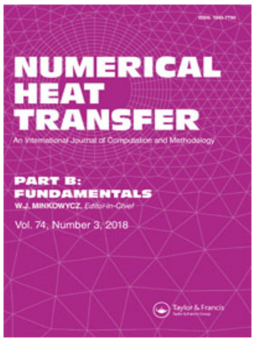

Numerical Heat Transfer, Part B: Fundamentals

An International Journal of Computation and Methodology

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: https://www.tandfonline.com/loi/unhb20

Adaptive inner iteration processes in pressure-based method for viscous compressible flows

Jin-Ping Wang, Jian-Fei Zhang, Zhi-Guo Qu & Wen-Quan Tao

To cite this article: Jin-Ping Wang, Jian-Fei Zhang, Zhi-Guo Qu & Wen-Quan Tao (2018) Adaptive inner iteration processes in pressure-based method for viscous compressible flows, Numerical Heat Transfer, Part B: Fundamentals, 74:3, 603-622, DOI: 10.1080/10407790.2018.1515330

To link to this article: https://doi.org/10.1080/10407790.2018.1515330

Published online: 11 Dec 2018.

NUMERICAL HEAT TRANSFER, PART B: FUNDAMENTALS
2018, VOL. 74, NO. 3, 603–622
https://doi.org/10.1080/10407790.2018.1515330

Adaptive inner iteration processes in pressure-based method for viscous compressible flows

Jin-Ping Wang, Jian-Fei Zhang, Zhi-Guo Qu, and Wen-Quan Tao

Key Laboratory of Thermo-Fluid Science and Engineering of MOE, School of Energy and Power Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, P. R. China

ABSTRACT

In some pressure-based methods, inner iteration processes are introduced to achieve efficient solutions. However, number of the inner iteration is fixed as 2 or 4 for different computations. In this paper, a mechanism is proposed to control inner iteration processes to make the number of inner iterations vary adaptively with different problems. The adaptive inner iteration processes are used in viscous compressible flows. Results reveal that by introducing inner iteration processes, computational efficiency is highly improved compared with that of the solution without inner iteration. In addition, adaptive inner iteration solutions have better robustness than fixed inner iteration solutions.

ARTICLE HISTORY
Received 17 June 2018
Accepted 18 August 2018

1. Introduction

In the existing numerical methods, density-based method is used traditionally for compressible flows. However, the performance of this method is not good for incompressible flow or low Mach number flow due to the slight change of density. Pressure-based method was devised originally for incompressible flows. As it can avoid the difficulties encountered in the density-based method, the pressure-based method is of great potential to be developed as a unified method for flows at different speeds.

SIMPLE algorithm is a famous pressure-based method, proposed initially by Patankar and Spalding [1] for incompressible flows, and it has been used in many researches. The two assumptions in SIMPLE algorithm result in a poor interconnection between pressure and velocity and a low convergence rate in calculations. To remove the first assumption in SIMPLE algorithm, SIMPLER algorithm was put forward, which improves the inherent interconnection between pressure and velocity [2]. After successful applications in incompressible flows, SIMPLE and SIMPLER algorithms have been extended for solving viscous compressible flows [3,4].

CLEAR algorithm was developed in 2004 to remove the second assumption in SIMPLE algorithm [5]. It is revealed that CLEAR algorithm can greatly enhance the convergence rate with reasonable robustness compared with SIMPLER algorithm in incompressible fluid flow [6]. With its success in incompressible flows, it was further extended for viscous compressible flows. Compared with SIMPLE algorithm, CLEAR algorithm can significantly enhance the convergence rate for subsonic, transonic, and supersonic flows at a small under-relaxation factor [7].

CONTACT Jian-Fei Zhang zhangjf@mail.xjtu.edu.cn; Zhi-Guo Qu zgqu@mail.xjtu.edu.cn Key Laboratory of Thermo-Fluid Science and Engineering of MOE, School of Energy and Power Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P. R. China

Nomenclature

A	Coefficient in the discretized equation
A	Surface area
b	Source term
E	Time step
P	Pressure
P'	Pressure correction
R	Gas constant
R_ϕ	Source term caused by pressure
S_ϕ	Source term caused by velocity
T	Temperature
u, v	Velocity components in the x- and y-directions
x, y	Cartesian coordinates
α	Under-relaxation factor
Γ	Nominal diffusion coefficient
ρ	Density
ϕ	General variable
Δt	Time step
ΔV	Volume
Subscripts	
e, w, …	east, west face of a control volume
E, W, …	East, West neighbor of the main grid point
nb	Neighbours of the P grid point
P	Grid point P
Superscripts	
$'$	Correction field
0	Values from previous time step or previous outer iteration
*	Values from intermediate calculation
**	Final values at present time step or present outer iteration

IDEAL algorithm is a pressure-based method developed for incompressible flows [8], which overcomes the two assumptions in SIMPLE algorithm successfully and guarantees a close coupling between velocity and pressure. After its initial construction on a staggered grid system in 2 D Cartesian coordinates, IDEAL algorithm was extended to staggered grid systems in 3 D Cartesian coordinates, collocated grid systems in 3 D Cartesian coordinates, and body-fitted collocated grid systems in 3 D non-orthogonal curvilinear coordinates. It was illustrated that IDEAL algorithm is robust and efficient and can converge almost at any under-relaxation factor [9]. In spite of the good performances in incompressible flows, there is no report on IDEAL algorithm in compressible flows.

The success of IDEAL algorithm lies on two inner iterations for solving pressure equations. The first inner iteration process is to update the pressure field to improve solutions of momentum equation. The second inner iteration process is to improve velocity fields in order to guarantee the mass conservation. Numbers of the inner iterations are controlled by N1 (for first inner iteration) and N2 (for second inner iteration). In Sun's study, N1 and N2 are fixed as 4 usually [8]. However, it is overestimated in some computations but underestimated in other computations.

In this paper, IDEAL algorithm is firstly extended for compressible flows. To ensure the stability of computations, pressure correction equations are solved instead of pressure equations in the inner iteration process. Based upon the two inner iteration processes in compressible IDEAL algorithm, a mechanism is proposed to achieve adaptive inner iteration processes. It is supposed that the number of inner iterations should vary with the calculation progress based on different problems in adaptive inner iteration processes.

The adaptive inner iteration processes are validated by classical compressible test models: flow over a bump and flow through a nozzle. The performance of adaptive inner iteration is compared with those of CLEAR algorithm (no inner iteration process) and IDEAL algorithm (fixed inner iteration processes).

In section 2, the governing equations of compressible flows and the discretization equations using FVM are presented. Implementation of the adaptive inner iteration processes is shown in section 3. The accuracy of the adaptive inner iteration is verified and its comparison with CLEAR and IDEAL algorithms is made in section 4. The conclusions are drawn in section 5.

2. Governing equations and discretization equations

2.1. Governing equations for compressible flows

The governing equation for two dimensional viscous compressible flows in the Cartesian coordinate is given as:

$$\frac{\partial(\rho\phi)}{\partial t}+\frac{\partial(\rho u\phi)}{\partial x}+\frac{\partial(\rho v\phi)}{\partial y}=\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)-R_\phi+S_\phi \tag{1}$$

In case of steady flows, the first item on the left side of the equation equals to zero. The symbols in Eq. (1) are listed in Table 1.

In addition, the state equation is included into the governing equation, which is written as:

$$P=\rho RT \tag{2}$$

In pressure-based method, density is calculated from pressure:

$$\rho=\frac{P}{RT} \tag{3}$$

2.2. Discretization equations

Finite volume method (FVM) is used to discretize the governing equations. The one- dimensional derivation process is taken as an example. It can be easily extended for two-dimensional and three-dimensional flows using the same methodology.

The momentum equation is discretized as,

$$a_e u_e=\sum a_{nb}u_{nb}+A_e(p_P-p_E)+b \tag{4}$$

$$a_w u_w=\sum a_{nb}u_{nb}+A_w(p_W-p_P)+b \tag{5}$$

where subscript e, w are the east and west interfaces of control volume P, respectively (Figure 1). Node P lies at the center of the control volume. W, E are the neighbor nodes of control volume $Pn.b$ stands for the neighbors of the point to be solved. Efficient a includes the effects of convection and diffusion terms. A is the area where pressure difference affects, and b is the source term.

The discretized continuity equation is given by:

$$\rho_e u_e A_e-\rho_w u_w A_w+\frac{\rho-\rho^0}{\Delta t}\Delta V=0 \tag{6}$$

where superscript 0 indicates the solutions from previous time step (unsteady flow) or previous outer iteration (steady flows). A denotes the area where the flow goes through. ΔV is volume of a control.

Table 1. Symbols in different equations.

Equations	Continuity	Momentum-u	Momentum-v	Temperature
ϕ	1	u	v	T
Γ	0	μ	μ	μ/Pr

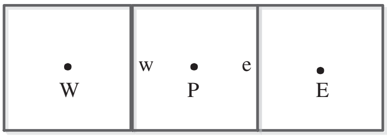

Figure 1. Control volumes, center nodes, and interfaces.

The temperature equation can be discretized into the following form:

$$A_P T_p = \sum A_{nb} T_{nb} + b \tag{7}$$

where A is coefficient that combines convection and diffusion effects.

The pressure correction equation is derived from discretized continuity and momentum equations. Pressure and velocity fields are updated from pressure correction. The derivation of pressure correction equation is shown as follows.

The discretized momentum equation of velocity u is rewritten:

$$a_e u_e^* = \sum a_{nb} u_{nb}^* + A_e \left(p_P^0 - p_E^0 \right) + b \tag{8}$$

After solving Eq. (8), the velocity u^* is obtained.

The explicit discretized momentum equation is written:

$$a_e u_e^{**} = \sum a_{nb} u_{nb}^* + \left(p_P^* - p_E^* \right) A_e + b \tag{9}$$

Subtract Eq. (8) from Eq. (9) to get:

$$a_e \left(u_e^{**} - u_e^* \right) = \left[\left(p_P^* - p_P^0 \right) - \left(p_E^* - p_E^0 \right) \right] A_e = \left(p'_P - p'_E \right) A_e \tag{10}$$

where $p_P^* - p_P^0 = p'_P$ and $p_E^* - p_E^0 = p'_E$

Rearrange Eq. (10) to give:

$$u_e^{**} = \left(p'_P - p'_E \right) A_e / a_e + u_e^* = \left(p'_P - p'_E \right) d_e + u_e^* \tag{11}$$

where $d_e = A_e / a_e$

Similarly, the velocity on west interface is given as:

$$u_w^{**} = \left(p'_W - p'_P \right) d_w + u_w^* \tag{12}$$

where $d_w = A_w / a_w$

The discretized continuity equation is rewritten as:

$$\rho_e{}^{**} u_e{}^{**} A_e - \rho_w{}^{**} u_w{}^{**} A_w + \frac{\rho^{**} - \rho^0}{\Delta t} \Delta V = 0 \tag{13}$$

The following equation is defined:

$$\rho^{**} = \rho^* + \rho' \tag{14}$$

According to Eq. (3):

$$\rho' = P' / RT \tag{15}$$

Substituting Eq. (15) in Eq. (14), there is:

$$\rho^{**} = \rho^* + P' / RT \tag{16}$$

Substitute Eqs. (11)–(12), and (16) into Eq. (13) to get pressure correction equation:

$$A_P P'_P = A_E P'_E + A_W P'_W + b \tag{17}$$

Coefficients A_P, A_E, A_w are dependent on density scheme. The FUD density scheme [10] is used as an example:

In the case $u > 0$:

$$A_P = \rho_e{}^* A_e d_e + \rho_w{}^* A_w d_w + \frac{1}{RT_P} u_e^* A_e + \frac{1}{RT_P} \frac{\Delta V}{\Delta t} \tag{18}$$

$$A_E = \rho_e{}^* A_e d_e \tag{19}$$

$$A_W = \rho_w{}^* A_w d_w + \frac{1}{RT_W} u_w^* A_w \tag{20}$$

$$b = \rho_w^* u_w^* A_w - \rho_e^* u_e^* A_e - \frac{\rho^* - \rho^0}{\Delta t} \Delta V \tag{21}$$

The items related to time disappear in solutions of steady flows.

After solving the pressure correction Eq. (17), pressure is updated by:

$$P^{**} = P^* + P' \tag{22}$$

Density is calculated from the state equation:

$$\rho^{**} = \frac{1}{RT} P^{**} \tag{23}$$

or

$$\rho^{**} = \rho^* + \frac{1}{RT} P' \tag{24}$$

At this point, we get all the discretized equations for velocity, temperature, pressure, and density.

To ensure a convergent solution, the under-relaxation technique is utilized. The under-relaxation form of discretized equation is written as:

$$\frac{A_P}{\alpha} \phi_p = \sum A_{nb} \phi_{nb} + b + \frac{1-\alpha}{\alpha} A_P \phi_p^0 \tag{25}$$

where ϕ stands for the variable to be solved. A are the coefficients. b is the source term. $\alpha$is the under-relaxation factor. Superscript 0 is variable from previous time step or previous outer iteration.

3. Derivation of adaptive inner iteration processes

In this part, two inner iteration processes for pressure correction equations are firstly presented in detail. Then, the implementation of adaptive inner iteration processes is written. At the end of this part, the overall procedure including adaptive inner iterations is shown in a flow chart.

3.1. Inner iteration processes for pressure correction equation

As shown in Eq. (8), the momentum equation is solved using pressure from the previous time step (unsteady flow) or previous outer iteration (steady flows), leading to a poor coupling between velocity and pressure. To solve this problem, the pressure correction equation is solved to update pressure field before the momentum equation is solved. The detail of the first inner iteration process for pressure correction is as follows:

The first inner iteration process:

u_e^0, u_w^0, ρ^0, and P^0 which are obtained from initial fields or from the previous time step are used for computing the coefficients of pressure correction equation:

$$A_P P'^{1,1}_P = A_E P'^{1,1}_E + A_W P'^{1,1}_W + b \tag{26}$$

where the first number 1 in the superscript means the first inner iteration and the second number 1 means the first time calculation.

After Eq. (26) is solved, $P'^{1,1}$is used to update velocity and density using Eqs. (11), (12), and (24) which are rewritten as:

$$tu_e^{1,1} = u_e^0 + \left(p'^{1,1}_P - p'^{1,1'}_E \right) d_e \tag{27}$$

$$u_w^{1,1} = u_w^0 + \left(p'^{1,1}_W - p'^{1,1}_P\right) d_w \tag{28}$$

$$\rho^{1,1} = \rho^0 + \frac{1}{RT} P'^{1,1} \tag{29}$$

Update coefficients and source term using $\rho^{1,1}$, $u_e^{1,1}$, and $u_w^{1,1}$ to solve pressure correction equation for the second time:

$$A_P P'^{1,2}_P = A_E P'^{1,2}_E + A_W P'^{1,2}_W + b \tag{30}$$

$u_e^{1,2}$, $u_w^{1,2}$, and $\rho^{1,2}$ are updated using Eqs. (11), (12), and (24). The coefficients and the source term are calculated again with the updated velocity and density to solve pressure correction equation for the third time. n steps are repeated until the first inner iteration process ends, at which stage the pressure $P^* = P^{1,n}$ is taken as the final pressure.

The intermediate density field is achieved from $\rho^* = \frac{1}{RT^0} P^*$. The final density field could be obtained after the temperature equation is solved.

The pressure and density fields are improved after the first inner iterative process ends, making the coupling between velocity, pressure, and density better in the solutions of compressible flows.

The momentum equations with the improved pressure field are given as:

$$a_e u_e^* = \sum a_{nb} u_{nb}^* + A_e\left(p_P^* - p_E^*\right) + b \tag{31}$$

$$a_w u_w^* = \sum a_{nb} u_{nb}^* + A_w\left(p_W^* - p_P^*\right) + b \tag{32}$$

The implicit momentum equations are solved for intermediate velocity u_e^* and u_w^*. The coupling between velocity and pressure is highly improved using the updated pressure instead of the pressure from assumed initial condition so that the efficiency and accuracy of calculations are optimized.

The second inner iteration process:

The velocity field obtained from Eqs. (31) and (32) could not in general satisfy the mass conservation. To make the continuity equation well satisfied, another inner iteration process is required.

$$A_P P'^{2,1}_P = A_E P'^{2,1}_E + A_W P'^{2,1}_W + b \tag{33}$$

where the first number 2 in superscript stands for the second inner iteration, and the second number 1 means the first calculation in this iterative process. u_e^*, u_w^*, and ρ^* obtained from the previous step are used to calculate coefficients A and source term b of pressure correction Eq. (33).

Eq. (33) is solved for $P'^{2,1}$ which is then substituted into Eq. (11) and Eq. (12):

$$u_e^{2,1} = \left(p'^{2,1}_P - p'^{2,1}_E\right) d_e + u_e^* \tag{34}$$

$$u_w^{2,1} = \left(p'^{2,1}_W - p'^{2,1}_P\right) d_w + u_w^* \tag{35}$$

A new velocity field is generated to calculate coefficients and source term of the second pressure correction equation:

$$A_P P'^{2,2}_P = A_E P'^{2,2}_E + A_W P'^{2,2}_W + b \tag{36}$$

After solving the pressure correction equation for the second time, the velocity field using expressions (11) and (12) is renewed. The coefficients and source term of pressure correction equation are calculated again with the updated velocity to solve the pressure correction equation for the third time.

The second inner iterative process is completed after repeating n steps. At this stage the velocity $u_e^{2,n}$ and $u_w^{2,n}$ are taken as the final velocity field.

The existence of the second inner iteration process is necessary for improving the accuracy of solutions. As shown in the work of reference [11], a minimum of two corrector steps is taken to legitimately obtain the velocity and pressure fields which satisfy the momentum and continuity equations. It is also suggested in the paper [12] that the pressure correction equation should be solved by two or three times to assure the continuity equation satisfied.

The temperature equation is solved in the implicit form after getting the final velocity:

$$A_P T_P = A_E T_E + A_W T_W + b \tag{37}$$

The temperature field T is obtained after Eq. (37) is solved.

The final density field is obtained from the state equation $\rho^{**} = \frac{1}{RT} P^*$.

3.2. Implementation of adaptive inner iterations

In IDEAL algorithm, the number of first inner iteration and the number of second inner iteration are fixed as 4, which is not always appropriate for different computational cases. To solve this problem, a mechanism is proposed to control inner iteration processes so that the number of inner iteration changes with computational progress in different computations.

As the first inner iteration process is to solve pressure correction equations to get a reasonable pressure field, the residual norm of pressure correction equation is used to trace the error of the pressure correction equation.

The residual norm of pressure correction is written as (One dimensional form is taken as an example):

$$R_{p'}{}^{(n)} = \left\{ \sum \left[\left(A_E P'_E + A_W P'_W + b - A_P P'_P \right)^{(n)} \right]^2 \right\}^{\frac{1}{2}} \tag{38}$$

where superscript n stands for the nth time of calculation in the inner iteration. The first inner iterative process ends when the criterion $R_{p'}{}^{(n)} / R_{p'}{}^{(0)} < 1$ is satisfied.

The second inner iteration is to get velocity fields, which satisfy the continuity equation. The global mass residual is used to check the error of mass conservation.

The global mass residual (1-D) is written as:

$$R_m{}^{(n)} = \sum \left(\rho_e u_e A_e - \rho_w u_w A_w + \left(\rho - \rho^0 \right) \Delta V / \Delta t \right)^{(n)} \tag{39}$$

where superscript n stands for the nth time calculation in the second inner iteration process. The second inner iterative process ends when the criterion $R_m{}^{(n)} / R_m{}^{(0)} < 1$ is satisfied.

If the criteria cannot be satisfied, a maximum number should be set in the procedure to avoid a limitless calculation. In the present work, the maximum number is set as 4 based on a number of tests.

By using this mechanism to control inner iteration processes, at least two inner iteration steps are conducted, which keeps the feature of iteration in the computation. At the same time, errors of the pressure correction equation and continuity equation are under surveillance and control.

3.3. Overall procedure

At this point, all the ingredients including adaptive inner iterations are available and reorganized as follows:

1. Velocity, pressure, temperature, and density fields are obtained from the initial condition or the previous time step.

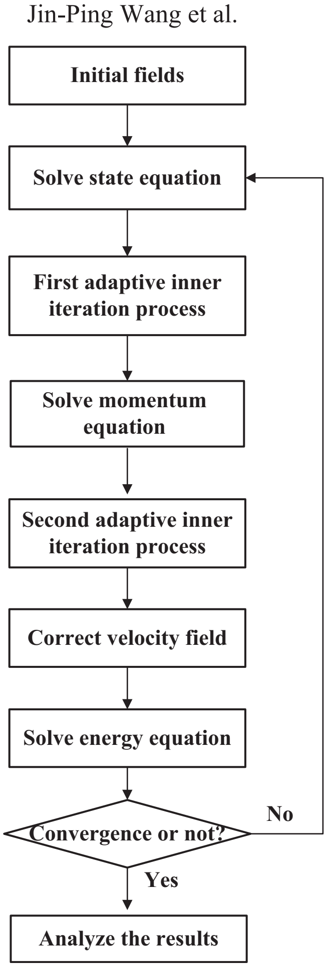

Figure 2. Flow chart of compressible procedure including adaptive inner iterations.

2. Coefficients and source term of the pressure correction equation are calculated.
3. The first adaptive inner iterative process is conducted to update the pressure and density fields.
4. The momentum equation is solved to obtain the intermediate velocity field.
5. The coefficients and source term of pressure correction equation are renewed using the intermediate velocity and updated density.
6. The second adaptive inner iterative process is performed to generate the final velocity field.
7. The temperature field is achieved after solving the temperature equation.
8. The steps 2 through 7 are repeated until the convergence is obtained for steady compressible flows. Return to step1 and repeat for unsteady compressible flows.

The flow chart of the procedure is shown in Figure 2.

4. Verification and comparison of adaptive inner iteration processes

In this section, the validity of the adaptive inner iteration processes is evaluated. To test the performance, a comparison is made with IDEAL and CLEAR algorithms.

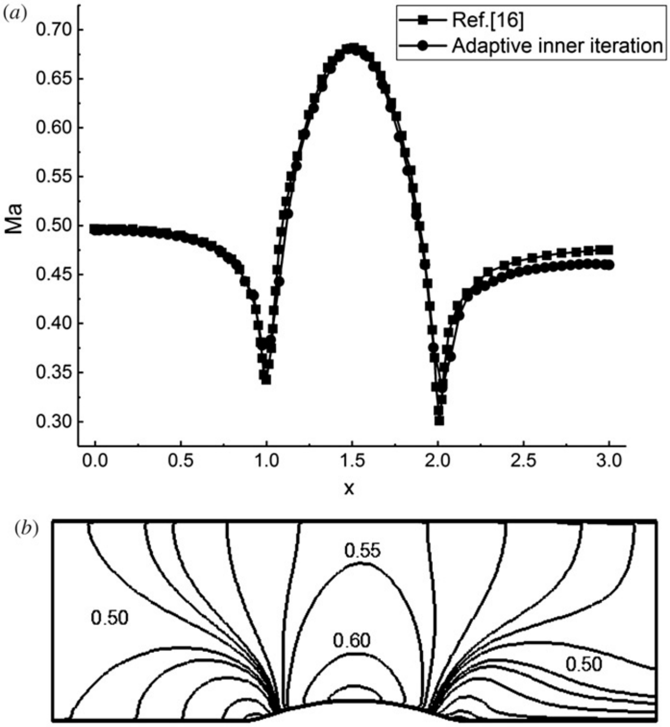

Figure 3. Results in subsonic flow over a bump using adaptive inner iteration processes. (*a*) Mach number profiles along the bump wall. (*b*) Mach number distribution.

4.1. Verification of the new algorithm

By solving the two dimensional viscous compressible flows (including flow over a bump and flow through a nozzle), solutions of the adaptive inner iteration processes are compared with experimental data and numerical results in literatures.

4.1.1. Two-dimensional viscous flow over a bump

Three different types of flows (subsonic, transonic and supersonic flows) in a channel with a circular arc bump are computed. The test cases were used by several researchers [13–15]. The width of the channel is equal to the length of the bump, and the channel length is equal to three times of the bump length. The thickness-to-chord ratio is 10% for subsonic and transonic flows and 4% for supersonic flow. The detail parameters of boundary conditions for these flows can be checked in the reference [7].

Subsonic flow:

Figure 3(*a*) presents the Mach number profile along the lower wall of the channel in subsonic flow. The agreement between the adaptive inner iteration solution and the result in the reference [16] is observed.

Figure 3(*b*) shows the predicted Mach number distributions in the present solution. The distributions are symmetric which confirms the expectation in this subsonic flow. The predictions match with those in the references [17–19].

Transonic flow:

The Mach number distributions for transonic flow are shown in Figure 4(*a*). It is seen that the distributions are asymmetric. The location of the shock is $x \simeq 1.72$ which is almost identical to that in Date's work [17].

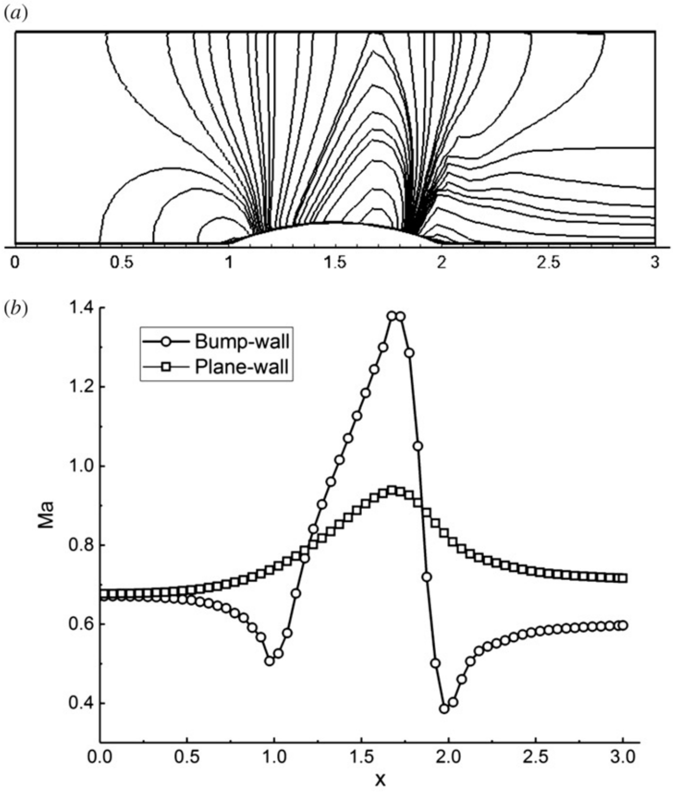

Figure 4. Predictions in transonic flow over a bump using adaptive inner iteration processes. (*a*) Mach number distributions. (*b*) Mach number profiles along the bump wall and plane channel wall.

Figure 4(*b*) displays the Mach number profiles along the bump wall and plane channel wall. The predicted maximum Mach number is 1.378, which is 1.41 in paper [16] and 1.332 in paper [17].

Supersonic flow:

Computations are performed for an inlet Mach number $M_{in} = 1.65$ in the flow over a bump. Figure 5 shows the Mach number profiles along the bump wall of the channel. Compared with the result in Moukalled and Darwish's work [16], the significant differences occur at the trailing edge of the bump and the channel exit, as seen in Figure 5. The relative error is 6.03% at the trailing edge and 5.41% at the channel exit.

4.1.2. Two dimensional viscous flow through a nozzle

The geometry of the nozzle is: the total length of the Laval is 11.56 cm; the half-height at the Laval throat is 1.37 cm; the half-height at the Laval inlet is 3.52 cm, and the half-height at the Laval exit is 1.49 cm. The stagnation pressure and temperature are prescribed at the inlet. A static-to-stagnation pressure ratio of 0.3367 is defined at the outlet.

Figure 6 shows the predicted solutions using adaptive inner iteration processes. The computed results are in good agreement with the experimental data along the centerline of the nozzle, as seen in Figure 6. The relative error is 5.64% between the predicted results and the experimental data [20].

From the above discussions, the validity of adaptive inner iteration is demonstrated. In the following part, the performance of adaptive inner iteration is tested.

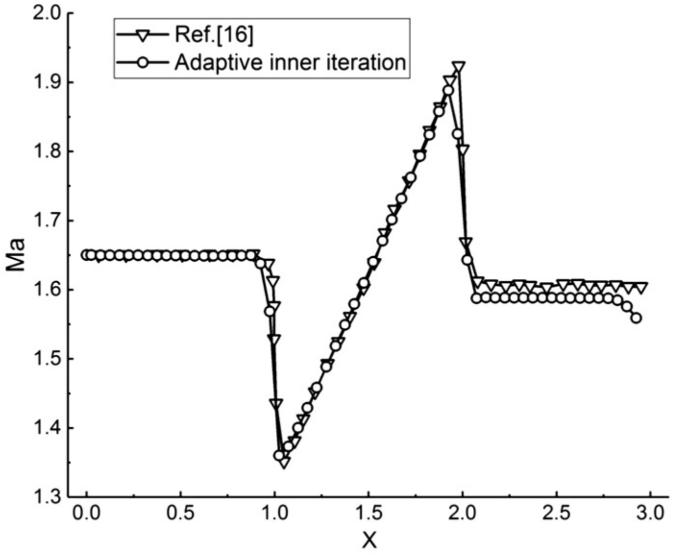

Figure 5. Mach number profiles along the bump wall in supersonic flow using adaptive inner iteration processes.

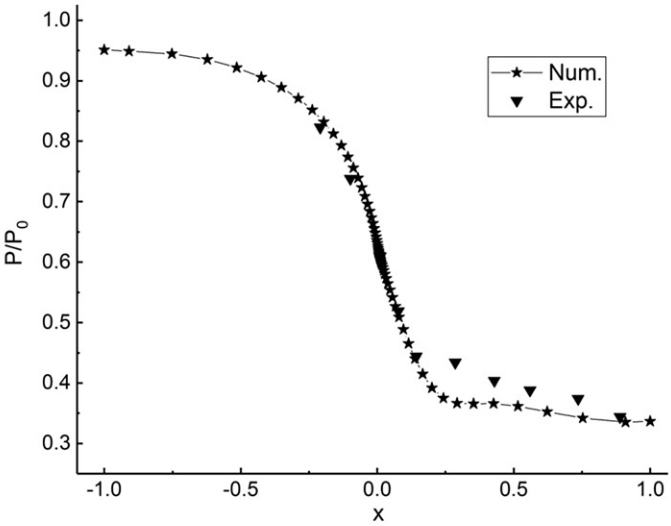

Figure 6. Pressure distribution along the nozzle centerline using adaptive inner iteration.

4.2. Comparison of adaptive inner iteration with IDEAL and CLEAR algorithms

CLEAR and IDEAL algorithms are proved to be efficient and robust in the published work [7,8]. There exists no inner iteration processes in CLEAR algorithm while the inner iteration steps in IDEAL algorithm are fixed (Usually numbers of inner iterations is 4 for first inner iteration and 4 for second iteration). To evaluate the performance of the proposed adaptive inner iteration, a comparison with these two methods is made.

To facilitate the comparison, the time step multiple E is introduced. The relation between α and E is defined as:

$$E = \frac{\alpha}{1 - \alpha} \quad (0 < \alpha < 1) \tag{40}$$

In this work, the range of under-relaxation factors is from 0.2 to 0.8, and the corresponding time step E is from 0.25 to 4.0.

4.2.1. Two-dimensional viscous flow over a bump

Subsonic flow:

The computing time using adaptive inner iteration, IDEAL algorithm, and CLEAR algorithm in the solution of subsonic flow over a bump is presented in Table 2. The alphabet D in the table means "Divergent solution". It is seen that solution of IDEAL algorithm is not convergent at an under-relaxation factor 0.7 or 0.8. The solutions of adaptive inner iteration and CLEAR algorithm are convergent in the whole region of under-relaxation factors, which illustrates the better robustness of adaptive inner iteration processes and CLEAR algorithm than IDEAL algorithm.

The comparison of computing time is shown in Figure 7. The average computing time is 28.05 s, 27.45 s, and 30.91 s using adaptive inner iteration processes, IDEAL algorithm, and CLEAR algorithm. Obviously, CLEAR algorithm is the most expensive program in this case.

To know how the number of inner iterations changes with computational progress in the adaptive inner iteration processes, variation of the inner iteration number during the computational progress at an under-relaxation factor of 0.8 is shown in Figure 8(a). It is seen that the number of first inner iteration stays as 3 and 4 for most of the time. The number of the second

Table 2. Computing time in subsonic flow over a bump using adaptive inner iteration, IDEAL algorithm and CLEAR algorithm.

	Computing times (s)		
E	Adaptive inner iteration	IDEAL	CLEAR
0.2500	43.29	41.82	47.11
0.4286	33.48	32.60	43.49
0.6667	29.17	28.33	9.09
1.0000	27.85	26.88	32.26
1.5000	24.93	7.63	18.16
2.3333	18.86	D	33.28
4.0000	18.08	D	32.95

D: divergent solution.

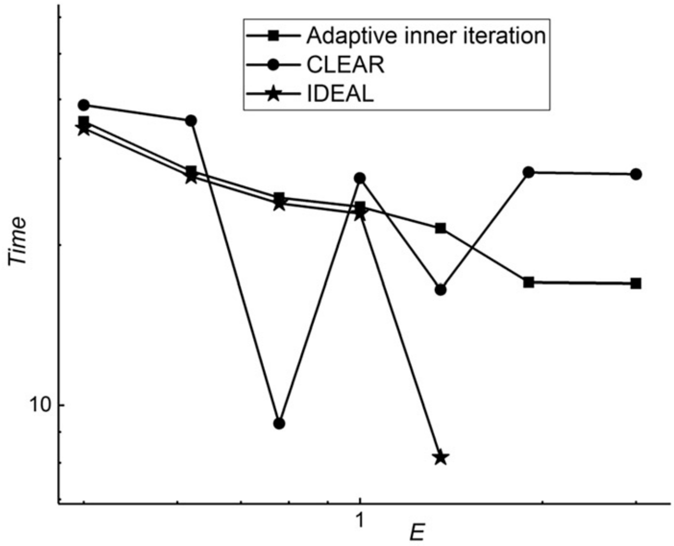

Figure 7. Comparison of computing time in subsonic flow over a bump.

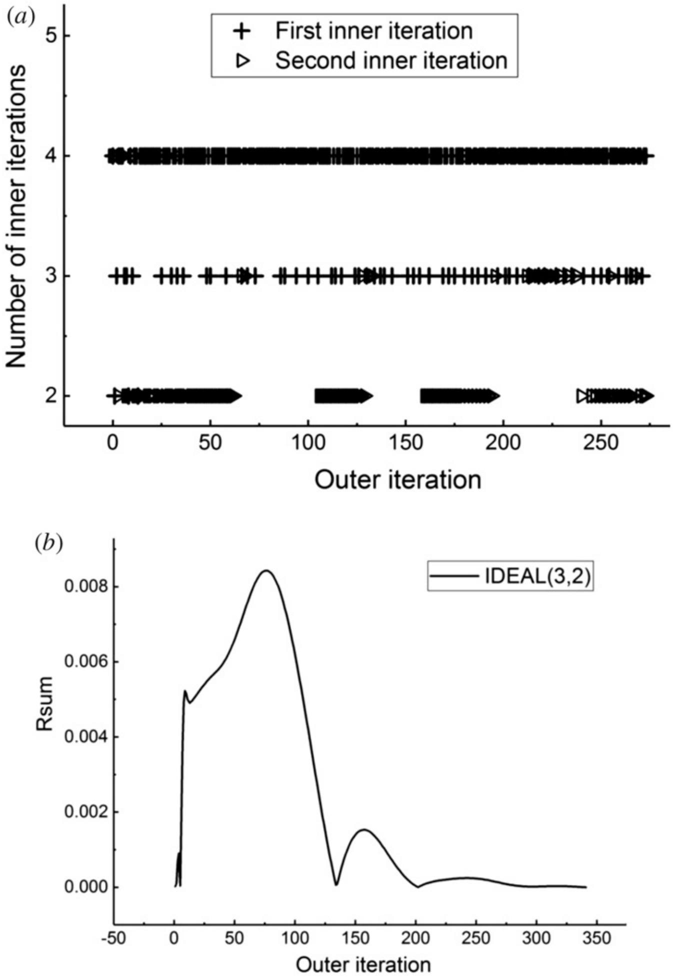

Figure 8. Use the numbers of adaptive inner iteration to reset the number of inner iteration in IDEAL algorithm ($\alpha = 0.8$) in subsonic flow over a bump. (*a*) Variation of the inner iteration numbers with computational progress at $\alpha = 0.8$. (*b*) Convergence history for IDEAL(3,2) at $\alpha = 0.8$.

inner iteration stays as 2 and 4 in most of the cases. Based on the numbers from the adaptive inner iteration solutions, the number of first inner iteration is reset as 3 and the number of second inner iteration is reset as 2 in IDEAL algorithm at an under-relaxation factor 0.8. It is found that when changing the fixed times from (4,4) (the first 4 in brackets denotes the number of first inner iteration and the second 4 is the number of second inner iteration in IDEAL algorithm) to (3,2) at an under-relaxation factor 0.8, the solution of IDEAL algorithm becomes convergent, showing the importance of inner iteration numbers in compressible flow solutions. Figure 8(*b*) shows the convergence history for IDEAL (3,2) at $\alpha = 0.8$. The computing time is 18.56 s which is a little more than that of adaptive inner iteration solution.

Transonic flow:

The computing time using adaptive inner iteration, IDEAL algorithm, and CLEAR algorithm in transonic flow over a bump is counted in Table 3. The solutions of IDEAL algorithm are not convergent at an under-relaxation factor 0.5, 0.6, 0.7 or 0.8, implying the worst robustness of IDEAL algorithm among these three methods.

The CPU time of these three methods is re-presented in Figure 9. It can be observed that the procedure of adaptive inner iteration costs much less time than CLEAR algorithm in the full

Table 3. Computing time in transonic flow over a bump using adaptive inner iteration, IDEAL algorithm and CLEAR algorithm.

	Computing times (s)		
E	Adaptive inner iteration	IDEAL	CLEAR
0.2500	64.76	63.87	70.92
0.4286	50.89	51.50	72.57
0.6667	45.22	44.93	74.32
1.0000	42.81	D	77.06
1.5000	42.12	D	76.00
2.3333	42.53	D	81.54
4.0000	43.66	D	87.06

D: divergent solution.

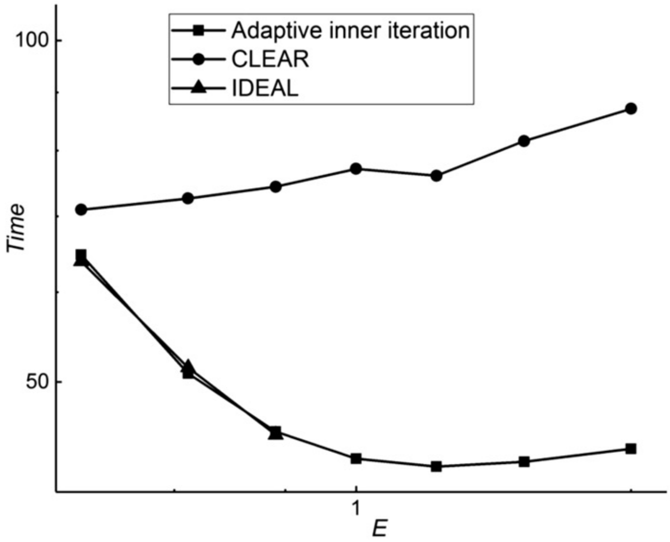

Figure 9. Comparison of computing time in transonic flow over a bump.

range of the under-relaxation factors. The computing time of IDEAL algorithm is almost the same as that of adaptive inner iteration processes in its convergent range.

The average time cost by CLEAR algorithm, IDEAL algorithm, and adaptive inner iteration are 77.07 s, 53.43 s, and 47.43 s, respectively. Compared with CLEAR algorithm, the efficiency is improved by 62.5% using the adaptive inner iteration.

Figure 10(*a*) shows the variation of inner iteration numbers in the solution of adaptive inner iteration processes at an under-relaxation factor 0.5. The number of the first inner iteration is 4 at the most of time and the number of the second inner iteration varies from 2 to 4. Based on the results, the number of inner iterations in IDEAL algorithm is reset as 4 (first inner iteration) and 2 (second inner iteration). It is noted that the solution of IDEAL algorithm at an under-relaxation factor 0.5 becomes convergent by adjusting inner iterative numbers. Figure 10(*b*) shows the convergence history of this solution. The computing time cost in the solution of IDEAL (4, 2) is 44.69 s which is more than that of adaptive inner iteration solution, showing the advantage of adaptive inner iteration on computational efficiency.

Supersonic flow:

Table 4 shows the CPU time using adaptive inner iteration, IDEAL algorithm, and CLEAR algorithm in supersonic flow over a bump. It should be noted that the solution of IDEAL

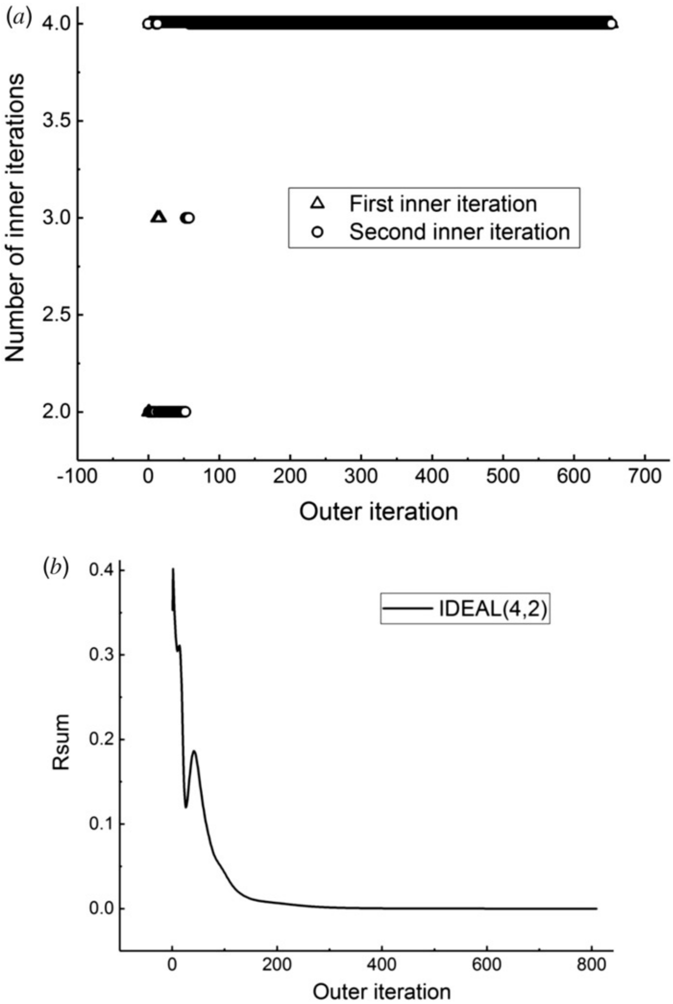

Figure 10. Use the numbers of adaptive inner iteration to reset the number of inner iteration in IDEAL algorithm ($\alpha = 0.5$) in transonic flow over a bump. (*a*) Variation of the inner iteration numbers with computational progress at $\alpha = 0.5$. (*b*) Convergence history for IDEAL(4,2) at $\alpha = 0.5$.

algorithm is convergent only at an under-relaxation factor 0.5 and divergent at other under-relaxation factors, showing the poor robustness of IDEAL algorithm in compressible supersonic flows.

Figure 11 presents the comparison of computing time by adaptive inner iteration and CLEAR algorithm. The red line of adaptive inner iteration is below the blue line of CLEAR algorithm which illustrates the higher computing efficiency of the adaptive inner iteration.

The average time is 10.74 s and 16.76 s for using adaptive inner iteration and CLEAR algorithm, respectively. Compared with CLEAR algorithm, the computational efficiency is improved by 56.1% using adaptive inner iteration.

Figure 12(*a*) shows variation of the inner iteration numbers in the adaptive inner iteration solution at an under-relaxation factor 0.2. It is seen that the number of first inner iteration is 2 and the number of second inner iteration is either 2 or 4. Based on the numbers, the inner iterative numbers in IDEAL algorithm is reset as 2 for the first inner iteration and 4 for the second inner iteration, and the convergence history is shown in Figure 12(*b*). The computing time in this case is 14.79 s which is a little more than that of adaptive inner iteration solution.

Table 4. Computing time in supersonic flow over a bump using adaptive inner iteration, IDEAL algorithm and CLEAR method.

	Computing times (s)		
E	Adaptive inner iteration	IDEAL	CLEAR
0.2500	14.63	D	18.74
0.4286	9.44	D	19.97
0.6667	9.88	D	13.67
1.0000	8.50	7.50	13.34
1.5000	9.72	D	20.45
2.3333	11.20	D	14.84
4.0000	11.78	D	16.32

D: divergent solution.

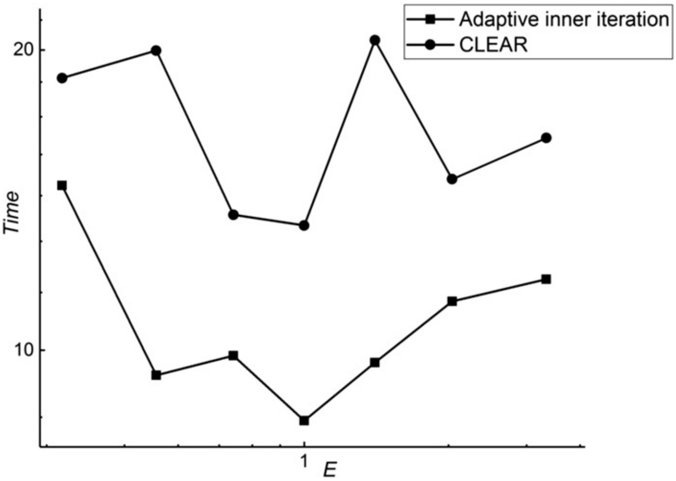

Figure 11. Comparison of computing time in supersonic flow over a bump.

4.2.2. Two dimensional viscous flow through a nozzle

The computing time of the adaptive inner iteration, IDEAL algorithm, and CLEAR algorithm in the solutions of flow through a nozzle is presented in Table 5. The solution of IDEAL algorithm is not convergent at an under-relaxation factor 0.7 or 0.8. The solution of CLEAR method is divergent at an under-relaxation factor 0.8. The results suggest the best robustness of the adaptive inner iteration solution.

The comparison of computing time between adaptive inner iteration, IDEAL algorithm, and CLEAR algorithm in flow through a nozzle is shown in Figure 13. The adaptive inner iteration line is almost coincident with the IDEAL algorithm line, suggesting the nearly same computing time by using these two methods. The CLEAR algorithm line is on the top of the figure, illustrating the most expensive method in this case.

The average computing time is 23.30 s, 20.00 s, and 53.92 s by using adaptive inner iteration processes, IDEAL method, and CLEAR algorithm, respectively. Compared with solution of the CLEAR algorithm, 1.31 times computing time is saved by using adaptive inner iteration.

Figure 14(*a*) is the variation of inner iteration numbers in the adaptive inner iteration solution of flow through a nozzle at an under-relaxation factor 0.7. It is seen that the number of first inner iteration varies from 2 to 4 and the number of second inner iteration is 4. Based on the results, the inner iterative numbers in IDEAL algorithm is reset as 3 for the first inner iteration and 4 for the second inner iteration, and the convergence history is shown in Figure 14(*b*). The computing time in this case is 36.22 s, which is more than that of adaptive inner iteration.

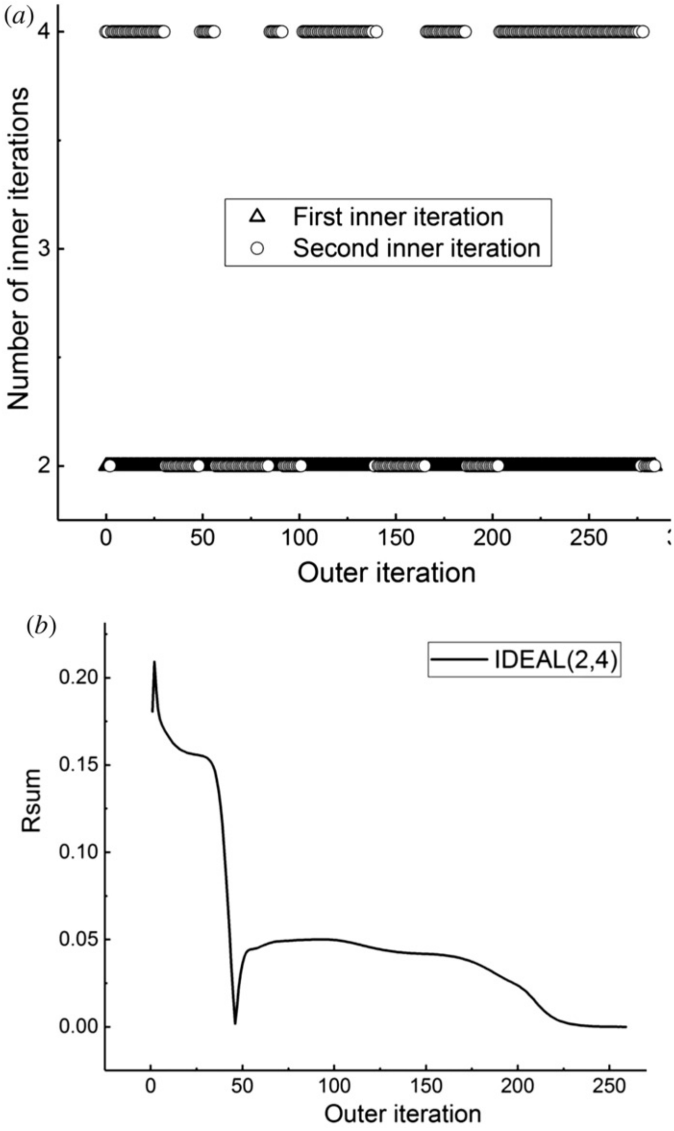

Figure 12. Use the numbers of adaptive inner iteration to reset the number of inner iteration in IDEAL algorithm ($\alpha = 0.2$) in supersonic flow over a bump. (*a*) Variation of the inner iteration numbers with computational progress at $\alpha = 0.2$. (*b*) Convergence history for IDEAL (2,4) at $\alpha = 0.2$.

Table 5. Computing time in the flow through a nozzle using adaptive inner iteration, IDEAL algorithm and CLEAR algorithm.

	Computing times (s)		
E	Adaptive inner iteration	IDEAL	CLEAR
0.2500	19.03	18.86	24.76
0.4286	16.26	16.27	33.13
0.6667	18.27	18.22	43.56
1.0000	21.59	21.56	63.18
1.5000	25.19	25.07	71.17
2.3333	29.25	D	87.73
4.0000	33.51	D	D

D: divergent solution.

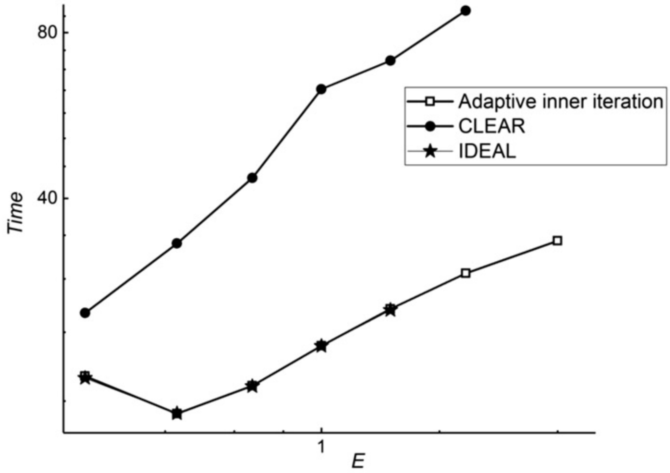

Figure 13. Comparison of computing time in flow through a nozzle.

5. Conclusions

The adaptive inner iteration processes based on pressure-based method for viscous compressible flows are presented with the number of inner iteration varying adaptively with computational progress based on different problems. This adaptive inner iteration is applied to the calculations of flow over a bump (subsonic, transonic, and supersonic flows) and flow through a nozzle. The results of these calculations amply demonstrate that the method is of good accuracy in compressible flows.

The performance of adaptive inner iteration in subsonic, transonic, and supersonic flows is compared with CLEAR algorithm (no inner iteration process) and IDEAL algorithm (fixed inner iteration (4,4)). In subsonic flow over a bump, IDEAL algorithm is not convergent at an under-relaxation factor 0.7 or 0.8. In the solutions of transonic flow over a bump, IDEAL algorithm is not convergent at an under-relaxation factor 0.5, 0.6, 0.7 or 0.8. The efficiency is improved by 62.5% using adaptive inner iteration compared with CLEAR algorithm. In supersonic flow over a bump, the solution of IDEAL algorithm is convergent only at an under-relaxation factor 0.5 and divergent at other under-relaxation factors. The efficiency is improved by 56.1% using the adaptive inner iteration compared with CLEAR algorithm. In solutions of flow through a nozzle, IDEAL algorithm is not convergent at an under-relaxation factor 0.7 or 0.8, and CLEAR method is divergent at an under-relaxation factor 0.8, but adaptive inner iteration solutions are convergent at all the under-relaxation factors.

From the above results, it is concluded that:

- The computational efficiency is highly improved by using adaptive inner iteration processes in compressible flows.
- In the compressible solutions of IDEAL algorithm (fixed inner iteration), several computations are not convergent. However, this issue is overcome by using adaptive inner iteration, which illustrates a good robustness.

Though the adaptive inner iteration processes are proposed based on IDEAL algorithm, the mechanism developed to control inner iteration processes depends on the features of iteration calculation and physical meaning of inner iterations processes. Therefore, the method to achieve adaptive inner iteration is not limited only for IDEAL algorithm. In any method where inner iteration processes exist, this mechanism can be used. Furthermore, the adaptive inner iteration processes can not only work for compressible flows, but also work well in incompressible flows.

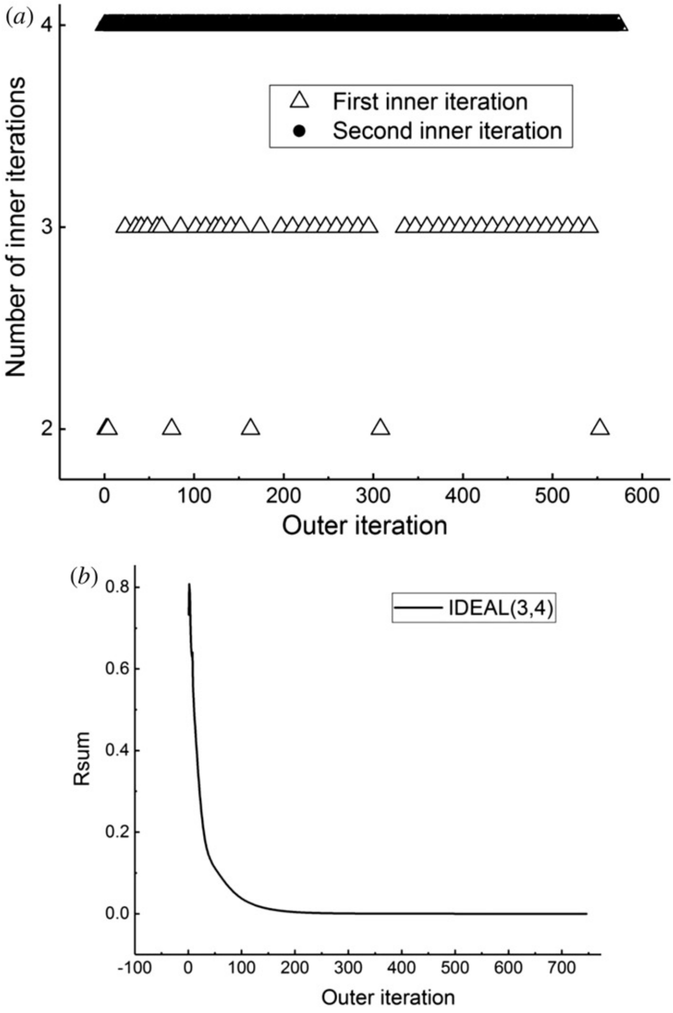

Figure 14. Use the numbers of adaptive inner iteration to reset the number of inner iteration in IDEAL algorithm ($\alpha = 0.7$) in flow through a nozzle. (*a*) Variation of the inner iteration numbers with computational progress at $\alpha = 0.7$. (*b*) Convergence history for IDEAL(3,4) at $\alpha = 0.7$.

Funding

The present work was supported by the National Natural Science Foundation of China [Grant No. 51206129]." to "The present work was supported by the National Natural Science Foundation of China [Grant No. 51576155] and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China [Grant No.51721004].

Reference

[1] S.V. Patankar, and D.B. Spalding, "A calculation procedure for heat mass and momentum transfer in three dimensional parabolic flows," *Int. J. Heat Mass Transfer*, vol. 15, no. 10, pp.1787–1806, 1972.

[2] S.V. Patankar. "A calculation procedure for two-dimensional elliptic situations," *Numerical. Heat Transfer*, vol. 4, no. 4, pp. 409–425, 1981.

[3] M. H. Hu, J. S. Wu and Y.S. Chen, "Development of a parallelized 2D/2D-axisymmetric Navier–Stokes equation solver for all-speed gas flows," *Comput Fluids*, vol. 45, pp. 241–248, 2011.

[4] Z. X. Sun. "The Application of Direct Simulation Monte Carlo Method for Solving Multi-scale Problems," Ph.D. thesis, Xi'an Jiaotong University, Xi'an, Shaanxi, PRC, 2011. (in Chinese)

[5] W.Q. Tao, Z.G. Qu, and Y.L. He, "A novel segregated algorithm for incompressible fluid flow and heat transfer problems—CLEAR (coupled and linked equations algorithm revised) part I: Mathematical formulation and solution procedure," *Numer. Heat Transf. B*, Vol. 45, no. 1, pp.1–17, 2004.

[6] Z.G. Qu. "Study on Advanced Numerical Algorithms on Fluid Flow and Heat Transfer Problems and Their Applications in Air Convective Heat Transfer Enhancement," Ph.D. thesis, Xi'an Jiaotong University, Xi'an, Shaanxi, PRC, 2005.(in Chinese)

[7] J. P. Wang, J. F. Zhang, Z. G. Qu et al. "Comparison of robustness and efficiency for SIMPLE and CLEAR algorithm with 13 high-resolution convection schemes in compressible flows," *Numerical Heat Transfer, part B: Fundamentals*. vol. 66, no. 2, pp. 133–161, 2014.

[8] D. L. Sun, Z. G. Qu, Y. L. He, and W. Q. Tao. "An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL(inner doubly iterative efficient algorithm for linked equations) part I: Mathematical formulation and solution procedure," *Numerical Heat Transfer, Part B*, vol. 53, no. 1, pp.1–17, 2008.

[9] D. L. Sun, "Development of Advanced Velocity-Pressure Coupling Algorithm and Interface Capturing Method," Ph.D. thesis, Xi'an Jiaotong University, Xi'an, Shaanxi, PRC, 2009.(in Chinese)

[10] J. F. Zhang, J. P. Wang, Z. G. Qu et al. "The study of different discretized schemes for density and convection terms in high speed compressible flow using the pressure-based method," *Appl. Therm. Eng.*, vol. 73, no. 2, pp. 1533–1540, 2014.

[11] R. Issa. "Solution of the implicit discretized fluid flow equations by operator splitting," *J Comput. Phys.*, vol. 62, no. 1, pp.40–65, 1986.

[12] K. C. Karki and S. V. Patankar. "Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations," *Aiaa J.*, vol. 27, no. 9, pp. 1167–1174, 1989.

[13] J. Rincon and R. Elder. "A high-resolution pressure-based method for compressible flows," *Comput Fluids*, vol. 26, no. 3, pp. 217–231, 1997.

[14] S. Tian, Z. He, G. Li, H. Wang, Z. Shen, and Q. Liu, "Influences of ambient pressure and nozzle-to-target distance on SC-CO2 jet impingement and perforation," *J. Nat. Gas Sci. and Eng.*, vol. 29, pp. 232–242, 2016.

[15] X. Nogueira, L. Ramírez, S.e Khelladi, J. C. Chassaing, and I. Colominas, "A high-order density-based finite volume method for the computation of all-speed flows," *Comp. Meth. App. Mech. and Eng.*, vol. 298, pp. 229–251, 2016.

[16] 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.

[17] A. W. Date. "Solution of Navier-Stokes equations on non-staggered grid at all speeds," *Numer. Heat Tran., Part B*, vol. 33, no. 4, pp. 451–467, 1998.

[18] N. Ron-Ho, "A multiple-grid scheme for solving the Euler equations," *Aiaa J.*, vol. 20, no. 11, pp. 1565–1571, 1982.

[19] S. Eidelman, P. Colella, and R. P. Shreeve, "Application of Godunov method and its Second-Order extension to Cascade flow modelling," *Aiaa J.*, vol. 22, no. 11, pp.1609–1615, 1984.

[20] M. L. Mason, L. E. Putnam, and R. J. Re, "The Effect of Throat Contouring in Two-dimensional Converging Diverging Nozzles at Static Conditions," NASA TP1704, 1980.