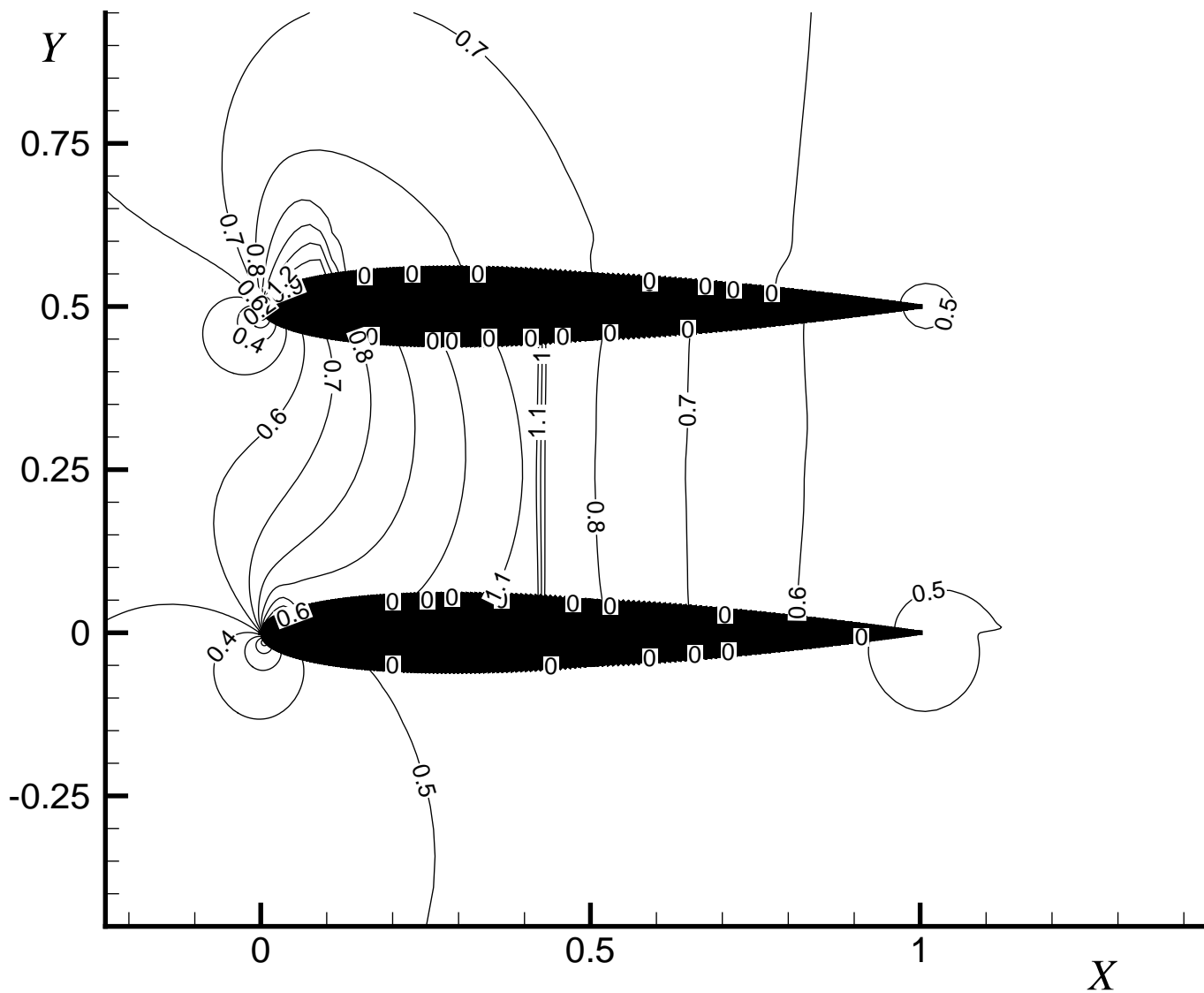


Figure 9: Cartesian grid GBCM calculation of bi-NACA 0012 double airfoil. Constant Mach number contours for $M_\infty = 0.55$, $\alpha = 6^\circ$.

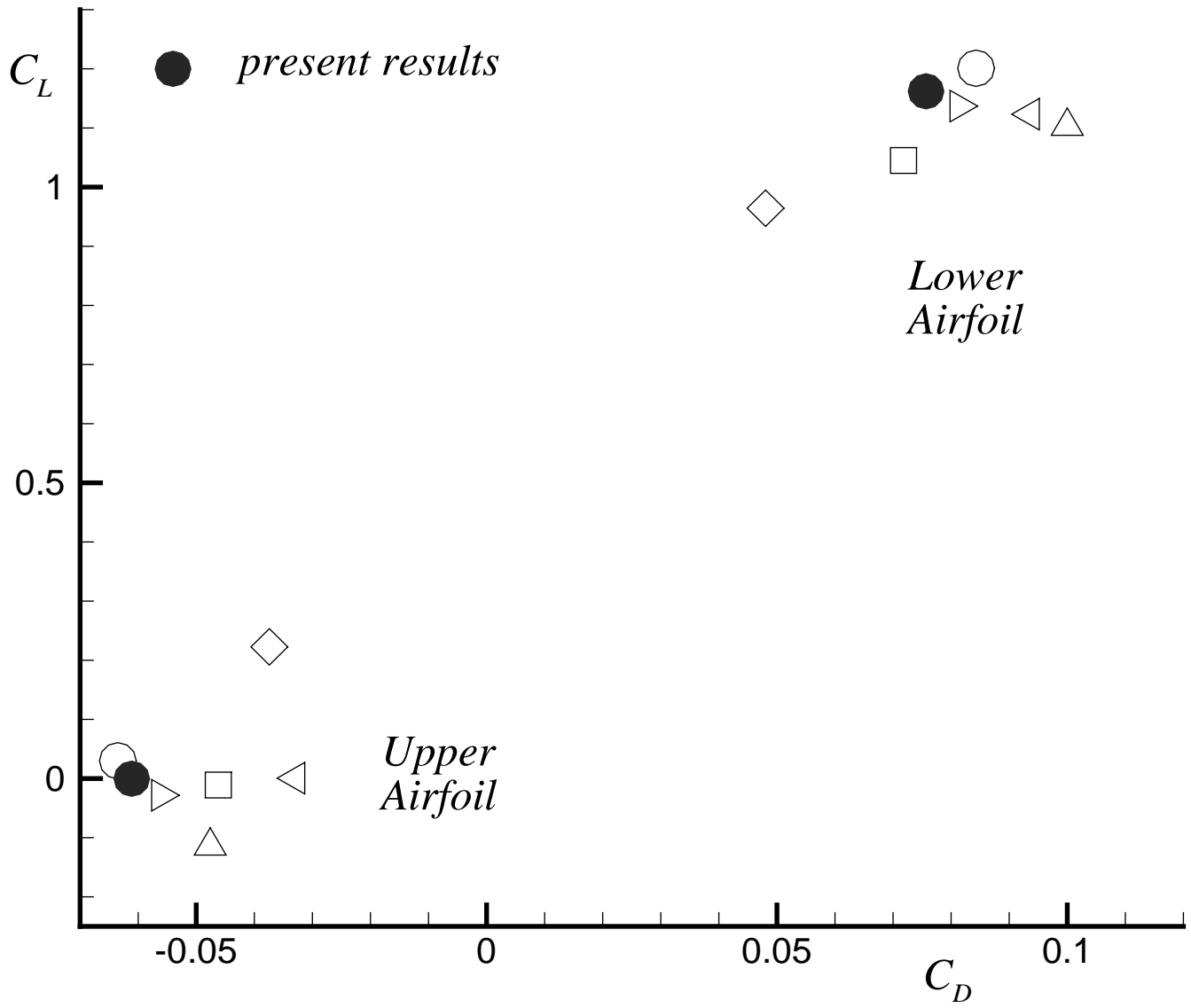


Figure 10: Bi-NACA 0012 double airfoil. Lift coefficient versus drag coefficient at $M_\infty = 0.55$, $\alpha = 6^\circ$. Five results from Ref. 24 (open symbols) and GBCM results (solid circle).

	i_l	i_l+1
j_l+3	■	□ ● ■
j_l+2	■	
j_l+1	■	□ ● ■
j_l	■	

Figure 11: Coarsening elements.

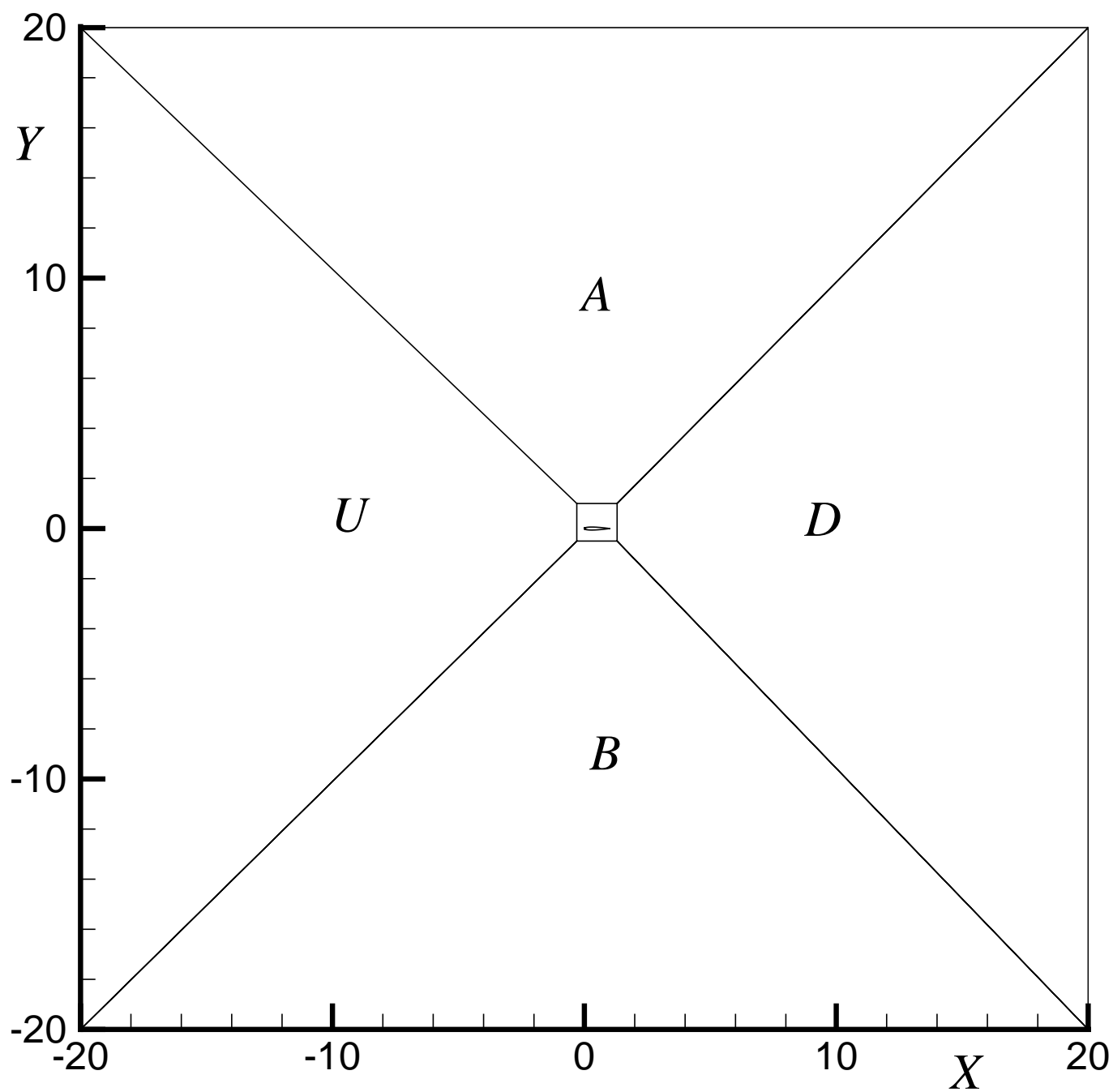


Figure 12: Coarsening regions.

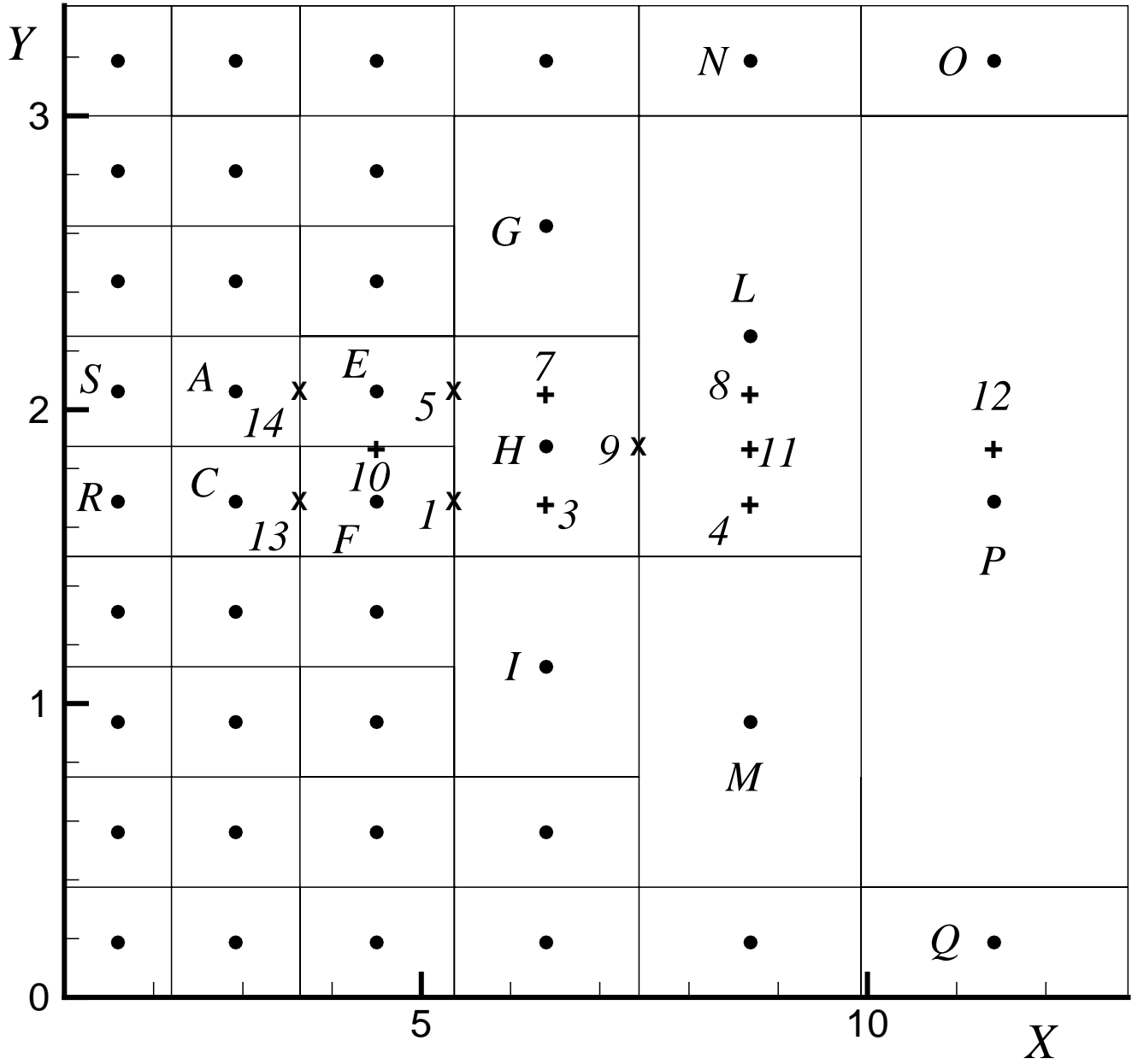


Figure 13: Cell centers surrounding a coarsening element in the downstream coarsening region.

$A \bullet$	$E \bullet$	$H \bullet$	
$B \bullet$			
$C \bullet$	$F \bullet$		
$D \bullet$			

Figure 14: Concatenated coarsening elements.

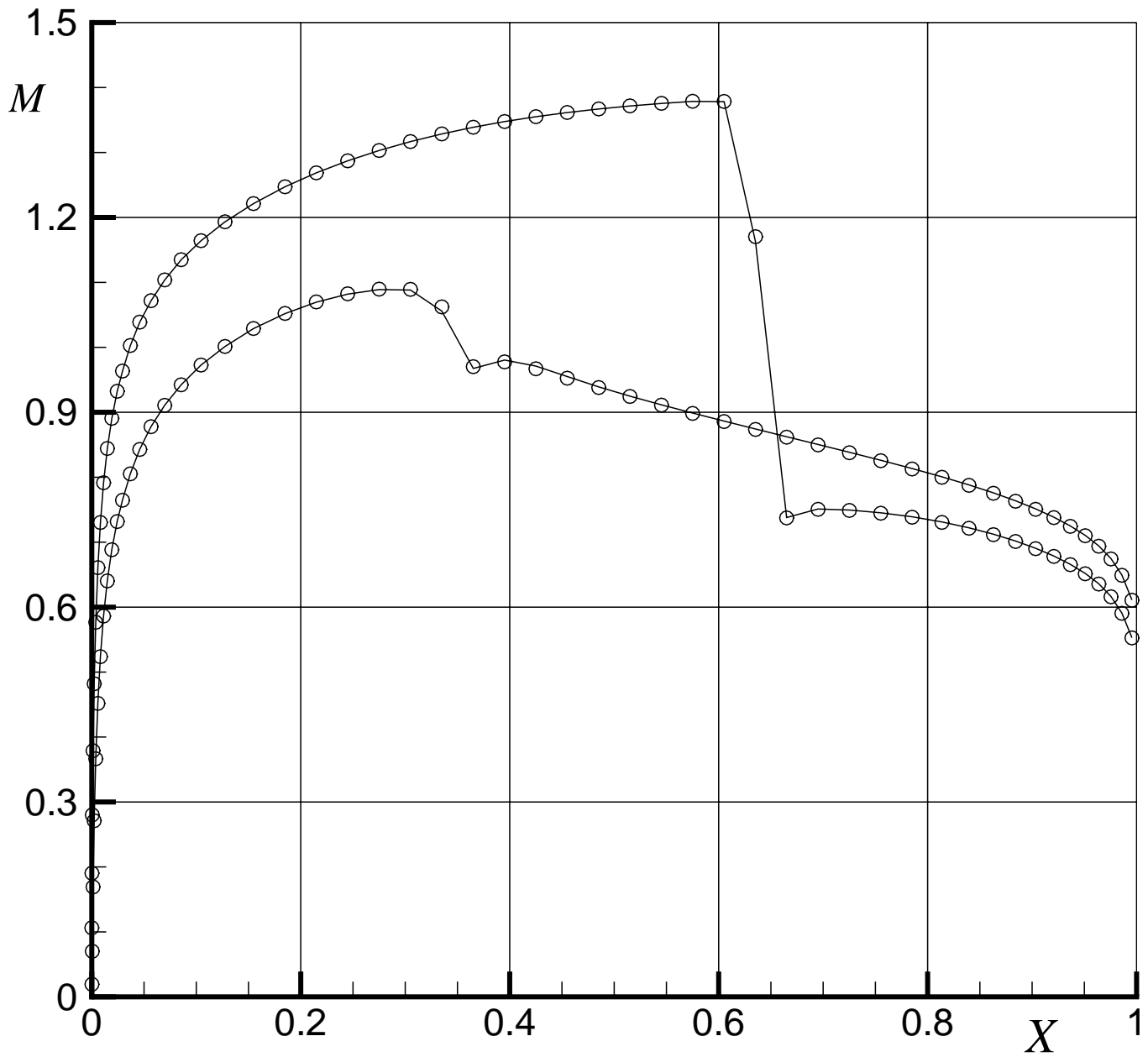


Figure 15: Surface Mach number distribution for NACA 0012 airfoil at $M_\infty = 0.80$, $\alpha = 1.25^\circ$. Coarsened grid (solid line) and uncoarsened grid (symbols).

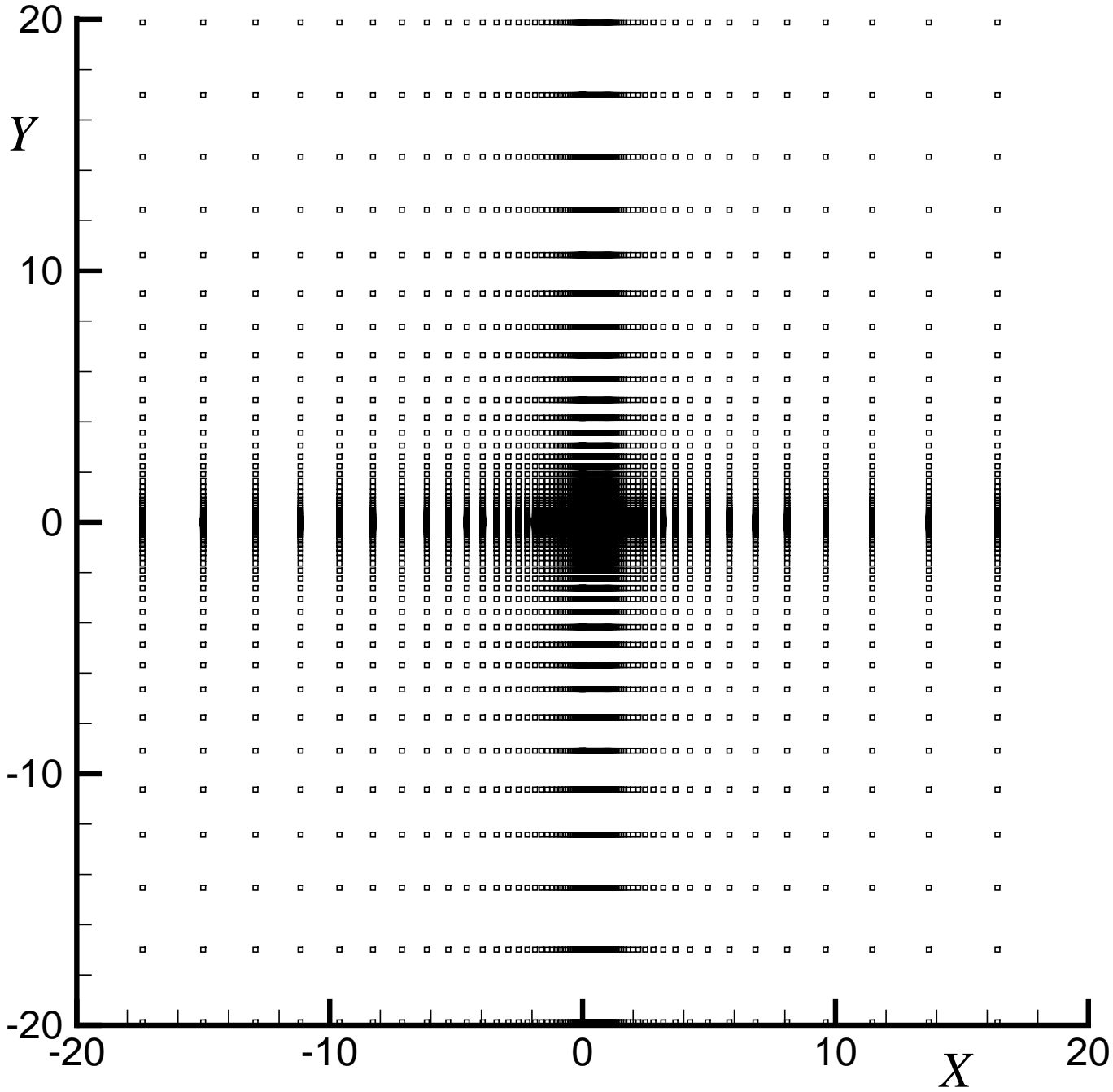


Figure 16: $M_\infty = 0.80$, $\alpha = 1.25^\circ$. Cell center net for the original uncoarsened grid.

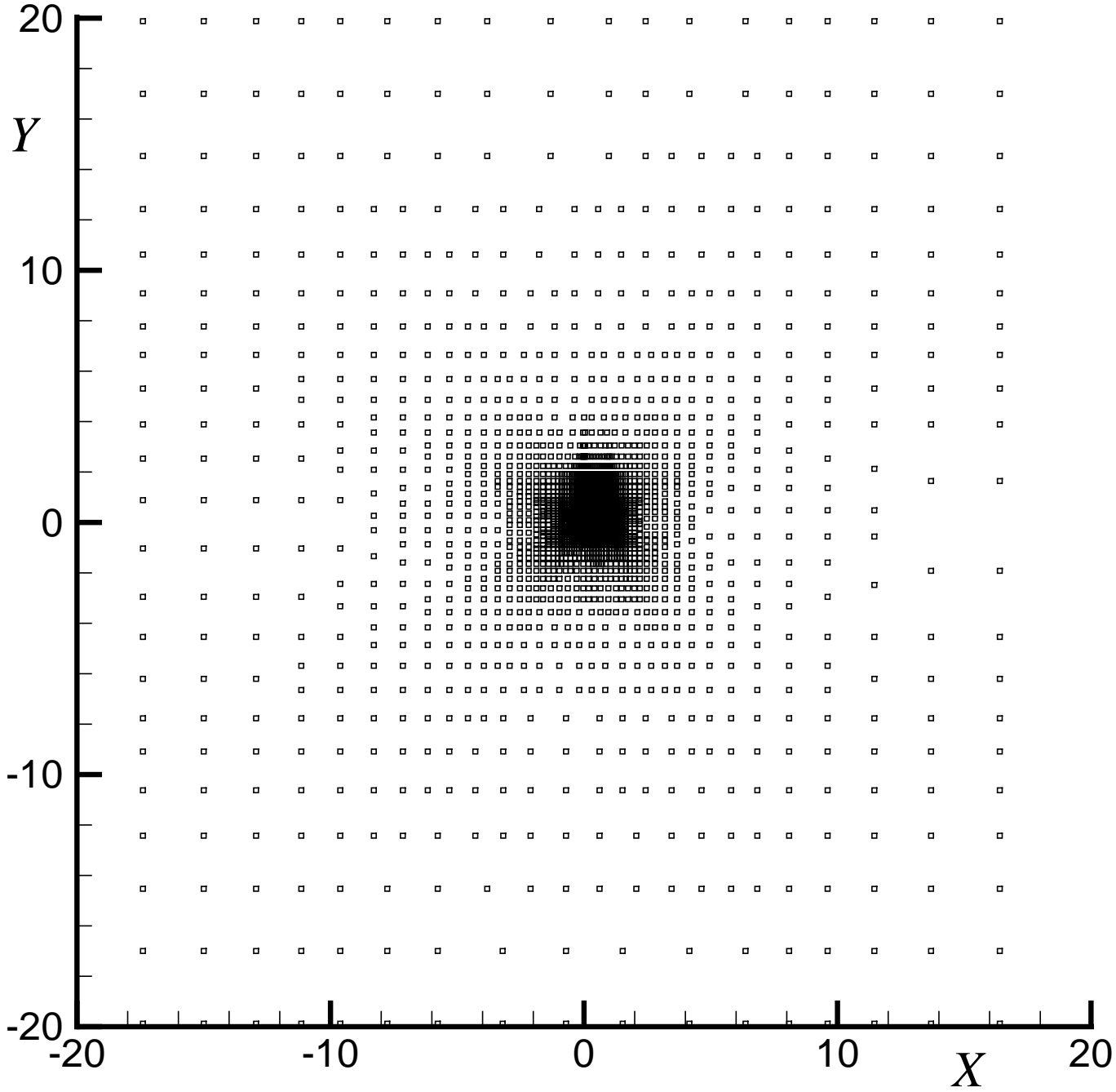


Figure 17: $M_\infty = 0.80$, $\alpha = 1.25^\circ$. Cell center net for the grid with far-field coarsening.

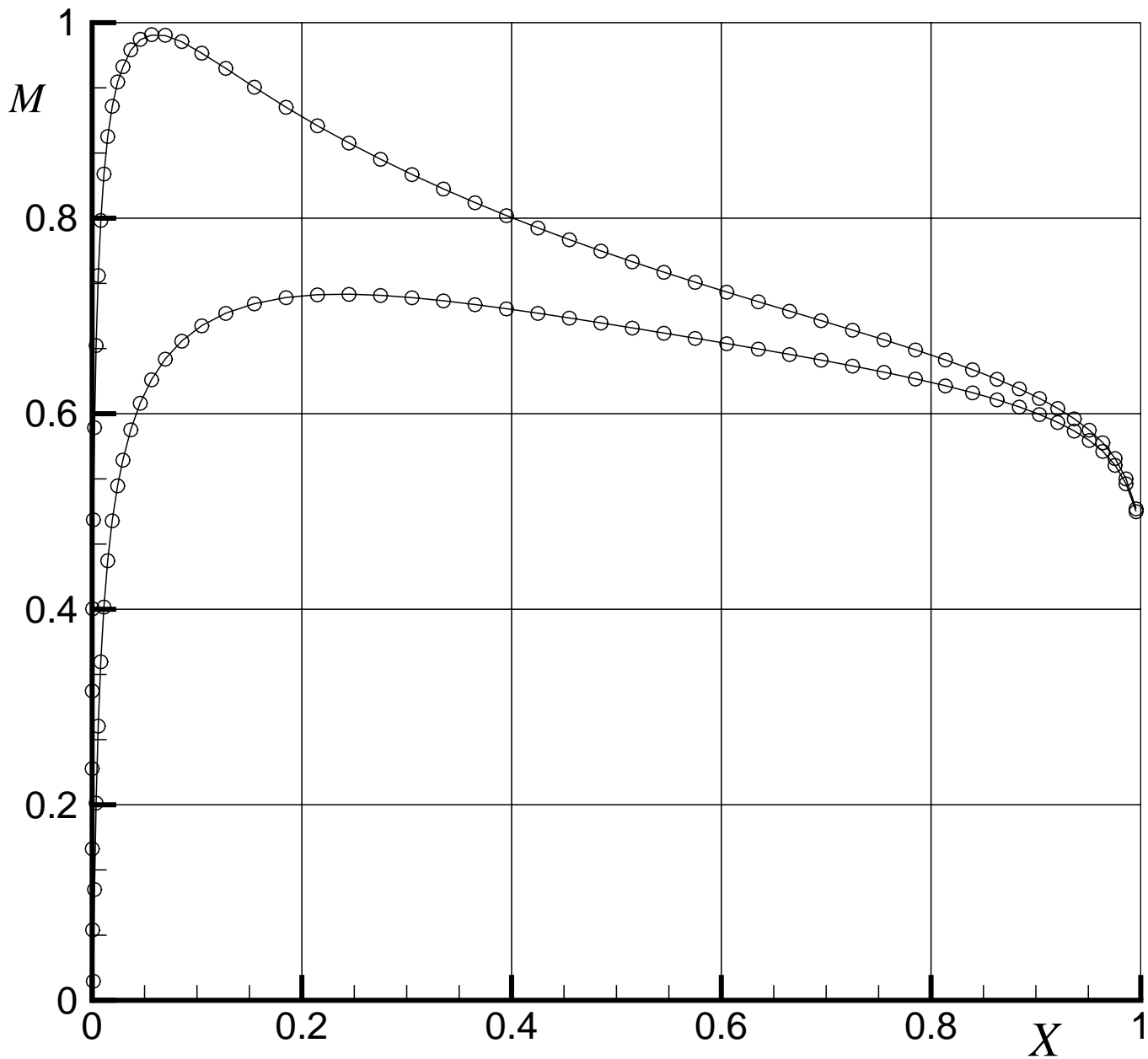


Figure 18: Surface Mach number distribution for NACA 0012 airfoil at $M_\infty = 0.63$, $\alpha = 2^\circ$. Coarsened grid (solid line) and uncoarsened grid (symbols).

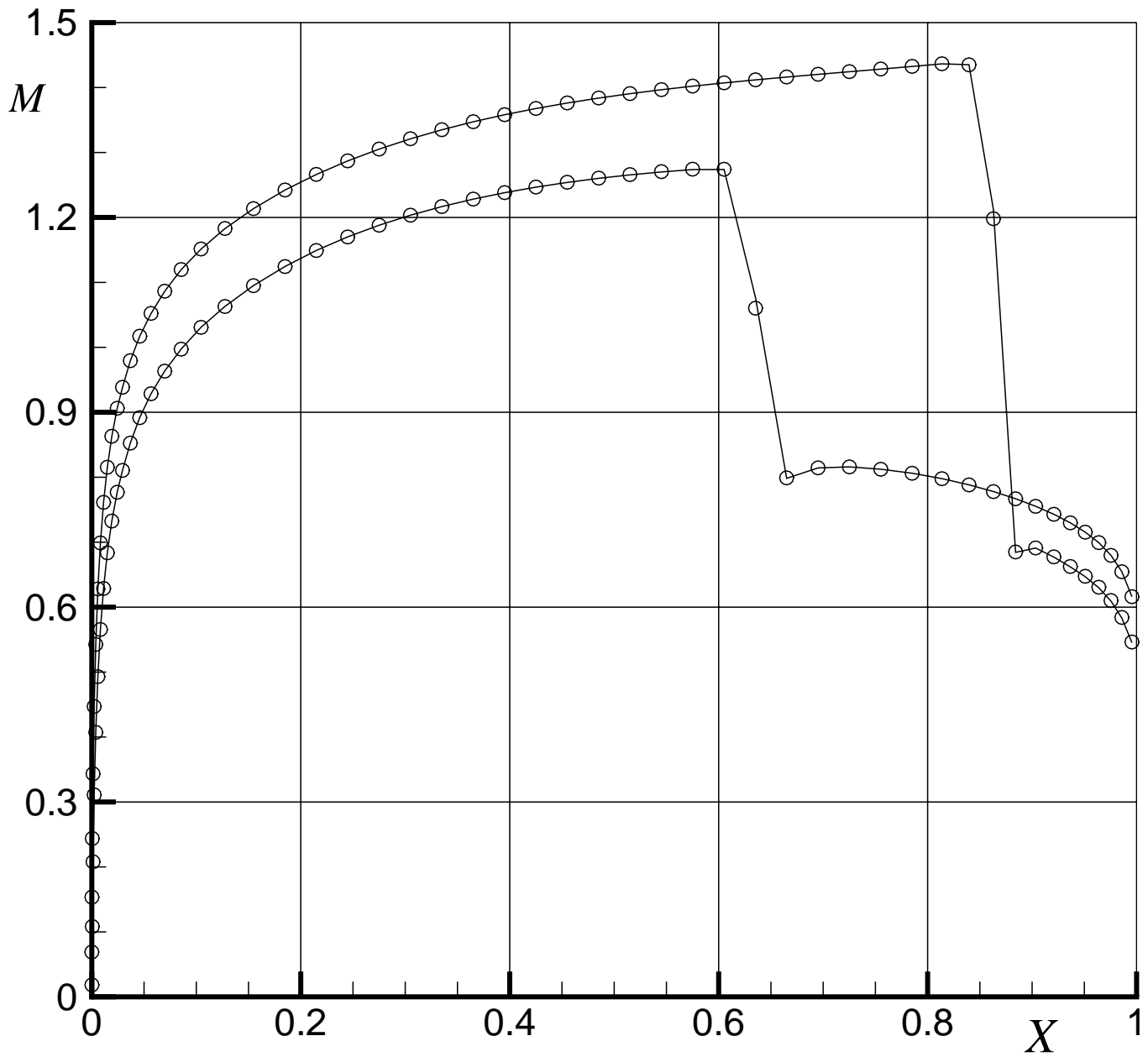


Figure 19: Surface Mach number distribution for NACA 0012 airfoil at $M_\infty = 0.85$, $\alpha = 1^\circ$. Coarsened grid (solid line) and uncoarsened grid (symbols).

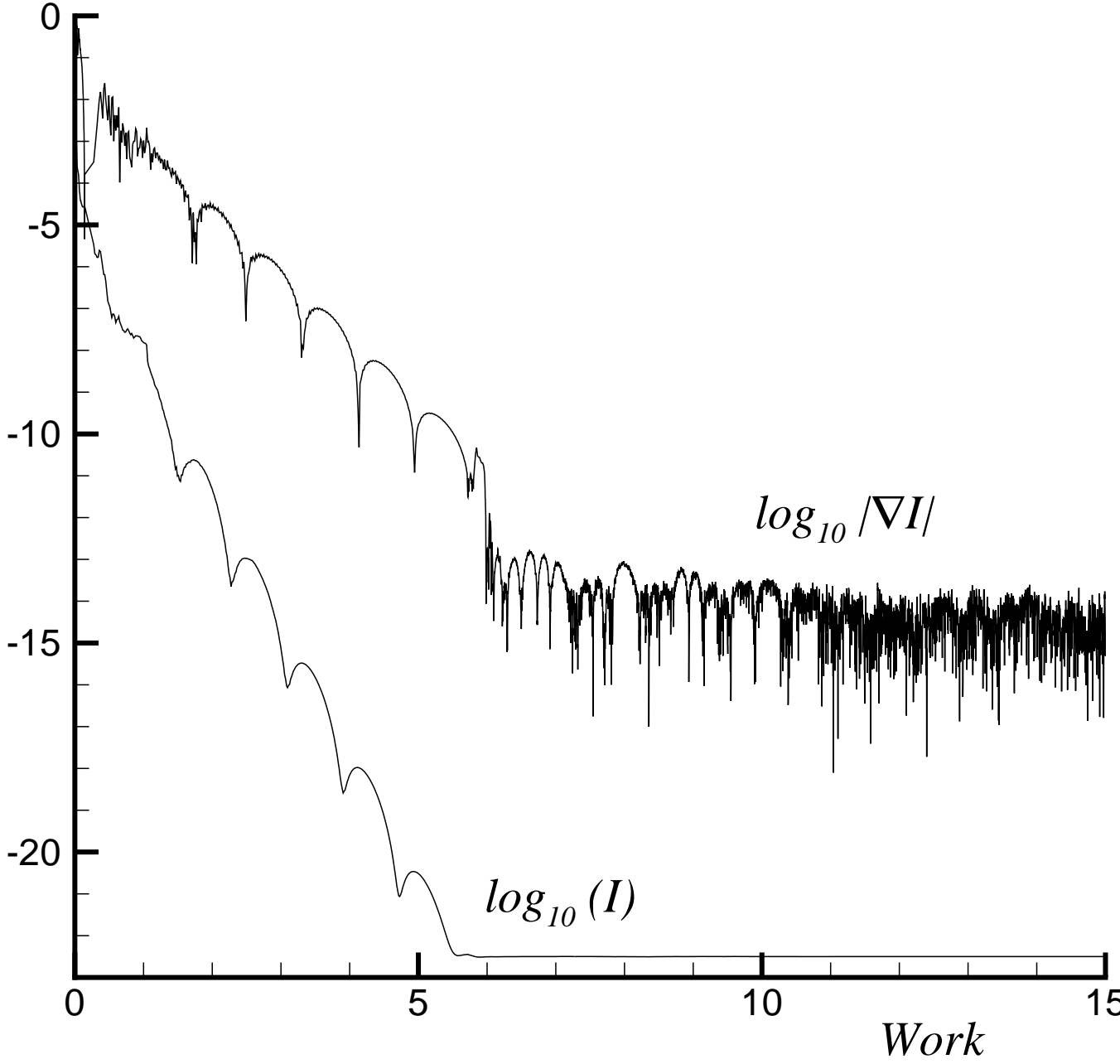


Figure 20: Convergence history for inverse design of a subsonic NACA 0012 airfoil. $M_\infty = 0.63$, $\alpha = 2^\circ$

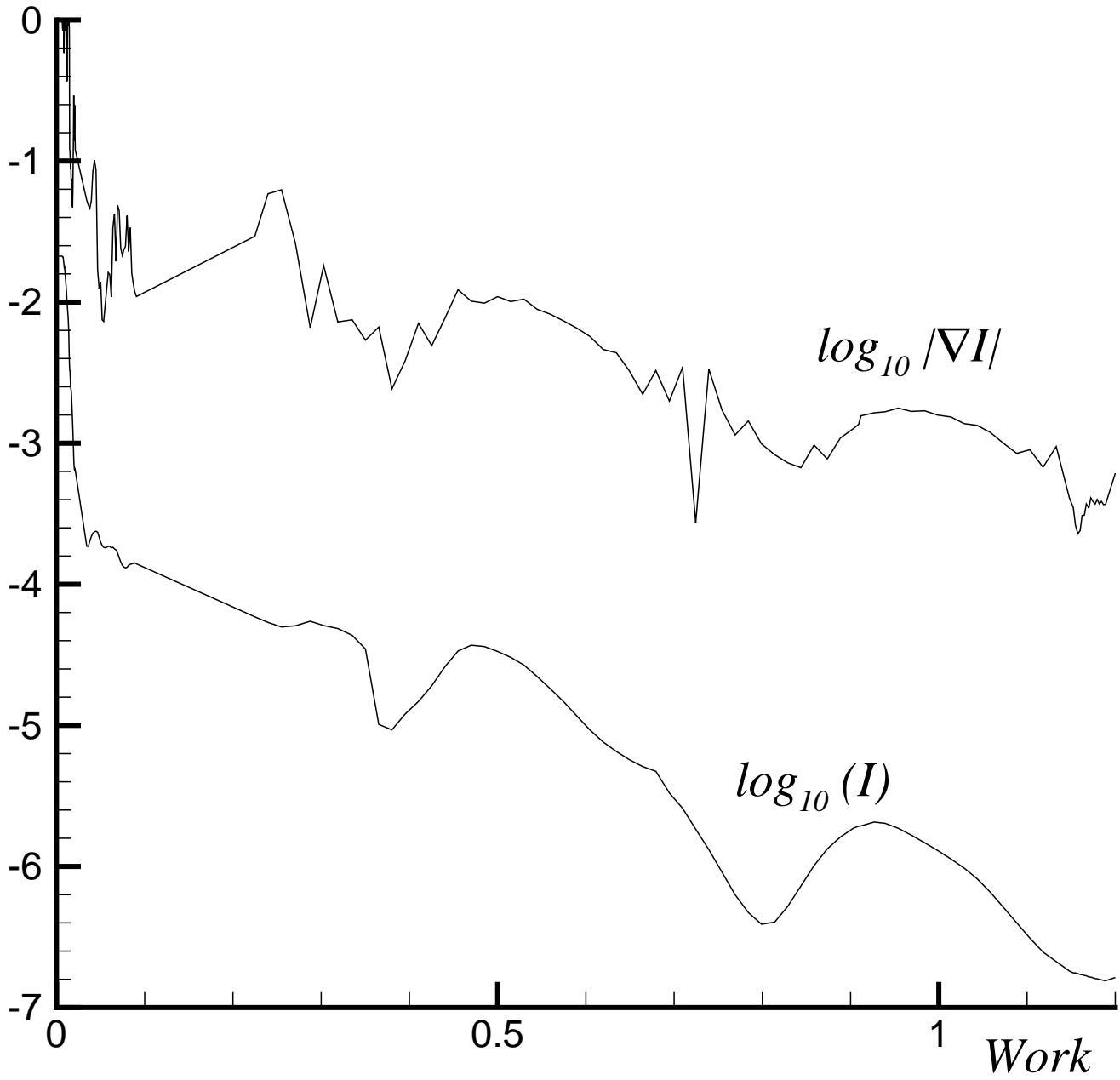


Figure 21: Convergence history for inverse design of a transonic NACA 0012 airfoil. $M_\infty = 0.80$, $\alpha = 1.25^\circ$.

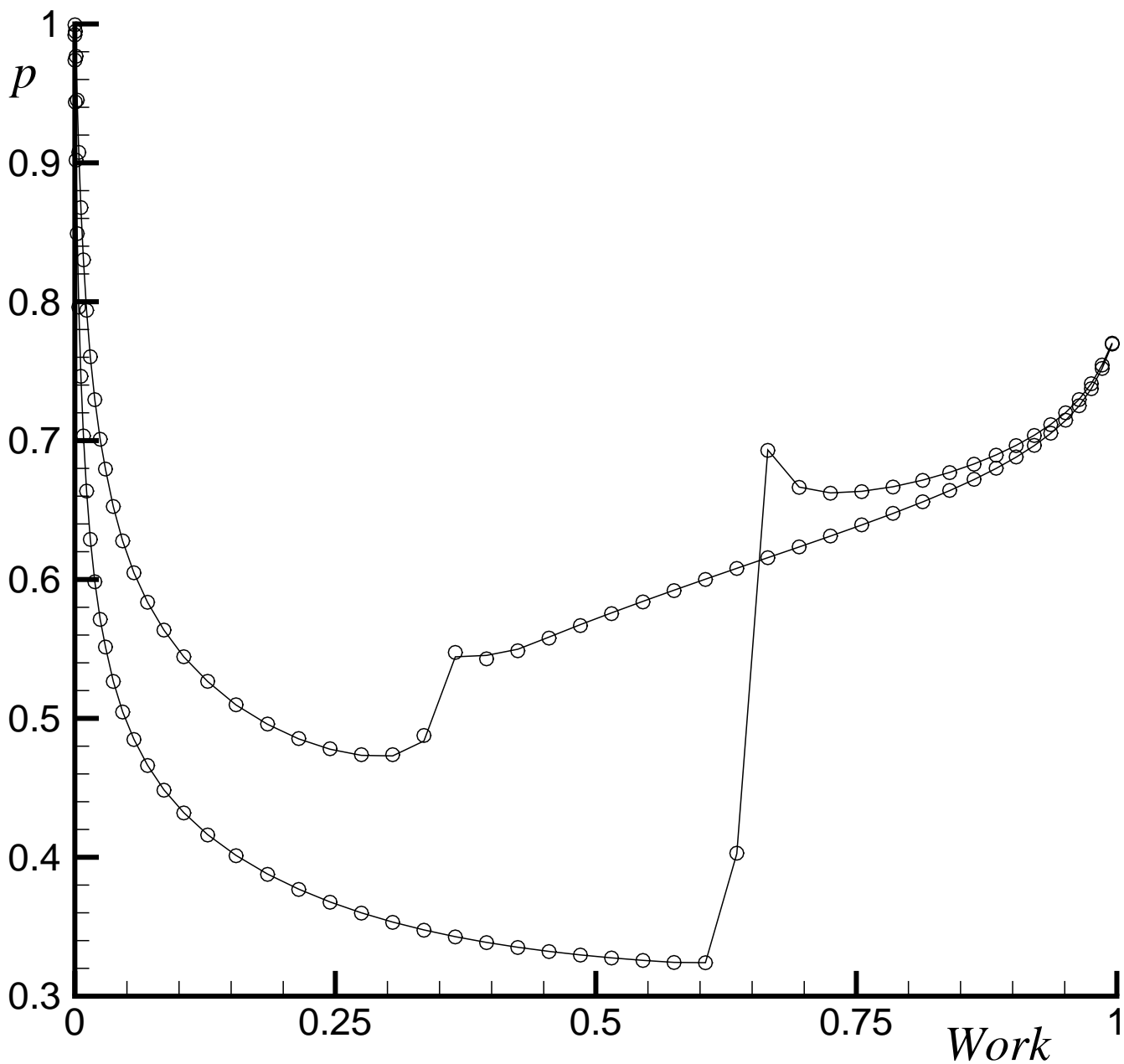


Figure 22: Transonic NACA 0012 airfoil. $M_\infty = 0.80$, $\alpha = 1.25^\circ$. Target (continuous line) and optimized (symbols) pressure distribution on airfoil surface.

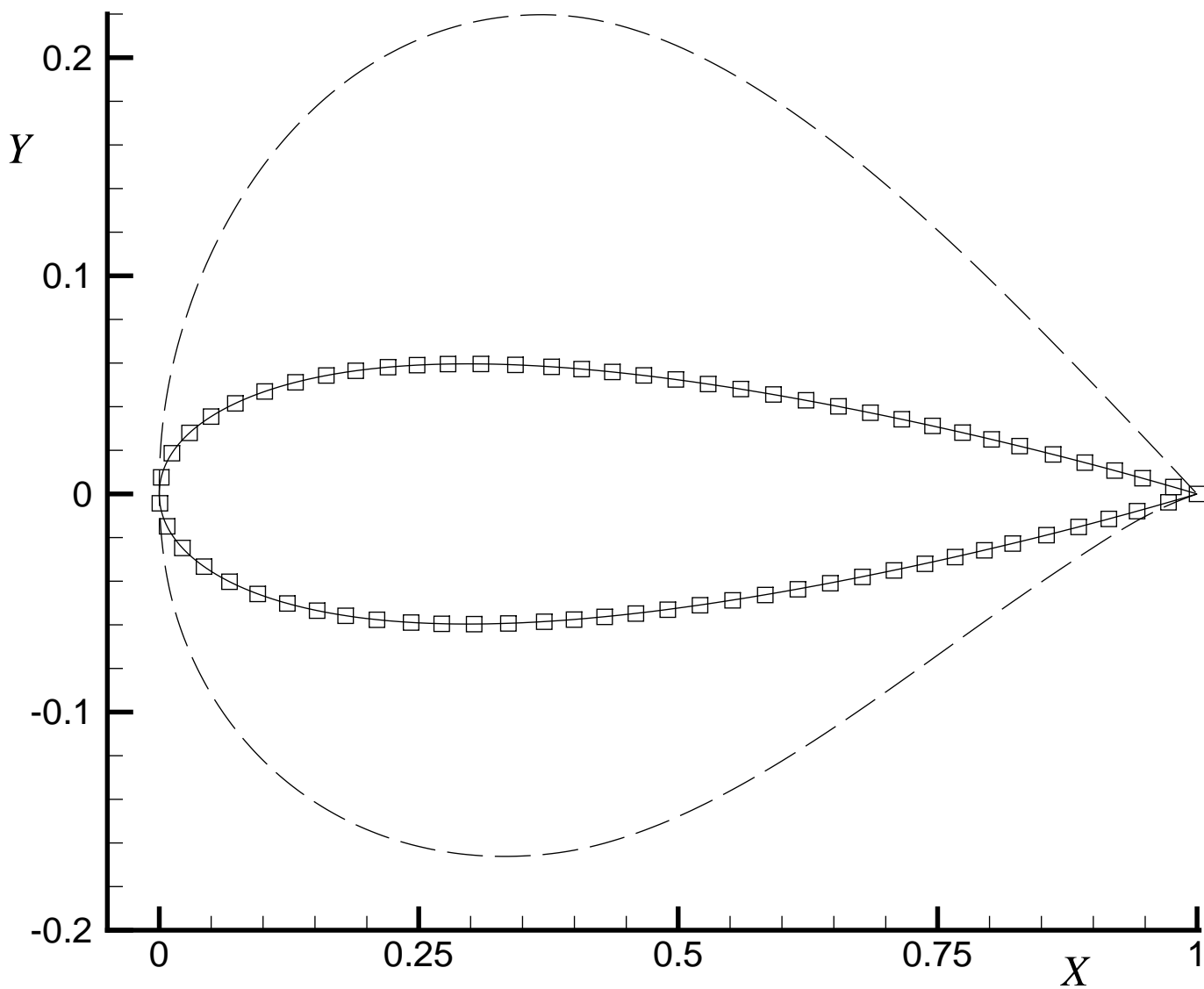


Figure 23: Transonic NACA 0012 airfoil. $M_\infty = 0.80$, $\alpha = 1.25^\circ$. Initial (dashed line), target (continuous line) and optimized (symbols) airfoil shape.

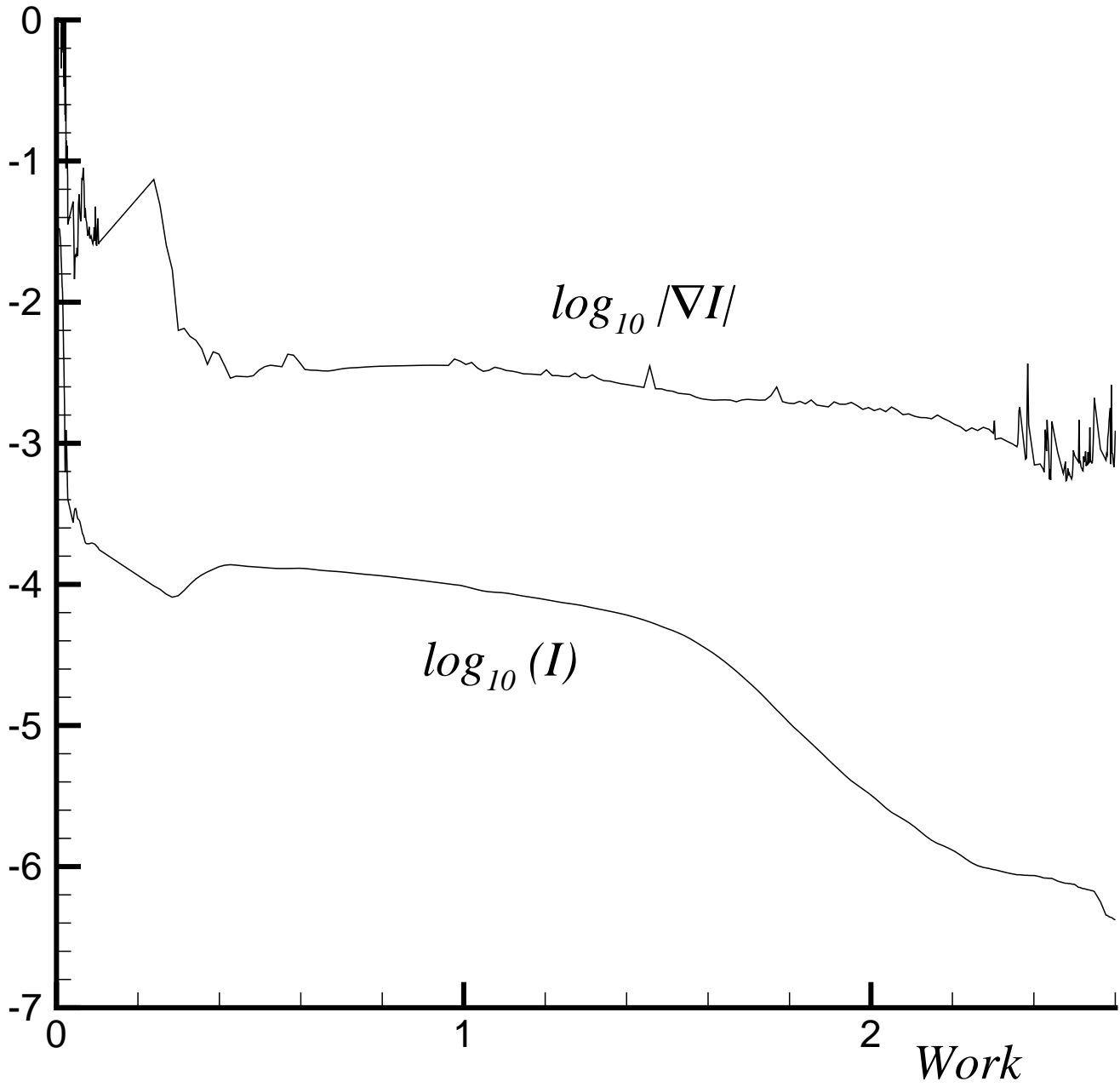


Figure 24: Convergence history for the inverse design of a transonic NACA 0012 airfoil. $M_\infty = 0.85$, $\alpha = 1^\circ$

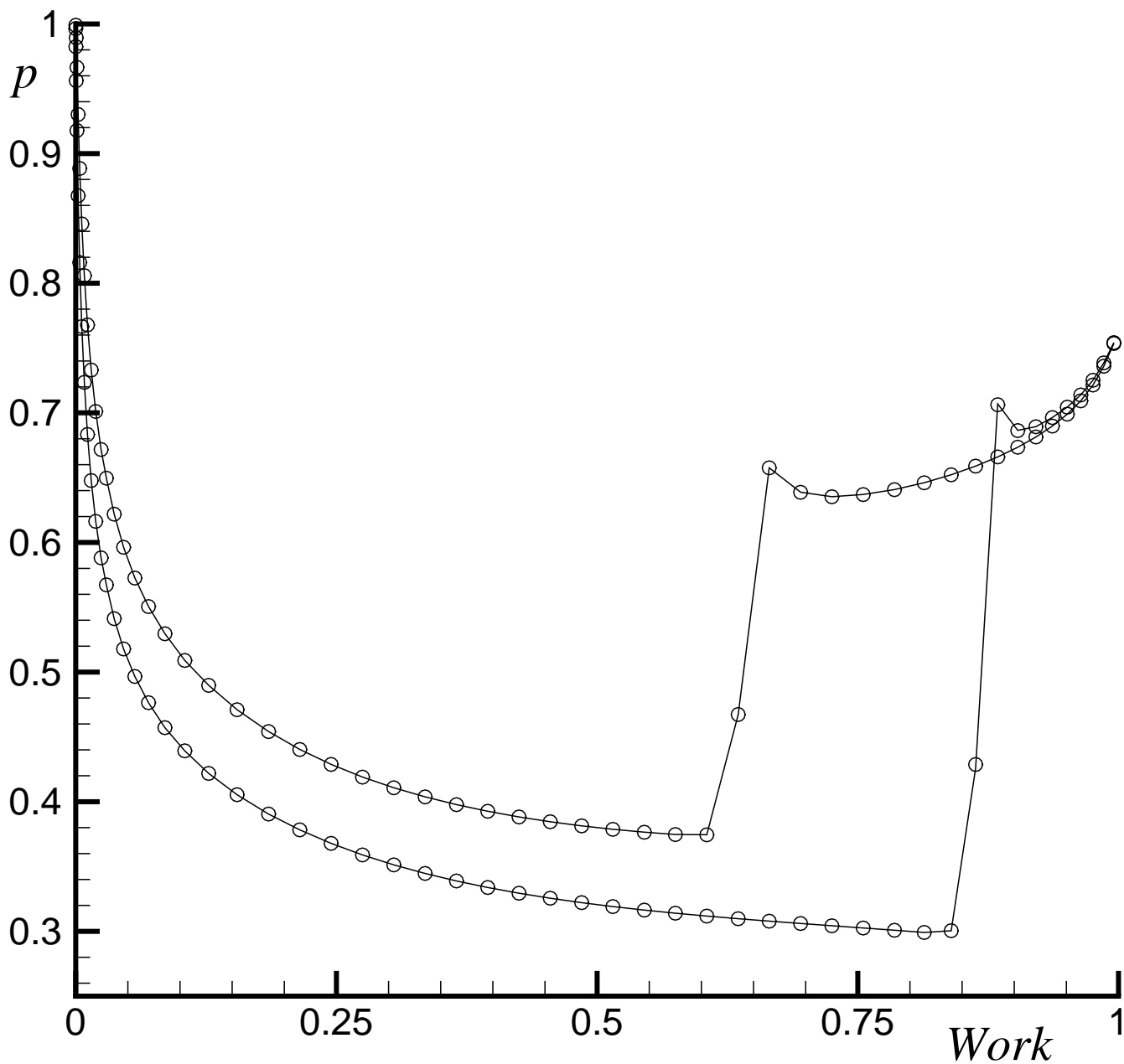


Figure 25: Transonic NACA 0012 airfoil. $M_\infty = 0.85$, $\alpha = 1^\circ$. Target (continuous line) and optimized (symbols) pressure distribution on airfoil surface.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSORING/MONITOR'S ACRONYM(S)	
				11. SPONSORING/MONITORING REPORT NUMBER	
12. DISTRIBUTION/AVAILABILITY STATEMENT					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19b. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)