Arch. Mech., **52**, 4–5, pp. 857–887, Warszawa 2000

The characteristic-based-split (CBS) algorithm, stability and boundary conditions

Dedicated to Professor Zenon Mróz on the occasion of his 70th birthday

O. C. ZIENKIEWICZ and P. NITHIARASU

Institute for Numerical Methods in Engineering
University of Wales Swansea, Swansea SA2 8PP, UK

RECENTLY SEVERAL PAPERS were published on the subject of the CBS algorithm. This paper presents a brief summary of the theory of the algorithm and some recent findings on stabilization procedures and appropriate boundary conditions. Several solutions are also presented in this paper.

1. Introduction

IN RECENT YEARS, we have published several papers concerning the basis and applications of the CBS algorithm to a large variety of fluid dynamics problems [1 – 9]. The present paper summarises the theory of the algorithm and demonstrates its practical applications. It also contains some other extensions which are important for increasing the accuracy of the CBS procedure.

The algorithm can be used in explicit, semi-implicit, nearly implicit and indeed in fully implicit forms. The algorithm was initially based on a series of preliminary studies conducted between 1990 and 1995 [10 – 12] but the split was correctly introduced only later in Refs. [1, 2].

Before the present algorithm was available, most successes of finite difference and finite element methods in fluid dynamics were based on some variant of the Lax-Wendroff [13] scheme which, by approximating better to the time derivative, also introduced stabilization of the convective terms. The method, when applied to finite elements, became known as the Taylor-Galerkin method [14]. Later it was discovered that for scalar variables, a direct algorithm utilizing the optimal approximation along the characteristics, the characteristic Galerkin method, could be shown to be identical to the Taylor-Galerkin [15, 16] method.

The identity of the two procedures exists only in the scalar case described by the well-known convection-diffusion equation which is often used as a model for fluid dynamics problems. However, for problems involving several variables,

typical of fluid dynamics, the application of characteristic Galerkin method is not possible since only a single characteristic speed must be involved. For this reason, the Taylor-Galerkin procedure has been widely used [17 – 19], though giving a sub-optimal approximation. Motivation of the development of the present algorithm came from the fact that characteristic Galerkin procedures for a scalar variable are optimal in the sense of approximation, and that with suitable splitting, these can be applied for the first stage of the solution of the fluid dynamics equations. The remainder of the split being self-adjoint, can then be solved optimally using the Galerkin procedure.

This split follows the general process initially introduced by CHORIN [20] for incompressible flow problems. Further motivation which originated the new algorithm is based on an additional benefit coming from the fact, already observed, that for incompressible situations, the algorithm permitted equal (and indeed arbitrary) interpolations for all the variables used. This sidesteps the Babuska-Brezzi requirement in finite elements and similarly, all difficulties in standard finite difference schemes. In the present form, the solution turns out to be fully accurate with arbitrary interpolation for velocity and pressure for full incompressibility.

The algorithm involves time integration and in general, the time-step size will be limited by the nature of the time stepping procedure adopted in each part of the split. If a fully explicit procedure is adopted in the first part of the split then the time step is governed by a Courant number defined in terms of the flow velocity $|\mathbf{u}|$ and viscosity ν. If an explicit scheme is also used for the second part of the split then a Courant number depending on the compressible wave celerity c is invoked and may control the time-step size. We shall thus have here several possible categories of problems.

(a) *Transonic and supersonic flows; Fully explicit process.* Here generally fully explicit computation is preferred as the time step limitations for both parts of the split are similar.

(b) *Low Mach number, incompressible flows with low viscosity, Semi-inplicit process.* Here the flow is nearly incompressible and the time step used will be governed by the first part of the split with the Courant number not affected much by viscosity.

(c) *Low Mach number with high viscosity; Nearly implicit process.* In this range of flow, the limiting time step again is governed by the first part of the split but viscosity may pose here very severe limits. In such a regime, a 'nearly' implicit procedure is recommended in which the viscous terms are treated implicitly.

2. The scalar convection-diffusion problem and the characteristic Galerkin explicit approximation

Before proceeding with the description of the full algorithm, we shall recall the application of the Characteristic Galerkin method in the explicit form to a typical convection-diffusion process with a scalar dependent variable ϕ. The governing equations can here be written always in a conservation form as

$$\frac{\partial V}{\partial t} + \frac{\partial F_i}{\partial x_i} + \frac{\partial G_i}{\partial x_i} + Q = 0, \tag{2.1}$$

with x_i being the i-th coordinate (i = 1, 2, 3),

$$F_i = u_i \phi \tag{2.2}$$

– the convected flux,

$$G_i = -k \frac{\partial \phi}{\partial x_i} \tag{2.3}$$

– the diffusion flux; here k is the diffusion coefficient.

$$Q = Q(x, t) \tag{2.4}$$

– the source term, and

$$u_i = u_i(x, t) \tag{2.5}$$

is the velocity field which is assumed to be known.

The full equation can thus be alternatively written as

$$\frac{\partial \phi}{\partial t} = -u_j \frac{\partial \phi}{\partial x_j} + \frac{\partial}{\partial x_i}\left(k \frac{\partial \phi}{\partial x_i}\right) - Q - \phi \frac{\partial u_j}{\partial x_j} = R(\phi) \tag{2.6}$$

in which only the first term on the RHS is not self-adjoint. As that term corresponds precisely to an advection wave moving with a velocity $\mathbf{u}$, a change of coordinates to the characteristic ones given by

$$dx_i' = dx_i - u_i dt \tag{2.7}$$

makes the offending term vanish, leaving a fully self-adjoint system.

For such a self-adjoint system it is known that the standard Galerkin approximation in space is optimal but the inconvenience of a moving coordinate system is introduced. However, this can be overcome with suitable remeshing and the procedure has been used in many early solutions of the above equation. Here the work of ADYE-BREBBIA [21] and Barbara Mróz should be mentioned followed by much later work. For a complete review the reader is directed to Ref. [23]. While

the exact coordinate transformation introduced no error, the simplified procedures using a Taylor approximation within the time step eliminated the costly process of remeshing (or mesh interpolation), introducing however a time-step limitation [23]. This process is fully described in [1, 2] and also in the recent text [23], and an explicit form can be written in fixed coordinates as

$$\Delta\phi = \phi^{n+1} - \phi^n = \Delta t R(\phi)^{n+\theta_3} - \frac{\Delta t^2}{2} u_i \frac{\partial}{\partial x_i} (R(\phi))^n + O(\Delta t^3) \tag{2.8}$$

where $0 \leq \theta_3 \leq 1$ and $\mathrm{R}(\phi)$ is defined by Eq. (2.6).

This, as already mentioned, is of an identical form to that resulting from the Taylor-Galerkin procedures and the second term adds the stabilizing diffusion in the streamline direction. Indeed a similar form can be obtained here in a variety of ways and recently OÑATE [24, 25] introduced an interesting form of stability control by using so-called "finite increment" calculus (FIC). However, only the characteristic form ensures the optimal approximation as Eq. (2.8) is derived from a self-adjoint form in which spatial discretization by the Galerkin method can be optimally used. We can write thus the approximation

$$\phi = \mathbf{N}\overline{\phi}, \tag{2.9}$$

where $\mathbf{N}$ is the shape function and $\overline{\phi}$ is a nodal quantity. We use the weights $\mathbf{N}^{\mathbf{T}}$ in the integrated residual expression. Thus we obtain

$$\mathbf{M}(\overline{\phi}^{\mathbf{n+1}} - \overline{\phi}^{\mathbf{n}}) = -\Delta t[(\mathbf{C}\overline{\phi}^{\mathbf{n}} + \mathbf{K}\overline{\phi}^{\mathbf{n}} + \mathbf{f}^{\mathbf{n}}) - \Delta t(\mathbf{K_u}\overline{\phi}^{\mathbf{n}} + \mathbf{f_s^n})] \tag{2.10}$$

if we assume $\theta_3 = 0$, as is generally done in explicit form and omit higher derivatives and source terms. In the above equation

$$\begin{aligned}
\mathbf{M} &= \int_\Omega \mathbf{N}^{\mathbf{T}}\mathbf{N} d\Omega, \\
\mathbf{C} &= \int_\Omega \mathbf{N}^{\mathbf{T}} \frac{\partial}{\partial x_i}(u_i \mathbf{N}) d\Omega, \\
\mathbf{K} &= \int_\Omega \frac{\partial \mathbf{N}^{\mathbf{T}}}{\partial x_i} k \frac{\partial \mathbf{N}}{\partial x_i} d\Omega, \\
\mathbf{f} &= \int_\Omega \mathbf{N}^{\mathbf{T}} \mathbf{Q} d\Omega + \text{b.t.}
\end{aligned} \tag{2.11}$$

and $\mathbf{K_u}$ and $\mathbf{f_s^n}$ come from the new term introduced by the discretization along the characteristics. b.t. stands for the boundary terms. After integration by parts, the expression of $\mathbf{K_u}$ and $\mathbf{f_s}$ is

$$\mathbf{K_u} = -\frac{1}{2}\int_\Omega \frac{\partial}{\partial x_i}(u_i \mathbf{N}^\mathbf{T})\frac{\partial}{\partial x_i}(u_i \mathbf{N})d\Omega,$$

(2.12)

$$\mathbf{f_s} = -\frac{1}{2}\int_\Omega \frac{\partial}{\partial x_i}(u_i \mathbf{N}^\mathbf{T})\mathbf{Q}d\Omega + \text{b.t.}$$

where b.t stands for integrals extending along the region boundaries [23].
The approximation is valid for any scalar convected quantity even if that is the velocity component u_i itself, as is the case with the momentum conservation equations. For this reason we have presented above all the details of the spatial approximation, since the matrices will be repeatedly used.

It is of interest that the explicit form of the Eq. (2.10) is only conditionally stable. For one-dimensional problems, the stability condition is given as (neglecting the effect of sources)

$$\Delta t \leq \Delta t_{\text{crit}} = \frac{h}{|\mathbf{u}|} \tag{2.13}$$

for linear elements in which h is the size of the element.

In 2D problems, the critical time step may be presented as [26]

$$\Delta t_{\text{crit}} = \frac{\Delta t_u \Delta t_\nu}{\Delta t_u + \Delta t_\nu}, \tag{2.14}$$

where Δt_u is given by Eq. (2.13) and $\Delta t_\nu = h^2/2k$ is the diffusive limit for the critical one-dimensional time step.

Further, with $\Delta t = \Delta t_{\text{crit}}$ the steady-state solution results in a diffusion change (almost) identical to that obtained by using the optimal streamline upwinding procedures [23]. Thus if steady state solutions are the main objective of the computation, such a value of Δt should be used in connection with the $\mathbf{K_u}$ term.

3. The CBS algorithm for the Navier-Stokes equations

3.1. The equations of flow – Navier-Stokes problem

The full conservation form of the Navier-Stokes equations for compressible flow is traditionally written as

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{F_i}}{\partial x_i} + \frac{\partial \mathbf{G_i}}{\partial x_i} + \mathbf{Q} = 0 \tag{3.1}$$

with

$$\mathbf{V}^{\mathbf{T}} = (\rho, \rho u_1, \rho u_2, \rho E) \tag{3.2}$$

being the independent variable vector,

$$\mathbf{F}_{\mathbf{i}}^{\mathbf{T}} = (\rho u_i, \rho u_i u_1 + \delta_{i1} p, \rho u_i u_2 + \delta_{i2} p, u_i(\rho E + p)) \tag{3.3}$$

defining the convective flux vector, and

$$\mathbf{G}_{\mathbf{i}}^{\mathbf{T}} = (0, -\tau_{i1}, -\tau_{i2}, q_i - \tau_{ij} u_j) \tag{3.4}$$

defining the diffusion fluxes. Finally,

$$\mathbf{Q}^{\mathbf{T}} = (0, \rho g_1, \rho g_2, \rho(g_i u_i + q''')) \tag{3.5}$$

gives the source terms.

In the above relations, stress components τ_{ij} are related to velocity gradients by

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right); \tag{3.6}$$

u_i are the velocity components; ρ is the density; E the specific energy; q_i the heat flux, q''' the heat generation per unit mass, p the pressure; g_i the acceleration due to gravity.

The equations are completed by the universal gas law

$$p = \rho R T \tag{3.7}$$

where R is the gas constant and T is the temperature.

The sound velocity is defined, assuming constant entropy, as

$$c^2 = \frac{\partial p}{\partial \rho} = \frac{\gamma p}{\rho}. \tag{3.8}$$

Here γ is the ratio of specific heats.

Further we can write conveniently

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial p} \frac{\partial p}{\partial t} = \frac{1}{c^2} \frac{\partial p}{\partial t} \tag{3.9}$$

though this expression assumes again constant entropy and is therefore only an approximation. In what follows we shall use Eq. (3.9) but elsewhere we discuss the possibility of correcting any errors involved by amendment of the algorithm [4].

While in gas flow all the equations are fully coupled, for incompressible flows in which $c = \infty$ the energy equations can be solved independently after the velocity field has been established. Nevertheless, a single algorithm for the solution of both problems is possible as we shall now show.

Although the form of Eq. (3.1) is identical to that of the convection-diffusion problem of Eq. (2.1), three wave speeds exist and the characteristic Galerkin procedure cannot be directly applied. In the next section we show how this can be done with a fractional step.

3.2. Characteristic-based-split (CBS) algorithm

For convenience we shall rewrite Eq. (3.2) in a more direct form, omitting initially the energy equation. These equations can be solved completely in a time increment Δt as the only coupling which exists is through the speed of sound c for which we shall simply use the value at time t_n due to the explicit nature of the time stepping algorithm. Equation (3.2) can be rewritten in separate equations as (neglecting energy equation):

Continuity

$$\frac{1}{c^2}\frac{\partial p}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0. \tag{3.10}$$

Momentum

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j U_i) = \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + Q, \tag{3.11}$$

where $U_i = \rho u_i$, $Q = -\dfrac{\partial p}{\partial x_i}$. Now the temporal discretization of the above equations is considered.

3.2.1 The split-temporal discretization. We can discretize Eq. (3.11) in time using the Characteristic Galerkin process. Except for the pressure term, this equation is similar to the convection-diffusion Eq. (2.6). This term can however be treated as a known (source type) quantity provided we have an independent way of evaluating the pressure. Before proceeding with the algorithm, we rewrite Eq. (3.11) in the form given below, to which the Characteristic Galerkin process can be applied

$$\frac{\partial U_i}{\partial t} = -\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + Q^{n+\theta_2} \tag{3.12}$$

with $Q^{n+\theta_2}$ being a known quantity evaluated at $t = t^n + \theta_2 \Delta t$ in a time increment Δt. In the above equation

$$Q^{n+\theta_2} = -\frac{\partial p}{\partial x_i}^{n+\theta_2} \tag{3.13}$$

with

$$\frac{\partial p}{\partial x_i}^{n+\theta_2} = \theta_2 \frac{\partial p}{\partial x_i}^{n+1} + (1-\theta_2)\frac{\partial p}{\partial x_i}^{n} \tag{3.14}$$

or

$$\frac{\partial p}{\partial x_i}^{n+\theta_2} = \frac{\partial p}{\partial x_i}^{n} + \theta_2 \frac{\partial \Delta p}{\partial x_i}. \tag{3.15}$$

In this formula

$$\Delta p = p^{n+1} - p^n. \tag{3.16}$$

Using the Eq. (2.8) and replacing ϕ by U_i, we can write

$$U_i^{n+1} - U_i^n = \Delta t \left[-\frac{\partial}{\partial x_j}(u_j U_i)^n + \frac{\partial \tau_{ij}}{\partial x_j}^{n} + Q^{n+\theta_2} - (\rho g_i)^n + \left(\frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j}(u_j U_i) - Q + \rho g_i \right) \right)^n \right]. \tag{3.17}$$

At this stage we have to introduce the 'split' in which we substitute a suitable approximation for Q which allows the calculation to proceed before p^{n+1} is evaluated. Two alternative approximations are useful and we shall describe these as *Split A* and *Split B*, respectively. In the first we remove all the pressure gradient terms from Eq. (3.17), in the second we retain in that equation the pressure gradient corresponding to the beginning of the step, i.e. $\dfrac{\partial p}{\partial x_i}^{n}$. Though it appears that the second split might be more accurate, there are other reasons for the success of the first split which we shall refer to later. Indeed the 'Split A' is the one which we shall universally recommend.

Split A

Here we introduce an auxiliary variable U_i^* such that

$$\Delta U_i^* = U_i^* - U_i^n = \Delta t \left[-\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} - \rho g_i + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j}(u_j U_i) + \rho g_i \right) \right]^n. \tag{3.18}$$

This equation will be solved subsequently by an explicit time step applied to the discretized form, and a complete solution is now possible. The "correction" given below is available once the pressure increment is evaluated,

$$\Delta U_i = U_i^{n+1} - U_i^n = \Delta U_i^* - \Delta t \frac{\partial p}{\partial x_i}^{n+\theta_2} - \frac{\Delta t^2}{2} u_k \frac{\partial Q}{\partial x_k}. \tag{3.19}$$

From Eq. (3.10) we have

$$\Delta\rho = \left(\frac{1}{c^2}\right)^n \Delta p = -\Delta t \frac{\partial U_i}{\partial x_i}^{n+\theta_1} . \tag{3.20}$$

Replacing U_i^{n+1} with the known intermediate, auxiliary variable U_i^* and rearranging (after neglecting higher order terms), we have

$$\Delta\rho = \left(\frac{1}{c^2}\right)^n \Delta p = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \left(\frac{\partial^2 p^n}{\partial x_i \partial x_i} + \theta_2 \frac{\partial^2 \Delta p}{\partial x_i \partial x_i}\right)\right] \tag{3.21}$$

where the U_i^* and pressure terms in the above equation come from Eq. (3.19).

The above equation is fully self-adjoint in the variables Δp (or $\Delta\rho$) which are the unknowns. Now standard Galerkin-type procedure can be optimally used for spatial approximation. It is clear that the governing equations can be solved after spatial discretization in the following order:
Eq. (3.18) to obtain ΔU_i^* ;
Eq. (3.21) to obtain Δp or $\Delta\rho$;
Eq. (3.19) to obtain ΔU_i thus establishing the values at t^{n+1}.

After completing the calculation to establish ΔU_i and Δp (or $\Delta\rho$), the energy equation is dealt with independently, and the value of $(\rho E)^{n+1}$ is obtained by the Characteristic Galerkin process.

It is important to remark that this sequence allows us to solve the governing Eqs. (3.1),in an efficient manner and with adequate numerical damping. *Note that these equations are written in conservation form. Therefore, this algorithm is well suited for dealing with supersonic and hypersonic problems, in which the conservation form ensures that shocks will be placed at right positions and unique solution will be achieved.*

Split B

In this split we also introduce an auxiliary variable U_i^{**} now retaining the known values of $Q^n = -\dfrac{\partial p^n}{\partial x_i}$, i.e.

$$\Delta U_i^{**} = U_i^{**} - U_i^n = \Delta t \left[-\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_i} - \rho g_i + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j}(u_j U_i) - Q + \rho g_i\right)\right]^n . \tag{3.22}$$

It may appear that now U_i^{**} is a better approximation of U^{n+1}. We can now write the correction as

$$\Delta U_i = U_i^{n+1} - U_i^n = \Delta U_i^{**} - \theta_2 \Delta t \frac{\partial \Delta p}{\partial x_i}, \tag{3.23}$$

i.e. the correction to be applied is smaller than that assuming 'Split A' (Eq. 3.19). Further if we use the fully explicit form with $\theta_2 = 0$, no mass velocity (U_i) correction is necessary. We proceed to calculate the pressure changes as in 'Split A' as

$$\Delta\rho = \frac{1}{c^2}\Delta p = -\Delta t \left[\frac{\partial U_i}{\partial x_i}^n + \theta_1 \frac{\partial \Delta U_i^{**}}{\partial x_i} - \Delta t \theta_1 \theta_2 \frac{\partial^2 \Delta p}{\partial x_i^2} \right]. \tag{3.24}$$

The solution stages follow the same steps as in 'Split A'. The final matrix form of the above steps and energy step are given as

Split A

Step 1. Intermediate momentum

$$\Delta\tilde{\mathbf{U}}^* = -\mathbf{M}_\mathrm{u}^{-1}\Delta t \left[(\mathbf{C}_\mathrm{u}\tilde{\mathbf{U}} + \mathbf{K}_\tau\tilde{\mathbf{u}} - \mathbf{f}) - \Delta t(\mathbf{K}_\mathrm{u}\tilde{\mathbf{u}} + \mathbf{f}_\mathrm{s}) \right]^n. \tag{3.25}$$

Step 2. Pressure

$$(\tilde{\mathbf{M}} + \Delta t^2 \theta_1 \theta_2 \mathbf{H})\Delta\tilde{\mathbf{p}} = \Delta t[\mathbf{G}_\mathrm{pu}^\mathrm{T}\tilde{\mathbf{U}}^\mathrm{n} + \theta_1 \mathbf{G}_\mathrm{pu}\Delta\tilde{\mathbf{U}}^* - \Delta t\theta_1 \mathbf{H}\tilde{\mathbf{p}} - \mathbf{f}_\mathrm{p}]^n. \tag{3.26}$$

Step 3. Momentum correction

$$\Delta\tilde{\mathbf{U}} = \Delta\tilde{\mathbf{U}}^* - \mathbf{M}_\mathrm{u}^{-1}\Delta t \left[\mathbf{G}_\mathrm{up}^\mathrm{T}(\tilde{\mathbf{p}}^\mathrm{n} + \theta_2\Delta\tilde{\mathbf{p}}) + \frac{\Delta t}{2}\mathbf{P}\tilde{\mathbf{p}}^\mathrm{n} \right]. \tag{3.27}$$

Step 4. Energy

$$\Delta\tilde{\mathbf{E}} = -\mathbf{M}_\mathrm{E}^{-1}\Delta t \Big[\mathbf{C}_\mathrm{E}\tilde{\mathbf{E}} + \mathbf{C}_\mathrm{p}\tilde{\mathbf{p}} + \mathbf{K}_\mathrm{T}\tilde{\mathbf{T}} + \mathbf{K}_{\tau\mathrm{E}}\tilde{\mathbf{u}} + \mathbf{f}_\mathrm{e} - \Delta t(\mathbf{K}_\mathrm{uE}\tilde{\mathbf{E}} + \mathbf{K}_\mathrm{up}\tilde{\mathbf{p}} + \mathbf{f}_\mathrm{es}) \Big]^n. \tag{3.28}$$

Split B

With the 'Split B', the discretization and solution procedure have to be slightly modified. Leaving the details of the derivation to the Reader and using identical discretization processes, the final steps can be summarised as:

Step 1

$$\Delta\tilde{\mathbf{U}}_\mathrm{i}^{**} = -\mathbf{M}_\mathrm{u}^{-1}\Delta t \Big[(\mathbf{C}_\mathrm{u}\tilde{\mathbf{U}} + \mathbf{K}_\tau\tilde{\mathbf{U}} + \mathbf{G}_\mathrm{up}^\mathrm{T}\tilde{\mathbf{p}} - \mathbf{f}) - \Delta t(\mathbf{K}_\mathrm{u}\tilde{\mathbf{U}} + \mathbf{f}_\mathrm{s} + \tfrac{\Delta t}{2}\mathbf{P}\tilde{\mathbf{p}}^\mathrm{n}) \Big]^n; \tag{3.29}$$

here all matrices are the same as in 'Split A' except the forcing term $\mathbf{f}$ which is

$$\mathbf{f} = \int_{\Omega} \mathbf{N}_{\mathbf{u}}^{\mathbf{T}} \rho \mathbf{g} d\Omega + \int_{\Gamma} \mathbf{N}_{\mathbf{u}}^{\mathbf{T}} \mathbf{t} d\Gamma, \tag{3.30}$$

Step 2

$$(\tilde{\mathbf{M}} + \Delta t^2 \theta_1 \theta_2 \mathbf{H}) \Delta \tilde{\mathbf{p}} = \Delta t [\mathbf{G}_{\mathbf{pu}}^{\mathbf{T}} \tilde{\mathbf{U}}^{\mathbf{n}} + \theta_1 \mathbf{G}_{\mathbf{pu}} \Delta \tilde{\mathbf{U}}^{**} - \mathbf{f}_{\mathbf{p}}]^n, \tag{3.31}$$

and

Step 3

$$\Delta \tilde{\mathbf{U}} = \Delta \tilde{\mathbf{U}}^{**} - \mathbf{M}_{\mathbf{u}}^{-1} \Delta t \left[\theta_2 \mathbf{G}_{\mathbf{up}}^{\mathbf{T}} \Delta \tilde{\mathbf{p}}\right]. \tag{3.32}$$

The 'Step 4' , calculation of energy, is unchanged. The Reader can notice the minor differences in the above equations from those of 'Split A'. For details of the matrices involved, the Reader can consult [1, 2, 7, 23].

4. Explicit, semi-implicit and nearly implicit forms

The algorithm described will always contain an explicit portion in the first Characteristic Galerkin step. However the second step, i.e. that of determination of the pressure increment can be made either explicit or implicit and various possibilities exist here, depending on the choice of θ_2. Now different stability criteria will apply. We refer to schemes being fully explicit or semi-implicit, depending on the choice of the parameter θ_2.

It is also possible to solve the first step in a partially implicit manner to avoid severe time-step restriction due to the viscous term. Now the viscous term is the one for which an implicit solution is sought. We refer to such schemes as quasi (nearly) implicit schemes.

4.1. Fully explicit form

In fully explicit forms, $1/2 \leq \theta_1 \leq 1$ and $\theta_2 = 0$. In general, the time step limitations explained for the convection diffusion equations are applicable, i.e.

$$\Delta t \leq \frac{h}{c + |\mathbf{u}|}, \tag{4.1}$$

since viscosity effects are generally negligible here.

This particular form is very successful in compressible flow computations and has been widely used by the authors for solving many complex problems [6].

4.2. Semi-implicit form

In semi-implicit form the following values apply:

$$\frac{1}{2} \leq \theta_1 \leq 1, \qquad \frac{1}{2} \leq \theta_2 \leq 1. \tag{4.2}$$

Again the algorithm is conditionally stable. The permissible time step is governed by the critical step of the Characteristic Galerkin explicit relation solved in Step 1 of the algorithm. This is the standard convection–diffusion problem and the same stability limits apply Eq. (2.14).

For slightly compressible or incompressible problems in which $\tilde{\mathbf{M}}$ is small or zero, the semi-implicit form is efficient and it should be noted that the matrix $\mathbf{H}$ of Eqs. (3.26) and (3.31) does not vary during the computation process. Therefore $\mathbf{H}$ can be partially inverted leading to an economical procedure.

4.3. Quasi (nearly) implicit form

To overcome the severe time step restriction made by the diffusion terms (viscosity, thermal conductivity etc.), these terms can be treated implicitly This involves solving separately an implicit form connecting the viscous terms with U_i^* or U_i^{**}. Here at each step, simultaneous equations need to be solved and this procedure can be of great advantage in certain cases such as high viscosity flows and low Mach number flows. Now the only time step limitation is $\Delta t \leq h/|\mathbf{u}|$ which appears to be a very reasonable and physically meaningful restriction.

4.4. Evaluation of time-step limit. Local and global time-steps

The time step limits, in spite of being defined in terms of element sizes, are best calculated at nodes of the element. In the calculation we shall specify, if the scheme is conditionally stable, the time step limit at each node by assgning the minimum value for such nodes calculated from all the surrounding elements. When a problem is being solved in real time, then obviously the smallest of all nodal values has to be adopted for the solution. In many problems transient calculation is adopted to find steady-state solutions, and *local time stepping* is convenient as it allows more rapid convergence and fewer time-steps to be used throughout the problems. Local time stepping can only be applied to problems in which (1) – the mass matrix is lumped, and (2) – the steady-state solution does not depend on the mass matrix. Thus with local time stepping we shall use at every node simply the minimum time-step found at that node. This is, of course, equivalent to assuming identical time steps for the whole problem and simply adjusting the lumped masses. Such a problem with lumped masses adjusted is

still physically and mathematically meaningful and we know that the convergence will be achieved as it invariably is.

Many of the steady-state problems were solved by means of such localized time stepping used in the calculations.

In the context of local and global time stepping it is interesting to note that the stabilizing terms introduced by the Characteristic Galerkin process will not take on the optimal value for any element in which the time-step differs from the critical one. However, on other occasions it may be useful to make sure that (a) in all elements we have introduced optimal damping, (b) the progressive time-step for all elements is identical. The latter of course is absolutely necessary if for instance we deal with transient problems where all time steps are real. For such cases it is possible to consider Δt as being introduced in two stages: (1) as the Δt_{ext} (external) which has of course to preserve stability and that must be left at minimum Δt calculated from any element, and (2): to use inside the calculation of each individual element the Δt_{int} (internal) which is optimal for an element as of course exceeding the stability limit does not matter there and we are simply adding a better damping characteristics.

This internal – external subdivision is of some importance when incompressibility effects are considered. As shown in the next section, the stabilizing diagonal term occurring in steady state depends on the size of the time-step. If the mesh is graded and very small elements dictate the time-step over the whole domain, we might find that the diagonal term introduced overall is not sufficient to preserve incompressibility. For such problems we recommend the use of internal and external time-step which differ and we introduce them in Ref. [9].

5. Circumventing the BB restrictions

In the previous sections we have not restricted the nature of the interpolating shape functions $\mathbf{N_u}$ and $\mathbf{N_p}$, i.e. shape functions for velocity and pressure, respectively. If we chose these interpolations in a manner satisfying the patch test conditions or BB restriction for incompressibility then, of course, completely incompressible problems can be dealt with without any special difficulties by both the 'Split A' and 'Split B' formulations. However, the 'Split A' of the formulation described introduces an important bonus which permits us to avoid any restrictions on the nature of the two shape functions. Let us examine here the structure of equations reached in steady conditions. For simplicity we shall consider here only the Stokes form of governing equations in which the convective terms disappear. Further we shall take the fluid as incompressible and thus uncoupled from the energy equations. Now the three steps of Eqs. (3.18), (3.21) and (3.19) are

written as

$$
\begin{aligned}
\Delta\tilde{\mathbf{U}}^* &= -\Delta t \mathbf{M}^{-1}[\mathbf{K}_\tau \mathbf{u}^n - \mathbf{f}], \\
\Delta\tilde{\mathbf{p}} &= \frac{1}{\Delta t \theta_1 \theta_2}\mathbf{H}^{-1}[\mathbf{G}_{pu}^T \tilde{\mathbf{U}}^n + \theta_1 \mathbf{G}_{pu}\Delta\mathbf{U}^* - \Delta t\theta_1 \mathbf{H}\tilde{\mathbf{p}}^n - \mathbf{f}_p], \\
\Delta\tilde{\mathbf{U}} &= \Delta\tilde{\mathbf{U}}^* - \Delta t \mathbf{M}^{-1}\mathbf{G}_{up}^T(\tilde{\mathbf{p}}^n + \theta_2\Delta\tilde{\mathbf{p}}).
\end{aligned} \tag{5.1}
$$

In steady state we have $\Delta\tilde{\mathbf{p}} = \Delta\tilde{\mathbf{U}} = 0$ and eliminating $\Delta\tilde{\mathbf{U}}^*$ we can write (dropping now the superscript $\mathbf{n}$)

$$
\mathbf{K}_\tau \tilde{\mathbf{u}} + \mathbf{G}_{up}^T \tilde{\mathbf{p}} = \mathbf{f} \tag{5.2}
$$

from the first and third of Eqs. (5.1), and

$$
\mathbf{G}_{pu}^T \tilde{\mathbf{U}} + \theta_1 \Delta t \mathbf{G}_{pu}\mathbf{M}^{-1}\mathbf{G}_{up}^T \mathbf{p} - \Delta t \theta_1 \mathbf{H}\mathbf{p} - \mathbf{f}_p = 0 \tag{5.3}
$$

from the second and third of Eqs. (5.1)

We finally have a system which can be written in the form

$$
\begin{bmatrix} \mathbf{K}_\tau/\rho & \mathbf{G}_{up}^T \\ -\mathbf{G}_{pu}^T & \Delta t\theta_1 \left[\mathbf{H} - \mathbf{G}_{pu}\mathbf{M}^{-1}\mathbf{G}_{up}^T\right] \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix}; \tag{5.4}
$$

here $\mathbf{f}_1$ and $\mathbf{f}_2$ follow from the forcing terms.

The system is now always positive definite and therefore leads to non-singular solution for *any interpolation functions* $\mathbf{N}_u$, $\mathbf{N}_p$ chosen. In all of the examples discussed in this paper and elsewhere, equal interpolation is chosen for both the U_i and p variables, i.e. $\mathbf{N}_u = \mathbf{N}_p$. We must however stress that any other interpolation can be used without violating the stability. This is an important reason for the preferred use of 'Split A' form.

It can be easily verified that if the pressure gradient term is retained as in Eq. (3.22), i.e. if we use 'Split B', the lower diagonal term of Eq. (5.4) is identically zero and the BB conditions in the full scheme cannot be avoided.

6. Boundary conditions

6.1. Fictitious boundaries

In a large number of fluid mechanics problems the flow in open domains is considered. In such problems, the boundaries are simply limits of computation and therefore they are fictitious. With suitable values specified at such boundaries however, accurate solution for the flow inside the isolated domain can be achieved.

1. If the flow is subsonic, the specification of all quantities except the density can be made on both the sides and entry boundaries.

2. For supersonic flows, all the variables can be prescribed at the inlet. At the exit however, no boundary conditions are imposed simply because by definition, the disturbances caused by the boundary conditions cannot travel faster than at the speed of sound.

With subsonic exit conditions the situation is somewhat more complex, and here various possibilities exist.

Condition A : Denoting the most obvious assumptions with regard to the traction and velocities.

Condition B : A more sophisticated condition of zero gradient of traction and stresses existing there. Such conditions will of course apply always to the exit domains for incompressible flow. The condition 'B' was first introduced by ZIENKIEWICZ *et al.* [10].

Of considerable importance especially in view of the new schemes, are the conditions which will be encountered on real boundaries.

6.2. Real boundaries

By real boundaries we mean limits of fluid domains which are physically defined; here three different possibilities exist.

1. *Solid boundaries with no slip conditions*: On such boundaries the fluid is assumed to stick or attach itself to the boundary and thus all velocity components become zero. Obviously, this boundary is only possible for viscous flows.

2. *Solid boundaries in inviscid, flow (slip conditions)*: When the flow is inviscid, we will always encounter slipping boundary conditions where only normal velocity component is specified and is in general equal to zero in steady-state motion. Such boundary conditions will invariably be imposed for problems of Euler flow, whether it is compressible or incompressible.

3. *Prescribed traction boundary conditions*: The last category is that on which tractions are prescribed. This includes zero traction in case of free surfaces of fluid or any prescribed tractions such as those caused by wind being imposed on the surface.

These three basic kinds of boundary conditions have to be imposed on the fluid and special consideration has to be given to these when split operator schemes are used.

7. Some solution of typical examples

In this section we illustrate the applications and show the advantages gained by the use of the CBS algorithm in various classes of problems. In all of the problems discussed, the same computer coding was used and only linear, triangular and quadrilateral elements are used in all examples.

The application of higher order elements and elements of different nature is of course possible under certain circumstances but we shall not make detailed comments on those here.

7.1. Fully explicit procedure, subsonic, transonic and supersonic flows

The use of fully explicit procedure is of course most preferred in aeronautical computations and the present CBS algorithm is well suited for this purpose.

Example 1. NACA 0012 aerofoil with zero angle of attack, $M = 0.5, 1.2$.

In Fig. 1, we show the density distribution along the centerline upto stagnation (see [3] for full details) for Mach number 0.5 with the fully explicit form of the algorithm. Here the performance of the CBS algorithm is compared with the performance of the previously used Taylor-Galerkin scheme.

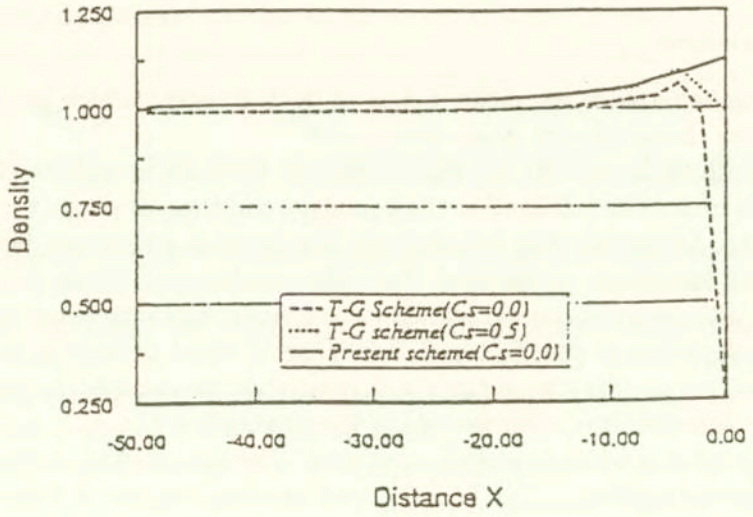


FIG. 1. Subsonic aerofoil inviscid flow past a NACA0012 aerofoil. Comparison of density along mid height. Analytical value at stagnation point 1.13. CBS scheme gives highly accurate solution.

It is interesting that the CBS algorithm improves the results dramatically near the stagnation point without the use of any additional artificial diffusion (which is essential to get any reasonable result using the Taylor-Galerkin scheme).

In Figs. 2, 3 and 4, the domain, mesh and results obtained are given for a supersonic case with Mach number equal to 1.2 of flow around the same aerofoil. The domain size taken is big enough to specify the free-stream conditions at the inlet. As can be seen, the results obtained are very smooth (Fig. 3) and compare excellently with the benchmark AGARD results [27] (Fig. 4).

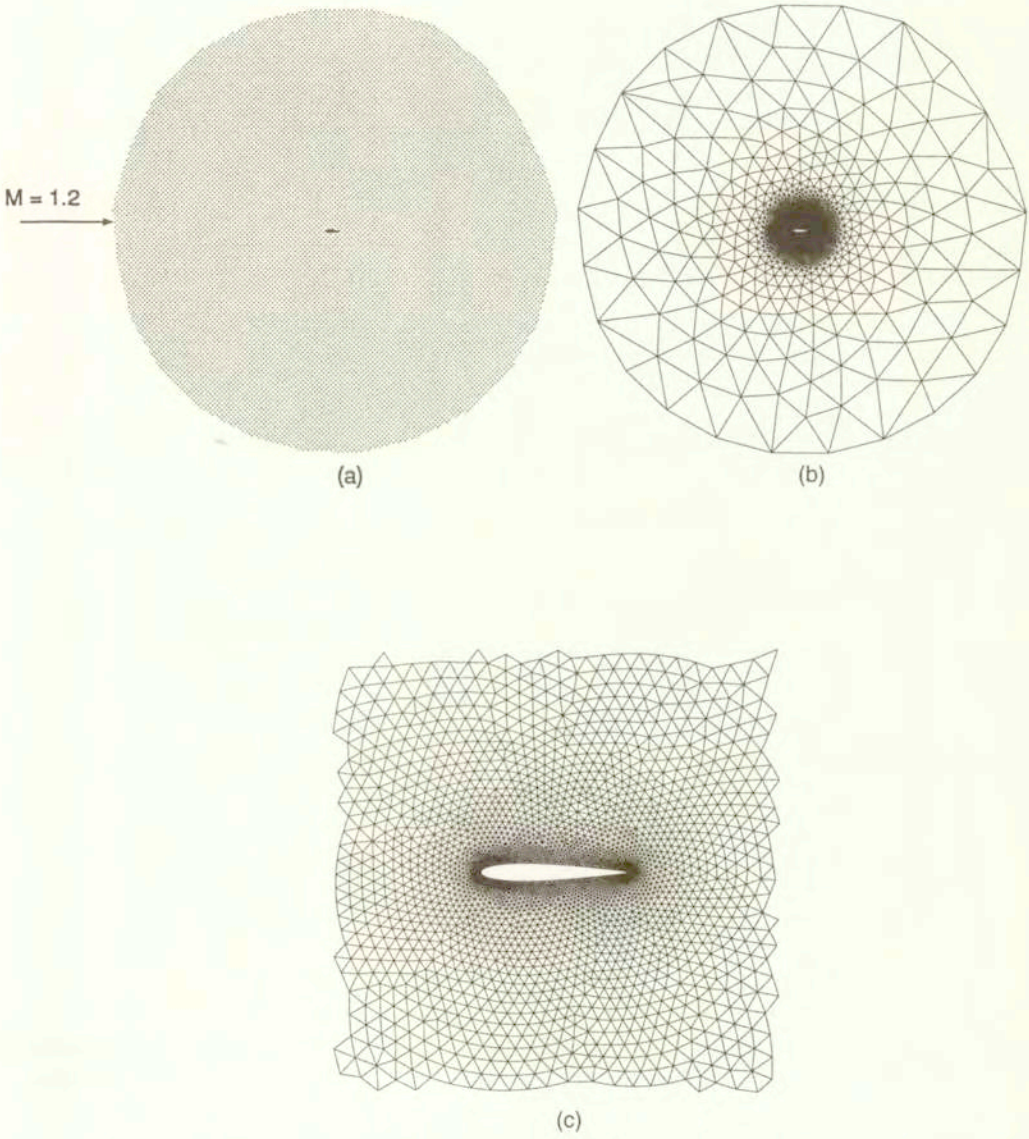


FIG. 2. Supersonic inviscid flow past NACA0012 aerofoil, $M = 1.2$, $\alpha = 0°$. (a) Domain (b) Linear traingular finite element mesh, Nodes: 3753, Elements: 7351 (c) Element distribution near aerofoil surface.

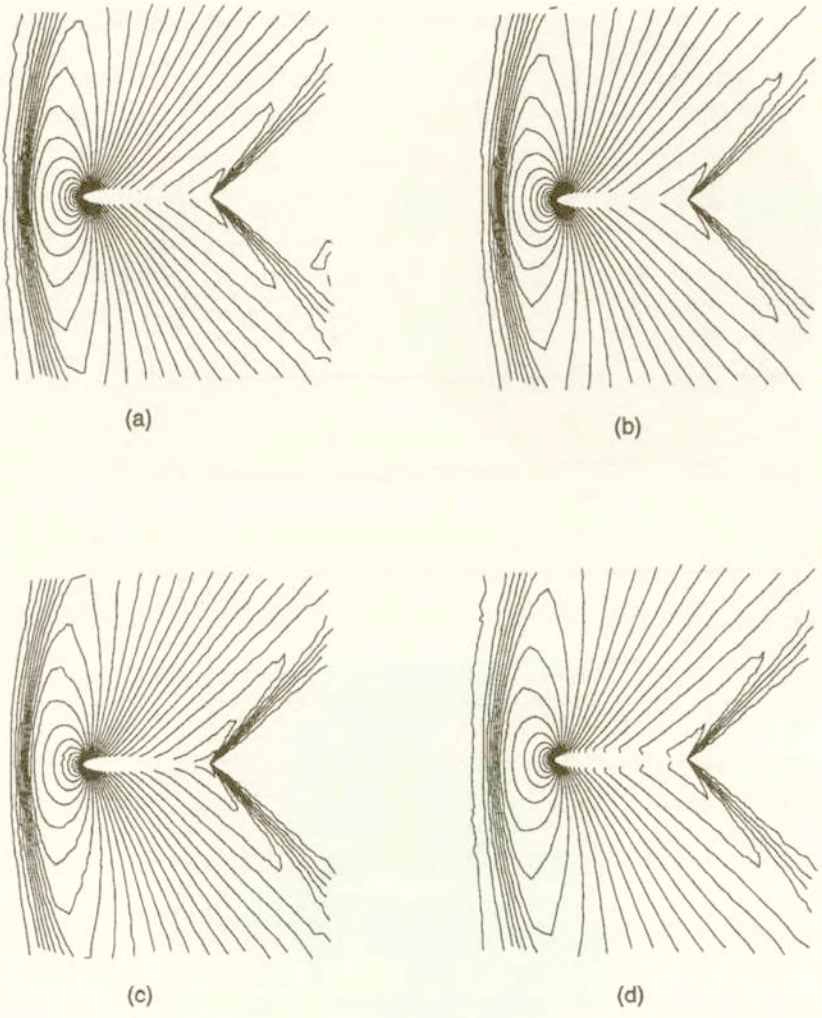


FIG. 3. Supersonic inviscid flow past NACA0012 aerofoil, $M = 1.2$, $\alpha = 0°$, contours of different variables. (a) Density (b) Pressure (c) Temperature (d) Mach number.

Example 2. Hypersonic flow past a cylinder.

Figure 5 shows the domain and quadrilateral mesh generated to study hypersonic flow past a quarter-cylinder. The inlet Mach number is taken as 6. At the inlet, velocity and density are prescribed.

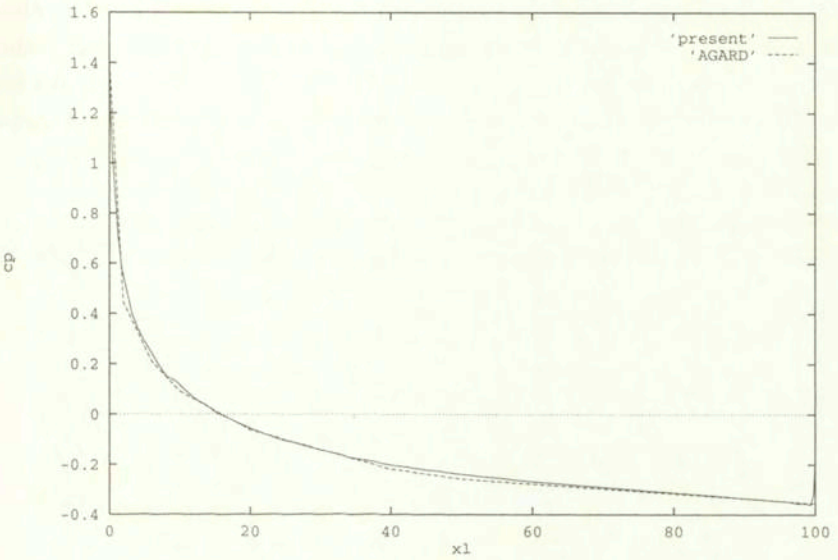


FIG. 4. Supersonic inviscid flow past NACA0012 aerofoil, $M = 1.2$, $\alpha = 0°$. Comparison of coefficient of pressure distribution on the surface of aerofoil.

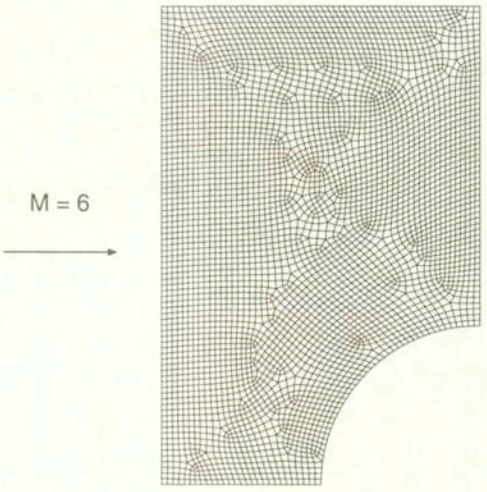


FIG. 5. Hypersonic inviscid flow past a quarter-cylinder, $M = 6$. Linear quadrilateral finite element mesh, Nodes: 4421, Elements: 4286.

In Fig. 6, the contours of all important variables are given. Even at a high Mach number of 6, we have found no oscillations. To smooth out the shocks, we have used some appropriate shock capturing viscosity [28]. Figure 7 slows the Mach number distribution along the bottom line of the domain. As seen, the shock is reasonably sharp even though adaptivity in any form is not applied.

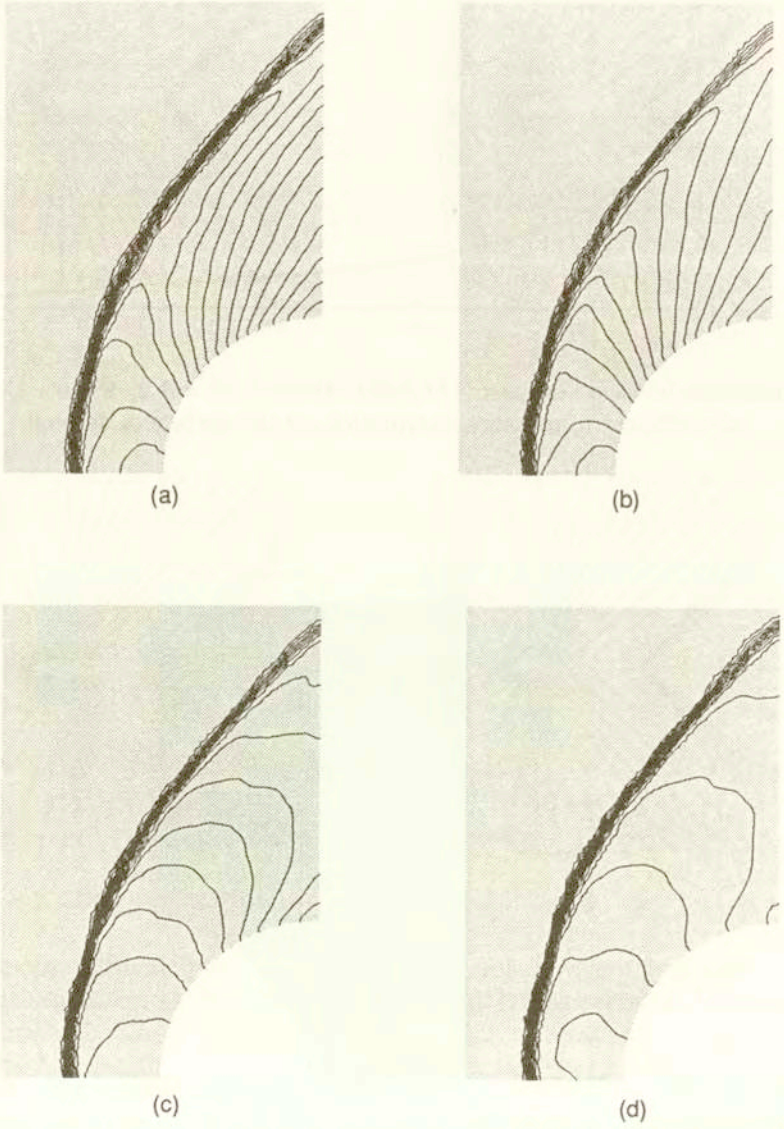


FIG. 6. Hypersonic inviscid flow past a quarter-cylinder, $M = 6$, contours of different variables. (a) Density (b) Pressure (c) Temperature (d) Mach number.

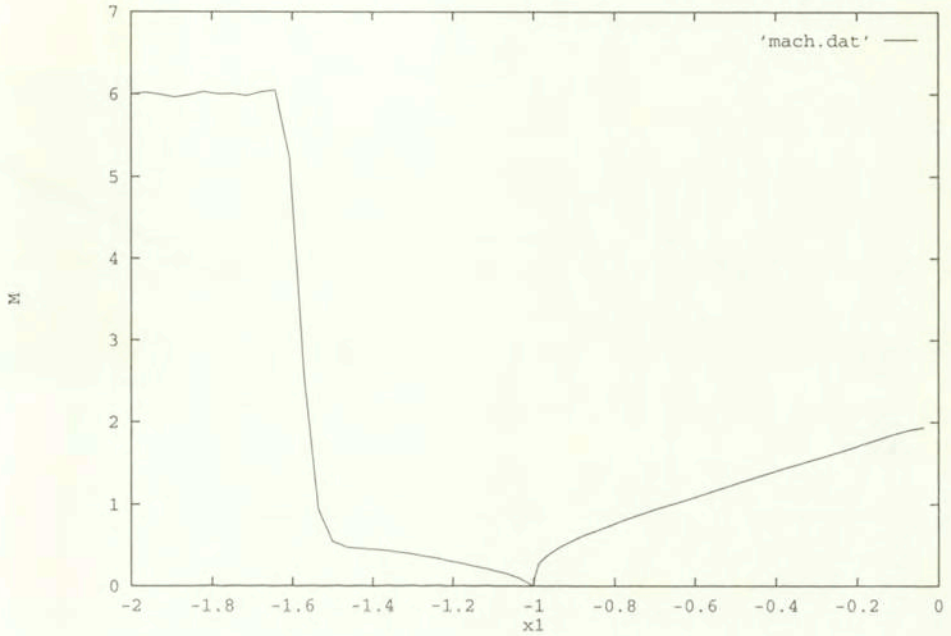


FIG. 7. Hypersonic inviscid flow past a quarter-cylinder, $M = 6$. Mach number distribution along bottom line.

Example 3. Viscous compressible flow past a plate ($M = 3.0$, $Re = 1000$) (Carter).

In this example, a fully explicit scheme is used. Here, the Mach number at the inflow is 3.0 and the inlet Reynolds number based on the length of the plate is 1000. This problem is also known as the Carter problem. The temperature of the plate is assumed to be constant and equal to the stagnation temperature given by

$$T_s = T_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2\right). \tag{7.1}$$

The temperature dependence of viscosity is accounted for through the Sutherland's law

$$\frac{\mu}{\mu_r} = \frac{T_r + S_o}{T + S_o} \left(\frac{T}{T_r}\right)^{1.5}, \tag{7.2}$$

where S_o is Sutherland's constant and is equal to 198.6° Rankine.

A uniform rectangular mesh with 8281 nodes and 8100 elements is used in this computation. The results obtained are shown in Figs. 8 and 9. It is seen that the contours are smooth and shocks are in the proper locations. The pressure distribution along the plate and density distribution at the outlet agree excellently

(Fig. 9) with the Carter [29] results (though the latter show some unatural oscillations near the leading edge).

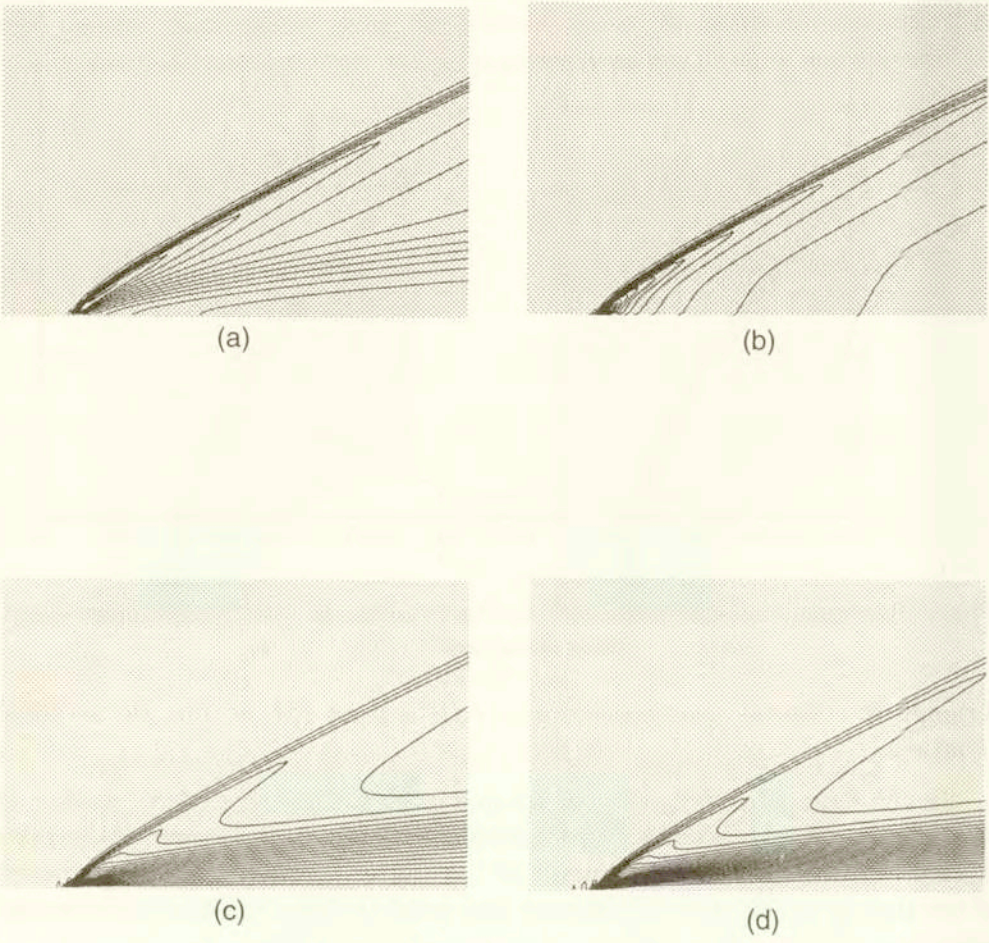
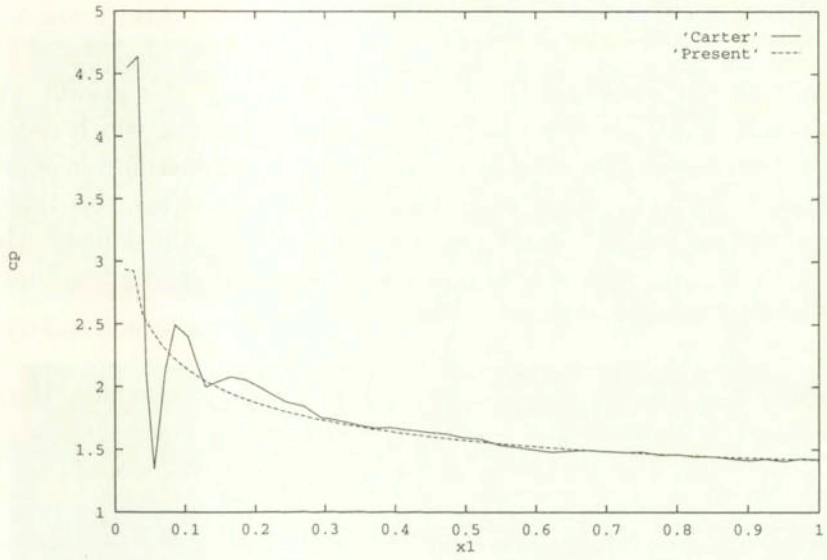


FIG. 8. Supersonic viscous flow past a flat plate, $M = 3$, $Re = 1000$, contours of different variables. (a) Density (b) Pressure (c) Temperature (d) Mach number

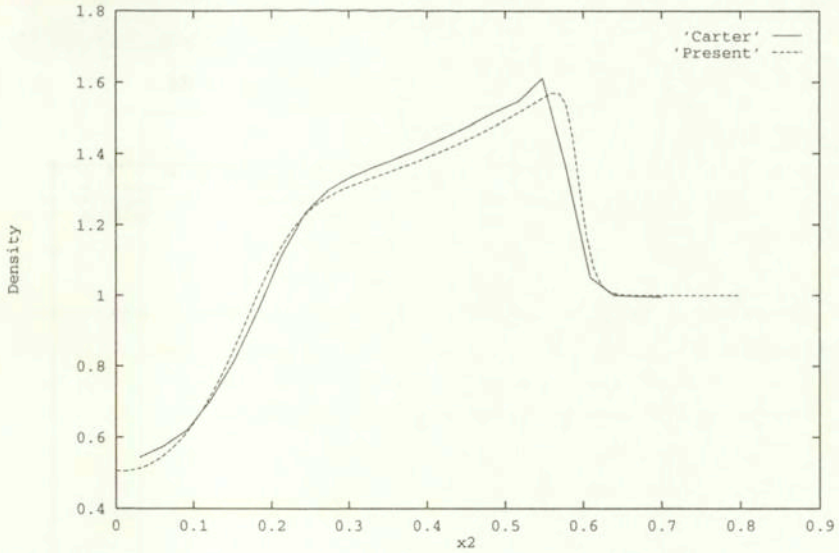
7.2. Semi-implicit procedure

Example 1. High Reynolds number flow in a lid-driven cavity, $M = 0$.

The first example is a fully viscous incompressible flow ($M = 0$) in a lid-driven cavity. This is a well known test case used by many authors. A high Reynolds number of 5000 is used in this study. Figure 10 shows the details of rectangular mesh used and comparison with the benchmark solution [30]. The agreement is excellent.



(a)



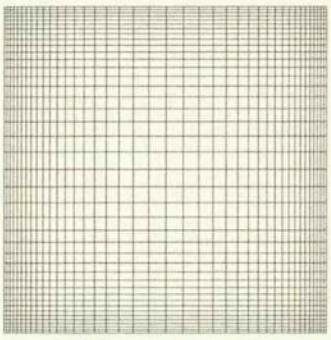
(b)

FIG. 9. Supersonic viscous flow past a flat plate, $M = 3$, $Re = 1000$; (a) Comparison of pressure distribution on the plate surface with CARTER [29]; (b) Comparison density distribution at exit with CARTER [29].

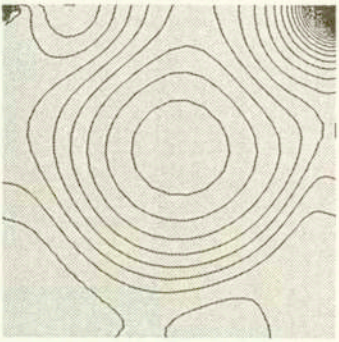
In Fig. 11, we give two adapted meshes using two different forms of adaptive procedure [31, 32] to solve the cavity problem for the same Reynolds number of 5000. It is seen that the results agree excellently with the benchmark.

7.3. Tests on incompressible stabilization

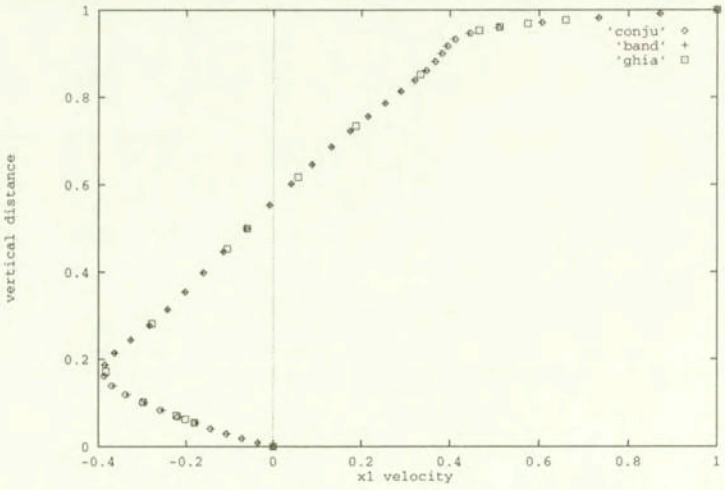
In Sec. 4.4, we mentioned two different time steps, the so-called internal and external time steps. The external time step is the one which needs to be calculated from the explicit step of the algorithm and sometimes it is necessary to use a safety factor to reduce it. In incompressible problems, this time step is taken as the minimum among the values calculated from the domain. However, the internal time step need not be equal to the external one and can improve the incompressible solution.



(a)

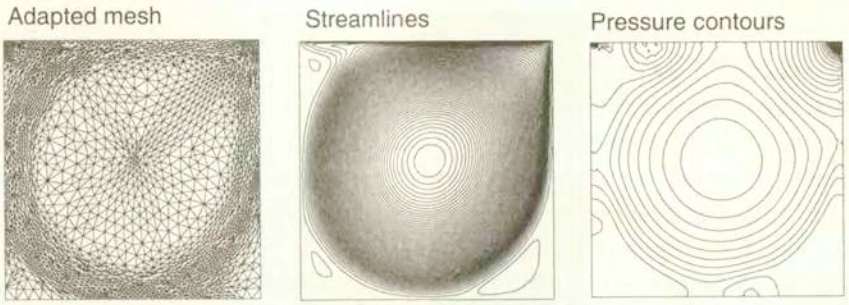


(b)

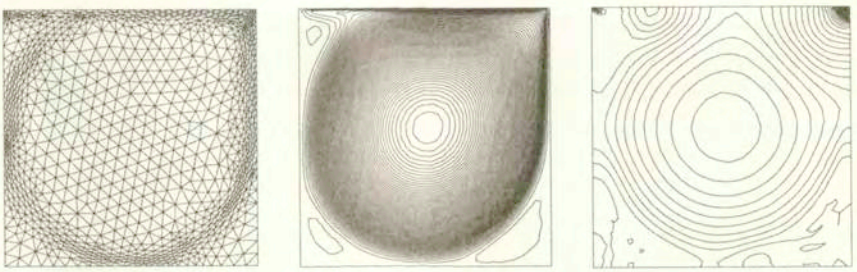


(c)

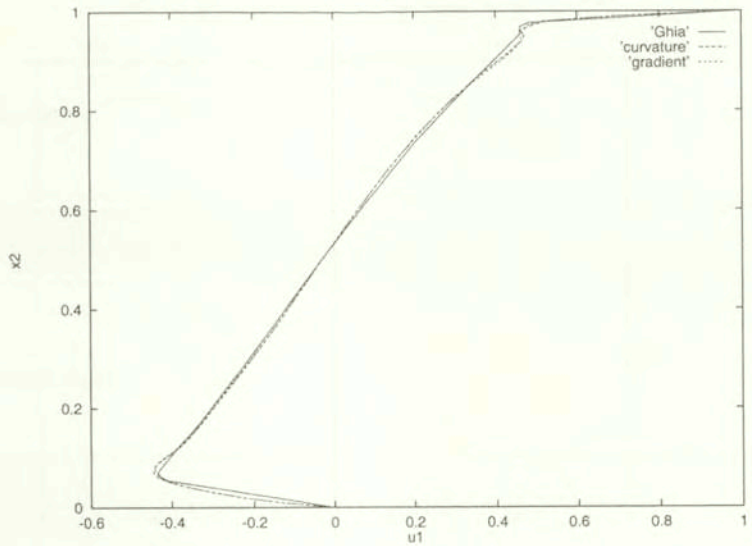
FIG. 10. Incompressible viscous flow in a lid-driven cavity, $Re = 1000$. (a) Linear quadrilateral mesh, Nodes: 1681, elements: 1600 (b) Pressure contours (c) Comparison of u_1 velocity distribution at mid-height with GHIA *et al.* [30].



(a) Curvature based procedure, Nodes: 2389, Elements: 4599



(b) Gradient based procedure, Nodes: 1034, Elements: 1962



(c) Comparison of velocity at mid-vertical plane

FIG. 11. Incompressible viscous flow in a lid-driven cavity, $Re = 5000$, (a) Curvature based adaptive procedure (b) Gradient based adaptive procedure (c) Comparison of u_1 velocity distribution at mid-height with GHIA *et al.* [30].

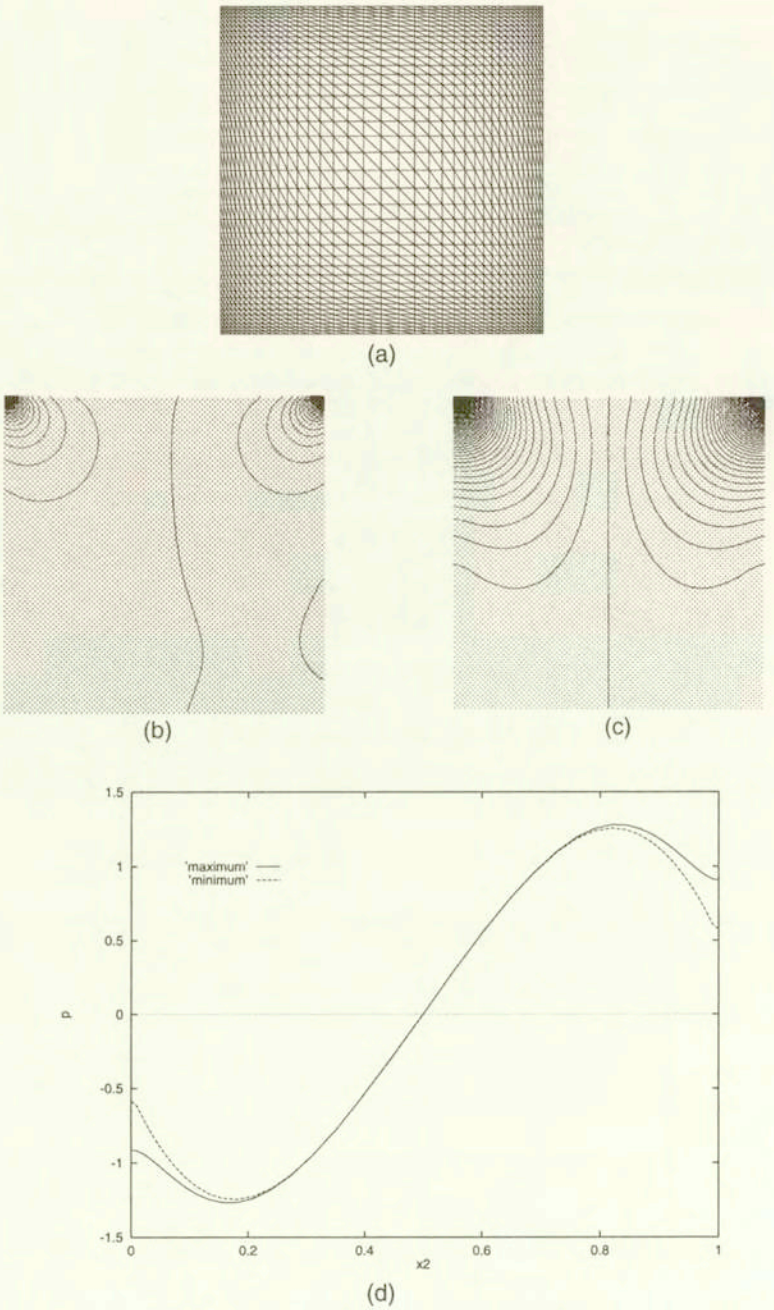
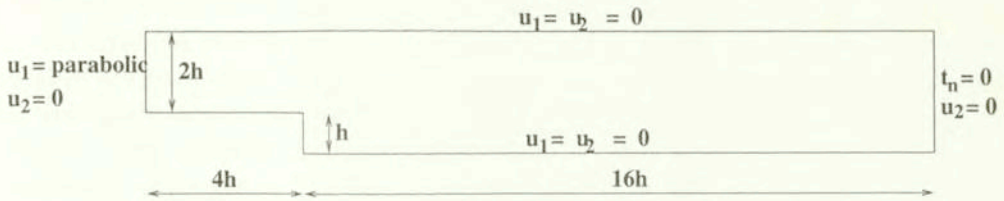
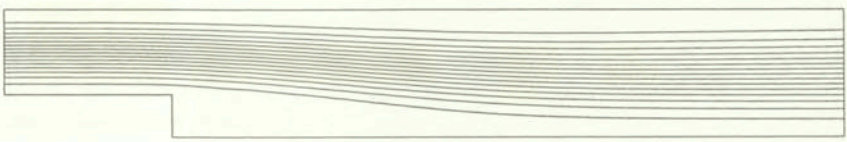


FIG. 12. Exercise on the incompressible stability of the CBS procedure on a Stokes flow problem. (a) Non-uniform triangular mesh (b) Pressure contours, $\Delta t_{\text{ext}} = \Delta t_{\text{crit}} = \Delta t_{\text{int}}$ (c) Pressure contours, $\Delta t_{\text{ext}} = \Delta t_{\text{crit}}$; $\Delta t_{\text{int}} = \Delta t_{\text{max}}$. (d) Comparison of pressure distribution across the cavity at mid-height.

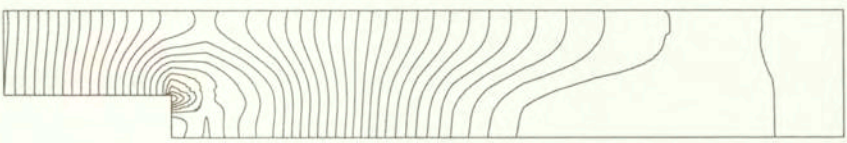
In Fig. 12, we show an example of the Stokes problem in a lid-driven cavity. The mesh used is non-uniform as shown in Fig. 12(a). Figures 12(b) and (c) show two different solutions obtained for the same problem with the same external time step and different internal time steps. In the first case (Fig. 12(b)), the internal time step is identically equal to the external time step, i.e. both the time steps are equal to the minimum value calculated from the problem domain. However, in the second case (Fig. 12(c)), the external time step is same as in case one but the internal time step is the maximum time step value calculated from the whole domain. As it may be seen, the solutions are different and higher internal time steps act as stabilizing factors. Figure 12(d) compares the pressure distribution along the mid-height of the cavity. It is seen that higher internal time step gives an improved solution and here the values are almost equal to the one given by the Galerkin Least Squares approach (GLS) with an optimum GLS factor [23]. A full derivation on the matter of internal and external time steps can be found in Ref. [9].



(a) Geometry

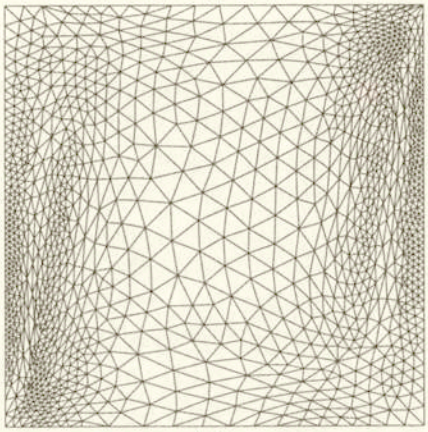


(b) Streamlines



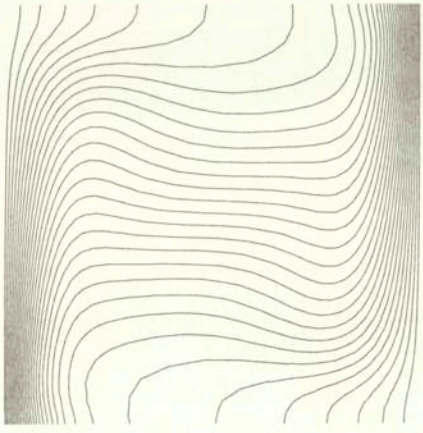
(c) Pressure contours

FIG. 13. Exercise on exit boundary conditions. Incompressible, viscous flow pas a backward facing step, $Re = 100$ (a) Geometry and boundary conditions (b) Stream lines (c) Pressure contours.



(a)

(b)



(c)

FIG. 14. Buoyancy driven flow in a square cavity, $Ra = 10^5$. (a) Final adapted mesh (b) Stream lines (c) Isotherms.

7.4. Boundary conditions

The problem considered here is the standard backward facing step. The domain and various boundary conditions are shown in Fig. 13(a). The inlet velocity profile is parabolic and the Reynolds number is 100.

Figures 13(b) and 13(c) show the results on exit boundary conditions mentioned earlier. As seen, the results are excellent and agree well with a longer domain results [8].

7.5. Buoyancy driven flows

The domain considered here is a two-dimensional square shape. The buoyancy flow is initiated in the cavity by differentially heating the vertical walls. Both the vertical walls are kept at two different temperatures; the left wall temperature is higher than that on the right one. Bottom and top walls are insulated. Non-slip boundary conditions are assumed on all four sides of the cavity.

An example on buoyancy-driven convection is given in Fig. 14. The adopted meshes are generated by the procedure explained in Ref. [31]. The solution given in this figure are extraordinarily smooth and symmetric for a Rayleigh number of 10^5. The quantitative solutions obtained (Nusselt number = 4.519) agree excellently with the available bench-mark solutions [33]. The difference is less than 0.5%.

8. Concluding remarks

We hope that in the present paper we have fully explained the logical background and the excellent performance of the CBS algorithm. The development of the algorithm has taken some time and the present authors would like to acknowledge the earlier works which pointed the way to the final algorithm. In this paper we addressed several new aspects of the algorithm including the incompressible stabilization procedures. However, further tests are needed to fully understand the nature in which the stabilization introduced affects the compressible and incompressible flow problems.

Acknowledgements

This research has been partially supported by NASA grant NAGW/2127, AMES Control Number 90 – 144.

References

1. O. C. ZIENKIEWICZ and R. CODINA, *Search for a general fluid mechanics algorithm*, Frontiers of Computational Fluid Dynamics, D. A. CAUGHEY and M. M. HAFEZ [Eds.], J. Wiley & Sons, 101–113, 1995.
2. O. C. ZIENKIEWICZ and R. CODINA, *A general algorithm for compressible and incompressible flow, Part I. The split characteristic based scheme*, Int. J. Num. Meth. Fluids, **20**, 869–885, 1995.
3. O. C. ZIENKIEWICZ, B. V. K. SATYA SAI, K. MORGAN, R. CODINA and M. VÁZQUEZ, *A general algorithm for compressible and incompressible flow, Part II. Tests on the explicit form*, Int. J. Num. Meth. Fluids, **20**, 887–913, 1995.
4. R. CODINA, M. VÁZQUEZ and O. C. ZIENKIEWICZ, *General algorithm for compressible and incompressible flows, Part III. A semi-implicit form*, Int. J. Num. Meth. Fluids, **27**, 13–32, 1998.
5. R. CODINA, M. VÁZQUEZ and O. C. ZIENKIEWICZ, *A fractional step method for the solution of compressible Navier-Stokes equations*, Frontiers of computational fluid dynamics, M. HAFEZ and K. OSHIMA [Eds.], **1**, 331–347, World Scientific Pub. co.Ltd, 1998.
6. B. V. K. SATYA SAI, O. C. ZIENKIEWICZ, M. T. MANZARI, P. R. M. LYRA and K. MORGAN, *General purpose Vs Special Algorithms for high speed flows with shocks*, Int. J. Num. Meth. Fluids, **27**, 57–80, 1998.
7. O. C. ZIENKIEWICZ, P. NITHIARASU, R. CODINA, M. VÁZQUEZ and P. ORTIZ, *An Efficient and Accurate Algorithm for Fluid Mechanics Problems. The Characteristic Based Split (CBS) Algorithm*, Int. J. Num. Meth. Fluids, **31**, 1, 359–392, 1999.
8. N. MASSAROATTI, P. NITHIARASU and O. C. ZIENKIEWICZ, *Characteristic-Based-Split (CBS) Algorithm for Incompressible Flow Problems with Heat Transfer*, International Journal of Numerical Methods for Heat and Fluid Flow, **8**, 8, 969–990, 1998.
9. P. NITHIARASU, and O. C. ZIENKIEWICZ, *On Stabilization of the CBS Algorithm. Internal and External Time Steps*, Int. J. Num. Meth. Engng., **48**, 875–880, 2000.
10. O. C. ZIENKIEWICZ, J. SZMELTER and J. PERAIRE, *Compressible and incompressible flow: An algorithm for all seasons*, Comp. Meth. Appl. Mech. Engng., **78**, 105–121, 1990.
11. O. C. ZIENKIEWICZ, *Explicit or semi-explicit general algorithm for compressible and incompressible flows with equal finite element interpolation*, Report 90/5, Chalmers University of Technology, 1990.
12. O. C. ZIENKIEWICZ and J. WU, *A general explicit of semi-explicit algorithm for compressible and incompressible flows*, Int. J. Num. Meth. Engng., **35**, 457–479, 1992.
13. P. D. LAX and B. WENDROFF, *Systems of conservation laws*, Comm. Pure. Appl. Math., **13**, 217–237, 1960.
14. J. DONEA, *A Taylor-Galerkin method for convective transport problems*, Int. J. Num. Meth. Engng., **20**, 101–119, 1984.
15. O. C. ZIENKIEWICZ, R. LÖHNER, K. MORGAN and J. PERAIRE, *High speed compressible flow and other advection dominated problems of fluid mechanics*, Finite elements in fluids, **6**, Ch. 2, 41–88, R. H. GALLAGHER, G. F. CAREY, J. T. ODEN and O. C. ZIENKIEWICZ [Eds.], J. Wiley & Sons, 1985.
16. R. LÖHNER, K. MORGAN and O. C. ZIENKIEWICZ, *The solution of non-linear hyperbolic equation system by the finite element method*, Int. J. Num. Meth. Fluids, **4**, 1043–1063, 1984.

17. H. LAVAL and L. QUARTAPELLE, *A fractional step Taylor-Galerkin method for unsteady incompressible flows*, Int. J. Num. Meth. Fluids, **11**, 501–513, 1990.

18. C. B. JIANG, M. KAWAHARA and K. KASHIYAMA, *A Taylor-Galerkin based finite element method for turbulent flows*, Fluid Dynamic Research, **9**, 165–178, 1992.

19. L. DEMKOWICZ, J. T. ODEN, W. RACHOWICZ and O. HARDY, *An h-p Taylor-Galerkin finite element method for compressible Euler equations*, Comp. Meth. Appl. Mech. Engng., **88**, 363–396, 1991.

20. A. J. CHORIN, *Numerical solution of Navier Stokes equations*, Math. Comp., **22**, 745–762, 1968.

21. R. A. ADEY and C. A. BREBBIA, *Finite element solution of effluent dispersion*, Numerical Methods in Fluid Mechanics, C. A. BREBBIA and J. J. CONNOR [Eds.], 325–354, Pentech Press, 1974.

22. O. PRIONNEAU, *On the transport diffusion algorithm to the Navier-Stokes equation*, Num. Math., **38**, 309–332, 1982.

23. O. C. ZIENKIEWICZ and R. L. TAYLOR, *The finite element method*, **3** Fluid Dynamics, 5^{th} ed., Butterworth, London 2000.

24. E. OÑATE, *Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems*, Comp. Meth. App. Mech. Engng., **151**, 233–265, 1998.

25. E. OÑATE and M. MANZÁN, *A general procedure for deriving stabilized space time finite element methods for advective diffusive equations*, Int. J. Num. Meth. Fluids, **31**, 203–221, 1999.

26. R. CODINA, *Stability analysis of forward Euler scheme for the convection diffusion equation using the SUPG formulation in space*, Int. J. Num. Meth. Engng., **36**, 1445–1464, 1993.

27. T. H. PULLIAM and J. T. BARTON, *Euler computations of AGARD working group 07 aerofoil test cases*, AIAA 23rd Aerospace Sciences Meeting, Jan 14 – 17, Reno, Nevada 1985.

28. P. NITHIARASU, O. C. ZIENKIEWICZ, B. V. K. S. SAI, K. MORGAN, R. CODINA, M. VÁZQUEZ, *Shock capturing viscosities for the general fluid mechanics algorithm*, Int. J. Num. Meth. Fluids, **28**, 1325–1353, 1998.

29. J. E. CARTER, *Numerical solutions of the Navier-Stokes equations for the supersonic laminar flow over a two-dimensional compression corner*, NASA TR-R-385, 1972.

30. U. GHIA, K. N. GHIA and C. T. SHIN, *High-Re solution for incompressible flow using the Navier-Stokes equations and multigrid method*, J. Comp. Phys., **48**, 387–411, 1982.

31. P. NITHIARASU and O. C. ZIENKIEWICZ, *Adaptive mesh generation for fluid mechanics problems*, Int. J. Num. Meth. Engng., **47**, 629–662, 2000.

32. J. PERAIRE, M. VAHDATI, K. MORGAN and O. C. ZIENKIEWICZ, *Adaptive remeshing for compressible flow computations*, J. Comp. Phys., **72**, 449–466, 1987.

33. G. de VAHL DAVIS, *Natural convection of air in a square cavity: a bench mark numerical solution*, Int. J. Num. Meth. Fluids, **3**, 249–264, 1983.

Received February 21, 2000; revised version July 4, 2000.