

High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations

F. Bassi* and S. Rebay†

**Dipartimento di Energetica, Università degli Studi di Ancona, Via Brecce Bianche, 60131 Ancona, Italy; †Dipartimento di Ingegneria Meccanica, Università degli Studi di Brescia,*

Via Branze 38, 25121 Brescia, Italy

E-mail: rebay@bsing.ing.unibs.it

Received August 1, 1995; revised February 23, 1996

This paper deals with a high-order accurate discontinuous finite element method for the numerical solution of the Euler equations. The method combines two key ideas which are at the basis of the finite volume and of the finite element method, the physics of wave propagation being accounted for by means of Riemann problems and accuracy being obtained by means of high-order polynomial approximations within elements. We focus our attention on two-dimensional steady-state problems and present higher order accurate (up to fourth-order) discontinuous finite element solutions on unstructured grids of triangles. In particular we show that, in the presence of curved boundaries, a meaningful high-order accurate solution can be obtained only if a corresponding high-order approximation of the geometry is employed. We present numerical solutions of classical test cases computed with linear, quadratic, and cubic elements which illustrate the versatility of the method and the importance of the boundary condition treatment. © 1997 Academic Press

Key Words: Euler equations; discontinuous Galerkin; higher order accuracy; boundary conditions.

1. INTRODUCTION

This paper deals with a high-order accurate discontinuous finite element method for the numerical solution of the Euler equations on unstructured grids. The method is based on a discontinuous finite element space discretization originally considered by Lesaint and Raviart [16, 17] for the solution of the neutron transport equation and more recently extended by Cockburn *et al.* [9–12] for the solution of nonlinear systems of conservation laws. The method combines two key ideas characterizing

finite element and (upwind) finite volume methods. As in classical (continuous) finite element methods, in fact, accuracy is obtained by means of high-order polynomial approximation within elements rather than by means of wide stencils as in finite volume schemes. As a consequence, there is no need for the interpolation procedures used in finite volume methods which, especially for unstructured meshes (see, e.g., Barth [2], Harten and Chakravarthy [15]), are not at all a trivial task. At the same time, the discontinuous approximation of the solution leads naturally to a treatment of the numerical flux at element interfaces based on the solution of Riemann problems, thus introducing a proper upwinding in the method which, in this respect, is similar to a finite volume scheme. The method is therefore particularly well suited to compute high-order accurate solutions of hyperbolic problems on unstructured grids.

Besides the upwind treatment of the fluxes at element interfaces, high-order discontinuous finite element methods require some additional nonlinear dissipation in order to capture the physically relevant weak solutions of the Euler equations without oscillations. A convenient dissipative mechanism is the local projection limiting procedure introduced by Cockburn and Shu in the series of papers [9–12].

Most of the published works on the discontinuous finite element method (see, e.g., Cockburn and Shu, Bey and Oden [7], and the recent work of Biswas, Devine, and Flaherty [8] on an h-p adaptive discontinuous finite element) concentrate on transient problems on polygonal domains and make use of quadratic polynomials at most. This work is in some sense complementary to the previous ones in that we focus our attention to steady-state subsonic and shockless transonic problems which are computed without any form of limiting and in that we consider domains with curved boundaries. In particular we show the feasibility and the convenience of highly accurate (up to fourth order) steady-state Euler computations on relatively coarse unstructured grids of triangles. Second, and more important, we put in evidence that the method is inaccurate at *curved* solid walls if a piecewise linear approximation of the geometry of the boundary is employed and show that a higher order boundary representation introduces a dramatic improvement in the accuracy of the numerical approximation.

Notice that the accuracy of most finite volume schemes is likewise strongly affected by the treatment of the solid wall slip condition. Over the years, several procedures for prescribing the slip condition have appeared in the literature. Solid wall boundary treatments of improved accuracy (see, for example, the procedure suggested in [13]) show that most of the extrapolation techniques commonly used in cell-centered finite volume methods lead to numerical solutions which are quite less accurate at boundary points than at internal points. However, the discontinuous finite element method requires a more accurate treatment of the slip condition than that allowed by common finite volume methods. In fact we are going to show that it is mandatory to represent accurately the real (curved) geometry of the boundary in order to compute meaningful solutions that otherwise cannot be obtained even on extremely refined grids.

In the section of computational results we present accurate (up to fourth-order) two-dimensional subsonic and shockless transonic computations that emphasize both the versatility of the proposed method and the influence of the solid wall boundary treatment on the solution accuracy.

2. DISCONTINUOUS GALERKIN FORMULATION

We consider the Euler equations written in conservation form,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \quad (1)$$

equipped with suitable initial-boundary conditions. The conservative variables \mathbf{u} and the cartesian components $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ of the flux function $\mathbf{F}(\mathbf{u})$ are given by

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{Bmatrix}; \quad \mathbf{f}(\mathbf{u}) = \begin{Bmatrix} \rho u \\ \rho u u + p \\ \rho v u \\ \rho h u \end{Bmatrix}; \quad \mathbf{g}(\mathbf{u}) = \begin{Bmatrix} \rho v \\ \rho u v \\ \rho v v + p \\ \rho h v \end{Bmatrix}, \quad (2)$$

where ρ is the fluid density, u and v are the velocity components, p is the pressure, and e is the total internal energy per unit mass. The total enthalpy per unit mass h is defined as $h = e + p/\rho$. By assuming that the fluid obeys the perfect gas state equation, p can be calculated as $p = (\gamma - 1)\rho(e - (u^2 + v^2)/2)$, where γ indicates the ratio between the specific heats of the fluid.

By multiplying by a “test function” \mathbf{v} , integrating over the domain Ω , and performing an integration by parts we obtain the weak statement of the problem

$$\int_{\Omega} \mathbf{v} \frac{\partial \mathbf{u}}{\partial t} d\Omega + \oint_{\partial\Omega} \mathbf{v} \mathbf{F}(\mathbf{u}) \cdot \mathbf{n} d\sigma - \int_{\Omega} \nabla \mathbf{v} \cdot \mathbf{F}(\mathbf{u}) d\Omega = 0 \quad \forall \mathbf{v}, \quad (3)$$

where $\partial\Omega$ denotes the boundary of Ω .

A discrete analogue of Eq. (3) is obtained by subdividing the domain Ω into a collection of nonoverlapping elements $\{E\}$ and by considering functions \mathbf{u}_h and \mathbf{v}_h , defined within each element, given by the combination of n shape functions ϕ_i ,

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{i=1}^n \mathbf{U}_i(t) \phi_i(\mathbf{x}), \quad \mathbf{v}_h(\mathbf{x}) = \sum_{i=1}^n \mathbf{V}_i \phi_i(\mathbf{x}) \quad \forall \mathbf{x} \in E. \quad (4)$$

The expansion coefficients $\mathbf{U}_i(t)$ and \mathbf{V}_i denote the degrees of freedom of the numerical solution and of the test function for an element E . The definition of the shape functions ϕ and of the geometry of the elements $\{E\}$ will be given in the following. Notice, however, that there is no global continuity requirement for \mathbf{u}_h and \mathbf{v}_h , which are therefore discontinuous functions across element interfaces.

By splitting the integrals over Ω appearing in Eq. (4) into the sum of integrals over the elements E and by admitting only the functions \mathbf{u}_h and \mathbf{v}_h , we obtain the semidiscrete equations for a generic element E , which can be written as

$$\frac{d}{dt} \int_E \mathbf{v}_h \mathbf{u}_h d\Omega + \oint_{\partial E} \mathbf{v}_h \mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n} d\sigma - \int_E \nabla \mathbf{v}_h \cdot \mathbf{F}(\mathbf{u}_h) d\Omega = 0 \quad \forall \mathbf{v}_h, \quad (5)$$

where ∂E denotes the boundary of element E .

Equation (5) must be satisfied for any element E and for any function \mathbf{v}_h . However, within each element, the \mathbf{v}_h are a linear combination of n shape functions ϕ_j , and Eq. (5) is, therefore, equivalent to the system of n equations

$$\frac{d}{dt} \int_E \phi_j \mathbf{u}_h d\Omega + \oint_{\partial E} \phi_j \mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n} d\sigma - \int_E \nabla \phi_j \cdot \mathbf{F}(\mathbf{u}_h) d\Omega = 0, \quad 1 \leq j \leq n. \quad (6)$$

The Galerkin mechanism therefore constructs automatically a system of n equations which is sufficient to determine the n degrees of freedom $\mathbf{U}_i(t)$ of the unknown solution \mathbf{u}_h .

Due to the discontinuous function approximation, flux terms are not uniquely defined at element interfaces. It is at this stage that the technique traditionally used in finite volume schemes is borrowed by the discontinuous finite element method. The flux function $\mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n}$ appearing in the second term of Eq. (5) is in fact replaced by a numerical flux function $\mathbf{h}(\mathbf{u}_h^-, \mathbf{u}_h^+)$ which depends on the internal interface state \mathbf{u}_h^- , on the neighbouring element interface state \mathbf{u}_h^+ , and on the direction \mathbf{n} normal to the interface. The states \mathbf{u}_h^- and \mathbf{u}_h^+ are calculated by means of the expansion appearing in the first relation of Eq. (4) evaluated at the element interfaces. In order to guarantee the formal accuracy of the scheme, $\mathbf{h}(\mathbf{u}^-, \mathbf{u}^+)$ is required to satisfy the relations

$$\mathbf{h}(\mathbf{u}, \mathbf{u}) = \mathbf{F}(\mathbf{u}) \cdot \mathbf{n}, \quad \mathbf{h}(\mathbf{u}, \mathbf{v}) = -\mathbf{h}(\mathbf{v}, \mathbf{u}), \quad (7)$$

which are called the consistency condition and the directional consistency condition, respectively. There are several numerical flux functions satisfying the above criteria such as the “exact” Riemann flux function or the approximate Lax–Friedrichs, Roe, Engquist–Osher, or Harten–Lax–van Leer (HLL) flux functions. In this work all the computations have been performed with the Riemann flux.

Notice that Eq. (5), with the flux function $\mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n}$, replaced by the numerical flux function $\mathbf{h}(\mathbf{u}_h^-, \mathbf{u}_h^+)$, is nothing but the standard Galerkin method applied to just one element in which the weakly imposed boundary data are obtained from the neighbouring elements. Considering, for example, the Roe flux function

$$\mathbf{h}(\mathbf{u}_h^-, \mathbf{u}_h^+) = \frac{1}{2}[\mathbf{F}(\mathbf{u}_h^+) + \mathbf{F}(\mathbf{u}_h^-)] \cdot \mathbf{n} - \frac{1}{2}|A|(\mathbf{u}_h^+ - \mathbf{u}_h^-) = \mathbf{F}(\mathbf{u}_h^-) \cdot \mathbf{n} + A^-(\mathbf{u}_h^+ - \mathbf{u}_h^-),$$

the equation for the element E can be rewritten as

$$\begin{aligned} \frac{d}{dt} \int_E \phi_j \mathbf{u}_h d\Omega + \oint_{\partial E} \phi_j \mathbf{F}(\mathbf{u}_h^-) \cdot \mathbf{n} d\sigma - \int_E \nabla \phi_j \cdot \mathbf{F}(\mathbf{u}_h) d\Omega - \oint_{\partial E} \phi_j A^- \mathbf{u}_h^- d\sigma \\ = - \oint_{\partial E} \phi_j A^- \mathbf{u}_h^+ d\sigma, \end{aligned}$$

where $A = [\partial \mathbf{F}(\mathbf{u}_h^R) / \partial \mathbf{u}] \cdot \mathbf{n}$ is the Jacobian matrix evaluated at the Roe average state \mathbf{u}_h^R , $|A| = L|\Lambda|R$, $A^- = L\Lambda^-R$, L and R being the matrices of the left and right eigenvectors of A , $|\Lambda| = \text{diag}\{|\lambda_k|\}$ and $\Lambda^- = \text{diag}\{\min(\lambda_k, 0)\}$ being the diagonal matrices of the absolute and of the negative eigenvalues of A . This amounts to

weakly prescribing the variables carried along the characteristics entering the element.

Despite the upwind treatment of the fluxes at the element interfaces, the numerical solutions computed with the method described so far (excluding the special case of a single constant shape function ϕ for each element) are oscillatory in the vicinity of a flow discontinuity if this is not located exactly at element interfaces. As shown by Godunov theorem [14], this is a typical behaviour of any second or high-order accurate numerical scheme in the vicinity of a discontinuous solution. In order to compute physically relevant solutions without oscillations, it is therefore necessary to introduce into the discontinuous finite element method some nonlinear dissipative mechanism which does not destroy the formal order of accuracy of the scheme, such as the local projection limiting strategy proposed by Cockburn *et al.* in [11]. Another possibility is to augment the scheme with a shock-capturing term similar to that commonly considered in the SUPG finite element method (see, e.g., [5]). In this work, however, due to the fact that we concentrate on shockless problems, we did not find it necessary to introduce any kind of limiting procedure.

Let us now describe the geometry of the elements and define the shape functions employed in this work. We have used 2D triangular elements with possibly curved edges in all the computations. It is common practice in the finite element method to deal with such curved elements by introducing a reference element \hat{E} in a nondimensional space and a geometric transformation which maps the reference element onto the real elements in the physical space. The mapping from the reference element to the real element is a polynomial function \hat{P}^m of order less than or equal to m defined in the space of the reference element for each independent variable. Since the mapping must guarantee the geometric continuity between neighbouring elements, it is commonly expressed in terms of Lagrangian functions $\hat{\psi}_i^m$ and Lagrangian node coordinates \mathbf{x}_i and is written as

$$\mathbf{x}_h(\xi) = \sum_{i=0}^l \mathbf{x}_i \hat{\psi}_i^m(\xi) \quad \forall \xi \in \hat{E}, \quad (8)$$

where ξ is the independent variable in the reference space of \hat{E} . Notice that the Lagrangian nodes of the element edges belonging to the boundary $\partial\Omega$ are placed on the real geometry of the boundary. A curved boundary is therefore approximated by a set of piecewise polynomial curved segments which individually are C^m , but which globally form a curve that in general is only C^0 .

Also the functions \mathbf{u}_h and ϕ_j are defined as polynomial functions on the reference element. We therefore introduce the functions $\hat{\mathbf{u}}_h$ and $\hat{\phi}_j$ which are polynomials \hat{P}^k of degree less than or equal to k within the reference element \hat{E} , i.e.

$$\hat{\mathbf{u}}_h(\xi, t) = \sum_{i=1}^n \mathbf{U}_i(t) \hat{\phi}_i^k(\xi) \quad \forall \xi \in \hat{E}. \quad (9)$$

The functions \mathbf{u}_h and ϕ_i^k for a generic real element E are formally obtained as

$$\mathbf{u}_h(\mathbf{x}_h, t) = \hat{\mathbf{u}}_h[\xi(\mathbf{x}_h), t], \quad \phi^k(\mathbf{x}_h) = \hat{\phi}^k[\xi(\mathbf{x}_h)] \quad \forall \mathbf{x}_h \in E, \quad (10)$$

where $\xi = \xi(\mathbf{x}_h)$ is the inverse of the mapping of Eq. (8). Notice that the functions \mathbf{u}_h and ϕ_j are not polynomials for a generic element. However, in the case of linear mapping (i.e., of real elements with straight edges), the functions \mathbf{u}_h and ϕ_j are in fact polynomials P^k also in the physical space.

Equation (10), however, is not required explicitly, since the integrals of Eq. (6) are evaluated in the space of the reference element by means of numerical quadrature formulae. As a consequence we only need Eq. (9) to represent the unknowns and Eq. (8) to compute the determinant of the inverse Jacobian matrix of the transformation at each quadrature point.

The number of quadrature points used is chosen to integrate exactly on the reference element polynomials of order $2k$. In the case of linear, quadratic, and cubic shape functions ϕ , we evaluate the volume integrals with Gauss formulae using three, six, and twelve points and the interface and the boundary integrals with formulae using two, three, and four points, respectively.

We can distinguish different types of elements by the order k of the shape functions $\hat{\phi}^k$ used to represent the unknowns and by the order m of the shape functions $\hat{\psi}^m$ of the geometric mapping. The element is called isoparametric if $m = k$, subparametric if $m < k$, and superparametric if $m > k$. In all the computations we are going to present in the following section we have used the complete polynomial basis of order m or k , both for the unknowns and for the geometric transformations.

By assembling together all the elemental contributions (6), the system of ordinary differential equations which govern the evolution in time of the discrete solution can be written as

$$M \frac{d\mathbf{U}}{dt} + R(\mathbf{U}) = 0, \quad (11)$$

where M denotes the mass matrix, \mathbf{U} is the global vector of the degrees of freedom, and $R(\mathbf{U})$ is the residual vector. Since the shape functions $\phi|_E$ are nonzero within element E only, the mass matrix M has a block diagonal structure which couples the n degrees of freedom of each unknown variable only within element E . As a consequence M can be very simply inverted considering one element at a time. The time integration of the semidiscrete system is accomplished by means of a Runge–Kutta explicit method for initial value problems. In practice we have used the Runge–Kutta methods advised by Cockburn and Shu in [9], which are linearly stable for a Courant number less than or equal to $1/(2k + 1)$. This rather restrictive CFL condition might suggest looking for a more efficient integration scheme. Preliminary work in this direction seems to indicate that an implicit method is indeed more efficient, especially for steady state computations. In the present work, however, this topic has not been addressed.

3. BOUNDARY TREATMENT

When the boundary ∂E of an element E belongs to $\partial\Omega$, the normal flux function $\mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n}$ must be consistent with the boundary condition to be imposed on $\partial\Omega$ and

will be denoted by \mathbf{h}_{bc} . The boundary normal flux function \mathbf{h}_{bc} is computed in terms of the numerical flux function as $\mathbf{h}_{bc} = \mathbf{h}(\mathbf{u}_{\bar{h}}, \mathbf{u}_{bc})$, where $\mathbf{u}_{\bar{h}}$ is the internal boundary state at the current time level and \mathbf{u}_{bc} is chosen according to the conditions that must be satisfied on the boundary. At inflow/outflow boundaries, the state \mathbf{u}_{bc} is computed by means of a locally one-dimensional characteristic analysis in a direction normal to the boundary by imposing the available data and the Riemann invariants associated to outgoing characteristics. On solid walls a symmetry technique is used whereby the state \mathbf{u}_{bc} has the same density, internal energy, and tangential velocity component of $\mathbf{u}_{\bar{h}}$ and the opposite normal velocity component.

The simplest element considered is the isoparametric linear element ($k = m = 1$). Since the geometric mapping is piecewise linear, the boundary of the domain is in this case discretized as the polygonal line that joins the grid points. Such a linear geometric discretization of the boundary gives rise to errors in the numerical solution which, for the Euler equations in conservation form, appear as spurious entropy production near solid walls. The usual remedy for this accuracy loss consists of clustering the grid points in the regions of high curvature. In general, accurate solutions are found, provided that the mesh is fine enough to represent smoothly the boundary geometry. However, recent work on boundary conditions for finite volume methods applied to the Euler equations indicates that, on a given mesh, much more accurate results are obtained by using a solid wall boundary treatment that takes into account the wall curvature (see, e.g., [13]).

During earlier attempts to compute the numerical solution of rather standard subsonic and transonic flows over airfoils [3–5], we have discovered that the discontinuous finite element method suffers from a piecewise linear approximation of the geometry of solid boundaries much more than standard finite volume schemes. The computations presented in the following section seem to indicate that a high-order geometric approximation of curved boundaries is *mandatory* if accurate numerical solutions are sought (i.e., we must use $m \geq 2, \forall k$). In fact, we have not been able to compute accurate numerical results using geometrically linear elements even on highly refined grids.

4. NUMERICAL RESULTS

The discontinuous finite element method has been implemented in a computer code that can use elements of different geometric shape and high order polynomial approximations both for the unknowns and for the geometric mapping functions.

Due to the explicit nature of the time stepping scheme, we have chosen an implementation aimed to minimize the computer time whereby the block diagonal mass matrix, the determinant of the Jacobian of the coordinate transformation, and the gradient of the shape functions are stored at the Gauss points of each element once for all at the beginning of the computation. With this choice, the memory requirement per element is of 22 integer words and 85 real (double precision) words for linear elements, of 33 integer words and 216 real words for quadratic elements and of 46 integer words and 518 real words for cubic elements. However, if the mass matrix, the Jacobian determinants, and the shape function gradients are recomputed at every iteration, the storage requirement reduces by a factor ranging from

2 for linear elements to 3.5 for cubic elements, while the computer time increases by about 50% in all cases.

All the computations (which converged to a steady-state solution) have been marched to a 10^{-10} error in the L_2 norm of the residuals and have been performed using double precision arithmetic on a HP735-125 workstation. The CPU time per time step (two Runge–Kutta stages) per element is of 0.104, 0.244, and 0.512 ms for linear, quadratic, and cubic elements, respectively. For the medium density 32×8 grid considered in the next section, the computations performed with linear, quadratic, and cubic elements took 6560, 21559, and 95578 time steps to converge to a density residual L_2 norm equal to 10^{-10} .

The results presented in the following sections are computed on very regular unstructured grids of triangles. Other results [5] show that the quality of the solution does not significantly reduce by employing more general triangulations obtained with mesh generation algorithms which can treat arbitrary domains.

4.1. Flow around a Circle

We first present the results obtained for the 2D subsonic flow around a circle at Mach number $M_\infty = 0.38$, a typical test case considered for the evaluation of numerical methods for the Euler equations.

In order to check the accuracy and the convergence properties of the method, we have performed various computations using different combinations of interpolation functions for the unknowns and for the geometric mapping from the reference element to the real elements. Different elements are denoted by P_kQ_m , where k indicates the order of the complete polynomials used to approximate the unknowns and m indicates the order of the complete polynomials used for the geometric mapping.

We present computations with linear, quadratic, and cubic approximation of the unknowns and with linear, quadratic, and cubic geometric mapping. The various computations have been performed on four successively refined grids shown in Figs. 1–4, which are the Delaunay triangulations obtained from the distributions of 16×4 , 32×8 , 64×16 , and 128×32 points, respectively. All the grids extend about 20 diameters away from the circle.

4.1.1. Second-Order Accurate Computations

Figures 5–8 show the Mach isolines computed with P1Q1 isoparametric elements on the four grids. The isolines are plotted for values of the Mach number given by $M_i = M_0 + i \Delta M$, $i = 0, 1, \dots$, where $M_0 = 0$ and $\Delta M = 0.038$. These solutions are very inaccurate, as put in evidence by the nonphysical “boundary layer” which develops along the solid wall and by the associated “wake” in the downstream region of the circle. Moreover, the computations do not converge to a steady solution because of the unsteadiness of the unphysical wake. The Mach isolines shown in the figures are therefore those at some instant in time which corresponds to 100,000 Runge–Kutta cycles. Notice that the poor behaviour of the numerical solution at the boundary does not disappear even on the very refined 128×32 grid.

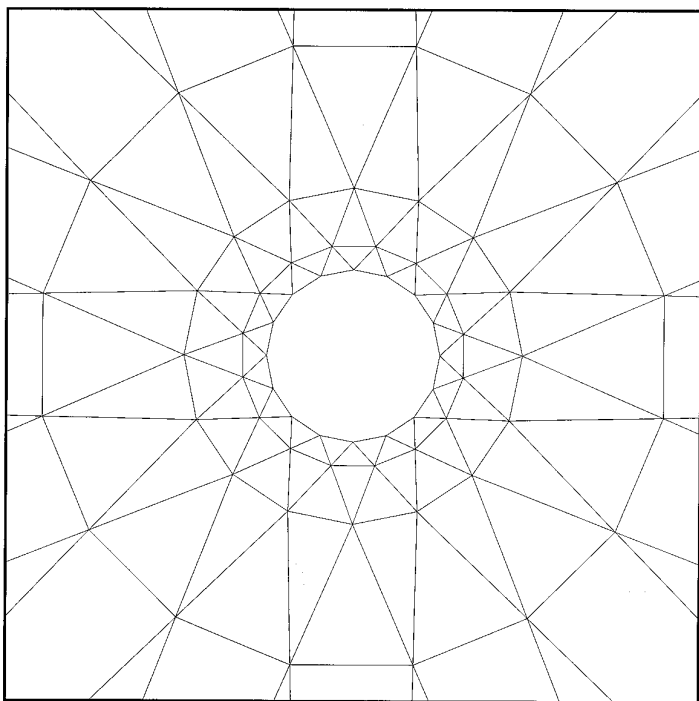


FIG. 1. 16×4 grid around a circle.

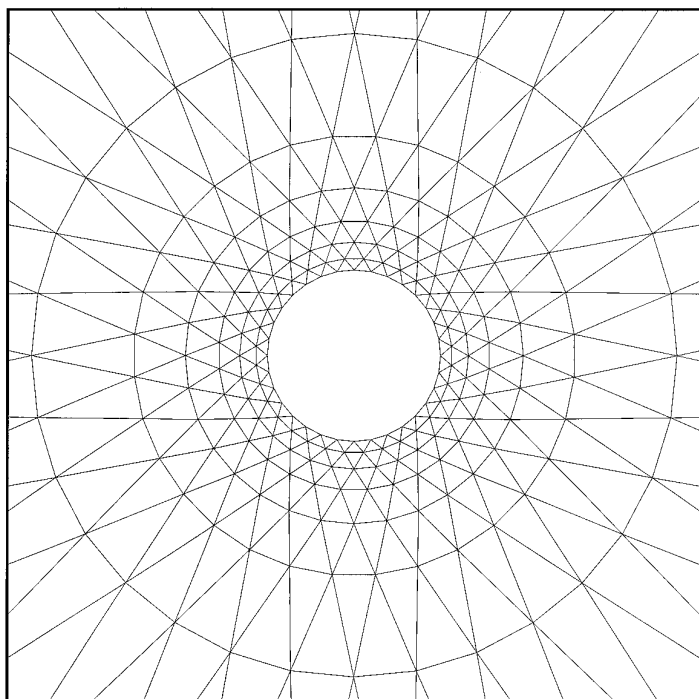


FIG. 2. 32×8 grid around a circle.

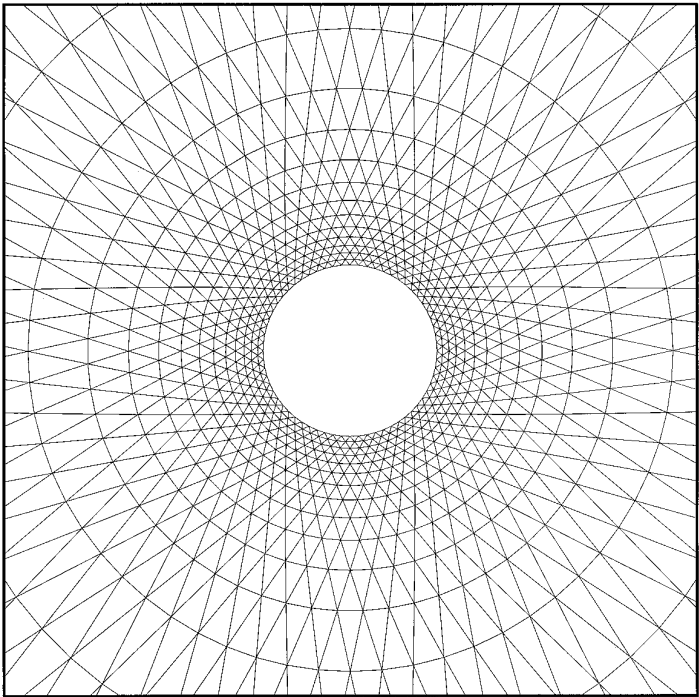


FIG. 3. 64×16 grid around a circle.

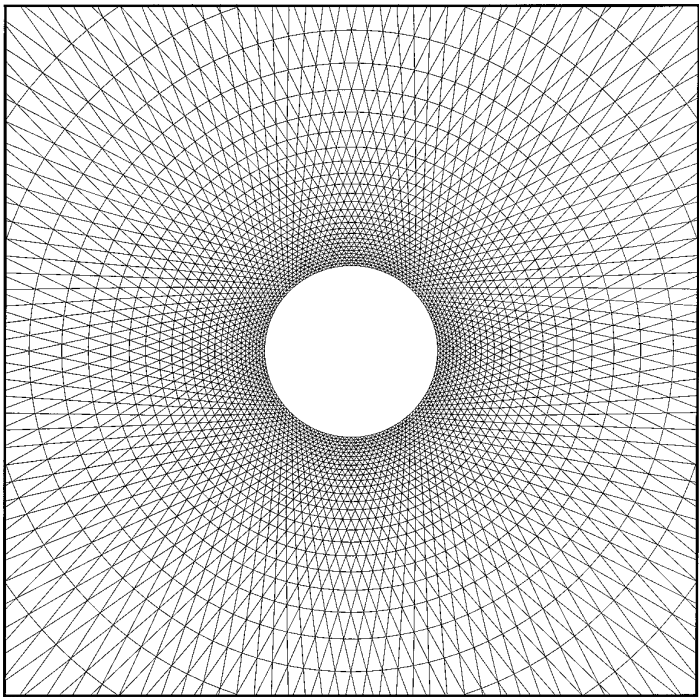


FIG. 4. 128×32 grid around a circle.

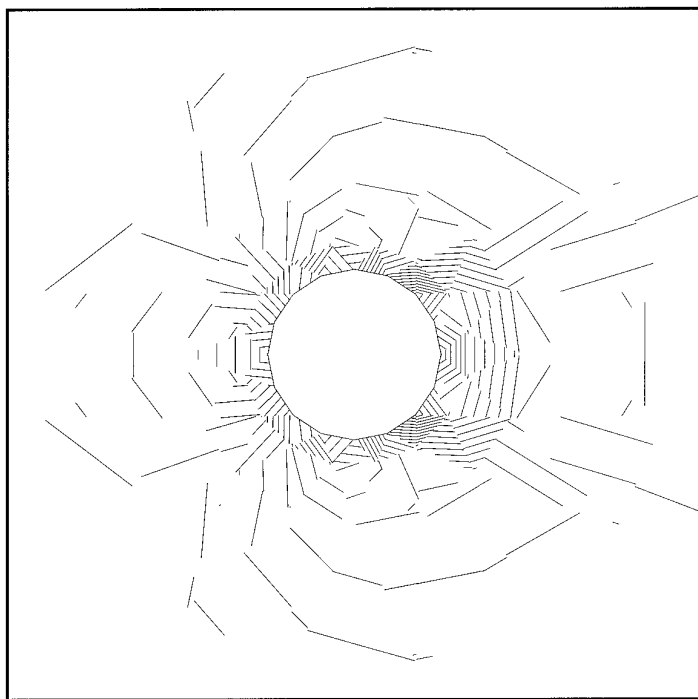


FIG. 5. Mach isolines around a circle with P1Q1 elements on the 16×4 grid.

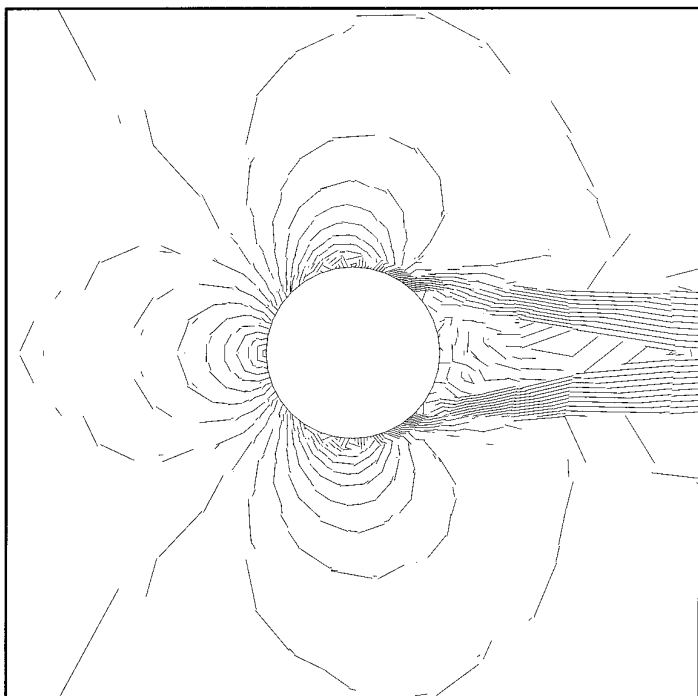


FIG. 6. Mach isolines around a circle with P1Q1 elements on the 32×8 grid.

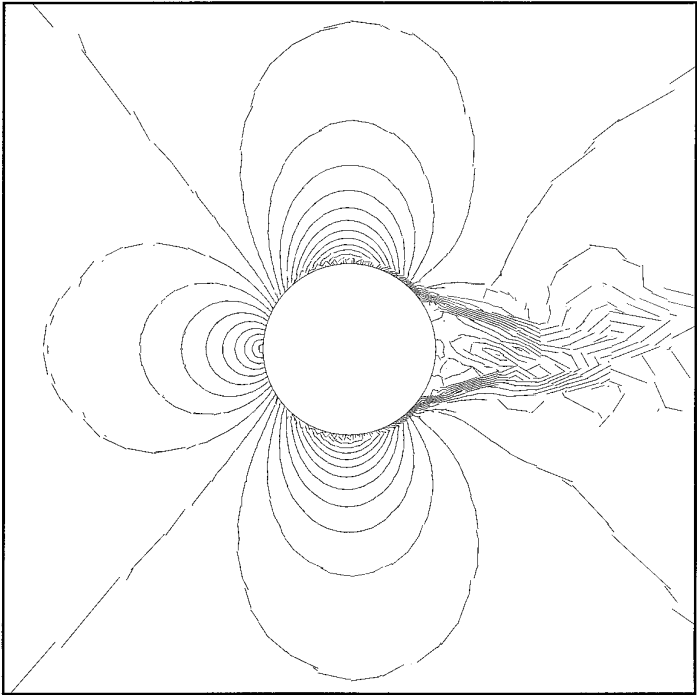


FIG. 7. Mach isolines around a circle with P1Q1 elements on the 64×16 grid.

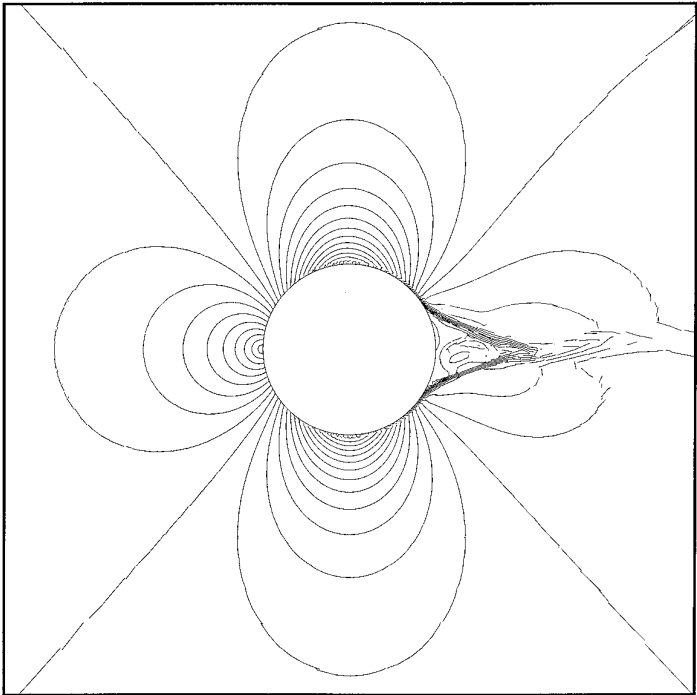


FIG. 8. Mach isolines around a circle with P1Q1 elements on the 128×32 grid.

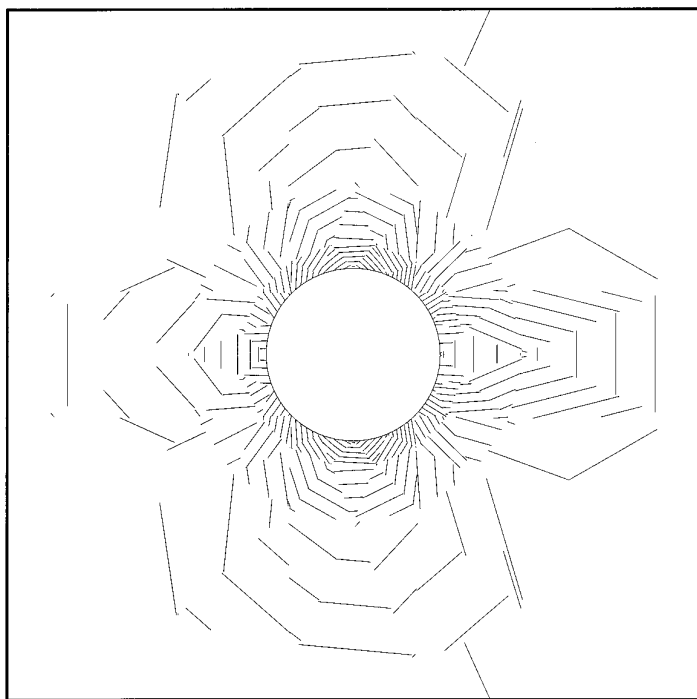


FIG. 9. Mach isolines around a circle with P1Q2 elements on the 16×4 grid.

The picture changes completely when using P1Q2 superparametric elements, as reported in Figs. 9–12, which show the Mach isolines computed on the four successively refined grids. In this case the numerical solution converges nicely to a solution which is (nearly) symmetric in the upstream–downstream direction and which is characterized by a limited entropy production along the wall, as reported in Table I which shows the L_2 entropy error and the order of convergence of the numerical solutions in comparison with other higher order ones (to be discussed in the following).

In order to gain some insight into the unexpected poor behaviour at a curved boundary of the P1Q1 discontinuous Galerkin finite element computations, we have performed some test calculations with a hybrid element (called P1Q1*) in which the boundary edges are regarded as straight lines as in Q1 elements but the normal direction changes locally along the boundary according to the real curved geometry as in Q2 elements.

Figures 13–16 show the Mach isolines computed with P1Q1* elements. The solution is more accurate than that computed with P1Q1 elements but remains much more inaccurate than that computed with P1Q2 elements. We have also performed other computations (not reported for reasons of brevity) using curved boundary edges and constant normal direction along each boundary edge, which resulted in less accurate solutions than those computed with P1Q1* elements. This means that the accuracy gain obtained by using a high-order representation of the geometry of the boundary is mainly related to the correct distribution of the normal

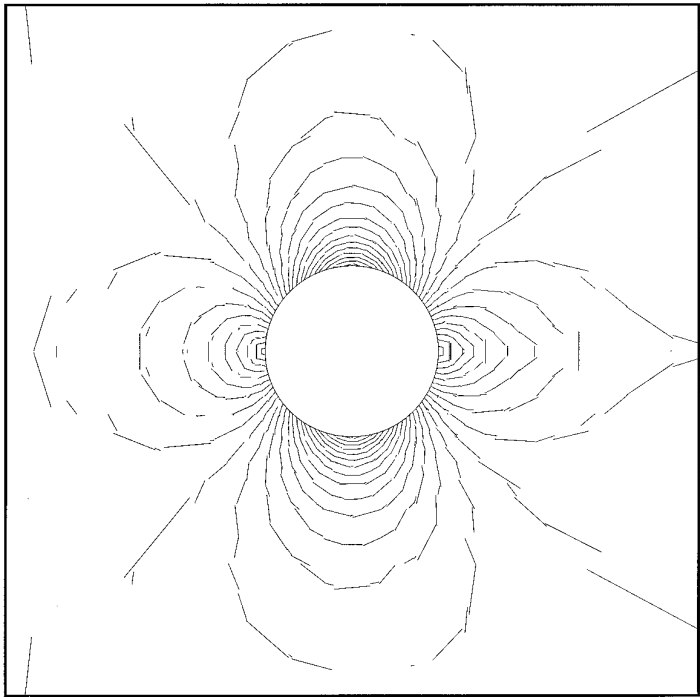


FIG. 10. Mach isolines around a circle with P1Q2 elements on the 32×8 grid.

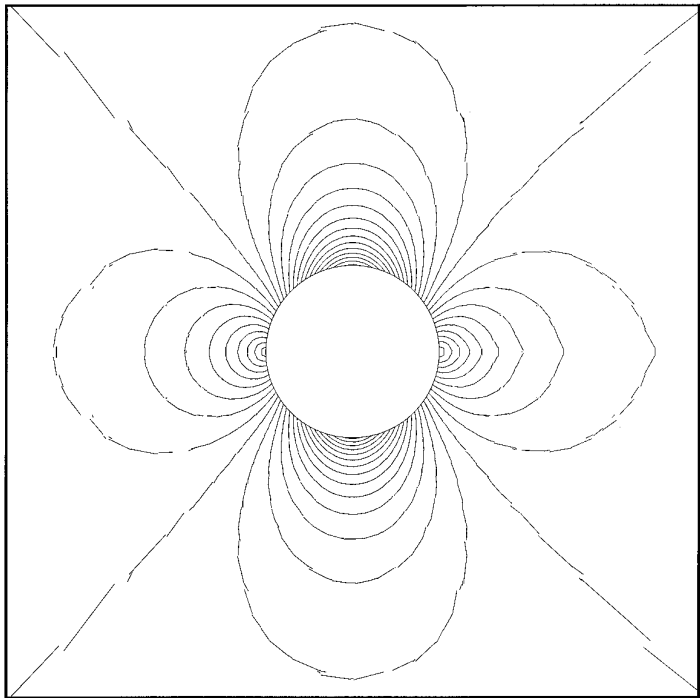


FIG. 11. Mach isolines around a circle with P1Q2 elements on the 64×16 grid.

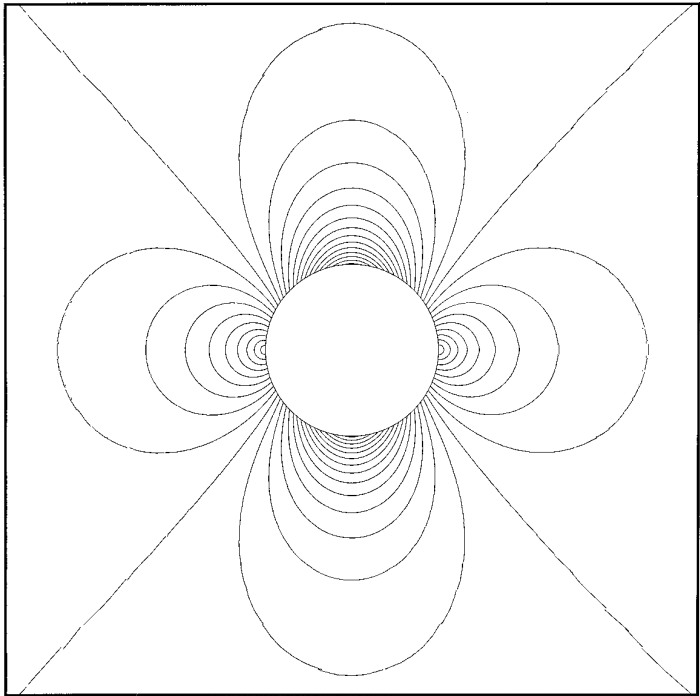


FIG. 12. Mach isolines around a circle with P1Q2 elements on the 128×32 grid.

direction along curved boundaries. However, our computations seem to indicate that a quadratic representation of the boundary is mandatory in order to obtain accurate solutions.

4.1.2. Higher Order Accurate Computations

We next present the result obtained with P2Q2 and P3Q3 isoparametric elements. Because of the inaccuracy already shown by the P1Q1 computations, no attempt has been made to use a linear geometric mapping in conjunction with high-order interpolation functions for the unknowns. The results of additional test calculations

TABLE I
 L_2 Entropy Errors and Orders of Convergence for the P1Q2, P2Q2, and P3Q3 Computations of the Flow around a Circle on Four Successive Grids

	Err_a	Err_b	Err_c	Err_d	Ord_{ab}	Ord_{bc}	Ord_{cd}
P1Q2	0.710×10^{-1}	0.102×10^{-1}	0.153×10^{-2}	0.231×10^{-3}	2.78	2.74	2.73
P2Q2	0.135×10^{-1}	0.941×10^{-3}	0.649×10^{-4}	0.614×10^{-5}	3.84	3.86	3.40
P3Q3	0.818×10^{-2}	0.415×10^{-3}	0.182×10^{-4}		4.30	4.51	

Note. The indices a, b, c , and d indicate the $16 \times 4, 32 \times 8, 64 \times 16$, and 128×32 grids, respectively. $Ord_{\alpha\beta}$, indicates the order of convergence obtained by comparing the solutions of grid α and of grid β and is computed as $Ord_{\alpha\beta} = \log(Err_\alpha/Err_\beta)/\log(h_\alpha/h_\beta)$, h being the mesh size of the grid.

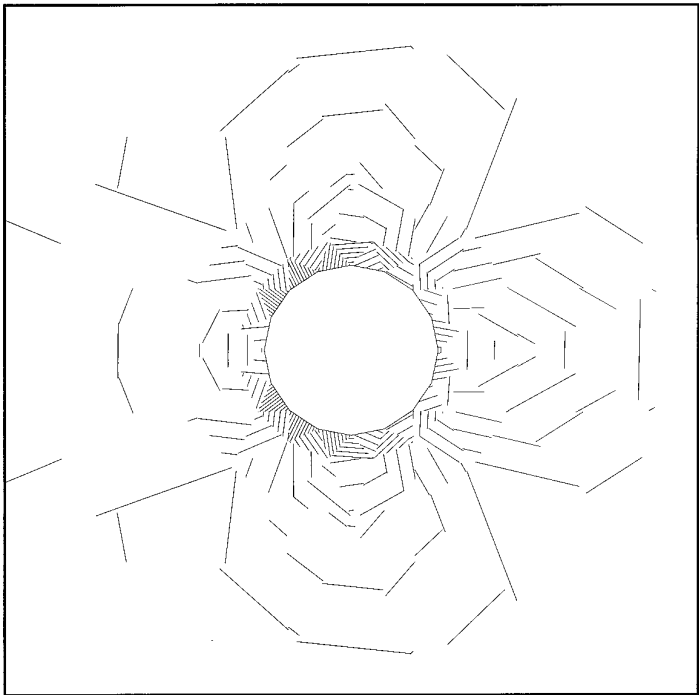


FIG. 13. Mach isolines around a circle with P1Q1* elements on the 16×4 grid.

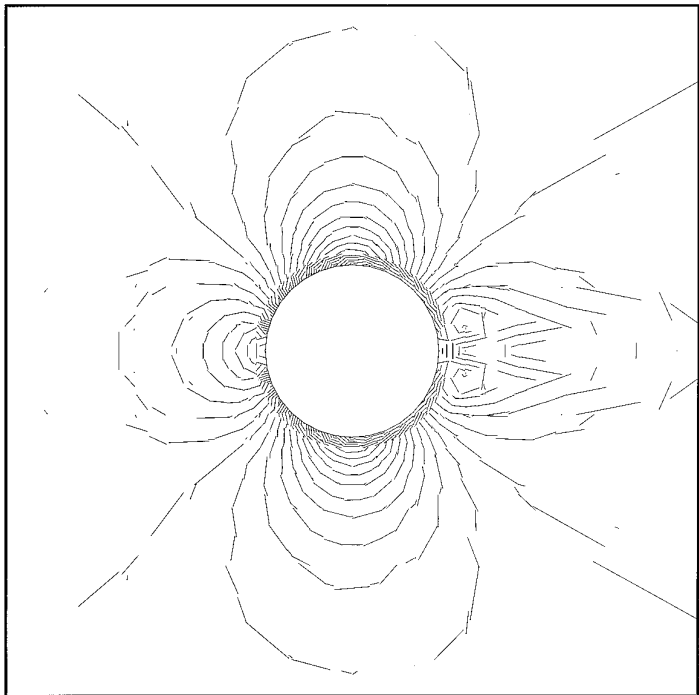


FIG. 14. Mach isolines around a circle with P1Q1* elements on the 32×8 grid.

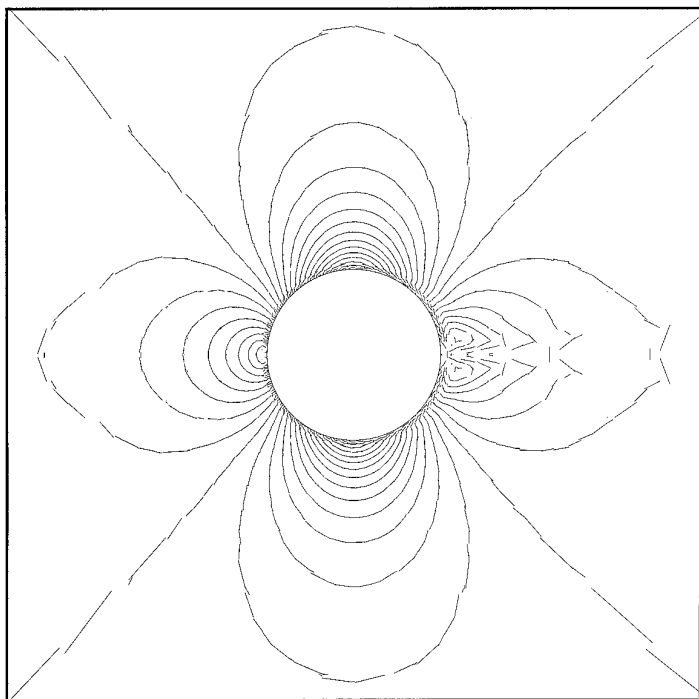


FIG. 15. Mach isolines around a circle with P1Q1* elements on the 64×16 grid.

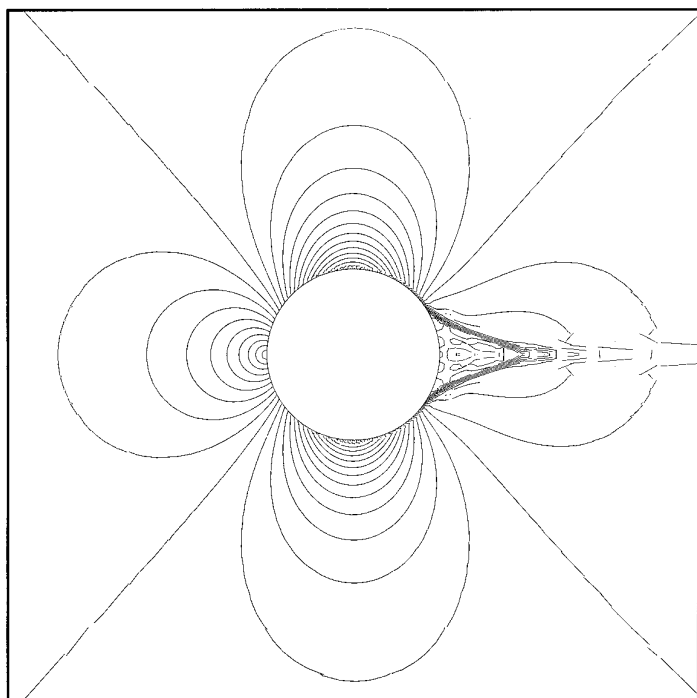


FIG. 16. Mach isolines around a circle with P1Q1* elements on the 128×32 grid.

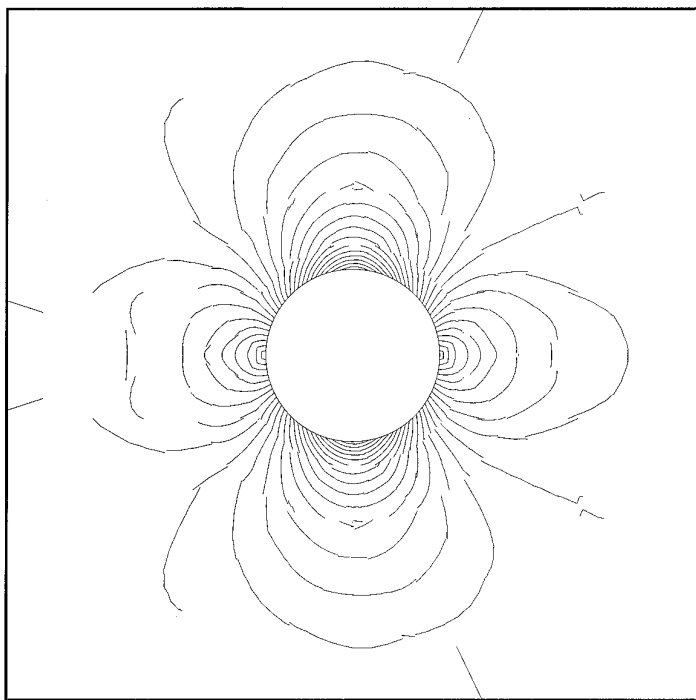


FIG. 17. Mach isolines around a circle with P2Q2 elements on the 16×4 grid.

(not reported for brevity) performed with P2Q3 and P3Q4 superparametric elements did not show any significant difference from those obtained with P2Q2 and P3Q3 elements.

Figures 17–20 show the Mach isolines computed with P2Q2 elements on the four grids, Figs. 21–24 show the Mach isolines computed with P3Q3 elements, and Table I reports the L_2 entropy error and the order of accuracy for the P1Q2, P2Q2, and P3Q3 computations (the result obtained for P1Q1 elements are not reported, since these computations do not converge to a steady-state solution).

These computations allow us to appreciate the potentialities offered by third- and fourth-order accurate elements in the numerical solutions of the Euler equations on unstructured grids. Notice, in fact, the remarkably small entropy error of these numerical solutions even on the very coarse 16×4 grid. Notice also that the order of convergence of these computations is consistent with the theoretical one.

The accuracy of the method is also evidenced in Figs. 25–28, which report the distribution of the pressure coefficient around the circle for the four different grids, and by Figs. 29–32, which report the distribution of the total pressure loss coefficient (defined as the ratio between the total pressure at a point and the total pressure of the undisturbed flow) for the various grids.

4.2. Ringleb Flow

We next present the result obtained for the more difficult Ringleb flow test case, which corresponds to an analytical smooth solution of the Euler equations obtained with the “hodograph method” (see, e.g., [1]). The Ringleb solution represents a

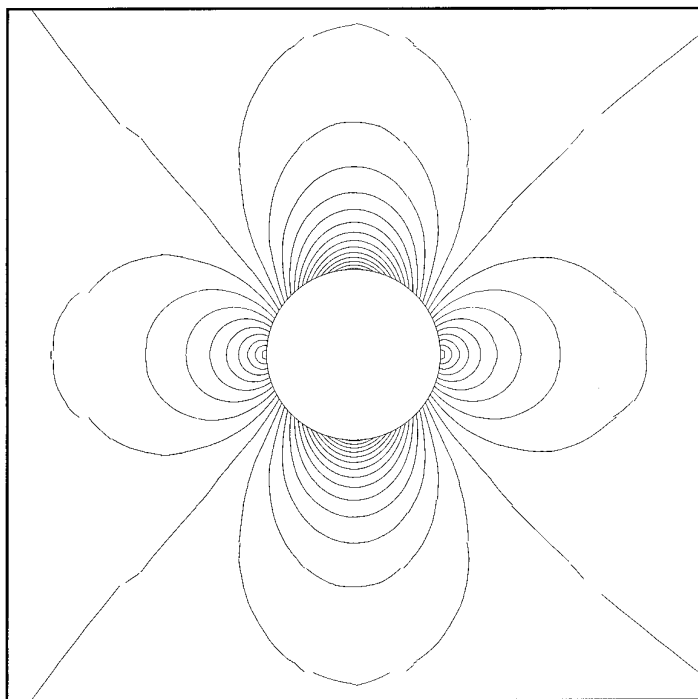


FIG. 18. Mach isolines around a circle with P2Q2 elements on the 32×8 grid.

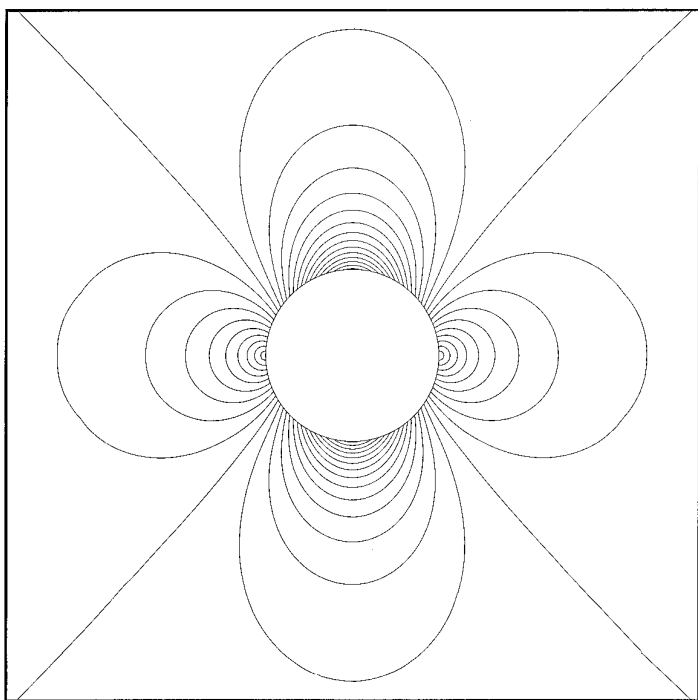


FIG. 19. Mach isolines around a circle with P2Q2 elements on the 64×16 grid.

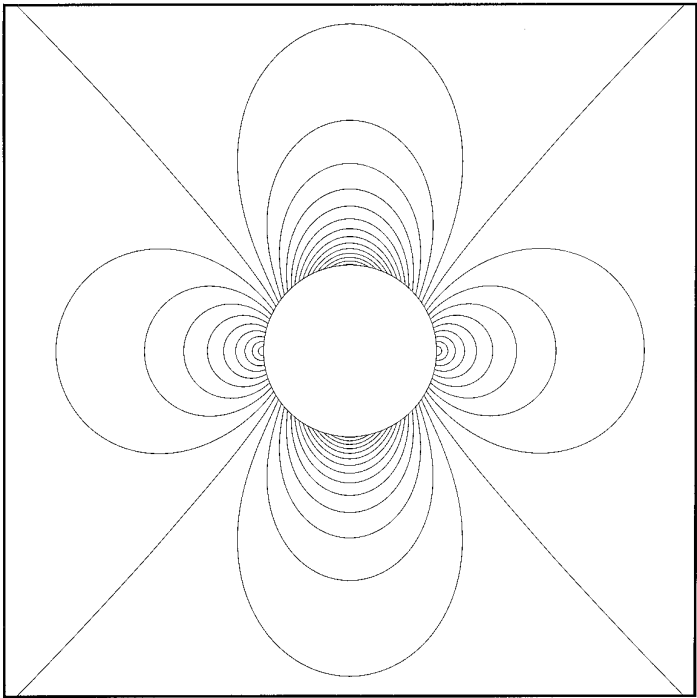


FIG. 20. Mach isolines around a circle with P2Q2 elements on the 128×32 grid.

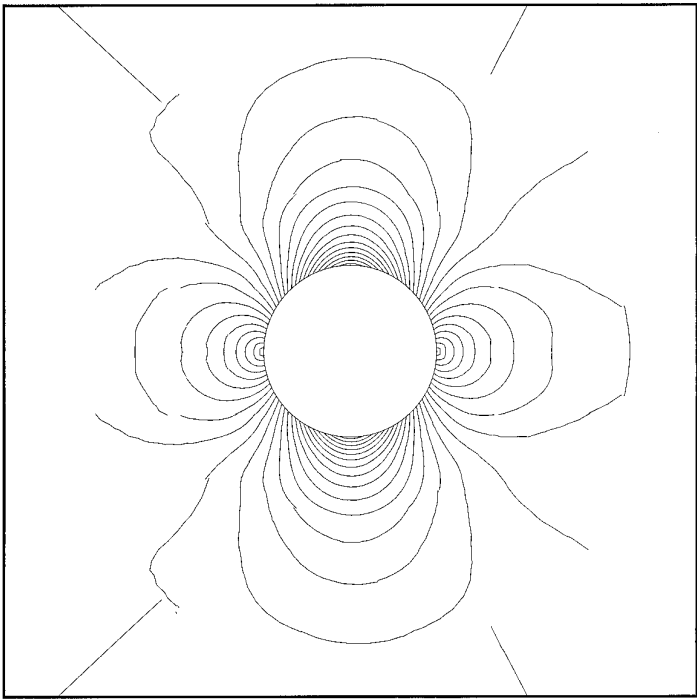


FIG. 21. Mach isolines around a circle with P3Q3 elements on the 16×4 grid.

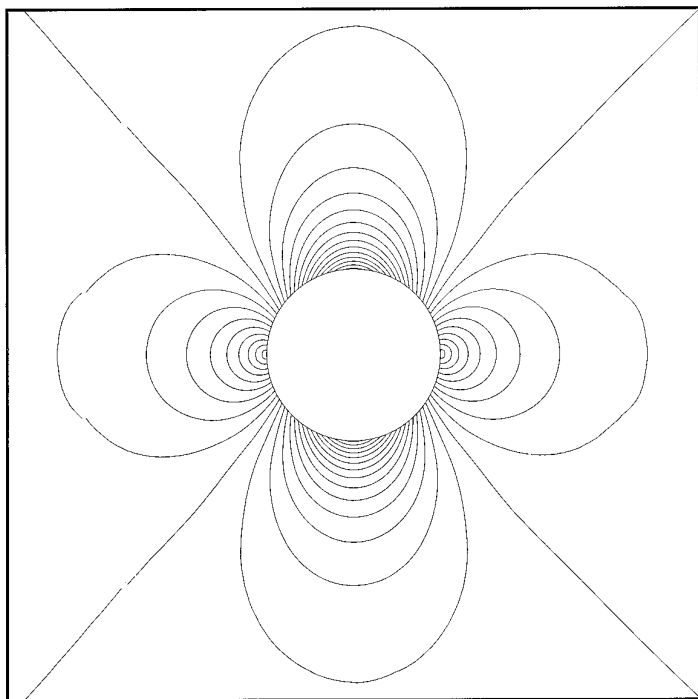


FIG. 22. Mach isolines around a circle with P3Q3 elements on the 32×8 grid.

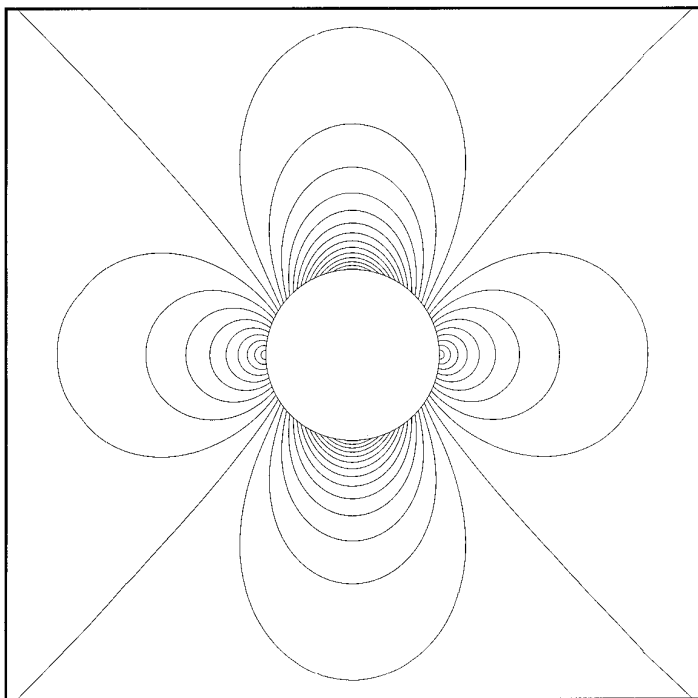


FIG. 23. Mach isolines around a circle with P3Q3 elements on the 64×16 grid.

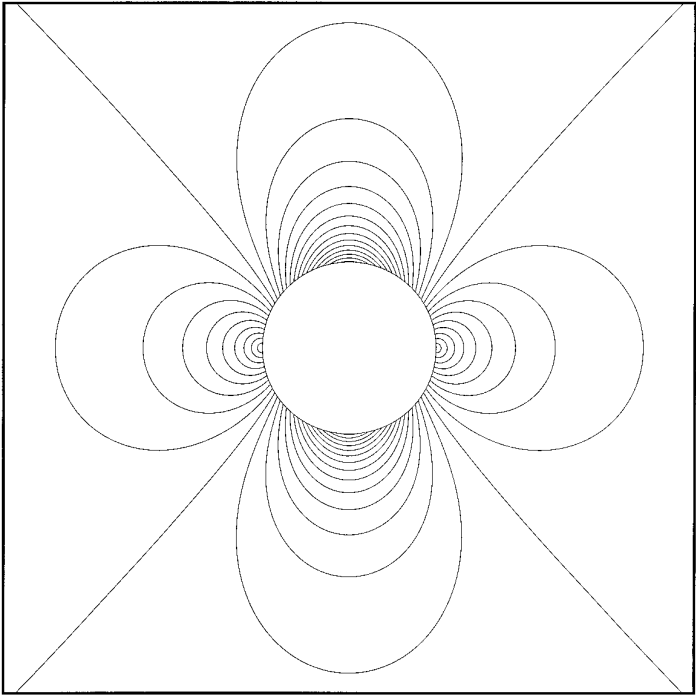


FIG. 24. Mach isolines around a circle with P3Q3 elements on the 128×32 grid.

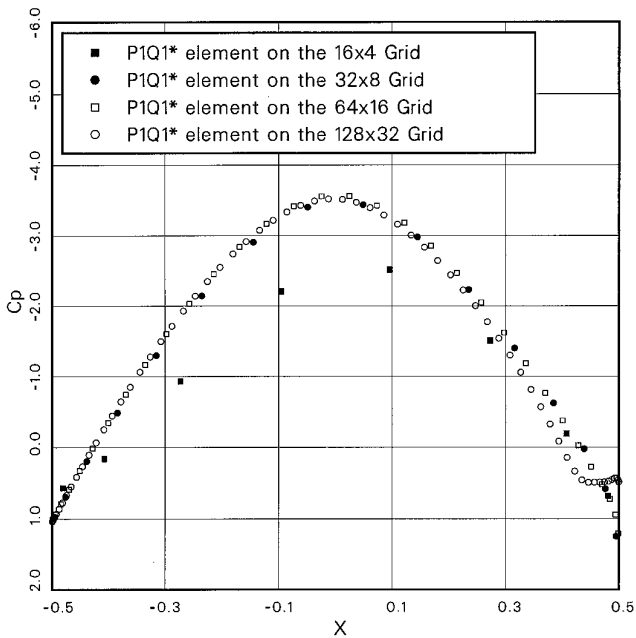


FIG. 25. Pressure coefficient distribution around a circle with P1Q1* elements.

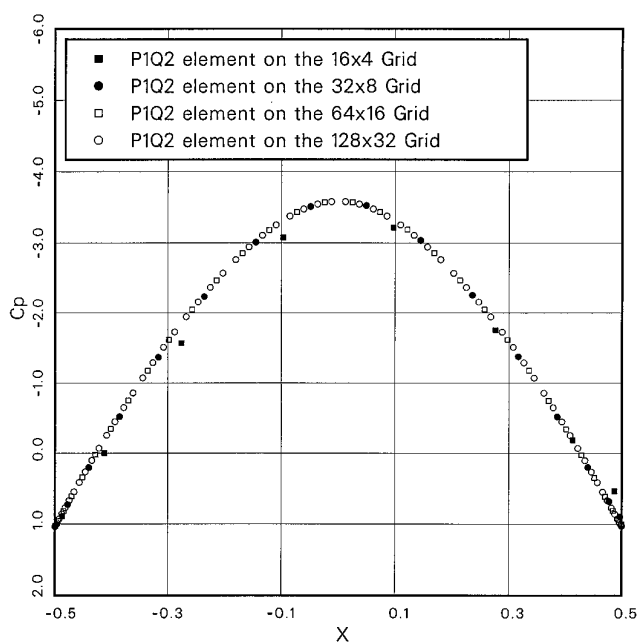


FIG. 26. Pressure coefficient distribution around a circle with P1Q2 elements.

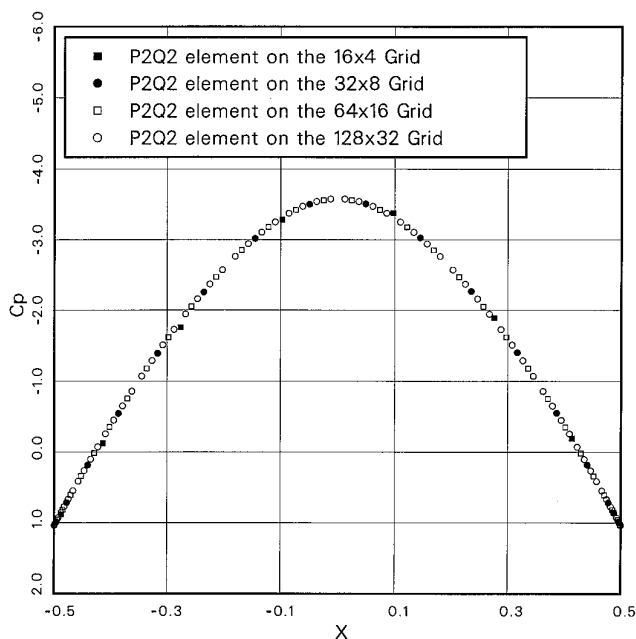


FIG. 27. Pressure coefficient distribution around a circle with P2Q2 elements.

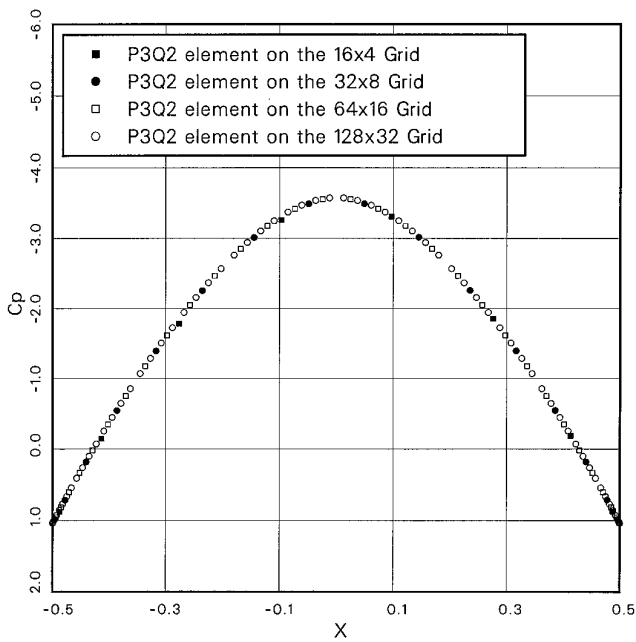


FIG. 28. Pressure coefficient distribution around a circle with P3Q3 elements.

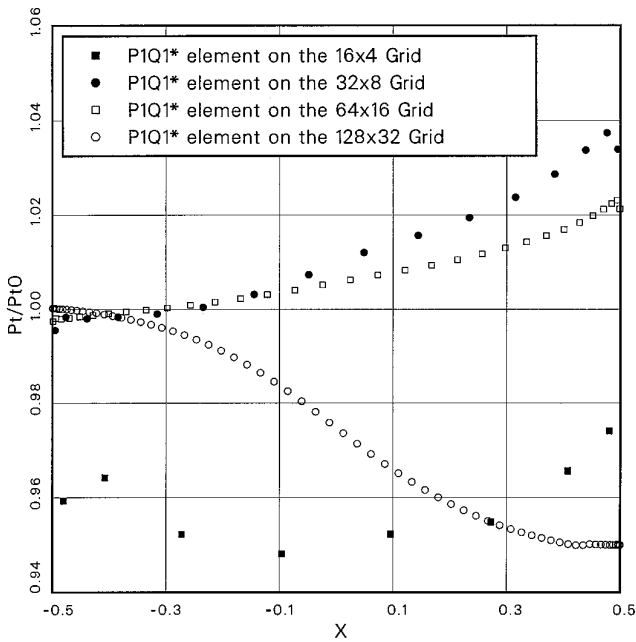


FIG. 29. Total pressure loss coefficient distribution around a circle with P1Q1* elements.

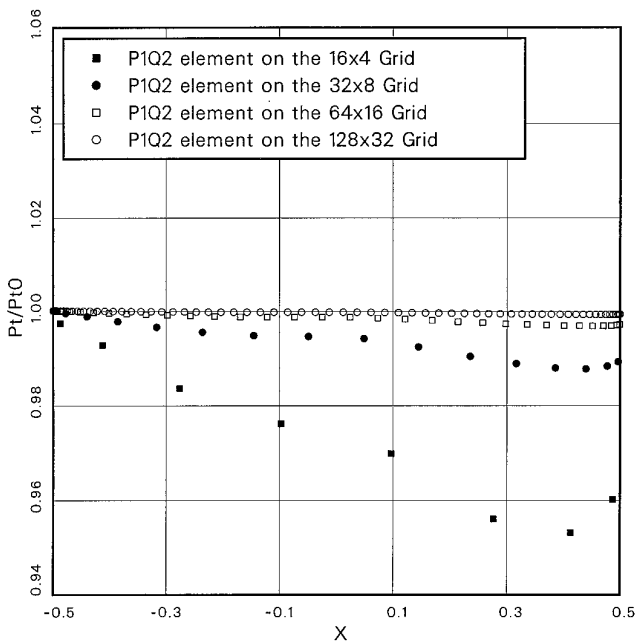


FIG. 30. Total pressure loss coefficient distribution around a circle with P1Q2 elements.

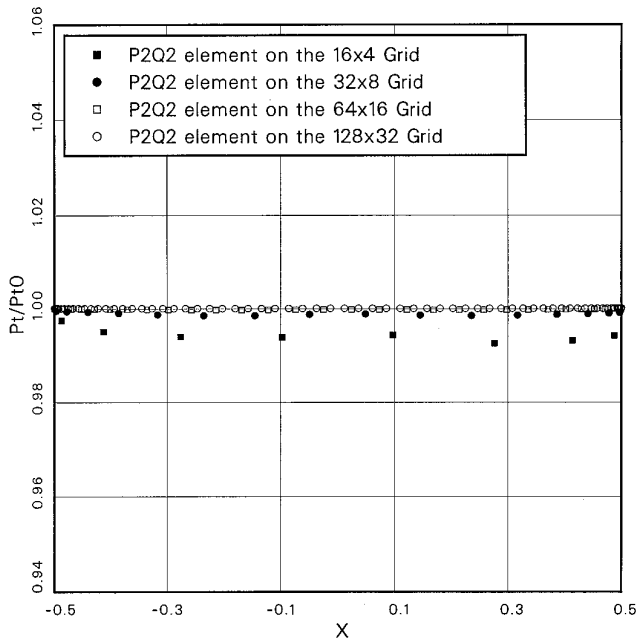


FIG. 31. Total pressure loss coefficient distribution around a circle with P2Q2 elements.

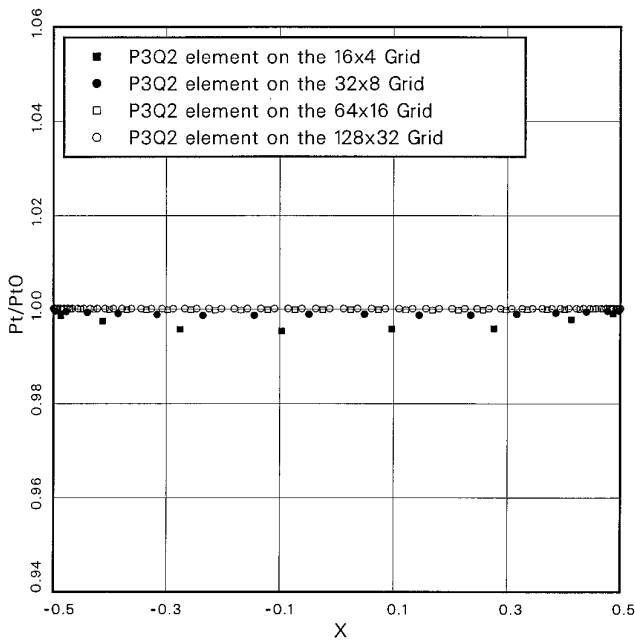


FIG. 32. Total pressure loss coefficient distribution around a circle with P3Q3 elements.

flow which turns around a symmetric obstacle. The flowfield is irrotational and isentropic, and the streamlines are symmetric along the axis of symmetry of the obstacle. Their shape tends to become parabolic far away from the body. The flow field is supersonic in a region around the nose of the obstacle.

Also in this test case we have performed various computations with different types of elements in order to check the convergence properties and the accuracy of the method at the solid wall. The elements are denoted as in the circle computations. We present calculations with linear, quadratic, and cubic approximation of the unknowns and with linear, quadratic, cubic, and fourth-order geometric mapping. The various computations have been performed on the four successively refined Delaunay grids shown in Figs. 33–36, which are the triangulations of sets of 16×4 , 32×8 , 64×16 , and of 128×32 points.

Figures 37 and 38 show the Mach isolines computed with P1Q1 isoparametric elements on the two finer grids, while Figs. 39 and 40 show a detail around the nose of the obstacle of the Mach isolines on the same grids. The isolines are plotted for values of the Mach number given by $M_i = M_0 + i \Delta M$, $i = 0, 1, \dots$, where $M_0 = 0$ and $\Delta M = 0.05$. Similarly to the circle test case, the numerical solutions are spoiled by a large spurious entropy production at solid walls if a linear approximation of the geometry is used. Even the solution obtained on the finest grid is very inaccurate near the solid walls. As shown in Figs. 41–44, however, this inaccuracy is completely cured by considering P1Q2 (or higher order) elements.

The accuracy of the high-order computations is evidenced in Figs. 45–52, showing the Mach isolines computed with P2Q3 and P3Q4 elements, and in Table II, which reports the L_2 entropy error and the order of convergence for various types of element. Similar results are obtained with P2Q2 and P3Q3 isoparametric elements.

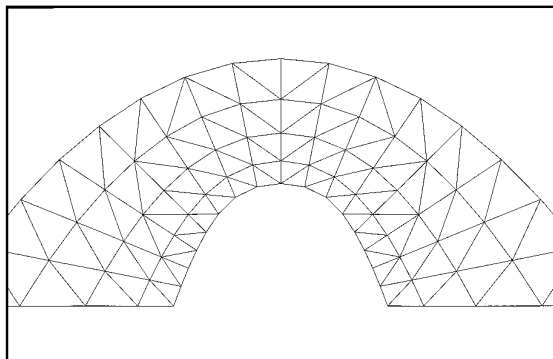


FIG. 33. 16×4 grid for the Ringleb flow.

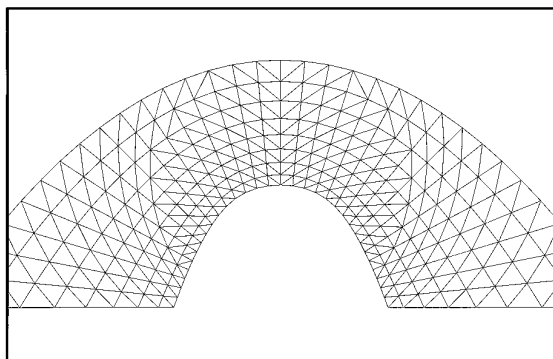


FIG. 34. 32×8 grid for the Ringleb flow.

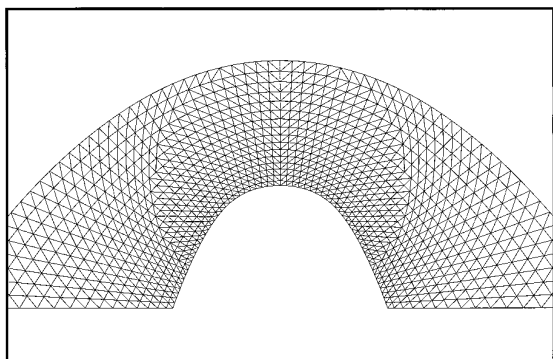


FIG. 35. 64×16 grid for the Ringleb flow.

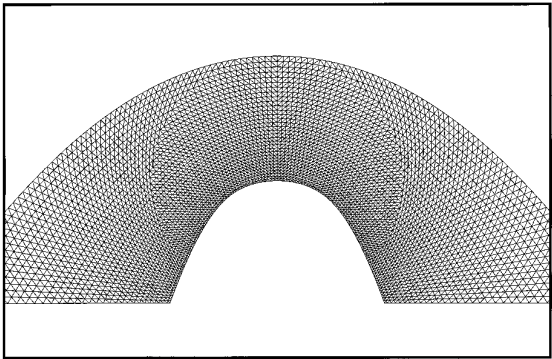


FIG. 36. 128×32 grid for the Ringleb flow.

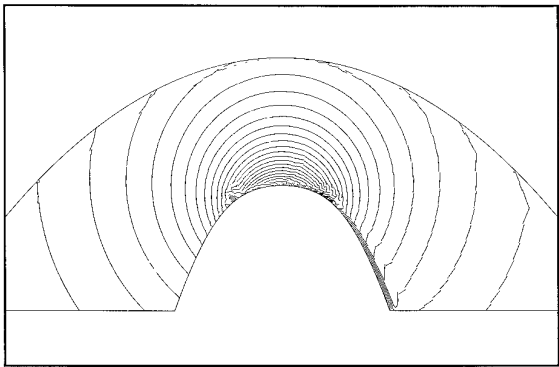


FIG. 37. Ringleb flow Mach isolines with P1Q1 elements on the 64×16 grid.

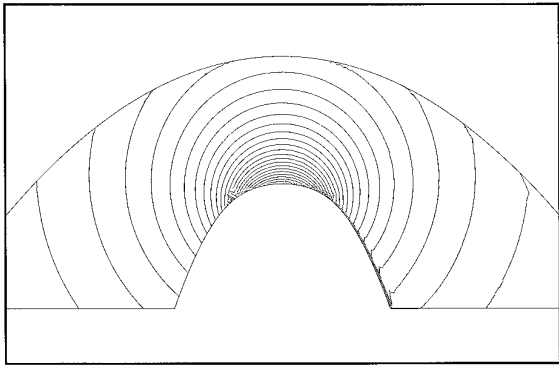


FIG. 38. Ringleb flow Mach isolines with P1Q1 elements on the 128×32 grid.

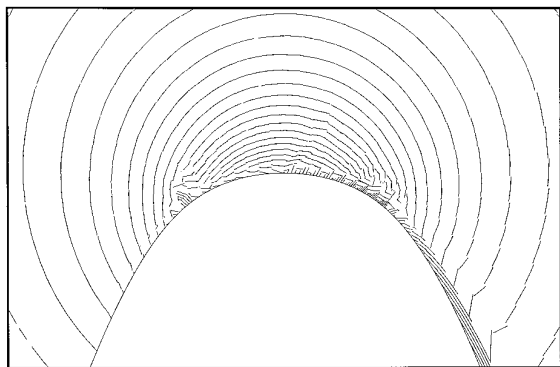


FIG. 39. Detail of the Ringleb flow Mach isolines on the 64×16 grid.

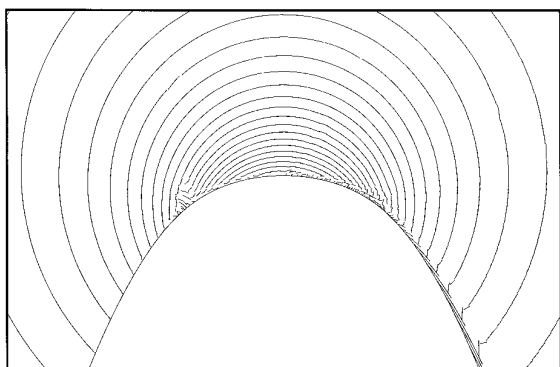


FIG. 40. Detail of the Ringleb flow Mach isolines on the 128×32 grid.

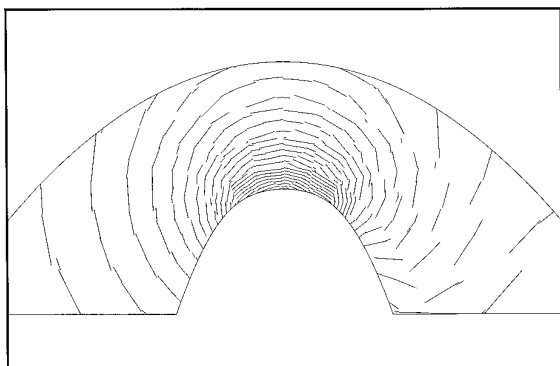


FIG. 41. Ringleb flow Mach isolines with P1Q2 elements on the 16×4 grid.

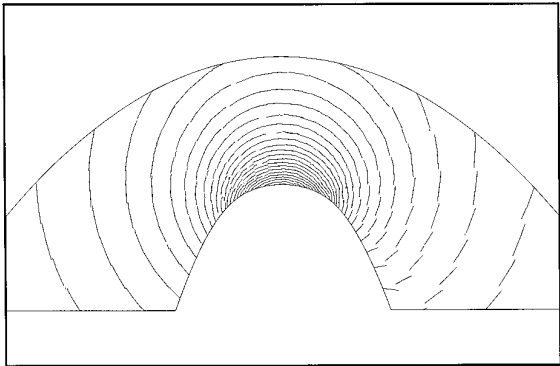


FIG. 42. Ringleb flow Mach isolines with P1Q2 elements on the 32×8 grid.

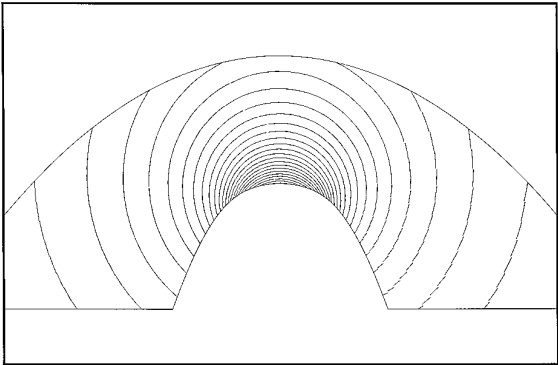


FIG. 43. Ringleb flow Mach isolines with P1Q2 elements on the 64×16 grid.

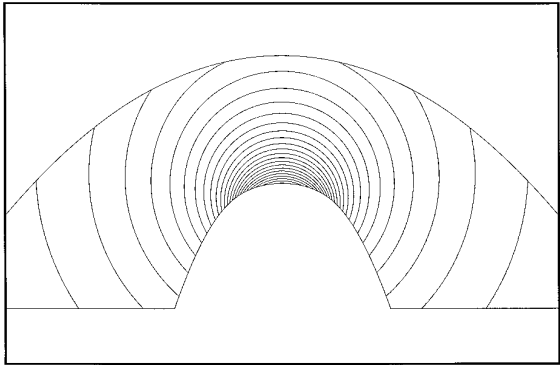


FIG. 44. Ringleb flow Mach isolines with P1Q2 elements on the 128×32 grid.

TABLE II
 L_2 Entropy Errors and Orders of Convergence for the P1Q2, P2Q2, P2Q3, P3Q3, and P3Q4 Computations of the Ringleb Flow on Four Successively Refined Grids

	Err_a	Err_b	Err_c	Err_d	Ord_{ab}	Ord_{bc}	Ord_{cd}
P1Q2	0.355×10^{-1}	0.806×10^{-2}	0.167×10^{-2}	0.395×10^{-3}	2.14	2.27	2.07
P2Q2	0.474×10^{-2}	0.803×10^{-3}	0.107×10^{-3}	0.150×10^{-4}	2.56	2.91	2.83
P2Q3	0.611×10^{-2}	0.105×10^{-3}	0.156×10^{-3}	0.222×10^{-4}	2.54	2.75	2.81
P3Q3	0.156×10^{-2}	0.914×10^{-4}	0.624×10^{-5}		4.09	3.87	
P3Q4	0.154×10^{-2}	0.970×10^{-4}	0.616×10^{-5}	0.438×10^{-6}	3.99	3.98	3.81

Note. The indices a, b, c , and d indicate the $16 \times 4, 32 \times 8, 64 \times 16$, and 128×32 grids, respectively. $\text{Ord}_{\alpha\beta}$, indicates the order of convergence obtained by comparing the solutions of grid α and of grid β and is computed as $\text{Ord}_{\alpha\beta} = \log(\text{Err}_\alpha/\text{Err}_\beta)/\log(h_\alpha/h_\beta)$, h being the mesh size of the grid.

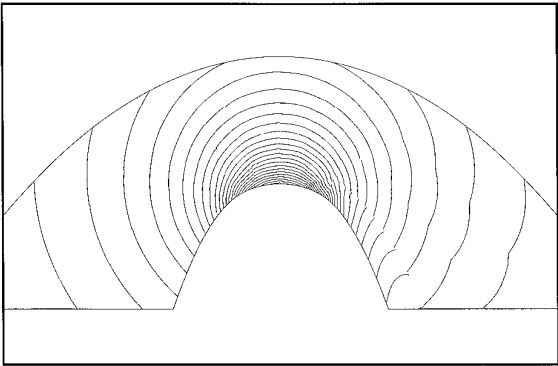


FIG. 45. Ringleb flow Mach isolines with P2Q3 elements on the 16×4 grid.

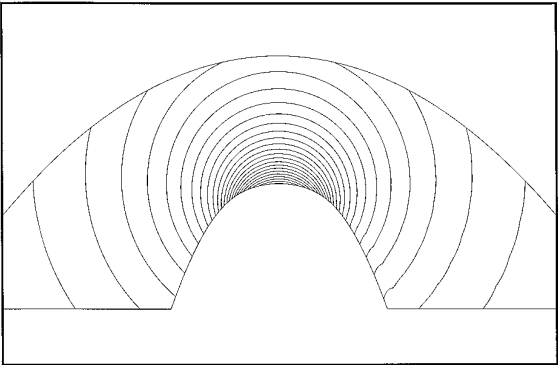


FIG. 46. Ringleb flow Mach isolines with P2Q3 elements on the 32×8 grid.

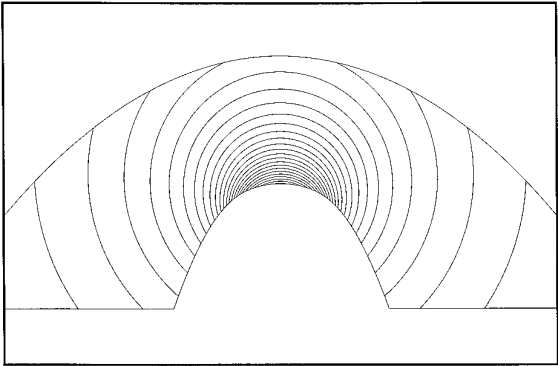


FIG. 47. Ringleb flow Mach isolines with P2Q3 elements on the 64×16 grid.

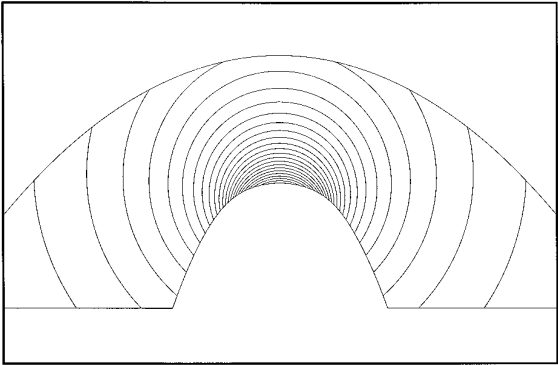


FIG. 48. Ringleb flow Mach isolines with P2Q3 elements on the 128×32 grid.

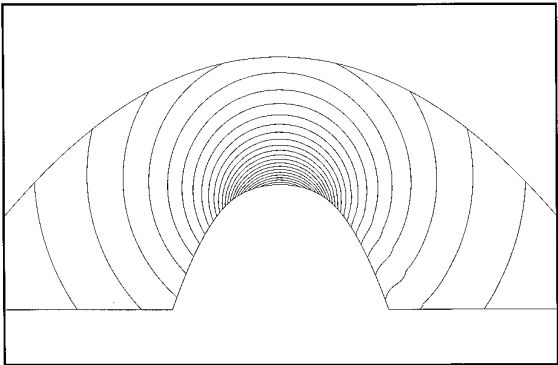


FIG. 49. Ringleb flow Mach isolines with P3Q4 elements on the 16×4 grid.

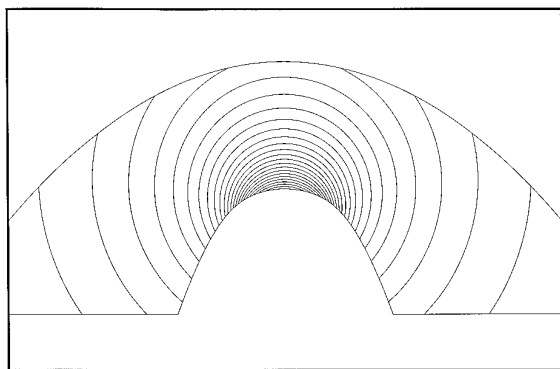


FIG. 50. Ringleb flow Mach isolines with P3Q4 elements on the 32×8 grid.

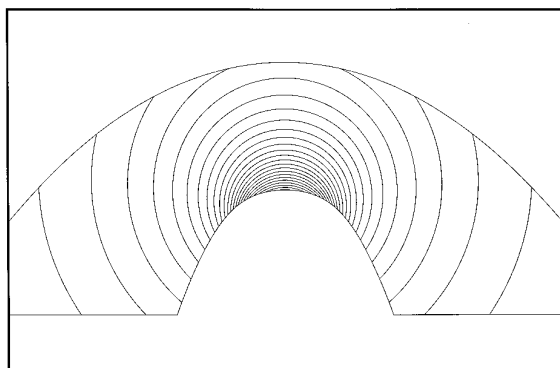


FIG. 51. Ringleb flow Mach isolines with P3Q4 elements on the 64×16 grid.

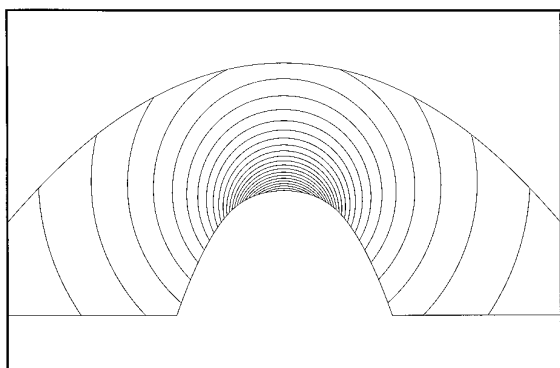


FIG. 52. Ringleb flow Mach isolines with P3Q4 elements on the 128×32 grid.

5. CONCLUSIONS

A high-order accurate discontinuous finite element method for the solution of the Euler equations has been presented. The method combines the ability of all finite element methods of constructing in an automatic and reliable way high-order approximations of differential problems on general unstructured meshes with the flux upwinding technique characterizing a class of finite-volume schemes. The method has proven to be very effective in the solution of classical two-dimensional steady-state test cases for the Euler equations. Our results show that accurate solutions on relatively coarse meshes can be computed by using a high-order representation of the unknowns and of the geometry of the boundary of the domain. In particular, our computations show that a geometric representation which takes into account (at least) the curvature of the boundary (i.e., a quadratic representation) is mandatory in order to obtain meaningful numerical solutions.

Work is in progress to extend the method to treat the full Navier–Stokes equations by resorting to a mixed formulation to discretize the viscous terms. Some preliminary results obtained with this approach have already been published in [6]. We plan to give a detailed report about the extension of the method to viscous flows in a forthcoming paper.

REFERENCES

1. *Test Cases for Inviscid Flow Field Methods*, AGARD-AR-211, 7 Ancelle 92200, Neuilly sur Seine, France, 1985.
2. T. J. Barth and P. O. Frederickson, *Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction*, AIAA-90-0013, 1990.
3. F. Bassi, S. Rebay, and M. Savini, A high resolution discontinuous Galerkin method for hyperbolic problems on unstructured grids, *Numerical Methods in Fluid Dynamics*, edited by M. J. Baines and K. W. Morton, in *III-rd ICFD Conference, Reading, April 1992*, (Clarendon Press, Oxford, 1993), p. 345.
4. F. Bassi, S. Rebay, and M. Savini, Discontinuous finite element Euler solutions on unstructured adaptive grids, *Lecture Notes in Physics, Vol. 414*, edited by M. Napolitano and F. Sabetta, in *XIIIth ICNMF, Rome, July 6–10, 1992*, (Springer-Verlag, New York/Berlin, 1993), p. 245.
5. F. Bassi and S. Rebay, Accurate 2D Euler computations by means of a high order discontinuous finite element method, *Lecture Notes in Physics*, in *XIVth ICNMF, Bangalore, July 11–15, 1994* (Springer-Verlag, New York/Berlin, 1996).
6. F. Bassi and S. Rebay (1995). Discontinuous finite element high order accurate numerical solution of the compressible Navier–Stokes equations, *Numerical Methods in Fluid Dynamics*, edited by M. J. Baines and K. W. Morton in *IVth ICFD Conference, Oxford, April 3–6, 1995*, (Clarendon Press, Oxford, UK).
7. K. S. Bey and J. T. Oden, *A Runge–Kutta Discontinuous Finite Element Method for High Speed Flows*, AIAA Paper 91-1575-CP, 1991, p. 541.
8. R. Biswas, K. D. Devine, and J. E. Flaherty, Parallel, adaptive finite element methods for conservation laws, *Appl. Numer. Math.* **14**, 255 (1994).
9. B. Cockburn and C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, *Math. Comput.* **52**, 411 (1989).
10. B. Cockburn and C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems, *J. Comput. Phys.* **84**, 90 (1989).

11. B. Cockburn, S. Hou, and C.-W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, *Math. Comput.* **54**, 545 (1990).
12. B. Cockburn and C.-W. Shu, “The P^1 –RKDG Method for two-dimensional Euler Equations of Gas Dynamics,” ICASE Report No. 91-32, 1991.
13. A. Dadone and B. Grossman (1994). “Surface Boundary Conditions for the Numerical Solution of the Euler Equations,” *AIAA Journal*, Vol. 32, No. 2, 285–293.
14. S. K. Godunov, “A Difference Scheme for Numerical Computation of Discontinuous Solutions of Hydrodynamic Equations” *Math Sb.* **47**, 271 (1959). [Russian]. [Translated by U.S. Joint Publ. Res. Service, JPRS, 7226 (1969).
15. A. Harten and S. R. Chakravarthy, “Multi-Dimensional ENO Schemes for General Geometries,” ICASE Report No. 91-76, 1991.
16. P. Lesaint, Sur la résolution des systèmes hyperboliques du premier ordre par des méthodes d’éléments finis, Ph.D. thesis, Université Paris 6, 1975.
17. P. Lesaint and P. A. Raviart, On a finite element method to solve the neutron transport equation, in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. de Boor, (Academic Press, New York, 1974), p. 89.