Building and Environment 115 (2017) 291–305

Contents lists available at ScienceDirect

Building and Environment

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

# Large-eddy simulation of indoor air flow using an efficient finite-volume method

Tobias Kempe*, Andreas Hantsch

*Institute of Air Handling and Refrigeration (ILK) Dresden gGmbH, Bertolt-Brecht-Allee 20, 01309 Dresden, Germany*

A R T I C L E   I N F O

*Article history:*
Received 11 November 2016
Received in revised form
10 January 2017
Accepted 15 January 2017
Available online 31 January 2017

*Keywords:*
Indoor air
Large-eddy simulation
$\sigma$-model
Heat transfer
Immersed boundary method
Real-time simulation

A B S T R A C T

A numerical scheme for the large-eddy simulation of indoor air flows including heat transfer is presented. It is based on a finite-volume discretization of the Navier–Stokes equations for incompressible fluids on Cartesian grids. The governing equations are solved with a projection approach in combination with a direct Poisson solver for the pressure employing Fourier transformations. Complex boundaries are implemented by a variant of the immersed boundary method based on geometrical blocking of the fluid cells. The Boussinesq approximation is used to account for thermal buoyancy effects. The recently developed $\sigma$-subgrid scale model is utilized for the modeling of unresolved turbulent scales. Extensive validation of the code is carried out employing numerical and experimental reference data for laminar and turbulent flows in complex geometries including heat transfer. The computational speed is investigated for the flow in a model room including a heat source. It is shown that with the proposed numerical scheme real-time simulations of this configuration can be carried out with high accuracy at moderate numerical effort.



## 1. Introduction

The fast computation of indoor air flows is of great interest in a vast number of applications in research and development, such as the design of efficient ventilation concepts under the constraint of thermal comfort or the minimization of the energy costs required for building air conditioning. The most classical approach for computational fluid dynamics (CFD) is the discretization of the Navier–Stokes equations, e.g. by a finite-volume method (FVM). For turbulent flows, usually the Reynolds averaged Navier–Stokes (RANS) equations are solved, supplemented by a turbulence model. However, since the whole turbulent spectrum including all length and time scales of the flow is modeled in a RANS simulation, physically complex turbulence models are necessary for general flow configurations. If the flow is transitional, specially designed transition models are required [1,2]. Opposing, all spatial and temporal turbulent scales are resolved by the numerical grid in direct numerical simulations (DNS) [3]. Due to limited computing resources, in most technically relevant configurations the size of the computational grid must be larger than the smallest structures of the turbulent flow. This is justified since the truncated small-scale, low-energy, high-frequency fluctuations are of minor interest in many applications. The unavailable small-scale information, however, is crucial for the proper evolution of large-scale structures in the flow. Therefore, the effect on the resolved scales of their nonlinear interactions with the unresolved subgrid scales (SGS) has to be represented by an appropriate model. Such an approach is denoted as large-eddy simulation (LES).

The main advantage of LES is that unsteady flows including transition can be tackled. Since the energy containing large eddies are resolved, the SGS model represents only a small part of the turbulent energy spectrum. The main task of the model is to account for the under-resolved viscous dissipation and to draw sufficient energy from the flow. The superiority of LES was demonstrated by Ref. [4] where various RANS and LES models were compared for the simulation of cross ventilation in a generic building.

In recent years, the lattice Boltzmann method (LBM) has become increasingly popular [5–7]. Originating in the kinetic theory of gases [8], it calculates the flow by means of the discretized Boltzmann equation. The fluid particles are forced to move on particular trajectories rather than moving in arbitrary direction. Macroscopic quantities, such as pressure, temperature and momentum are obtained by moments of the particle distribution function. The

* Corresponding author.
*E-mail address:* tobias.kempe@ilkdresden.de (T. Kempe).

http://dx.doi.org/10.1016/j.buildenv.2017.01.019
0360-1323/

numerical algorithm consists of two steps only—propagation and collision of fluid particles. It does not need any iterations for the pressure calculation [9] and allows also complex and thermal fluid flow [10,11]. The method is fully local allowing very efficient parallel implementations on both central processing units (CPUs) and graphical processing units (GPUs) [12].

An appealing approach for real-time computations is fast fluid dynamics (FFD), originally introduced by Stam [13,14] for computer games. Zuo & Chen [15] substantially improved the algorithm by introducing turbulence modeling and applied the scheme to the air flow in buildings. Further steps were the GPU acceleration of the algorithm [16] and the extension to buoyancy driven flows [17].

The subject of this manuscript is a FVM for the computation of turbulent and buoyancy-affected flows which is significantly faster than general purpose CFD codes. The long-term objective is the numerical optimization [18–20] and online control of indoor air flows by means of LES.

## 2. Numerical method

### 2.1. Governing equations and discretization

The solver employed in the present study is the in-house code CARIBOU (CARtesian grid with Immersed BOUndaries). The governing equations are the unsteady three-dimensional Navier—-Stokes equations for Newtonian fluids [21].

$$\nabla\cdot\mathbf{u} = 0, \tag{1}$$

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{uu})\right) = \nabla\cdot\boldsymbol{\tau} + \mathbf{f}, \tag{2}$$

with the hydrodynamic stress tensor

$$\boldsymbol{\tau} = -p\mathbf{I} + \rho\nu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right), \tag{3}$$

and the equation for the temperature for fluids of constant specific heat [22].

$$\frac{\partial T}{\partial t} + \nabla\cdot(\mathbf{u}T) = \nabla\cdot(a\nabla T) + Q. \tag{4}$$

The nomenclature is as usual, with $\mathbf{u} = (u, v, w)^T$ denoting the velocity along the Cartesian coordinates $x, y, z$, while $t$ represents time, $p$ pressure, $\rho$ fluid density, $\mathbf{I}$ identity matrix, $\nu$ kinematic viscosity, $\mathbf{f}$ volumetric forces, $T$ temperature, $a$ thermal diffusivity and $Q$ the heat source.

The most general approach for the solution of the NSE is the simultaneous solution of the pressure and the velocity field in every time step. The key problem for incompressible flows is that no explicit transport equation for the pressure is available. The role of the pressure can be interpreted as an operator which projects the velocity into a divergence-free vector field [23]. An efficient approach for the numerical solution of problems involving incompressible fluids is the projection method, as proposed by Chorin [24]. The advantage of the method is that the pressure and the velocity field can be computed decoupled in a sequential manner, thus significantly easing the solution and potentially increasing the computational speed. The solution of the nonlinear system of Eqs. (1) and (2) is split into two consecutive steps. First, a non divergence-free intermediate velocity field is computed using the pressure field of the previous time level. Second, the pressure is determined at the new time level and the intermediate velocity field is projected onto a divergence-free field. This step requires the solution of an elliptic equation - the pressure Poisson equation. It is derived by applying the divergence operator to the time discrete momentum equation and using the incompressibility condition. Codes utilizing projection methods are routinely used for high-performance computing of unsteady single- and multiphase flow problems on billions of grid points and thousands of processors [25,26]. It is also used in the present paper.

The time advancement is accomplished by the explicit third-order Runge–Kutta (RK) scheme of [27]. The scheme has an excellent stability up to Courant numbers of $C \approx 1.73$, hence allowing large time steps despite its explicit nature. Furthermore, it is a so-called low-storage scheme, thus requiring only two memory registers for intermediate variables. In contrast to multipoint methods, e.g. Adams-Bashforth, RK methods can be started using only data from the initial conditions [28]. The temporal discretization scheme can be summarized as follows:

$$\begin{aligned}\frac{\tilde{\mathbf{u}} - \mathbf{u}^{k-1}}{\Delta t} = & -2\alpha^k\frac{1}{\rho}\nabla p^{k-1} - \gamma^k\left(\nabla\cdot(\mathbf{uu})^{k-1} + \nu\nabla^2\mathbf{u}^{k-1}\right) \\ & -\zeta^k\left(\nabla\cdot(\mathbf{uu})^{k-2} + \nu\nabla^2\mathbf{u}^{k-2}\right) + \mathbf{f}^{k-1},\end{aligned} \tag{5a}$$

$$\nabla^2\phi^k = \nabla\cdot\tilde{\mathbf{u}}, \tag{5b}$$

$$\mathbf{u}^k = \tilde{\mathbf{u}} - \nabla\phi^k, \tag{5c}$$

$$p^k = p^{k-1} + \frac{\phi^k}{2\alpha^k\Delta t}, \tag{5d}$$

$$\begin{aligned}\frac{T^k - T^{k-1}}{\Delta t} = & -\gamma^k\left(\nabla\cdot(\mathbf{u}T)^{k-1} + \nabla\cdot(a\nabla T)^{k-1}\right) - \zeta^k\left(\nabla\cdot(\mathbf{u}T)^{k-2}\right. \\ & \left. + \nabla\cdot(a\nabla T)^{k-2}\right) + Q^{k-1},\end{aligned} \tag{5e}$$

$$\mathbf{f}_b^k = \mathbf{g}\beta\left(T_{\text{ref}} - T^k\right), \tag{5f}$$

$$\mathbf{f}^k = \mathbf{f}_b^k + \mathbf{f}_{\text{IBM}}^k. \tag{5g}$$

The superscript $k$ in Eq. (5) denotes the RK sub-step with the corresponding coefficients $\alpha^k$, $\gamma^k$ and $\zeta^k$ [27]:

$$\begin{aligned} &\alpha_1 = 4/15, \quad &\alpha_2 = 1/15, \quad &\alpha_3 = 1/6, \\ &\gamma_1 = 8/15, \quad &\gamma_2 = 5/12, \quad &\gamma_3 = 3/4, \\ &\zeta_1 = 0, \quad &\zeta_2 = -17/60, \quad &\zeta_3 = -5/12. \end{aligned} \tag{6}$$

First, an intermediate velocity field $\tilde{\mathbf{u}}$ (5a) is computed using the pressure of the previous sub-step, $p^{k-1}$. Eq. (5a) is derived by applying the RK scheme to the momentum equation (2). The solution of the Poisson equation, (5b), to obtain the pressure correction $\phi$ and the projection step, Eq. (5c), yields the divergence-free velocity field $\mathbf{u}^k$. The pressure field subsequently is updated by Eq. (5d). After the determination of velocity and pressure, the temperature is computed using Eq. (5e), constituting the time discrete variant of Eq. (4). The total volumetric force imposed to the fluid, Eq. (5g), consists of two different contributions. First, in order to account for heat-induced convection the buoyancy force $\mathbf{f}_b$ is introduced, which is modeled by the Boussinesq approximation, Eq. (5f). Here, $T_{\text{ref}}$ is the reference temperature, $\beta$ the thermal expansion of the fluid, and $\mathbf{g}$ the gravitational acceleration. The second force term in Eq. (5g), $\mathbf{f}_{\text{IBM}}$, is required for the implementation of complex geometries using the immersed boundary method and will be explained later. Explicit time integration schemes as the present one usually require time steps $\Delta t$ smaller