Nomenclature

,	Velocity, *m/s*		Peclet number
	Density, kg/m^3		Prandtl number
	Internal energy, *J*	CTR	Computing time ratio
	Temperature, *K*	Superscripts	
R	Gas constant for air, /	, − 1	Time step index
,	Specific heat at constant pressure and volume, /(·)	Subscripts	
	Dynamic viscosity, ·	*eff*	Effective
	Turbulent viscosity, ·	*r, l, t, b*	Location index of the cell boundaries
	Thermal conductivity, /(·)	, , ,	Location index of the cell
	Length, *m*		

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

2.1 State-Space Model

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

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

where t is time, x the state vector, u the input vector, A the system matrix, and B the input matrix. The system matrix A is used to describe the interactions between different variables in the state vector x, while the input matrix B represents the influences from the input vector u to the state vector x. Matrices A and 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):

$$x(k+1) = A(k)x(k) + B(k)u(k), \quad (2)$$

where k is the time step index, $A(k)$ and $B(k)$ are constant within the time step k and vary before entering time step $k+1$. In the context of airflow simulation, the state vector $x(k)$ represents the different variables (e.g. velocity, temperature, and density) and the input vector $u(k)$ 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 $A(k)$. The influences from the boundary conditions are described in the input matrix $B(k)$. 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.

$$\frac{(\quad)}{\quad} + \frac{(\quad)}{\quad} + \frac{(\quad)}{\quad} = -\frac{}{} + \frac{}{}\left[\quad (2\frac{}{} - \frac{2}{3}(\quad\cdot\quad))\right] + \frac{}{}\left[\quad (\frac{}{} + \frac{}{})\right] + \quad , \tag{3}$$

$$\frac{(\quad)}{\quad} + \frac{(\quad)}{\quad} + \frac{(\quad)}{\quad} = -\frac{}{} + \quad +$$

$$= 0.03874 \quad , \tag{9}$$

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

To fit Equation (3) to (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 ($\frac{(\quad)}{\quad}$, $\frac{(\quad)}{\quad}$, —, $\frac{(\quad)}{\quad}$).

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,

$$= \quad . \tag{10}$$

Then the energy variable is represented as:

$$= \quad - - = \quad , \tag{11}$$

where 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 and effective thermal conductivity are often non-linear depending on the turbulence model being used. The pressure term is also non

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 s

$$= \frac{}{}, \tag{12}$$

where

$$\times = \left({}^{0}_{1} + {}_{1} \right) \times \left({}^{0}_{2} + {}_{2} \right) \quad {}^{0}_{1} \times {}^{0}_{2} + {}^{0}_{1} \times \quad + {}^{0}_{2} \times \quad , \tag{16}$$

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:

Momentum Equation $$ {}^{-1}(\quad) \quad 2 \quad {}^{-1}(\quad {}^{-1} \cdot \Delta \quad) + \quad {}^{-1}(\quad) \quad {}^{-1}, \tag{17}$$

Continuity Equation $$ (\quad) \quad [\quad {}^{-1}(\Delta \quad) + \quad {}^{-1}(\Delta \quad) \;] + (\quad) \quad {}^{-1}, \tag{18}$$

Figure 2 Workflow of SFD

In current status, a SFD solver for 2-D airflow simulation has been programmed on Matlab to t

a

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 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 Figure 3. We took the reference data from Ghia, Ghia and Shin [18].

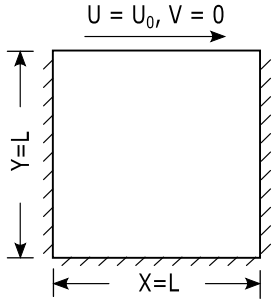

Figure 3 Schematic of lid driven cavity case

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 = 0.5L is shown in Figure 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 Figure 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.

(a) Different grid resolutions (b) Different time step sizes

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

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 forck to

finer grid moderates the underestimation because the numerical viscosity is reduced

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

(a) (b)

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

Figure 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 consistently better results than the other

a CTR of 15.5, while FFD has a CTR of 13.6. The difference between them is not so significant.

(a) U profile (b) Temperature profile

Figure 14 Comparison of U profile (a) and temperature profile (b) at x = 0.5L between

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

Figure 15 (b). The size of the coefficient matrix is "(− 1) × + (− 1) × + 2 × × + ", where 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 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.

(a)

(b)

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

In addition, the current SFD code simply uses the backslash operation in Matlab, i.e. x = A\B, to solve the equation system " = ". This method doesn't take any advantage of the sparse characteristic of the coefficient matrix. However, since there are a lot of

avail0(f)133/F1 12 Tf1 0 0 1 8516.42 286.85 Tm0 g098.66.vfic(the5l5,40110003≥2900.4 rii0 G[(15)] TJETC

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 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 implem

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-

[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, Journal of Building Performance Simulation (2017) 1-14.
[10] M. Jin, W. Liu, Q. Chen, Simulating buoyancy-driven airflow in buildings by coarse-grid fast fluid dynamics, Building and Environment 85 (2015) 144-152.
[11] W. Zuo, M. Jin, Q. Chen, Reduction of Numerical Diffusion in FFD Model, Engineering Applications of Computational Fluid Mechanics 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, Journal of Building Performance Simulation 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, Building and environment 64 (2013) 26-37.
[15] S.T. Parker, D.M. Lorenzetti, M.D. Sohn, Implementing state-space methods for multizone contaminant transport, Building and Environment 71 (2014) 131-139.
[16] A. Sempey, C. Inard, C. Ghiaus, C. Allery, A State Space Model for Real-Time Control of the aM4600912 0 612 792 reW* nBT/F1 -9(a)4(rse)/MCID 11>BDC q0.00000912 0 612 792 reW*

343.