AIAA JOURNAL
Vol. 34, No. 2, February 1996

Characteristic-Based Pressure Correction at All Speeds

M. H. Kobayashi* and J. C. F. Pereira†
Instituto Superior Técnico, 1096 Lisbon Codex, Portugal

A new characteristic-based pressure-correction method designed for computation of flows at all speeds is presented. The method includes the implementation within the pressure-correction framework of interpolation of the characteristic variables, the approximation of the nonlinear operator by a stable segregate linear operator, the use of an approximate Riemann solver for the computation of the cell face inviscid flux, and the projection of the correction fields into the flux space. Application test cases include 1) supersonic inviscid arc bump flow, 2) transonic viscous double throat nozzle, and 3) incompressible lid driven cavity flow. The method is uniformly of high order and nonoscillatory, which leads to crisp shock representation without the creation of spurious oscillations. The results show a substantial improvement over existing pressure-correction methods for computation of compressible flows.

Nomenclature

A	= coefficients matrix
a	= coefficients in the discrete equations
b	= nonhomogeneous term
c	= sound velocity; numerator of the discrete counterpart of G; curvature term
D	= computational domain
e	= eigenvector of $\partial_Q N^1$
F	= average mass flux
f	= forcing term
G	= transpose of cofactor matrix of inverse transformation matrix
H	= total flux of ϕ; total enthalpy
h	= enthalpy
J	= determinant of X^i_α
k	= thermal conductivity; kinetic energy
L	= linear, nonlinear operator due to Navier–Stokes discretization
M	= molecular weight of the gas; minmod function
Ma	= Mach number
N	= inviscid flux
n	= space dimension
p	= static pressure
Q	= state vector
q	= heat flux; upwind indicator
R	= reconstruction polynomial
$\bar{R}$	= universal gas constant
S_m	= pseudomass source term
s	= linearized source term
T	= static temperature
v	= velocity vector
X	= direct, $X^\alpha_i = \partial_i x^\alpha$, and inverse, X^i_α: $X^i_\alpha X^\alpha_j = \delta^i_j$, transformation matrix
x	= oblique curvilinear coordinate system
y	= Cartesian coordinate system
α	= strength, slope of characteristic variable variation; underrelaxation factor
β	= limited slope of characteristic variable
Δ	= variation
δ	= Kronecker delta; centered difference operator
δ Vol	= volume
$\partial_i, \partial_\alpha$	= $\partial/\partial y^i$, $\partial/\partial x^\alpha$
$\dot{\vartheta}$	= velocity divergence
κ	= coefficient of diffusivity of ϕ
λ	= speed of the characteristic variable
μ	= dynamic viscosity
π	= diffusive flux of ϕ
ρ	= mass density
σ	= stress tensor
τ	= limited slope of characteristic variable, $[x^1_w, x^1_e] \times [x^2_s, x^2_n]$
ϕ	= general transported scalar
χ	= characteristic function
Ψ	= discrete convection diffusion of momentum operator
Ω	= oriented-area operator
$\langle\cdot\rangle$	= arithmetic mean

Subscripts

ap	= approximate
e, w, n, s	= cell faces
i, j, k	= Cartesian components
l, r	= left, right
m	= mass
τ	= control volume
ϕ	= transported scalar

Superscripts

e, w, n, s, ww	= cell face
i, j, k	= Cartesian component
n	= nth outer iteration
$P, E, W, N,$ S, WW, EE	= cell center
T	= transpose of a matrix
α, γ	= oblique contravariant component
$+, -$	= right and left limit values, respectively
$\sim$	= Roe-averaged values
$'$	= correction values
$*$	= predicted value

Introduction

THE numerical solution of the viscous transonic and supersonic flows is usually carried out by time-marching schemes that solve the set of the coupled system of equations governing the flux of mass, momentum, and energy, using accurate high-resolution total variation diminishing[1] (TVD) schemes employing Roe's Riemann solver.[2] On the other hand, simulation of incompressible fluid flows with engineering interest are usually pursued with finite volume formulation, using primitive variables in conjunction with some variant of the semi-implicit pressure-correction method.[3] In this method the momentum equations are solved in a segregated fashion while an equation for the pressure field or pressure-correction field is derived

Received July 23, 1994; revision received July 18, 1995; accepted for publication July 24, 1995.

*Assistant Professor, Department of Mechanical Engineering.
†Full Professor, Department of Mechanical Engineering. Member AIAA.

combining the discrete momentum and continuity equations so that the pressure field is driven towards a level where the continuity equation over each control volume is satisfied to any prescribed level.[3–5] Because this procedure in its standard form results in an elliptic equation for pressure (or pressure correction) it cannot cope with the hyperbolic nature of the signal propagation in compressible transonic or supersonic flows. By contrast those methods that are very efficient for computation of compressible hyperbolic flows become increasingly ill conditioned as the Mach number decreases. Although some remedy for convergence stagnation exists such as artificial compressibility or preconditioning, in practice, these techniques are not well suited for computation of flows with extensive regions of low Mach numbers (lower than 0.3).

Several attempts have been made by incompressible fluid flow numerical researchers, towards the unification of numerical methods developed for incompressible and compressible flows.[6–11] The main goal consists in the development of methods for computation of viscous flows at all Mach numbers by extending the pressure-correction formulation to ensure shock-capturing properties. Yet, as just mentioned, to fully attain these goals, some puzzle in the formalism of these methods has to be settled, for example, accounting for the spreading of acoustic waves and the Riemann problems in the framework of the pressure correction. No previous work gives a satisfactory answer to these questions apart from one-dimensional flow.[10,11] Generally these problems translate themselves into insufficient shock-capturing capabilities and have so far prevented a wider use of pressure-correction methods in the framework of all speed calculations.

In the present work a first positive answer to these problems is provided by introducing a new methodology—called SIMPLENO for simple essentially nonoscillatory. This methodology incorporates into a pressure-correction framework the ENO[12] technique for the reconstruction of the characteristic variables, together with the use of a Riemann solver for computation of the fluxes at the interfaces.

In the paper the general discretization procedure of the scalar convection-diffusion equation is presented as well as a detailed description of the SIMPLENO method. The paper ends with the simulation of flows in different regimes for assessment of the general features of the new algorithm, viz., shock resolution, shock/viscous layer interaction, and vortices' interactions in incompressible flows.

Mathematical and Numerical Model

The steady continuum flow of a nonreacting thermally conductive Newtonian fluid is considered. The mathematical models expressing transport of mass, momentum, and energy are summarized in the Navier–Stokes equations, which in a curvilinear oblique coordinate system and for the Cartesian components of vectors and tensors are

1) Continuity:

$$\partial_\alpha J\rho v^i X_i^\alpha = 0 \tag{1}$$

2) Motion:

$$\partial_\alpha J(\rho v^j v^i + \sigma^{ij}) X_j^\alpha = 0 \tag{2}$$

3) Energy:

$$\partial_\alpha J(\rho v^\alpha h + q^\alpha) = -\partial_\alpha J\sigma^{ki} v^j \delta_{ji} X_k^\alpha \tag{3}$$

where all of the vector and tensor component indices fore and aft run from one to the dimension of the space. In the present work we use the summation convention whereby summation is implied when an index is repeated for diagonal positions.

For a Newtonian fluid the stress tensor σ can be written as

$$\sigma^{ij} = \left(p + \tfrac{2}{3}\mu\dot{\vartheta}\right)\delta^{ij} - \mu\left(X_k^\alpha \partial_\alpha v^i \delta^{kj} + X_k^\alpha \partial_\alpha v^j \delta^{ki}\right) \tag{4}$$

The heat flux vector q is defined by a Fourier law, i.e.,

$$q^i = -kX_k^\alpha \partial_\alpha T \delta^{ki} \tag{5}$$

Equations of continuity, motion, and energy can each be cast into the following general transport equation that is useful for presentation of the discretization procedure:

$$\partial_\alpha(\rho v^i \phi + \pi^i) G_i^\alpha = J\rho s_\phi \tag{6}$$

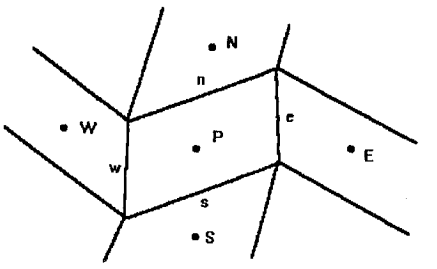


Fig. 1 Control volume configuration and nomenclature.

where π is defined as

$$\pi^i = -\kappa\left(G_k^\alpha / J\right)\partial_\alpha \phi \delta^{ki} \tag{7}$$

To close the governing equations we need an equation of state. In the present work we consider the perfect gas relation

$$p = \rho RT \tag{8}$$

with R defined by

$$R = \bar{R}/M \tag{9}$$

Discretization of General Transport Equation

The discrete form of Eq. (6) is obtained by using the finite volume method based on a collocated, oblique grid (Fig. 1). The main steps for its derivation are briefly discussed next. The reader not familiar with the method can find extensive material on the subject in Refs. 13–16. For the sake of conciseness we consider a two-dimensional geometry.

The finite volume method belongs to the class of weighted residual methods with the characteristic function χ for each control volume (CV) taken as the weight function, i.e.,

$$\int_D \partial_\alpha\left[(\rho v^i \phi + \pi^i) G_i^\alpha\right]\chi_\tau = \int_D J\rho s_\phi \chi_\tau \tag{10}$$

Use of the divergence theorem in Eq. (10) gives rise to the following flux balance equation over each control volume:

$$H^{1e} - H^{1w} + H^{2n} - H^{2s} = s \tag{11}$$

The total flux H of ϕ consists of the convective flux, denoted by H_C, and the diffusive flux, denoted by H_D. In what follows the details of the derivation will be given for the e face only. The evaluation of flux through other cell faces is performed similarly.

The diffusion term is approximated by central differences yielding the following equation:

$$-H_D^{1e} = \int_{\partial\tau_e} \frac{\kappa}{J}\left(G_i^\gamma \delta^{ij} G_j^1 \partial_\gamma \phi\right) \mathrm{d}x^2$$

$$\cong \left(\frac{\kappa}{\delta \mathrm{Vol}}\right)_e D_e^{11}(\phi_E - \phi_P) + \left(\frac{\kappa}{\delta \mathrm{Vol}}\right)_e D_e^{12}(\phi_n - \phi_s)_e \tag{12}$$

where D is given by

$$D^{\alpha\gamma} = c_k^\gamma \delta^{ki} c_i^\alpha \tag{13}$$

and c_i^α is the numerator of the discrete counterpart (using central differences) of cofactor G_i^α, i.e.,

$$\left[c_i^\alpha\right]_e = \begin{bmatrix} \left(y_n^2 - y_s^2\right)_e & -\left(y_n^1 - y_s^1\right)_e \\ -\left(y_E^2 - y_P^2\right) & \left(y_E^1 - y_P^1\right) \end{bmatrix}$$

The convective flux may be written as

$$H_C^{1e} = \int_{\partial\tau_e} \rho v^i G_i^1 \,\mathrm{d}x^2 \cong F_e^1 \phi_e \tag{14}$$

where F_e^1 is the average mass flux passing through face e. In symbols,

$$F_e^1 = \left[\rho\left(v^1 c_1^1 + v^2 c_2^1\right)\right]_e \tag{15}$$

Computation of the convective flux of ϕ is performed by a nonoscillatory uniformly high-order reconstruction of the variable within each CV followed by solution of a Riemann problem for flux evaluation. So, given the average values of ϕ at the center of each CV, the problem at hand is to reconstruct closely at each point the distribution of the variable inside each CV by a piecewise polynomial (to high accuracy in any neighborhood where the function is smooth) and then evaluate the flux at the interface by solving a Riemann problem with right and left values given by their respective limits at the interface.

Clearly any multidimensional high-order scheme can be incorporated into the procedure. For the sake of completeness we consider the uniformly second-order-accurate nonoscillatory UNO2 scheme of Harten.[17] In this case, given the averaged values at control volume centers the reconstruction polynomial R takes the form

$$R(x^1, x^2; \phi) = \phi_P + r_P\left(x^1 - x_P^1\right) + s_P\left(x^2 - x_p^2\right) + d_P\left(x^1 - x_P^1\right)\left(x^2 - x_P^2\right), \qquad \forall (x^1, x^2) \in \tau \tag{16}$$

Because R is bilinear within τ, the average of R over cell face e is identical to its point value at (x_{e-}^1, x_P^2)—Fig. 1. Consequently, reconstruction reduces to calculating the coefficients r and s, and since evaluation of the variable at interfaces does not involve any cross term, we drop the indices and treat each direction separately; for instance, r_P is computed as

$$r_P = M\left(t_P^+, t_P^-\right) \tag{17}$$

where M stands for the minmod function

$$(a, b) \in \boldsymbol{R}^2 : M(a, b) = \begin{cases} \min(|a|, |b|), & \text{if } ab > 0 \\ 0, & \text{if } ab \le 0 \end{cases} \tag{18}$$

and $t_P^{\pm}$ are defined by

$$\begin{aligned} t_P^- &= \phi\left[x_W^1, x_P^1\right] + c_w \delta_w x^1 \\ t_P^+ &= \phi\left[x_P^1, x_E^1\right] - c_e \delta_e x^1 \end{aligned} \tag{19}$$

where the curvature terms are computed as

$$\begin{aligned} c_w &= M\left[\phi\left(x_{WW}^1, x_W^1, x_P^1\right), \phi\left(x_W^1, x_P^1, x_E^1\right)\right] \\ c_e &= M\left[\phi\left(x_W^1, x_P^1, x_E^1\right), \phi\left(x_P^1, x_E^1, x_{EE}^1\right)\right] \end{aligned} \tag{20}$$

where $\phi[(x_a^1)_{a \in A}]$ is the divided difference of ϕ over support A, and

$$\delta_e x^1 = x_E^1 - x_P^1 \tag{21}$$

Other terms and slopes are computed similarly. The interface value of ϕ is obtained by solving a Riemann problem with corresponding interface limit values, i.e.,

$$\phi_e = \mathrm{Rie}\left(\phi_e^-, \phi_e^+\right) \tag{22}$$

where Rie is the solution at the interface of the Riemann problem with left and right values given by ϕ_e^- and ϕ_e^+, respectively.

Substitution of Eq. (22) into Eq. (14) and the latter with Eq. (12) into Eq. (11) gives rise to a nonlinear system of equations for the nodal values of ϕ. In the case of the Navier–Stokes equations the resulting system of equations is nonlinear and highly coupled. Integration and linearization of the Navier–Stokes equations are carried out in next section where the SIMPLENO method is introduced.

SIMPLENO Method

The discretization procedure developed in the previous section applies to a scalar transport equation only. Its extension to the Navier–Stokes equations is not a trivial task. It must accommodate into the numerical scheme the complex physics of signal propagation that takes place within the flow at different Mach number regimes. To attain this goal we incorporate for the first time into a semi-implicit pressure-correction method the interpolation of characteristic variables and the solution of a Riemann problem for computation of the inviscid flux. The resulting general methodology is called SIMPLENO.

The characteristic variables and Roe's approximate Riemann solver[2] are used for the computation of the inviscid flux, whereas for the diffusive flux we use the second-order-accurate practice explained earlier. Denote by Q and N the state vector and its inviscid flux, respectively, i.e.,

$$\begin{aligned} Q &= (\rho, \rho v^1, \rho v^2, \rho E) \\ N^1(w) &= \left(\rho U^1, \rho U^1 v^1 + p G_1^1, \rho U^1 v^2 + p G_2^1, \rho U^1 H\right) \\ N^2(w) &= \left(\rho U^2, \rho U^2 v^1 + p G_1^2, \rho U^2 v^2 + p G_2^2, \rho U^2 H\right) \end{aligned} \tag{23}$$

where U is defined as

$$U^\alpha = v^i G_i^\alpha \tag{24}$$

Then by using Roe's approximate Riemann solver the averaged inviscid flux that crosses face e is evaluated as follows:

$$N_e^1(Q_l, Q_r) = \frac{1}{2}\left(N_l^1 + N_r^1\right) - \frac{1}{2}\sum_k \alpha_k |\lambda_k| e_k \tag{25}$$

where

$$\alpha_1 = -\tilde{\rho}\Delta u^1 + (\Delta p/\tilde{c}), \qquad \alpha_2 = -\tilde{\rho}\Delta w^1 - (\Delta p/\tilde{c}) + \tilde{c}\Delta\rho$$
$$\alpha_3 = \tilde{\rho}\Delta w^1 - (\Delta p/\tilde{c}) + \tilde{c}\Delta\rho, \qquad \alpha_4 = \tilde{\rho}\Delta u^1 + (\Delta p/\tilde{c}) \tag{26}$$

$$\lambda_1 = (\tilde{u}^1 - \tilde{c})l, \qquad \lambda_2 = \lambda_3 = \tilde{u}^1 l, \qquad \lambda_4 = (\tilde{u}^1 + \tilde{c})l \tag{27}$$

and e the eigenvector of the Jacobian $\partial_Q N^1$, i.e.,

$$\begin{aligned} e_1 &= (1/2\tilde{c})\left(1, \tilde{v}^1 - \tilde{c}n_1, \tilde{v}^2 - \tilde{c}n_2, \tilde{H} - \tilde{c}\tilde{u}^1\right)^T \\ e_2 &= (1/2\tilde{c})\left(1, \tilde{v}^1 - \tilde{c}n_2, \tilde{v}^2 + \tilde{c}n_1, \tilde{c}\tilde{w}^1 + \tilde{K}\right)^T \\ e_3 &= (1/2\tilde{c})\left(1, \tilde{v}^1 + \tilde{c}n_2, \tilde{v}^2 - \tilde{c}n_1, \tilde{c}\tilde{w}^1 - \tilde{K}\right)^T \\ e_4 &= (1/2\tilde{c})\left(1, \tilde{v}^1 + \tilde{c}n_1, \tilde{v}^2 + \tilde{c}n_2, \tilde{H} + \tilde{c}\tilde{u}^1\right)^T \end{aligned} \tag{28}$$

where

$$\tilde{\rho}^2 = \rho_l \rho_r, \qquad \tilde{v}^i = \frac{\left\langle \rho^{\frac{1}{2}} v^i \right\rangle}{\left\langle \rho^{\frac{1}{2}} \right\rangle}, \qquad \tilde{H} = \frac{\left\langle \rho^{\frac{1}{2}} H \right\rangle}{\left\langle \rho^{\frac{1}{2}} \right\rangle}$$
$$\tilde{c}^2 = (\gamma - 1)(\tilde{H} - \tilde{K}), \qquad \tilde{K} = \tfrac{1}{2}[(\tilde{v}^1)^2 + (\tilde{v}^2)^2]$$
$$l^2 = \left(c_1^1\right)^2 + \left(c_2^1\right)^2 \tag{29}$$
$$n_i = c_i^1 / l, \qquad \tilde{u}^1 = \tilde{v}^i n_i, \qquad \tilde{w}^1 = \tilde{v}^2 n_1 - \tilde{v}^1 n_2$$

For simplicity the same letter N is used for both the vector flux of Q and its averaged value. The left and right states are obtained by reconstruction of the characteristic variables, e.g., for the left state it follows that

$$Q_l = R\left(x_{e-}^1; Q\right) = Q_P + \frac{1}{2}\sum_k \hat{\alpha}_{k,P} e_{k,P} \tag{30}$$

where

$$\hat{\alpha}_{k,P} = M\left(\tau^{-}_{k,P}, \tau^{+}_{k,P}\right) \tag{31}$$

$$\begin{aligned} \tau^{-}_{k,P} &= \alpha_{k,w} + \tfrac{1}{2}\beta_{k,w} \\ \tau^{+}_{k,P} &= \alpha_{k,e} - \tfrac{1}{2}\beta_{k,e} \end{aligned} \tag{32}$$

$$\begin{aligned} \alpha_{1,w} &= -\tilde{\rho}_w\left(u^1_P - u^1_W\right) + \frac{(p_P - p_W)}{\tilde{c}_w} \\ \alpha_{2,w} &= -\tilde{\rho}_w\left(w^1_P - w^1_W\right) - \frac{(p_P - p_W)}{\tilde{c}_w} + \tilde{c}_w(\rho_P - \rho_W) \\ \alpha_{3,w} &= \tilde{\rho}_w\left(w^1_P - w^1_W\right) - \frac{(p_P - p_W)}{\tilde{c}_w} + \tilde{c}_w(\rho_P - \rho_W) \\ \alpha_{4,w} &= \tilde{\rho}_w\left(u^1_P - u^1_W\right) + \frac{(p_P - p_W)}{\tilde{c}_w} \\ \alpha_{1,e} &= -\tilde{\rho}_e\left(u^1_E - u^1_P\right) + \frac{(p_E - p_P)}{\tilde{c}_e} \\ \alpha_{2,e} &= -\tilde{\rho}_e\left(w^1_E - w^1_P\right) - \frac{(p_E - p_P)}{\tilde{c}_e} + \tilde{c}_e(\rho_E - \rho_P) \\ \alpha_{3,e} &= \tilde{\rho}_e\left(w^1_E - w^1_P\right) - \frac{(p_E - p_P)}{\tilde{c}_e} + \tilde{c}_e(\rho_E - \rho_P) \\ \alpha_{4,e} &= \tilde{\rho}_e\left(u^1_E - u^1_P\right) + \frac{(p_E - p_P)}{\tilde{c}_e} \end{aligned} \tag{33}$$

$$\beta_{k,e} = M(\beta_{k,P}, \beta_{k,E}), \qquad \beta_{k,w} = M(\beta_{k,W}, \beta_{k,P}) \tag{34}$$

$$\begin{aligned} \beta_{k,W} &= \alpha_{k,w} - \alpha_{k,ww}, \qquad \beta_{k,P} = \alpha_{k,e} - \alpha_{k,w} \\ \beta_{k,E} &= \alpha_{k,ee} - \alpha_{k,e} \end{aligned} \tag{35}$$

Similar expressions follow for the right state and other cell faces.

At low Mach number (Ma), pressure and density are weakly linked. As a result, use of these expressions in regions of the flow where the local Mach number is low ($Ma < 0.3$) could lead to numerical instabilities. To overcome these difficulties and at the same time to extend its applicability down to the incompressible limit, in the SIMPLENO method the characteristic interpolation practice and the use of Roe's Riemann solver are only employed in regions where the local Mach number exceeds 0.3. Below this mark the pressure-weighted interpolation method (PWIM),[18] together with reconstruction of the primitive variables, is used for computation of the inviscid flux at the interfaces,

$$N^1_e = \left(F^1_e, F^1_e v^1_e + p_e c^1_1, F^1_e v^2_e + p_e c^1_2, F^1_e H_e\right) \tag{36}$$

where the mass flux F is evaluated by the PWIM and the interface values of v, ρ, and H are computed, for instance, for density:

$$\rho_e = q^+ \rho^+_e + q^- \rho^-_e \tag{37}$$

where $\rho^{\pm}_e$ are calculated as explained in the previous section and q is defined as

$$q^+ = \begin{cases} \dfrac{F^1_e - |F^1_e|}{2F^1_e}, & \text{if } F^1_e \neq 0 \\ 0, & \text{if } F^1_e = 0 \end{cases} \tag{38}$$

and $q^- = 1 - q^+$. Similarly for the other variables. In Eq. (36) pressure is interpolated linearly to preserve the elliptic character of the flow in this range of Mach numbers.

Substitution of the preceding equations into the discrete Navier–Stokes equations gives rise to a highly coupled, nonlinear system of equations that we denote by

$$L(Q) = f \tag{39}$$

where L is the nonlinear operator acting on the nodal values after discretization. Linearization of this equation is a crucial step in the algorithm because it must reconcile computer storage requirements with CPU needs for convergence. Since these are antagonistic requirements, a compromise is established with lower levels of computer storage. This is achieved by using a segregate semi-implicit pressure-correction method along with the approximate technique proposed by Orzag[19] (for use with implicit spectral methods), i.e.,

$$L_{\text{ap}} Q^{n+1} = L_{\text{ap}} Q^n - \alpha[L(Q^n) - f] \tag{40}$$

where L_{ap} is a linear robust approximate operator to L [for brevity we denote the linear operator $L_{\text{ap}}(Q)$ by $L_{\text{ap}} Q$. In the present work we define L_{ap} as PWIM together with the first-order upwind scheme for the convective flux, the second-order central differences for the longitudinal diffusive flux, and lumping of the cross derivatives. Having linearized the discrete equations, we proceed with the definition of a solution procedure for the overall linearized system of equations. This procedure can be arranged into the following predictor-corrector scheme.

Predictor Level

First we rewrite the linearized momentum equations as follows:

$$v^i_P = \Psi v^i_P + (\Omega^{1i} p + \Omega^{2i} p)_P + b^i_P \tag{41}$$

where the operators Ψ and Ω are defined by

$$\Psi v^i_P = \frac{\sum_{m \in NB} a_m \underline{v^i_m}}{\underline{a^{v^i}_P}} \tag{42}$$

$$\Omega^{\alpha i} p = -\frac{\delta_P\left(p c^{\alpha}_{\underline{k}} \delta^{\underline{k}i}\right)}{\underline{a^{v^i}_P}} \tag{43}$$

The underlined indexes indicate an exception to the summation convention, NB is the index set corresponding to the points involved in the cell face flux evaluation, and the a_i ($i = m, p$) coefficients consist of all contributions that come from convective and diffusive flux in L_{ap}.

In the predictor step velocity fields are computed separately by solving Eq. (41), i.e.,

$$v^{i*}_P = \Psi v^{i*}_P + (\Omega^{1i} p^n + \Omega^{2i} p^n)_P + \left(b^i_P\right)^n \tag{44}$$

The enthalpy field is updated similarly. This procedure (characteristic of the semi-implicit pressure correction schemes and retained in SIMPLENO) allows for the use of an efficient pentadiagonal solver for the segregate solution of the primitive variables. [In the present work we use the strongly implicit procedure (SIP) of Stone.[20]] This enhance the robustness of the overall algorithm while requiring low levels of computer storage.

The available density and velocity fields satisfy the linearized momentum equations but usually do not satisfy mass balance over the control volumes, i.e.,

$$S_m = F^{1*}_e - F^{1*}_w + F^{2*}_n - F^{2*}_s \neq 0 \tag{45}$$

$$F^{1*}_e = \left[\rho^n\left(v^{1*} c^1_1 + v^{2*} c^1_2\right)\right]_e \tag{46}$$

Corrector Level

In the corrector step correction fields are derived that yield density and velocity fields satisfying continuity and some linearized form of the momentum equations. First, we update the velocity field as follows:

$$v^{i**}_P = \Psi v^{i*}_P + (\Omega^{1i} p^* + \Omega^{2i} p^*)_P + \left(b^i_P\right)^n \tag{47}$$

Subtraction of Eq. (44) from Eq. (47) yields the following relation between the velocity and pressure corrections fields:

$$v^{i'}_P = (\Omega^{1i} p' + \Omega^{2i} p')_P \tag{48}$$

where the correction fields are defined as

$$v_P^{i'} = v_P^{i**} - v_P^{i*} \tag{49}$$

$$p' = p^* - p^n \tag{50}$$

Now consider the density field. Here we simply assume that the latter undergoes an isentropic process between old and new values. This is equivalent to a functional dependence of the type $\rho = \rho(p)$. Linearizing this relation means expanding in a power series about $p = p^n$ as

$$\rho^* = \rho^n + \left(\frac{\mathrm{d}\rho}{\mathrm{d}p}\right)^n p' + \mathcal{O}(p'^2) \tag{51}$$

and neglecting all terms involving second or higher order powers of p'. In terms of correction fields this reads

$$\rho' = (1/c^2)p' \tag{52}$$

Using the updated fields of density and velocity we can compute mass flux through cell faces as

$$F_e^{1**} = (\rho^n + \rho')_e\left[(v^{1*} + v^{1'})c_1^1 + (v^{2*} + v^{2'})c_2^1\right]_e$$

$$\approx F_e^{1*} + \rho_e'\left(v^{1*}c_1^1 + v^{2*}c_2^1\right)_e + \rho_e^*\left(v^{1'}c_1^1 + v^{2'}c_2^1\right)_e \tag{53}$$

where we omit second-order terms. As can be seen from Eq. (53) we need correction fields at the CV interfaces and again interpolation of these values is a crucial step since it must be robust and of high-order accuracy. To achieve these goals we first notice that at convergence all correction fields are zero, and so far as the method converges and the corrections vanish we can use any level of approximation for the interpolation of the correction field because at convergence the high-level accuracy of the interpolation of the mass flux is recovered. Taking these facts into account we evaluate the interface values of velocity and density corrections as follows: 1) we consider at interface, by analogy with the nodal relation, the following dependence between velocity and pressure correction fields:

$$v_e^{i'} = (\Omega^{1i}p' + \Omega^{2i}p')_e \tag{54}$$

and 2) we use first-order upwind (with convective velocity) for the density correction at interfaces. Similar expressions follow for other cell faces.

In the first assumption we assume the existence of an irrotational velocity correction field in direct (adjusted to match units) relation to the pressure field. This procedure is justified in the incompressible case where the projection of the velocity field into the null divergence space is obtained by subtracting from it its orthogonal complement (taken in the closure of the adjoint operator, which in this case is the additive symmetric of the gradient operator); see Refs. 21 and 22. For the steady compressible case it is the mass flux that is projected in the null divergence space. In the present method the gradient operator is replaced by the equivalent form in the right-hand side (RHS) of Eq. (54).

Remark 1: It is a known fact that the use of relations (48) and (54) strongly overpredicts the values of pressure corrections, which in turn can lead to numerical instabilities in regions of low Mach number. To remedy this problem we replace them by the relations proposed by the authors,[23] i.e.,

$$v_{e(P)}^{i'} = (1/\alpha_P)(\Omega^{1i}p' + \Omega^{2i}p')_{e(P)} \tag{55}$$

where $\alpha_P \in (0, 1]$.

We proceed by requiring that the corrected flux satisfy mass conservation over each control volume, i.e.,

$$F_e^{1**} - F_w^{1**} + F_n^{2**} - F_s^{2**} = 0 \tag{56}$$

Substitution of Eqs. (52) and (55) into Eq. (53) and the latter into Eq. (56) gives rise to the following linear system of equations for the pressure correction:

$$Ap' = S_m \tag{57}$$

where A is linear and represents the contributions from correction field relations between pressure correction and velocity correction and between pressure correction and density correction as explained earlier.

Remark 2: The coefficient matrix A with the preceding definitions is nondiagonal. This can be further simplified if we notice that if the nonorthogonality is not too severe, the cross term in the pressure velocity correction fields can be neglected in comparison to the normal terms. With this fact in mind, Eq. (55) for the cell faces is modified to

$$v_e^{i'} = (1/\alpha_P)\tilde{\Omega}^{1i}p'|_e \tag{58}$$

where

$$\tilde{\Omega}^{\alpha i}p = -\frac{c_k^\alpha \delta^{ki}\delta_P p}{a_P^{v_i^i}} \tag{59}$$

These simplifications lead to a pentadiagonal system of equations that can be solved using the same algorithm (SIP) as for the other variables. In addition they strengthen the diagonal dominance of A and facilitate the implementation of boundary conditions of the pressure-correction field. In some pathological situations where the nonorthogonality of the grid is too severe over a large extension of the computational domain convergence can be difficult.[24] In such situations a better grid can fix the problem. In any event the complete nondiagonal matrix could easily be solved by, for instance, the generalized minimal residual (GMRES) method.[25] Here it should be stressed again that this modification does influence the accuracy of the converged solution.

The correction step can now be concluded by prescribing the manner in which the field variables and their flux are updated. This is done in accordance with the local Mach number as follows.

1) If the local Mach number is less than 0.3, in this case we update nodal variables with the corrected fields, and with these values we compute inviscid cell face flux as explained earlier.

2) If the local Mach number is greater than or equal to 0.3, variables are updated as in case 1. However, evaluation of the inviscid flux with the available nodal variables would require an additional (one is done before the corrector step) solution of the Riemann problem at interfaces. This procedure would in turn strongly penalize the efficiency of the overall method (solution of the Riemann problem is one of the most time-consuming parts of the method.) This problem can be overcome by noticing that Roe's linearization gives rise to an incremental equation for the inviscid flux, i.e.,

$$\Delta M = \sum_k \alpha_k \lambda_k e_k \tag{60}$$

where

$$\alpha_1 = -\bar{\rho}\Delta u^1 + (\Delta p/\bar{c}), \qquad \alpha_2 = -\bar{\rho}\Delta w^1 - (\Delta p/\bar{c}) + \bar{c}\Delta\rho$$
$$\alpha_3 = \bar{\rho}\Delta w^1 - (\Delta p/\bar{c}) + \bar{c}\Delta\rho, \qquad \alpha_4 = \bar{\rho}\Delta u^1 + (\Delta p/\bar{c}) \tag{61}$$

$$\lambda_1 = (\bar{u}^1 - \bar{c})\cdot l, \qquad \lambda_2 = \lambda_3 = \bar{u}^1 \cdot l$$
$$\lambda_4 = (\bar{u}^1 + \bar{c})\cdot l \tag{62}$$

$$e_1 = (1/2\bar{c})\left(1, \bar{v}^1 - \bar{c}n_1, \bar{v}^2 - \bar{c}n_2, \bar{H} - \bar{c}\bar{u}^1\right)^T$$
$$e_2 = (1/2\bar{c})\left(1, \bar{v}^1 - \bar{c}n_2, \bar{v}^2 + \bar{c}n_1, \bar{c}\bar{w}^1 + \bar{K}\right)^T$$
$$e_3 = (1/2\bar{c})\left(1, \bar{v}^1 + \bar{c}n_2, \bar{v}^2 - \bar{c}n_1, \bar{c}\bar{w}^1 - \bar{K}\right)^T \tag{63}$$
$$e_4 = (1/2\bar{c})\left(1, \bar{v}^1 + \bar{c}n_1, \bar{v}^2 + \bar{c}n_2, \bar{H} + \bar{c}\bar{u}^1\right)^T$$

where the overbar indicates linear interpolation.

Then the second solution of the Riemann problem for interface flux computation can be avoided by 1) interpolation of the interface correction fields (in the present work we use first order upwind) and 2) projection of this fields into flux space by using Eqs. (60–63) with Δ replaced by its corresponding correction field, e.g., $\Delta p = p'$.

In both cases the enthalpy field is corrected by using the following isentropic relation:

$$h' = p'/\rho \tag{64}$$

The SIMPLENO method can be summarized by the following steps.

1) Initialize field variables: density, velocity, pressure, and enthalpy.

2) With the available variable fields calculate the coefficients of the approximate operator L_{ap} and the RHS of Eq. (40).

3) Solve Eq. (44) separately for each velocity component by using the SIP of Stone.

4) Solve the corresponding equation for enthalpy.

5) With the predicted fields compute the inviscid flux by solving a Riemann problem.

6) Compute the pseudomass source term S_m in Eq. (45) and solve Eq. (57) for p'.

7) Update nodal variables and inviscid flux by using Eqs. (49–63).

8) Repeat steps 2–7 until convergence is achieved.

At this point some observations are pertinent concerning the uniformity of the method in the various flow regimes and the transition in the interpolation practices from low Mach number (below 0.3) to higher (above 0.3) Mach numbers. The uniformity of the method comes from the fact that we use the same procedure in the iterative process for the whole spectrum of Mach numbers. What changes for the different flow regimes is how one evaluates the flux, which is nevertheless uniquely defined for each cell face. Thus, in the SIMPLENO method the interpolation practice is the same for all variables. It employs the reconstruction practice of the ENO methods to build piecewise polynomials within each control volume, which in turn provides the values of the variables at the interfaces. With these values we compute the fluxes using a Riemann solver. The main difference in the treatment of different flow regions is the Riemann solver used in the computation of the inviscid flux: the Roe's and a linear approximate solver, and also the variable used in the interpolation, the characteristic or primitive variables, both at $Ma \geq 0.3$ and <0.3, respectively. At this low value of the Mach number the flow is smooth and the discontinuity a the limit values at the interface will be of the order of the interpolation method. Thus both methods produce very similar values for the fluxes (again notice that only one of these Riemann solvers is used in the evaluation of the flux). With the flux defined in each cell face we derive an equation for the pressure correction from the continuity equation. Consequently, the pressure-correction equation works as a transfer function that receives a nonsolenoidal mass field (computed in the predictor step) to which it responds with a correction field that assures that the corrected fields meet mass conservation and satisfy some linearized form of the Navier–Stokes equations.

The linear Riemann solver is selected for $Ma < 0.3$ to stabilize the procedure, since at very low values of Mach number the continuity becomes a weak equation for the density. This fact reflects itself in the loss of effectiveness of Roe's Riemann solver, which is then replaced by the linear one. The limit value of 0.3 also saves CPU time since the PWIM approach used in this quasi-incompressible region provides high accuracy at a much lower cost.

Results and Discussion

Arc Bump Flow

As a first test case we consider the inviscid supersonic flow over a circular arc bump as proposed by Eidelman et al.[26] The configuration corresponds to a wall-mounted 4% thick circular arc bump. To the upper and lower boundaries of the channel, the solid wall boundary condition was applied. At the inlet (left boundary) the condition of $Ma_{-\infty} = 1.65$ is used and at outlet all variables are linearly extrapolated (details can be found in the reference cited). Figure 2 shows the geometry and the grid comprising 80×30 control volumes used in the computations. Although the mathematical model is simpler than the Navier–Stokes equations the presence of discontinuities and their interactions (shock-shock interaction and shock reflection) serve as a good test for the capabilities of the method to resolve shock waves and their interactions.

Figure 3 depicts the results obtained using the SIMPLE method together with a third-order interpolation for v and h and the first-order upwind scheme for ρ. Apart from the interpolation of primitive or conservative variables, this is the standard approach used together with pressure-correction methods.[6–11] This figure clearly indicates the inadequacy of the latter for simulating flows with shocks: the shock profiles are too smeared. This situation can be significantly improved by using a second-order nonoscillatory procedure for all variables, including density, as done by the authors in a previous work.[27] Figure 4 reproduces from Ref. 27 the results obtained using the latter technique. The resolution of shock waves has been clearly improved. However, the use of a nonconservative discretization for pressure and the use of convective velocity as upwind indicator still diffuses shocks and creates spurious oscillations.

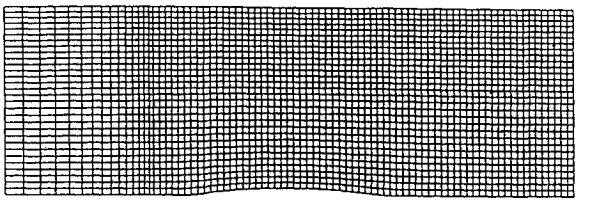


Fig. 2 Arc bump flow: geometry and 80 × 30 mesh.

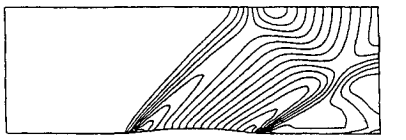


Fig. 3 Arc bump flow: isomach lines using QUDS (v, h) and UDS (ρ).

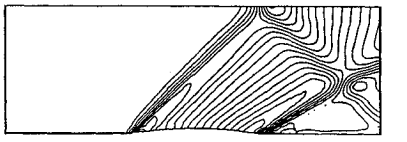


Fig. 4 Arc bump flow: isomach lines using minmod (v, h, ρ).

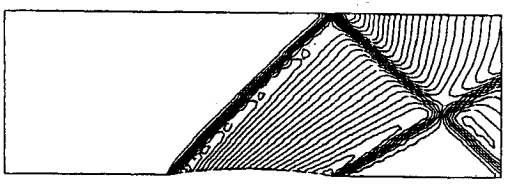


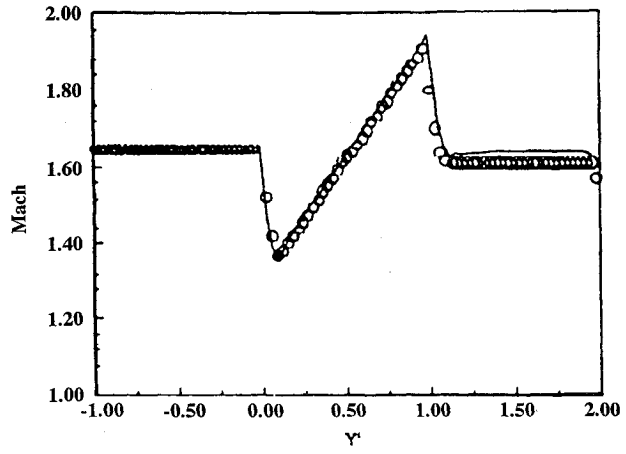
Fig. 5 Arc bump flow: isomach lines using SIMPLENO.

Figure 5 shows the results obtained with the new methodology SIMPLENO. As can be seen from this figure all shock structures are very well resolved. Their representation is crisp, nonoscillatory, and in perfect agreement with the results of Eidelman et al.,[26] who used a density-based second-order Godunov method.

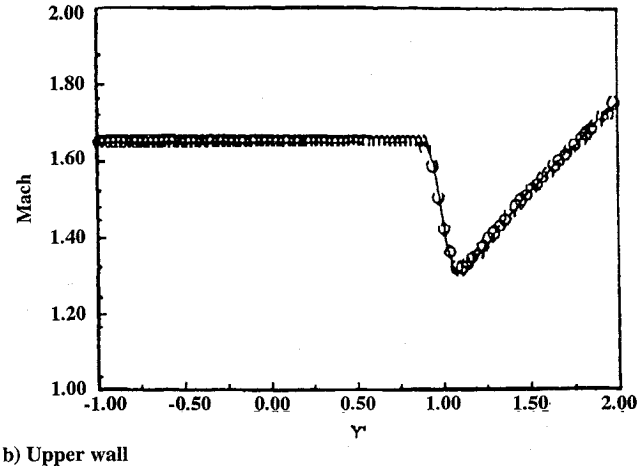
Figure 6 displays the quantitative comparison between the present results and the results of Eidelman et al.[26] These results only corroborate the preceding qualitative assertions about SIMPLENO, while letting no doubt exist about its capacity to resolve shock waves and their interactions.

Double Throat Nozzle

This flow problem first appeared in a workshop on numerical computation of compressible viscous flows.[28] In the workshop very accurate data were obtained that can be used as reference values. Figure 7 shows the geometry and the grid comprising 80×20 control volumes used in the simulation. The relevant parameters are the Reynolds number based on the stagnation properties and the Prandtl number, which are equal to 4×10^2 and 0.72, respectively. At the inlet, total pressure and total enthalpy are imposed, at the wall no slip is used, at the centerline symmetry conditions are applied, and at outlet all variables are linearly extrapolated. At this level of Reynolds number an extended separated zone is created attached to the upper wall caused by its curvature in the converging part of the second throat. This separated bubble gives rise to a strong shear layer that interacts with the normal shock wave formed underneath it. All these features make this test case an effective means for assessing



a) Lower wall



b) Upper wall

Fig. 6 Arc bump flow: pressure distribution.

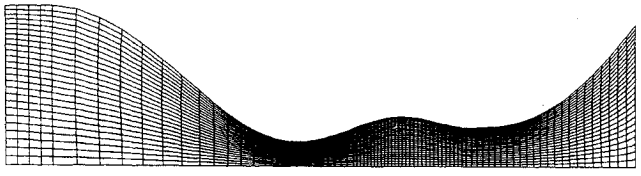
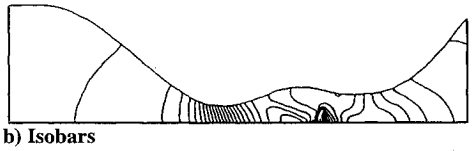


Fig. 7 Double throat flow: geometry and 80 × 20 mesh.

a) Isomach lines



b) Isobars

Fig. 8 Double throat flow.

the capacity of the method for simulating shock wave/viscous layer interactions.

Figure 8 shows the isomach and isobars obtained with SIMPLENO. Again the new methodology predicts a very sharp shock profile with no spurious oscillations or instabilities in the low Mach number regions. In particular it provides, in passing, an answer to the question on the effect of the flux limiters on the viscous layer, where strong yet continuous gradients arise. This is quantified in Figs. 9–11, where the present results are compared against

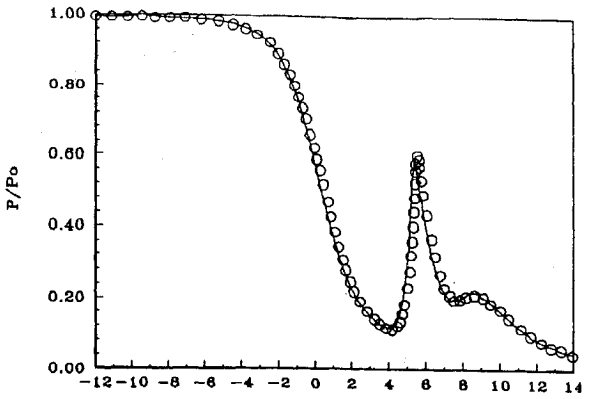


Fig. 9 Double throat flow: pressure distribution at the centerline.

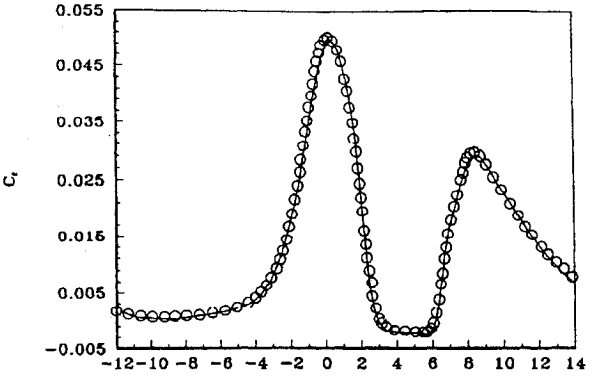


Fig. 10 Double throat flow: friction coefficient distribution at the upper wall.

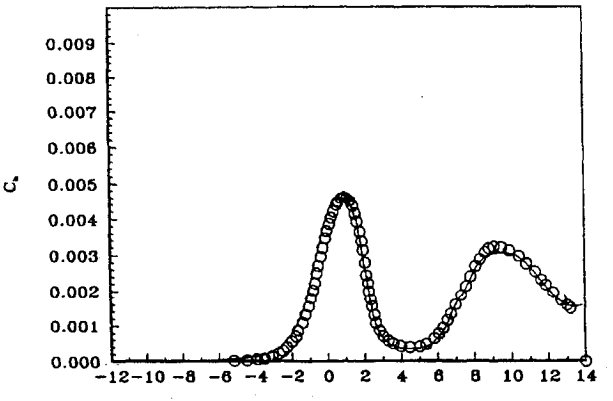


Fig. 11 Double throat flow: heat flux coefficient distribution at the upper wall.

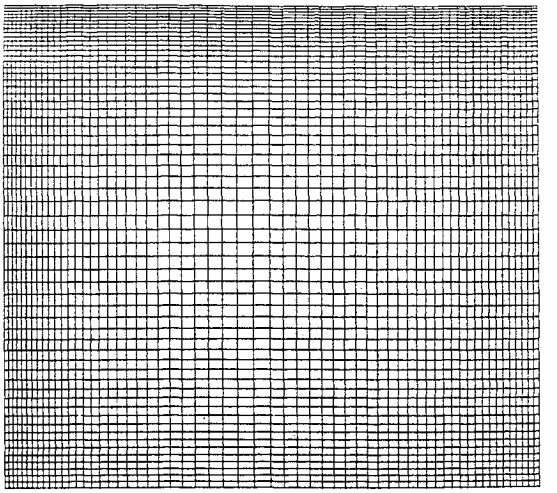


Fig. 12 Lid-driven cavity flow: 60 × 60 nonuniform grid.

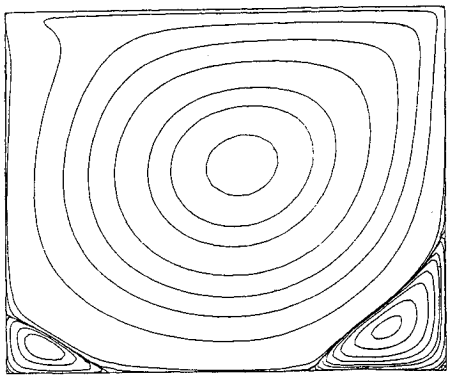
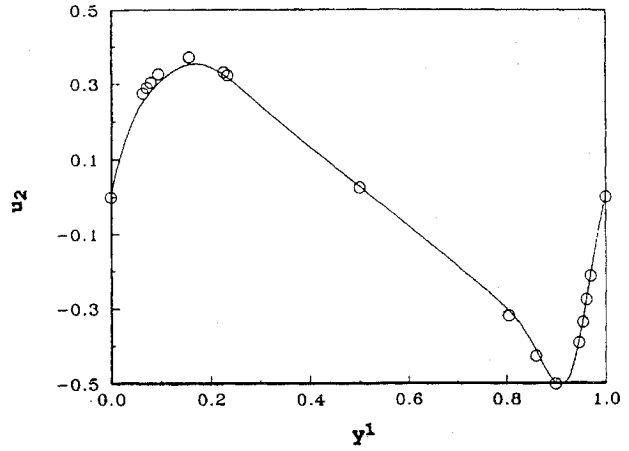
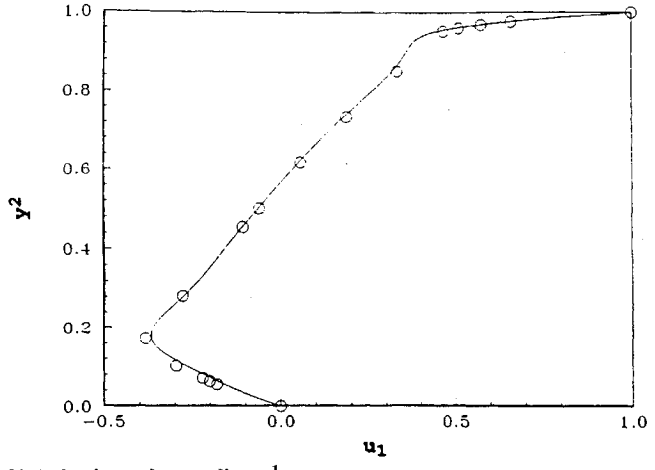


Fig. 13 Lid-driven cavity flow: streamlines. $\Psi = \{-1.0 \times 10^{-10}, -1.0 \times 10^{-7}, -1.0 \times 10^{-5}, -1.0 \times 10^{-4}, -0.01, -0.03, -0.05, -0.07, -0.09, -0.1, -0.11, -0.115, -0.1175, 1.0 \times 10^{-8}, 1.0 \times 10^{-7}, 1.0 \times 10^{-8}, 1.0 \times 10^{-6}, 1.0 \times 10^{-5}, 5.0 \times 10^{-5}, 1.0 \times 10^{-4}, 2.5 \times 10^{-4}, 5.0 \times 10^{-4}, 1.0 \times 10^{-3}, 1.5 \times 10^{-3}, 0.0 \times 10^{-3}\}$.



a) At vertical centerline, v^2



b) At horizontal centerline, v^1

Fig. 14 Lid-driven cavity flow: velocity profiles

reference values[28] obtained using a density-based method. It is evident from this figure that the flux limiters do not prevent the high accuracy of the method in viscous flows neither in the shock region with perfect agreement with the reference values nor in the shear layers where both the extension and position of the separated zones are in excellent agreement with reference values.

Lid-Driven Cavity Flow

To complete this section the test case of the flow induced by a moving wall is considered. Because of the extent and intensity of the vortices and the vortices[9] interactions present in the domain, this flow has became a standard test case for assessing Navier–Stokes solvers and convection discretization schemes. The reference values selected here are those of Ghia et al.,[29] who used a very high number of grid nodes and a second-order-accurate scheme, which are generally accepted as converged solutions. The boundary conditions for this test case are no-slip conditions, where all of the walls are at rest except the upper one, which has velocity equal to 1.

Figure 12 shows the 60×60 nonuniform grid used in the computations. Figure 13 depicts the streamlines obtained with SIMPLENO for the incompressible lid-driven cavity flow, at $Re = 1 \times 10^3$. The results display very good agreement with the reference lines with good resolution of the separated zones, both in its location and extension.

The quantitative comparison is provided in Fig. 14, where the present results are plotted together with the reference values of Ghia et al. There we can see that the agreement with the reference values of Ghia et al. is very good. Again no adverse effect due to the use of flux limiters is perceptible. The profiles are sharp and free of spurious oscillations.

Conclusions

This study has described a formulation of an accurate all-speed and multidimensional calculation procedure based on a pressure-velocity formulation that can be applied with equal effectiveness to compressible and incompressible flow.

The SIMPLENO algorithm introduces an array of new features: the implementation of interpolation of the characteristic variables, the approximation of the nonlinear operator by a robust segregate linear operator, the use of an approximate Riemann solver for the computation of the cell face flux in the pressure-correction framework, and the projection of the correction on the fluxes' space. Simulation of flows under severe conditions (shock/shock interaction, shock reflection, shock/viscous layer interaction, etc.) has demonstrated that the method is as accurate as its counterparts, which use density as a primary dependent variable in the hyperbolic regions of the flows, while extending its range of applications to the limit of very low (zero) Mach number flows.

Acknowledgments

The authors would like to acknowledge the referees' comments, which contributed to the improvement of the paper.

References

[1]Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, 1983, pp. 357–393.

[2]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[3]Patankar, S. V., and Spalding, D. B., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows," *International Journal of Heat and Mass Transfer*, Vol. 15, 1972, p. 1782.

[4]Van Doormaal, J. P., and Raithby, G. D., "Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows," *Numerical Heat Transfer*, Vol. 7, 1984, p. 147.

[5]Issa, R. I., "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics*, Vol. 62, 1985, pp. 40–62.

[6]Issa, R. I., and Lockwood, F. C., "On the Prediction of Two-Dimensional Supersonic Viscous Interactions Near Walls," *AIAA Journal*, Vol. 15, No. 2, 1977, pp. 182–188.

[7]Van Doormaal, J. P., Raithby, G. D., and McDonald, B. H., "The Segregated Approach to Predicting Viscous Compressible Fluid Flows," *Journal of Turbomachinery*, Vol. 109, April 1987, pp. 268–277.

[8]Shyy, W., "A Numerical Study of Two-Dimensional Compressible Navier–Stokes Flows," *Numerical Heat Transfer*, Vol. 14, 1988, pp. 323–341.

[9]Karki, K. C., and Patankar, S. V., "Pressure Based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations," *AIAA Journal*, Vol. 27, No. 9, 1989, pp. 1167–1174.

[10]McGuirk, J. J., and Page, G. L., "Shock Capturing Using a Pressure-Correction Method," *AIAA Journal*, Vol. 28, No. 10, 1990, pp. 1751–1757.

[11]Lien, F.-S., and Leschziner, M. A., "A Pressure-Velocity Solution Strategy for Compressible Flow and Its Application to Shock/Boundary-Layer Interaction Using Second-Moment Turbulence Closure," *Journal of Fluids Engineering*, Vol. 115, Dec. 1993, pp. 717–725.

[12]Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R., "Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III," *Journal of Computational Physics*, Vol. 71, No. 2, 1987, pp. 231–303.

[13]Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.

[14]Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere, New York, 1984.

[15]Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics*, Vols. 1 and 2, Springer–Verlag, Berlin, 1988.

[16]Hirsch, C., *Numerical Computation of Internal and External Flows*, Vols. 1 and 2, Wiley, New York, 1990.

[17]Harten, A., and Osher, S., "Uniformly High-Order Accurate Nonoscillatory Schemes., I," *Journal on Numerical Analysis*, Vol. 24, No. 2, 1987, pp. 279–309.

[18]Miller, T. F., and Schmidt, F. W., "Use of a Pressure-Weighted Interpolation Method for the Solution of the Incompressible Navier–Stokes Equations on a Non-Staggered Grid System," *Numerical Heat Transfer*, Vol. 14, 1988, pp. 213–233.

[19]Orzag, S. A., "Spectral Methods for Problems in Complex Geometries," *Journal of Computational Physics*, Vol. 37, 1980, pp. 70–92.

[20]Stone, H. L., "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations," *Journal on Numerical Analysis*, Vol. 5, No. 3, 1968, pp. 530–573.

[21]Chorin, A. J., "Numerical Solution of the Navier–Stokes Equations," *Mathematics of Computation*, Vol. 22, 1968, pp. 745–762.

[22]Bell, J. B., Solomon, J. M., and Szymczak, W. G., "Projection Method for Viscous Incompressible Flows on Quadrilateral Grids," *AIAA Journal*, Vol. 32, No. 10, 1994, pp. 1961–1969.

[23]Kobayashi, M. H., and Pereira, J. C. F., "Comparison of Several Convection Discretization Schemes for All Mach Arbitrary 2D Flows," *Proceedings of the 7th International Conference on Numerical Methods in Laminar and Turbulent Flows*, Vol. 7, Pt. 2, Pineridge, Swansea, Wales, UK, 1991, pp. 863–873.

[24]Coelho, P., and Pereira, J. C. F., "Calculation Procedure for 3D Laminar Flows in Complex Geometries Using a Nonstaggered Nonorthonormal Grid System," *Applied Mathematical Modeling*, Vol. 17, 1993, pp. 562–575.

[25]Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Non-symmetrical Linear Systems," *SIAM Journal on Scientific and Statistical Computation*, Vol. 17, No. 3, 1986, pp. 856–869.

[26]Eidelman, S., Colella, P., and Shreeve, R. P., "Application of the Godunov Method and Its Second-Order Extension to Cascade Flow Modeling," *AIAA Journal*, Vol. 22, No. 11, 1984, pp. 1609–1615.

[27]Kobayashi, M. H., and Pereira, J. C. F., "Predictions of Compressible Viscous Flows at All Mach Number Using Pressure Correction, Collocated Primitive Variables and Non-Orthogonal Meshes," AIAA Paper 92-0426, Jan. 1982.

[28]Bristeau, M. O., Glovinsk, R., Periaux, J., and Viviand, H., *Numerical Simulation of Compressible Navier–Stokes Flows*, GAMM Workshop, Vol. 18, Vieweg, Brunswick, Germany, 1987.

[29]Ghia, U., Ghia, K. N., and Shin, C. T., "High-Re Solutions for Incompressible Flow Using the Navier–Stokes Equations and a Multigrid Method," *Journal of Computational Physics*, Vol. 48, 1982, p. 387.