Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 1930–1951
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fractional step method for solving the compressible Navier–Stokes equations

Kunlun Liu *, Richard H. Pletcher

Department of Mechanical Engineering, 2025 Black Engr. Building, Iowa State University, Ames, IA 50011, United States

Received 1 May 2006; received in revised form 19 June 2007; accepted 21 June 2007
Available online 5 July 2007

Abstract

This paper proposes a fractional step method for the calculation of compressible Navier–Stokes equations. The purpose of this study is to develop a robust and efficient numerical method for the simulation of low Mach number flows in which the poorly distributed eigenvalues usually result in the numerical difficulties. The method takes advantage of the pressure-based and the density-based methods. It thus accelerates the numerical convergence by adjusting the eigenvalues of the Jacobian matrix. In order to control the numerical instability and the spurious wave reflections at the numerical boundaries, a characteristic boundary condition is formulated. This numerical method as well as the boundary condition treatment extend the traditional fractional step method to the simulation of compressible Navier–Stokes equations. The performance and the accuracy of the method to calculate the flows with large amplitude of acoustic waves and strong heat transfer have been demonstrated by the direct numerical simulation of three distinct cases: a one-dimensional Euler equation with large amplitude of acoustic waves, an adiabatic turbulent boundary layer, and a turbulent boundary layer with heated wall.

Keywords: Fractional Step Method; Projection method; Compressible Navier–Stokes equations; Preconditioning method

1. Introduction

Traditional fractional step or projection methods for incompressible Navier–Stokes equations were introduced by Chorin [1] and Temam [2] in the late 1960s. The reformulation of these methods by Kim and Moin [3] in 1985 was a milestone in the development of computational fluid dynamics. After that, the fractional step method has enjoyed widespread popularity in the simulation of incompressible flows because it provides an efficient way to remove most of the computational difficulties arising from the low Mach number limit. The main idea of fractional step method is to replace the singular matrix with some proper submatrices by using a factorization technique. These submatrices can be solved easily because they have a better eigenvalue distribution, thereby increasing the efficiency of numerical convergence. The success of the fractional step

* Corresponding author.
E-mail addresses: liukl@umn.edu, kunlunliu@gmail.com (K. Liu).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.06.026

method in incompressible Navier–Stokes equations inspires us to seek an analogous algorithm to improve the simulation of compressible Navier–Stokes equations.

In low Mach number flows, the influence of density fluctuations on the turbulence boundary layers is negligible as long as the mean density remains constant across the boundary layers [4]. Therefore, such flows can be treated as incompressible flows and traditional fractional steps methods for incompressible flows can be applied. However, the density varies with the distance from the wall once the heat transfer is involved. The typical cases are the thermal turbulent boundary layers, which greatly affect the turbulent structures. As a result, the mean velocity profiles will eventually be deformed even though the Mach number is small [5]. Huang et al. [6] and Nicoud and Bradshaw [5] suggested using some density-weighted transformations to modify the velocity profiles such that the similarity law of the incompressible flows can be applied to the turbulent boundary layers with strong heat transfer. Under the circumstances that the heat transfer is not strong and the amplitude of the acoustic waves is not large, the low Mach number flows can be approached by the incompressible Navier–Stokes equations, while solving the overall compressible Navier–Stokes equations is an option which is able to capture both compressibility and variable density effects [7].

Currently, there are three distinct methods for the simulation of the compressible Navier–Stokes equations with a low Mach number: shifting operator methods [8], density-based methods [9–11] and the pressure-based methods [12–14]. The main idea of the shifting operator methods is to adjust the eigenvalues by using the artificial viscous terms or diagonal matrices, for example, the LU-SGS scheme proposed by Yoon and Jameson [8]. The disadvantage of the methods is that the acoustic modes may be damped. In the case the acoustic effects are significant, these methods should be improved.

The density-based methods contain two classes: the preconditioning schemes and the asymptotic schemes. Asymptotic schemes decompose the pressure into thermodynamic pressure and hydrodynamic pressure by using the perturbation method [9]. This treatment will accelerate the convergence if the pressure fluctuations are small. However, this assumption is usually violated by turbulence in which the pressure fluctuations have the same order of magnitude with the velocity fluctuations [15]. The preconditioning schemes offer an easy, efficient, and accurate way to calculate the compressible Navier–Stokes equations with a high Reynolds number under this circumstance. The preconditioning methods optimize the Jacobian matrix by using a left multiplier [10,11]. A well-designed preconditioning matrix will cluster the eigenvalues, which will eventually accelerate the convergence.

The pressure-based methods [14,12,13] are analogous to the fractional step methods [3] for the incompressible flows. Unlike the density-based methods, the pressure-based methods [3,14] adjust the Jacobian matrix by using a factorization matrix. For low Mach number flows in which the compressibility effects are not negligible, the acoustic waves are transported with a speed significantly faster than the convective waves. As a result, the time step of the traditional explicit schemes was limited to a very small value. To avoid the acoustic CFL restriction, Wall et al. [14] proposed a semi-implicit scheme, in which velocity components and pressure were staggered in time by one-half time step with respect to the scalar variables. This method prevented the artificial damping of acoustic waves, in contrast to the LU-SGS scheme. Two differences between the fractional step methods for the compressible Navier–Stokes equations and incompressible Navier–Stokes equations shall be addressed: firstly, the pressure equation is a Poisson equation for the incompressible Navier–Stokes equations but a Helmholtz equation for the compressible Navier–Stokes equations [14]. Secondly, when the compressibility and variable density effects become significant in the system, the energy equation should be solved together with the momentum and continuity equations. The traditional fractional step methods cannot solve the energy equation implicitly or semi-implicitly.

Furthermore, according to the Ostrowski-Reich theorem [16], the Gauss–Seidel scheme or SOR scheme converges if and only if the Jacobian matrix is positive definite. However, this restriction is usually violated by the low Mach number flows or incompressible flows, in which some eigenvalues may be negative or extremely large. The negative eigenvalues will give rise to the divergence of some iterative schemes such as the Gauss–Seidel scheme, Newton Gauss–Seidel scheme, or SOR scheme. In order to overcome this problem, this paper proposes a method to optimize the distribution of eigenvalues of the Jacobian matrix by using a right multiplier and a left multiplier. In other words, the method deduced by this paper is a combination of a preconditioning method and a factorization treatment, or a combination of pressure-based and density-based methods.

The rest of this paper is organized as follows: In Section 2, we present the governing equations. The preconditioning and the factorization methods will be addressed and analyzed in this section. In Section 3, the numerical algorithm will be described in detail. A characteristic-based method for treating inflow and outflow boundary conditions will be discussed in Section 4. And in Section 5, we will present the results of numerical practises.

2. Governing equations and numerical solver

In this paper, variables are normalized as the following:

$$\begin{aligned} x_i &= \frac{x_i^*}{L_{\text{ref}}}, \quad u_i = \frac{u_i^*}{U_{\text{ref}}}, \quad t = \frac{t^*}{(L_{\text{ref}}/U_{\text{ref}})}, \quad \rho = \frac{\rho^*}{\rho_{\text{ref}}} \\ p &= \frac{p^*}{\rho_{\text{ref}} U_{\text{ref}}^2}, \quad T = \frac{T^*}{T_{\text{ref}}}, \quad e = \frac{e^*}{U_{\text{ref}}^2}, \quad \mu = \frac{\mu^*}{\mu_{\text{ref}}} \\ k &= \frac{k^*}{k_{\text{ref}}}, \quad c_v = \frac{c_v^*}{(U_{\text{ref}}^2/T_{\text{ref}})}, \quad R = \frac{R^*}{(U_{\text{ref}}^2/T_{\text{ref}})}, \quad c_p = \frac{c_p^*}{(U_{\text{ref}}^2/T_{\text{ref}})} \end{aligned} \tag{1}$$

The reference Mach number is $M_{\text{ref}} = U_{\text{ref}}/\sqrt{\gamma R^* T_{\text{ref}}}$. As usual, the flows are assumed to be an ideal gas and the non-dimensional equation of state is given by $p = \rho RT$. The non-dimensional coefficients of viscosity and thermal conductivity are evaluated as: $\mu = T^s$, where s is assumed to be 0.71. The specific heats, c_p and c_v, are considered constant. Hence, the governing equation is as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \tag{2}$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{\partial \sigma_{ij}}{\partial x_j} \tag{3}$$

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial[(\rho E + p)u_j]}{\partial x_j} = \frac{\partial(u_i \sigma_{ij})}{\partial x_j} - \frac{\partial q_j}{\partial x_j} \tag{4}$$

Here, $E = c_v T + \frac{1}{2} u_i u_j$, $\sigma_{ij} = \frac{2\mu}{Re_r}\left(S_{ij} - \frac{1}{3} S_{kk}\delta_{ij}\right)$, and $q_j = -\frac{c_p \mu}{Re_{\text{ref}} P_r} \frac{\partial T}{\partial x_j}$.

2.1. Discretization

The discretization of quantities is represented by the operator $[\cdot]_{I,J,K}^{n,m}$, where the superscript n stands for the physical time step and m stands for the pseudo-time step; subscript I, J, and K are corresponding to the streamwise, normal, and spanwise stations, respectively. For instance, $[\Phi]_{I,J,K}^{n,m}$ is the value of function Φ in the location I, J, K and at the moment n, m.

We let the discretization operator $[\cdot]_{I,J,K}^{n,m}$ preserve the addition and multiplication laws. That is

$$[\Phi + \Psi]_{I,J,K}^{n,m} = \Phi_{I,J,K}^{n,m} + \Psi_{I,J,K}^{n,m} = [\Phi]_{I,J,K}^{n,m} + [\Psi]_{I,J,K}^{n,m} \tag{5}$$

and

$$[\Phi \cdot \Psi]_{I,J,K}^{n,m} = \Phi_{I,J,K}^{n,m} \cdot \Psi_{I,J,K}^{n,m} = [\Phi]_{I,J,K}^{n,m} \cdot [\Psi]_{I,J,K}^{n,m} \tag{6}$$

(5) means that the discrete value of the summary of two functions is equal to the summary of their discrete values. Also, (6) means that the discrete value of the multiplication of two functions is equal to the multiplication of their discrete values.

Usually, to capture the turbulent structures at the low Mach number limit, the central difference scheme in space and the fully implicit or semi-implicit scheme in time are recommended. In this study, the second order central difference scheme was applied for the space derivative, and the second order Euler backward scheme was utilized for the time derivative. For the sake of simplicity, the discretization of $\frac{\partial \Phi}{\partial t}$, $\left[\frac{\delta \Phi}{\delta t}\right]_{I,J,K}^{n,m}$, is given by

$$\left[\frac{\delta \Phi}{\delta t}\right]_{I,J,K}^{n,m} = \frac{3\Phi_{I,J,K}^{n,m} - 4\Phi_{I,J,K}^{n-1,m} + \Phi_{I,J,K}^{n-2,m}}{2\Delta t} \tag{7}$$

Likewise, the discretization of $\frac{\partial \Psi}{\partial x}$, $\left[\frac{\delta \Psi}{\delta x}\right]^{n,m}_{I,J,K}$, is given by

$$\left[\frac{\delta \Psi}{\delta x}\right]^{n,m}_{I,J,K} = \frac{\Psi^{n,m}_{I+1,J,K} - \Psi^{n,m}_{I-1,J,K}}{2\Delta x} \tag{8}$$

To conduct the pseudo-time iteration, an operator δ_m is introduced which is defined by

$$[\delta_m \Phi]^{n,m}_{I,J,K} := [\Phi]^{n,m+1}_{i,j,k} - [\Phi]^{n,m}_{i,j,k} \tag{9}$$

where $\delta_m \Phi$ is a variation of Φ in the pseudo-time direction. Owing to (6) and the definition (7)–(9), we have

$$\left[\delta_m \frac{\delta \Phi}{\delta t}\right]^{n,m}_{x,y,z} = \left[\frac{\delta}{\delta t}\delta_m \Phi\right]^{n,m}_{x,y,z} \tag{10}$$

$$\left[\delta_m \frac{\delta \Phi}{\delta x_i}\right]^{n,m}_{x,y,z} = \left[\frac{\delta}{\delta x_i}\delta_m \Phi\right]^{n,m}_{x,y,z} \tag{11}$$

$$\left[\Psi \frac{\delta \Phi}{\delta x}\right]^{n,m}_{I,J,K} = [\Psi]^{n,m}_{I,J,K}\left[\frac{\delta \Phi}{\delta x}\right]^{n,m}_{I,J,K} \tag{12}$$

Since the mathematical rigor is not the focus of this paper, the detailed proofs about equalities (10)–(12) are neglected here.

2.2. Discretization of the governing equations

Thus, the discretization of the system (2)–(4) is given by

$$\left[\frac{\delta \rho}{\delta t}\right]^{n,m+1}_{I,J,K} + \left[\frac{\delta(\rho u_j)}{\delta x_j}\right]^{n,m+1}_{I,J,K} = 0 \tag{13}$$

$$\left[\frac{\delta(\rho u_i)}{\delta t}\right]^{n,m+1}_{I,J,K} + \left[\frac{\delta(\rho u_i u_j)}{\delta x_j}\right]^{n,m+1}_{I,J,K} + \left[\frac{\delta p}{\delta x_i}\right]^{n,m+1}_{I,J,K} = \left[\frac{\delta \sigma_{ij}}{\delta x_j}\right]^{n,m}_{I,J,K} \tag{14}$$

$$\left[\frac{\delta(\rho E)}{\delta t}\right]^{n,m+1}_{I,J,K} + \left[\frac{\delta[(\rho E + p)u_j]}{\delta x_j}\right]^{n,m+1}_{I,J,K} = \left[\frac{\delta(u_i \sigma_{ij})}{\delta x_j}\right]^{n,m}_{I,J,K} - \left[\frac{\delta q_j}{\delta x_j}\right]^{n,m}_{I,J,K} \tag{15}$$

Above governing equations are fully implicit with respect to the physical time step n. Therefore, a pseudo-time iteration should be employed for solving the variables. When the pseudo-time iteration converges, both of the states $[\cdot]^{n,m+1}_{I,J,K}$ and $[\cdot]^{n,m}_{I,J,K}$ will approach the state $[\cdot]^{n,\infty}_{I,J,K}$. In this manner, the distinction between the state $[\cdot]^{n,m+1}_{I,J,K}$ and $[\cdot]^{n,m}_{I,J,K}$ is negligible, provided m is large enough and the pseudo-time iteration converges. The pseudo-time iteration will be stopped at the moment that a threshold is reached. For simplification, we denote $[\cdot]^{n,\infty}_{I,J,K}$ by $[\cdot]^{n}_{I,J,K}$.

By using (9), Eqs. (13)–(15) end up with

$$\left[\delta_m \frac{\partial \rho}{\partial t}\right]^{n,m}_{I,J,K} + \left[\delta_m \frac{\delta(\rho u_j)}{\delta x_j}\right]^{n,m}_{I,J,K} = -[F_1]^{n,m}_{I,J,K} \tag{16}$$

$$\left[\delta_m \frac{\delta(\rho u_i)}{\delta t}\right]^{n,m}_{I,J,K} + \left[\delta_m \frac{\delta(\rho u_i u_j)}{\delta x_j}\right]^{n,m}_{I,J,K} + \left[\delta_m \frac{\delta p}{\delta x_i}\right]^{n,m}_{I,J,K} = -[F_{i+1}]^{n,m}_{I,J,K} \tag{17}$$

$$\left[\delta_m \frac{\delta(\rho E)}{\delta t}\right]^{n,m}_{I,J,K} + \left[\delta_m \frac{\delta[(\rho E + p)u_j]}{\delta x_j}\right]^{n,m}_{I,J,K} = -[F_5]^{n,m}_{I,J,K} \tag{18}$$

Here, $i, j = 1, 2, 3$ and

$$-[F_1]^{n,m}_{I,J,K} = -\left[\frac{\delta \rho}{\delta t}\right]^{n,m}_{I,J,K} - \left[\frac{\delta(\rho u_j)}{\delta x_j}\right]^{n,m}_{I,J,K} \tag{19}$$

$$-[F_{i+1}]^{n,m}_{I,J,K} = -\left[\frac{\delta(\rho u_i)}{\delta t}\right]^{n,m}_{I,J,K} - \left[\frac{\delta(\rho u_i u_j)}{\delta x_j}\right]^{n,m}_{I,J,K} - \left[\frac{\delta p}{\delta x_i}\right]^{n,m}_{I,J,K} + \left[\frac{\delta \sigma_{ij}}{\delta x_j}\right]^{n,m}_{I,J,K} \tag{20}$$

$$-[F_5]^{n,m}_{I,J,K} = -\left[\frac{\delta(\rho E)}{\delta t}\right]^{n,m}_{I,J,K} - \left[\frac{\delta[(\rho E + p)u_j]}{\delta x_j}\right]^{n,m}_{I,J,K} + \left[\frac{\delta(u_i \sigma_{ij})}{\delta x_j}\right]^{n,m}_{I,J,K} - \left[\frac{\delta q_j}{\delta x_j}\right]^{n,m}_{I,J,K} \tag{21}$$

Let

$$Q_1 = \frac{\delta}{\delta t}\delta_m \rho + \frac{\delta}{\delta x_j}\delta_m(\rho u_j) \tag{22a}$$

$$Q_{i+1} = \frac{\delta}{\delta t}\delta_m(\rho u_i) + \frac{\delta}{\delta x_j}\delta_m(\rho u_i u_j) + \frac{\delta}{\delta x_i}\delta_m p \tag{22b}$$

$$Q_5 = \frac{\delta}{\delta t}\delta_m(\rho E) + \frac{\delta}{\delta x_j}\delta_m[(\rho E + p)u_j] \tag{22c}$$

Eqs. (16)–(18) become

$$[Q]^{n,m}_{I,J,K} = -[F]^{n,m}_{I,J,K} \tag{23}$$

2.3. Numerical solver

The best-known and widely applied method for the nonlinear equation is the Quasi-Newton method, which provides a global convergence and enables us to calculate complex Navier–Stokes equations with a rough initial guess. The main idea of the Quasi-Newton method is to achieve a Cauchy sequence of the numerical solutions by solving the corresponding linearized system called the Jacobian equation. In this study, Newton method is employed for the pseudo-time iteration.

Denote $\delta_m v$ by $[\delta_m P, \delta_m u_1, \delta_m u_2, \delta_m u_3, \delta_m T]^{\mathrm{T}}$. The vector Q can be represented as

$$Q = \mathbf{J}\delta_m v$$

where $\mathbf{J}$ is the Jacobian matrix given by

$$\mathbf{J} = \begin{bmatrix} \frac{1}{RT}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} & \frac{P}{RT}\frac{\delta}{\delta x_1} & \frac{P}{RT}\frac{\delta}{\delta x_2} & \frac{P}{RT}\frac{\delta}{\delta x_3} & -\frac{P}{RT^2}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} \\ \frac{u_1}{RT}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} + \frac{\delta}{\delta x_1} & \frac{P}{RT}\left(u_1\frac{\delta}{\delta x_1} + \frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right) & \frac{Pu_1}{RT}\frac{\delta}{\delta x_2} & \frac{Pu_1}{RT}\frac{\delta}{\delta x_3} & -\frac{Pu_1}{RT^2}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} \\ \frac{u_2}{RT}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} + \frac{\delta}{\delta x_2} & \frac{Pu_2}{RT}\frac{\delta}{\delta x_1} & \frac{P}{RT}\left(u_2\frac{\delta}{\delta x_2} + \frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right) & \frac{Pu_2}{RT}\frac{\delta}{\delta x_3} & -\frac{Pu_2}{RT^2}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} \\ \frac{u_3}{RT}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} + \frac{\delta}{\delta x_3} & \frac{Pu_3}{RT}\frac{\delta}{\delta x_1} & \frac{Pu_2}{RT}\frac{\delta}{\delta x_2} & \frac{P}{RT}\left(u_3\frac{\delta}{\delta x_3} + \frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right) & -\frac{Pu_3}{RT^2}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} \\ \frac{H}{RT}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} - \frac{\partial}{\partial t} & \frac{P}{RT}\left(H\frac{\delta}{\delta x_1} + u_1\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right) & \frac{P}{RT}\left(H\frac{\delta}{\delta x_2} + u_2\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right) & \frac{P}{RT}\left(H\frac{\delta}{\delta x_3} + u_3\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right) & -\frac{P(H-CpT)}{RT^2}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} \end{bmatrix}$$

Here, $H = CpT + \frac{1}{2}(u_1^2 + u_2^2 + u_3^2) = E + \frac{P}{\rho}, R = \frac{\gamma}{M^2}$, and $\frac{\overline{\mathrm{D}}}{\mathrm{D}t} = \frac{\delta}{\delta t} + u_i\frac{\delta}{\delta x_i}$ is the discretization of the material derivative.

In this study, all of the terms are implicit in terms of the physical time steps, but the viscous terms are treated explicitly in the pseudo-time iteration. This treatment is based on three considerations: firstly, keeping the viscous terms in the Jacobian matrix will cause the difficulty on the analysis of eigenvalues. Secondly, for the simulation of turbulence, which is usually associated with a large Reynolds number, the influence of viscous terms on the numerical stability of pseudo-time iteration is negligible. At last, once the pseudo-time iteration convergence, the errors produced by the terms which are lagged in the pseudo-time steps are negligible. With the Quasi-Newton method, Jacobian matrix shall be adjusted by a preconditioning and a factorization technique. The main implements are proposed as follows.

2.4. Preconditioning

When a Newton-like method is used, governing equations can be simplified as

$$[\mathbf{J}\delta_m v]^{n,m}_{I,J,K} = -[F]^{n,m}_{I,J,K} \tag{24}$$

Clearly, the Jacobian matrix $\mathbf{J}$ can be decomposed as

$$\mathbf{J} := \Gamma\frac{3}{2\Delta t} + A\frac{\delta}{\delta x_1} + B\frac{\delta}{\delta x_2} + C\frac{\delta}{\delta x_3}$$

In general, matrices A, B, and C are not diagonally dominant. For accelerating the numerical convergence, we suggest adjusting the eigenvalues of matrix **J** by using a left multiplier **P** and right multiplier **M** so that the resolved matrices can maintain diagonal dominance and be positive definite. Thus, an improved solver is:

$$[\mathbf{P}]^{n,m}_{I,J,K}[\mathbf{JMM}^{-1}\delta_m v]^{n,m}_{I,J,K} = -[\mathbf{P}]^{n,m}_{I,J,K}[F]^{n,m}_{I,J,K} \tag{25}$$

By using Eq. (6), it yields

$$[\mathbf{PJMM}^{-1}\delta_m v]^{n,m}_{I,J,K} = -[\mathbf{P}F]^{n,m}_{I,J,K} \tag{26}$$

The left multiplier **P** is essentially a preconditioning matrix, while the right multiplier **M** implements a factorization.

The preconditioning matrix proposed by Pletcher and Chen [10] is utilized in this paper. It is

$$\mathbf{P} = \begin{pmatrix} \frac{q}{C_v} & -\frac{u_1}{C_v} & -\frac{u_2}{C_v} & -\frac{u_3}{C_v} & \frac{1}{C_v} \\ -\frac{RTu_1}{P} & \frac{RT}{P} & 0 & 0 & 0 \\ -\frac{RTu_2}{P} & 0 & \frac{RT}{P} & 0 & 0 \\ -\frac{RTu_3}{P} & 0 & 0 & \frac{RT}{P} & 0 \\ \frac{RT}{P}\left(\frac{q}{C_v} - T\right) & -\frac{RTu_1}{PC_v} & -\frac{RTu_2}{PC_v} & -\frac{RTu_3}{PC_v} & \frac{RT}{PC_v} \end{pmatrix} \tag{27}$$

where

$$q = \frac{(u_1^2 + u_2^2 + u_3^2)}{2}$$

Taking a left multiplication to both sides of Eq. (24) via the above preconditioning matrix, it yields

$$\mathbf{PJ}\delta_m v = \begin{pmatrix} \frac{1}{R}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} & b\frac{\delta}{\delta x_1} & b\frac{\delta}{\delta x_2} & b\frac{\delta}{\delta x_3} & 0 \\ e\frac{\delta}{\delta x_1} & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 & 0 & 0 \\ e\frac{\delta}{\delta x_2} & 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 & 0 \\ e\frac{\delta}{\delta x_3} & 0 & 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 \\ 0 & \frac{TR}{C_v}\frac{\delta}{\delta x_1} & \frac{TR}{C_v}\frac{\delta}{\delta x_2} & \frac{TR}{C_v}\frac{\delta}{\delta x_3} & \frac{\overline{D}}{Dt} \end{pmatrix} \begin{pmatrix} \delta_m P \\ \delta_m u_1 \\ \delta_m u_2 \\ \delta_m u_3 \\ \delta_m T \end{pmatrix} = -\mathbf{P}F \tag{28}$$

where $b = \frac{P}{R} + \frac{P}{C_v}$, $e = \frac{TR}{P}$, and $c = \frac{TR}{C_v}$. Such a preconditioning matrix offers us two advantages. Firstly, it transfers the large off-diagonal terms into the diagonal part so that the numerical scheme becomes more robust no matter how small the Mach number is. Secondly, this preconditioning technique enables us to decouple the temperature variance and velocity variance in the Jacobian matrix.

Without the preconditioning, the eigenvalues of the Jacobian matrix are u_i and $u_i \pm c$ where $c = \sqrt{\gamma RT}$ is acoustic speed. But the eigenvalues with preconditioning become u_i and $\left[u_i(1+R) \pm \sqrt{u_i^2(R-1)^2 + 4Rc^2}\right] \Big/ (2R)$. Obviously, these eigenvalues cluster near 1. But it is still possible for them to be zero or negative. The appearance of negative eigenvalues results in the loss of the positive definiteness. Such non-positive eigenvalues are usually associated with the numerical difficulty for the Quasi-Newton solvers, and may even cause the divergence of the iterative scheme. Therefore, a proper factorization technique will be needed.

2.5. Fractional step factorization

Denote the matrix $\mathbf{M}^{-1}$ in Eq. (26) by

$$\mathbf{M}^{-1} = \begin{bmatrix} \frac{1}{R}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} - b\frac{\delta}{\delta x_i}\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 0 \\ \left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_1} & 1 & 0 & 0 & 0 \\ \left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_2} & 0 & 1 & 0 & 0 \\ \left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_3} & 0 & 0 & 1 & 0 \\ -\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} c\frac{\delta}{\delta x_i}\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 1 \end{bmatrix} \tag{29}$$

Substitute (29) into the identity (26); thus, Eq. (28) is immediately followed by

$$\left[\begin{bmatrix} 1 & b\frac{\delta}{\delta x_1} & b\frac{\delta}{\delta x_2} & b\frac{\delta}{\delta x_3} & 0 \\ 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 & 0 & 0 \\ 0 & 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 & 0 \\ 0 & 0 & 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 \\ 0 & c\frac{\delta}{\delta x_1} & c\frac{\delta}{\delta x_2} & c\frac{\delta}{\delta x_3} & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} \end{bmatrix} \begin{bmatrix} \frac{1}{R}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} - b\frac{\delta}{\delta x_i}\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 0 \\ \left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_1} & 1 & 0 & 0 & 0 \\ \left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_2} & 0 & 1 & 0 & 0 \\ \left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_3} & 0 & 0 & 1 & 0 \\ -\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} c\frac{\delta}{\delta x_i}\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 1 \end{bmatrix} \delta_m v \right]_{I,J,K}^{n,m} = -[\mathbf{P}F]_{I,J,K}^{n,m} \tag{30}$$

where $b = \frac{P}{R} + \frac{P}{C_v}$, $e = \frac{TR}{P}$, and $c = \frac{TR}{C_v}$. Note that the second matrix on the left of (30) is $\mathbf{M}^{-1}$, and the first matrix thus is **PJM**.

This factorization provides a way to decompose variance $\delta_m P$, $\delta_m u_i$, and, $\delta_m T$. By choosing a proper time step Δt, the eigenvalues of matrix **PJM** and $\mathbf{M}^{-1}$ can be guaranteed to be positive and cluster near 1, and hence meet the requirement of the convergence of iterative procedures.

3. Numerical procedure

In the identity (30), $\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1}$ stands for the inverse operator of the material derivative. In [3], Kim and Moin used Δt to replace $\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1}$. Choi and Moin [18] found that the time difference Δt can be up to 0.4 without reducing the accuracy in their simulation of turbulence channel flow. By a conservative scheme, Ham et al. [19] obtained accurate results using a time step as lager as 1.6. Theoretical analysis of this implementation was given by Perot [17] and Lee et al. [20].

In this study, we follow this idea and replace $\left(\frac{\overline{\mathrm{D}}}{\mathrm{D}t}\right)^{-1}$ by Δt. Thus, the discretization of identity (30) becomes

$$\left[\begin{bmatrix} 1 & b\frac{\delta}{\delta x_1} & b\frac{\delta}{\delta x_2} & b\frac{\delta}{\delta x_3} & 0 \\ 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 & 0 & 0 \\ 0 & 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 & 0 \\ 0 & 0 & 0 & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} & 0 \\ 0 & c\frac{\delta}{\delta x_1} & c\frac{\delta}{\delta x_2} & c\frac{\delta}{\delta x_3} & \frac{\overline{\mathrm{D}}}{\mathrm{D}t} \end{bmatrix} \begin{bmatrix} \frac{1}{R}\frac{\overline{\mathrm{D}}}{\mathrm{D}t} - \Delta t b\frac{\delta}{\delta x_i} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 0 \\ \Delta t e\frac{\delta}{\delta x_1} & 1 & 0 & 0 & 0 \\ \Delta t e\frac{\delta}{\delta x_2} & 0 & 1 & 0 & 0 \\ \Delta t e\frac{\delta}{\delta x_3} & 0 & 0 & 1 & 0 \\ -\Delta t^2 c\frac{\delta}{\delta x_i} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 1 \end{bmatrix} \delta_m v \right]_{I,J,K}^{n,m} = -[\mathbf{P}F]_{I,J,K}^{n,m} \tag{31}$$

In summary, the resolved equations suggested by this paper are given by

$$\left[\begin{pmatrix} 1 & b\frac{\delta}{\delta x_1} & b\frac{\delta}{\delta x_2} & b\frac{\delta}{\delta x_3} & 0 \\ 0 & \frac{\overline{D}}{Dt} & 0 & 0 & 0 \\ 0 & 0 & \frac{\overline{D}}{Dt} & 0 & 0 \\ 0 & 0 & 0 & \frac{\overline{D}}{Dt} & 0 \\ 0 & c\frac{\delta}{\delta x_1} & c\frac{\delta}{\delta x_2} & c\frac{\delta}{\delta x_3} & \frac{\overline{D}}{Dt} \end{pmatrix} \delta_m W_i \right]_{I,J,K}^{n,m} = -[\mathbf{P}F]_{I,J,K}^{n,m} \tag{32}$$

and

$$\left[\begin{pmatrix} \frac{1}{R}\frac{\overline{D}}{Dt} - \Delta t b\frac{\delta}{\delta x_i} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 0 \\ \Delta t e\frac{\delta}{\delta x_1} & 1 & 0 & 0 & 0 \\ \Delta t e\frac{\delta}{\delta x2} & 0 & 1 & 0 & 0 \\ \Delta t e\frac{\delta}{\delta x_3} & 0 & 0 & 1 & 0 \\ -(\Delta t)^2 c\frac{\delta}{\delta x_i} e\frac{\delta}{\delta x_i} & 0 & 0 & 0 & 1 \end{pmatrix} \delta_m v \right]_{I,J,K}^{n,m} = [\delta_m W_i]_{I,J,K}^{n,m}$$

Such a decomposition, essentially, is a fractional step procedure. The corresponding numerical procedure follows:

Algorithm 1 (*New Fractional Step Method*)

1: Give an initial guess of $[u_i]_{I,J,K}^{-1}, [p]_{I,J,K}^{-1}, [T]_{I,J,K}^{-1}$. Let $n = 1$, $m = 1$, and

$$[u_i]_{I,J,K}^{n,m} = [u_i]_{I,J,K}^{0} = [u_i]_{I,J,K}^{-1}$$
$$[p]_{I,J,K}^{n,m} = [p]_{I,J,K}^{0} = [p]_{I,J,K}^{-1}$$
$$[T]_{I,J,K}^{n,m} = [T]_{I,J,K}^{0} = [T]_{I,J,K}^{-1}$$

where $i = 1, 2, 3$.

2: Calculate Eqs. (19)–(21) for $[F]_{I,J,K}^{n,m}$, and then compute $[\mathbf{P}F]_{I,J,K}^{n,m}$, where $\mathbf{P}$ is given by the identity (27) and

$$[\mathbf{P}F]_{I,J,K}^{n,m} = [\mathbf{P}]_{I,J,K}^{n,m}[F]_{I,J,K}^{n,m}$$

which follows the formula (6). Decide to go to the next physical time step or not, according to the modulus of $[F]_{I,J,K}^{n,m}$.

3: Determine $\delta_m W_2$, $\delta_m W_3$, and $\delta_m W_4$ by solving the equation

$$\left[\begin{pmatrix} \frac{\overline{D}}{Dt} & 0 & 0 \\ 0 & \frac{\overline{D}}{Dt} & 0 \\ 0 & 0 & \frac{\overline{D}}{Dt} \end{pmatrix} \begin{pmatrix} \delta_m W_2 \\ \delta_m W_3 \\ \delta_m W_4 \end{pmatrix} \right]_{I,J,K}^{n,m} = \left[\begin{pmatrix} -\mathbf{P}F_2 \\ -\mathbf{P}F_3 \\ -\mathbf{P}F_4 \end{pmatrix}\right]_{I,J,K}^{n,m} \tag{33}$$

The discretization of the material derivative, $\frac{\overline{D}}{Dt} = \frac{\delta}{\delta t} + u_i\frac{\delta}{\delta x_i}$, is implemented in this study.

4: Calculate $\delta_m W_1$ and $\delta_m W_5$ by

$$[\delta_m W_1]_{I,J,K}^{n,m} = \left[-\mathbf{P}F_1 - b\frac{\delta}{\delta x_i}\delta_m W_i\right]_{I,J,K}^{n,m}, \tag{34}$$

$$\left[\frac{\overline{D}}{Dt}\delta_m W_5\right]_{I,J,K}^{n,m} = \left[\Delta t\left(-\mathbf{P}F_5 - c\frac{\delta}{\delta x_i}\delta_m W_i\right)\right]_{I,J,K}^{n,m} \quad (i = 1, 2, 3) \tag{35}$$

5: Determine $\delta_m P$ by solving the equation

$$\left[\left(\frac{1}{R}\frac{\overline{D}}{Dt} - b\Delta t\frac{\delta}{\delta x_i}e\frac{\delta}{\delta x_i}\right)\delta_m P\right]_{I,J,K}^{n,m} = [\delta_m W_1]_{I,J,K}^{n,m} \tag{36}$$

6: Calculate the velocity variance $\delta_m u_i$ by

$$[\delta_m u_i]^{n,m}_{I,J,K} = \left[\delta_m W_{i+1} - \Delta t e \frac{\delta}{\delta x_i}\delta_m P\right]^{n,m}_{I,J,K} \quad (i = 1,2,3) \tag{37}$$

7: Calculate the temperature variance $\delta_m T$ by

$$[\delta_m T]^{n,m}_{I,J,K} = \left[\delta_m W_5 + \Delta t^2 c \frac{\delta}{\delta x_i} e \frac{\delta}{\delta x_i}\delta_m P\right]^{n,m}_{I,J,K} \quad (i = 1,2,3) \tag{38}$$

8: Renew u_i, p, and T with $u_i + \delta u_i$, $p + \delta P$, and $T + \delta T$, respectively. Let $m = m + 1$. And then go back to step 2.

As an eight-step procedure, the resolution of steps 3, 4 and 5 requires proper iterative procedures, but the rest can be solved by only one step. For steps 3 and 4, since $\frac{\overline{D}}{Dt}$ is the discretization of the material derivative, a small enough time step Δt can ensure the diagonal dominance, which essentially guarantees the convergence of the iteration. In this study, LU decomposition on the whole matrix was utilized for solving (33) and (35). Eq. (37) is the Helmholtz equation as in the Wall et al.'s method [14].

Considering that the determinant of preconditioning matrix $\mathbf{P}$ is $\frac{R^4T^5}{P^4C_v}$, which is bounded above and below for most of flows, the convergence of matrix $\mathbf{P}\vec{F}$, thus, is equivalent to the convergence of $\vec{F}$. In this sense, either $\vec{F}$ or $\mathbf{P}\vec{F}$ can be used to determine the stop of the iterative procedure in step 2.

4. Characteristic boundary conditions

The preceding section describes a new fractional step method for solving the compressible Navier–Stokes equations. An appropriate treatment for the boundary condition associated with this numerical method needs to be constructed. In the simulation of compressible flows, characteristic-based boundary conditions [21] have been widely utilized. The method is remarkably effective in the controlling of numerical instabilities and in removing spurious wave reflections at the computational boundary. But the traditional characteristic boundary treatment needs to be altered as long as preconditioning is applied because the preconditioning matrix changes the eigenvalues and eigenvectors.

Let us consider a boundary located at $x_1 = L$. Multiplying the governing Eqs. (2)–(4) by the preconditioning matrix (27) yields

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x_1} + B\frac{\partial W}{\partial x_2} + C\frac{\partial W}{\partial x_3} - Vis. = 0$$

where *Vis.* is the viscous term, $W = [P, u_1, u_2, u_3, T]^{\mathrm{T}}$,

$$A = \begin{pmatrix} \frac{u_1}{R} & \frac{P}{R} + \frac{P}{C_v} & 0 & 0 & 0 \\ \frac{TR}{P} & u_1 & 0 & 0 & 0 \\ 0 & 0 & u_1 & 0 & 0 \\ 0 & 0 & 0 & u_1 & 0 \\ 0 & \frac{TR}{C_v} & 0 & 0 & u_1 \end{pmatrix}$$

and

$$B = \begin{pmatrix} \frac{u_2}{R} & 0 & \frac{P}{R} + \frac{P}{C_v} & 0 & 0 \\ 0 & u_2 & 0 & 0 & 0 \\ \frac{TR}{P} & 0 & u_2 & 0 & 0 \\ 0 & 0 & 0 & u_2 & 0 \\ 0 & 0 & \frac{TR}{C_v} & 0 & u_2 \end{pmatrix}, \quad C = \begin{pmatrix} \frac{u_3}{R} & 0 & 0 & \frac{P}{R} + \frac{P}{C_v} & 0 \\ 0 & u_3 & 0 & 0 & 0 \\ 0 & 0 & u_3 & 0 & 0 \\ \frac{TR}{P} & 0 & 0 & u_3 & 0 \\ 0 & 0 & 0 & \frac{TR}{C_v} & u_3 \end{pmatrix}$$

Referring to the appendix, the eigenvalues of A are the following:

$$\lambda_1 = \left[u_1(1+R) - \sqrt{u_1^2(R-1)^2 + 4R^2T\left(1+\frac{R}{C_v}\right)}\right] \Big/ (2R)$$

$$\lambda_2 = \left[u_1(1+R) + \sqrt{u_1^2(R-1)^2 + 4R^2T\left(1+\frac{R}{C_v}\right)}\right] \Big/ (2R)$$

$\lambda_3 = u_1$, $\lambda_4 = u_1$, and $\lambda_5 = u_1$. Define a 5×5 diagonal matrix Λ by

$$\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5]$$

A can be split as

$$A = Q\Lambda R$$

where

$$Q = \begin{pmatrix} q_{11} & q_{12} & 0 & 0 & 0 \\ q_{21} & q_{22} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad R := Q^{-1} = \begin{pmatrix} p_{11} & p_{12} & 0 & 0 & 0 \\ p_{21} & p_{22} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ p_{51} & p_{52} & 0 & 0 & 1 \end{pmatrix}$$

The definition of the entries are given in the appendix.

If we denote $\Lambda R\frac{\partial W}{\partial x_1}$ by L_i, it yields

$$L_1 = \lambda_1\left(p_{11}\frac{\partial P}{\partial x_1} + p_{12}\frac{\partial u_1}{\partial x_1}\right) \tag{39}$$

$$L_2 = \lambda_2\left(p_{21}\frac{\partial P}{\partial x_1} + p_{22}\frac{\partial u_1}{\partial x_1}\right) \tag{40}$$

$$L_3 = \lambda_3\frac{\partial u_2}{\partial x_1} \tag{41}$$

$$L_4 = \lambda_4\frac{\partial u_3}{\partial x_1} \tag{42}$$

$$L_5 = \lambda_5\left(p_{51}\frac{\partial P}{\partial x_1} + p_{52}\frac{\partial u_1}{\partial x_1} + \frac{\partial T}{\partial x_1}\right) \tag{43}$$

where L_i are the amplitude of characteristic waves associated with each characteristic velocity λ_i. Unlike the traditional characteristic system, the propagation waves corresponding to L_1 and L_2 will no longer be the acoustic waves. By using these L_i,

$$A\frac{\partial W}{\partial x_1} = \begin{bmatrix} q_{11}L_1 + q_{12}L_2 \\ q_{21}L_1 + q_{22}L_2 \\ L_3 \\ L_4 \\ L_3 + L_4 + L_5 \end{bmatrix}$$

It is known that the upstream-propagating waves are associated with the negative eigenvalues, and the downstream-propagating waves are associated with the positive eigenvalues. In order to control the spurious wave reflections and achieve numerical stability, the forward scheme is recommended for the downstream-propagating waves and the backward scheme is recommended for the upstream-propagating waves. Without loss of generality, assume $\lambda_1 \leqslant 0$ and $\lambda_2 \geqslant 0$. Thus,

$$L_1 = \lambda_1\left(p_{11}\frac{P_{I+1,j,k}-P_{I,j,k}}{\Delta x_1}+p_{12}\frac{u_{1_{I+1,j,k}}-u_{1_{I,j,k}}}{\Delta x_1}\right)$$

and

$$L_2 = \lambda_2\left(p_{21}\frac{P_{I,j,k}-P_{I-1,j,k}}{\Delta x_1}+p_{22}\frac{u_{1_{I,j,k}}-u_{1_{I-1,j,k}}}{\Delta x_1}\right)$$

Since the numerical boundary condition must be given in order to calculate L_i, Poinsot and Lele [21] suggested a series of methods to evaluate these L_i. But these need to be modified since the preconditioning is applied.

4.1. Outlet characteristic boundary conditions for subsonic flows

We discuss the outlet boundary condition for the subsonic flows here. Clearly, λ_2, λ_3, λ_4, and, λ_5 are positive. Hence, L_2, L_3, L_4, and, L_5 are calculated by the backward difference. But L_1 must be specified. Poinsot and Lele [21] suggested applying

$$L_1 = \sigma(1-M^2)(P-P_\infty)c/L$$

where M is the maximum Mach number in the flow, c is the acoustic speed, L is the characteristic size of the domain, and σ is a constant. Clearly, the transfer of numerical information in the system with preconditioning is not carried by the acoustic wave, but by an upstream-propagating wave with an eigenvalue λ_1 satisfying $|\lambda_1| < |u_1 - c|$. Hence, directly using the L_1 proposed by Poinsot and Lele [21] in this system will overestimate the speed of upstream-propagating waves. Certainly, it is not an optimum choice for the system with preconditioning. For the subsonic flow, $R = \frac{\gamma}{M_\infty^2} > 1$,

$$\lambda_1 = \left[u_1(1+R) - \sqrt{u_1^2(R-1)^2 + 4R^2T\left(1+\frac{R}{C_v}\right)}\right]\Big/(2R) = \frac{\frac{u_1^2}{R} - T\left(1+\frac{R}{C_v}\right)}{u_1\left(\frac{1+R}{2R}\right) + \sqrt{u_1^2\left(\frac{R-1}{2R}\right)^2 + T\left(1+\frac{R}{C_v}\right)}}$$

$$= -\frac{T\left(1+\frac{R}{C_v}\right)}{u_1\left(\frac{1+R}{2R}\right)}\left[\frac{1-\frac{u_1^2 C_v}{TR(C_v+R)}}{1+\sqrt{1+T\left(1+\frac{R}{C_v}\right)\Big/\left[u_1^2\left(\frac{R-1}{2R}\right)^2\right]}}\right] = -2\alpha\frac{T(C_v+R)R}{u_1(1+R)C_v}$$

where $\alpha \in (0,1)$. We therefore suggest that

$$L_1 = \frac{\sigma(1-M^2)(P-P_\infty)}{L}\frac{T^\infty(C_v+R)R}{u_1^\infty(1+R)C_v}$$

where u_1^∞ is the free stream streamwise velocity and T^∞ is the free stream temperature. σ is suggested to be 0.25.

5. Numerical practices

For the low Mach number flows in which the compressibility or variable density effects are not negligible, the compressible Navier–Stokes equations rather than the incompressible Navier–Stokes equations should be solved. The variable density effects are often associated with the strong heat transfer. For the low Mach number thermal turbulent boundary layers, the variable density effects instead of the compressibility effects are significant. In such flows, the pressure change is small but the heat transfer is strong. The coupling between the density and temperature need to be well treated. Usually, the low Mach number flows are incompressible. However, when the acoustic waves have a large amplitude, the low Mach number flows may present a strong compressibility. Under this circumstance, the coupling between the pressure and density will cause the numerical difficulties. (For details, refer to [14].)

Thus, to validate the method, a one-dimensional Euler equation with the large amplitude acoustic waves at a Mach number of 0.02, an adiabatic turbulent boundary layer at a Mach number of 0.06, and a turbulent boundary layer with heated wall at the Mach number of 0.06 are calculated by using the direct numerical sim-

ulations. For both turbulent boundary layers, the free stream pressure gradient is zero. Owing to the large amplitude acoustic waves, the compressibility effects are dominant in the first case.

5.1. A one-dimensional Euler equation

As we know, the magnitude of the sound speed for the low Mach number compressible flows is much larger than the convective velocity u. It is hard to converge the code under this circumstance. For testing the properties of the method at the low Mach number limit, a one-dimensional Euler equation,

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} = 0 \tag{44}$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} = -\frac{\partial p}{\partial x} \tag{45}$$

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial[(\rho E + p)u]}{\partial x} = 0 \tag{46}$$

was calculated by the present method. The calculated Mach number M_0 was chosen as 0.02. Clearly, Eqs. (44)–(46) form a hyperbolic system. The unsteady structures are transported to the downstream or upstream via the convective or acoustic waves. The characteristic line method enable us to anticipate the propagations of these convective and acoustic waves. The rearrangement of terms in Eqs. (44)–(46), based on the idea of the characteristic lines, results in the following equations:

$$\frac{\partial p}{\partial t} + (u-c)\frac{\partial p}{\partial x} = \rho c\frac{\partial u}{\partial t} + \rho c(u-c)\frac{\partial u}{\partial x} \tag{47}$$

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} = \frac{1}{c^2}\frac{\partial p}{\partial t} + \frac{u}{c^2}\frac{\partial p}{\partial x} \tag{48}$$

$$\frac{\partial p}{\partial t} + (u+c)\frac{\partial p}{\partial x} = -\rho c\frac{\partial u}{\partial t} - \rho c(u+c)\frac{\partial u}{\partial x} \tag{49}$$

where c is the sound speed given by $c = \sqrt{\gamma RT}$. (For details, refer to [21]).

The quantities were normalized by the reference system proposed by Section 2. The dimensionless initial velocity and density were 1.0. The dimensionless initial pressure was $P(x) = P_0$, where $P_0 = \frac{1}{\gamma M_0^2}$. At $x = 0$, the velocity was specified to vary sinusoidally in time about a mean value, more precisely, $u(0) = 1.0 + 0.2\sin(t)$, while density was specified to be constant. At $x = 800$, the pressure was specified to be a sinusoidal function, where $P(800) = P_0 + 0.08P_0\sin(t)$. The numerical domain was from 0 to 800. The ratio of the amplitude of the pressure oscillation at $x = 800$ to the amplitude of the velocity oscillation at $x = 0$ was $\frac{0.08}{\gamma M_0^2}/0.2 = 1607.14$. Thus, the acoustic waves were much faster than the convective waves. Such a large amplitude acoustic wave usually results in numerical instability. It slows the numerical convergence and even causes the divergence of the code.

The above case was calculated by four distinct methods: a semi-implicit pressure-based method proposed by Wall et al. [14], a density-based method [24], a characteristic line method, and the present method. Note that the variables are assumed to be constant along the characteristic direction by the characteristic line method. Thus, an explicit scheme of Eqs. (47)–(49) could be utilized to obtain the solution by means of characteristic line method. In summary, a second order central difference for the space derivative was implemented for all of these four methods. The first order forward difference in time was utilized for the characteristic line method, while the second order Euler backward difference in time was implemented for the others. $\Delta x = 0.1$ and $\Delta t = \frac{0.2}{1+1/M_0}\Delta x$ for the characteristic line method, but $\Delta x = 2$ and $\Delta t = 0.1$ for the rest. CFL number was $0.2u$ for the characteristic line method, but $2.55u$ for the rest cases. Since the rest method were implicit or semi-implicit scheme, the large CFL number can be applied.

Fig. 1 shows the convergent history of the maximum residual errors by the present method and the semi-implicit pressure-based method [14]. Both of the methods solved the pressure Helmholtz equation. The eigenvalues of the Jacobian matrix was adjusted by using the preconditioning matrix for the present scheme. The results reveal that a better distribution of the eigenvalues allows a high performance of the convergence.

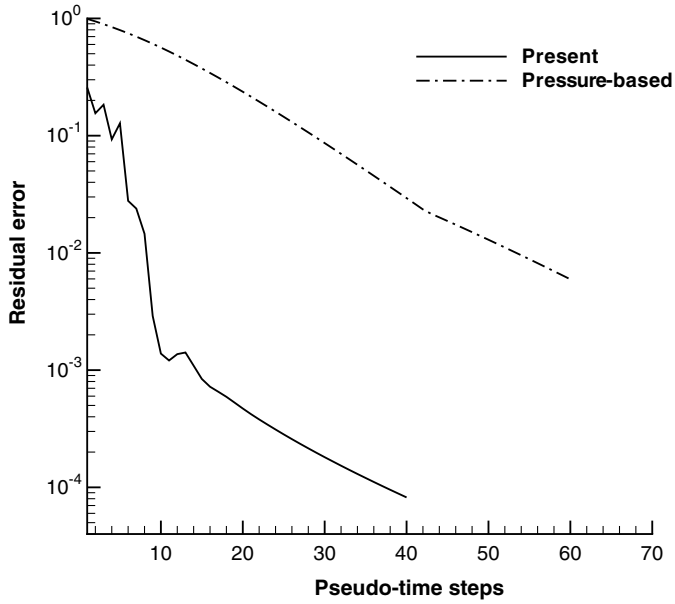

Fig. 1. Plots of the maximum residual error vs. the pseudo-time step. The solid line was calculated by the present method. And the dash-dotted line was calculated by a pressure-based method [14].

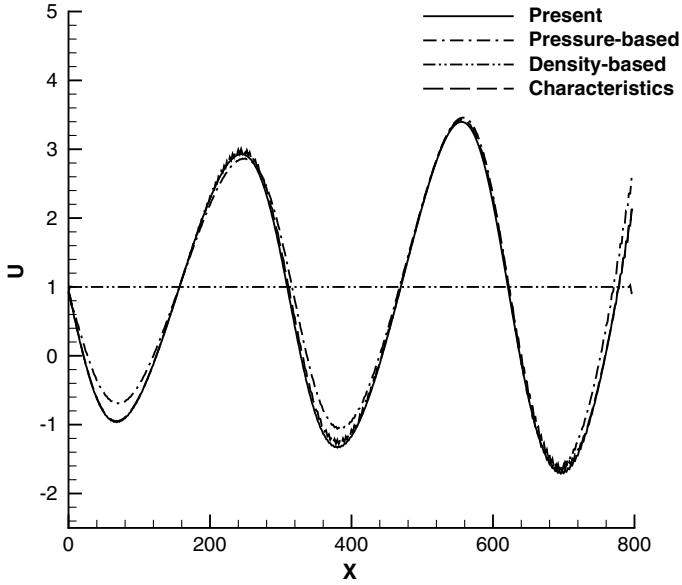

Fig. 2. Plots of the velocity vs. the x-axis, where the density-based scheme was reported by [24], the pressure-based scheme was reported by [14], and characteristics corresponds to the characteristic line method.

Figs. 2 and 3 plot the velocity and pressure distribution calculated by four different methods at the moment $T = 1000$, respectively. The amplitude of velocity in the downstream was significantly enlarged by the acoustic waves, although it was only 0.2 in the inlet. This phenomena was captured by the semi-implicit pressure-based method [14], characteristic line method, and the present method. However, by using the LU-SGS method [24], the large amplitude acoustic waves were damped by the artificial dissipative terms, even though the preconditioning technique was implemented. Since the amplitude of the pressure oscillation at the outlet was too large. In order to converge the code, we had to add a large artificial dissipative term in the case of the LU-SGS method. Owing to this, our calculation by using

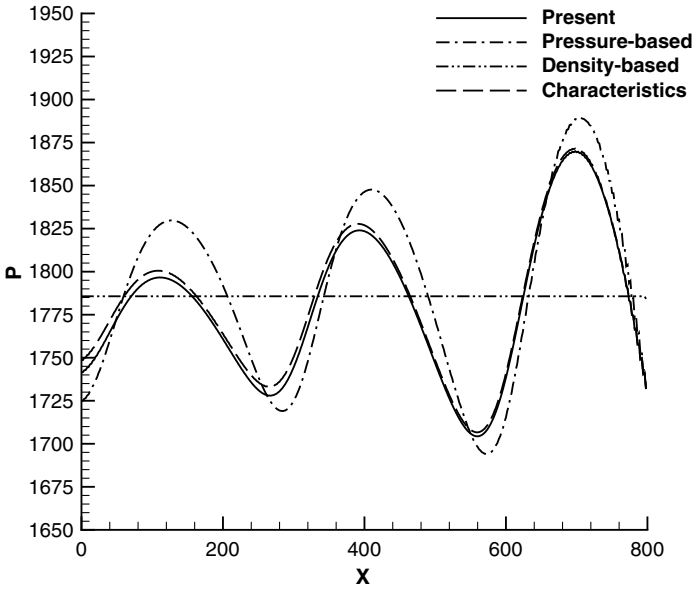

Fig. 3. Plots of the pressure vs. the x-axis, where the density-based scheme was reported by [24], the pressure-based scheme was reported by [14], and characteristics corresponds to the characteristic line method.

LU-SGS method did not capture the coupling between the pressure and velocity. The results reveal that the artificial damping may not be a good method for the simulation of the low Mach number flows with large amplitude acoustic waves.

In summary, the present method which solves the pressure Helmholtz equation can capture the large amplitude acoustic waves for the low Mach number flows. Moreover, it converges faster than the pressure-based method because it takes the advantage of the density-based method and has a better distribution of eigenvalues than the pressure-based method [14].

5.2. Turbulent boundary layers

In order to demonstrate this numerical method, the turbulent boundary layers with or without heat transfer were calculated by the proposed method. The calculated Reynolds number was from approximately 1800–2200 in terms of displacement thickness. The Prandtl number was 0.71. Direct numerical simulation (DNS) was implemented.

A second order central difference finite volume scheme was applied. The second order Euler backward scheme was utilized on the time difference. A fully implicit scheme was implemented. The numerical procedure included two loops: an inner loop and an outer loop.

A recycling and rescaling inlet condition proposed by Liu and Pletcher [22] was utilized for the velocities and temperature. The inlet pressure was extrapolated from the interior. On the upper boundary, the pressure and temperature were specified as free stream. The numerical mesh was $350 \times 90 \times 192$ in the streamwise, normal, and spanwise directions, respectively. The mesh was uniform in the streamwise and spanwise directions and stretched in the normal direction, where $\Delta x^+ = 20.0, \Delta z^+ = 6.7, \Delta y^+_{\min} = 0.3, \Delta y^+_{\max} = 65$.

A random function was used to generate the initial fluctuations. And the Blasius profile was used as the initial mean profile. The characteristic boundary conditions proposed in this paper were applied at the outlet.

5.2.1. Case2: Adiabatic turbulent boundary layer

The first case was a flow at a Mach number of 0.06, a level at which a traditional compressible scheme would experience very poor convergence or even divergence.

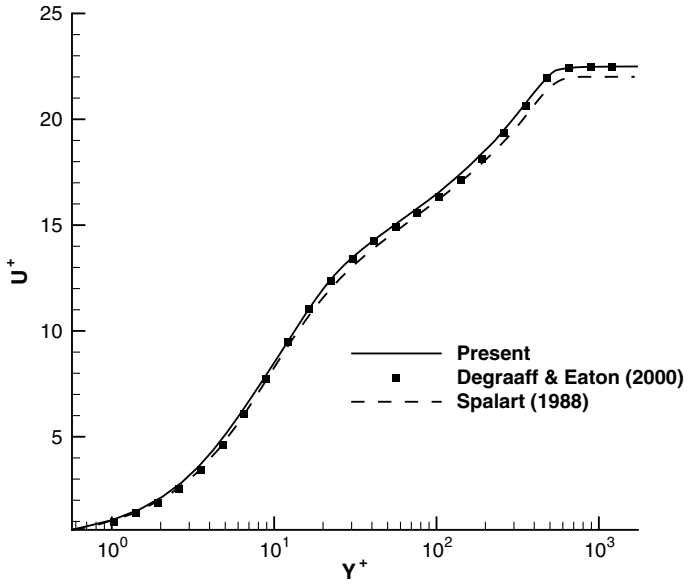

Fig. 4. Comparison of mean velocity profile in a turbulent boundary layer $Re_{\delta_d} = 2000$ over adiabatic wall, where δ_d is the displacement thickness. The dashed line gives a DNS profile by Spalart [15], and the square symbols are experimental data by DeGraaff and Eaton [23].

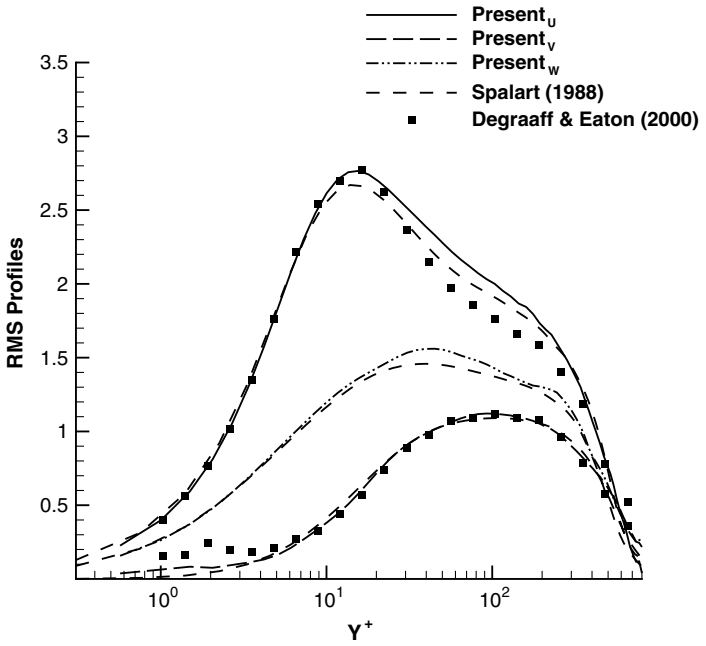

Fig. 5. Comparison of rms of velocity profiles in a turbulent boundary layer $Re_{\delta_d} = 2000$ over adiabatic wall : The dashed line gives a DNS profile by Spalart [15], and the square symbols are experimental data by DeGraaff and Eaton [23].

The first and second order statistics at $Re_d = 2000$ were shown in Figs. 4 and 5, respectively. The compared DNS results were calculated by Spalart [15] and experiment data were reported by DeGraaff and Eaton [23]. We found a fairly close agreement between our DNS results and previous data.

Fig. 6 plots the turbulent kinetic energy budget. The turbulent kinetic energy budget reveals the detailed information and mechanism of the turbulence. The governing equation for the turbulent kinetic energy budget is given by

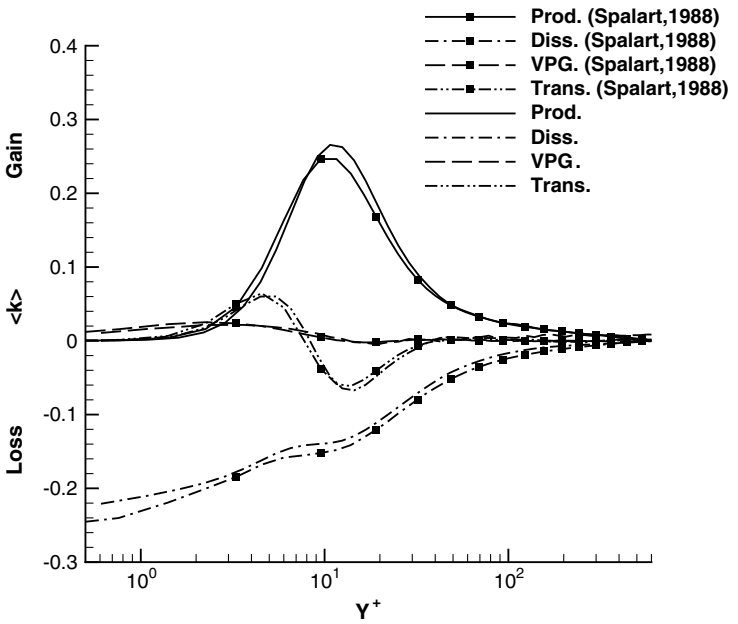

Fig. 6. Comparison of the turbulent kinetic energy budget in a turbulent boundary layer $Re_{\delta_d} = 2000$ over adiabatic wall, where Prod., Diss., VPG, and Trans. stand for the turbulent production, turbulent dissipation, Velocity pressure gradient, and turbulent transport, respectively.

$$\frac{\partial k}{\partial t} + U_k \frac{\partial k}{x_k} = P_{ii} + T_{ii} + D_{ii} + \Pi_{ii} - \varepsilon_{ii} \tag{50}$$

where $k = \frac{\langle u_i u_i \rangle}{2}$, P_{ii} is the turbulent production tensor, T_{ii} is the turbulent transport tensor, D_{ii} is the viscous dissipation tensor, Π_{ii} is the velocity pressure gradient tensor, and ε_{ii} is the turbulent dissipation tensor. The representation of those terms are given by,

$$P_{ii} = -2\langle u_i u_k \rangle \frac{\partial U_i}{\partial x_k}, \quad \varepsilon_{ii} = 2\nu \left\langle \frac{\partial u_i}{\partial x_k} \frac{\partial u_i}{\partial x_k} \right\rangle, \quad T_{ii} = -\left\langle \frac{\partial u_i u_i u_k}{\partial x_k} \right\rangle, \quad \Pi_{ii} = -\frac{2}{\rho} \left\langle u_i \frac{\partial p}{\partial x_i} \right\rangle, \quad D_{ii} = \nu \Delta \langle u_i u_i \rangle$$

The numerical results was compared with Spalart's data [15]. The agreement is fairly good.

5.2.2. Case3: Turbulent boundary Layer with heat transfer

In the third case, turbulent boundary layer developing over a heated wall was simulated. The wall temperature T_{wall} was fixed at $1.4T_e$, where T_e is the environment temperature. The calculated Mach number was 0.06. Wall properties were used to normalize the quantities. A density-weighted transformation proposed by Huang, Bradshaw, and Coakley [6] was utilized to compute a modified velocity, which was

$$U_c = \sqrt{B}\left[\sin^{-1}\left(\frac{A+u}{D}\right) - \sin^{-1}\left(\frac{A}{D}\right)\right] \tag{51}$$

where

$$A = q_w/\tau_w, \quad B = \frac{2C_{p_\infty} T_w}{Pr_t}, \quad D = \sqrt{A^2 + B}$$

subscript w stands for the wall property, and subscript ∞ stands for the free stream property. Fig. 7 shows the comparison of modified velocity U_c and u^+ with experimental results of incompressible turbulent boundary layer. This figure verifies that in the boundary layer with heated wall the velocity profiles are deformed due to the change of the density across the boundary layer. Such a deformation causes the u^+ to depart from the law of the wall. But the modified velocity U_c maintains the law of the wall for the incompressible flow. These numerical results match the theoretical and experimental conclusions of [6].

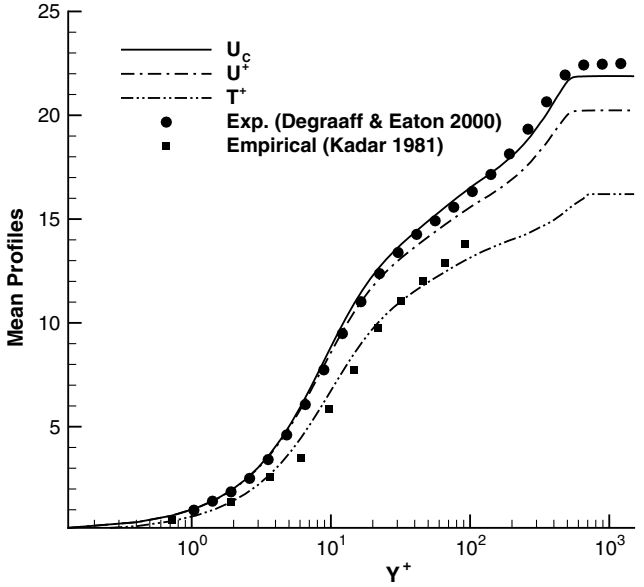

Fig. 7. Mean velocity profile in the turbulent boundary layer with heated wall, where $T_{\text{wall}} = 1.4T_e$. The dash-dotted curve is normalized by U_τ. The solid curve is U^c given by Eq. (51). The circle symbols are the experimental data of the incompressible turbulent boundary layer by DeGraaff and Eaton [23]. The square symbols are the empirical temperature profile suggested by Kadar [25].

The calculated mean temperature profile for the turbulent boundary layer with $T_w/T_\infty = 1.4$ is presented by Fig. 7 also. The square symbols are the empirical mean temperature profile suggested by Kadar [25], which followed:

$$\theta^+ = Pry^+ \exp(-\Gamma) + \left\{ 2.12 \ln \left[(1+y^+) \frac{2.5(2-y/\delta)}{1+4(1-y/\delta)^2} \right] + \beta(Pr) \right\} \exp(-\Gamma)$$

where $\theta^+ = \frac{T_w - T}{t_f}, t_f = \frac{q_w}{\rho_w C_p u_\tau}$, and,

$$\Gamma = \frac{0.01(Pry^+)^4}{1+5Pr^3 y^+}, \quad \beta(Pr) = (3.85(Pr)^{-1/3} - 1.3)^2 + 2.12 \ln Pr$$

The present result agrees with the Kadar's formula [25] in the viscous sublayer and logarithmic layer.

Since the density and temperature varied with the distance from the wall, in order to describe the influence of the temperature on the statistics of the turbulent fluctuations, we normalized the rms of the streamwise velocity fluctuations in three different ways. Fig. 8 shows the comparison of the rms of the streamwise velocity fluctuations. The results shown by the dash-dotted line were normalized by the wall properties, where $U^+_{\text{rms}} = \frac{U_{\text{rms}}}{U_{\tau,w}}$ and $Y^+ = Y\frac{U_{\tau,w}}{\nu_w}$. It matches with DeGraaff and Eaten's incompressible data [23] in the outer part of turbulent boundary layer. The dash line results were normalized by the global properties, where $U^+_{\text{rms}} = \frac{U_{\text{rms}}}{U_{\tau,\infty}}$ and $Y^+ = Y\frac{U_{\tau,\infty}}{\nu_\infty}$. This curve is close to DeGraaff and Eaten's experimental results [23] on the viscous sublayer. The solid line results were normalized by the local properties. More precisely, $U^+_{\text{rms}} = \frac{U_{\text{rms}}}{U_{\tau,\text{local}}}$ and $Y^+ = Y\frac{U_{\tau,\text{local}}}{\nu_{\text{local}}}$. These results give the best match with the peak value of the experimental data. By using the local properties, the U^+_{rms} of thermal turbulent boundary layers collapses the U^+_{rms} of incompressible turbulent boundary layers in the log layer.

The same turbulent boundary layer flow ($T_w/T_\infty = 1.4$) was also calculated by the density-based method [24], the semi-implicit pressure-based method [14], and the artificial damping method [8] which is an LU-SGS scheme. Fig. 9 shows the comparison of the average residual errors versus the iteration number of pseudo-time step loops (or inner loop) by these four different schemes. The magnitude of the average residual

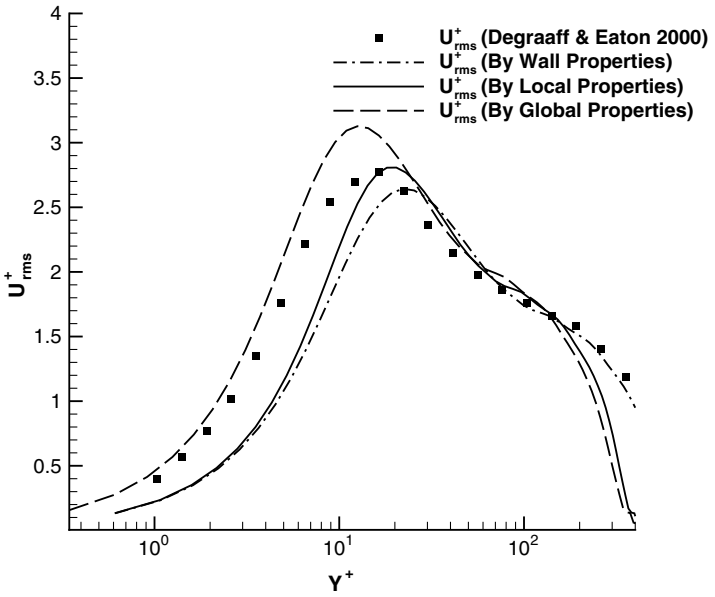

Fig. 8. RMS fluctuation profiles of an turbulent boundary layer with heated wall $T_{wall} = 1.4T_e$. The square symbols are the experimental data of the incompressible turbulent boundary layer by DeGraaff and Eaton [23].

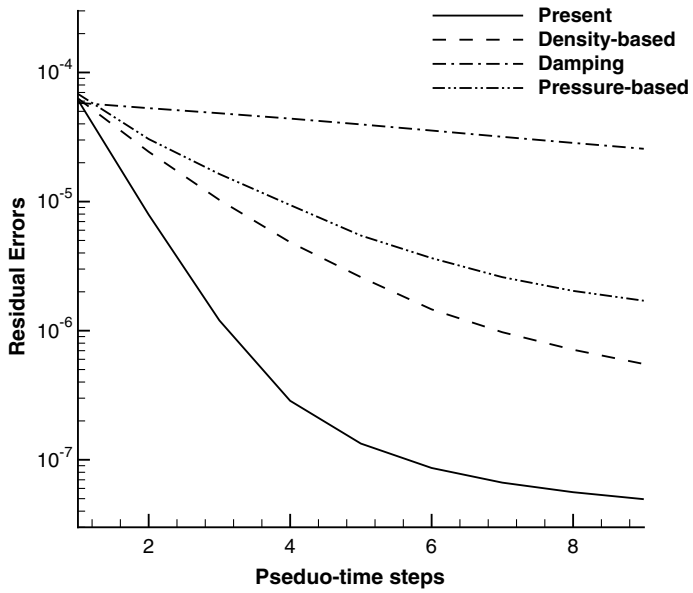

Fig. 9. Convergence history of residual errors vs iteration steps in the pseudo-time loops, where the density-based scheme was reported by [24], the pressure-based scheme was reported by [14], and the damping method was reported by [8].

errors drops down 3 orders in just 9 steps by the present fractional step scheme. But during these 9 steps, the decay in the order of the magnitude using the density-based method or pressure-based method is no more than 2. The convergence history shows the superior performance of the present method for the simulation of the subsonic flows.

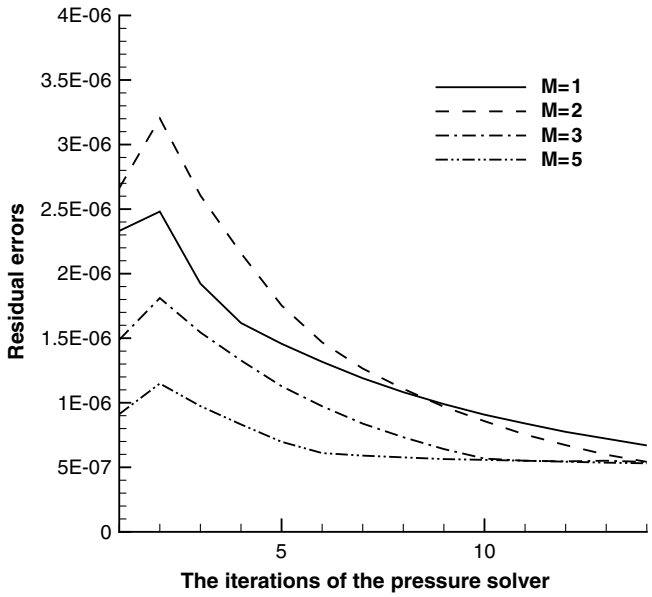

Fig. 10. Convergence history of residual errors vs iteration steps in the pseudo-time loops.

5.3. Pressure solver

Since both the present method and the pressure-based method for the compressible Navier–Stokes equations need to solve a pressure Helmholtz equation, which usually takes much CPU time, the performance of the pressure solver needed to be investigated. (Note that, in each pseudo-time step, an iteration is required to solve the pressure Helmholtz equation.) In this study, the inexact Newton method [26] and the Red Black Gauss Seidel method [27,28] were implemented to improve the convergence of the pressure solver.

The Red Black Gauss Seidel method is one type of classic method to solve the elliptic equations. It helps to smooth the numerical oscillation caused by the even–odd coupling and accelerate the convergence of the elliptic equation. (For details of the method, refer to [27,28].)

There are two Newton methods to solve the pressure Helmholtz equation: get accurate pressure results at each pseudo-time step, which is the exacted Newton method, or get accurate pressure results only at each physical time step, which is the inexact Newton method. The former takes more CPU time. However, it may not reduce the amount of the pseudo-time steps since velocities and temperature will be updated at each pseudo-time step and the convergence of the velocities and temperature depends on the whole system including the momentum equations and the pressure Helmholtz equation. The best way is to converge the pressure, velocities, and temperature simultaneously. Based on these considerations, we use the inexact Newton method [26].

The main idea of the inexact Newton method [26] is to stop the Newton iteration if the threshold, ε, is reached or the residual error drops to $\eta\varepsilon_m$, where ε_m is the initial residual error of the pressure Helmholtz equation in the mth pseudo-time step and η is a small positive constant less than 1. Usually, η is from 0.01 to 0.5 and $\eta\varepsilon_m$ may be larger than the threshold ε, which was 5×10^{-7} in this study. Therefore, the inexact Newton method saves CPU time for solving the pressure Helmholtz equation.

The total CPU time t_{tot} for each physics time step obeys $t_{\text{tot}} = M \times (t_{\text{pre}} + t_{\text{rest}})$ where M is the amount of the pseudo-time steps, t_{pre} is the CPU time for the pressure solver, and t_{rest} is the CPU time for the rest of the calculation. In our study, we found that the inexact Newton method can significantly reduce the CPU time. By using the inexact Newton method, the pressure solver takes roughly 60% of the total CPU time. Otherwise, the pressure solver will take roughly 85% of the total CPU time. Fig. 10 plots the convergent history of the pressure Helmholtz equation, where M is the ordering of the pseudo-time step. The results show that the residual error can converge to 30% of the initial residual error or the threshold value in just 10 iterations. In

our study, iteration of the pressure Helmholtz equation is stopped when the residual error reaches $0.3\varepsilon_m$ or the threshold.

6. Conclusions

A novel fractional step method is proposed by this paper to decouple the temperature, velocity, and pressure variances in the Jacobian matrix of the compressible Navier–Stokes solver, by which the traditional fractional step algorithm has been extended to the simulation of compressible flows. In contrast to the widely used LU-SGS scheme, the proposed method enhances the convergence speed of the residual error for the subsonic flows, particularly for the low Mach number flows.

To accelerate the convergence, Jacobian and residual errors were multiplied by a preconditioning matrix, which helped optimize the distribution of the eigenvalues. Combined with a proposed factorization, the Jacobian matrix was further decomposed into two parts. A proper choice of the time step can guarantee that the submatrices are positive definite regardless of the Mach number, which renders the computing robust and efficient.

An optimal numerical boundary condition was developed in this study by reconstructing the eigenvalues and eigenvectors of the Jacobian for which the preconditioning was taken into account.

This numerical method was applied to the simulations of a one-dimensional Euler equation with large amplitude of acoustic waves, an adiabatic turbulent boundary layer and a turbulent boundary layer with significant heat transfer. The numerical results show fairly good agreement with both DNS and experimental results. The results show that, compared with the pressure-based method, the present method has a better convergent route because of the better distributed eigenvalues. Compared with the density-based method, the present method can correctly capture the large amplitude of the acoustic waves because the artificial damping is avoided.

Acknowledgments

The authors are grateful to the Air Force Office of Scientific Research for their support through grant F49620-01-1-0113. Iowa State High Performance Computing Center and University of Minnesota Supercomputing Institute provided computational resources needed for this research.

Appendix

Suppose matrix A to be given by

$$A = \begin{pmatrix} b & c & 0 & 0 & 0 \\ d & a & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & a & 0 \\ 0 & e & 0 & 0 & a \end{pmatrix}$$

Then the eigenvalue of matrix A will be $\lambda_1 = \frac{a+b-f}{2}, \lambda_2 = \frac{a+b+f}{2}, \lambda_3 = a, \lambda_4 = a$, and $\lambda_5 = a$, where

$$f = \sqrt{(a-b)^2 + 4cd}$$

Denote the diagonal matrix Λ by $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5]$. Hence, there exists an invertible matrix Q such that

$$A = Q\Lambda Q^{-1}$$

It is easy to know that

$$Q = \begin{pmatrix} q_{11} & q_{12} & 0 & 0 & 0 \\ q_{21} & q_{22} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad Q^{-1} = \begin{pmatrix} p_{11} & p_{12} & 0 & 0 & 0 \\ p_{21} & p_{22} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ p_{51} & p_{52} & 0 & 0 & 1 \end{pmatrix}$$

where

$$q_{11} = \frac{2cd + (a-b)(a-b+f)}{2de}, \quad q_{12} = \frac{2cd + (a-b)(a-b-f)}{2de}$$
$$q_{21} = \frac{b-a-f}{2e}, \quad q_{22} = \frac{b-a+f}{2e}$$
$$p_{11} = \frac{e(b-a+f)}{2cf}, \quad p_{12} = \frac{[(a-b)(f-a+b) - 2cd]e}{2cdf}$$
$$p_{21} = \frac{e(a-b+f)}{2cf}, \quad p_{22} = \frac{[(a-b)(f+a-b) + 2cd]e}{2cdf}$$
$$p_{51} = \frac{-e}{c}, \quad p_{52} = \frac{(b-a)e}{cd}$$

References

[1] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[2] R. Temam, Sur I'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires ii, Arch. Rat. Mech. Anal. 33 (1969) 377–385.
[3] J. Kim, P. Moin, Application of a fractional step method to incompressible flows, J. Comput. Phys. 59 (1985) 308–323.
[4] P. Bradshaw, Compressible turbulent shear layers, Ann. Rev. Fluid. Mech. 9 (1977) 33–54.
[5] F. Nicoud, P. Bradshaw, A velocity transformation for heat and mass transfer, Phys. Fluids 12 (1) (2000) 237–238.
[6] P.G. Huang, P. Bradshaw, T.J. Coakley, Skin friction and velocity profile family for compressible turbulent boundary layers, AIAA J. 31 (9) (1993) 33–54.
[7] K. Liu, R.H. Pletcher, Compressibility and variable density effects of turbulent boundary layers, J. Heat Transfer 129 (2007) 1–9.
[8] S. Yoon, A. Jameson, An LU-SGS scheme for the Euler and Navier–Stokes equations, AIAA paper 87-600.
[9] A.G. Tomboulides, S.A. Orzag, A quasi-two-dimensional benchmark problem for low Mach number compressible codes, J. Comput. Phys. 146 (1998) 691–706.
[10] R.H. Pletcher, K.-H. Chen, On solving the compressible Navier–Stokes equations for unsteady flows at very low Mach numbers, AIAA Paper 3368, 1993.
[11] E. Turkel, Preconditioning techniques in computational fluid dynamics, Annu. Rev. Fluid Mech. 31 (1999) 385–416.
[12] K.C. Karki, S.V. Patankar, Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations, AIAA J. 27 (1989) 1167.
[13] H. Bijl, P. Wesseling, A unified method for computing incompressible and compressible flows in boundary fitted coordinates, J. Comput. Phys. 141 (1998) 153–173.
[14] C. Wall, C.D. Pierce, P. Moin, A semi-implicit method for resolution of acoustic waves in low mach number flows, J. Comput. Phys. 181 (2002) 545–563.
[15] P.R. Spalart, Direct numerical simulation of turbulent boundary layer up to $R_\theta = 1410$, J. Fluid. Mech. 187 (1988) 61–98.
[16] R.S. Varga, Matrix Iterative Analysis, Springer-Verlag, New York, 2000 (Chapter 3).
[17] J.B. Perot, An analysis of the fractional step Method, J. Comput. Phys. 108 (1993) 51–58.
[18] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, J. Comput. Phys. 113 (1994) 1–4.
[19] F.E. Ham, F.S. Lien, A.B. Strong, A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids, J. Comput. Phys. 177 (2002) 117–133.
[20] M.J. Lee, B.D. Oh, Y.B. Kim, Canonical fractional-step methods and consistent boundary conditions for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 73–100.
[21] T.J. Poinsot, S.K. Lele, Boundary conditions for direct simulations of compressible viscous flows, J. Comput. Phys. 101 (1992) 104–129.
[22] K. Liu, R.H. Pletcher, Inflow conditions for the large eddy simulation of turbulent boundary layer: a dynamic recycling procedure, J. Comput. Phys. 219 (2006) 1–6.

[23] D.B. DeGraaff, J.K. Eaton, Reynolds-number scaling of the flat-plate turbulent boundary layer, J. Fluid. Mech. 422 (2000) 319–346.
[24] X. Xu, J.S. Lee, R.H. Pletcher, A compressible finite volume formulation for large eddy simulation of turbulent pipe flows at low Mach number in Cartesian coordinates, J. Comput. Phys. 203 (2005) 22–48.
[25] B.A. Kadar, Temperature and concentration profiles in fully turbulent boundary layers, Int. J. Heat Mass Transfer. 24 (9) (1981) 1541–1544.
[26] S.C. Eisenstat, H.P. Walker, Globally convergent inexact newton methods, SIAM J. Optimiz. 4 (2) (1994) 393–422.
[27] I. Yavneh, Multigrid smoothing factors for red-black Gauss–Seidel relaxation applied to a class of elliptic operators, SIAM J. Numer. Anal. 32 (4) (1995) 1126–1138.
[28] J. Zhang, Fast and high accuracy multigrid solution of the three dimensional poisson equation, J. Comput. Phys. 143 (1998) 449–461.