Journal of Statistical Physics, Vol. 107, Nos. 1/2, April 2002 (© 2002)

Evaluation of PowerFLOW for Aerodynamic Applications

David P. Lockard,[1] Li-Shi Luo,[2] Seth D. Milder,[2, 3] and Bart A. Singer[1]

Received December 21, 2000; accepted July 27, 2001

A careful comparison of the performance of a commercially available Lattice-Boltzmann Equation solver (PowerFLOW) was made with a conventional, block-structured computational fluid-dynamics code (CFL3D) for the flow over a two-dimensional NACA-0012 airfoil. The results suggest that the version of PowerFLOW used in the investigation produced solutions with large errors in the computed flow field; these errors are attributed to inadequate resolution of the boundary layer for reasons related to grid resolution and primitive turbulence modeling. The requirement of square grid cells in the PowerFLOW calculations limited the number of points that could be used to span the boundary layer on the wing and still keep the computation size small enough to fit on the available computers. Although not discussed in detail, disappointing results were also obtained with PowerFLOW for a cavity flow and for the flow around a generic helicopter configuration.

KEY WORDS: PowerFLOW; lattice Boltzmann method; lattice gas automata; 2D flow past NACA-0012 airfoil; aerodynamic simulations; turbulence modeling; wall function.

1. INTRODUCTION

In recent years, methods of the lattice-gas automata (LGA) and the lattice Boltzmann equation (LBE) have become an alternative to conventional computational fluid dynamics (CFD) methods for various systems (see collections of papers,[1] proceedings,[2, 3] reviews,[4–6] and monographs[7–9]).

[1] Computational Modeling and Simulation Branch, Mail Stop 128, NASA Langley Research Center, Hampton, Virginia 23681-2199; e-mails: d.p.lockard@larc.nasa.gov and b.a.singer@larc.nasa.gov

[2] ICASE, Mail Stop 132C, NASA Langley Research Center, 3 West Reid Street, Building 1152, Hampton, Virginia 23681-2199; e-mail: luo@icase.edu

[3] School of Computational Sciences, MS 5C6, George Mason University, 4400 University Drive, Fairfax, Virginia 22030.

The lattice-gas and lattice Boltzmann methods have been particularly successful in dealing with fluid flow applications involving complex fluids and complicated boundaries, such as multi-component fluids through porous media.(10, 11) Although the promising potential of the LGA and LBE methods as viable CFD tools has been demonstrated in a number of cases of laminar flow with low Reynolds number, there is no experience to demonstrate that the method can be applied to turbulent flow in the presence of a streamlined body at high Reynolds number. In fact, application of the LBE method to turbulent flows at high Reynolds number remains as an area of future development.(12, 6)

The fundamental philosophy of the lattice gas automata and the lattice Boltzmann equation is to construct simple models based on kinetic theory that preserve the conservation laws and necessary symmetries such that the emerging behavior of these models obeys the desired macroscopic equations. The theoretical foundation for using simplified kinetic models to simulate hydrodynamic systems (or other complex systems) is based upon the observation that in nature macroscopic phenomenon such as hydrodynamics is rather insensitive to the underlying details in the microscopic, or even mesoscopic, dynamics. In hydrodynamic systems, details of microscopic dynamics can affect the numerical values of the transport coefficients, but not the form of the Navier–Stokes equations. Therefore, as long as these models preserve conservation laws and necessary symmetries (e.g., Galilean invariance) satisfactorily, they can be used to simulate hydrodynamic systems. The primary design criteria for the development of LGA and LBE models are efficiency and the preservation of the physical properties of the system. The LGA and LBE models resemble molecular dynamics methods or the Boltzmann kinetic equation in concept, but do not have their computational complexity. However, some artificial physical limitations are typically placed on the system to improve the efficiency of the algorithm.

The kinetic nature of the LGA and LBE methods leads to the following features that distinguish the LGA and LBE methods from any other conventional CFD method. First, the convection operator of the LGA or LBE models is linear in phase space, similar to that of the Boltzmann kinetic equation but different than the Euler or the Navier–Stokes equations. Second, the pressure is obtained through an equation of state, as opposed to solving a Poisson equation in the incompressible Navier–Stokes equations. Third, with the Navier–Stokes equations, the constitutive relations are input from empirical data, while with the LGA and LBE methods, the constitutive relations emerge as a result of proper modeling of interparticle potentials. Fourth, unlike the Navier–Stokes or Euler equations, in which macroscopic conservation laws are discretized, the LGA and LBE methods utilize a set of discrete particle velocities such that the conserved quantities are preserved up to machine accuracy in the calculations. In addition, the LGA and LBE methods have the following computational characteristics:

1. Intrinsic parallelism due to nearest neighbor data communications of the streaming (convection) process and purely local calculation of the collision process;
2. Ability to handle complex boundaries without compromising computational speed.

The intrinsic parallelism of the LGA and LBE methods is an appealing feature in light of the clear trend to use massively parallel computers. The ability to handle complex boundaries is important to practical applications such as flow through porous media. The capability to include model interactions is crucial to simulations of some physical systems such as multiphase or multi-component flows. Furthermore, the LGA method has the following features, in addition to the above, due to its Boolean nature:

1. Unconditional stability;
2. Memory efficiency;
3. Suitability for special-purpose computers.

The theoretical advantages of the LGA and LBE methods are still largely untested in the realm of CFD. Moreover, precisely what advantages the LGA and LBE methods exhibit over conventional methods of solving the Navier–Stokes equations are expected to be problem dependent. The areas where the LGA and LBE methods are more suitable depend on the physical nature of the problems. Unfortunately, due to the lack of research effort in this area, a sufficient amount of numerical evidence to verify the theoretical advantages of the LGA and LBE methods does not exist, especially in the areas where conventional CFD methods have proven to be useful, such as in the field of aerodynamics.

Recently, a commercial CFD software product, PowerFLOW, based on the lattice-gas and the lattice Boltzmann methods became available.[13–16] PowerFLOW has been applied to simulate high Reynolds number turbulent external flow over an automobile body in ground proximity.[17] However, PowerFLOW has not been subjected to a set of thorough validation tests. The focus of this study is to explore the possibility of using PowerFLOW in typical aerodynamic applications of CFD simulations of various configurations. In this report, we present test results for a 2-D flow past a streamlined airfoil (NACA-0012)[18] with Reynolds numbers ($\mathrm{Re} = UL/\nu$, where U is the free-stream velocity, L is the chord length, and

ν is the kinematic viscosity) ranging from 500 to 0.5×10^9 using PowerFLOW and compare the results with those from a conventional CFD solver CFL3D[19–24] and the lattice Boltzmann nine-velocity model in two dimensions (D2Q9).[25, 26] Our test results regarding PowerFLOW are rather negative: in its present state, PowerFLOW, release 3.0 is inadequate for aerodynamic applications. Nevertheless, with further development, software based on LGA and LBE methods may prove to be useful tools for aerodynamics.

This report is organized as follows. In the next section, the basic notion of lattice-gas automata and the lattice Boltzmann equation is briefly reviewed. The lattice Boltzmann equation is presented as a special finite-difference form of the continuous Boltzmann equation, and the boundary conditions and turbulence models in the lattice Boltzmann method are discussed. Sections 3 and 4 briefly describe PowerFLOW and CFL3D, respectively. Section 5 presents the simulation results of flow past a NACA-0012 airfoil for Reynolds numbers between 500 and 0.5×10^9. Section 6 discusses the results and concludes this report.

2. LATTICE-GAS AUTOMATA AND LATTICE BOLTZMANN EQUATION

2.1. Lattice Gas Automata

In a series of articles published in the 1980's,[27] Wolfram showed that cellular automata, despite their simple construction, have sufficient complexity to accomplish universal computing: that is, beginning with a particular initial state, the evolution of some automaton could realize any chosen finite algorithm. Based upon kinetic theory and the previous experience of the Hardy–Pomeau–de Pazzis (HPP) model[28, 29] that a two-dimensional square lattice does not possess sufficient symmetry for hydrodynamics, Frisch, Hasslacher, and Pomeau[30] and Wolfram[31] independently discovered that a simple cellular automaton on a two-dimensional triangular lattice can simulate the Navier–Stokes equations. The two-dimensional Frisch–Hasslacher–Pomeau (FHP) model was immediately generalized to a three-dimensional LGA model.[32]

The LGA model proposed by Frisch, Hasslacher, and Pomeau, and Wolfram evolves on a two-dimensional triangular lattice space. The particles have momenta that allow them to move from one site on the lattice to another in discrete time steps. A particular lattice site is occupied by either no particle or one particle with a particular momentum pointing to a nearest neighbor site. Therefore, at most a lattice site can be simultaneously occupied by six particles, hence this model is called the six-velocity model

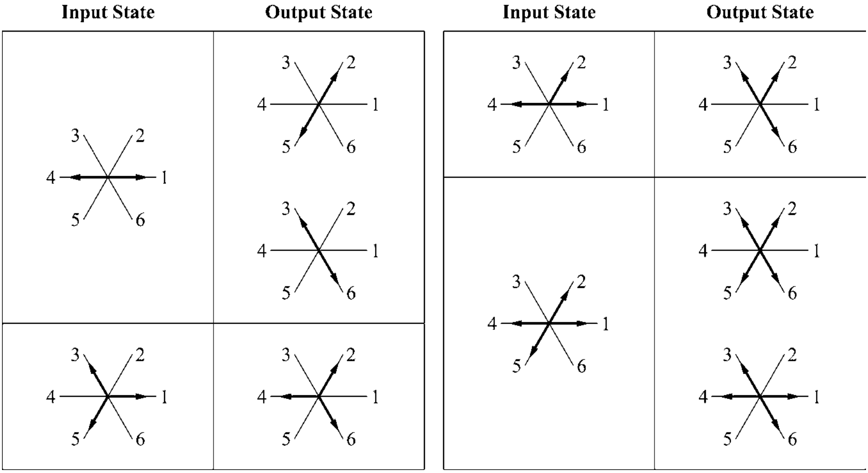

Fig. 1. Collisions of FHP LGA model. Note that the figure does not included those collisions that can be obtained by applying rotations of multiple $\pi/3$ to input and output states simultaneously.

or FHP-I model. The evolution of the LGA model consists of two steps: collision and advection. The collision process is partially described in Fig. 1. For example, two particles colliding with opposite momenta will rotate their momenta 60° clockwise or counter-clockwise with equal probability. In Fig. 1, we do not list those configurations which can be easily obtained by rotational transformation, and which are invariant under the collision process. The particle number, the momentum, and the energy are conserved in the collision process locally and exactly. (Because the FHP model has only one speed, the energy is no longer an independent variable, it is equivalent to the particle number. However, for multi-speed models, the energy is an independent variable.)

The evolution equation of the LGA can be written as:

$$n_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta_t, t+\delta_t) = n_\alpha(\mathbf{x}, t) + C_\alpha \tag{1}$$

where n_α is the Boolean particle number with the velocity $\mathbf{e}_\alpha$, C_α is the collision operator, $\mathbf{x}$ is a vector in the lattice space with lattice constant δ_x, t denotes discrete time with step size δ_t. We usually set both δ_x and δ_t to unity. The subscript α is an index for velocity; as illustrated in Fig. 1, for the FHP model, α runs from 1 to 6. In the figure, collisions that can be obtained by rotations of the input and output states are not shown. Also note that momentum must be conserved during collisions, therefore many output states must be equal to their input states. These trivial collisions are

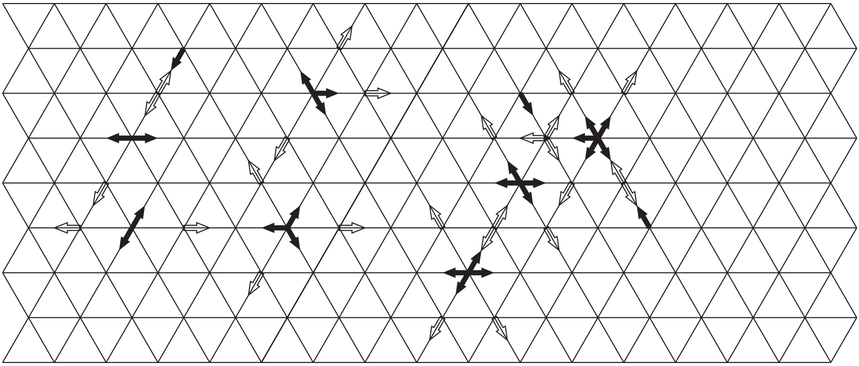

Fig. 2. Evolution of FHP LGA model. Solid and hollow arrows represent particles with velocities corresponding to times t and $t+1$, respectively. That is, the hollow arrows are the final configurations of the initial configurations of solid arrows after one cycle of collision and advection.

not shown in the figure. After colliding, particles advect to the next site according to their velocities. Figure 2 illustrates the evolution of the system in one time step from t to $t+\delta_t$. In this figure, solid and hollow arrows represent particles with corresponding velocity at time t and $t+\delta_t$, respectively. The system evolves by iteration of the collision and advection processes.

According to the collision rules prescribed in Fig. 1, the collision operator, C_α, can be written as follows:

$$C_\alpha(\{n_\alpha(\mathbf{x}, t)\}) = \sum_{\mathbf{s}, \mathbf{s}'} (s'_\alpha - s_\alpha)\, \xi_{\mathbf{ss}'} \prod_\sigma n_\sigma^{s_\sigma}(1-n_\sigma)^{(1-s_\sigma)} \tag{2}$$

where $\mathbf{s} \equiv \{s_1, s_2, \ldots, s_6\}$ and $\mathbf{s}' \equiv \{s'_1, s'_2, \ldots, s'_6\}$ are possible incoming and outgoing configurations at a given site $\mathbf{x}$ and time t, respectively; $\xi_{\mathbf{ss}'}$ is a Boolean random number in space and time that determines the transition between state $\mathbf{s}$ and $\mathbf{s}'$ satisfying the following normalization condition:

$$\sum_{\mathbf{s}'} \xi_{\mathbf{ss}'} = 1\,, \qquad \forall \mathbf{s} \tag{3}$$

The Boolean random number $\xi_{\mathbf{ss}'}$ must also have rotational symmetry, i.e., for any states $\mathbf{s}$ and $\mathbf{s}'$, $\xi_{\mathbf{ss}'}$ is invariant if states $\mathbf{s}$ and $\mathbf{s}'$ are both subjected to simultaneous clockwise or counterclockwise rotations. It is obvious that for Boolean numbers n_σ and s_σ, the following equation holds:

$$n_\sigma^{s_\sigma}(1-n_\sigma)^{(1-s_\sigma)} = \delta_{n_\sigma s_\sigma} \tag{4}$$

where $\delta_{n_\sigma s_\sigma}$ is the Kronecker delta symbol with two indices. Therefore, Eq. (2) can be written as

$$C_\alpha(\{n_\alpha(\mathbf{x}, t)\}) = \sum_{\mathbf{s},\,\mathbf{s}'} (s'_\alpha - s_\alpha)\,\xi_{\mathbf{ss}'}\delta_{\mathbf{ns}} \tag{5}$$

where $\delta_{\mathbf{ns}} \equiv \delta_{n_1 s_1}\delta_{n_2 s_2}\cdots\delta_{n_b s_b}$. Equation (2), or (5), is rather abstract. A specific example of the collision operator for the two-body collision is

$$\begin{aligned} C^{(2)}_\alpha = {} & \xi^{(2)}_R n_{\alpha+1} n_{\alpha+4} \bar{n}_\alpha \bar{n}_{\alpha+2} \bar{n}_{\alpha+3} \bar{n}_{\alpha+5} + \xi^{(2)}_L n_{\alpha+2} n_{\alpha+5} \bar{n}_\alpha \bar{n}_{\alpha+1} \bar{n}_{\alpha+3} \bar{n}_{\alpha+4} \\ & - (\xi^{(2)}_R + \xi^{(2)}_L)\, n_\alpha n_{\alpha+3} \bar{n}_{\alpha+1} \bar{n}_{\alpha+2} \bar{n}_{\alpha+4} \bar{n}_{\alpha+5} \end{aligned} \tag{6}$$

where $\bar{n}_\alpha \equiv 1 - n_\alpha$ is the complement of n_α, $\xi^{(2)}_R$ and $\xi^{(2)}_L$ are Boolean random numbers which determine the outcome of head-on two-body collisions. The Boolean random numbers reflect the randomness of the outcomes of the two-body collision. Obviously, for the collision operator to satisfy the complete lattice symmetry group statistically (on average), they must satisfy

$$\langle \xi^{(2)}_R \rangle = \langle \xi^{(2)}_L \rangle \tag{7}$$

where $\langle \cdot \rangle$ denotes the ensemble average. The conservation laws of the particle number, momentum, and energy of the LGA micro-dynamics can be written as follows:

$$\sum_\alpha (s'_\alpha - s_\alpha) = 0, \tag{8a}$$

$$\sum_\alpha (s'_\alpha - s_\alpha)\,\mathbf{e}_\alpha = 0, \tag{8b}$$

$$\sum_\alpha (s'_\alpha - s_\alpha)\,\tfrac{1}{2}\mathbf{e}^2_\alpha = 0 \tag{8c}$$

In practice, the collision can be implemented with various algorithms. One can either use logical operations [as indicated by Eq. (6)], or by table-lookup. The collision rules shown in Fig. 1 can also be represented by a collision table, as shown by Table I. In Table I, each bit in a binary number represents a particle number n_α, $\alpha = 1, 2, \ldots, 6$, from right to left. The limitation of table lookup is the size of the table, which is 2^b, where b is the number of bits of the model. Both logic operations and table lookup can be extremely fast on digital computers, and especially so on dedicated computers.(33–35)

Table I. Collision Table for the Six-Velocity FHP Model on a Triangular Lattice

Input State	Output State
001001	010010
	100100
010101	101010
001011	100110
011011	110110
	101101

2.2. Hydrodynamics of Lattice Gas Automata

The ensemble average of Eq. (1) leads to a lattice Boltzmann equation:

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta_t,\, t+\delta_t) = f_\alpha(\mathbf{x},t) + \Omega_\alpha \tag{9}$$

where $f_\alpha \equiv m\langle n_\alpha \rangle$ and $\Omega_\alpha \equiv \langle C_\alpha \rangle$, where m is the particle mass. In addition, correlations among colliding particles are assumed to be negligible, i.e.,

$$\langle n_\alpha n_\beta \cdots n_\gamma \rangle = \langle n_\alpha \rangle \langle n_\beta \rangle \cdots \langle n_\gamma \rangle \tag{10}$$

The above approximation is equivalent to the celebrated molecular chaos assumption of Boltzmann (*Stosszahlansatz*).[36] With the molecular chaos approximation, the lattice Boltzmann collision operator is given by

$$\Omega_\alpha(\{f_\alpha(\mathbf{x},t)\}) = \sum_{\mathbf{s},\,\mathbf{s}'} (s'_\alpha - s_\alpha)\, A_{\mathbf{ss}'} \prod_\sigma f_\sigma^{s_\sigma}(1-f_\sigma)^{(1-s_\sigma)} \tag{11}$$

where $A_{\mathbf{ss}'} \equiv \langle \xi_{\mathbf{ss}'} \rangle$ is the transition probability from state $\mathbf{s}$ and $\mathbf{s}'$. The hydrodynamic moments are given by:

$$\rho = \sum_\alpha f_\alpha\,, \qquad \rho\mathbf{u} = \sum_\alpha \mathbf{e}_\alpha f_\alpha\,, \qquad \rho\varepsilon = \sum_\alpha \tfrac{1}{2}(\mathbf{e}_\alpha - \mathbf{u})^2 f_\alpha \tag{12}$$

where ρ, $\mathbf{u}$, and ε are the mass density, the velocity, and the internal energy density, respectively.

Equation (9) can be expanded in a Taylor series of δ_t up to the second order:

$$(\partial_t + \mathbf{e}_\alpha \cdot \nabla) f_\alpha + (\partial_t + \mathbf{e}_\alpha \cdot \nabla)^2 f_\alpha = \Omega_\alpha \tag{13}$$

The equilibrium distribution, $f_\alpha^{(0)}$, which is the solution of $\Omega_\alpha(\{f_\alpha\}) = 0$, must be a Fermi–Dirac distribution because the system is a binary one,[31] that is,

$$f_\alpha^{(0)} = \frac{1}{1 + \exp(a + b\mathbf{u} \cdot \mathbf{e}_\alpha)} \tag{14}$$

where coefficients a and b are functions of ρ and $\mathbf{u}^2$ in general. Because the coefficients a and b in $f_\alpha^{(0)}$ cannot be determined exactly,[31] $f_\alpha^{(0)}$ must be expanded in a Taylor series of $\mathbf{u}$—the small velocity (low Mach number) expansion. With the small velocity expansion of the equilibrium $f_\alpha^{(0)}$ and through Chapman–Enskog analysis, one can obtain the following hydrodynamic equations from the FHP LGA model:[30, 31]

$$\partial_t \rho + \nabla \cdot \rho \mathbf{u} = 0, \tag{15a}$$

$$\partial_t \rho \mathbf{u} + \nabla(g \rho \mathbf{u}\mathbf{u}) = -\nabla P + \nu \nabla^2 \rho \mathbf{u} \tag{15b}$$

where g is a function of ρ,

$$P = c_s^2 \rho \left[1 - g \frac{\mathbf{u}^2}{c^2} \right], \tag{16a}$$

$$c_s = \frac{1}{\sqrt{2}} c, \tag{16b}$$

$$\nu = -\frac{1}{4} \left(\frac{1}{\lambda} + \frac{1}{2} \right) c \delta_x \tag{16c}$$

c_s is the sound speed, $c = \delta_x / \delta_t$, and λ is an eigenvalue of the linearized collision operator:[7]

$$J_{\alpha\beta} \equiv \left. \frac{\partial \Omega_\alpha}{\partial f_\beta} \right|_{f_\beta = f_\beta^{(0)}}$$

The defects of the LGA hydrodynamics are obvious from the above equations [Eqs. (15a) and (16a)]:

1. Simulations are intrinsically noisy because of the large fluctuation in n_α;

2. The factor $g(\rho)$ is not unity, thus Galilean invariance is destroyed;

3. Increasing the Reynolds number Re is difficult;

4. The equation of state depends on $\mathbf{u}^2$ (unphysical);

5. Spurious (unphysical) conserved quantities exist due to the simple symmetry of the lattice-gas automata.

However, all these defects can be fixed by using more sophisticated LGA models,[37] or the other alternative—the lattice Boltzmann equation.

2.3. Lattice Boltzmann Equation

Historically, models of the lattice Boltzmann equation[38, 39] evolved from their Boolean counterparts: the lattice-gas automata.[30, 31] Recently, the lattice Boltzmann equation has been shown to be a special discretized form of the continuous Boltzmann equation.[40–42]

For the sake of simplicity but without lose of generality, the Boltzmann equation with the Bhatnagar–Gross–Krook (BGK) approximation[43] is used in the following analysis. The BGK Boltzmann equation can be written in the form of an ordinary differential equation:

$$D_t f + \frac{1}{\lambda} f = \frac{1}{\lambda} f^{(0)} \tag{17}$$

where $D_t \equiv \partial_t + \xi \cdot \nabla$ is the Lagrangian derivative along the (microscopic) particle velocity ξ, $f \equiv f(\mathbf{x}, \xi, t)$ is the single particle distribution function, λ is the relaxation time due to collision, and $f^{(0)}$ is the Maxwell–Boltzmann distribution function:

$$f^{(0)} \equiv \frac{\rho}{(2\pi RT)^{D/2}} \exp\left[-\frac{(\xi - \mathbf{u})^2}{2\theta}\right] \tag{18}$$

in which D is the dimension of the space; $\theta = k_B T/m$ is the normalized temperature; T, k_B and m are temperature, the Boltzmann constant, and particle mass, respectively. The macroscopic variables are the moments of the distribution function f with respect to particle velocity ξ:

$$\rho = \int f \, d\xi\,, \qquad \rho\mathbf{u} = \int \xi f \, d\xi\,, \qquad \rho\theta = \tfrac{1}{2}\int (\xi - \mathbf{u})^2 f \, d\xi \tag{19}$$

Equation (17) can be formally integrated over a time interval δ_t:

$$f(\mathbf{x} + \xi\delta_t, \xi, t + \delta_t) = e^{-\delta_t/\lambda} f(\mathbf{x}, \xi, t) + \frac{1}{\lambda} e^{-\delta_t/\lambda} \int_0^{\delta_t} e^{t'/\lambda} f^{(0)}(\mathbf{x} + \xi t', \xi, t + t')\, dt' \tag{20}$$

Assuming that δ_t is small enough and $f^{(0)}$ is smooth enough locally, and neglecting the terms of the order $O(\delta_t^2)$ or smaller in the Taylor expansion of the right hand side of Eq. (20), we obtain

$$f(\mathbf{x}+\xi\delta_t, \xi, t+\delta_t)-f(\mathbf{x}, \xi, t)=-\frac{1}{\tau}\left[f(\mathbf{x}, \xi, t)-f^{(0)}(\mathbf{x}, \xi, t)\right] \tag{21}$$

where $\tau \equiv \lambda/\delta_t$ is the dimensionless relaxation time.

The equilibrium distribution function $f^{(0)}$ can be expanded as a Taylor series in $\mathbf{u}$. By retaining the Taylor expansion up to $\mathbf{u}^2$, we obtain:

$$f^{(\mathrm{eq})}=\frac{\rho}{(2\pi\theta)^{D/2}}\exp\left(-\frac{\xi^2}{2\theta}\right)\left[1+\frac{(\xi\cdot\mathbf{u})}{\theta}+\frac{(\xi\cdot\mathbf{u})^2}{2\theta^2}-\frac{\mathbf{u}^2}{2\theta}\right] \tag{22}$$

For the purpose of deriving the Navier–Stokes equations, the above second-order expansion is sufficient.

To derive the Navier–Stokes equations, the following moment integral must be evaluated exactly:

$$\int \xi^m f^{(\mathrm{eq})}\, d\xi \tag{23}$$

where $0 \leqslant m \leqslant 3$ for isothermal models. The above integral contains the following integral which can be evaluated by Gaussian-type quadrature:

$$I=\int \exp(-\xi^2/2\theta)\,\psi(\xi)\,d\xi=\sum_\alpha W_\alpha \exp(-\xi_\alpha^2/2\theta)\,\psi(\xi_\alpha) \tag{24}$$

where $\psi(\xi)$ is a polynomial in ξ, and W_α and ξ_α are the weights and the abscissas (or discrete velocities) of the quadrature, respectively. Accordingly, the hydrodynamic moments of Eqs. (19) can be computed by quadrature as well:

$$\rho=\sum_\alpha f_\alpha\,, \qquad \rho\mathbf{u}=\sum_\alpha \xi_\alpha f_\alpha\,, \qquad \rho\theta=\tfrac{1}{2}\sum_\alpha (\xi_\alpha-\mathbf{u})^2 f_\alpha \tag{25}$$

where $f_\alpha \equiv f_\alpha(\mathbf{x}, t) \equiv W_\alpha f(\mathbf{x}, \xi_\alpha, t)$. We shall use the nine-velocity (9-bit) isothermal LBE model on square lattice space as an example to illustrate the derivation of LBE models: the evolution equation (21) on a discretized phase space and time with a proper equilibrium distribution function leads to the Navier–Stokes equations.

To derive the nine-velocity LBE model, a Cartesian coordinate system is used, and accordingly, we set $\psi(\xi) = \xi_x^m \xi_y^n$. The integral of Eq. (24) becomes:

$$I = (\sqrt{2\theta})^{(m+n+2)}\, I_m I_n \tag{26}$$

where

$$I_m = \int_{-\infty}^{+\infty} e^{-\zeta^2} \zeta^m \, d\zeta \tag{27}$$

and $\zeta = \xi_x/\sqrt{2\theta}$ or $\xi_y/\sqrt{2\theta}$. Naturally, the third-order Hermite formula is the optimal choice to evaluate I_m for the purpose of deriving the nine-velocity LBE model, i.e., $I_m = \sum_{j=1}^{3} \omega_j \zeta_j^m$. The three abscissas ($\zeta_j$) and the corresponding weights (ω_j) of the quadrature are:

$$\begin{aligned} &\zeta_1 = -\sqrt{3/2}, &&\zeta_2 = 0, &&\zeta_3 = \sqrt{3/2}, \\ &\omega_1 = \sqrt{\pi}/6, &&\omega_2 = 2\sqrt{\pi}/3, &&\omega_3 = \sqrt{\pi}/6 \end{aligned} \tag{28}$$

Then, the integral of Eq. (26) becomes:

$$I = 2\theta \left[\omega_2^2 \psi(\mathbf{0}) + \sum_{\alpha=1}^{4} \omega_1 \omega_2 \psi(\xi_\alpha) + \sum_{\alpha=5}^{8} \omega_1^2 \psi(\xi_\alpha) \right] \tag{29}$$

where ξ_α is the zero velocity vector for $\alpha = 0$, the vectors of $\sqrt{3\theta}(\pm 1, 0)$ and $\sqrt{3\theta}(0, \pm 1)$ for $\alpha = 1$–4, and the vectors of $\sqrt{3\theta}(\pm 1, \pm 1)$ for $\alpha = 5$–8. Note that the above quadrature is exact for $(m+n) \leqslant 5$.

Now the momentum space is discretized with nine discrete velocities $\{\xi_\alpha \mid \alpha = 0, 1, \ldots, 8\}$, as shown in Fig. 3. To obtain the nine-velocity model, the configuration space is discretized accordingly, i.e., it is discretized into a square lattice with lattice constant $\delta_x = \sqrt{3\theta}\,\delta_t$. It should be stressed that

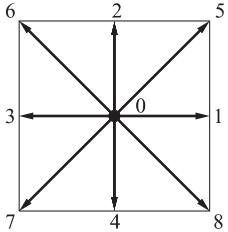

Fig. 3. Discrete velocities of the nine-velocity model on a 2D square lattice.

the temperature θ has no physical significance here because we are only dealing with an isothermal model. We can therefore choose δ_x to be a fundamental quantity instead, thus $\sqrt{3\theta} = c \equiv \delta_x/\delta_t$, or $\theta = c_s^2 = c^2/3$, where c_s is the sound speed of the model.

By comparing Eqs. (24) and (29), we can identify the weights defined in Eq. (24):

$$W_\alpha = 2\pi\theta \exp(\xi_\alpha^2/2\theta)\, w_\alpha \tag{30}$$

where

$$w_\alpha = \begin{cases} 4/9, & \alpha = 0, \\ 1/9, & \alpha = 1, 2, 3, 4, \\ 1/36, & \alpha = 5, 6, 7, 8 \end{cases} \tag{31}$$

Then, the equilibrium distribution function of the nine-velocity model is:

$$\begin{aligned} f_\alpha^{(\mathrm{eq})} &= W_\alpha f^{(\mathrm{eq})}(\mathbf{x}, \xi_\alpha, t) \\ &= w_\alpha \rho \left\{ 1 + \frac{3(\mathbf{e}_\alpha \cdot \mathbf{u})}{c^2} + \frac{9(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right\} \end{aligned} \tag{32}$$

where

$$\mathbf{e}_\alpha = \begin{cases} (0, 0), & \alpha = 0\,, \\ (\cos\phi_\alpha, \sin\phi_\alpha)\, c, & \alpha = 1, 2, 3, 4, \\ (\cos\phi_\alpha, \sin\phi_\alpha)\sqrt{2}\, c, & \alpha = 5, 6, 7, 8 \end{cases} \tag{33}$$

and $\phi_\alpha = (\alpha - 1)\,\pi/2$ for $\alpha = 1$–4, and $(\alpha - 5)\,\pi/2 + \pi/4$ for $\alpha = 5$–8, as shown in Fig. 3. The Navier–Stokes equation derived from the above LBE model is:

$$\rho\partial_t \mathbf{u} + \rho\mathbf{u} \cdot \nabla\mathbf{u} = -\nabla p + \rho\nu\nabla^2\mathbf{u} \tag{34}$$

where the equation of state is the ideal gas one, $p = c_s^2 \rho$, the sound speed $c_s = c/\sqrt{3}$, and the viscosity

$$\nu = \tfrac{1}{3}(\tau - 1/2)\, c\delta_x = (\tau - 1/2)\, c_s^2 \delta_t \tag{35}$$

for the nine-velocity model.

Similarly, we can also derive two-dimensional six-velocity, seven-velocity, and three-dimensional twenty-seven-velocity LBE models.(41)

In the above derivation, the discretization of phase space is accomplished by discretizing the momentum space in such a way that a lattice structure in configuration space is simultaneously obtained. That is, the discretization of configuration space is determined by that of momentum space. Of course, the discretization of momentum space and configuration space can be done independently. This consideration has two immediate consequences: arbitrary mesh grids and significant enhancement of the Reynolds number in LBE hydrodynamic simulations.

To implement arbitrary mesh grids with the LBE method, one first discretizes the configuration space by generating a mesh adapted to the physics of the particular problem. Then at each grid point, one can discretize momentum space as before. Now, a local LBE is built on each mesh grid point. The evolution of this discretized Boltzmann equation (DBE) consists of the following three steps. The first two steps are the usual collision and advection process as in the previous LBE models. After collision and advection, interpolation follows. The interpolation process is what distinguishes the DBE from the LBE method. Because the mesh grids can be arbitrary, the distribution function f_α at one mesh grid point, say $\mathbf{x}_i$, cannot go to another grid point in general through the advection process as it can in previous LBE models. Therefore, the interpolation step becomes necessary to construct $f_\alpha(\mathbf{x}_i, t)$ on each and every mesh grid point from $f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha \delta_t, t)$ after the advection process. Of course, interpolation brings in additional numerical error, but it can be justified so long as the error induced by interpolation does not affect the DBE algorithm as a whole.(44) In addition, the separate discretization of momentum and configuration space allows one to increase the Reynolds number significantly in numerical simulations without enlarging mesh sizes or decreasing the viscosity by adjusting τ.(44) In other words, the limitation posed by the lattice Reynolds number is completely overcome(45) and the stability of the LBE method is greatly improved.(44, 45)

For the sake of simplicity, so far we have only discussed the lattice BGK model based on the BGK equation.(43) With a single relaxation parameter, the BGK model has some obvious shortcomings such as fixed Prandtl number ($\mathrm{Pr} = 1$ for BGK models) and fixed ratio between the kinematic and bulk viscosities. These shortcomings due to the single-relaxation-time approximation can be overcome by introducing the multiple-relaxation-time lattice Boltzmann model.(46, 26, 47) The lattice BGK model is thus a special case of the multiple-relaxation-time lattice Boltzmann model. As opposed to the above derivation which connects the lattice Boltzmann equation to the continuous Boltzmann equation, the lattice

Boltzmann equation can also be presented as a self-contained mathematical object representing a dynamical system with a finite number of moments in discrete space and time (equivalent moment system).(46, 26, 47) It can be explicitly shown that the lattice Boltzmann equation is a relaxation system for the incompressible Navier–Stokes equations, with artificial compressibility.(48, 49) It can also be shown that the lattice Boltzmann equation approximates the incompressible Navier–Stokes equations with a second-order centered difference in space (using non-standard stencils) and a first-order Euler discretization in time.(48, 49)

2.4. Boundary Conditions and Turbulence Modeling

The boundary conditions for the lattice gas automata are relatively simple. The simplest boundary conditions for the velocity field are bounce-back and reflective boundary conditions. The former mimics the noslip boundary condition and the latter simulates the slip boundary condition. In the bounce back boundary condition, a particle hitting a wall simply reverses its momentum, whereas in the reflective boundary condition the particle hitting a solid boundary only reverses its momentum normal to the boundary and maintains its momentum tangential to the boundary. In addition, fully developed boundary conditions for the velocity field can be obtained by equating the distribution functions in two adjacent lattice lines. The pressure boundary condition can be implemented *via* the ideal-gas equation of state by setting appropriate mass densities in the desired locations. Body forces can also be implemented by probabilistically making particles gain momentum along the forcing direction. The Boolean nature of the lattice gas automata sometimes makes the boundary conditions difficult to alter. Furthermore, the accuracy of the aforementioned boundary conditions could be of first order (in space), instead of second order.

The lattice Boltzmann method has greater freedom than lattice gas automata methods to implement various boundary conditions accurately.(50) Various conventional techniques such as interpolation or extrapolation can be applied to improve the accuracy of the LBE boundary conditions.(50) Furthermore, arbitrary nonuniform meshes(44) and mesh refinement(51) can be implemented in the LBE method.

There are two types of turbulence models implemented in the lattice Boltzmann method: subgrid modeling(52) and wall function modeling.(17) The subgrid modeling is illustrated as follows by using the example of the Smagorinsky subgrid model. In a subgrid model, the total viscosity v can be divided in two parts,

$$v = v_0 + v_s \tag{36}$$

where ν_0 and ν_s are the physical viscosity (for the resolved scales) and eddy viscosity (for the unresolved subscales), respectively. The eddy viscosity is obtained as follows:

$$\nu_s = (C_S \delta_x)^2 \, |\bar{S}| \tag{37a}$$

$$|\bar{S}| = \sqrt{2 \sum_{i,j} \bar{S}_{ij} \cdot \bar{S}_{ij}} \tag{37b}$$

$$\bar{S}_{ij} \equiv \tfrac{1}{2}(\partial_j \bar{u}_i + \partial_i \bar{u}_j) \tag{37c}$$

where $\bar{\mathbf{u}}$ is the resolved velocity field, $\bar{S}_{ij}$ is the corresponding strain-rate tensor, and $C_S > 0$ is the Smagorinsky constant. The relaxation time is evaluated at each grid as follows:

$$\tau = 6(\nu_0 + \nu_s) + \tfrac{1}{2} \tag{38}$$

where ν_0 is given *a priori*, as determined by the Reynolds number of the simulated flow. The relaxation time now varies locally to model the subgrid-scale dynamics, depending on the flow field.

An alternative to the above subgrid modeling of turbulence is through a wall function.[17] Instead of resolving the boundary layer adjacent to the wall, the computation starts at the first cell above the surface, where a "slip" velocity U_s is typically nonzero.[53] The turbulent law of the wall is assumed to hold at the location y_s of the first cell off the wall; hence, the shear stress at y_s is the same as the wall shear stress τ_w. A local skin-friction coefficient C''_f is defined through the relation

$$\tau_w \equiv C''_f \, \tfrac{1}{2} \, \rho U_s^2 \tag{39}$$

The local friction velocity U_τ is defined as

$$U_\tau \equiv \sqrt{\frac{\tau_w}{\rho}} \tag{40}$$

Application of the law of the wall at the distance y_s from the wall yields the logarithmic profile of the boundary layer

$$\frac{U_s}{U_\tau} = \frac{1}{\kappa}\left[\ln\left(\frac{U_\tau y_s}{\nu}\right) + B\kappa\right] \tag{41}$$

where κ is the von Karman constant ($\kappa \approx 0.4$) and B is a constant of integration ($B \approx 5.0$). From Eqs. (39) and (40),

$$\frac{U_s}{U_\tau} = \sqrt{\frac{2}{C_f''}} \tag{42}$$

and the friction coefficient can be expressed as

$$C_f'' = \frac{2\kappa^2}{\left[\ln\left(\frac{U_\tau y_s}{\nu}\right) + B\kappa\right]^2} \tag{43}$$

In a simple mixing-length model a turbulent, or eddy viscosity ν_t is proportional to the mean velocity gradient multiplied by the square of a length scale. If y_s is in the log layer, κy_s is an appropriate length scale and $U_\tau/(\kappa y_s)$ is the mean velocity gradient, hence

$$\nu_t = \kappa y_s U_\tau \tag{44}$$

Anagnost *et al.*[17] assume that the lattice viscosity ν^* is equal to the eddy viscosity ν_t at the height y_s of the first grid cell. Why such an assumption is necessarily valid is not clear; however, this assumption leads to the basic form of the relationship for the local skin-friction coefficient

$$C_f'' = \frac{2\kappa^2}{\left[\ln\left(\frac{\nu^*}{\kappa\nu}\right) + B\kappa\right]^2} \tag{45}$$

The effect of local adverse pressure gradients can be incorporated into the model by modifying the relationship between the lattice viscosity and the eddy viscosity. The modified equation for C_f'' is

$$C_f'' = \frac{2\kappa^2}{\left[\ln\left\{\left(\frac{\nu^*}{\kappa\nu}\right)(1 + \chi(\partial_x p))\right\} + B\kappa\right]^2} \tag{46}$$

where the function $\chi(\partial_x p)$ is negative in an adverse pressure gradient and zero otherwise. The C_f'' effectively determines a "slip" at the boundary. This "slip" boundary condition can be accomplished by a proper ratio between bounce-back and reflection of the distribution function colliding with the wall.

3. POWERFLOW

PowerFLOW is a software package based on lattice gas cellular automata.(13–16). PowerFLOW uses three speeds in three-dimensional space.(53) The equilibrium distribution function in the software is of Maxwellian type:

$$f_{\alpha}^{(\mathrm{eq})} = \rho w_{\alpha}(T)\exp\left(\frac{1}{2T}(2\mathbf{e}_{\alpha}\cdot\mathbf{u}-\mathbf{u}^{2})\right) \tag{47}$$

where $w_{\alpha}(T)$ can be determined by the conservation constraints.(37) In particular, for a three-speed model:(53)

$$w_{\alpha}(T) = \begin{cases} \dfrac{1}{d_0}(3T^2-3T+1), & |\mathbf{e}_{\alpha}| = 0, \\ \dfrac{1}{24}2T(2-3T), & |\mathbf{e}_{\alpha}| = 1, \\ \dfrac{1}{24}T(3T-1), & |\mathbf{e}_{\alpha}| = \sqrt{2} \end{cases} \tag{48}$$

where d_0 is an adjustable positive constant and is usually set to 6. The distribution function in PowerFLOW is represented by an eight-bit integer.(53) Some special care must be taken to ensure exact preservation of the conservation laws.

PowerFLOW uses a structured, cubic Cartesian mesh (voxels), and a grid refinement scheme which refines the grid size in each spatial dimension equally and successively by a factor of 2. The boundary surface is tessellated into triangular planes (surfels), of which each side has a length that is less than 1/10 the linear dimension of a voxel in the region.

Various boundary conditions are implemented in PowerFLOW by specifying a velocity profile, static pressure, and mass flux at the boundary.(16) At the boundary, momentum fluxes are computed and redistributed to enforce desirable boundary conditions.(15) Because PowerFLOW is a commercial software package, and source code was unavailable to the authors, specific implementation details discussed in Sections 2 and 3 may not be completely relevant to the release of the PowerFLOW software used in this work.

4. CFL3D

A standard computer code, CFL3D,(19–24) was used to compute flow fields for comparison with PowerFLOW results. The computer code CFL3D was developed at NASA Langley Research Center to solve the three-dimensional, time-dependent, thin-layer (in each coordinate direction) Reynolds-averaged Navier–Stokes (RANS) equations using a finite-volume formulation. The code uses upwind-biased spatial differencing for the inviscid terms and flux-limiting to obtain smooth solutions in the vicinity of shock waves. The viscous derivatives are computed by second-order central differencing. Fluxes at the cell faces are calculated by Roe's flux-difference-splitting method.(54) An implicit three-factor approximate factorization method is used to advance the solution in time. Here, only steady solutions have been computed and convergence is accelerated with local time-stepping, grid sequencing, and multigridding.

4.1. Turbulence Modeling

The CFL3D flow solver provides a wide range of turbulence models from which to choose. In this work, most of the cases employed the model of Spalart and Allmaras.(55) One case used Wilcox's k–ω model.(56)

In the model of Spalart and Allmaras(55) the turbulent eddy viscosity is obtained directly from the solution of a transport equation. The model is tuned for aerodynamic flows, is robust, and is applicable to flows with complex geometry. Several studies for both steady and unsteady flows(57, 22) show the model to have better predictive capability than algebraic and other one-equation models for a wide range of aerodynamic flows. The CFL3D implementation uses Version Ia of the model.(58)

The Wilcox k–ω model(56) is a two-equation model, one for the turbulent kinetic energy k, the other for ω, which is proportional to the turbulent energy dissipation rate per unit kinetic energy in the CFL3D implementation.(24) The resultant turbulent eddy viscosity is the ratio k/ω.

Although not required by CFL3D, in cases in which a turbulence model is used, the flow is considered fully turbulent from the onset, i.e., no laminar region is assumed. In addition, both the Spalart–Allmaras model and the Wilcox model are integrated all the way to the wall. As with all turbulence models in CFL3D, the solution of either turbulence model is decoupled from the solution of the flow equations. Information from the flow equations that is required in the turbulence equation lags one iteration. This is not a problem for sufficiently converged solutions. Additional implementation details are available in the CFL3D manual.(24)

4.2. Boundary Conditions

As shown in Fig. 5, a C-grid was used for the CFL3D calculations. This grid requires boundary conditions to be defined on three distinct geometrical surfaces: (1) the wing body, (2) the downstream plane, and (3) the external surface that defines the upstream, and upper and lower portions of the computational domain.

For the inviscid calculations with CFL3D, slip-wall boundary conditions (CFL3D boundary-condition type 1005) are used on the wing body (surface 1). These conditions require that the component of velocity normal to the wall be equal to zero. In CFL3D this boundary condition approximates the cell-face boundary value for the density as the density of the nearest cell-center point on the grid. The boundary values for the pressure are obtained similarly. For the inviscid cases, the external computational surfaces (surfaces 2 and 3) are modeled with the use of local one-dimensional Riemann invariants (CFL3D boundary condition type 1003). The conditions outside the computational domain are taken from the prescribed free-stream Mach number and flow direction.

For the viscous calculations, no-slip, no-penetration, adiabatic wall conditions (CFL3D boundary-condition type 2004 with $Twtype = 0$ and $Cq = 0$) were used on the wing body (surface 1). In the viscous cases, the condition imposed at the downstream plane (surface 2) involved a zeroth-order extrapolation from the computational interior (CFL3D boundary-condition type 1002). The far-field, upstream computational surface (surface 3) had its boundary condition modeled with the local one-dimensional Riemann invariant, as with the inviscid cases.

More details regarding the implementation of the boundary conditions can be found in the CFL3D manual.(24)

5. TWO-DIMENSIONAL SIMULATION OF FLOW PAST NACA-0012 AIRFOIL

5.1. NACA-0012 Airfoil

The NACA-0012 airfoil is a popular wing section that is used for many purposes. Coordinates and a generation formula are given in ref. 18. Figure 4 shows a NACA-0012 wing section. The NACA-0012 airfoil has zero camber and a maximum thickness to chord ratio of 12 percent. A 12 percent thickness ratio results in maximum lift for many 4 and 5 digit NACA wing sections.(18) Experimental lift versus angle-of-attack curves, reproduced in graphical form in ref. 18, suggest that on aerodynamically smooth wings, the NACA-0012 airfoil does not experience massively separated flow up to an angle of attack of 16° for Reynolds numbers of 3, 6, and 9 million. For the wing with a standard roughness, massive flow separation occurs at an angle of attack of about 12° at a Reynolds number of 6 million. Because the NACA-0012 airfoil is symmetric, it generates no

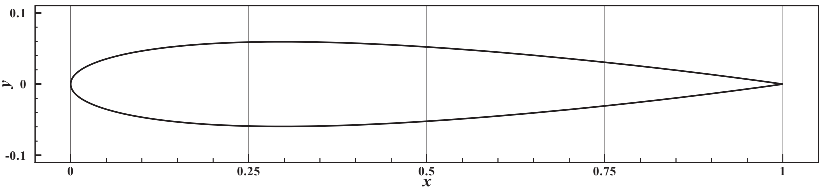

Fig. 4. NACA-0012 airfoil.

lift at zero angle of attack. At an angle of attack of 3°, the lift coefficient is expected to be about 0.35. At an angle of attack of 7°, the lift coefficient is expected to be between about 0.74 and 0.78, depending upon conditions.

5.2. Computational Domain and Boundary Conditions

A series of 2-D simulations of flow past the NACA-0012 airfoil at low speed were performed by using PowerFLOW and the results are compared with those from CFL3D. The Mach number M is set to 0.2 in the simulations. Several values of angle of attack, α, are selected in the simulations: 0, 3, 7, and 12°.

Both CFL3D and PowerFLOW are three-dimensional codes. However, the simulations of the flow past the airfoil are two-dimensional. In the case of CFL3D, this is simply done by using the two-dimensional solver option in the code. In the case of PowerFLOW, the two-dimensional simulations were accomplished by shrinking the third dimension to two cells and employing a periodic boundary condition.[59] The simulations were done on an SGI ORIGIN2000 with 16 R10000 processors. The CPU times and resolutions used in the simulations of both CFL3D and PowerFLOW are summarized in Table II.

For PowerFLOW, a typical computational domain is $L_x \times L_y = 9 \times 8$ in units of chord length. PowerFLOW uses Cartesian grids. The resolution at the surface is approximately 500 points per chord. In the PowerFLOW simulations there are seven variable resolution regions, i.e., the grid spacing is varied from 2^0 to 2^{-6}. The resolution used in PowerFLOW is usually 383,724 voxels (grid points) and 1,275 surfels (surface elements). The aspect ratio of grid spacings is always 1:1 in the PowerFLOW simulations.

The boundary conditions used in PowerFLOW are: a free-slip boundary condition for upper and lower boundaries; a driving boundary condition with a fixed velocity at the inlet boundary, and a fixed static pressure for exit boundary condition. At the low Reynolds number (Re = 500), a no-slip boundary condition is used for the airfoil surface. At high Reynolds

Table II. Summary of Results for Simulations for Flow Past a NACA-0012 Airfoil. (Top) Drag Coefficient C_d and Lift Coefficient C_l Obtained from CFL3D and PowerFLOW. The Numbers in Parentheses Are Bounds Estimated from the Unsteady Results of the PowerFLOW Simulations. (Bottom) Resolutions and CPU Times of Simulations Using CFL3D and PowerFLOW. CPU Times Are Given in Seconds and (Hours)

			CFL3D		PowerFLOW	
	α	Re	C_d	C_l	C_d	C_l
1a	0°	500	0.17618	$0.115 \cdot 10^{-6}$	0.171721	$0.22703 \cdot 10^{-3}$
1b	0°	500	0.1741	$-0.538 \cdot 10^{-5}$	0.180716	$-0.211 \cdot 10^{-3}$
2a	0°	$0.5 \cdot 10^6$	0.016268	$0.1769 \cdot 10^{-5}$	0.0212145	$-0.2 \cdot 10^{-3}$
2b	0°	$0.5 \cdot 10^6$	0.012174	$0.9241 \cdot 10^{-9}$	0.0201036	$-0.3518 \cdot 10^{-3}$
2c	0°	$0.5 \cdot 10^6$	0.012815	$-0.616 \cdot 10^{-7}$	—	—
3	3°	$0.5 \cdot 10^6$	0.012986	0.32372	0.0132659	0.188275
4a	3°	inviscid	0.001256	0.3608	—	—
4b	3°	$0.5 \cdot 10^9$	—	—	−0.014939	0.351683
4c	3°	$0.5 \cdot 10^9$	—	—	0.00304585	0.0512701
5	7°	$0.5 \cdot 10^6$	0.015739	0.7449	(−0.017, 0.0019)	(0.72, 1.2)
6	12°	$0.5 \cdot 10^6$	0.027468	1.1809	(−0.054, −0.12)	(0.7, 1.56)

			CFL3D		PowerFLOW	
	α	Re	$N_\theta \times N_r$	CPU time	Voxel × Surfel	CPU time
1a	0°	500	257 × 65	3500 (1.0)	159060 × 828	36795 (10.2)
1b	0°	500	373 × 141	23143 (6.4)	418800 × 1275	70900 (19.7)
2a	0°	$0.5 \cdot 10^6$	257 × 65	1650 (0.5)	145748 × 828	23003 (6.4)
2b	0°	$0.5 \cdot 10^6$	373 × 141	17671 (4.9)	383724 × 1275	63034 (17.5)
2c	0°	$0.5 \cdot 10^6$	373 × 141	17662 (4.9)	—	—
3	3°	$0.5 \cdot 10^6$	373 × 141	29962 (8.3)	383724 × 1275	60772 (16.9)
4a	3°	inviscid	225 × 65	1630 (0.5)	—	—
4b	3°	$0.5 \cdot 10^9$	—	—	383724 × 1275	66940 (18.6)
4c	3°	$0.5 \cdot 10^9$	—	—	63612 × 604	112528 (31.3)
5	7°	$0.5 \cdot 10^6$	373 × 141	29962 (8.3)	383724 × 1275	61627 (17.1)
6	12°	$0.5 \cdot 10^6$	373 × 141	22680 (6.3)	383724 × 1275	67320 (18.7)

numbers ($\mathrm{Re} \geqslant 0.5 \times 10^6$), the "turbulent wall boundary condition" using a wall function is applied to the airfoil surface.

PowerFLOW is a time explicit scheme with a small time step: the Courant-Friedrichs-Lewy number is one with respect to the grid. PowerFLOW has significant fluctuations in its instantaneous results because of its integer nature. Therefore hydrodynamic outputs of PowerFLOW are usually averaged over a number of time steps. The interval for the time-average for the NACA-0012 airfoil simulations is 1,000 steps unless otherwise specified.

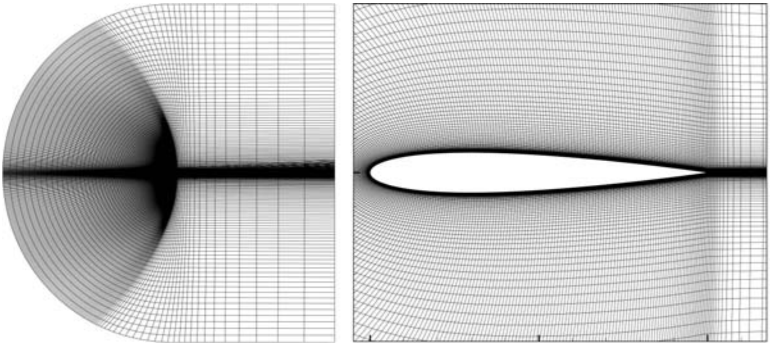

Fig. 5. A typical body-fitted mesh used in CFL3D. (left) the entire computational domain with grid resolution 373 × 141, and (right) the mesh around the airfoil, which is the center part of the domain.

Figure 5 shows the body-fitted computational mesh (C-grid) used in the CFL3D simulations. With the grid nondimensionalized with the chord length, the computational domain consists of a half disc of radius 20 in front of the airfoil and a rectangle of 20 × 40, as shown in Fig. 5. The resolution is typically 373 × 141, or 52,593 cells. The grid spacing normal, y_1, and tangential, x_1, to the surface near the middle of chord are approximately 0.005 and 0.015 for inviscid cases, respectively, and 0.00012 and 0.008 for viscous cases, respectively. With these fine grids, CFL3D can resolve the boundary layer in the simulations.

Although all flows possess some unsteadiness, for attached flows the fluctuations typically are on a time scale that is so small that they are usually of no practical interest for aerodynamic quantities like lift and drag. Most experimental force data are actually time-averaged quantities. Some sensors implicitly average because they are incapable of resolving the small, unsteady fluctuations. To obtain mean data from a time-accurate CFD simulation requires performing a similar time-average on the numerical results. This is the approach used in PowerFLOW. Although only fractions of a second are needed to obtain a significant sample in an experiment, considerable computer resources are required to generate all of the required samples from a numerical simulation. To simplify and accelerate the solution procedure, CFD schemes commonly solve equations that have been time-averaged so that all of the dependent variables are actually time-mean quantities. In the case of the Navier–Stokes equations applied to turbulent flows, this process is called Reynolds averaging and produces several additional terms which must be modeled to close the system of equations. This modeling of the effects of the small scale, unsteady turbulent flow on the mean quantities is nontrivial, but many models have been developed and validated for many important flows. Because the equations are nonlinear, the typical solution procedure is to iterate until the solution becomes steady. Many acceleration techniques have been developed that take advantage of the fact that the intermediate solution need not be time accurate. This is the approach used in CFL3D.

5.3. Simulation Results

The local results discussed below include u and v, the x and y components of the velocity field normalized by the inlet velocity U_∞; and the pressure coefficient

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{49}$$

where p is the local pressure, p_∞ is the far-field pressure, and ρ_∞ is the far-field density. The velocity and pressure are measured at five stations: $x/L = 0$, $1/4$, $1/2$, $3/4$, and 1, where L is the chord length. In addition, the lift coefficient

$$C_l = \frac{F_l/A}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{50}$$

and the drag coefficient

$$C_d = \frac{F_d/A}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{51}$$

are obtained by integrating the lift force F_l and the drag force F_d per unit span over the surface of the airfoil. (A is the platform area, which in two dimensions is the chord length L.)

The resultant lift coefficients C_l and drag coefficients C_d are summarized in Table II. The Reynolds numbers Re in these simulations range from 500 to 0.5×10^9. For $\mathrm{Re} = 500$ (Case 1a and 1b in Table II), the results were obtained by directly simulating the laminar flow. For $\mathrm{Re} \geqslant 0.5 \times 10^6$, turbulence modeling was used in both PowerFLOW and CFL3D (Cases 2–6 in Table II). PowerFLOW uses wall functions, described in Sec. 2.4, to model the boundary layer. CFL3D incorporates various turbulence models in the code. Two turbulence models were used in the simulations of flow past a NACA-0012 airfoil: the Spalart-Allmaras model and the k–ω model. The k–ω model was only used in Case 2c of Table II to demonstrate the level of variability in the CFL3D result with different turbulence models. The results show little dependence on the turbulence model for this flow. The remainder of CFL3D simulations used the Spalart-Allmaras model.

Case 1. $\mathrm{Re} = 500$, $\alpha = 0$. The CFL3D simulations used two resolutions: 257×65 and 373×141, and PowerFLOW used 159, 060 voxels and 828 surfels (Case 1a), and 418, 800 voxels and 1275 surfels (Case 1b). The value of drag coefficient C_d obtained from PowerFLOW is about 3.8% larger than the value obtained from CFL3D. Although a symmetric airfoil at zero angle of attack should generate zero lift, the PowerFLOW lift coefficient varied between ± 0.0002. The CFL3D solutions maintained zero lift to an accuracy almost two orders of magnitude better.

At a Reynolds number $\mathrm{Re} = 500$, the flow is laminar and the results were obtained by direct numerical simulation—no turbulence modeling was used. Along the surface of airfoil, the no-slip boundary condition is used in PowerFLOW. After 25,000 time steps, a steady-state solution was attained. Figure 6 shows the time history of C_d and C_l. The variations of C_d and C_l are on the order of 10^{-4}.

Figures 7, 8, and 9 show $u(x, y)$, $v(x, y)$, and $C_p(x, y)$ at five vertical sections of $x/L = 0$, $1/4$, $1/2$, $3/4$, and 1. These figures show the results with the higher resolution case (Case 1b). The differences between the results of PowerFLOW and CFL3D are already visible at $\mathrm{Re} = 500$. The difference in the x-component of velocity u is relatively small. However, the

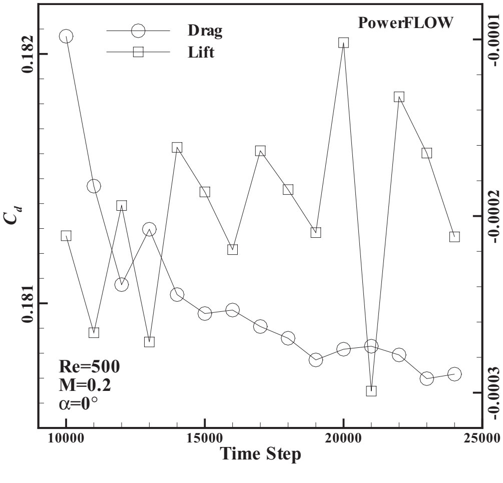

Fig. 6. Time history of C_d and C_l in the PowerFLOW simulation. $\mathrm{Re} = 500$, $\mathrm{M} = 0.2$, and $\alpha = 0°$.

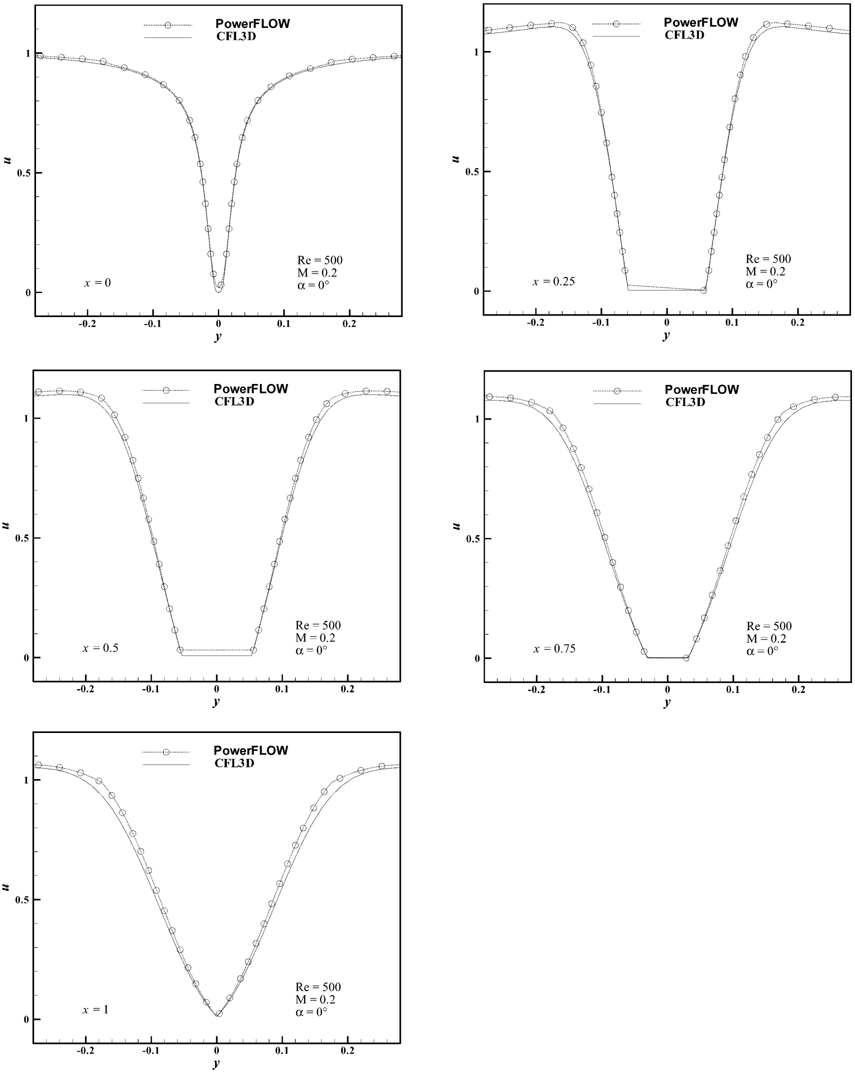

Fig. 7. NACA-0012 airfoil with angle of attack $\alpha = 0°$, $\mathrm{Re} = 500$, and $\mathrm{M} = 0.2$. The x-component of velocity $u(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 418, 800 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

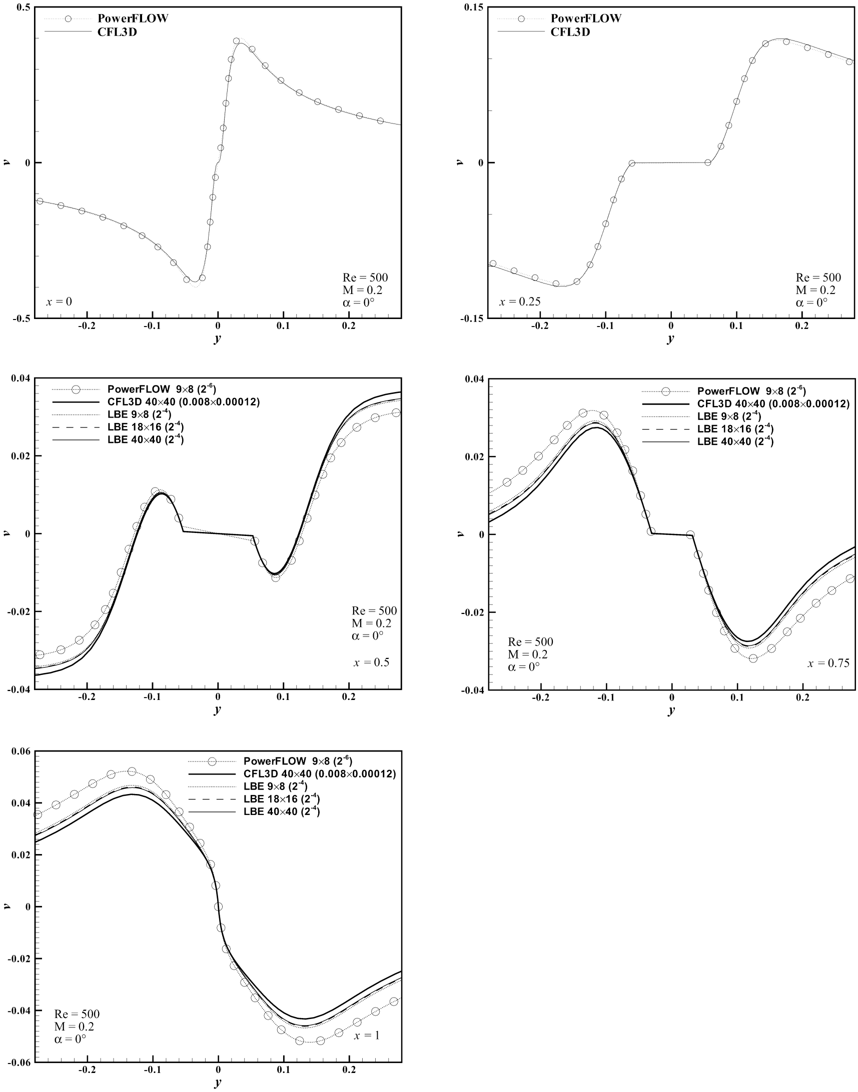

Fig. 8. NACA-0012 airfoil with angle of attack $\alpha = 0°$, $\mathrm{Re} = 500$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 418, 800 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D. The results of the LBE simulations with a resolution of 320 points on the airfoil surface and three different domain sizes are also shown at $x/L = 0.5$, 0.75, and 1.0.

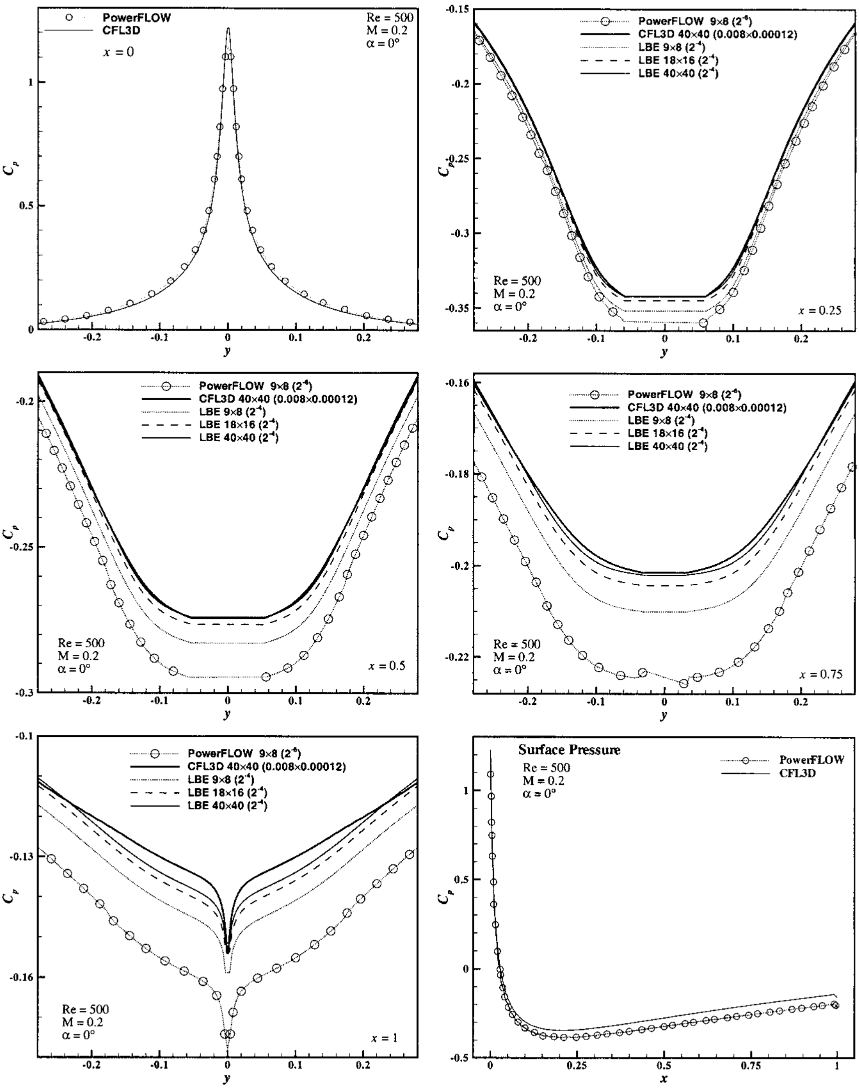

Fig. 9. NACA-0012 airfoil with angle of attack $\alpha = 0°$, $\mathrm{Re} = 500$, and $\mathrm{M} = 0.2$. The pressure coefficient $C_p(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length, and along the surface. The resolution is 418, 800 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D. The results of the LBE simulations with a resolution of 320 points on the airfoil surface and three different domain sizes are also shown at $x/L = 0.25$, 0.5, 0.75 and 1.0.

relative difference in the y-component of velocity v is significantly greater, and the difference is amplified downstream. Similar discrepancies exist in the vertical pressure distribution and the surface pressure distribution. Although the no-slip condition is satisfied in the CFL3D simulations, the figures do not show $u \to 0$ near the airfoil surface. This is because the data were extracted at uniformly spaced points in the y direction, and a point on the airfoil surface is usually not included. This is also true for the PowerFLOW data.

To identify the sources which cause the discrepancy between the results obtained by using CFL3D and PowerFLOW, we conducted a series of simulations by using a lattice Boltzmann equation with nine velocities in two-dimensions (D2Q9 model).[25, 26] We are particularly concerned with two issues related to boundary conditions: the noslip boundary conditions at the wing body and the far field boundary conditions. We performed a convergence study of the simulation on the computational domain size and resolution along the wing surface. We used a nested-block binary grid refinement, similar to that used in PowerFLOW in terms of grid structure, i.e., the grid size at nth level refinement is $2^{1-n}\delta_x$, where δ_x $(=1)$ is the coarsest grid size. We used the bounce-back scheme with second-order interpolations for the noslip boundary conditions at the wing body.[60] At the lower and upper boundaries, free slip boundary conditions are applied. At upstream, a constant pressure and a constant velocity (of $\mathrm{M} = 0.2$) are enforced. And at the downstream, diffusive boundary conditions are used. The computations were performed on a Pentium III 650MHz PC. The results are summarized in Table III.

We first performed simulations of the flow with a computational domain size of 9×8 in units of chord length L, and with three different numbers of nested binary grid refinement blocks around the wing body,

Table III. Simulations of Flow Past a NACA-0012 Airfoil at Re = 500, M = 0.2, and $\alpha = 0°$. The LBE D2Q9 Model Is Used in the Calculations. The LBE Results of the Drag Coefficient C_d Agree Well with the CFL3D Results Given in Table II

LBE D2Q9 Model, Re = 500, M = 0.2, $\alpha = 0°$						
C_d	C_l	2^{-n}	Grids	$L_x \times L_y$	Time steps	CPU time
0.1758	$8.5 \cdot 10^{-16}$	2^{-2}	39,074	9×8	14,000	1,319.0 (0.37)
0.1763	$3.7 \cdot 10^{-16}$	2^{-3}	55,362	9×8	14,000	4,611.9 (1.28)
0.1759	$2.5 \cdot 10^{-15}$	2^{-4}	105,570	9×8	10,000	17,938.7 (4.98)
0.1760	$-2.4 \cdot 10^{-15}$	2^{-5}	241,218	9×8	10,000	93,005.2 (25.9)
0.1742	$1.5 \cdot 10^{-15}$	2^{-4}	189,411	18×16	10,000	19,404.2 (4.98)
0.1734	$4.2 \cdot 10^{-16}$	2^{-4}	715,611	40×40	10,000	28,338.8 (7.87)

$n = 4$, 5, and 6. These three refinement levels correspond to the resolutions of 160, 320, and 640 points on the airfoil surface, respectively. These resolutions are coarser than the typical resolutions of the corresponding PowerFLOW calculations. The chord length L is equal to $20\delta_x$, where δ_x is the coarsest grid size ($\delta_x = 1$). Apparently the solutions with these three simulations with different grid refinement levels have no visible differences, in flow fields of v and C_p where the CFL3D solution and the PowerFLOW solution display their greatest discrepancies, as shown in parts of Figs. 8 and 9, respectively. The variation in the drag coefficient due to the different grid refinement levels is very little, as shown in Table III. This indicates that even the coarsest resolution ($n = 4$, 160 points on the wing surface) is sufficient in this calculation. A coarser resolution of 80 points on the wing surface ($n = 3$) was also tested but it is found to be too coarse. The solutions obtained by using the LBE D2Q9 model is between that of CFL3D and PowerFLOW when the computational domain size is 9×8. (Note that the domain size of the PowerFLOW calculations is 9×8 and that of the CFL3D calculations is 40×40.)

We further performed simulations with different computational domain sizes, while maintaining the grid refinement of 5 levels (320 points on the airfoil surface). As the computational domain size becomes larger and larger, from 9×8 (in the units of chord length), to 18×16, and then to 40×40, the discrepancy between the LBE and the CFL3D solutions becomes smaller and smaller, as shown most clearly in Fig. 9 for C_p at $x = 1$.

Even with equal domain sizes, there still remains significant discrepancy between the solutions of CFL3D and the LBE D2Q9 model. We suspect that the discrepancy could be due to the different far field boundary conditions: while CFL3D uses a sophisticated far field boundary conditions that minimize the wave coming in from the far field boundaries, the LBE method usually uses simple boundary conditions, such as periodic or free-slip boundary conditions. Implementation of non-reflective far-field boundary conditions in the LBE method is an interesting subject for future study.

Case 2. $\mathrm{Re} = 0.5 \times 10^6$, $\alpha = 0°$. When $\mathrm{Re} = 0.5 \times 10^6$, turbulence modeling was used. CFL3D used the Spalart-Allmaras model (Case 2a and 2b in Table II) and k-ω model (Case 2c). PowerFLOW used the wall function discussed in Section 2.4 (Case 2a and 2b). The value of drag C_d obtained from PowerFLOW is about 66% larger than the value obtained from CFL3D. The value of C_l from the PowerFLOW simulation is 2 to 6 orders of magnitude larger than that of CFL3D. As before, for $\alpha = 0°$, the value of C_l should be 0.

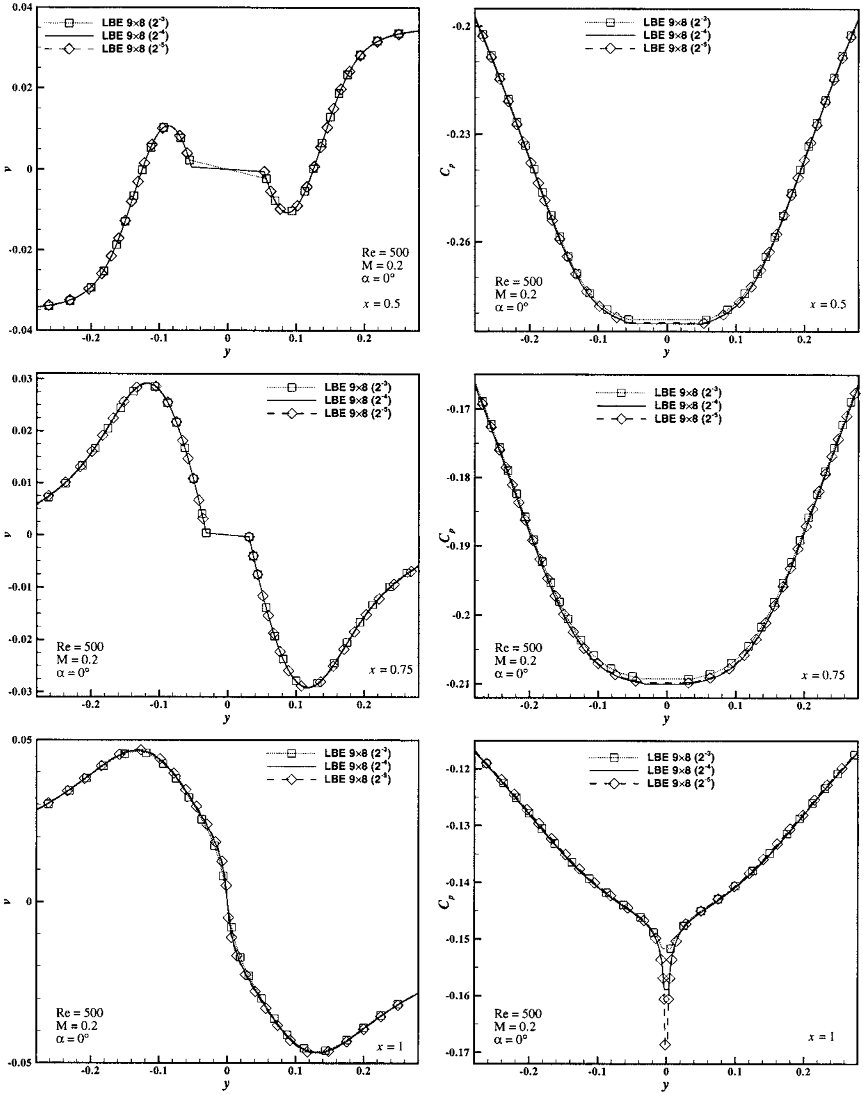

Fig. 10. NACA-0012 airfoil with angle of attack $\alpha = 0°$, $\mathrm{Re} = 500$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ (left) and the pressure coefficient $C_p(x, y)$ (right) at $x/L = 0.5$, 0.75, and 1.0, obtained by using the nine-velocity LBE model (D2Q9). Three different resolutions were used to demonstrate the convergence of the results. Squares (□), solid lines (—), and diamonds (◇) correspond to 160, 320, and 640 points on the airfoil surface, respectively.

Figures 11, 12, and 13 show $u(x, y)$, $v(x, y)$, and $C_p(x, y)$ at the five vertical sections, for the results with the higher resolution (Case 2b). The differences between the CFL3D results and PowerFLOW results are quite significant on the surface of the airfoil, and relatively negligible in the flow region away from the surface. Obviously, the differences in the flow fields near the boundary result in the significant difference in the value of drag C_d. This raises the question of adequacy of turbulence modeling by a wall function in PowerFLOW, even for cases in which the flow is attached.

Case 3. $\mathrm{Re} = 0.5 \times 10^6$, $\alpha = 3°$. The results of the drag coefficient C_d obtained from the two methods are very close in this case: C_d from PowerFLOW is only about 2.1% larger than that from CFL3D. With $\alpha = 3°$, the lift coefficient is non-zero. The lift coefficient C_l obtained from PowerFLOW simulations is about 58% smaller than the result from CFL3D. Because drag includes both a frictional component and a component related to lift, the close agreement of the drag coefficient is likely to be fortuitous.

Figures 14, 15, and 16 show $u(x, y)$, $v(x, y)$, and $C_p(x, y)$ at the five vertical sections. The velocity fields obtained from PowerFLOW and CFL3D simulations significantly disagree near the airfoil surface, which in turn leads to the disagreement in the region away from the boundary. Furthermore, qualitative disagreement in pressure $C_p(x, y)$ starts to appear (e.g., $C_p(x, y)$ at $x = 3/4$ and 1 in Figs. 16). Figure 17 shows the streamlines of PowerFLOW and CFL3D simulations in the vicinity of the airfoil. The streamlines do not stay completely attached to the airfoil surface in the PowerFLOW simulations. Note that the PowerFLOW calculation shows separated flow near the trailing edge while the CFL3D solution remains completely attached. The separation shown in Fig. 17 could also be surmised from the surface pressure distribution near the trailing edge in Fig. 16.

Case 4. Inviscid flow, $\mathrm{Re} = \infty$, $\alpha = 3°$. In the CFL3D simulation of inviscid flow, an Euler scheme is used by neglecting the viscous flux terms. In the PowerFLOW simulation, the Reynolds number is 0.5×10^9, and the free slip boundary condition is applied at the surface of the airfoil[(16)] to mimic the inviscid effect. In this case, the value of C_d from one PowerFLOW simulation is negative, as indicated in Table II (Case 4b).

Figures 18, 19, and 20 show $u(x, y)$, $v(x, y)$, and $C_p(x, y)$ at the five vertical sections, for Case 4a (CFL3D) and 4b (PowerFLOW). The results of $u(x, y)$ from CFL3D and the PowerFLOW simulations are qualitatively different near the airfoil surface. The CFL3D results were compared against those from other CFD codes, and the agreement was excellent.

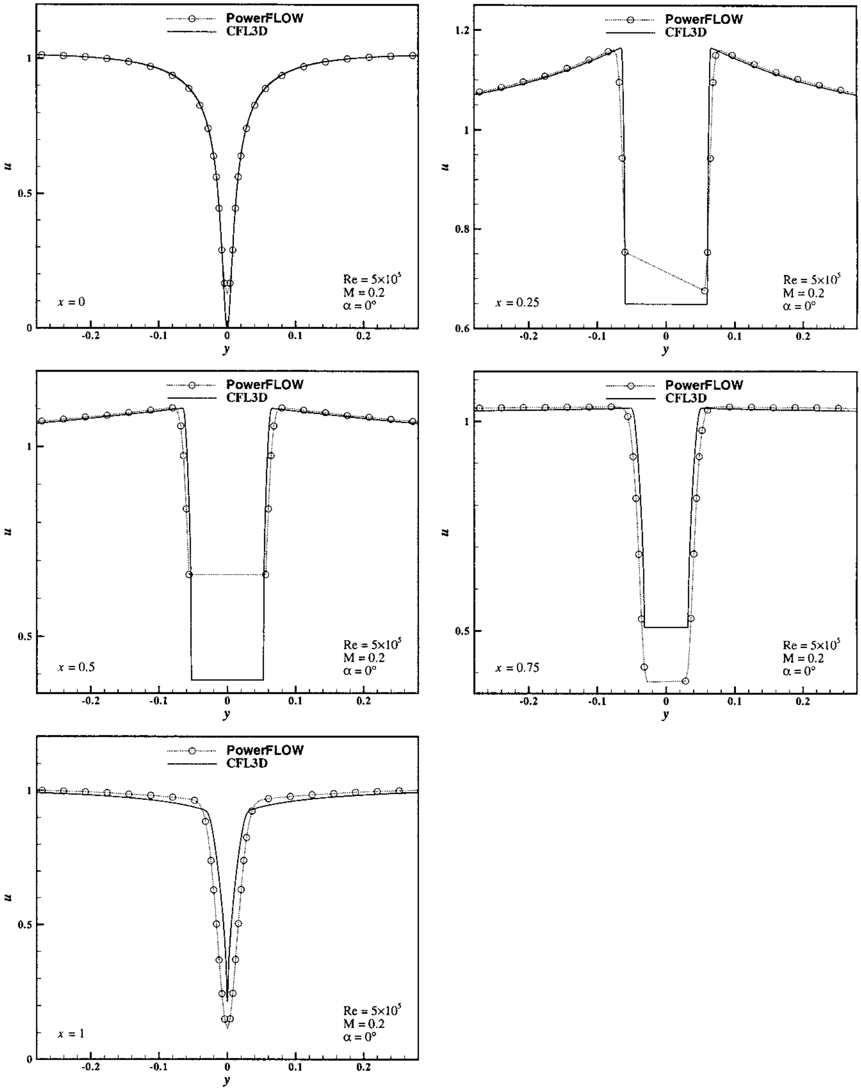

Fig. 11. NACA-0012 airfoil with angle of attack $\alpha = 0°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The x-component of velocity $u(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

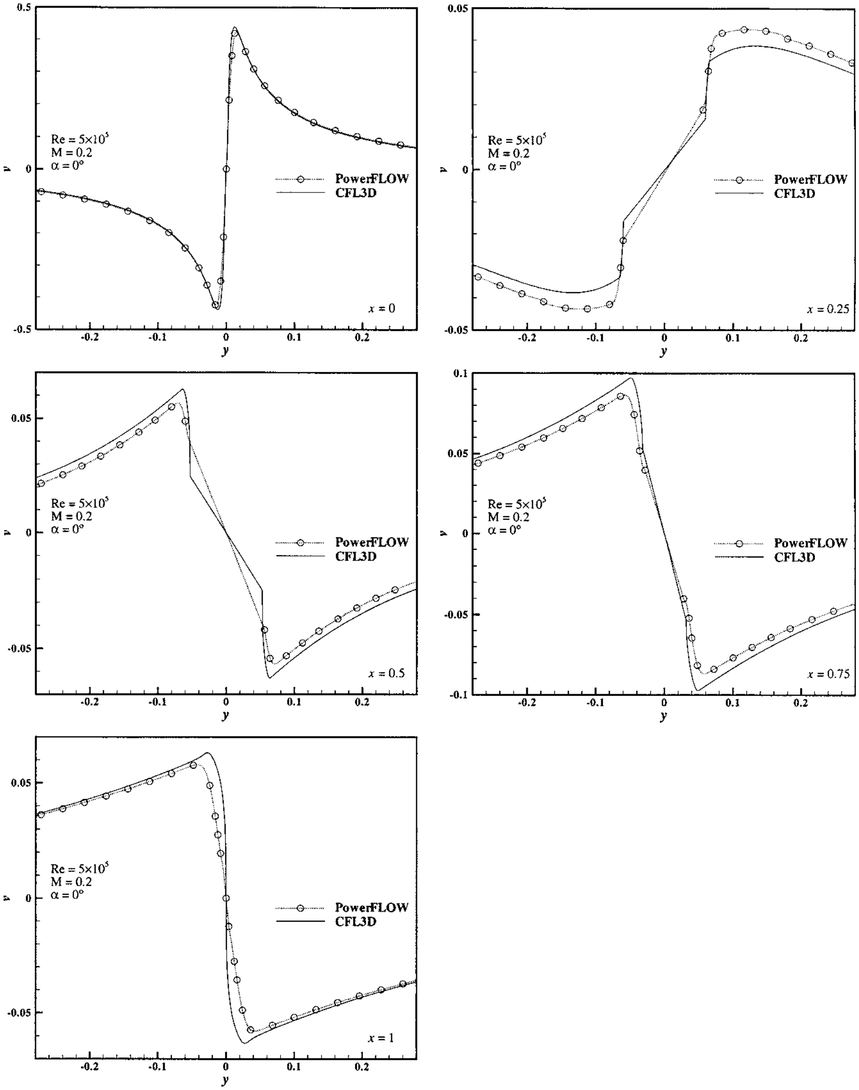

Fig. 12. NACA-0012 airfoil with angle of attack $\alpha = 0°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

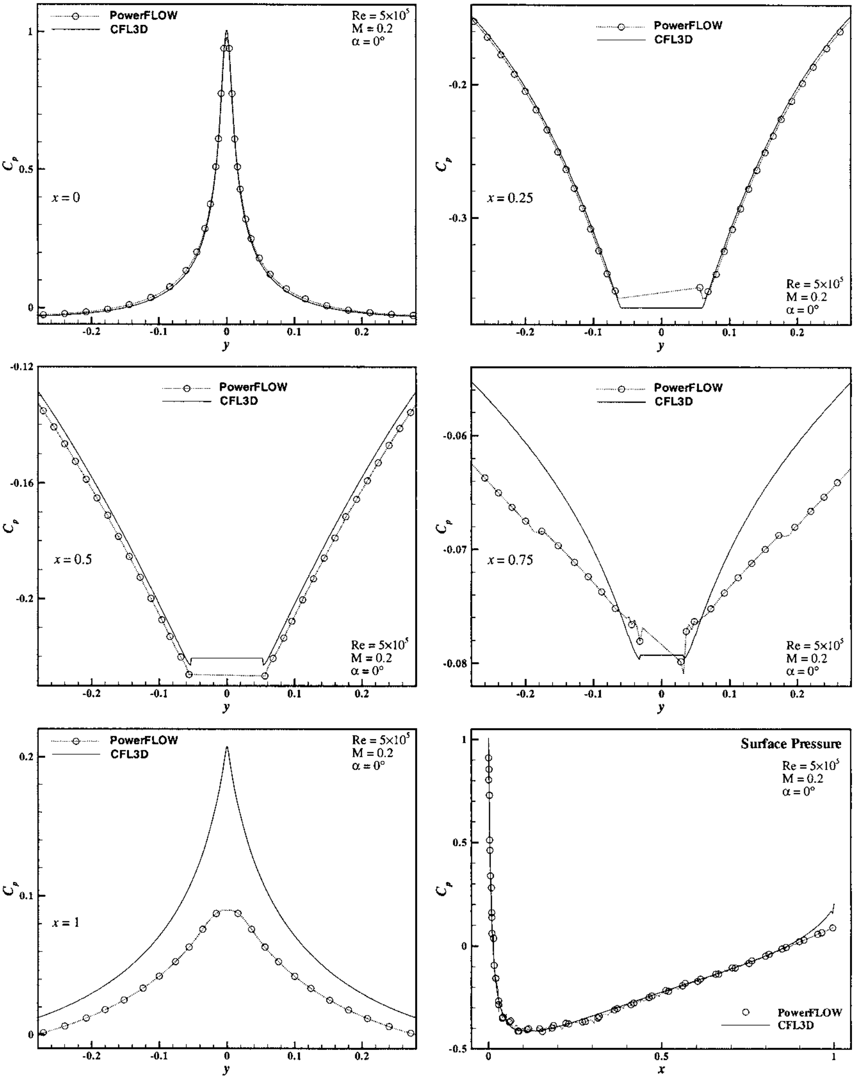

Fig. 13. NACA-0012 airfoil with angle of attack α = 0°, Re = 0.5 × 10^6, and M = 0.2. The pressure coefficient $C_p(x, y)$ at $x/L = 0.0, 0.25, 0.5, 0.75$, and 1.0, where L is the chord length, and along the surface. The resolution is 383, 724 voxels and 1, 275 surfels for POWERFLOW and 373 × 141 for CFL3D.

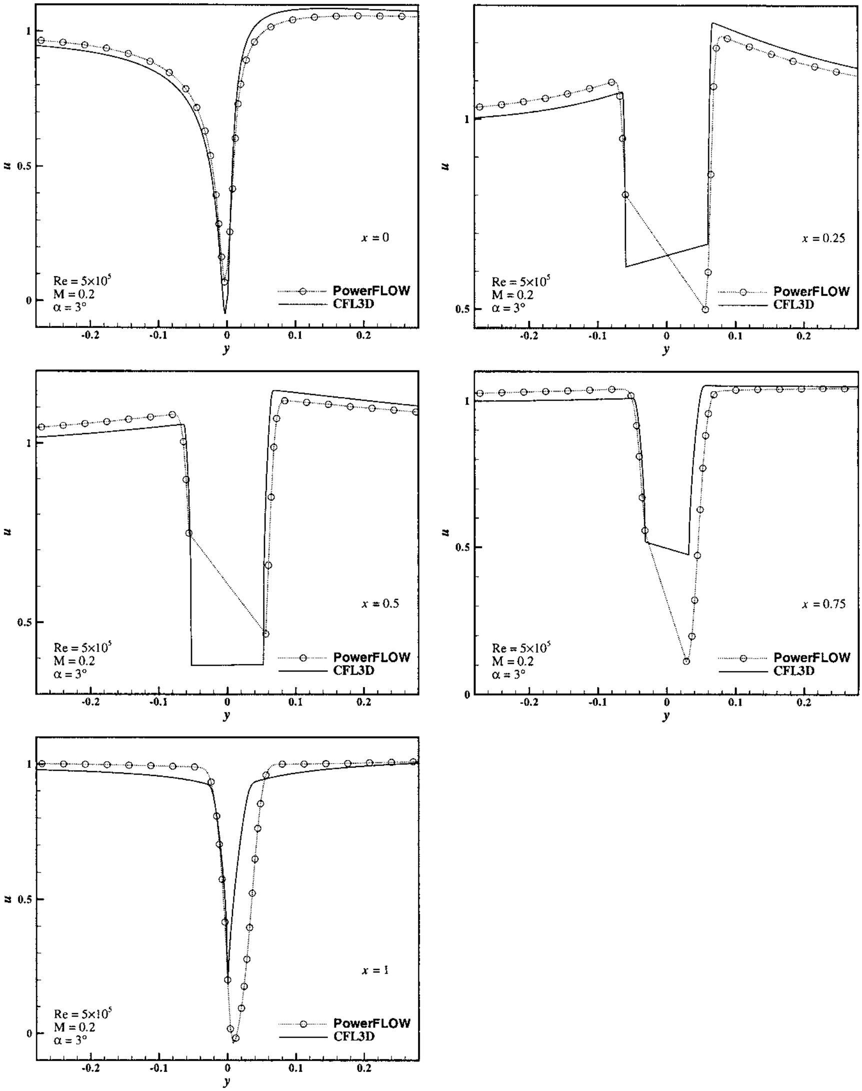

Fig. 14. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The x-component of velocity $u(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

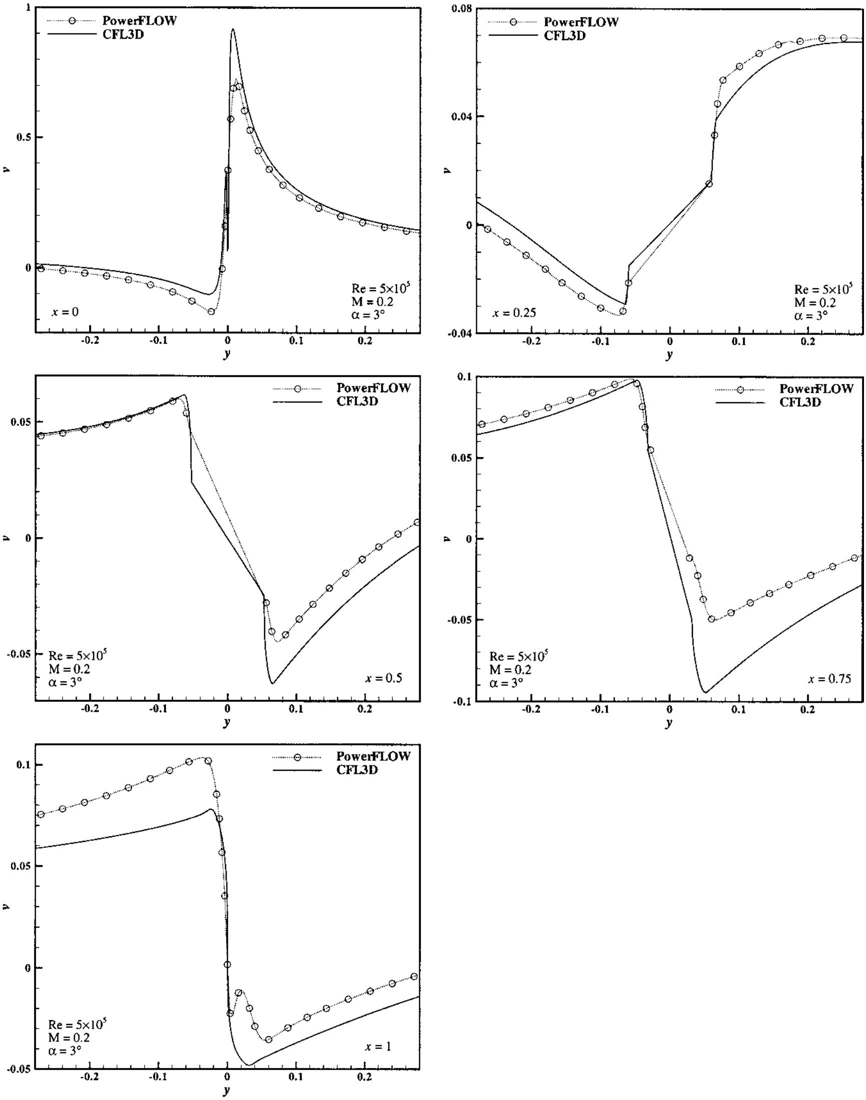

Fig. 15. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

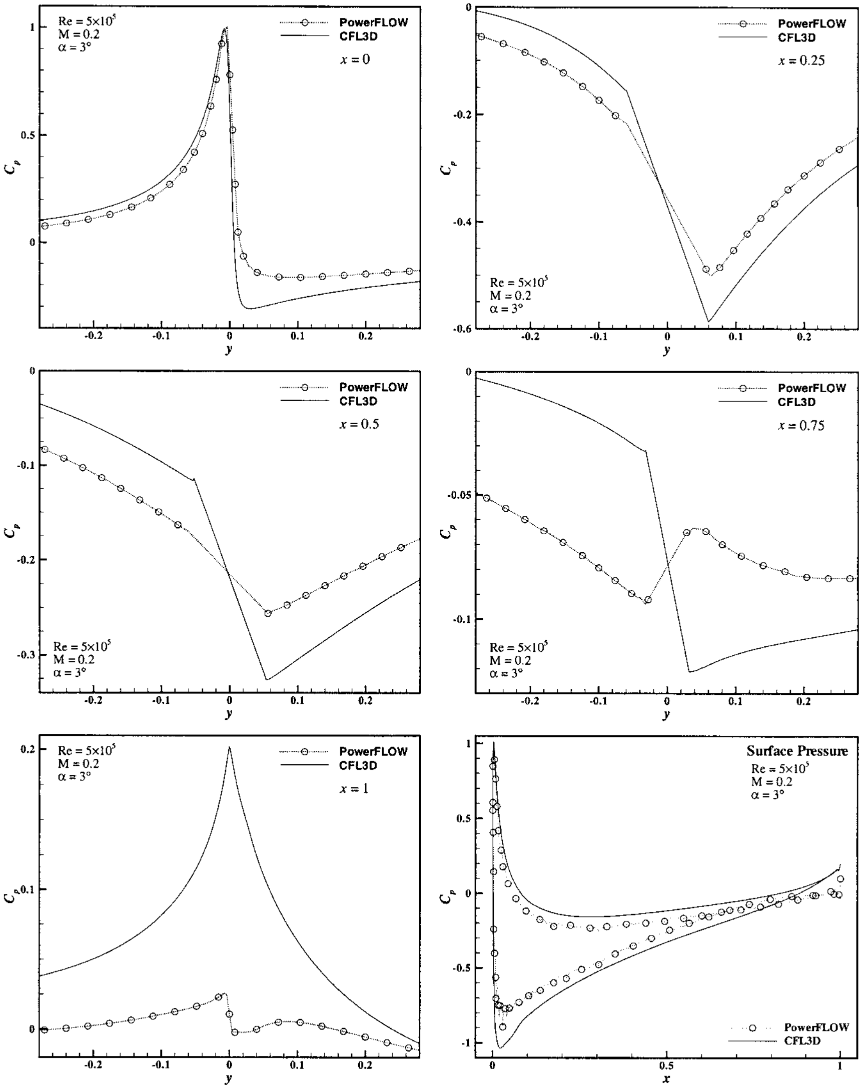

Fig. 16. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The pressure coefficient $C_p(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length, and along the surface. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

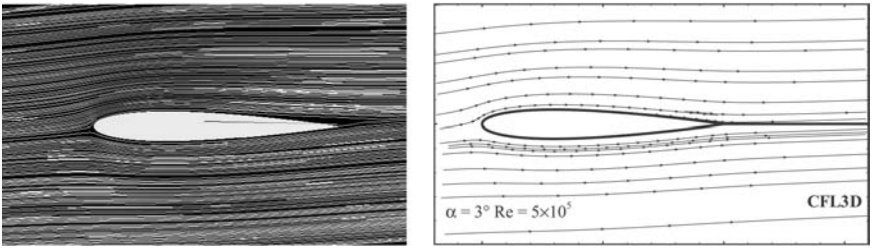

Fig. 17. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. Streamlines of PowerFLOW simulation (left) vs. CFL3D simulation (right).

Figure 21 shows the time history of C_d and C_l in PowerFLOW simulations with two different resolutions and computational domain sizes. In the first case (Case 4b), the resolution is 383,724 voxels and 1,275 surfels, the resolution on the airfoil surface is 500 points along the chord, the computational domain is 9×8, and there are seven variable resolution regions (2^0 to 2^{-6}). The drag coefficient C_d converges to a negative value after about 20,000 iterations, as shown in the left part of Fig. 21.

In the second case (Case 4c), the resolution is 63,612 voxels and 604 surfels, the resolution on the airfoil surface is 200 points along the chord, the computational domain is 21×20, and there are seven variable resolution regions (2^0 to 2^{-7}). This case was run to investigate whether the original computational domain was too small. With a coarser resolution but a larger computational domain, the results of C_d and C_l do not seem to converge after 60,000 iterations, as shown in the right part of Fig. 21. The values of C_d and C_l given in Table II (Case 4c) are the final values of the simulations. They are apparently far away from the correct values. It is also noted that the CPU time for the PowerFLOW simulation is almost doubled in this case, despite the fact that the resolution is much coarser.

Case 5. $\mathrm{Re} = 0.5 \times 10^6$, $\alpha = 7°$. With $\alpha = 7°$, the mean flow should be steady and attached. However, the PowerFLOW simulation displays massive flow separation, and flow field becomes highly unsteady. Figure 22 shows a snapshot of streamlines in both the PowerFLOW and the CFL3D simulation. The two results are completely different from each other. Because PowerFLOW failed to converge to a steady mean flow after 30,000 time steps, the values of C_l and C_d can only be given in their estimated bounds, as shown in Fig. 23.

Figures 24, 25, and 26 show $u(x, y)$, $v(x, y)$, and $C_p(x, y)$ at the five vertical sections. In this case the flow fields obtained from the two methods do not have any resemblance; while the solution from CFL3D remains

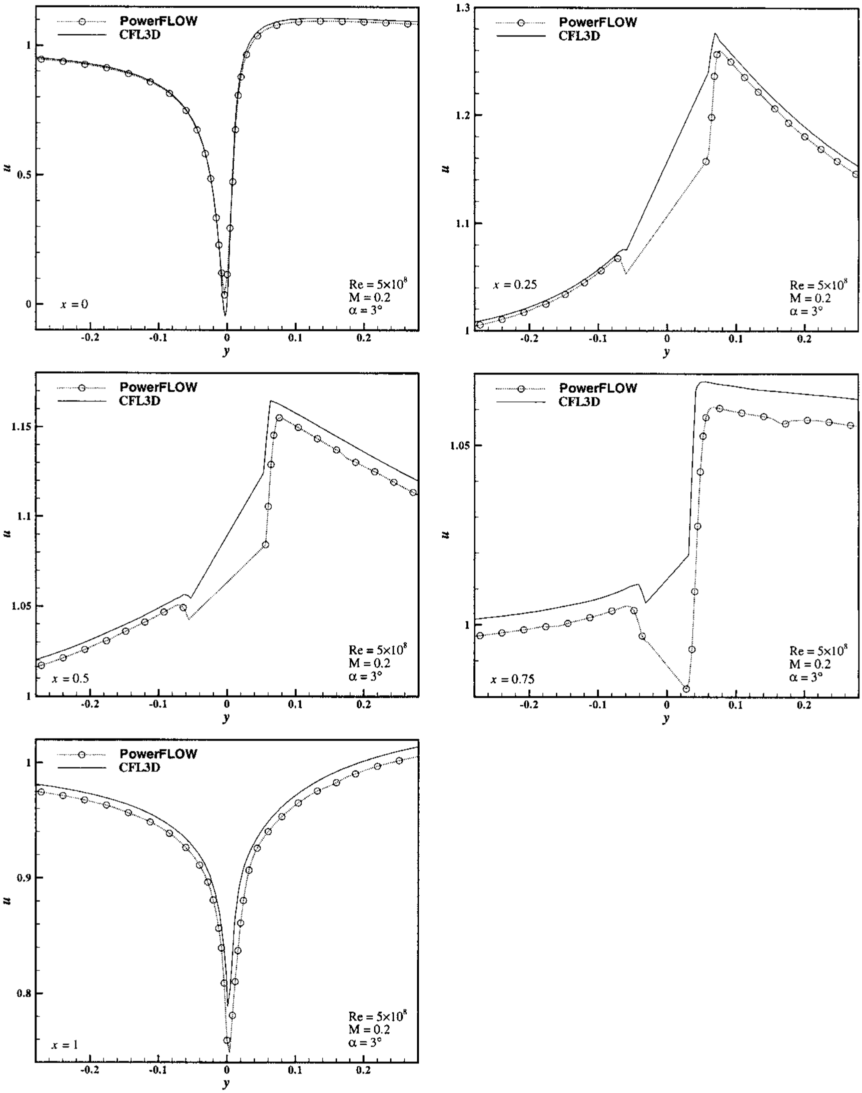

Fig. 18. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^9$, and $\mathrm{M} = 0.2$. The x-component of velocity $u(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

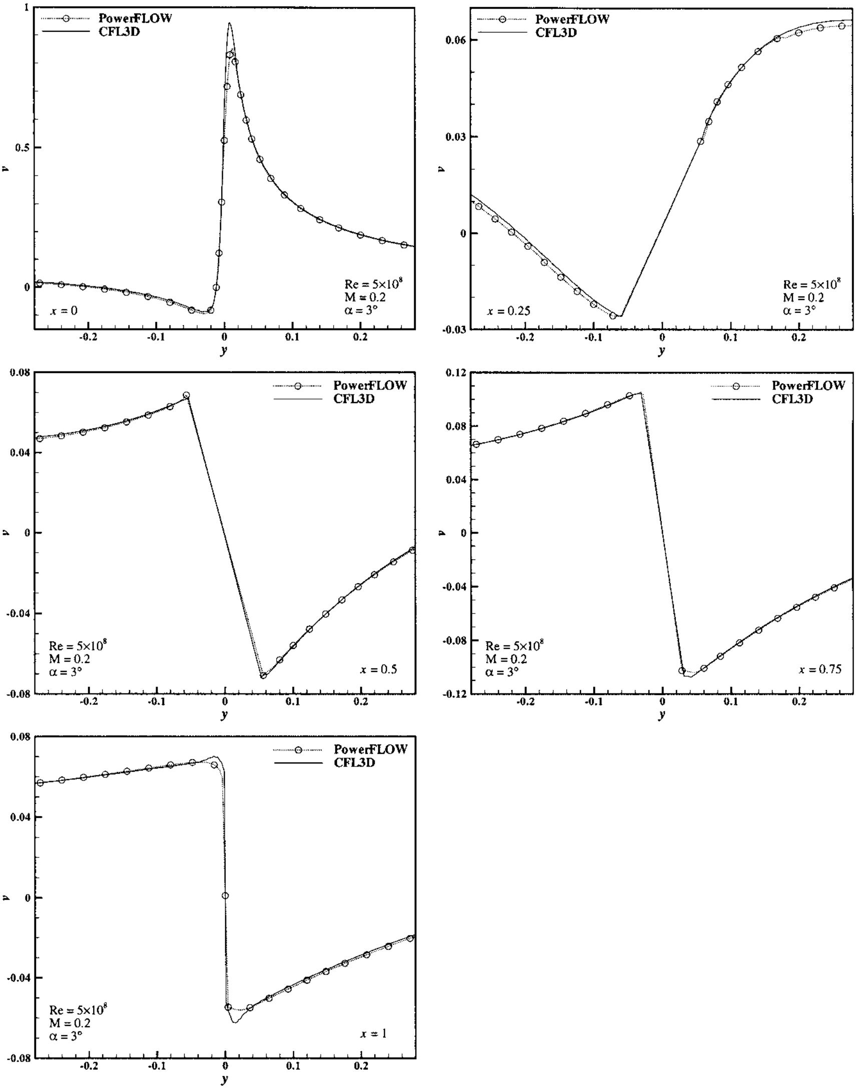

Fig. 19. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^9$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

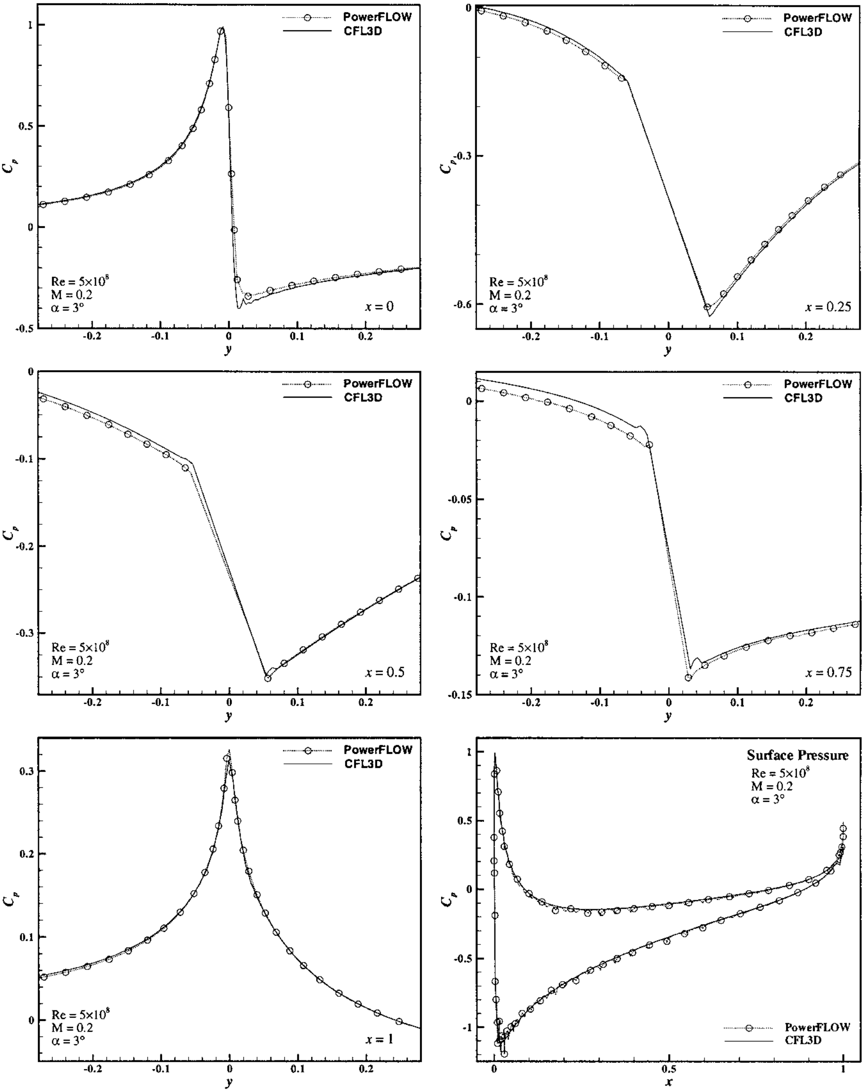

Fig. 20. NACA-0012 airfoil with angle of attack $\alpha = 3°$, $\mathrm{Re} = 0.5 \times 10^9$, and M = 0.2. The pressure coefficient $C_p(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length, and along the surface. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

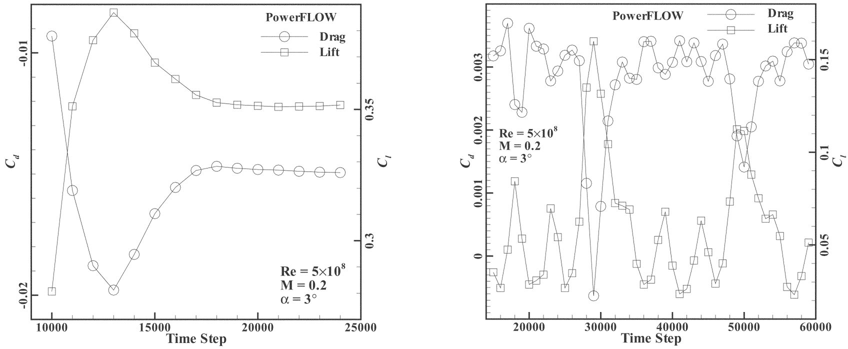

Fig. 21. Time history of C_d and C_l in the PowerFLOW simulation. $\mathrm{Re} = 0.5 \times 10^9$, $\mathrm{M} = 0.2$, and $\alpha = 3°$. (left) Computational domain $L_x \times L_y = 9 \times 8$. The total resolution is 383, 724 voxels and 1, 275 surfels. The resolution on the airfoil surface is 500 points along the chord. Note that C_d converges to a negative value. (right) Computational domain $L_x \times L_y = 21 \times 20$. The total resolution is 63, 612 voxels and 604 surfels. The resolution on the airfoil surface is 200 points along the chord. Neither C_d nor C_l converge after 60,000 steps of iteration.

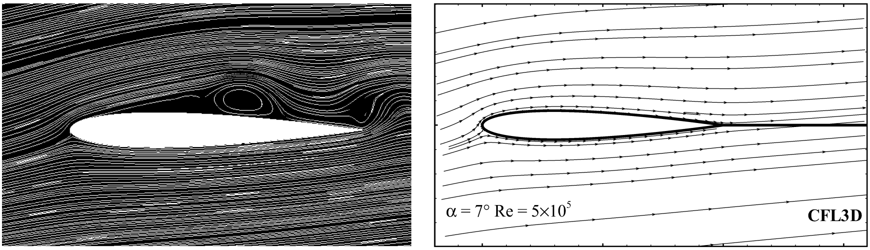

Fig. 22. NACA-0012 airfoil with angle of attack $\alpha = 7°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. Streamlines of PowerFLOW simulation (left) vs. CFL3D simulation (right).

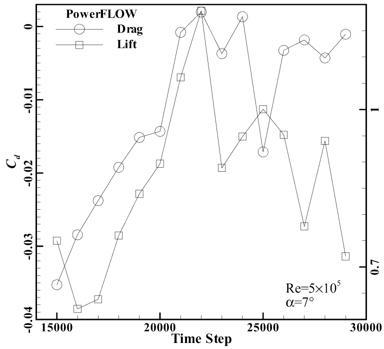

Fig. 23. Time history of C_d and C_l in the PowerFLOW simulation. $\mathrm{Re} = 0.5 \times 10^6$, $\mathrm{M} = 0.2$, and $\alpha = 7°$. Note that the value of C_d is negative most of time, and both C_d and C_l do not converge after 30,000 steps of iteration.

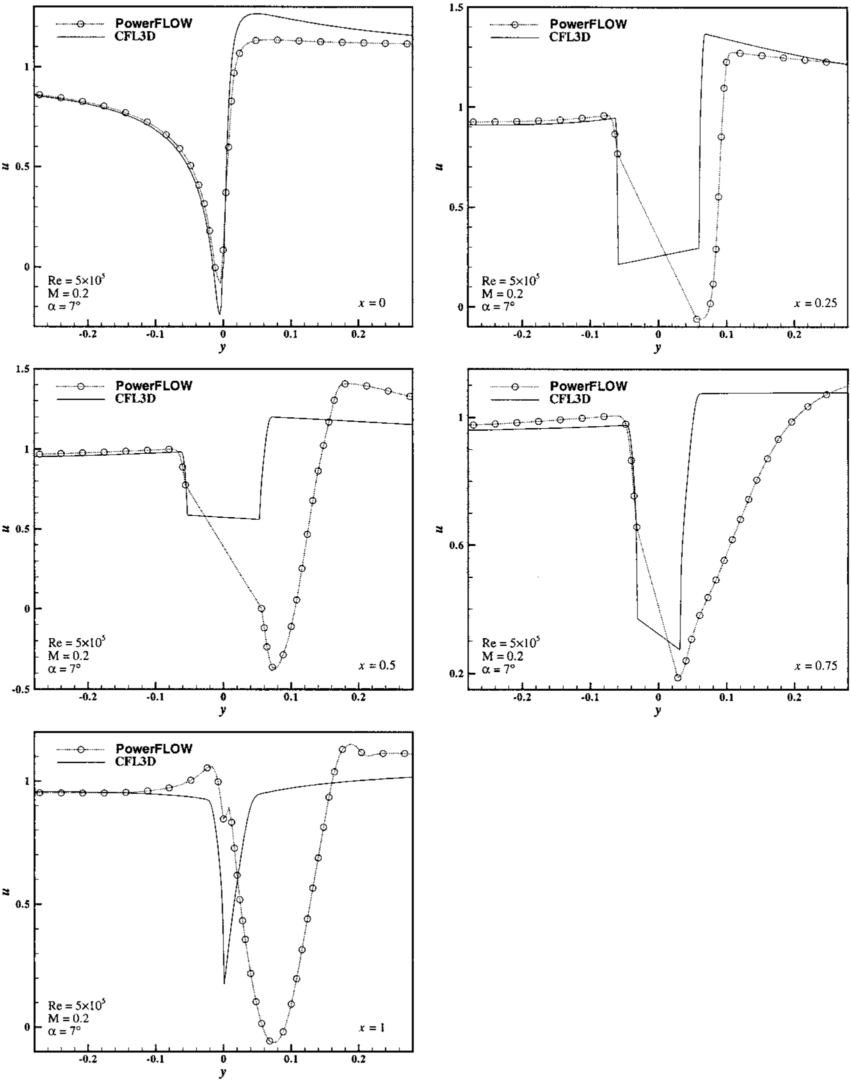

Fig. 24. NACA-0012 airfoil with angle of attack $\alpha = 7°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The x-component of velocity $u(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

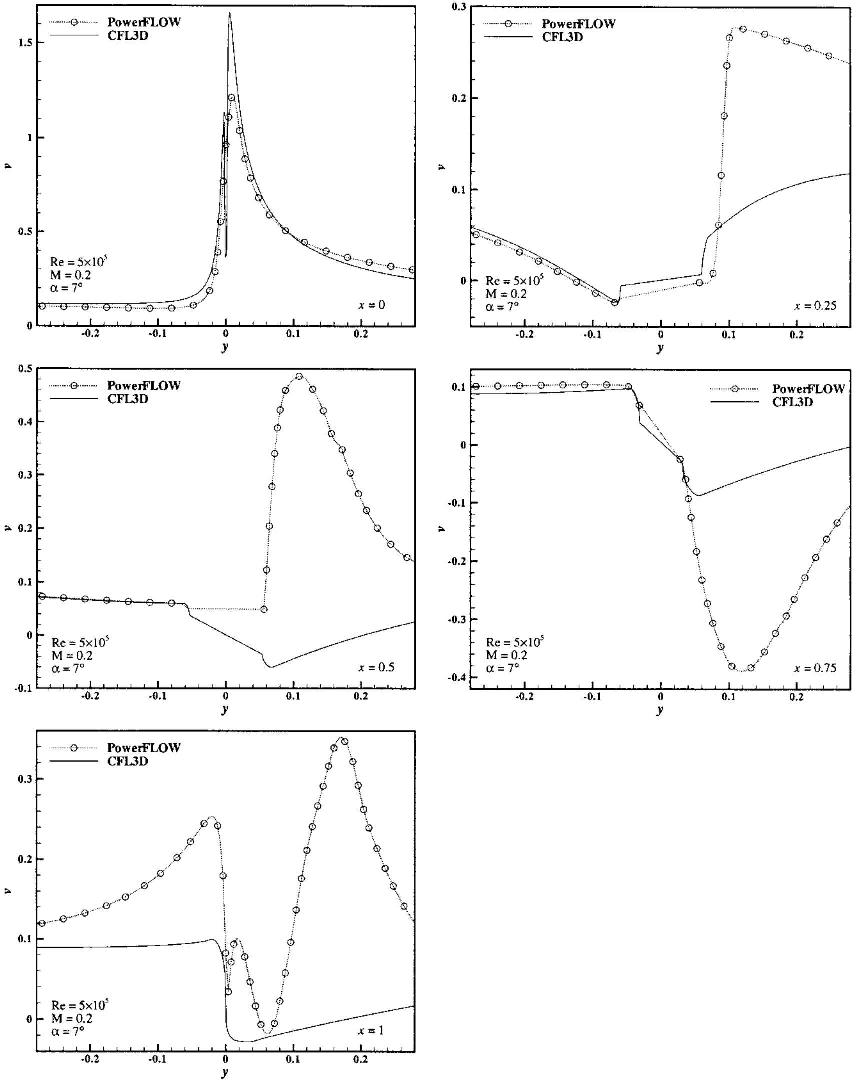

Fig. 25. NACA-0012 airfoil with angle of attack $\alpha = 7°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

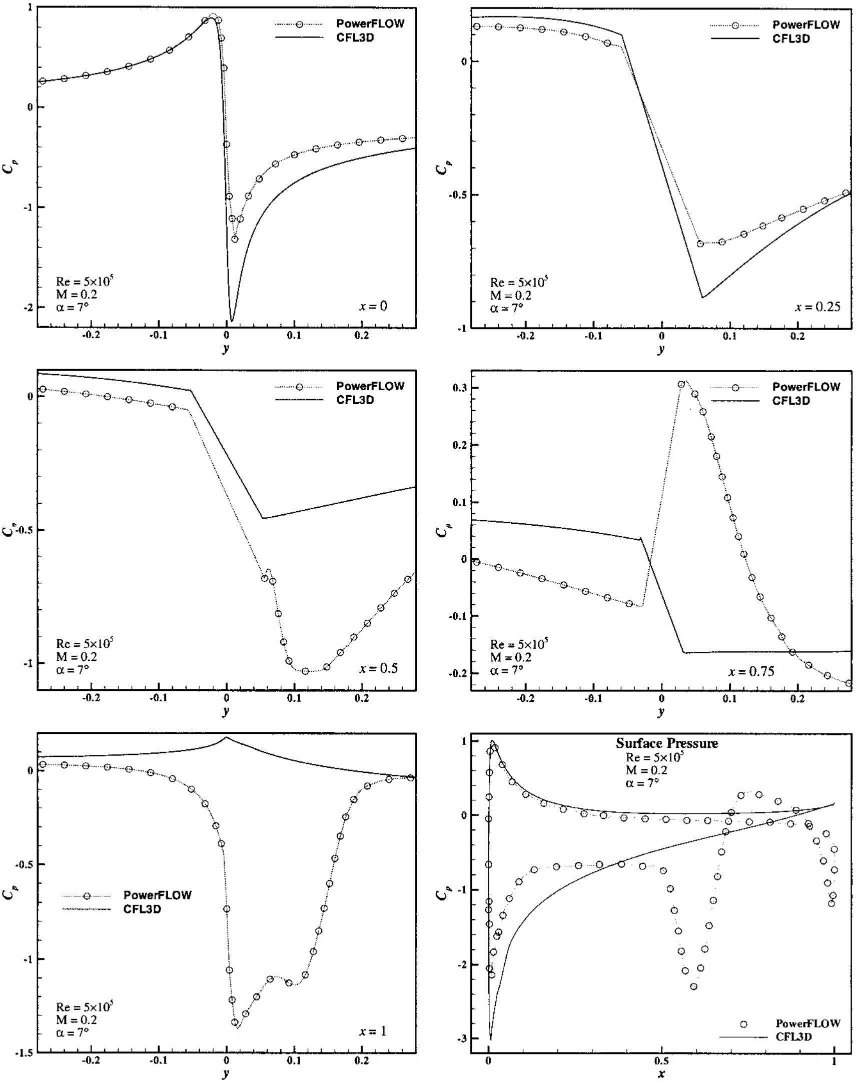

Fig. 26. NACA-0012 airfoil with angle of attack α = 7°, Re = 0.5×10^6, and M = 0.2. The pressure coefficient $C_p(x, y)$ at $x/L = 0.0, 0.25, 0.5, 0.75$, and 1.0, where L is the chord length, and along the surface. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373 × 141 for CFL3D.

completely attached, the solution from PowerFLOW undergoes a massive separation, as also shown in Fig. 22.

Case 6. $\mathrm{Re} = 0.5 \times 10^6$, $\alpha = 12°$. At $\alpha = 12°$, the observation is similar to the previous case of $\alpha = 7°$; in the PowerFLOW simulation the flow separates massively while it should not. The mean flow fails to converge to a steady state, as it should. Therefore, the flow fields obtained in PowerFLOW simulation are qualitatively incorrect, as shown in Figures 27, 28, and 29.

6. DISCUSSION AND CONCLUSION

In this work we present the results of a thorough investigation using PowerFLOW to simulate two-dimensional flow past a NACA-0012 airfoil with a Reynolds number ranging from 500 to 0.5×10^9. The PowerFLOW results were compared with the results obtained from CFL3D.

In all cases we observed deficiencies with the PowerFLOW calculations. We hypothesize that the principal culprit for the observed deficiencies is the the lack of wall-normal grid resolution near the surface of the airfoil at high Reynolds numbers. Because PowerFLOW requires grid cells to be square, properly sizing the grid cells in the wall-normal direction requires grid cells in the streamwise direction to be so small as to make even a two-dimensional calculation infeasible. Errors in the near-wall region degrade the accuracy of the overall aerodynamic quantities, such as the lift and drag coefficients. Although the use of wall functions for the high-Reynolds number turbulent cases relieves some of the resolution requirements for those cases, the data suggest that the resolution used was still insufficient.

At a low Reynolds number of $\mathrm{Re} = 500$, the results of the lattice Boltzmann simulations with the D2Q9 model agree well with the CFL3D results, as shown in Figs. 8 and 9. However, our tests show that the PowerFLOW results disagree with those obtained with the lattice Boltzmann D2Q9 model and CFL3D near the boundary. This clearly indicates that the implementation of fluid-solid interaction at the boundary may also be a source of error.

Figure 30 shows a velocity profile normal to the wing surface as computed by CFL3D. The variables plotted are in wall coordinates (i.e., $y^+ = u_\tau y/v$ and $u^+ = U/u_\tau$ where $u_\tau = \sqrt{\tau_{\mathrm{wall}}/\rho}$ and τ_{wall} is the wall shear stress), which emphasize the profile in the near-wall region. The vertical lines indicate the locations of the first through fifth grid points from the surface in the PowerFLOW grid. Although the first PowerFLOW grid point is located at an appropriate position for implementing a wall model,

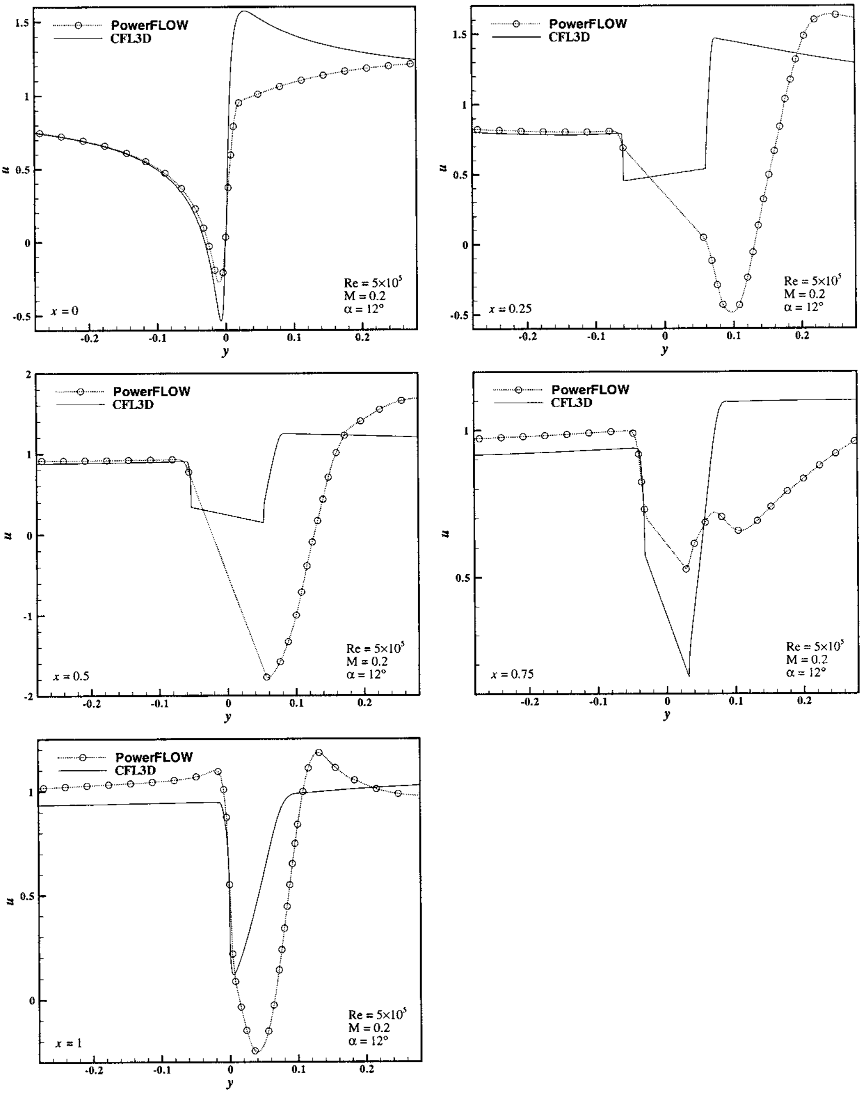

Fig. 27. NACA-0012 airfoil with angle of attack $\alpha = 12°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The x-component of velocity $u(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

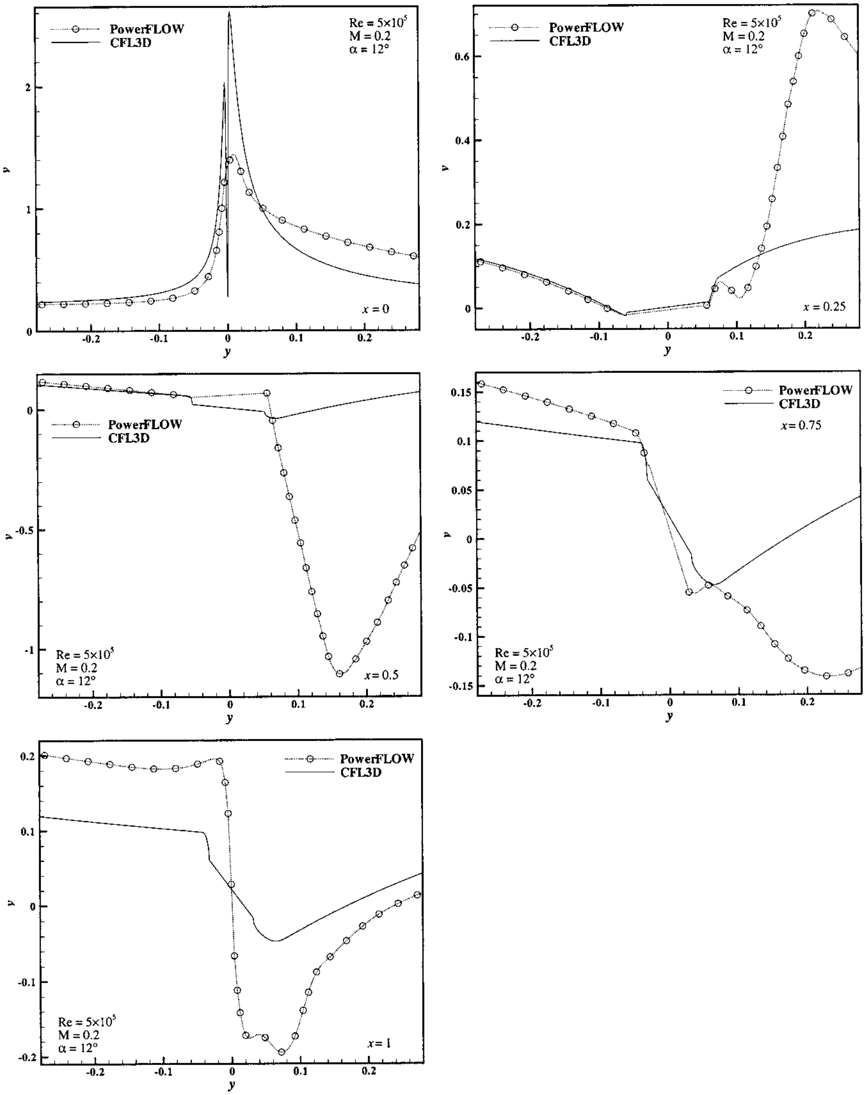

Fig. 28. NACA-0012 airfoil with angle of attack $\alpha = 12°$, $\mathrm{Re} = 0.5 \times 10^6$, and $\mathrm{M} = 0.2$. The y-component of velocity $v(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373×141 for CFL3D.

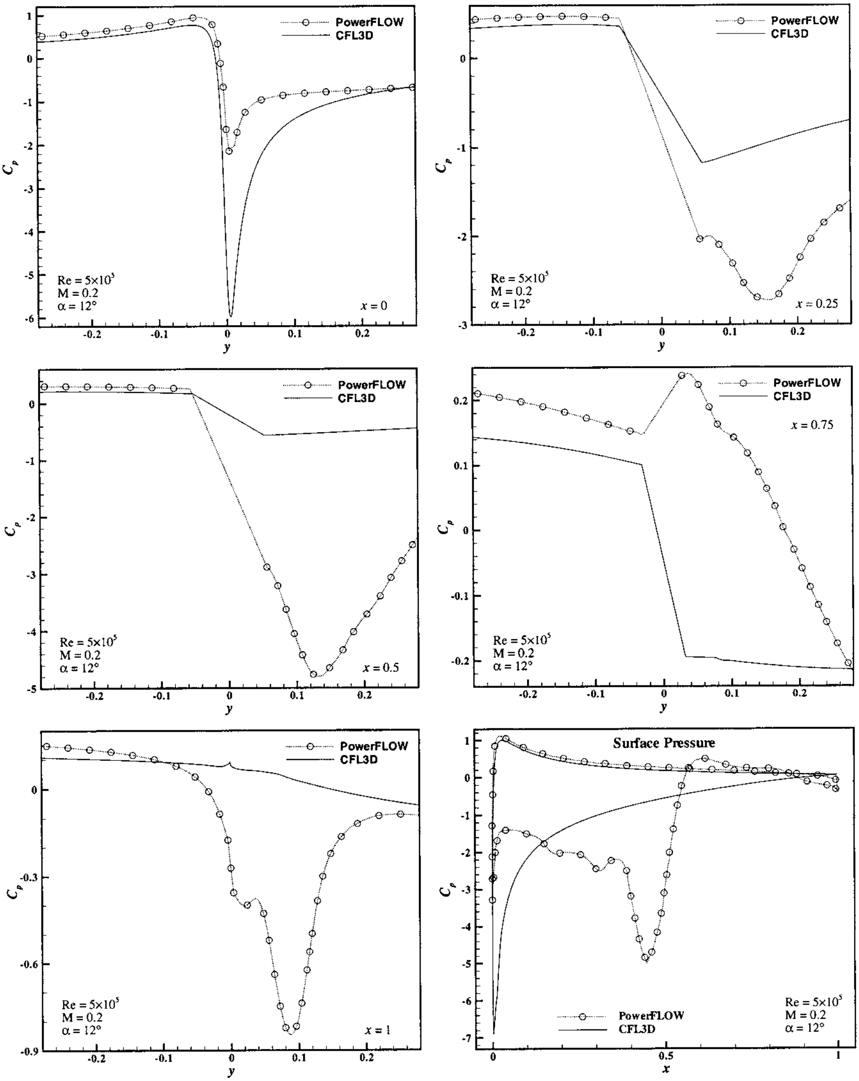

Fig. 29. NACA-0012 airfoil with angle of attack α = 12°, Re = 0.5 × 10^6, and M = 0.2. The pressure coefficient $C_p(x, y)$ at $x/L = 0.0$, 0.25, 0.5, 0.75, and 1.0, where L is the chord length, and along the surface. The resolution is 383, 724 voxels and 1, 275 surfels for PowerFLOW and 373 × 141 for CFL3D.

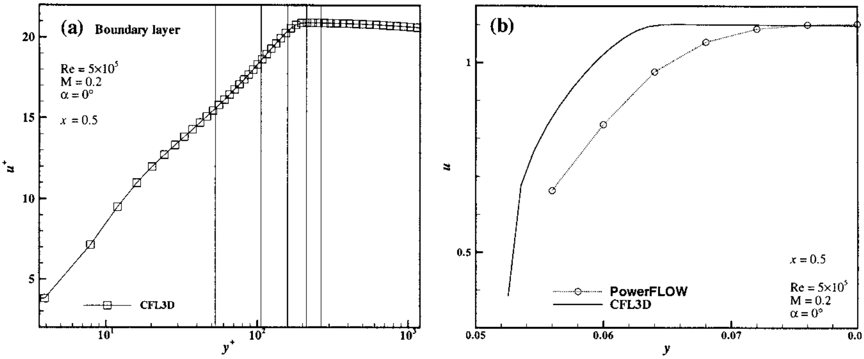

Fig. 30. Velocity profile normal to the wing surface at $x/L = 0.5$. $\alpha = 0°$, $\mathrm{M} = 0.2$, and $\mathrm{Re} = 0.5 \times 10^6$. (a) Velocity profile normal to the wing surface computed by CFL3D. The variables plotted are in wall coordinates. The vertical lines indicate the locations of the first through fifth grid points from the surface in the PowerFLOWgrid. (b) $u(x, y)$ at $x = 0.5$ computed by CFL3D (solid line) and PowerFLOW (dotted-line and circles) near the boundary.

the PowerFLOW grid only has about 5 points in the entire boundary layer. That level of grid resolution is inadequate for properly computing the boundary layer.

In this flow, incorrectly computing the boundary layer produces catastrophic problems. Flow separation occurs when the downstream momentum of the fluid in the boundary layer is insufficient to maintain the downstream movement of the fluid against an adverse pressure gradient. The momentum of the fluid in the boundary layer is decreased by the shear stress at the wall, but is increased by the (usually turbulent) diffusion of momentum from the free stream into the boundary layer. Incorrectly computing the boundary layer can dramatically alter where the flow first separates and the degree of separation. After massive flow separation occurs, the computed flow has little resemblance to its attached-flow counterpart.

In the test problems studied here, minor separation was observed in the PowerFLOW results for a case with a three-degree angle of attack. Massive separation was observed in the PowerFLOW results for a seven-degree angle of attack case. After separation occurred, even the gross aerodynamic quantities could not be computed well. It should also be pointed out that the PowerFLOW is designed for unsteady, time-dependent flows, while CFL3D is formulated to directly compute a turbulent steady state through an efficient iterative process. In order to obtain a steady-state solution from PowerFLOW, the unsteady solution must be computed long enough to obtain a statistically meaningful average. For certain problems, this averaging time can be very long. Due to resource constraints, only limited variations in the averaging time were investigated. The significant differences between the solution procedures in CFL3D and PowerFLOW may partially explain the differences observed in the results.

Two additional tests were carried out at NASA Langley Research Center: (a) simulation of three-dimensional cavity flow,[61] and (b) three-dimensional flow past the ROBIN (ROtor Body INteraction) configuration, a generic helicopter configuration consisting of a body and an engine-transmission pylon.[62]

The geometry of the cavity flow is very simple. The Mach number and the unit Reynolds number of the problem are 0.3 and $6\times 10^6/\text{ft.}$, respectively. The resolution required to obtain reasonable results is 9,240,000 voxels and 286,500 surfels. A run of 20,000 time steps required approximately 60 hours on an SGI Origin2000 with sixteen 250MHz processors. Although quantitative comparisons with experimental results of mean wall pressure distributions, the sound pressure level, and length scales of the flow were not satisfactory,[61] the simulation did capture the acoustic interaction with the shear layer in the flow. Constraints on time and memory size precluded further simulations with finer resolutions.

The ROBIN is a generic three-dimensional helicopter configuration. The Reynolds number of the flow is 9.2×10^6 based on the body length (2 meters). Five levels of grid resolution were used in the PowerFLOW simulations (2^0 to 2^{-4}). The resulting grid had 4.3×10^6 voxels. The PowerFLOW simulations were compared with results obtained by using Navier–Stokes solvers on a structured grid (CFL3D) and an unstructured grid (USM3D), and with experimental observations. The main concerns with the PowerFLOW simulations of the flow past ROBIN are: (i) too large a value for the maximum pressure coefficient C_p; (ii) an excessive amount of oscillation in the pressure coefficient distributions and skin friction coefficient distributions; (iii) significant differences in surface flow patterns and non-dimensional stagnation pressure distributions from those predicted by the Navier–Stokes solvers; and (iv) drag and lift coefficients that differ significantly from those predicted by the Navier–Stokes solvers.[62]

Based on our investigation of the two-dimensional flow past the NACA-0012 airfoil, it is our conclusion that PowerFLOW at its present state is inadequate for aerodynamic applications in two aspects: performance and accuracy. PowerFLOW is generally much less accurate than CFL3D and much slower as well.

Some successful examples using PowerFLOW include simulating the flow past a vehicle.[17] The reason for this success, we suspect, is that in the case of the flow past a vehicle, the bluff body has geometric discontinuities that dictate the location of flow separation. Therefore, the flow separation, which is the most important flow feature in this case, is not affected by other factors as much. In this case, PowerFLOW seems to be able to deliver better results.[17] However, when dealing with a smooth or streamlined body such as an airfoil, PowerFLOW does not provide reliable results. We also observed the fact that simulations using PowerFLOW to obtain a steady mean state generally require much larger memory and longer CPU time than the CFL3D calculations.

Although the results of the aerodynamic simulations revealed a less than adequate performance of PowerFLOW compared with CFL3D, further advancements in LGA and LBE methodology may enable this approach to become a viable alternative to traditional CFD. LBE methods have already been used successfully for applications involving porous media and bluff bodies, and further research should enable successful applications to streamlined aerodynamic bodies, as shown in our preliminary study. Some researchers have also obtained good solutions for airfoil flows with Reynolds numbers as high as 5000 by using interpolation schemes at boundaries and multi-block grids to handle surface curvature.[63] Continued refinements by the LBE community and Exa corporation will undoubtedly have occurred by the time this paper is disseminated.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Stanley Willner of the Hydrodynamic/Hydroacoustic Technology Center of the Naval Sea Systems Command for providing resources for this investigation. The authors are grateful to Dr. Gregory Jones and Dr. Raymond Mineck for sharing and discussing their test results using PowerFLOW[61, 62] with the authors, and to Dr. Michael Barton, Prof. Dominique d'Humières, Dr. Manfred Krafczyk, Prof. Pierre Lallemand, Prof. Renwei Mei, Dr. Cord Rossow, Dr. Robert Rubinstein, and Prof. Wei Shyy for helpful discussions. We would also like to thank Dr. James L. Thomas and Mr. William Sellers of NASA Langley Research Center, and Dr. Manuel D. Salas of ICASE for their support of this work. Technical assistance from Exa Corporation during the test is also acknowledged.

REFERENCES

1. G. D. Doolen, ed., *Lattice Gas Methods for Partial Differential Equations* (Addison-Wesley, New York, 1990).
2. B. M. Boghosian, F. J. Alexander, and P. Coveney, eds., Proceedings of 1996 conference on discrete models for fluid mechanics, *Internat. J. Modern Phys. C* **8**:637–1011 (1997)
3. B. M. Boghosian, ed., Proceedings of the 7th international conference on the discrete simulation of fluids, *Internat. J. Modern Phys. C* **9**:1123–1617 (1998).

4. R. Benzi, S. Succi, and M. Vergassola, The lattice Boltzmann equation: Theory and applications, *Phys. Rep.* **222**:145–197 (1992).
5. S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* **30**:329–364 (1998).
6. L.-S. Luo, The future of lattice-gas and lattice Boltzmann methods, in *Computational Aerosciences in the 21st Century*, M. D. Salas and W. K. Anderson, eds. (Kluwer, Dordrecht, 2000), pp. 165–187.
7. D. H. Rothman and S. Zaleski, *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics* (Cambridge University Press, Cambridge, 1997).
8. B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems* (Cambridge University Press, Cambridge, 1998).
9. S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Clarendon Press, Oxford, 2001).
10. J. Buckles, R. Hazlett, S. Chen, K. G. Eggert, D. W. Grunau, and W. E. Soll, Flow through porous media using lattice Boltzmann method, *Los Alamos Sci.* **22**:112–121 (1994).
11. N. S. Martys and H. Chen, Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method, *Phys. Rev. E* **53**:743–750 (1994).
12. S. Succi, Lattice Boltzmann equation: Failure or success? *Physica A* **240**:221–228 (1997).
13. H. Chen, P. C. Churchill, R. A. Iannucci, K. Molvig, G. Papadopoulos, S. A. Remondi, C. M. Teixeira, and K. Traub, Computer system for simulating physical processes using multiple-integer state vectors, US Patent 5,594,671 (1997).
14. H. Chen, Volumetric formulation of the lattice Boltzmann method for fluid dynamics: Basic concept, *Phys. Rev. E* **58**:3955–3963 (1998).
15. H. Chen, C. Teixeira, and K. Molvig, Realization of fluid boundary conditions via discrete Boltzmann dynamics, *Internat. J. Modern Phys. C* **9**:1281–1292 (1998).
16. Exa Corp., *PowerFLOW User's Guide 3.0* (Exa Corp., Lexington, MA, 1998).
17. A. Anagnost, A. Alajbegovic, H. Chen, D. Hill, C. Teixeira, and K. Molvig, Digital physics analysis of the Morel body in ground proximity, SAE Technical Paper Series No. 970139 (1997).
18. I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections, Including a Summary of Airfoil Data* (Dover, New York, 1959).
19. V. N. Vatsa, J. L. Thomas, and B. W. Wedan, Navier–Stokes computations of prolate spheroids at angle of attack, *J. Aircraft* **26**:986–993 (1989).
20. J. L. Thomas, S. L. Krist, and W. K. Anderson, Navier–Stokes computations of vortical flows over low aspect-ratio wings, *AIAA Journal* **28**:205–212 (1990).
21. J. L. Thomas, Reynolds number effects on supersonic asymmetrical flows over a cone, *J. Aircraft* **30**:488–495 (1993).
22. C. L. Rumsey, M. D. Sanetrik, R. T. Biedron, N. D. Melson, and E. B. Parlette, Efficiency and accuracy of time-accurate turbulent Navier–Stokes computations, *Comput. Fluids* **25**:217–236 (1996).
23. C. L. Rumsey, T. B. Gatski, S. X. Ying, and A. Bertelrud, Prediction of high-lift flows using turbulent closure models, AIAA Paper 97-2260 (1997).
24. S. L. Krist, R. T. Biedron, and C. L. Rumsey *CFL3D User's Manual*, (Version 5.0), 2nd Edition, NASA Langley Research Center, Hampton, Virginia, 1997. Also see the online version at: http://fmad-www.larc.nasa.gov/~rumsey/CFL3D/WEB/CFL3D-1.html.
25. Y. H. Qian, D. d'Humières, and P. Lallemand, Lattice BGK models for Navier–Stokes equation, *Europhys. Lett.* **17**:479–484 (1992).
26. P. Lallemand and L.-S. Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, *Phys. Rev. E* **61**:6546–6562 (2000).

27. S. Wolfram, *Cellular Automata and Complexity: Collected Papers* (Addison-Wesley, New York, 1994).
28. J. Hardy, Y. Pomeau, and O. de Pazzis, Time evolution of a two-dimensional classical lattice system, *Phys. Rev. Lett.* **31**:276–279 (1973).
29. J. Hardy, Y. Pomeau, and O. de Pazzis, Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions, *J. Math. Phys.* **14**:1746–1759 (1973).
30. U. Frisch, B. Hasslacher, and Y. Pomeau, Lattice-gas automata for the Navier–Stokes equation, *Phys. Rev. Lett.* **56**:1505–1508 (1986).
31. S. Wolfram, Cellular automaton fluids 1: Basic theory, *J. Statist. Phys.* **45**:471-526 (1986).
32. U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet, Lattice gas hydrodynamics in two and three dimensions, *Complex Systems* **1**:649–707 (1987).
33. A. Clouqueur and D. d'Humières, RAP1, a cellular automaton machine for fluid dynamics, *Complex Systems* **1**:585–597 (1987).
34. T. Toffoli and N. Margolus, *Cellular Automata Machines* (MIT Press, Cambridge, 1987).
35. C. Adler, B. M. Boghosian, E. G. Flekkøy, N. Margolus, and D. H. Rothman, Simulating three-dimensional hydrodynamics on a cellular-automata machine, *J. Statist. Phys.* **81**: 105–128 (1995).
36. L. Boltzmann, *Lectures on Gas Theory* (Dover, New York, 1995).
37. H. Chen, C. Teixeira, and K. Molvig, Digital physics approach to computational fluid dynamics: Some basic theoretical features, *Internat. J. Modern Phys. C* **8**:675–684 (1997).
38. G. McNamara and G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata, *Phys. Rev. Lett.* **61**:2332–2335 (1988).
39. H. Chen, S. Chen, and W. H. Matthaeus, Recovery of the Navier–Stokes equations using a lattice gas Boltzmann method, *Phys. Rev. A* **45**:R5339–R5342 (1992).
40. X. He and L.-S. Luo, A priori derivation of the lattice Boltzmann equation, *Phys. Rev. E* **55**:R6333–R6336 (1997).
41. X. He and L.-S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev. E* **56**:6811–6817 (1997).
42. X. Shan and X. He, Discretization of the velocity space in the solution of the Boltzmann equation, *Phys. Rev. Lett.* **80**:65–68 (1998).
43. P. L. Bhatnagar, E. P. Gross, and M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* **94**:511–525 (1954).
44. X. He, L.-S. Luo, and M. Dembo, Some progress in lattice Boltzmann method. Part I. Nonuniform mesh grids, *J. Comput. Phys.* **129**:357–363 (1996).
45. X. He, L.-S. Luo, and M. Dembo, Some progress in lattice Boltzmann method. Reynolds number enhancement in simulations, *Physica A* **239**:276–285 (1997).
46. D. d'Humières, Generalized lattice-Boltzmann equations, in *Rarefied Gas Dynamics: Theory and Simulations*, *Prog. Astro. Aeor.*, Vol. 159, B. D. Shizgal and D. P. Weaver, eds. (AIAA, Washington, DC, 1992).
47. D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, to appear in *Proc. Roy. Soc. London A* (2002).
48. M. Junk and A. Klar, Discretizations for the incompressible Navier–Stokes equations based on the lattice Boltzmann method, *SIAM J. Sci. Comput.* **22**:1–19 (2001).
49. M. Junk, A finite difference interpretation of the lattice Boltzmann method, *Numer. Methods Partial Differential Eq.* **17**:383–402 (2001).
50. R. Mei, L.-S. Luo, and W. Shyy, An accurate curved boundary treatment in the lattice Boltzmann method, *J. Comput. Phys.* **155**:307–330 (1999).

51. O. Filippova and D. Hänel, Grid refinement for lattice-BGK models, *J. Comput. Phys.* **147**:219–228 (1998).
52. S. Hou, J. Sterling, S. Chen, and G. D. Doolen, A lattice Boltzmann subgrid model for high Reynolds number flows, in *Fatten Formation and Lattice Gas Automata*, *Fields Institute Communications*, Vol. 6, A. T. Lawniczak and R. Kapral, eds. (Am. Math. Soc., Providence, 1996).
53. C. Teixeira, Incorporating turbulence models into the lattice Boltzmann method, *Internat. J. Modern Phys. C* **9**:1159–1175 (1998).
54. P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**:357–372 (1981).
55. P. R. Spalart and S. Allmaras, A one-equation turbulence model for aerodynamic flows, AIAA Paper 92-0439 (1992).
56. D. Wilcox, *Turbulence modeling for CFD* (DCW Industries, Inc., La Canada, CA, 1993).
57. S. W. Rogers, F. R. Menter, P. A. Durbin, and N. N. Mansour, A comparison of turbulence models in computing multi-element airfoil flows, AIAA Paper 94-9201 (1994).
58. P. R. Spalart and S. Allmaras, A one-equation turbulence model for aerodynamic flows, *La Recherche Aerospatiale* **1**:5–21 (1994).
59. C. Teixeira, private communication.
60. M. Bouzidi, M. Firdaouss, and P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, *Phys. Fluids* **13**:3452–3459 (2001).
61. G. Jones, NASA LaRC active flow cavity program, preprint (1999).
62. R. E. Mineck, Evaluation of a lattice gas automata technique for predicting aerodynamic characteristics of Rotorcraft, preprint (1998).
63. D. Yu, R. Mei, and W. Shyy, A Multi-block lattice Boltzmann method for fluid flows, AIAA Paper 2000-2614 (2000).