

A COMPARATIVE ASSESSMENT WITHIN A MULTIGRID ENVIRONMENT OF SEGREGATED PRESSURE-BASED ALGORITHMS FOR FLUID FLOW AT ALL SPEEDS

M. Darwish , D. Asmar & F. Moukalled

To cite this article: M. Darwish , D. Asmar & F. Moukalled (2004) A COMPARATIVE ASSESSMENT WITHIN A MULTIGRID ENVIRONMENT OF SEGREGATED PRESSURE-BASED ALGORITHMS FOR FLUID FLOW AT ALL SPEEDS, Numerical Heat Transfer, Part B: Fundamentals, 45:1, 49-74, DOI: [10.1080/1040779049025487](https://doi.org/10.1080/1040779049025487)

To link to this article: <http://dx.doi.org/10.1080/1040779049025487>



Published online: 17 Aug 2010.



Submit your article to this journal [↗](#)



Article views: 69



View related articles [↗](#)



Citing articles: 8 View citing articles [↗](#)

A COMPARATIVE ASSESSMENT WITHIN A MULTIGRID ENVIRONMENT OF SEGREGATED PRESSURE-BASED ALGORITHMS FOR FLUID FLOW AT ALL SPEEDS

M. Darwish, D. Asmar, and F. Moukalled

*Faculty of Engineering & Architecture, Mechanical Engineering Department,
American University of Beirut, Beirut, Lebanon*

This article deals with the evaluation of six segregated high-resolution pressure-based algorithms, which extend the SIMPLE, SIMPLEC, PISO, SIMPLEX, SIMPLEST, and PRIME algorithms, originally developed for incompressible flow, to compressible flow simulations. The algorithms are implemented within a single grid, a prolongation grid, and a full multigrid method and their performance assessed by solving problems in the subsonic, transonic, supersonic, and hypersonic regimes. This study clearly demonstrates that all algorithms are capable of predicting fluid flow at all speeds and qualify as efficient smoothers in multigrid calculations. In terms of CPU efficiency, there is no global and consistent superiority of any algorithm over the others, even though PRIME and SIMPLECT are generally the most expensive for inviscid flow problems. Moreover, these two algorithms are found to be very unstable in most of the cases tested, requiring considerable upwind bleeding (up to 50%) of the high-resolution scheme to promote convergence. The most stable algorithms are SIMPLEC and SIMPLEX. Moreover, the reduction in computational effort associated with the prolongation grid method reveals the importance of initial guess in segregated solvers. The most efficient method is found to be the full multigrid method, which resulted in a convergence acceleration ratio, in comparison with the single grid method, as high as 18.4.

INTRODUCTION

The multigrid method, initially developed for the solution of elliptic partial differential equations [1–3] for which near-optimum convergence characteristics have been demonstrated (i.e., computational time directly proportional to grid size [4, 5]), has gained wide acceptance within the computational fluid dynamics (CFD) community and has become an essential acceleration technique for solving industrial-type flow problems [6–13]. Multigrid strategies for the incompressible Navier-Stokes equations using pressure-based methods, such as SIMPLE [14], as smoothers have shown substantial increase in convergence rate and improvement in overall robustness for both staggered [15, 16] and collocated [17–20] grids. In these

Received 23 April 2003; accepted 11 June 2003.

The financial support provided by the University Research Board at the American University of Beirut is gratefully acknowledged.

Address correspondence to M. Darwish, American University of Beirut, Mechanical Engineering Department, P.O. Box 11-0236, Beirut, Lebanon. E-mail: darwish@aub.edu.lb

NOMENCLATURE

$a_p^\phi, a_E^\phi, \dots$	Coefficients in the discretized equation for ϕ	$\mathbf{v}$	velocity vector ($= u\mathbf{i} + v\mathbf{j}$)
b_p^ϕ	source term in the discretized equation for ϕ	β	thermal expansion coefficient
C_p	Coefficient equals to $1/RT$	Γ^ϕ	diffusion coefficient for ϕ
$\mathbf{d}$	distance vector between a coarse and a fine grid point	$\Delta[\phi]$	Δ operator
$\mathbf{D}[\phi]$	$\mathbf{D}$ operator	μ	viscosity
h	enthalpy	ρ	density
$H[\phi]$	H operator	ϕ	dependent variable
$\mathbf{H}[\phi]$	vector form of the H operator	ϕ_f	scalar value at cell face f
I_h^H	restriction operator	Φ	dissipation term in energy equation
I_H^h	prolongation operator	Ω	cell volume
P	pressure	Subscripts	
Q^ϕ	source term in the conservation equation for ϕ	f	refers to control-volume face f
r_h	residuals on grid size h	NB	refers to neighbors of grid point P
R	gas constant	P	refers to the grid point P
$\mathbf{S}_f$	surface vector	Superscripts	
T	temperature	ϕ	refers to dependent variable
u, v	velocity components in the x and y directions	$*$	refers to the updated value at the current iteration
		$'$	refers to correction field

algorithms, a pressure-correction equation is used to enforce mass conservation by correcting both the pressure and the momentum satisfying velocity field. Moreover, being elliptic in nature, the use of a multigrid technique in solving the incompressible pressure-correction equation [21] is expected to decrease the computational time in comparison with a single-grid method. This elliptic form is very different from the hyperbolic form assumed by the equation when derived for all-speed compressible flows [22, 23]. Nevertheless, multigrid methods have been successfully employed in the simulation of high-speed compressible flows, with density-based algorithms as smoothers [24, 25].

In density-based methods, the continuity equation acts as an equation for density, while pressure is obtained from the energy and state equations. This means that for low-Mach-number flows, as small disturbances in density may result in large variations in the pressure field, density-based algorithms become unstable and their convergence rate greatly diminishes. Despite the extension of this class of algorithms to predicting incompressible flows through the use of the so-called pseudo or artificial compressibility techniques [26–28], the difficulties encountered in efficiently avoiding the stiff solution matrices generated by these methods have led to the extension of pressure-based algorithms to this class of flows [22, 23, 29–33], thus encompassing the entire subsonic-to-hypersonic spectrum. In pressure-based all-speed flow algorithms the pressure-correction equation contains an advection term whose effect dominates the other terms in the equation for supersonic and hypersonic flows, while for transonic flows its effect increases with the Mach number. This subtle change in the nature of the equation from elliptic to hyperbolic does affect the performance of multigrid methods.

The aim of this article is to evaluate the performance of a number of high-resolution (HR) pressure-based all-speed flow algorithms as smoothers in a multigrid framework. In addition to the all-speed flow algorithms based on SIMPLE [14], SIMPLEC [34], PISO [35], and SIMPLEST [37], two all-speed flow algorithms, based on SIMPLEX [36] and PRIME [21], are also implemented and tested for the first time in solving high-Mach-number flows. To this end a set of problems ranging from subsonic to hypersonic flows is solved on increasingly finer grids using the above algorithms with a single-grid (SG), a prolongation grid (PG), and a full multigrid (FMG) methodology.

In what follows the governing equations for compressible flow are presented and their discretization outlined. This is followed by a brief description of all-speed pressure-based algorithms and their treatment within a multigrid framework. Finally, test problems are introduced and results discussed.

THE GOVERNING EQUATIONS

The equations of motion for steady compressible viscous flow in conservative form are:

Conservation of mass:

$$\nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

Conservation of momentum:

$$\nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla P + \nabla \cdot \tau \quad (2)$$

where

$$\tau = \mu (\nabla \mathbf{v} + \nabla \mathbf{v}^T) - \frac{2}{3} \mu (\nabla \cdot \mathbf{v}) \mathbf{I} \quad (3)$$

Conservation of energy:

$$\nabla \cdot (\rho \mathbf{v} h) = \{ \nabla \cdot (k \nabla T) + \beta T [\nabla \cdot (P \mathbf{v}) - P \nabla \cdot (\mathbf{v})] + \Phi \} \quad (4)$$

where

$$\Phi = \mu \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 - \frac{2}{3} (\nabla \cdot \mathbf{v})^2 \right\} \quad (5)$$

An adequate manipulation of these equations allows their representation in a unified equation of a general scalar variable ϕ as follows:

$$\nabla \cdot (\rho \mathbf{v} \phi) = \nabla \cdot (\Gamma^\phi \nabla \phi) + Q^\phi \quad (6)$$

The equation of state for an ideal gas is written as

$$\rho = \frac{P}{RT} = C_p P \quad (7)$$

THE DISCRETIZATION PROCEDURE

Discretization of the Transport Equations

The discretization of the momentum and energy equations using the finite-volume method is detailed in [22] and therefore only an outline of the procedure is given here. The first step consists of integrating the general equation over a differential control volume to yield

$$\int_{\Omega} \nabla \cdot (\rho \mathbf{v} \phi) d\Omega = \int_{\Omega} \nabla \cdot (\Gamma^{\phi} \nabla \phi) d\Omega + \int_{\Omega} Q^{\phi} d\Omega \quad (8)$$

Using the divergence theorem, the convection and diffusion volume integrals are transformed into surface integrals to give

$$\oint_{\partial\Omega} (\rho \mathbf{v} \phi) \cdot \mathbf{dS} = \oint_{\partial\Omega} (\Gamma^{\phi} \nabla \phi) \cdot \mathbf{dS} + \int_{\Omega} Q^{\phi} d\Omega \quad (9)$$

Employing the midpoint integration rule, the surface integral about the cell faces is set equal to a summation of fluxes at the cell faces centers, while the volume integral is evaluated at the cell center. Thus, Eq. (9) becomes

$$\sum_{\sim \text{faces}} \{(\rho \mathbf{v} \cdot \mathbf{S})\phi - \Gamma^{\phi} \nabla \phi \cdot \mathbf{S}\} = Q\Omega \quad (10)$$

These fluxes are then related to the values at the cell centers and their neighboring nodes by using a suitable interpolation profile in a local coordinate direction (a high-resolution scheme applied within the context of the NVSF methodology [38]). Substitution of the interpolation profiles into Eq. (10) gives the final form of the discretized equation as

$$a_P^{\phi} \phi_P + \sum_{\text{NB}(P)} a_{\text{NB}}^{\phi} \phi_{\text{NB}} = b_P^{\phi} \quad (11)$$

where ϕ_P is the value of ϕ at the main grid point P , NB designates the neighbors of P (i.e., north, south, east, and west), and b_P^{ϕ} includes the original source term, the deferred-correction treatment of the high-resolution advection fluxes, and the nonorthogonal diffusion fluxes. For later use Eq. (11) can either be written as

$$\phi_P = H[\phi] + B \quad (12)$$

where

$$H[\phi] = \frac{-\sum_{\text{NB}(P)} a_{\text{NB}}^{\phi} \phi_{\text{NB}}}{a_P} \quad B = \frac{b_P}{a_P} \quad (13)$$

or, using matrix notation, as

$$a(\phi)\phi = b(\phi) \quad (14)$$

In a similar manner, the algebraic relation for the momentum equation can be written as

$$\mathbf{v} = \mathbf{H}[\mathbf{v}] + \mathbf{B} - \mathbf{D}\nabla P \quad (15)$$

where the pressure gradient term is treated explicitly and separated from other sources, and $\mathbf{D}$ is

$$\mathbf{D} = \begin{bmatrix} \Omega/a_p^u & 0 \\ 0 & \Omega/a_p^v \end{bmatrix} \quad (16)$$

Discretization of the Continuity Constraint

As is clear from Eqs. (1–4), there is no explicit equation for pressure, thus a pressure or pressure-correction equation has to be derived using a combination of the continuity and momentum equations. The continuity equation for some velocity field $\mathbf{v}^*$ and density field ρ^* is written as

$$\nabla \cdot (\rho^* \mathbf{v}^*) = 0 \quad (17)$$

However, in the above equation the equality is not satisfied because generally the velocity field obtained by solving the momentum equations using some guessed pressure P^* does not satisfy continuity. It is only at convergence that the continuity and momentum equations are satisfied simultaneously. Thus corrections are sought to the velocity, pressure, and density fields to enforce conservation of both mass and momentum equations. Denoting the corrections by $\mathbf{v}'$, P' , and ρ' , one can write

$$\nabla \cdot [(\rho^* + \rho')(\mathbf{v}^* + \mathbf{v}')] = 0 \quad (18)$$

or

$$\nabla \cdot (\rho^* \mathbf{v}' + \mathbf{v}^* \rho' + \rho^* \mathbf{v}^* + \rho' \mathbf{v}') = 0 \quad (19)$$

where now the corrected fields given by

$$\mathbf{v} = \mathbf{v}^* + \mathbf{v}', \quad P = P^* + P', \quad \text{and} \quad \rho = \rho^* + \rho' \quad (20)$$

satisfy the mass conservation equation.

Noting that

$$\rho = \frac{P}{RT} = C_p P \quad (21)$$

the correction for density can be written as

$$\rho' = C_p P' \quad (22)$$

The relation between velocity and pressure corrections can be derived by considering the following discretized form of the momentum satisfying velocity field $\mathbf{v}^*$,

$$\mathbf{v}^* = \mathbf{H}[\mathbf{v}^*] + \mathbf{B} - \mathbf{D}\nabla P^* \quad (23)$$

and the continuity and momentum satisfying fields:

$$\mathbf{v} = \mathbf{H}[\mathbf{v}] + \mathbf{B} - \mathbf{D}\nabla P \quad (24)$$

Subtracting Eq. (23) from Eq. (24) yields:

$$\mathbf{v}' = \mathbf{H}[\mathbf{v}'] - \mathbf{D}\nabla P' \quad (25)$$

Substitution in Eq. (18) gives the pressure-correction equation as:

$$\nabla \cdot \{(\rho^* + \rho')(\mathbf{v}^* + [\mathbf{H}[\mathbf{v}'] + -\mathbf{D}\nabla P'])\} = 0 \quad (26)$$

or, using Eq. (22), as:

$$\nabla \cdot \left(\rho^* \left(\underline{\mathbf{H}[\mathbf{v}']} - \mathbf{D}\nabla P' \right) + (\mathbf{v}^* C_\rho)_f P' \right) = -\nabla \cdot ((\rho^* \mathbf{v}^*) + (\rho' \mathbf{v}')) \quad (27)$$

The segregated algorithms differ in the treatment of the underlined term [22]. Notwithstanding these differences, the pressure-correction equation becomes

$$\nabla \cdot (\mathbf{v}^* C_\rho P') - \nabla \cdot (\rho^* \mathbf{D}\nabla P') = -\nabla \cdot (\rho^* \mathbf{v}^*) - \nabla \cdot (\rho^* \underline{\mathbf{H}[\mathbf{v}']}) - \nabla \cdot (\rho' \mathbf{v}') \quad (28)$$

The first term on the left-hand side of Eq. (28) is an advection term, while the second one is the standard elliptic term of the incompressible pressure-correction equation. The last term on the right-hand side is usually neglected, as it is small in comparison with other terms, while depending on the algorithm used, approximations are introduced to the underscored term, as detailed in [22]. None of these approximations, however, affects the final solution, since at convergence $\mathbf{H}[\mathbf{v}']$ is zero.

The Rhie-Chow Interpolation

The discretization of Eq. (28) yields:

$$\sum_{\sim \text{faces}} (\mathbf{v}^* C_\rho P')_f \mathbf{S}_f + \sum_{\sim \text{faces}} (\rho^* \mathbf{D}\nabla P')_f \mathbf{S}_f = - \sum_{\sim \text{faces}} (\rho^* \mathbf{v}^*) \cdot \mathbf{S}_f - \dots \quad (29)$$

For collocated variables, the interpolation of the velocity to the cell faces in the above equation is not trivial: a simple linear interpolation leads to the well-known checkerboard problem [39]. To overcome this difficulty a special interpolation procedure, known as Rhie-Chow interpolation [40], has to be used to enforce the proper coupling between the pressure and velocities. What the Rhie-Chow interpolation achieves is basically a virtual re-formulation of the momentum Eq. (15) at cell faces, which is accomplished by interpolating the coefficients $\mathbf{H}[\]$, $\mathbf{D}$, and $\mathbf{B}$ at all the cell faces from the cell values either following the momentum-weighted interpolation method (MWIM) [41],

$$\mathbf{v}_f^* = \bar{\mathbf{H}}_f [\mathbf{v}_f^*] + \bar{\mathbf{B}}_f - \bar{\mathbf{D}}_f \nabla P_f^* \quad (30)$$

or the pressure-weighted interpolation method (PWIM) [40],

$$\mathbf{v}_f^* = \bar{\mathbf{v}}_f^* - \bar{\mathbf{D}}_f (\nabla P_f^* - \nabla P_f^*) + (\bar{\mathbf{B}}_f - \mathbf{B}_f) \quad (31)$$

where the overlined coefficients are obtained by linear interpolation from their cell equivalents (the reader is referred to [22] for details).

The Pressure-Correction Equation

Expansion of Eq. (29) yields the final algebraic form of the pressure correction equation as:

$$a_P P'_P + \sum_{NB(P)} a_{NB} P'_{NB} = b'_P \quad (32)$$

The P' field obtained from the solution of this equation is then used in the correction of the velocity, density, and pressure fields.

The overall algorithm for pressure-based methods can now be summarized as follows. The Navier-Stokes equations in their algebraic forms are solved sequentially. First, the momentum equations are assembled and solved by treating the fields of pressure, mass fluxes, and energy as known. Second, the modified continuity equation is employed to calculate a pressure-correction field that is used to force the velocity and density fields to satisfy mass conservation. Then the energy equation is solved and the sequence of events repeated until convergence, which is achieved when the normalized residuals fall below a prescribed value.

THE MULTIGRID METHOD

The procedure described in the previous section applies to a single-grid method whose convergence rate is high during the very first iterations but then stalls thereafter, and the situation gets worse on finer grids. This behavior is attributed to the smoothing property [42] of the algorithm, which is efficient in removing only the high-frequency (short-wavelength) Fourier components of the error. The low-frequency components, which have long wavelength compared to the grid spacing, are not properly resolved. This is clearly noticed when using any of the solvers on progressively denser grids, in which case the convergence rate decreases more rapidly on the finer grids. This behavior is attributed to the fact that on increasingly finer grids, the portion of the low-frequency error components increases. The performance of these algorithms can thus be substantially improved by combination with the multigrid method.

In the multigrid approach the high-frequency errors are initially eliminated on the finest grid, then, when the convergence rate degrades, the process is repeated on a coarser grid, where part of the low-frequency-component errors of the finer grid are transformed into high-frequency error components on the coarser grid that can be efficiently removed. This step is recursively applied on coarser grids, and more of the error components are reduced. Results are then interpolated back from coarser to finer grids. It is clear that for multigrid methods what is needed is a good *smoother*, i.e., a scheme that is capable of efficiently attenuating the high-frequency error. This is contrary to the single-grid method, in which a solver that can attenuate lower-frequency errors is needed, even at the cost of a degradation of efficiency in dealing with high-frequency errors.

Prolongation and Restriction

Any multigrid method relies on two important notions: restriction and prolongation. Restriction, denoted by I_h^H , defines the manner in which the variables are transferred from a fine to a coarse grid, while prolongation, denoted by I_H^h , applies to the transfer of the variable and to the transfer of corrections from a coarse to a fine grid. Another deciding factor is the cycling, i.e., the sequence of prolongation and restriction that occur during the multigrid iterations.

In the restriction step the coarse-grid variables are computed from the fine grid values as:

$$\tilde{\phi}_H = I_h^H(\phi_h) = \frac{1}{4} \sum_{i=1-4} (\phi_{h_i} + \nabla \phi_{h_i} \cdot \mathbf{d}_{h_i \mathbf{H}}) \quad (33)$$

the tilde, $\sim$, indicates a restricted value on the coarse grid.

The prolongation can be applied to the solution fields of the coarse grid, or to the field corrections. The first case is used when the solution of the coarse grid is to be used as an initial guess for the fine grid, while the second case is used when the solution of the coarse grid is to be used to correct the fine-grid fields, i.e., to remove some of the low-frequency errors on the fine grid. For the first case the prolongation takes the form:

$$\phi_{h_i} = I_H^h[\phi_H] = \phi_H + \nabla \phi_H \cdot \mathbf{d}_{Hh_i} \quad (34)$$

while for the second case it takes the form:

$$\phi'_{h_i} = I_H^h[\phi'_H] = \phi'_H + \nabla \phi'_H \cdot \mathbf{d}_{Hh_i} \quad (35)$$

where

$$\phi'_H = \phi_H - \tilde{\phi}_H \quad (36)$$

The Multigrid Correction Process

An exact solution of the discretized equations for variable $\phi = (\mathbf{v}, P, h, \dots)$ satisfies the equation:

$$a_h \phi_h = b_h \quad (37)$$

where the subscript h indicates that the equation is solved on the fine grid h .

After several outer iterations on the whole set of equations (as explained previously), an approximate solution ϕ_h^* is obtained which satisfies Eq. (37) to a residual r_h , defined as:

$$r_h(\phi_h^*) = b_h(\phi_h^*) - a_h(\phi_h^*)\phi_h^* \quad (38)$$

where $b_h(\phi_h^*)$ and $a_h(\phi_h^*)$ are approximations to b_h and a_h based on the approximate solution ϕ_h^* . Subtracting the above two equations, one gets:

$$a_h \phi_h = b_h + \underline{a_h^*(\phi_h^*)\phi_h^* - b_h^*(\phi_h^*) + r_h(\phi_h^*)} \quad (39)$$

Restriction of this equation to the coarse grid leads to the following fundamental coarse-grid, full approximation scheme equation:

$$I_h^H[a_h\phi_h] = I_h^H[b_h] + \underline{I_h^H[a_h^*(\phi_h^*)\phi_h^*] - I_h^H[b_h^*(\phi_h^*)] + I_h^H[r_h(\phi_h^*)]} \quad (40)$$

which also can be written as:

$$a_H\phi_H = b_H + \underline{a_H^*(I_h^H[\phi_h^*])I_h^H[\phi_h^*] - b_H^*(I_h^H[\phi_h^*]) + I_h^H[r_h(\phi_h^*)]} \quad (41)$$

or, after simplification, as:

$$a_H\phi_H = b_H + \underline{a_H^*(\tilde{\phi}_H^*)\tilde{\phi}_H^* - \tilde{b}_H^*(\tilde{\phi}_H^*) + \tilde{r}_H} \quad (42)$$

where now a_H , ϕ_H , and b_H are approximations to the exact values of a_h , ϕ_h , and b_h , respectively, and are computed on the coarse grid H , while $\tilde{\phi}_H^*$, $\tilde{b}_H^*$, and $\tilde{r}_H$ are the restricted vector values of ϕ_h^* , b_h^* , and r_h , respectively, on the coarse grid H .

Once Eq. (42) is solved on the coarse mesh, the correction defined in Eq. (35) is prolonged rather than the solution, and the fine-grid solution is corrected using the formula:

$$\phi_h = \phi_h + I_h^h[\phi_H'] = \phi_h + (\phi_H' + \nabla\phi_H' \cdot \mathbf{d}_{HH}) \quad (43)$$

This is needed to ensure that when $r_h = 0$, i.e., when the fine-grid solution is satisfied, the prolonged coarse-grid correction is also zero.

The two-grid multigrid cycle explained above can be applied recursively through sequences of grid levels. There are different strategies to control the intergrid transfer, which may depend on dynamic criteria such as residual reduction rate or can be initiated following a prescribed numbers of iteration sweeps. In this study the second approach, in which the number of pre- and postsmoothing relaxation sweeps is assigned a priori, is adopted in combination with a V-cycle (Figure 1b). Moreover, as it is relatively much cheaper to compute an initial solution on a coarser mesh and then prolong it to the finer mesh, the current iteration procedure starts on the coarsest grid. After convergence, the solution is prolonged to the next finer grid to be used as an initial approximation. Then a V-cycle (Figure 1b) is initiated on this level. The procedure is repeated consecutively until the finest grid level is reached, where conventional multigrid V-cycle begins. The sequence of operations is as follows.

1. Start with an initial guess on the coarsest grid and iterate to convergence.
2. Prolong solution to the next fine-grid to use it as an initial guess.
3. Apply a number of V-cycles on the fine grid to get the solution.
4. Repeat steps 2 and 3 until a converged solution on the finest grid is obtained.

In addition to the full multigrid strategy, the prolongation grid approach is also tested. This approach differs from the FMG method in that the solution moves in one direction from the coarse to the fine grids with the initial guess on level $n + 1$ obtained by interpolation from the converged solution on level n . As such, the acceleration over the single-grid method obtained with this approach is an indication of the effect of initial guess on convergence.

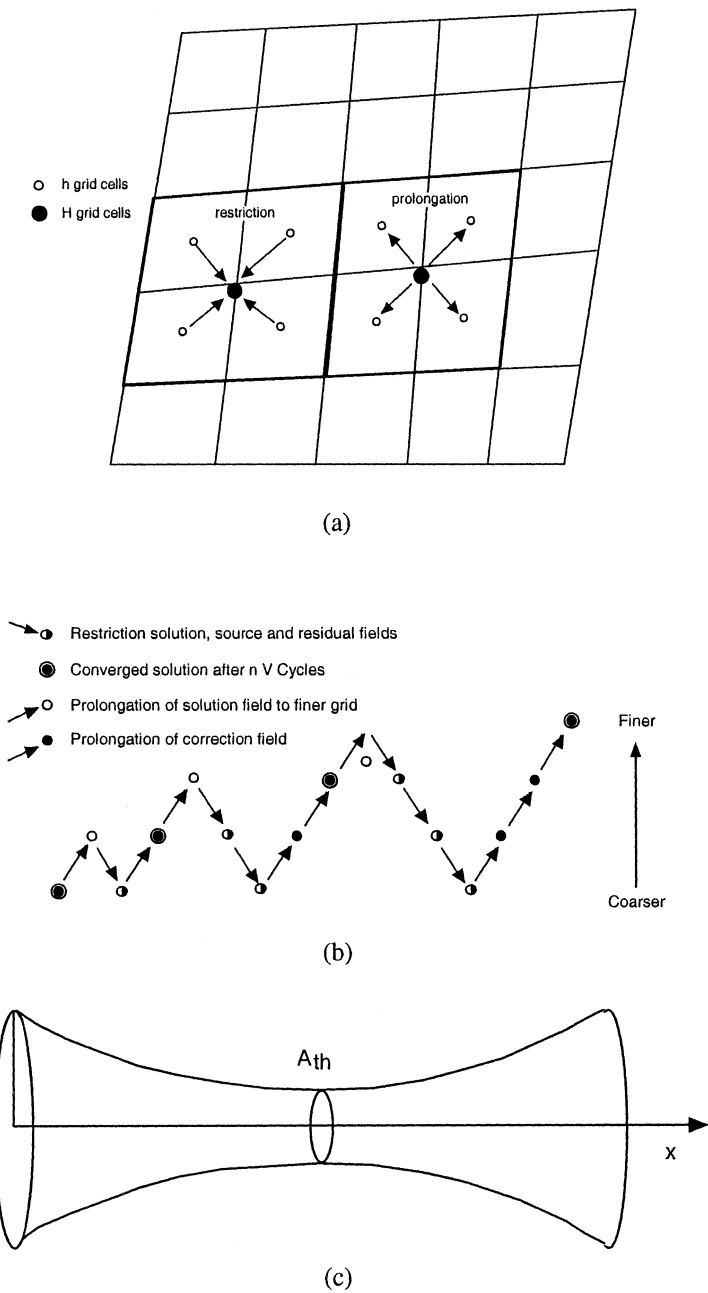


Figure 1. (a) Prolongation and restriction. (b) Full multigrid algorithm cycles. (c) Converging-diverging nozzle test problem.

RESULTS AND DISCUSSION

The validity of the above-described procedure is demonstrated in this section by presenting solutions to the following test cases: (1) subsonic and hypersonic flow in a converging diverging nozzle; (2) subsonic, transonic, and supersonic flow over a bump; and (3) transonic flow over an airfoil. The problems are solved using the single grid (SG), prolongation grid (PG), and full multigrid (FMG) methods. With the prolongation grid and full multigrid methods, four grid levels are used for all results presented here. Experimentation with more grid levels did not show any noticeable difference. For all problems, results obtained are verified by comparison against values reported in previous publications [43, 44]. This is followed by a performance comparison of the different pressure-based algorithms in terms of number of iterations and CPU time to reach the desired level of convergence. In the convergence history plots, residuals are displayed on the finest grid level. Moreover, the residual of a variable ϕ at the end of an outer iteration is defined as:

$$\text{RES}_\phi = \sum_{c.v} \left| A_p \phi_p - \sum_{\text{all } p \text{ neighbours}} A_{nb} \phi_{nb} - B_p \right| \quad (44)$$

All residuals are normalized by their respective inlet fluxes. Computations are terminated when the maximum normalized residual of all variables drops below a very small number ε_s ($\simeq 10^{-5}$). For a given problem, the same value of ε_s is used with all algorithms.

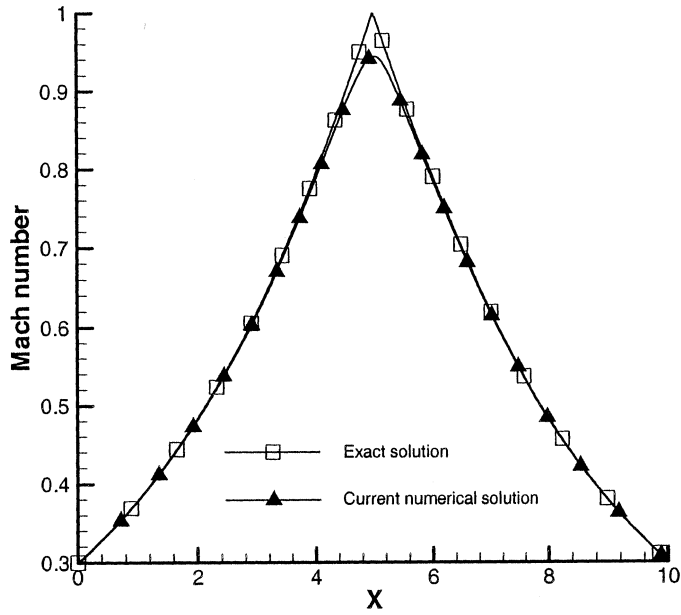
Flow in a Converging-Diverging Nozzle

The flow in a converging-diverging nozzle is a standard test that has been used by several researchers [32, 44] to validate their new numerical methodologies. The nozzle cross-sectional area (Figure 1c) varies according to the relation [23]:

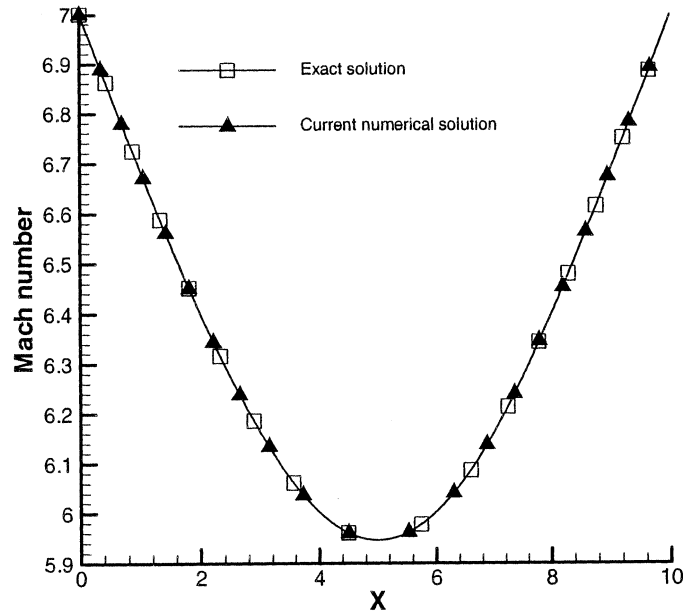
$$A(x) = A_{\text{th}} + (A_i - A_{\text{th}}) \left(1 - \frac{x}{5} \right)^2 \quad (45)$$

where $A_i = 2.035$ and $A_{\text{th}} = 1$ are the inlet and throat areas, respectively, and $0 \leq x \leq 10$. An axisymmetric grid of size 290×34 is used, and solutions for subsonic ($M_{\text{in}} = 0.3$) and hypersonic ($M_{\text{in}} = 7$) flow are generated. For the subsonic case, all flow variables except pressure are specified at the inlet and the pressure is prescribed at the outlet. For the hypersonic situation, all variables are given at the inlet, while at the outlet they are all extrapolated from the interior domain. The area-averaged axial Mach number profiles are compared in Figure 2 against one-dimensional exact solutions. As can be seen, predictions agree well with the exact solutions for both subsonic (Figure 2a) and hypersonic (Figure 2b), flows with numerical results falling on top of analytical calculations except near the throat of the nozzle in the subsonic case, where the Mach number is slightly underpredicted. This could be improved by further refining the grid in that region.

Convergence history plots for the single-grid, prolongation grid, and full multigrid solution methods using the different algorithms are presented in Figure 3. A larger number of iteration is required to converge the solution to the desired level for the subsonic (Figures 3a–3c) than for the hypersonic flow (Figures 3d–3f) with all



(a)



(b)

Figure 2. Comparison between the analytical and numerical Mach number distribution along the nozzle at (a) subsonic ($M_{\text{inlet}} = 0.3$) and (b) hypersonic ($M_{\text{inlet}} = 7$) speeds.

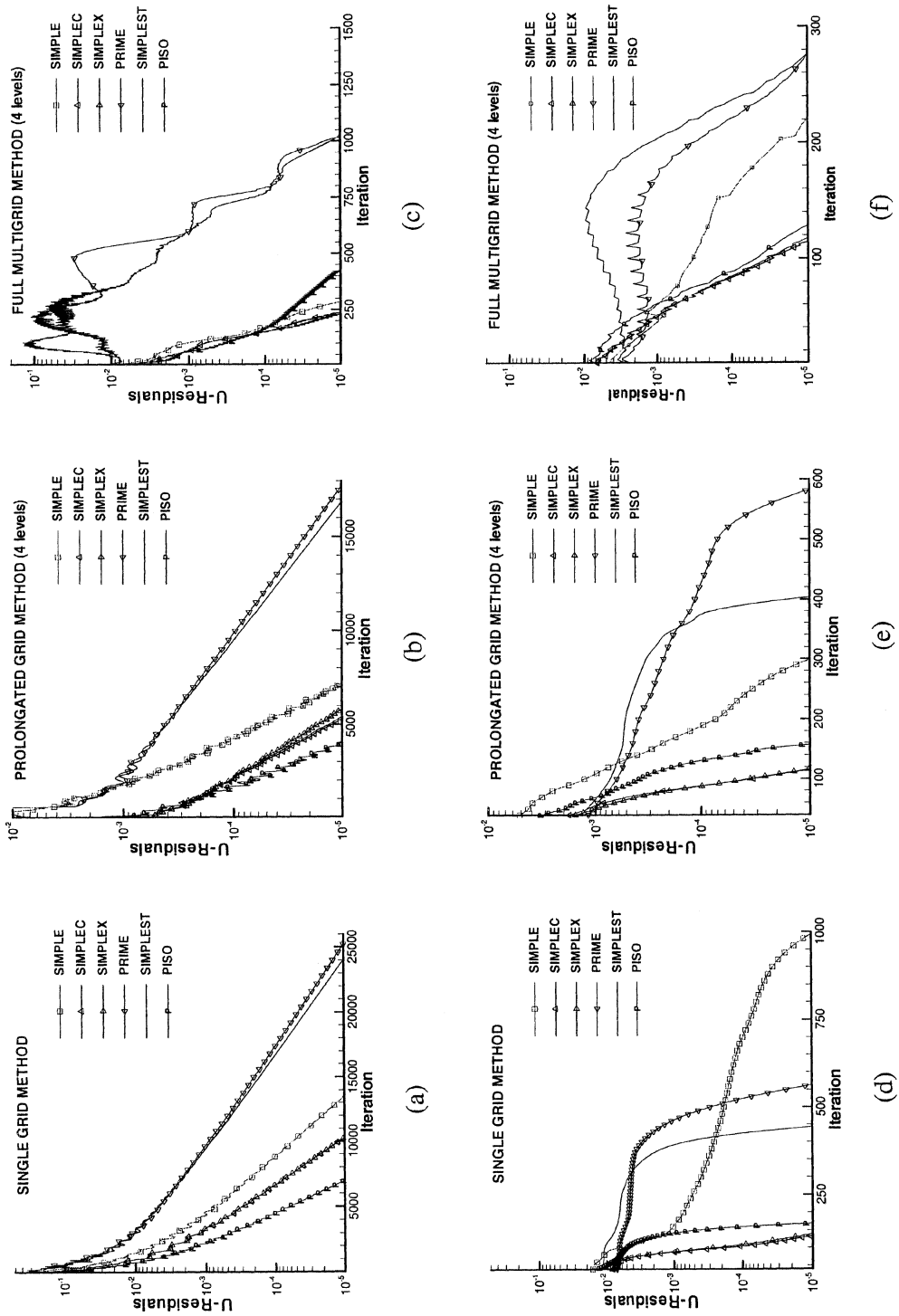


Figure 3. Convergence history plots for flow inside a nozzle at subsonic (a-c) and hypersonic (d-f) speeds.

methodologies and algorithms. Starting with the subsonic case, the convergence plots of the different algorithms using the single-grid method (Figure 3a) indicate that PRIME and SIMPLEST are by far the least-performing algorithms. The SIMPLEC and SIMPLEX algorithms behave similarly and require the lowest number of iterations following PISO. As depicted in Figure 3b, the use of the prolongation grid method reduces the number of iterations required by all algorithms by about 40%. This improvement in performance is due to what amounts to a better initial guess, which becomes increasingly more important as the nonlinearity increases. However, it also affects the relative performance of these algorithms, indicating the sensitivity of some of the smoothers to the initial conditions. While the performance of PRIME and SIMPLEST is still the worst, SIMPLEC is now the best-performing algorithm, requiring nearly half the number of iterations needed by PISO, which is followed by SIMPLEX and SIMPLE. The convergence history plots for the various algorithms using the full multigrid method are displayed in Figure 3c. With all algorithms, a drastic decrease in the number of iterations is noticed. For PRIME and SIMPLEST, the number of iterations is reduced by a factor of 25 and 18 compared to the single-grid and multigrid methods, respectively. The SIMPLEC and SIMPLEX are the best performers, followed by SIMPLE, which outperforms PISO. It is worth mentioning here that PISO requires nearly twice the computational effort of any of the other algorithms because of the four-step solution procedure, which requires solving the pressure-correction equation twice in a global iteration [35].

The convergence history plots of the various algorithms for the hypersonic case are depicted in Figures 3d–3f. For the single-grid method (Figure 3d), SIMPLEC and SIMPLEX require the least number of iterations, followed by PISO, which is contrary to the subsonic case. Also to be noted is the unexpected performance of the SIMPLE algorithm, which requires the largest number of iterations. The use of the prolongation grid method (Figure 3e) does not noticeably reduce the number of iterations needed by PISO, SIMPLEX, SIMPLEC, SIMPLEST, and PRIME. However, it greatly improves the performance of SIMPLE by reducing the number of iterations required for convergence by a factor of 3.33 in comparison with the single-grid method. The full multigrid method (Figure 3f) seems to better improve the performance of SIMPLE, SIMPLEST, and PRIME, where a $\sim 30\%$ reduction in the number of iterations, as compared to the prolongation grid method, is obtained. The performance of SIMPLEX, SIMPLEC, and PISO appears to be unaffected.

CPU time: Flow in a converging-diverging nozzle. The CPU times required by the various algorithms using the different solution methodologies are displayed in Figure 4. For the subsonic case (Figure 4a), the performance of the various algorithms can be divided into two groups, with SIMPLEST and PRIME forming one group and the remaining algorithms composing the second. As depicted, the computational effort of the various algorithms belonging to a group is nearly the same, with the first group requiring (on average) nearly double the computational time needed by the second group with all methodologies. Moreover, the substantial savings accomplished through the use of the full multigrid method with all algorithms is obvious. Furthermore, the use of a four-grid-level arrangement decreases slightly the CPU time in comparison with the three-grid-level configuration.

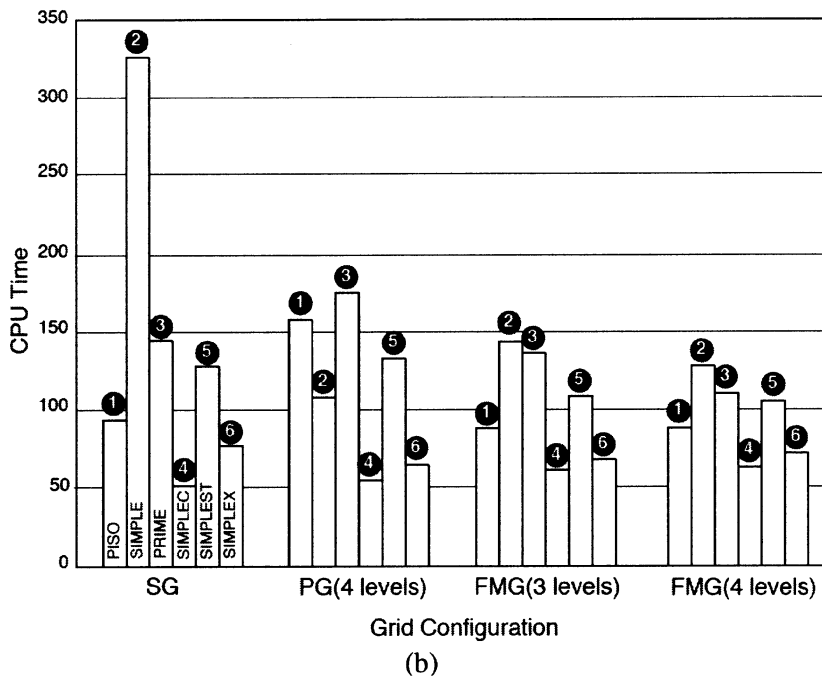
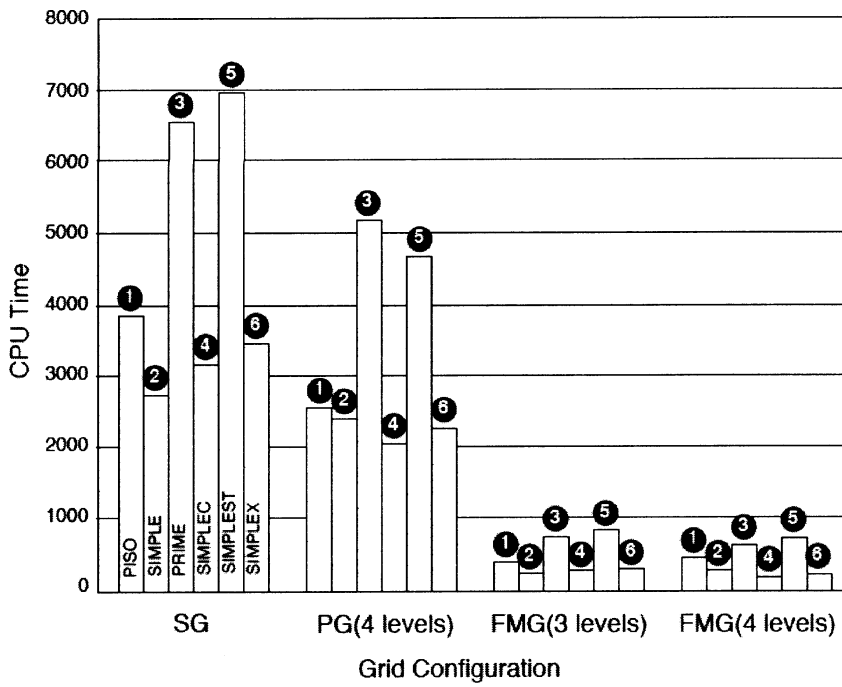


Figure 4. Comparison of CPU time needed by the various algorithms using the different solution methodologies for (a) subsonic and (b) hypersonic flow in a converging-diverging nozzle.

For the hypersonic case (Figure 4b), the maximum CPU time needed is one-twentieth that of the subsonic case. The worst performer with the single-grid method is SIMPLE, which requires almost double the CPU time of PRIME. This CPU time is reduced by half with the prolongation grid and full multigrid methods. Other algorithms are less affected by the method used, with slight increase or decrease in the CPU time required. This behavior is expected since the full multigrid method was originally developed for elliptic flows, and hypersonic flows are of the hyperbolic type.

Flow over a Circular Arc Bump

The flow over a circular arc bump is a good test for the stability and accuracy of numerical algorithms [43, 44]. The physical situation consists of a channel of width equal to the length of the circular arc bump and of total length equal to three lengths of the bump. Results are presented for three different types of flow (subsonic, transsonic, and supersonic). For subsonic and transonic calculations, the thickness-to-chord ratio is 10% and for supersonic flow calculations it is 4%.

Subsonic flow over a circular arc bump. With an inlet Mach number of 0.5, the inviscid flow in the channel is fully subsonic and symmetric across the middle of the bump. A symmetric grid with respect to the vertical centerline of the channel is used. At the inlet, the flow is assumed to have uniform properties and all variables, except pressure, are specified. At the outlet section, the pressure is prescribed and all other variables are extrapolated from the interior of the domain. The flow tangency condition is applied at the walls.

The computed isobars in the channel, which are almost symmetric and in excellent agreement with similar results reported in the literature [44], are displayed in Figure 5a. Figure 5b shows a comparison between predicted Mach number profiles along the inner and outer walls of the channel and those documented by Favini [43]. The two sets of results are almost identical.

The residual history plots, as a function of the number of iterations, for the various algorithms using the different solution methodologies are presented in Figures 5c–5e. The relative performance of the various algorithms appears to be similar with the various approaches. The SIMPLEST and PRIME algorithms require the largest number of iterations, while PISO requires the lowest number. The performance of SIMPLE, SIMPLEC, and SIMPLEX is nearly identical. For all algorithms, the use of the prolongation grid approach (Figure 5d) reduces the number of iterations, on average, by half (with SIMPLE requiring less iteration than SIMPLEC and SIMPLEX). On the other hand, the use of the full multigrid strategy (Figure 5e) diminishes the number of iterations required to reach the desired level of convergence, in comparison with the single-grid method, by a factor varying between 10.5 (for PRIME) and 21.3 (for SIMPLEC and SIMPLEX).

Transonic flow over a circular arc bump. With the exception of the inlet Mach number being set to 0.675, the grid distribution and the implementation of boundary conditions are identical to those described for subsonic flow. Although a shock over the bump is predicted as shown by the pressure contours displayed in Figure 6a, since it does not propagate in the direction of the flow, there is no

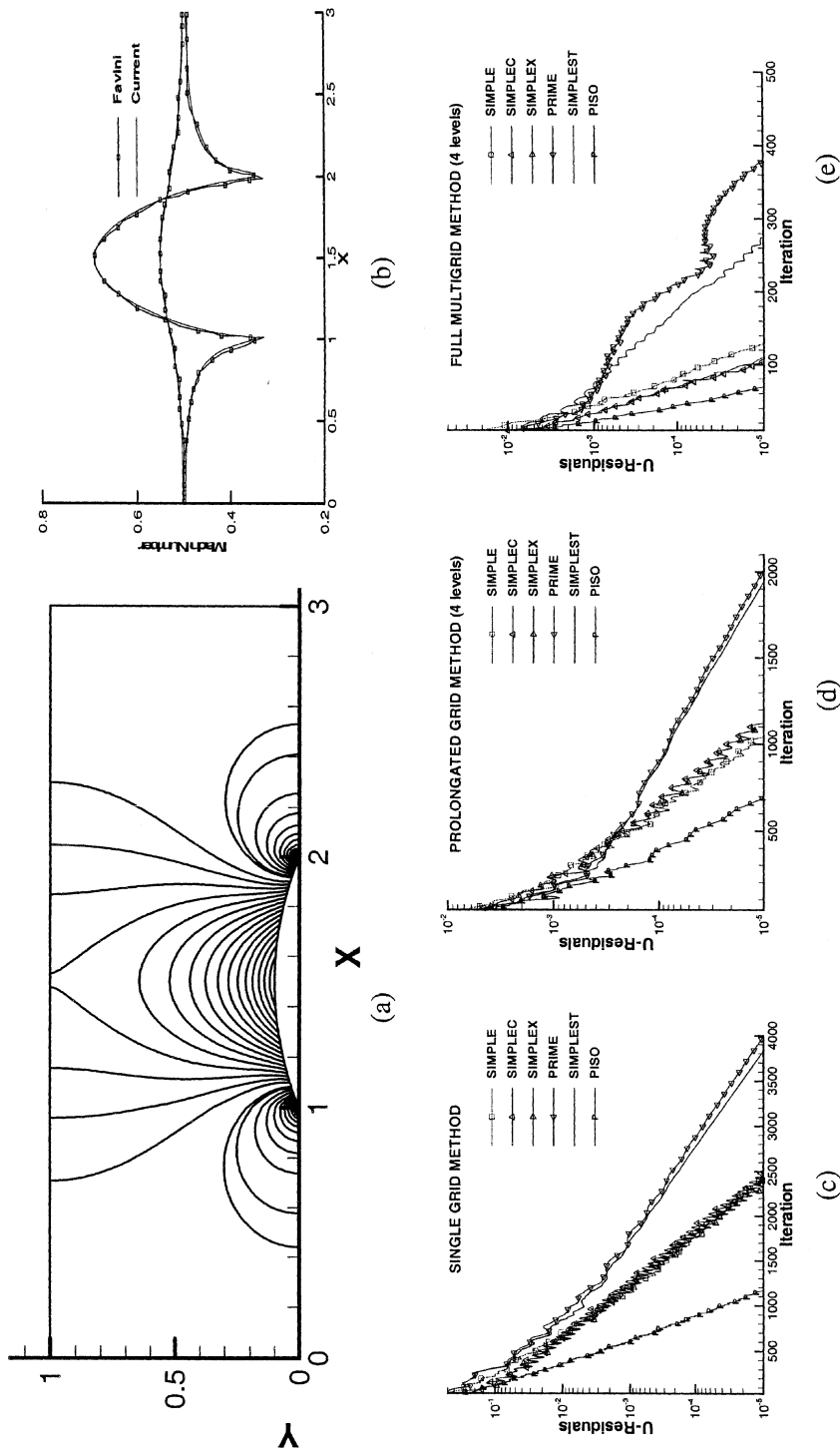


Figure 5. (a) Pressure contours and (b) Mach number distributions along the upper and lower walls for subsonic flow over a bump ($M_{\text{inlet}} = 0.5$). Convergence history plots of the various algorithms using the (c) single-grid, (d) prolonged grid, and (e) multigrid methodologies for subsonic flow over a bump ($M_{\text{inlet}} = 0.5$).

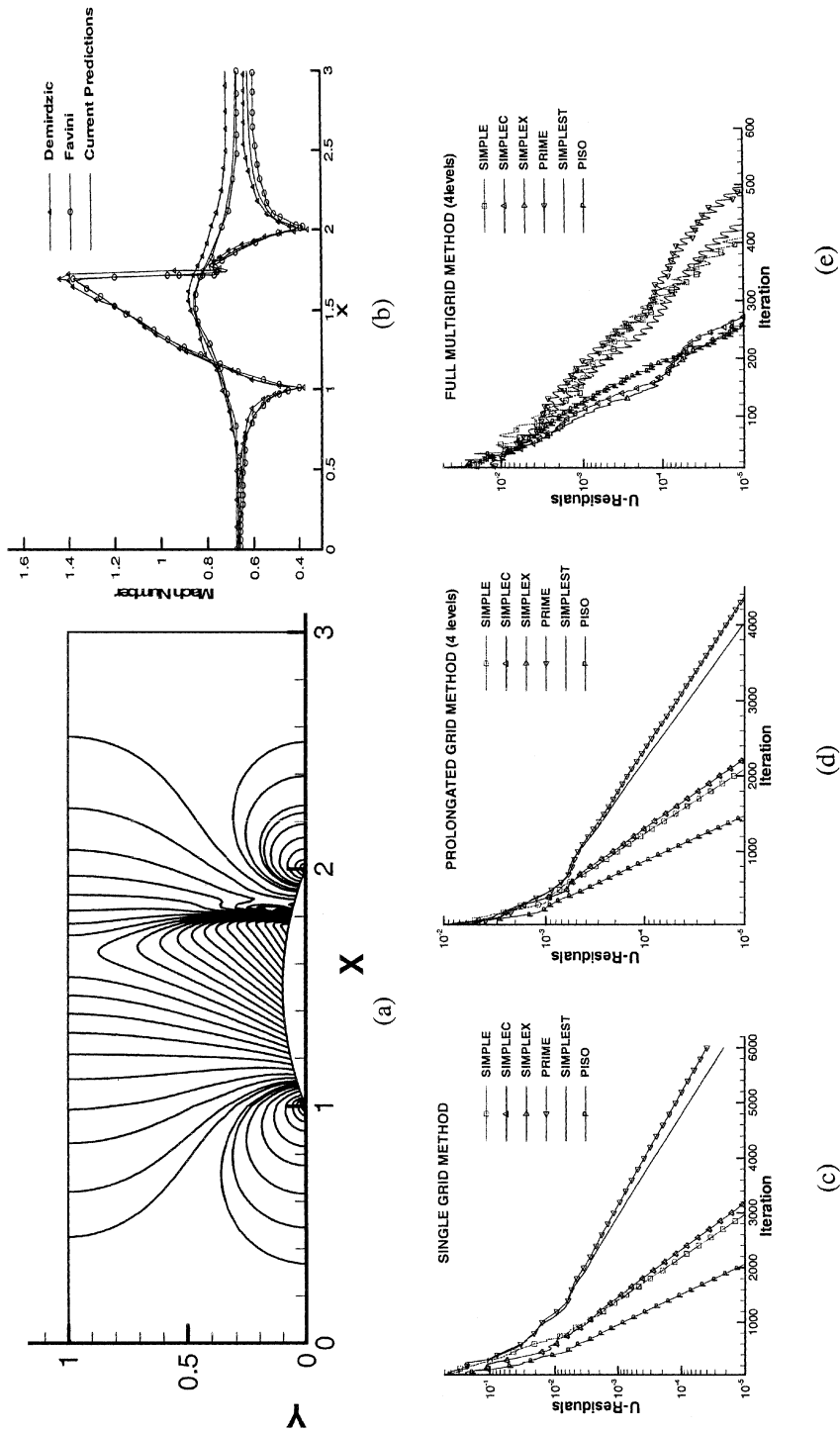


Figure 6. (a) Pressure contours and (b) Mach number distributions along the upper and lower walls for transonic flow over a bump ($M_{inlet} = 0.675$). Convergence history plots of the various algorithms using the (c) single-grid, (d) prolonged grid, and (e) multigrid methodologies for transonic flow over a bump ($M_{inlet} = 0.675$).

need to change the density of the grid downstream of the bump. As depicted in Figure 6b, Mach number distributions along the lower and upper walls of the channel are in good agreement with similar results reported by Demirdzic [44] and Favini [43], which is a good indication of the correctness of the solution procedure.

The convergence history plots using the single-grid method displayed in Figure 6c reveal that PISO requires the lowest number of iterations, followed by SIMPLE, SIMPLEC, and SIMPLEX. On the other hand, the number of iterations entailed by PRIME and SIMPLEST is more than thrice the number needed by PISO. As shown in Figure 6d, the use of the prolongation grid method reduces the number of iterations of all algorithms by one-third their original values, while the relative performance of all algorithms remains unchanged. The use of the full multigrid method changes the performance of the various algorithms dramatically, as depicted in Figure 6e. In this case, the number of iterations required is reduced by a factor varying from 7.4 with PISO to over 16 by SIMPLEST. The number of iterations required by PISO, SIMPLEC, and SIMPLEX becomes nearly identical, while that required by SIMPLEST approaches the one needed by SIMPLE. Nevertheless, PRIME consumes the largest number of iterations.

Supersonic flow over a circular arc bump. Computations are presented for an inlet Mach number of 1.4. For this value of inlet Mach number and the geometry used, the flow is also supersonic at the outlet. Thus, all variables at the inlet are prescribed, and at the outlet all variables are extrapolated. Isobars displayed in Figure 7a, revealing the formation of shock waves at the leading and trailing edges of the bump, are in excellent agreement with similar results reported in [44]. This is also demonstrated in Figure 7b by the good concurrence between the computed Mach number distributions along the lower and upper walls of the channel and similar profiles extracted from Favini [43].

Residual histograms of the various algorithms and for the different solution methodologies as a function of the iteration number are depicted in Figures 7c–7d. For the single-grid method (Figure 7c), the number of iterations required by the different algorithms varies between 450 iterations for PISO and 1,500 iterations for PRIME (which is appreciably lower than the values obtained in the subsonic and transonic cases). SIMPLE is the second best performer after PISO, while SIMPLEC and SIMPLEX have identical performance, which is close to that of SIMPLE. The number of iterations needed by SIMPLEST is about 17% less than PRIME. The use of the prolongation grid method (Figure 7d) reduces the number of iterations required on the finest level by all algorithms, except PRIME, by over 30%. For the PRIME algorithm, the number of iterations has increased by about 25%. On the other hand, the full multigrid method (Figure 7e) appreciably decreases the number of iterations required by all algorithms. An average reduction of about 80% is achieved in comparison with the single-grid method. The relative performance of the algorithms remains unchanged with the exception of SIMPLEST, which in this case requires the largest number of iterations on the finest mesh, following PRIME.

CPU time: Flow over a circular arc bump. Figure 8 displays the CPU times required by the various algorithms using the different solution methodologies in the various Mach number regimes for the flow over a circular arc bump. As for the case of a converging-diverging nozzle, subsonic flow results depicted in Figure 8a suggest

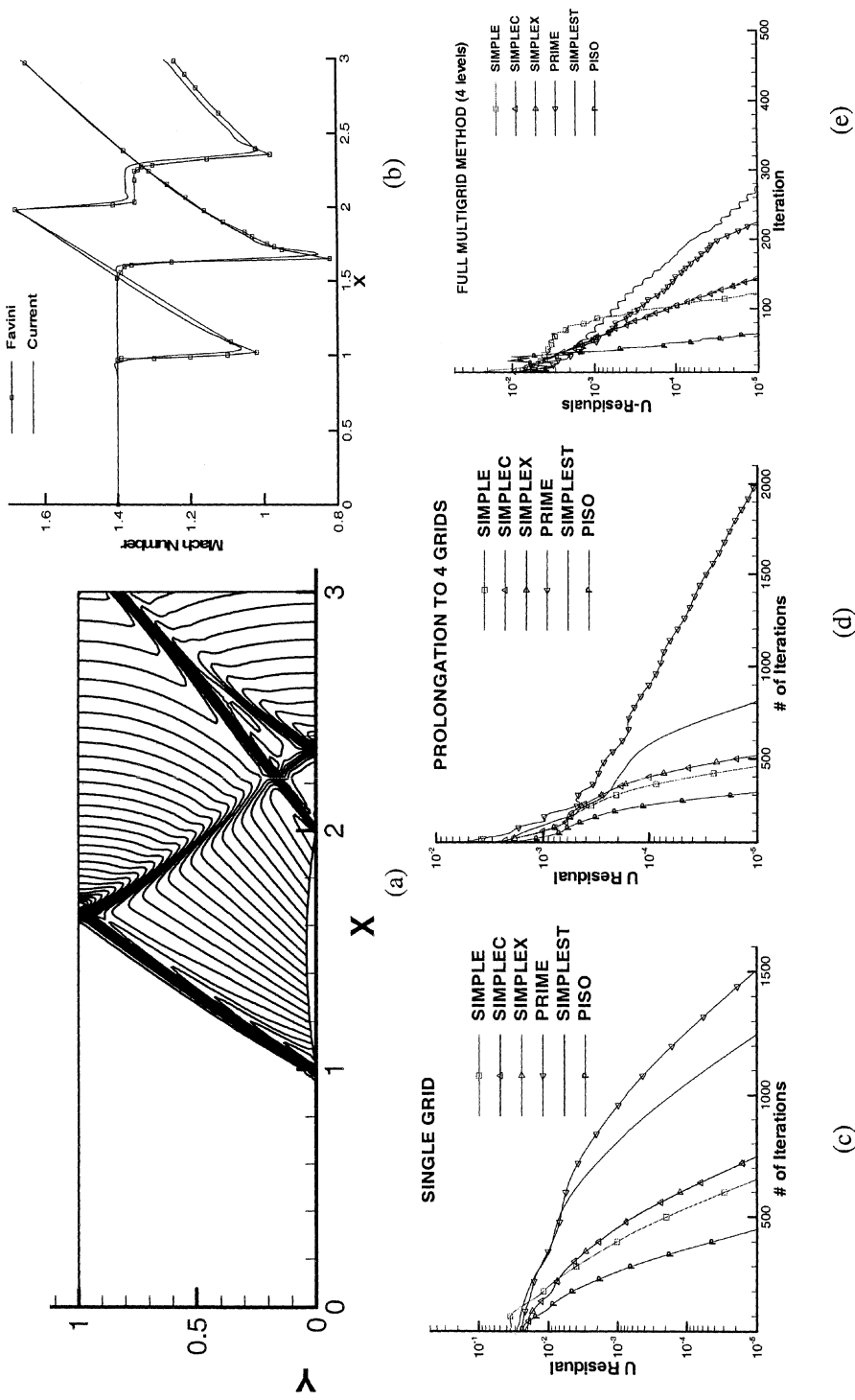


Figure 7. (a) Pressure contours and (b) Mach number distributions along the upper and lower walls for supersonic flow over a bump ($M_{\text{inlet}} = 1.4$). Convergence history plots of the various algorithms using the (c) single-grid, (d) prolongation grid, and (e) multigrid methodologies for supersonic flow over a bump ($M_{\text{inlet}} = 1.4$).

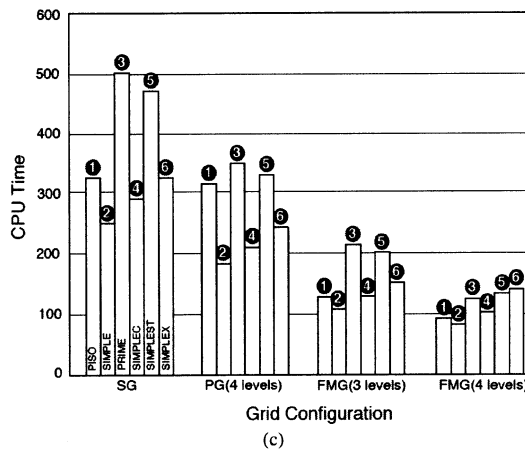
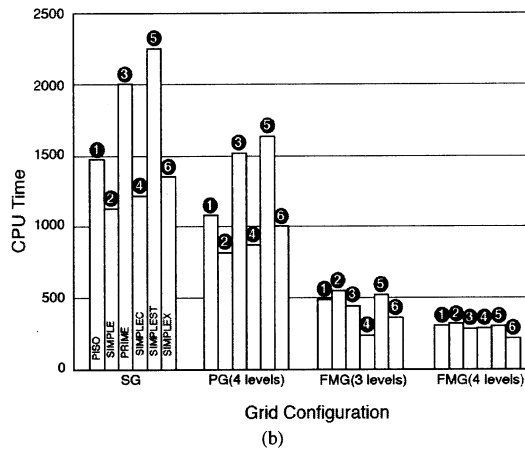
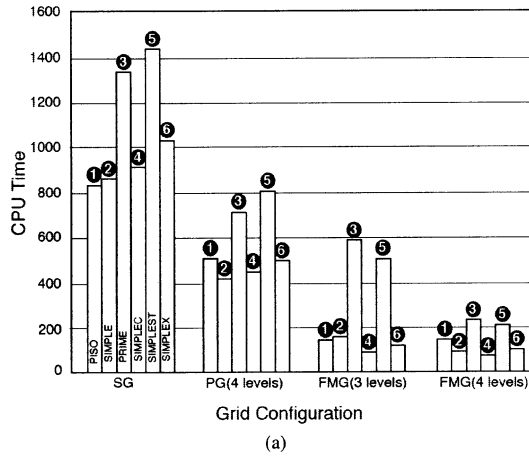


Figure 8. Comparison of CPU time needed by the various algorithms using the different solution methodologies for (a) subsonic, (b) transonic, and (c) supersonic flow over a bump.

dividing the various algorithms into two groups, with SIMPLEST and PRIME forming one group and the remaining algorithms composing the second. As shown, the computational effort of the various algorithms belonging to a group is nearly the same, with the first group requiring (on average) nearly 55% more computational time than the second group with the single-grid method, 40% more with the prolongation grid method, and around 45% more with the full multigrid method over four grid levels (no appreciable improvement in performance is seen for group 1 when using three grid levels). Nevertheless, the CPU time is decreased almost 7-fold through the use of the full multigrid method.

Similar behavior is noticed for the transonic (Figure 8*b*) and supersonic (Figure 8*c*) flows with the single-grid and prolongation grid methods. With the full multigrid method, the performance of both groups is closer than in the subsonic case. Also to be noticed is the fact that no single algorithm is showing universal superiority over other algorithms. Rather, superiority is case-based. The conclusion to be drawn here is the effectiveness of the full multigrid method in greatly reducing the computational cost with all algorithms.

Flow over a NACA 0012 Airfoil

The final test case considered is for transonic flow around a NACA 0012 airfoil, which has become a standard test case used by several researchers to evaluate their methodologies [43, 45–48]. The flow approaches the airfoil with a Mach number of 0.85 at 1° angle of attack. An O-type grid is used in the solution with a density on the finest mesh of size 192×96 control volumes. Moreover, the computational domain extends 100-chord lengths in all directions around the airfoil. Furthermore, all variables except pressure are prescribed at the inlet, while at the outlet the pressure is assigned and all other variables are extrapolated from the interior.

Computed results using SIMPLEC and SIMPLEX are presented in Figure 9. The use of these two algorithms is linked to the fact that they were the only ones that did not require additional upwind bleeding for the advection terms. The remaining algorithms required a large percentage of upwind bleeding to the high-resolution schemes and are termed numerically unstable for the current application. Figure 9*a* displays isobars around the airfoil. Two shock waves, one on the upper and the second on the lower side of the airfoil, are obtained at two distinct streamwise locations. These isobars are in excellent agreement with published results [43]. As a further check for accuracy, the computed pressure coefficients along both sides of the airfoil are compared against similar ones reported in [43]. As depicted in Figure 9*b*, the agreement is very good, which is additional evidence of the correctness of the solution procedure. The convergence history plots presented in Figures 9*c*–9*e* indicate the capability of both algorithms to predict external transonic flows. In terms of number of iterations, the SIMPLEC algorithm performs slightly better with the prolongation grid method (Figure 9*d*) than with the full multigrid method (Figure 9*e*). The opposite is true with SIMPLEX. However, both methods reduce the number of iterations, for both algorithms, by a factor of nearly 3.5 over the single-grid method. This reduction ratio appears to equally hold for the CPU time displayed in Figure 9*f*. To be noticed in that figure is the fact that the best performance is obtained with the full multigrid method over three grid levels.

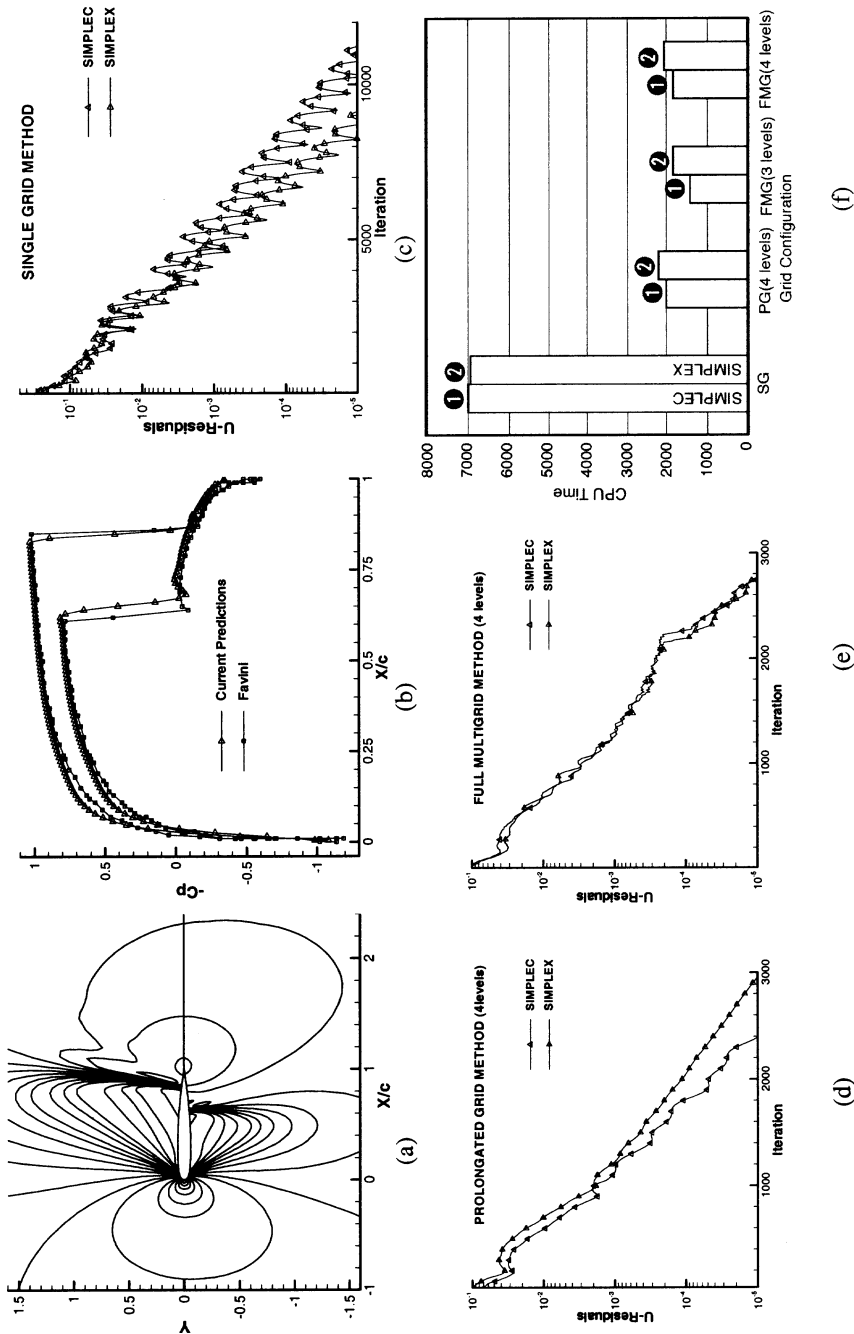


Figure 9. (a) Pressure contours and (b) C_p distributions along the lower and upper surfaces of a NACA 0012 airfoil at transonic speed. Convergence history plots for transonic flow around a NACA 0012 airfoil at transonic speed using the (c) single-grid, (d) prolongation grid, and (e) full multigrid methods. (f) Comparison of CPU time needed by SIMPLEX and SIMPLEX for the transonic flow around a NACA 0012 airfoil.

CLOSING REMARKS

In this article, six incompressible flow algorithms (SIMPLE, SIMPLEC, SIMPLEX, SIMPLEST, PISO, and PRIME) were extended to predict compressible fluid flow at all speeds and implemented within a single-grid, a prolongation grid, and a full multigrid methodology. The relative merits of these extended algorithms were compared by solving several problems encompassing flows in the subsonic, transonic, supersonic, and hypersonic regimes. The full multigrid method was found to be the most efficient in all problems solved and for all algorithms, with its efficiency, as expected, decreasing with increasing Mach number values. Based on the computations performed, SIMPLEC and SIMPLEX were the most stable algorithms, while SIMPLEST and PRIME were the most expensive. Even though PISO required the least number of iterations, it was prone to numerical instability in problems that involved shock waves. The performance of SIMPLE in terms of CPU time was acceptable in most of the problems, but it was also sensitive to shocks and became numerically unstable in such problems.

Future work will concentrate on dynamic agglomeration of fine grid cells in multigrid methods. In such methods, instead of structured agglomeration of four fine-grid cells to produce one coarse-grid cell, as was done in this article, the flow conditions could be considered as a criterion in the design of the multigrid system. For example, if a shock wave exists in the computational domain it would probably be more efficient not to agglomerate grid points across it.

REFERENCES

1. F. V. Poussin, An Accelerated Relaxation Algorithm for Iterative Solution of Elliptic Equations, *SIAM J. Numer. Anal.*, vol. 5, pp. 340–351, 1968.
2. A. Brandt, Multi-Level Adaptive Solutions for Boundary Value Problems, *Math. Comput.*, vol. 31, pp. 333–390, 1977.
3. A. Brandt, *Guide to Multigrid Development*, Springer-Verlag, Berlin, 1982.
4. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
5. U. Trottenberg, C. Oosterlee, and A. Schuller, *Multigrid*, Academic Press, Cornwall, England, 2001.
6. A. Brandt and N. Dinar, Multigrid Solutions to Flow Problems, in *Numerical Methods for Partial Differential Equations*, Part S, pp. 43–147, Academic Press, New York, 1979.
7. C. P. Thompson and P. Lezeau, Application of the Full Approximation Storage Method to the Numerical Simulation of Two-Dimensional Steady Incompressible Viscous Multiphase Flows, *Int. J. Numer. Meth. Fluids*, vol. 28, no. 8, pp. 1217–1239, 1998.
8. M. C. Thompson and J. H. Ferziger, An Adaptive Multigrid Technique for the Incompressible Navier-Stokes Equations, *J. Comput. Phys.*, vol. 82, pp. 94–121, 1989.
9. P. Rubini, H. E. Becker, E. Grandmaison, A. Pollard, A. Sobisak, and C. Thurgood, Multigrid Acceleration of Three Dimensional Turbulent Variable Density Flows, *Numer. Heat Transfer B*, vol. 22, pp. 163–177, 1992.
10. S. Zeng and P. Wesseling, Multigrid Solution of the Incompressible Navier-Stokes in Generalized Coordinates, *SIAM J. Numer. Anal.*, vol. 31, pp. 1764–1784, 1994.
11. M. Peric, M. Rüger, and G. Scheuerer, A Finite Volume Multigrid Method for Calculating Turbulent Flows, *Proc. 7th Symp. on Turbulent Shear Flows*, Stanford University, Stanford, CA, 1989, vol. 1, pp. 7.3.1–7.3.6.

12. M. F. Paisley, Multigrid Solution of the Incompressible Navier-Stokes Equations for Three-Dimensional Recirculating Flow: Coupled and Decoupled Smoothers Compared, *Int. J. Numer. Meth. Fluids*, vol. 30, pp. 441–459, 1999.
13. S. J. Kiao and F. Mashayek, A Multigrid Approach for Steady State Laminar Viscous Flows, *Int. J. Numer. Meth. Fluids*, vol. 37, pp. 107–123, 2001.
14. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. Mass Transfer*, vol. 15, pp. 1787–1806, 1972.
15. S. Sivaloganathan and G. J. Shaw, An Efficient Non-linear Multigrid Procedure for the Incompressible Navier-Stokes Equations, *Int. J. Numer. Meth. Fluids*, vol. 8, pp. 417–440, 1988.
16. W. Shyy and C. S. Sun, Development of a Pressure-Correction/Staggered Grids Based Multigrid Solver for Incompressible Recirculating Flow, *Comput. Fluids*, vol. 22, pp. 51–76, 1993.
17. C. Becker, J. H. Ferziger, M. Perić, and G. Scheurer, Finite Volume Multigrid Solutions of the Two-Dimensional Incompressible Navier-Stokes Equations, *Notes Numer. Fluid Mech.*, vol. 23, pp. 37–47, 1988.
18. K. M. Smith, W. K. Cope, and S. P. Vanka, Multigrid Procedure for Three Dimensional Flows on Non-orthogonal Collocated Grids, *Int. J. Numer. Meth. Fluids*, vol. 17, pp. 887–904, 1993.
19. F. S. Lien and M. A. Lechziner, Multigrid Acceleration for Recirculating Laminar and Turbulent Flow Computed on a Non-orthogonal, Collocated Finite Volume Scheme, *Comput. Meth. Appl. Mech. Eng.*, vol. 118, pp. 351–371, 1994.
20. T. Gjesdal and M. E. H. Lossius, Comparison of Pressure Correction Smoothers for Multigrid Solution of Incompressible Flow, *Int. J. Numer. Meth. Fluids*, vol. 25, pp. 393–405, 1997.
21. C. R. Maliska and G. D. Raithby, Calculating 3-D Fluid Flows Using Non-orthogonal Grid, *Proc. Third Int. Conf. on Numerical Methods in Laminar and Turbulent Flows*, Seattle, WA, 1983, pp. 656–666.
22. F. Moukalled and M. Darwish, A Unified Formulation of the Segregated Class of Algorithms for Fluid Flow at All Speeds, *Numer. Heat Transfer B*, vol. 37, no. 1, pp. 103–139, 2000.
23. F. Moukalled and M. Darwish, A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds, *J. Comput. Phys.*, vol. 168, no. 1, pp. 101–133, 2001.
24. D. J. Mavriplis and A. Jameson, Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes, *AIAA J.*, vol. 26, no. 7, pp. 824–831, 1988.
25. S. D. Connel and D. G. Holmes, A 3D Unstructured Adaptive Multigrid Scheme for the Euler Equations, *AIAA J.*, vol. 32, no. 8, pp. 1626–1632, 1994.
26. A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *J. Comput. Phys.*, vol. 2, pp. 12–26, 1967.
27. A. Rizzi and L. E. Eriksson, Computation of Inviscid Incompressible Flow with Rotation, *J. Fluid Mech.*, vol. 153, no. 3, pp. 275–312, 1985.
28. D. Choi and C. L. Merkle, Application of Time-Iterative Schemes to Incompressible Flow, *AIAA J.*, vol. 23, pp. 1518–1524, 1985.
29. K. C. Karki, A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries, Ph.D. thesis, University of Minnesota, Twin Cities, MN, June 1986.
30. C. M. Rhie, A Pressure Based Navier-Stokes Solver Using the Multigrid Method, *AIAA Paper 86-0207*, 1986.
31. W. Shyy and M. E. Braaten, Adaptive Grid Computation for Inviscid Compressible Flows Using a Pressure Correction Method, *AIAA paper 88-3566-CP*, 1988.

32. F. S. Lien and M. A. Leschziner, A Pressure-Velocity Solution Strategy for Compressible Flow and Its Application to Shock/Boundary-Layer Interaction Using Second-Moment Turbulence Closure, *J. Fluids Eng.*, vol. 115, pp. 717–725, 1993.
33. E. S. Politis and K. C. Giannakoglou, A Pressure-Based Algorithm for High-Speed Turbomachinery Flows, *Int. J. Numer. Meth. Fluids*, vol. 25, pp. 63–80, 1997.
34. J. P. Van Doormaal and G. D. Raithby, Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, vol. 7, pp. 147–163, 1984.
35. R. I. Issa, Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting, *Mech. Eng. Rep.* FS/82/15, Imperial College, London, UK, 1982.
36. J. P. Van Doormaal and G. D. Raithby, An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, ASME paper 85-HT-9, Presented at the National Heat Transfer Conference, Denver, CO, 4–7 August, 1985.
37. D. B. Spalding, Mathematical Modelling of Fluid Mechanics, Heat Transfer and Mass Transfer Processes, *Mech. Eng. Dept. Rep.* HTS/80/1, Imperial College of Science, Technology and Medicine, London, UK, 1980.
38. M. S. Darwish and F. Moukalled, Normalized Variable and Space Formulation Methodology for High Resolution Schemes, *Numer. Heat Transfer B*, vol. 26, pp. 79–96, 1994.
39. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1981.
40. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past and Airfoil with Trailing Edge Separation, *AIAA J.*, vol. 21, pp. 1525–1532, 1983.
41. M. Peric, A Finite Volume Method for the Prediction of Three Dimensional Fluid Flow in Complex Ducts, Ph.D. thesis, Imperial College, London, UK, 1985.
42. S. Sivaloganathan and G. J. Shaw, On the Smoothing Properties of the SIMPLE Pressure Correction Algorithm, *Int. J. Numer. Meth. Fluids*, vol. 8, pp. 441–461, 1988.
43. B. Favini, R. Broglia, and A. Di Mascio, Multigrid Acceleration of Second-Order ENO Schemes from Low Subsonic to High Supersonic Flows, *Int. J. Numer. Meth. Fluids*, vol. 23, pp. 589–606, 1996.
44. I. Demirdzic, Z. Lilek, and M. Peric, A Collocated Finite Volume Method for Predicting Flows at all Speeds, *Int. J. Numer. Meth. Fluids*, vol. 16, pp. 1029–1050, 1993.
45. C. J. Hwang and S. J. Wu, Adaptive Finite Volume Upwind Approach on Mixed Quadrilateral-Triangular Meshes, *AIAA J.*, vol. 31, pp. 61–67, 1993.
46. J. Y. Yang, Y. C. Perng, and R. H. Yen, Implicit Weighted Essentially non-Oscillatory Schemes for the Navier-Stokes Equations, *AIAA J.*, vol. 39, pp. 2082–2090, 2001.
47. J. T. Batina, Implicit Flux-Split Euler Schemes for Unsteady Aerodynamic Analysis Involving Unstructured Dynamic Meshes, *AIAA J.*, vol. 29, pp. 1836–1843, 1991.
48. Y. Yaldin and D. Caughy, Block Multigrid Implicit Solution of the Euler Equations of Compressible Fluid Flow, *AIAA J.*, vol. 29, pp. 712–719, 1991.