Building and Environment 126 (2017) 184–194

Contents lists available at ScienceDirect

Building and Environment

journal homepage: www.elsevier.com/locate/buildenv

A state-space method for real-time transient simulation of indoor airflow

Qiujian Wang[a], Yiqun Pan[a,*], Mingya Zhu[a], Zhizhong Huang[a], Wei Tian[b], Wangda Zuo[c], Xu Han[c], Peng Xu[a]

[a] Tongji University, Shanghai, China
[b] University of Miami, Coral Gables, FL, USA
[c] University of Colorado, Boulder, CO, USA

ARTICLE INFO

Keywords:
CFD
FFD
State-space fluid dynamics (SFD)
Transient air flow simulation
Real-time simulation

ABSTRACT

Inhomogeneous airflow distribution is common in air-conditioned rooms, especially the large open spaces. To evaluate the thermal comfort of such space, or the control performance of the Heating, Ventilation, and Air Conditioning (HVAC) systems in an efficient way, one will need a fast prediction method to simulate the airflow and temperature distribution. This paper proposes a discrete state-space method, called state-space fluid dynamics (SFD), for the fast indoor airflow simulation. To handle time-varying velocity and temperature field, SFD converts all the governing equations of fluid dynamics into the form of a state-space model. Four typical cases are selected to evaluate both the accuracy and speed of SFD, compared with fast fluid dynamics (FFD), which is another fast airflow simulation program. Results show that SFD is capable of achieving faster-than-real-time airflow simulation with an accuracy similar to FFD. The computing time of SFD is longer than FFD when the time step size is the same. However, SFD can generally produce better results than FFD when the time step size is larger, which allows SFD run faster than FFD.

1. Introduction

Advanced controls of Heating, Ventilation, and Air Conditioning (HVAC) systems are intensively being implemented to improve energy efficiency and thermal comfort for buildings. Stratified airflow distribution, such as displacement ventilations, in a space is usually introduced by those systems. To evaluate the control performance of the systems using simulation, predictions of temperature and airflow distribution are critical. As summarized by Chen [1], there are mainly three types of models for airflow prediction: multizone models, zonal models, and computational fluid dynamics (CFD) models. Multizone models are computationally fast. However, they are not suitable for the simulation of non-uniform airflow and temperature distribution, as they assume that the air is well-mixed [2]. Zonal models [3] divide the room into several subzones and use empirical formulas to simulate airflow in specific area. But users of zonal model should be aware in advance of where the specific areas are, which is not very flexible. CFD models are versatile and can provide the most comprehensive information. It is nevertheless too computationally demanding to be applied in control simulations [4].

To resolve the obstacle imposed on CFD models, many researchers have explored various alternatives in order to reduce the computing time for non-uniform airflow and temperature distribution. These include the coarse-grid techniques, and some efficient equation-solving techniques. The coarse-grid techniques are more straightforward solutions to speed up CFD simulations, as calculation loads drop substantially with the decrease of grid nodes. Wang and Zhai [5] systematically examined the accuracy and speed improvement of coarse-grid CFD under several cases. Coarse-grid CFD can reduce the computing time by more than 16 times (even 100 times for some cases) compared to fine-grid CFD and achieve acceptable accuracy at the same time (even more accurate for some cases). However, one should bear in mind that the pre-acquisition of the airflow prior to applying the coarse-grid techniques is pivotal.

A more fundamental way to speed up CFD is to develop efficient equation-solving techniques such as fast fluid dynamics (FFD) and state-space methods. FFD uses the time splitting method and solves the governing equations sequentially after dividing the complex equation into several simple ones according to the number of terms in the equation. FFD was first developed by Stam [6] for simulating fast fluid movement in computer games. Zuo and Chen first introduced FFD into the airflow simulation in building area, by systematically evaluating its accuracy and speedup over CFD [7]. With a certain loss of accuracy, FFD is found to be 50 times faster than CFD models when running on CPU, and another 30 times speedup can be achieved if one is running simulations on a graphics processing unit (GPU) [8]. Furthermore, Tian,

* Corresponding author. 4800 CaoAn Highway, Shanghai 201804, China.
E-mail address: yiqunpan@tongji.edu.cn (Y. Pan).

http://dx.doi.org/10.1016/j.buildenv.2017.09.032
Received 6 July 2017; Received in revised form 27 September 2017; Accepted 29 September 2017
Available online 30 September 2017

Nomenclature

u, v	Velocity, m/s
ρ	Density, kg/m^3
e	Internal energy, J
T	Temperature, K
R	Gas constant for air, J/kg
C_p, C_v	Specific heat at constant pressure and volume, $J/(kg{\cdot}K)$
μ	Dynamic viscosity, $Pa{\cdot}s$
μ_t	Turbulent viscosity, $Pa{\cdot}s$
λ	Thermal conductivity, $W/(m{\cdot}K)$
l	Length, m
Pe	Peclet number
Pr	Prandtl number
CTR	Computing time ratio

Superscripts

n, $n-1$	Time step index

Subscripts

Eff	Effective
r, l, t, b	Location index of the cell boundaries
R, L, T, B	Location index of the cell

Sevilla, Li, Zuo and Wetter [9] combined FFD with an in situ adaptive tabulation (ISAT) algorithm that is essentially a reduced order model. In addition, Jin, Liu and Chen [10] applied coarse grid technique to FFD in simulation of buoyancy-driven flow in buildings to improve the FFD speed even further. They integrated an analytical model of plume to improve the simulation performance for the heat source whose physical size is smaller than the mesh size. Additionally, the simulation speed was also accelerated.

After all the descriptions above, FFD seems like a more versatile framework with which to do fast flow simulation. However, FFD usually requires a small time step size (sometimes as small as 0.01s) when using semi-lagrangian method for the convection term, which introduces a numerical viscosity related to the time step size. Additionally, the first-order time splitting method imposes restrictions on choosing the time step size. Thus, FFD can be costly in computation time when simulating the slowly changing flows because of the long simulation time required. To solve this problem, some researchers have tried to use higher order interpolation to mitigate the numerical diffusion caused by semi-lagrangian method [11] or directly replacing the semi-lagrangian with other implicit schemes [12]. But a more complicated scheme or method is likely to impact the speed and convergence.

Compared to FFD, state-space method works towards the same goal with a different approach. Conventionally, state-space method only solves the energy conservation equations. Precedent CFD calculations are often utilized to calculate the mass flow rate and thermal conductivity on the boundaries in order to define the mutual influence of adjacent cells. The matrices of state-space model are generated based on the precedent CFD results. More details of state-space method are given in section 2.1.

Peng and van Paassen [13] used the state-space method to simulate the dynamic response of temperature distribution for a room with forced convection air flow. They assumed the flow field was fixed and used CFD to pre-calculate the velocity field and turbulence viscosity. After that, the discretized energy conservation equations were transformed into the state-space model and solved. By not solving the momentum conservation equations, they saved a great amount of computing time. Yao, Yang, Huang and Wang [14] also used state-space method to calculate the dynamic response of not only the temperature but also humidity in a three-zoned room. In addition, Parker, Lorenzetti and Sohn [15] implemented it for the multi-zone contaminant transport.

To further improve the speed of state-space method, researchers offered several different options. Some of them [13,14] merged the adjacent small cells into relatively large zones to directly reduce the size of state-space model. Sempey, Inard, Ghiaus and Allery [16] used Proper Orthogonal Decomposition (POD) method to find an optimal basis using several snapshots extracted from precedent transient CFD simulation result and achieved the reduced order. Parker, Lorenzetti and Sohn [15] proposed to solve the equations analytically using the matrix exponential and achieved speed improvement as well.

However, for the conventional way of using state-space method in fluid dynamics, a major limitation is the assumption of fixed velocity field. When the objective room is an open space with multiple independently controlled supply air inlets, the dimensions of the problem increases substantially, leaving it impractical to train the model using the pre-calculated CFD results.

In this paper, we propose to apply the state-space technique to the Navier-Stokes equations together with all the other governing equations in CFD. For ease of illustration, we call the proposed method State-space Fluid Dynamics (SFD) in the rest of the paper. First, we describe the SFD model in the methodology section. Then, we evaluate the accuracy and speed of SFD by using multiple typical airflows in buildings, including a lid driven cavity flow, a forced convection flow, a natural convection flow, and a mixed convection flow. We compare the results of SFD with its counterpart FFD. Finally, we propose the future work for SFD.

2. Methodology

This section will first introduce the method of state-space model and the governing equations of fluid dynamics. Then we will focus on the discretization and linearization of the equations.

2.1. State-space model

As we know, the state space model is usually used to describe a linear system represented as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t), \tag{1}$$

where t is time, $\boldsymbol{x}$ the state vector, $\boldsymbol{u}$ the input vector, $\boldsymbol{A}$ the system matrix, and $\boldsymbol{B}$ the input matrix. The system matrix $\boldsymbol{A}$ is used to describe the interactions between different variables in the state vector $\boldsymbol{x}$, while the input matrix $\boldsymbol{B}$ represents the influences from the input vector $\boldsymbol{u}$ to the state vector $\boldsymbol{x}$. Matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can be time-invariant or time-variant as shown in Equation (1). When they are time-variant, we can also use discrete time-variant state-space model to describe the system as Equation (2):

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k), \tag{2}$$

where k is the time step index, $\boldsymbol{A}(k)$ and $\boldsymbol{B}(k)$ are constant within the time step k and vary before entering time step $k+1$. In the context of airflow simulation, the state vector $\boldsymbol{x}(k)$ represents the different variables (e.g. velocity, temperature, and density) and the input vector $\boldsymbol{u}(k)$ represents the boundary conditions (e.g. inlet velocity and temperature, and outlet pressure etc.). The convection and diffusion between cells are represented by the system matrix $\boldsymbol{A}(k)$. The influences from the boundary conditions are described in the input matrix $\boldsymbol{B}(k)$. Since the velocity and diffusion coefficients on the cell boundaries are always changing as the flow field develops, we use the form of discrete time-variant state-space model to fit in the descriptive equations.

2.2. Governing equations

The two-dimensional form of governing equations used in the flow simulation are described as follows, including the Navier-Stokes equations (3) and (4), continuity equation (5), and energy conservation equation (6). In the energy conservation equation (6), we ignore the dissipation and the work done by surface force and body force, because the impact of them for indoor airflow is negligible.

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho uu)}{\partial x} + \frac{\partial(\rho vu)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[\mu_{eff}\left(2\frac{\partial u}{\partial x} - \frac{2}{3}(\nabla\cdot\vec{\boldsymbol{u}})\right)\right] + \frac{\partial}{\partial y}\left[\mu_{eff}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + k_x, \tag{3}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho vv)}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[\mu_{eff}\left(2\frac{\partial v}{\partial y} - \frac{2}{3}(\nabla\cdot\vec{\boldsymbol{u}})\right)\right] + \frac{\partial}{\partial x}\left[\mu_{eff}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + k_y, \tag{4}$$

$$\frac{\partial\rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0, \tag{5}$$

$$\frac{\partial(\rho e)}{\partial t} + u\frac{\partial(\rho e)}{\partial x} + v\frac{\partial(\rho e)}{\partial y} = \frac{\partial}{\partial x}\left[\lambda_{eff}\left(\frac{\partial T}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\lambda_{eff}\left(\frac{\partial T}{\partial y}\right)\right] + S, \tag{6}$$

where u, v are the velocity on x and y dimension, p is pressure, $\vec{\boldsymbol{u}}$ is the velocity vector, k_x, k_y are the body force on x and y dimension, e is the internal energy, T is temperature, S is the heat source. The μ_{eff} is the effective viscosity [17], which represents the turbulent influences and is the sum of turbulent viscosity and dynamic viscosity:

$$\mu_{eff} = \mu_t + \mu. \tag{7}$$

The λ_{eff} is the effective thermal conductivity and calculated by:

$$\lambda_{eff} = \frac{\mu_{eff}}{Pr_{eff}}C_p, \tag{8}$$

where Pr_{eff} is the effective Prandtl number and equals to 0.9 [17]. C_p is the specific heat.

To calculate the turbulent viscosity, we use the zero-equation model [17] as Equation (9) because it can avoid introducing extra differential equation. The zero-equation model is defined as:

$$\mu_t = 0.03874\rho Vl, \tag{9}$$

where V is the local velocity value and l is the characteristic length, which is the distance from the center of the cell to the nearest wall.

To fit Equations (3)–(6) into the form of state-space, three criteria should be satisfied. First, every equation should contain a first-order differential term to fit in the form of state-space model. By setting the fluid as compressible, each equation can have a first-order differential term ($\frac{\partial(\rho u)}{\partial t}$, $\frac{\partial(\rho v)}{\partial t}$, $\frac{\partial\rho}{\partial t}$, $\frac{\partial(\rho e)}{\partial t}$).

Second, the equation system should be closed. As the pressure is an unknown variable in this method, an extra governing equation should be added. Hereby, we use the equation of state for ideal gas,

$$p = \rho RT. \tag{10}$$

Then the energy variable is represented as:

$$e = C_pT - \frac{p}{\rho} = C_vT, \tag{11}$$

where C_v is the specific heat at constant volume.

Third, the equation system should be linear. There exist some non-linear terms in the governing equations. For example, the convection terms in the momentum and energy conservation equations are quadratic. The effective viscosity μ_{eff} and effective thermal conductivity λ_{eff} are often non-linear depending on the turbulence model being used.

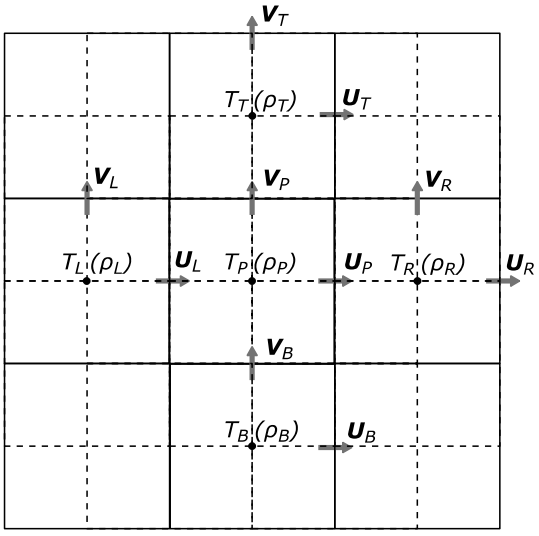


Fig. 1. Schematic of the location of discretization variables.

The pressure term is also non-linear, as it is the function of density and temperature as Equation (10). In section 2.3, we will introduce the linearization process of those terms.

2.3. Linearization of the nonlinear governing equations

This section will introduce the linearization of the non-linear terms through the process of discretization of the governing equations.

2.3.1. Discretization of governing equations

For the discretization of the transient terms, we use a first-order implicit scheme, which can guarantee the unconditional stability of the simulation. A staggered grid is adopted to store the vector and scalar variables separately as shown in Fig. 1.

A center differencing scheme is used to discretize the diffusion term. The convection term is discretized using a hybrid differencing method of a first order upwind differencing scheme and a second order central differencing scheme. The switch between the two schemes is determined by the Peclet number Pe:

$$Pe = \frac{w\rho l}{d_c}, \tag{12}$$

where w is the velocity on the cell boundary, ρ the local density, l the cell length, and d_c the diffusion coefficient, i.e. effective viscosity μ_{eff} and thermal conductivity λ_{eff}. The Pe represents the relative scale of convection and diffusion in the flow motion. If $Pe > 2$ *or* $Pe < -2$, the flow is convection dominant and we can use the first order upwind scheme to account for transportiveness, which is first-order accurate. If the $Pe \in [-2,2]$, the diffusion is comparable to the convection. Thus, a central differencing scheme is used, which is second-order accurate.

The discretized form of Equations (3)–(6) are illustrated in Equations (13)–(15). To avoid redundancy, the discretized form of momentum Equation (4) in y direction is not presented, as it shares the same form as momentum Equation (13) in x direction. As shown in Equation (13)–(15), the density variables ρ in the transient terms and convection terms in the momentum and energy equations are the known ones from the previous time step. The effective viscosity μ_{eff} and effective thermal diffusive coefficient λ_{eff} are also calculated using the result from the previous time step for the purpose of convenience of linearization afterwards. By doing so, the transient terms, the effective viscosity μ_{eff} and effective thermal diffusive coefficient λ_{eff} are linearized.

$$\rho^{n-1}\frac{(u)^n-(u)^{n-1}}{\delta t}\delta x\delta y+[\rho_r^{n-1}(uu)_r^n-\rho_l^{n-1}(uu)_l^n]\delta y+[\rho_t^{n-1}(vu)_t^n-\rho_b^{n-1}(vu)_b^n]\delta x=-[(\rho RT)_r^n-(\rho RT)_l^n]\delta y+\frac{4}{3}\left[\left(\mu_{eff}^{n-1}\frac{u_R-u_P}{\delta x}\right)_r^n-\left(\mu_{eff}^{n-1}\frac{u_P-u_L}{\delta x}\right)_l^n\right]\delta y+\left[\left(\mu_{eff}^{n-1}\frac{u_T-u_P}{\delta y}\right)_t^n-\left(\mu_{eff}^{n-1}\frac{u_P-u_B}{\delta y}\right)_b^n\right]\delta x+\frac{1}{3}\left[\left(\mu_{eff}^{n-1}\frac{v_R-v_P}{\delta x}\right)_t^n-\left(\mu_{eff}^{n-1}\frac{v_P-v_L}{\delta x}\right)_b^n\right]\delta x+k_x\delta x\delta y \tag{13}$$

$$\frac{(\rho)^n-(\rho)^{n-1}}{\delta t}\delta x\delta y+[(\rho u)_r^n-(\rho u)_l^n]\delta y+[(\rho v)_t^n-(\rho v)_b^n]\delta x=0 \tag{14}$$

$$\rho^{n-1}C_v\frac{(T)^n-(T)^{n-1}}{\delta t}\delta x\delta y+[\rho_r^{n-1}C_v(uT)_r^n-\rho_l^{n-1}C_v(uT)_l^n]\delta y+[\rho_t^{n-1}C_v(vT)_t^n-\rho_b^{n-1}C_v(vT)_b^n]\delta x=\left[\left(\lambda_{eff}^{n-1}\frac{T_R-T_P}{\delta x}\right)_r^n-\left(\lambda_{eff}^{n-1}\frac{T_P-T_R}{\delta x}\right)_l^n\right]\delta y+\left[\left(\lambda_{eff}^{n-1}\frac{T_T-T_P}{\delta y}\right)_t^n-\left(\lambda_{eff}^{n-1}\frac{T_P-T_B}{\delta y}\right)_b^n\right]\delta x+S \tag{15}$$

Where the superscripts "n" means the current time step and "$n-1$" means the previous time step. The subscripts "r, l, t, b" represent the "right", "left", "top" and "bottom" boundaries of the current cell, respectively. The subscripts "P, R, L, T, B" represent the current cell and neighboring cells on the right, left, top, and bottom, respectively.

2.3.2. Linearization of discretized equations

After discretization, we only have two quadratic terms remaining in the equation system. One is the convection term $\rho^{n-1}(uu)^n$, $(\rho u)^n$ and $\rho^{n-1}C_v(uT)^n$ in momentum, continuity, and energy conservation equations, respectively. The other one is the pressure term $(\rho RT)^n$. Here we adopted the method used in Ref. [14] to tackle with second-order terms, when constructing the state-space model. They rewrote the state variables as the sum of initial value (θ^0) and an increment ($\Delta\theta$), omitted the higher-order term $\Delta\theta_1\Delta\theta_2$, and changed the independent variables of the equation from the original variables themselves to the increments of them, as Equation (16). Then the equation system was linearized.

$$\theta_1\times\theta_2=(\theta_1^0+\Delta\theta_1)\times(\theta_2^0+\Delta\theta_2)\cong\theta_1^0\times\theta_2^0+\theta_1^0\times\Delta\theta_2+\theta_2^0\times\Delta\theta_1, \tag{16}$$

Inspired by their work, we linearized all the quadratic terms and altered the independent variables for our problem. The original variables, like velocity, temperature, and density, were redefined as the initial value at the beginning of the current time step plus the increment during the current time step. Thus, for quadratic terms like the convection term, we have:

$$\text{Momentum Equation}\quad \rho^{n-1}(uu)^n\cong 2\rho^{n-1}(u^{n-1}\cdot\Delta\boldsymbol{u}^n)+\rho^{n-1}(uu)^{n-1}, \tag{17}$$

$$\text{Continuity Equation}\quad (\rho u)^n\cong[\rho^{n-1}(\Delta\boldsymbol{u})^n+u^{n-1}(\Delta\boldsymbol{\rho})^n]+(\rho u)^{n-1}, \tag{18}$$

$$\text{Energy Equation}\quad \rho^{n-1}C_v(uT)^n\cong[\rho^{n-1}C_vu^{n-1}(\Delta\boldsymbol{T})^n-\rho^{n-1}C_vT^{n-1}(\Delta\boldsymbol{u})^n]+\rho^{n-1}C_v(uT)^{n-1}. \tag{19}$$

For the pressure term, we have:

$$\text{Momentum Equation}\quad (\rho RT)^n\cong[\rho^{n-1}R(\Delta\boldsymbol{T})^n-T^{n-1}R(\Delta\boldsymbol{\rho})^n]+(\rho RT)^{n-1}. \tag{20}$$

For other terms which are originally linear, like the transient term we have:

$$\text{Momentum Equation}\quad \rho^{n-1}\frac{(u)^n-(u)^{n-1}}{\delta t}=\rho^{n-1}\frac{(\Delta\boldsymbol{u})^n}{\delta t}, \tag{21}$$

$$\text{Continuity Equation}\quad \frac{(\rho)^n-(\rho)^{n-1}}{\delta t}=\frac{(\Delta\boldsymbol{\rho})^n}{\delta t}, \tag{22}$$

$$\text{Energy Equation}\quad \rho^{n-1}C_v\frac{(T)^n-(T)^{n-1}}{\delta t}=\rho^{n-1}C_v\frac{(\Delta\boldsymbol{T})^n}{\delta t}. \tag{23}$$

For the diffusion term, we have:

$$\text{Momentum Equation}\quad \left(\mu_{eff}^{n-1}\frac{u_T-u_P}{\delta y}\right)^n=\mu_{eff}^{n-1}\left(\frac{\Delta\boldsymbol{u_T}-\Delta\boldsymbol{u_P}}{\delta y}\right)^n+\mu_{eff}^{n-1}\left(\frac{u_T-u_P}{\delta y}\right)^{n-1}, \tag{24}$$

$$\text{Energy Equation}\quad \left(\lambda_{eff}^{n-1}\frac{T_R-T_P}{\delta y}\right)^n=\lambda_{eff}^{n-1}\left(\frac{\Delta\boldsymbol{T_R}-\Delta\boldsymbol{T_P}}{\delta y}\right)^n+\lambda_{eff}^{n-1}\left(\frac{T_R-T_P}{\delta y}\right)^{n-1}. \tag{25}$$

Besides the mutual influence among state variables themselves, the boundary variables also have influences on them. Currently, the SFD solver can deal with the following boundary conditions: (1) velocity inlet, (2) pressure outlet, (3) Non-slip wall with known temperature, (4) Non-slip wall with known heat flux. The reason why we don't have outflow outlet boundary condition is that in most cases we have multiple outlets, and we don't know the flow allocation among these outlets, and we want it to be calculated based on the known boundary pressure. For the velocity inlet boundary, all the variables, including normal, tangential velocity, temperature and density, are directly implemented into Equation (13)–(15). To calculate the normal velocity on the pressure outlet, we extend the calculation domain towards the outside direction by half of the boundary cell size and solve a momentum equation of the normal outlet velocity. For the thermal boundary conditions, the known wall heat flux is directly taken into the source term of the energy conservation equation. The heat transfer coefficient on the surface with known temperature is calculated as:

$$h=\frac{\mu_{eff}}{Pr_{eff}}\frac{C_p}{\Delta x}, \tag{26}$$

where Δx is the distance between the surface and the first-layer grid [17]. All these known boundary variables are defined in the input vector, while the variables of normal velocities on the pressure outlets are in the state vector.

Eventually all the equations can be rewritten into the following form as Equation (27). Since all the parts on the right side of Equation (27) are known, the equation system can be further presented as a linearized algebraic equation system, i.e. $Ax=B$, which can be solved easily.

$$MM\times\begin{bmatrix}\Delta U\\ \Delta V\\ \Delta T\\ \Delta D\end{bmatrix}^n=II\times[\Delta(BC)]^n+RMM\times\begin{bmatrix}U\\ V\\ T\\ D\end{bmatrix}^{n-1}+RII\times[BC]^{n-1} \tag{27}$$

Where BC represents the corresponding variables on the boundary, which are defined in the boundary condition; MM, II are the coefficient matrices of the increment vector of state variables and boundary input variables for the current time step, while RMM, RII are the coefficient matrices of the initial value vector of state and boundary input variables for the current time step.

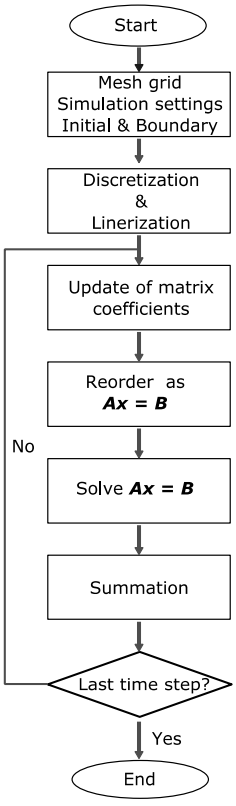


Fig. 2. Workflow of SFD.

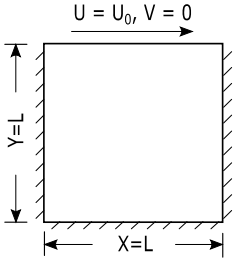


Fig. 3. Schematic of lid driven cavity case.

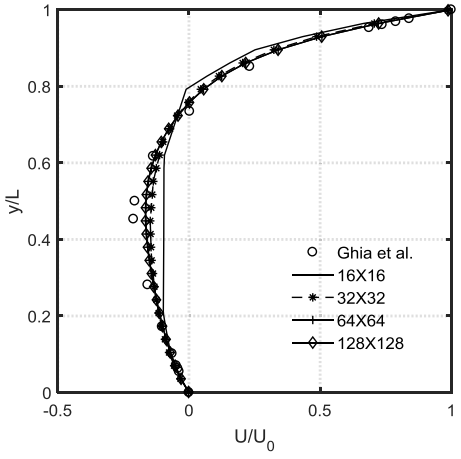
After all the above descriptions, we can generalize the calculation process of SFD in Fig. 2. First, we set up the grid, define the initial and boundary conditions and also the simulation period and time step size. Then we discretize the equation system. After that, we linearize the system, and change the independent variables to the increment variables of velocity, temperature and density. During each time step, we update the coefficients in system and input matrices and reorder the equations as $A\boldsymbol{x} = B$. When the algebraic equation system $A\boldsymbol{x} = B$ is solved, we have the increment results. By summing up the increments and the values from previous time step, we obtain the result of velocity, temperature and density for the current time step.

In current status, a SFD solver for 2-D airflow simulation has been programmed on Matlab to test feasibility. All the following case studies are run on Matlab and the computing time results represent the efficiency of the current Matlab code.

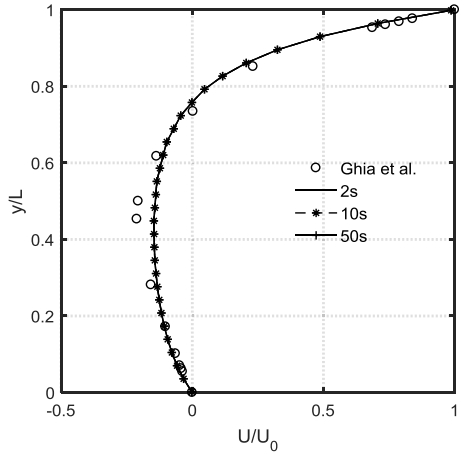
3. Performance evaluation of SFD

We chose four cases to evaluate the performance of SFD, including a lid driven cavity flow with Re number of 100 [18], a forced convection flow [19], a natural convection flow [20], and a mixed convection flow [21]. Each of them are with high quality flow and temperature data. For each case, we first ran several SFD simulations using different mesh grids with a relatively small time step size (0.5/1s) to investigate the impact of grid resolution on the SFD result. Then we chose a proper grid resolution and proceeded another several SFD simulations using different time step sizes to evaluate the sensitivity to time step size of SFD converging on a stable solution. The SFD results were compared to the original referenced data for evaluating the accuracy.

As another reference, we also conducted the corresponding FFD simulation for each case. The FFD code we used was the one implemented in the Modelica Buildings Library [22] through the research of coupling FFD with Modelica [23]. This FFD code uses the laminar viscosity, first-order time splitting method for solving the equations, and linear semi-lagrangian scheme for the convection term instead of the high-order hybrid interpolation scheme [11] for the consideration of speed performance. The grid and time step size settings for FFD were chosen from several FFD papers [7,24] to keep neutral on the FFD side. We compared the final steady results of SFD and FFD with each other using the same mesh grid and also to the referenced data. Both the SFD and FFD results were time-averaged since the original referenced data is steady. Hereby, the time step sizes used in SFD and FFD are not necessarily the same. FFD runs at a smaller time step size due to the time-splitting method and semi-lagrangian scheme it adopts, while SFD can use a relatively large time step size since it adopts the implicit scheme. The simulation time periods were set so that the flow field could



(a) Different grid resolutions



(b) Different time step sizes

Fig. 4. SFD results of U profile at x = 0.5 L (a) using different grid resolutions with 1s time step size, (b) using different time step sizes with 32 × 32 grid.

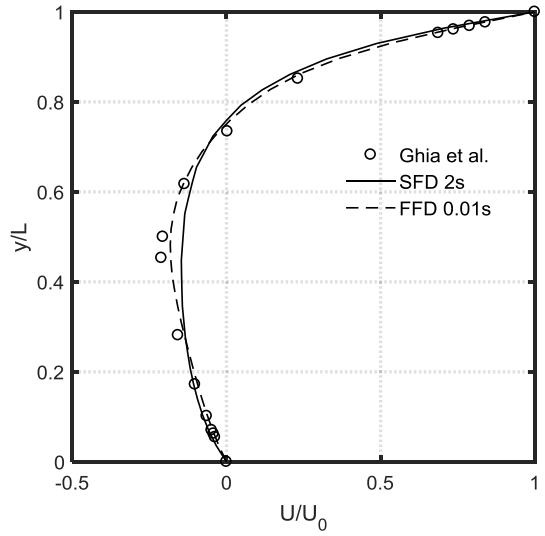


Fig. 5. Comparison of U profile at x = 0.5 L between SFD and FFD for lid driven cavity case.

fully develop from an initially still condition to a new steady state or so SFD or FFD could run for 100 steps, depending on which one was longer. The computing time ratio (CTR), i.e. simulation time period divided by computing time, was chosen as the index for evaluating the speed of SFD compared to FFD. The larger the CTR it has, the faster the solver is.

3.1. Accuracy

3.1.1. Lid driven cavity

The lid driven cavity case is similar to the air circulation of a room under a jet attached to the top. We chose the case with a Reynold number of 100. The schematic of the lid driven cavity is shown in Fig. 3.

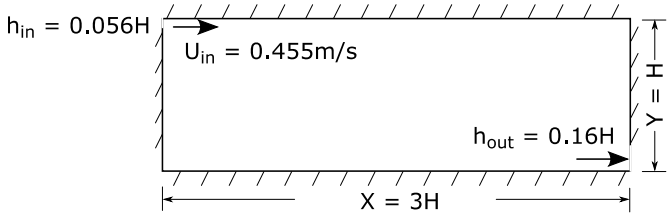


Fig. 6. Schematic of forced convection case.

We took the reference data from Ghia, Ghia and Shin [18].

We ran the SFD simulations with four different meshes (16 × 16, 32 × 32, 64 × 64, and 128 × 128) using 1s as the time step size. The result of velocity profile at the plane of x = 0.5 L is shown in Fig. 4 (a). We can see that the numerical result is approaching the reference data as the mesh goes finer. Then, we evaluated the sensitivity of SFD to the time step size by running simulations with three different time step sizes (2/10/50s) using a grid of 32 × 32. As presented in Fig. 4 (b), the profiles overlap with each other precisely. Even at the largest time step size of 50s, the prediction is still fairly stable and accurate. Thus, the time step size is not found to have a large impact on the final steady result of SFD for this case. One thing should be noticed is that when using extremely large time step size, the numerical calculation process is more like a steady state calculation, since no dynamic features can be captured in this way. However, with SFD the users can have more flexibilities to determine the time step size so that they can choose the detailed level of dynamics to be captured.

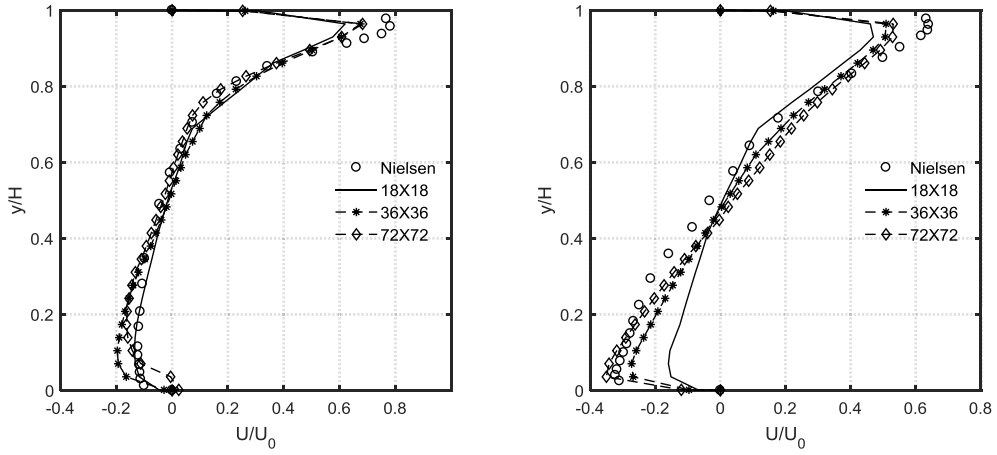
The corresponding FFD simulation was performed using the same grid of 32 × 32 with 0.01s time step size [24]. Fig. 5 compares the FFD result with SFD using 2s time step size, with which it can still capture the sufficiently detailed dynamics. FFD predicts the results slightly better than SFD. However, since SFD can use a much larger time step size, the CTR of SFD is 3.3, while the CTR of FFD is 1.8.

3.1.2. Forced convection

A forced convection case was derived from the experiment by Ref. [19]. Fig. 6 shows the schematic of it. The width of the inlet, which is located on the upper-left corner, is 0.056H. The outlet on the lower-right corner is 0.16H wide. The height of the room H is 3 m and the length of the entire room is 3H. The inlet velocity is 0.455 m/s, which leads to a Reynold number of 5000 based on the width of inlet.

We conducted the SFD simulations with three different grids (18 × 18, 36 × 36, and 72 × 72) using 1s as the time step size and the results are presented in Fig. 7. Then the grid of 36 × 36 was selected to proceed another four SFD simulations using different time step sizes (2/10/50s). It was found that although the final steady results from using the three time step sizes overlapped with each other, using time step size of 10s or 50s would introduce observable numerical oscillation during the beginning when the flow field was changing rapidly. Thus, we moved back to time step size of 5s for capturing the dynamics correctly. The corresponding steady SFD results from using different time step sizes are not presented to avoid redundancy.

Fig. 7 (a) and (b) show the U profiles predicted by SFD at the vertical plane of x = H and 2H, respectively. As the grid resolution is doubled from 18 × 18 to 36 × 36 and 72 × 72, the U velocity profile



(a) x = H (b) x = 2H

Fig. 7. SFD results of U profile (a) at x = H, (b) x = 2H.

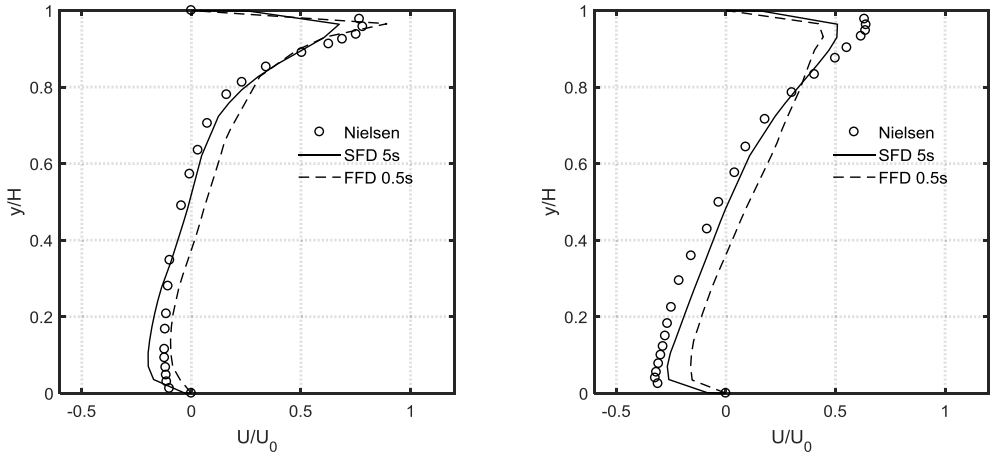


Fig. 8. Comparison of U profile at x = H (a) and x = 2H (b) between SFD and FFD for forced convection case.

at the plane of x = H starts to have a slight overshoot (36 × 36) or have a small inverse flow (72 × 72) in the lower area, as shown in Fig. 7 (a). However, at the plane of x = 2H shown in Fig. 7 (b), the U velocity profile is much closer to the reference data when finer grids are used.

Fig. 8 (a) and (b) compare the results from SFD with time step size of 5s to FFD with 0.5s using the same grid of 36 × 36 [7]. At the plane of x = H, SFD has a slight overshoot at the lower area and an underestimate in the jet zone, while FFD has a better prediction in the jet zone but a consistent underestimate for the area outside the jet zone. For the plane of x = 2H, both of them underestimate the velocity profile but it can be seen that SFD is better than FFD. As to the speed, the CTR of SFD is 5.7, while FFD has a much better speed performance with CTR of 101.5. The reason for the difference of speed performance will be systematically discussed in the next section.

The difference of the respective deviations of SFD and FFD results from the reference data is mainly because of the viscosity calculation method they adopt and the numerical viscosity they introduce. It is because of that the velocity profile is strongly impacted by the distribution of viscosity, which includes not only the one calculated by the turbulence model but also the numerical one introduced by discretization. SFD uses the zero equation to calculate the effective viscosity and generates the numerical diffusion by using a hybrid scheme to discretize the convection term. As shown in the lower part of Fig. 7 (b), using finer grid moderates the underestimation because the numerical viscosity is reduced as the mesh size decreases. FFD adopts the constant viscosity and uses the numerical viscosity generated by a linear semi-Lagrangian method to imitate the effective viscosity. As a result, when the numerical viscosity in FFD is not at the same level as the real effective viscosity, a significant deviation will occur.

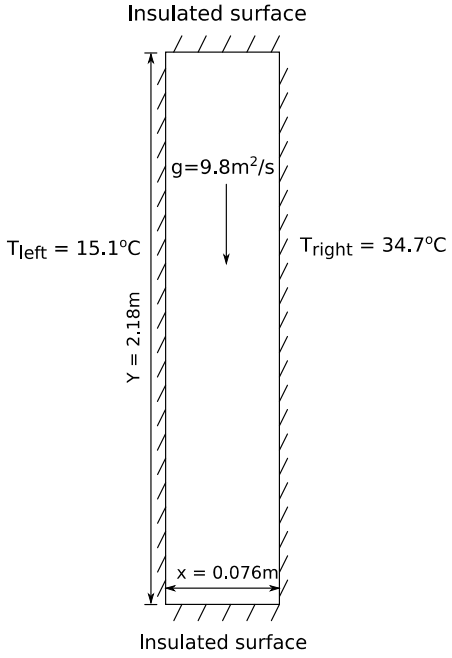


Fig. 9. Schematic of natural convection case.

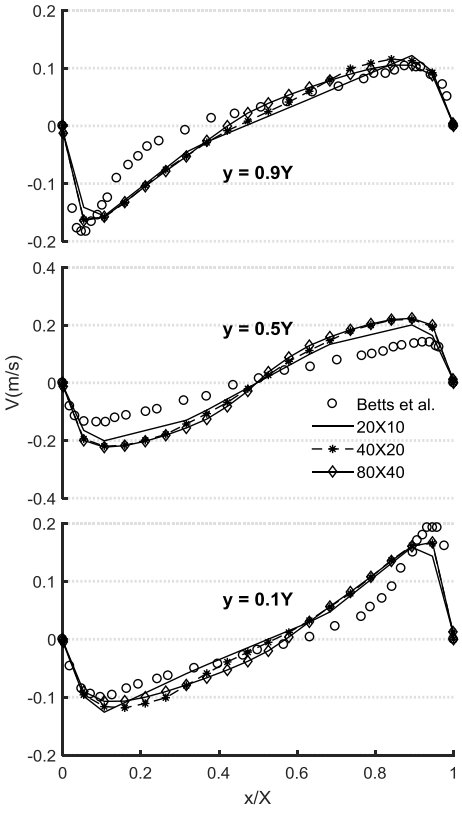
3.1.3. Natural convection

For the case of natural convection, we referenced the experiment from Betts and Bokhari [20]. As shown in Fig. 9, it's a tall cavity with a hot surface on the right and a cold on the left. The top and bottom surfaces are insulated. The gravity acceleration at Y direction is 9.8 m/s^2. Thus, the expected flow field inside should be counter-clockwise.

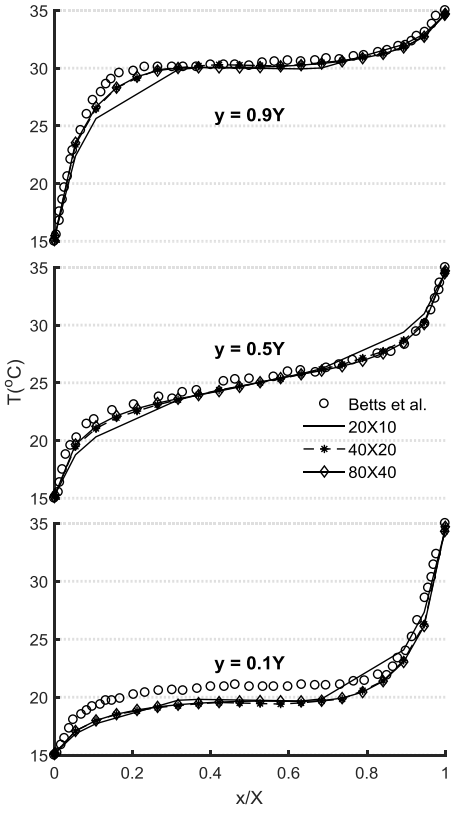
We conducted the SFD simulations with three different mesh sizes (20 × 10, 40 × 20 and 80 × 40) using a time step size of 1s. The grid of 20 × 10 was chosen to proceed another three SFD simulations using time step sizes of 2/10/50s. It was found that 50s is too large for SFD to converge on a stable solution. The SFD result from using the time step size of 2s was chosen to compare with the FFD result.

Fig. 10 presents the SFD results of the velocity and temperature profiles at three planes with the respective height of y = 0.9Y, y = 0.5Y and y = 0.1Y using different grids. It shows that the difference between the results from different grids is not significant. It is because the calculation domain of this case is relatively small and the impact from the numerical viscosity is mitigated. The velocity profile at the plane of y = 0.5Y in Fig. 10 (a) has a slight overshoot. One possible explanation for the velocity profile discrepancy is the turbulence viscosity calculated by the zero-equation model is not that accurate. On the other hand, the corresponding temperature profile at y = 0.5Y in Fig. 10 (b) is close to the referenced data points. This is mainly because the effective conductivity and the heat transfer coefficients on the surface are calculated by additional equations as Equations (8) and (26). By setting the value of Pr_{eff} as 0.9, it compensates the discrepancy of viscosity distribution. In addition, it also benefits from the fact that the height of the first grid we used for this case is suitable.

Fig. 11 compares the results from SFD and FFD using time step size of 2s and 0.05s respectively [7]. Neither of the solvers can achieve

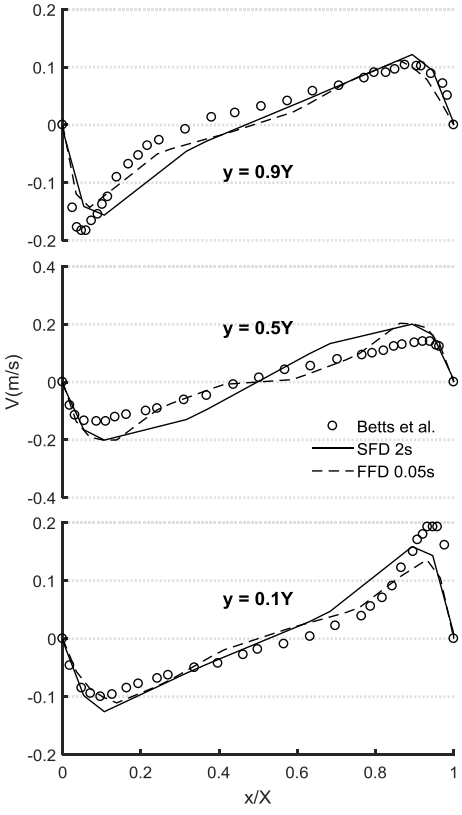


(a)

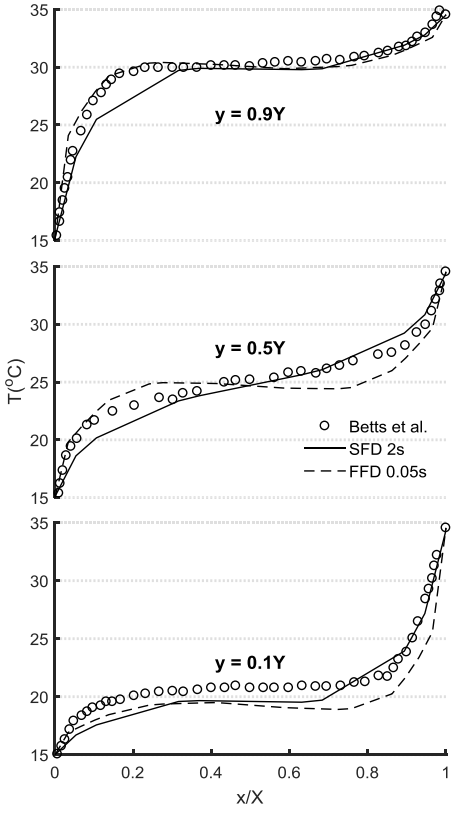


(b)

Fig. 10. SFD results of V velocity (a) and temperature profile (b) at different height using different grids for natural convection case.



(a)



(b)

Fig. 11. Comparison of V velocity (a) and temperature profiles (b) at different planes between SFD and FFD for natural convection case.

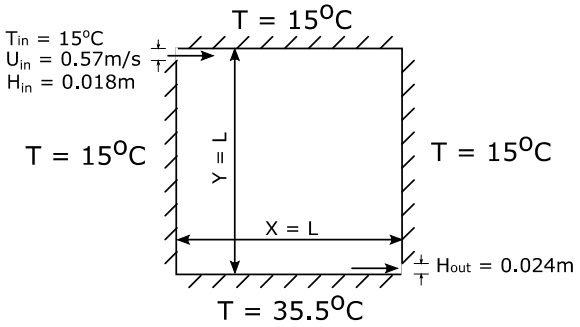


Fig. 12. Schematic of mixed convection case.

consistently better results than the other one for all predictions. FFD and SFD have a similar result of the velocity profiles at the plane of y = 0.1Y and y = 0.9Y. At the plane of y = 0.5Y, FFD only overestimates the velocity near the vertical surface, while SFD has an overall overshoot. For the temperature prediction, FFD has a larger overshoot at the y = 0.5Y, while SFD has a significant underestimation at y = 0.9Y. The difference between the velocity profiles is caused by the calculation method for turbulence viscosity as mentioned in the forced convection case. For the temperature profile difference, it's not only affected by the effective thermal conductivity but also by the heat transfer coefficient on the surface. FFD simply uses a constant surface heat transfer coefficient for the sake of speed. Thus, more consideration should be taken for choosing a suitable value for the coefficient, while SFD just uses the zero-equation model to calculate it. As to the speed performance, SFD has a CTR of 24.8, while FFD has a CTR of 58.6.

3.1.4. Mixed convection

The mixed convection flow, which is determined by both the inertial force and buoyancy force, is deemed complex for modelling. We used the data from Blay, Mergui and Niculae [21]. The schematic of the investigated room is a square room with the length and height of L = 1.04 m, as shown in Fig. 12, together with all the geometry parameters and boundary conditions.

We performed three SFD simulations with the grids of 20 × 20, 40 × 40 and 80 × 80 using a time step size of 0.5s. Then the grid of 20 × 20 was selected to proceed another two simulations using time step sizes of 0.5s. In this case the time step sizes are significantly smaller than the previous cases. It is because the initial condition is a balanced steady-still field where all the velocities are equal to zero and all the temperatures 15 °C. When the calculation starts, the inlet velocity and the temperature of bottom surface jump to 0.57 m/s and 35.5 °C, respectively. Thus, both the velocity and temperature field change rapidly and have a strong mutual impact on each other. If the time step size is too large, SFD cannot obtain a converged solution. For comparison, the FFD simulation used a grid size of 20 × 20 and a time step size of 0.02s [7].

The U velocity and temperature profile on the plane of x = 0.5X predicted by SFD using the time step size of 0.5s is presented in Fig. 13(a) and (b). When using the coarse grid of 20 × 20, SFD can capture the general patterns of the velocity profiles but has a slight underestimation because of the numerical viscosity. By using finer mesh grid, the accuracy of SFD is improved. When the grid of 80 × 80 is used, both the velocity and temperature profiles are quite close to the reference data.

Fig. 14 (a) and (b) compare the results of SFD and FFD using the time step size of 2s and 0.02s respectively. For the velocity prediction, the FFD result is closer to the reference data in most areas and only overestimates the velocity near the upper boundary. However, SFD has a much better prediction for the temperature field than FFD. It's because the surface heat transfer coefficients calculated by SFD and FFD are different. For this mixed convection case, it seems that the value of the heat transfer coefficient chosen for FFD is not large enough which leads to an insufficient heat transfer from the bottom hot surface, while SFD has a better prediction for it using the zero-equation model. As to the speed performance, SFD has a CTR of 15.5, while FFD has a CTR of 13.6. The difference between them is not so significant.

3.2. Speed

To evaluate the speed of SFD, we use the CTR of FFD as a benchmark. Both the SFD and FFD simulations were run on a same PC with the following specifications: Intel(R) Xeon(R) CPU E5-2609, 2.4 GHz; 16 GB RAM; 64-bit Win 7. Both SFD and FFD simulations for each case were conducted for 5 times to obtain an averaged CTR result. The

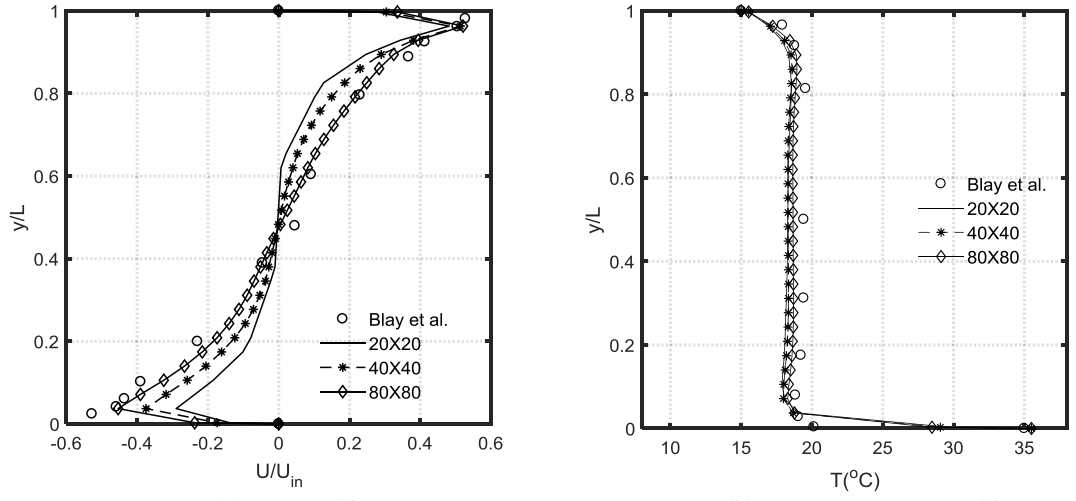


Fig. 13. SFD results of U profile (a) and temperature profile (b) at x = 0.5 L.

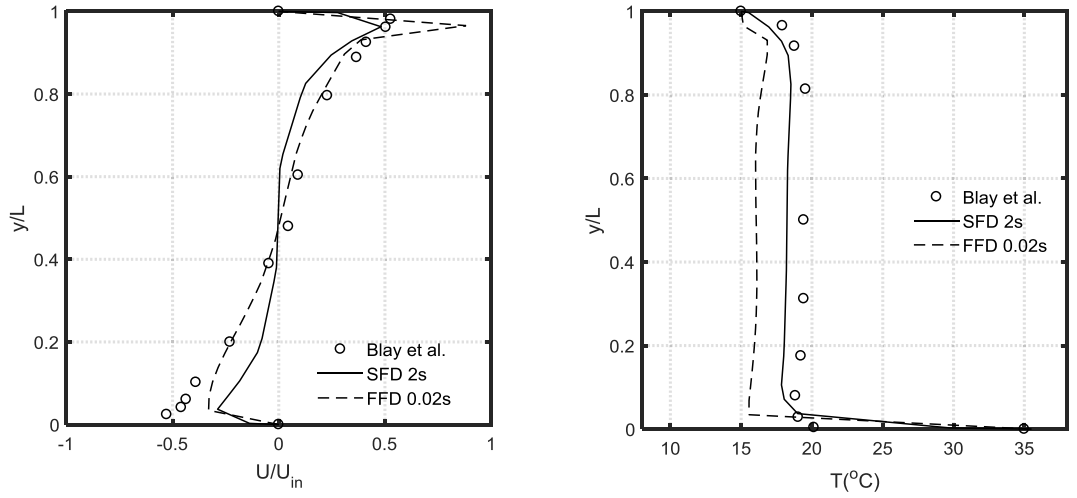


Fig. 14. Comparison of U profile (a) and temperature profile (b) at x = 0.5 L between SFD and FFD for mixed convection case.

Table 1
Speed performance comparison between SFD and FFD.

Case	Mesh	Time step size/s		CTR	
		SFD	FFD	SFD	FFD
Lid Driven Cavity	32 × 32	2	0.01	3.3	1.8
Forced Convection	36 × 36	5	0.5	5.7	101.5
Natural Convection	20 × 10	2	0.05	24.8	58.6
Mixed Convection	20 × 20	2	0.02	15.5	13.6

information of mesh, time step size and corresponding CTR results are listed in Table 1.

Generally speaking, for the studied cases the current SFD code can achieve faster-than-real-time simulation of indoor air flow. The CTR of SFD varies from 3.3 (lid driven cavity) to 24.8 (natural convection). For the three cases using the same time step size of 2s, the speed performance of SFD improves as the number of mesh cells decreases. For the forced convection case, although the number of cells (36 × 36) is larger than the one used in the lid driven cavity case (32 × 32), the CTR for forced convection is still larger than the other one, since a larger time step size is used. Compared with FFD, the speed of SFD is comparable, or for the cases of lid driven cavity and mixed convection, even slightly better than FFD in terms of CTR. It is mainly because in those cases, the time step sizes used by SFD are much larger than the ones by FFD, i.e. 200 times larger for the lid driven cavity case, and 100 times larger for mixed convection. However, for the cases of forced and natural convection where the multiples of the time step sizes between SFD and FFD are not that huge, SFD is found to be not as efficient as FFD.

The reason for the difference between the speed performances of SFD and FFD is that, during one time step, SFD solves a huge matrix simultaneously with all the equations (momentum, energy and continuity) included, while FFD solves several relatively small matrices separately. For a 2-D simulation as an example, if a grid of M × N is used, the amount of momentum equations is $(M-1)\times N$ and $(N-1)\times M$ for U velocity and V velocity respectively and the amounts of energy and continuity equations are both M × N. According to the time splitting method that FFD adopts, the procedure of FFD is shown in Fig. 15 (a) and the sizes of the matrices to be solved are also listed underneath. Instead of solving small matrices for several times, SFD solves a large-scale matrix only once in one time step, as in Fig. 15 (b). The size of the coefficient matrix is "$(M-1)\times N+(N-1)\times M+2\times M\times N+N_o$", where N_o is the amount of the pressure outlet, since SFD uses N-S equation to solve the normal velocity on the outlet boundary. As a result, if no special consideration is taken for method selection of solving the large-scale algebraic equation system, the time consumed by SFD for one time step will be much

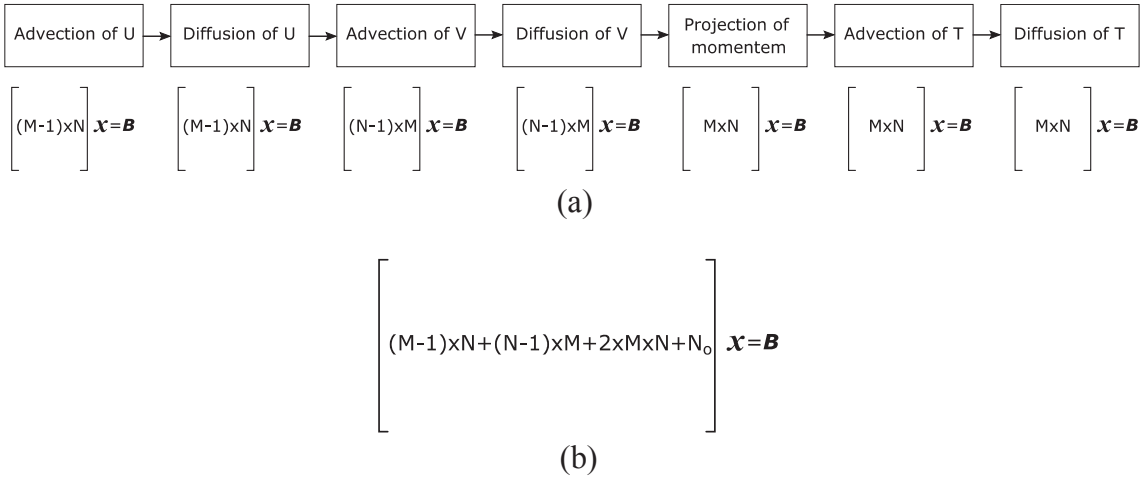


Fig. 15. Calculation procedure of FFD (a) and SFD (b) in one singe time step.

longer than FFD. Thus, if the time step size for SFD is not large enough to considerably reduce the number of time steps, the speed of SFD will not be satisfactory.

In addition, the current SFD code simply uses the backslash operation in Matlab, i.e. x = A\\B, to solve the equation system "$Ax = B$". This method doesn't take any advantage of the sparse characteristic of the coefficient matrix. However, since there are a lot of available efficient numerical algorithms for solving large-scale sparse matrices, an improvement for the speed performance of SFD can be expected.

4. Conclusion and future work

We systematically evaluate the feasibility of applying the state-space model in solving the distribution of airflow and temperature by developing the state-space fluid dynamics (SFD) method. For the studied cases, SFD can capture the general velocity and temperature profiles with acceptable discrepancies and achieve faster-than-real-time airflow simulation.

SFD converts all the governing equations into the form of a state-space model and solves all the fluid variables (velocity, temperature and density) simultaneously during each time step. The zero-equation model is used in SFD to model the turbulence to avoid introducing extra equations and save the computational cost. For the discretization of convection term, SFD adopts the hybrid scheme, which may introduce the numerical diffusion. However, from the results of the studied cases, the impact from the numerical diffusion can be mitigated with careful considerations for the grid resolution and distribution (especially the height of the first grid). For the discretization of transient term, SFD uses the first-order implicit scheme so that it can utilize a relatively large time step size. Thus, the users can have the flexibility to choose different time step sizes from a wider range to determine the detailed level of dynamics to be captured. This theoretical characteristic makes SFD suitable for applications that adopt relatively large time step sizes, such as coupled simulation with building energy simulation with the consideration of inhomogeneous indoor airflow distribution.

The future works of SFD include:

1. To improve the accuracy of SFD. The current SFD code uses hybrid scheme to discretize the convection term which will introduce the numerical viscosity and artificially smooths the variable profile. Thus, in the future, higher-order discretization schemes can be implemented in SFD to reduce the numerical viscosity.
2. To improve the speed of SFD. The current SFD code only uses the basic operation method to solve the algebraic equations and is implemented on the platform of Matlab, which has limitations on the computational speed. If SFD can be implemented using C/C++ and include some efficient numerical algorithms specifically for solving sparse matrix, an improved speed performance can be expected. Besides, SFD can be further accelerated by taking advantages of parallel computing techniques, like running on a graphics processing unit (GPU).
3. Applications of SFD. Since SFD can achieve real-time or faster-than-real-time simulation speed, it can be used for control application in co-simulation platform. By re-programming SFD in the way compliant to the Functional Mock-up Interface standard (FMI), it can be broadly adopted to perform co-simulations with many other simulation programs.

Acknowledgement

The authors would like to thank the support of the project in the National Science & Technology Pillar Program during the thirteenth Five-year Plan Period (2015BAL04B00) and the international cross-discipline doctoral joint-supervision program from Tongji University under the program number 2017XKJC-011.

References

[1] Q. Chen, Ventilation performance prediction for buildings: a method overview and recent applications, Build. Environ. 44 (4) (2009) 848–858.
[2] L.Z. Wang, W.S. Dols, Q.Y. Chen, Using CFD capabilities of CONTAM 3.0 for simulating airflow and contaminant transport in and around buildings, Hvac & R Res. 16 (6) (2010) 749–763.
[3] A.C. Megri, F. Haghighat, Zonal modeling for simulating indoor environment of buildings: review, recent developments, and applications, Hvac & R Res. 13 (6) (2007) 887–905.
[4] Z. Zhai, Q. Chen, P. Haves, J.H. Klems, On approaches to couple energy simulation and computational fluid dynamics programs, Build. Environ. 37 (8–9) (2002) 857–864.
[5] H. Wang, Z. Zhai, Application of coarse-grid computational fluid dynamics on indoor environment modeling: optimizing the trade-off between grid resolution and simulation accuracy, Hvac & R Res. 18 (5) (2012) 915–933.
[6] J. Stam, Stable fluids, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co, 1999, pp. 121–128.
[7] W. Zuo, Q. Chen, Real-time or faster-than-real-time simulation of airflow in buildings, Indoor air 19 (1) (2009) 33–44.
[8] W. Zuo, Q. Chen, Simulations of air distributions in buildings by FFD on GPU, Hvac & R Res. 16 (6) (2010) 785–798.
[9] W. Tian, T.A. Sevilla, D. Li, W. Zuo, M. Wetter, Fast and self-learning indoor airflow simulation based on in situ adaptive tabulation, J. Build. Perform. Simul. (2017) 1–14.
[10] M. Jin, W. Liu, Q. Chen, Simulating buoyancy-driven airflow in buildings by coarse-grid fast fluid dynamics, Build. Environ. 85 (2015) 144–152.
[11] W. Zuo, M. Jin, Q. Chen, Reduction of numerical diffusion in FFD model, Eng. Appl. Comput. Fluid Mech. 6 (2) (2012) 234–247.
[12] W. Liu, R. You, J. Zhang, Q. Chen, Development of a fast fluid dynamics-based adjoint method for the inverse design of indoor environments, J. Build. Perform. Simul. 10 (3) (2017) 326–343.
[13] X. Peng, A. van Paassen, Simplified Modeling of Indoor Dynamic Temperature Distributions, CLIMA 2000, Brussels, Belgium (1997).
[14] Y. Yao, K. Yang, M. Huang, L. Wang, A state-space model for dynamic response of indoor air temperature and humidity, Build. Environ. 64 (2013) 26–37.
[15] S.T. Parker, D.M. Lorenzetti, M.D. Sohn, Implementing state-space methods for multizone contaminant transport, Build. Environ. 71 (2014) 131–139.
[16] A. Sempey, C. Inard, C. Ghiaus, C. Allery, A state space model for real-time control of the temperature in indoor space-principle, calibration and results, Int. J. Vent. 6 (4) (2008) 327–336.
[17] Q. Chen, W. Xu, A zero-equation turbulence model for indoor airflow simulation, Energy Build. 28 (2) (1998) 137–144.
[18] U. Ghia, K.N. Ghia, C. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (3) (1982) 387–411.
[19] P.V. Nielsen, Specification of a Two-dimensional Test Case:(IEA), Institut for Bygningsteknik, Aalborg Universitet, 1990.
[20] P. Betts, I. Bokhari, Experiments on turbulent natural convection in an enclosed tall cavity, Int. J. Heat Fluid Flow 21 (6) (2000) 675–683.
[21] D. Blay, S. Mergui, C. Niculae, Confined Turbulent Mixed Convection in the Presence of a Horizontal Buoyant Wall Jet vol 213, ASME-PUBLICATIONS-HTD, 1993 65–65.
[22] M. Wetter, W. Zuo, T.S. Nouidui, X. Pang, Modelica buildings library, J. Build. Perform. Simul. 7 (4) (2014) 253–270.
[23] W. Zuo, M. Wetter, W. Tian, D. Li, M. Jin, Q. Chen, Coupling indoor airflow, HVAC, control and building envelope heat transfer in the Modelica Buildings library, J. Build. Perform. Simul. 9 (4) (2016) 366–381.
[24] W. Zuo, Advanced Simulations of Air Distributions in Buildings, Purdue University, 2010.