WP 5 Project Deliverable D5.4

Software User Guide V1.0

Project Number	IST-2000-29266
Project Title	Virtual Real Time Fire Emergency Simulator
Deliverable Type	User Guide
Deliverable Class	Internal

Deliverable Number	D5.4
Title of Deliverable	Software User Guide V1.0
Nature of the Deliverable	User Guide
Contributing WPs	WP 5
Contractual Date of Delivery	December 31st, 2002
Actual Date of Delivery	January 16th, 2003
URL	www.virtualfires.org
Authors	Christian Redl (CD), Wilhelm Brandstätter (CD)
Contact Details	Institute for Structural Analysis / SiTu Research Univ. Prof. Dipl.-Ing. Dr. techn. Gernot Beer Lessingstrasse 25/II 8010 Graz / Austria Tel.: +43 316 8736180 Fax: +43 316 8736185 Email: gernot.beer@ifb.tu-graz.ac.at

Abstract	This report describes the physical and mathematical background and the turbulence modelling involved in the Virtualfires Software V1.0. The program structure is presented in detail and all necessary input data to run the software are specified. Finally a reference example is given.
Keywords	Tunnel Fires , Large-Eddy Simulation, Lattice Boltzmann Method

CONTENTS

1. DISCLAIMER

Christian-Doppler-Laboratory for Applied Computational Thermofluiddynamics (CD) makes no warranty, expressed or implied, to users of the Virtualfires software named ICE, and accepts no responsibility for its use. Users of ICE assume sole responsibility for determining the appropriateness of its use in any particular application; for any conclusions drawn from the results of its use; and for any actions taken or not taken as a result of analyses performed using this tool.

Users are warned that ICE is intended for use only by those competent in the fields of fluid dynamics, thermodynamics, combustion and heat transfer. It is intended only to supplement the informed judgement of the qualified user. The physical phenomena are too complex to be described in all detail using today's computer power. Therefore assumptions and simplifications are to be made and the user has to decide in which situations these are justified.

Contact information

Christian-Doppler-Laboratory for Applied Computational Thermofluiddynamics
at the University of Leoben
Franz-Josef-Straße 18
A-8700 Leoben

Phone : +43/3842 402 – 9940
Fax : +43/3842 402 – 9942

Email : sekcdfd@unileoben.ac.at

2. INTRODUCTION

The rapid growth of computing power and the corresponding maturing of Computational Fluid Dynamics (CFD) has lead to the development of CFD-based "field" models applied to fire research problems. Virtually all this work is based on the conceptual framework provided by the Reynolds averaged form of the fundamental conservation equations and the effects of turbulence are taken into account by the so-called k-ε model [1]. The use of such a model was demonstrated in the first phase of the Virtualfires project (see D5.1), where the commercial CFD solver FLUENT formed the basis of the analysis. However, CFD solvers like FLUENT have a fundamental limitation for fire applications – the averaging procedure at the root of the model equations. This causes a smoothed appearance of the results of even the most-highly resolved fire simulations. The smallest resolvable length scales are determined by the product of the local velocity and the averaging time, rather than the spatial resolution of the underlying computational grid.

Unfortunately, the evolution of large eddy structures characteristic of most fire plumes is lost with such an approach, as is the prediction of local transient events. The application of Large-Eddy Simulation (LES)-techniques is aimed at extracting greater temporal and special fidelity from simulations of fire. The basic idea behind the LES technique is that turbulent eddies that account for the fluctuation of the velocity field and mixing are large enough to be calculated with reasonable accuracy from the equations of fluid dynamics. Using LES-techniques in this way more realistic impressions of fires can be obtained.

In the first part of this report the underlying physical and mathematical framework, i.e. the fundamental conservation equations which describe the conservation of mass, momentum, energy and species, are described. Thereafter various turbulence models are discussed and the LES-technique is introduced. Then the Lattice Boltzmann (LB) method, which is used for the numerical solution of the corresponding differential equations, is presented. Compared to conventional CFD methods as mentioned above, the LB method is attractive for various reasons in context with fire simulations. As an explicit numerical scheme it does not require iterations. Secondly code parallelisation is extremely simply. Last but not least, as already discussed in the Virtualfires proposal, it shows potential for real time fire simulations.

The second part of the report is devoted to the documentation of the Virtualfires Software V1.0 named ICE V1.0. This version according to the work programme enables the user to simulate turbulent flame spread in tunnels containing arbitrarily shaped "objects" (cars, trucks and trains). It permits the computation and visualisation of the spatial and time-dependent smoke spread, temperature distribution, dynamic pressure variations, velocity field, fresh air/exhaust product stratification, etc. during the event of a hazard.

The structure of the Virtualfires software V1.0 is described in detail and all necessary files to run the program are specified. Furthermore the coupling to the Virtual Reality (VR) environments using the communication software CoVise is presented. Finally a reference example for simulating a tunnel fire is given.

3. GOVERNING EQUATIONS

The behaviour of continuum (both solid and fluid) is governed by the so-called transport equations based on the following basic laws of physics expressing balance (conservation) of:

- *mass,*
- *momentum (Newton's second law) and*
- *energy (First law of thermodynamics)*

The preferred way of writing equations expressing these laws is in differential form valid for an arbitrary point within the continuum:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j\right) = 0 \qquad \text{(Equ. 1)}$$

$$\frac{\partial \rho c_l}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j c_l\right) = \frac{\partial}{\partial x_j}\left[D_l \frac{\partial c_l}{\partial x_j}\right] + S_{c_l} \qquad \text{(Equ. 2)}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j u_i\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu \frac{\partial u_m}{\partial x_m}\delta_{ij}\right] + \rho g_i \qquad \text{(Equ. 3)}$$

$$\frac{\partial \rho h}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j h\right) = \frac{\partial}{\partial x_j}\left[\frac{\lambda}{C_P}\frac{\partial h}{\partial x_j}\right] + \frac{\partial p}{\partial t} \qquad \text{(Equ. 4)}$$

u_i is the velocity in direction x_i; ρ is density; p is pressure; g_i is the component of the gravitational acceleration vector in direction x_i; μ, D_l and λ are viscosity, diffusion coefficient of species l and heat conductivity respectively. S_{c_l} is the species source or sink.

In the case when the continuum is a mixture of various species:

- *which are mixed at the molecular level*
- *which share the same velocity, pressure and temperature fields, and*
- *in which the mass transfer between phases takes place by convection and diffusion*

the balance equation for the lth species concentration c_l or mass fraction defined as the ratio of the mass of the lth species m_l to the mass of the mixture m at a point

$$c_l = \frac{m_l}{m} \qquad \text{(Equ. 5)}$$

has to be solved. Since from the definition of mass fractions (Equ. 5) follows, that

$$\sum_{i=0}^{N} c_i = 1 \qquad \text{(Equ. 6)}$$

it is not necessary to solve transport equations (Equ. 2) for all the species. The species labelled by 0 is called *background* or *carrier fluid* and transport equations (Equ. 2) are solved for the mass fractions of the *additional species* (labelled from 1 to N).

Finally the total enthalpy is defined by:

$$h = C_p T + \frac{1}{2} u_i^2 \qquad \text{(Equ. 7)}$$

In this equation C_p represents the specific heat at constant pressure and T is the temperature.

Apart from the assumption, that the dissipation of kinetic energy into heat can be ignored for low Mach number flows, the above equations contain no approximations.

3.1 TURBULENCE MODELLING

Most engineering fluid flows are in a particular state of continuous instability called turbulence and can be said to be steady on an average basis only, since small scale, high frequency fluctuations of all the hydrodynamic variables in both space and time are always present. A flow exhibiting these macroscopic fluctuations is called turbulent flow.

Generally the degree of turbulence of a given flow is expressed by the Reynolds number (see Appendix A), e.g. for a pipe flow a Reynolds number higher than 2000 indicates a turbulent flow.

Assuming air as the ambient fluid and the tunnel diameter as a characteristic length one can suppose turbulent flows in case of tunnel fires.

The turbulent flow is well described by the Navier-Stokes equations. However their numerical solution requires a mesh with spacing smaller than the length scale of the smallest eddies and time steps smaller than the smallest time scale of turbulent fluctuations. Except for low Reynolds number flows in simple geometries this is not possible to achieve with today's computing power.

In a Large Eddy Simulation (LES) only the largest unsteady motions are resolved and all the smaller scales are modelled. This modelling can be done at a local basis for the Lattice BGK method using a Smagorinsky subgrid scale model. Details of LES can be found in the books by McComb [2], Pope [3], Wilcox [1] and Orlandi [4].

3.2 REYNOLD AVERAGED NAVIER-STOKES

The RANS equations are obtained by using a statistical description of turbulent motion, formulated in terms of averaged quantities.

One such description uses the Reynolds averaging, whereby each dependent variable is expressed as the sum of its mean, or time-averaged value $\overline{\phi}$, and fluctuating component ϕ'.

$$\phi = \overline{\phi} + \phi', \qquad \text{(Equ. 8)}$$

where

$$\overline{\phi}(\mathbf{r},\mathbf{t}) = \frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \phi(\mathbf{r},\mathbf{t}+\xi)\mathbf{d}\xi, \qquad \text{(Equ. 9)}$$

and the time interval τ is large enough with respect to the time scale λ_t of the turbulent fluctuations, but small with respect to the scale of other time dependent effects (see Fig. 1). This practice is termed "ensemble-averaging".

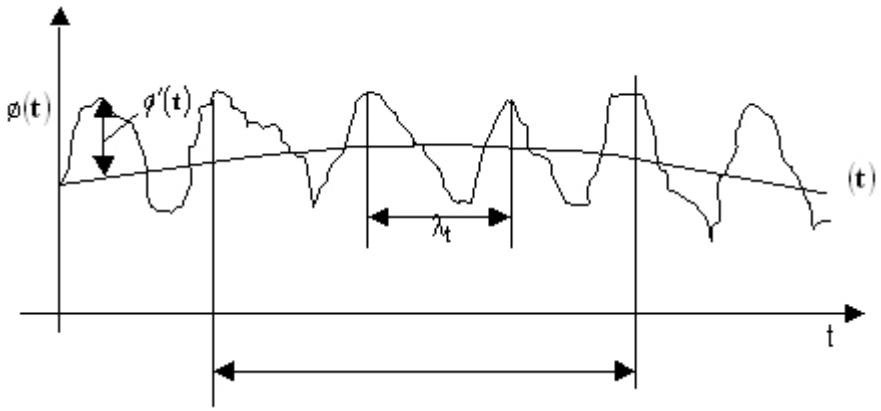

Figure 1: *Ensemble averaging procedure*

Applied to the governing equations of Section 3.1; the following equations for mass, energy and momentum balance in turbulent flows are obtained:

$$\frac{\partial\overline{\rho}}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{\rho}\,\overline{u}_j\right) = 0 \qquad \text{(Equ. 10)}$$

$$\frac{\partial\overline{\rho}\,\overline{c}_l}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho\overline{u}_j\overline{c}_l\right) = \frac{\partial}{\partial x_j}\left[D_l\frac{\partial\overline{c}_l}{\partial x_j} - \overline{\rho u_j' c_l'}\right] + \overline{S_{c_l}} \qquad \text{(Equ. 11)}$$

$$\frac{\partial\overline{\rho}\,\overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{\rho}\,\overline{u}_j\overline{u}_i\right) = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right) - \overline{\rho u_j' u_i'} - \frac{2}{3}\mu\frac{\partial\overline{u}_m}{\partial x_m}\delta_{ij}\right] + \rho g_i \qquad \text{(Equ. 12)}$$

$$\frac{\partial\overline{\rho}\,\overline{h}}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{\rho}\,\overline{u}_j\overline{h}\right) = \frac{\partial}{\partial x_j}\left[\frac{\lambda}{C_p}\frac{\partial\overline{h}}{\partial x_j} - \overline{\rho u_j' h'}\right] + \frac{\partial\overline{p}}{\partial t} \qquad \text{(Equ. 13)}$$

It is evident by comparing (Equ. 1) to (Equ. 4) with the above that the averaging procedure has produced equations where all instantaneous variables are replaced by their ensemble-averaged counterpart. Additionally new terms containing products of fluctuating quantities (so called "correlations") have appeared. In what follows the averaging signs (overbars) are retained only for the correlations and all dependent variables are considered as averaged values.

The averaging procedure produced a set of new unknowns in the momentum and energy conservation equations, respectively:

Turbulent mass flux

$$\mathbf{q}_c^t = -\overline{\rho \mathbf{c}_l' \mathbf{u}_j'} \qquad \text{(Equ. 14)}$$

Turbulent momentum flux (Reynolds stress)

$$\mathbf{T}^t = -\overline{\rho \mathbf{u}_j' \mathbf{u}_i'} \qquad \text{(Equ. 15)}$$

Turbulent heat flux

$$\mathbf{q}_h^t = -\overline{\rho \mathbf{u}_j' \mathbf{h}'} \qquad \text{(Equ. 16)}$$

Since these quantities are unknown, the averaged equations are accompanied by the so-called *turbulence models*, which provide these unknowns by expressing the correlations of the fluctuations in terms of the mean quantities. To do so, one has to rely on experimental data and knowledge obtained from DNS. No single model can be expected to reproduce well the effects of turbulence on the mean flow in all practical applications.

The most popular turbulence models are *eddy-viscosity models*, which postulate an analogy between the turbulent and viscous diffusion (Boussinesq eddy-viscosity hypothesis) and model the effects of turbulence by introducing turbulent diffusivity and viscosity coefficients:

$$\begin{aligned}
\mathbf{q}_c^t &\approx \rho \mathbf{D}_{l,t} \frac{\partial \overline{\mathbf{c}}_l}{\partial \mathbf{x}_j}, \\
\mathbf{T}^t &\approx 2\mu_t \left(\frac{\partial \overline{\mathbf{u}}_i}{\partial \mathbf{x}_j} + \frac{\partial \overline{\mathbf{u}}_j}{\partial \mathbf{x}_i} \right) - \frac{2}{3} \left(\mu_t \frac{\partial \overline{\mathbf{u}}_m}{\partial \mathbf{x}_m} + \rho \mathbf{k} \right) \delta_{ij}, \\
\mathbf{q}_h^t &\approx \lambda_t \frac{\partial \mathbf{T}}{\partial \mathbf{x}_j},
\end{aligned} \qquad \text{(Equ. 17)}$$

where μ_t represents the turbulent viscosity and $\mathbf{k}$ stands for the turbulent kinetic energy. These coefficients are non-linear functions of the flow parameters and usually vary several orders of magnitude within the flow region.

3.3 FILTERED NAVIER-STOKES EQUATIONS

Contrary to Reynolds averaging in conventional CFD simulations the Navier-Stokes equations are required in filtered form to be used with LES.

The filter operation can be done either in spectral space (components greater than a given cut off frequency are suppressed) or in physical space (weighted averaging in a given volume).

The filtering can be described mathematically by

$$\overline{\mathbf{u}}_i(\mathbf{x},\mathbf{t}) = \int\int\int \mathbf{G}(\mathbf{x}-\xi,\Delta)\mathbf{u}_i(\xi,\mathbf{t})\mathbf{d}^3\xi \qquad \text{(Equ. 18)}$$

For the application with LBGK models the Navier-Stokes equations are filtered in physical space using a volume-average box filter .

$$\mathbf{G}(\mathbf{x}-\xi,\Delta) = \begin{cases} \frac{1}{\Delta^3}, & |\mathbf{x}_i - \xi_i| < \frac{\Delta \mathbf{x}_i}{2} \\ 0, & \textbf{otherwise} \end{cases} \qquad \text{(Equ. 19)}$$

In what follows the filtering operation is denoted by $\overline{\mathbf{u}_i} = \mathbf{G} * \mathbf{u}_i$.

Filtering the instantaneous balance equations leads to equations formally similar to the Reynolds averaged balance equations.

The instantaneous velocity can be decomposed into a resolvable filtered velocity and a sub grid scale velocity:

$$\mathbf{u}_i = \overline{\mathbf{u}}_i + \mathbf{u}'_i \qquad \text{(Equ. 20)}$$

Using the definitions above the incompressible continuity and Navier-Stokes equations can be written as:

$$\frac{\partial \overline{\mathbf{u}}_i}{\partial \mathbf{x}_i} = 0$$

$$\frac{\partial \overline{\mathbf{u}}_i}{\partial \mathbf{x}_i} + \frac{\partial}{\partial \mathbf{x}_j}(\overline{\mathbf{u}_i \mathbf{u}_j}) = -\frac{1}{\rho}\frac{\partial \overline{\mathbf{p}}}{\partial \mathbf{x}_i} + \nu \frac{\partial^2 \overline{\mathbf{u}}_i}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \qquad \text{(Equ. 21)}$$

The filtered non-linear term can be written as

$$\overline{\mathbf{u}_i \mathbf{u}_j} = \mathbf{G} * (\mathbf{u}_i \mathbf{u}_j) = \mathbf{G} * (\overline{\mathbf{u}}_i \overline{\mathbf{u}}_j + \mathbf{u}'_i \overline{\mathbf{u}}_j + \overline{\mathbf{u}}_i \mathbf{u}'_j + \mathbf{u}'_i \mathbf{u}'_j) \qquad \text{(Equ. 22)}$$

The first term on the right hand side contains only explicit scales and the filter function and can be written as:

$$\mathbf{G} * (\overline{\mathbf{u}}_i \overline{\mathbf{u}}_j) = \overline{\mathbf{u}}_i \overline{\mathbf{u}}_j + \mathbf{L}_{ij} \qquad \text{(Equ. 23)}$$

$\mathbf{L}_{ij}$ is usually referred to as Leonard stresses. There is no fundamental closure problem to calculate the above term, but if a finite-difference scheme of second order (as the LBGK model described in chapter 5) is used the Leonard stresses are in the same order as the truncation error. Therefore they are implicitly accounted for.

The remaining terms containing sub grid scales can be lumped together as

$$\mathbf{G}*(\mathbf{u}_i'\overline{\mathbf{u}}_j+\overline{\mathbf{u}}_i\mathbf{u}_j'+\mathbf{u}_i'\mathbf{u}_j')=\overline{\overline{\mathbf{u}}_i\mathbf{u}_j'}+\overline{\mathbf{u}_i'\overline{\mathbf{u}}_j}+\overline{\mathbf{u}_i'\mathbf{u}_j'}=\mathbf{C}_{ij}+\mathbf{R}_{ij} \quad \text{(Equ. 24)}$$

where $\mathbf{C}_{ij}=\overline{\overline{\mathbf{u}}_i\mathbf{u}_j'}+\overline{\mathbf{u}_i'\overline{\mathbf{u}}_j}$ are the cross-term stresses and $\mathbf{R}_{ij}=\overline{\mathbf{u}_i'\mathbf{u}_j'}$ is the Reynolds stress tensor. As mentioned above the filtered equations are somewhere similar to the Reynolds averaged form but not identical.

Neglecting the Leonard stress the filtered Navier-Stokes equations can be rearranged

$$\frac{\partial\overline{\mathbf{u}}_i}{\partial\mathbf{x}_i}+\frac{\partial}{\partial\mathbf{x}_j}(\overline{\mathbf{u}_i\mathbf{u}_j})=-\frac{1}{\rho}\frac{\partial\overline{\mathbf{p}}}{\partial\mathbf{x}_i}+\nu\frac{\partial^2\overline{\mathbf{u}}_i}{\partial\mathbf{x}_i\partial\mathbf{x}_j}+\frac{\partial\tau_{ij}}{\partial\mathbf{x}_j} \quad \text{(Equ. 25)}$$

where τ_{ij} contains the cross-term and Reynolds stress.

3.4 LARGE EDDY SIMULATION

The basic idea of a Large Eddy Simulation (LES) is only to resolve the largest unsteady turbulent motions. The role of the small eddies is limited to give the satisfactory role of dissipation that it is concentrated at small scales. The large energy carrying length scales of turbulence are very problem dependent, whereas the small scales are assumed to be more universal.

Referring to the filtered equations given above the modelling effort of a LES is taken on the sum of the cross-term and Reynolds stress. The subgrid scale model most often used is based on the Eddy viscosity concept and was derived by Smagorinsky in 1963.

Using the Boussinesq approximation τ_{ij} can be expressed as

$$\tau_{ij}=-2\nu_t\overline{\mathbf{S}}_{ij} \quad \text{(Equ. 26)}$$

Assuming the eddy viscosity to be proportional to a characteristic length and velocity of the small scales, the expression for υ_t is given by

$$\upsilon_t=\mathbf{l}^2\sqrt{2\overline{\mathbf{S}}_{ij}\overline{\mathbf{S}}_{ij}} \quad \text{(Equ. 27)}$$

The resolved strain rate tensor can be obtained locally within the LBGK method as a linear combination of the non-equilibrium distribution functions and reads in continuum form as

$$\overline{\mathbf{S}}_{ij}=\frac{1}{2}\left(\frac{\partial\overline{\mathbf{u}}_i}{\partial\mathbf{x}_j}+\frac{\partial\overline{\mathbf{u}}_j}{\partial\mathbf{x}_i}\right) \quad \text{(Equ. 28)}$$

The length $\mathbf{l}$ is represents the scale of the small eddies and it must be related to the filter size. The expression best suited for a LBGK method working on uniform grids with spacing Δ is

$$\mathbf{l} = \mathbf{C}_s \Delta^2 \qquad \text{(Equ. 29)}$$

In the above equation C_s is the so-called Smagorinsky constant.

3.5 FILTER WIDTH AND SMAGORINSKY CONSTANT

Fig. 2 shows the energy spectrum for a turbulent flow in log-log scale.

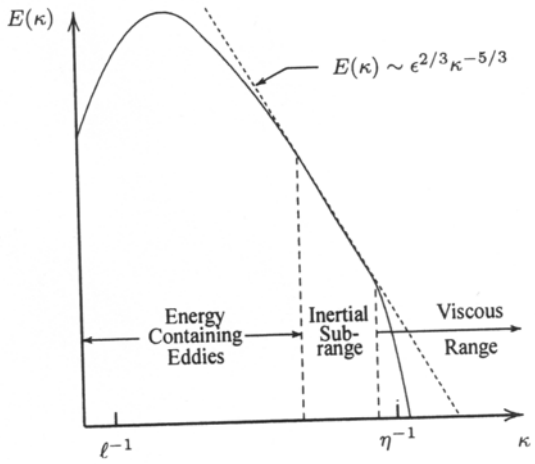

Figure 2: *Turbulent energy spectrum*

If the filter size were in the inertial sub range and sufficiently larger than the Kolmogorov viscous length scale the Smagorinsky constant would be universal and its theoretical value about 0.18 [1]. As this is rarely the case the Smagorinsky "coefficient" has to be calibrated for different kind of flows. Usually it lies in the range of 0.10 and 0.28. A lot of research is devoted to overcome the deficiency of the fixed static Smagorinsky constant. In these models the Smagorinsky constant is computed during the simulation from the flow variables.

If the grid were too coarse to resolve a substantial fraction of the turbulent kinetic energy (TKE) the method is classified as Very Large Eddy Simulation. Pope [3] claims a limiting value of 80 [%] of the resolved TKE as the criterion for a LES. As the fraction of the resolved energy is seldom estimated it is not always clear whether a simulation is a LES or a VLES.

As ICE currently does not use wall functions to account for near wall effects and the grid is very coarse it is appropriate to classify it as a VLES.

4. REPRESENTATION OF FIRE

The fire is represented as a source of smoke and energy in a pre-defined region. This model, which does not simulate the combustion process itself, is known as Volumetric Heat Source (VHS) model. It is widely used for simulating fires in enclosures.

The heat release rate can be computed by a simple equation [5, 6, 7]

$$\mathbf{Q} = \dot{\mathbf{m}}_{\mathbf{fuel}} \mathbf{H}_{\mathbf{fuel}} \eta \qquad \text{(Equ. 30)}$$

where $\dot{\mathbf{m}}_{\mathbf{fuel}}$ is the fuel consumption, $\mathbf{H}_{\mathbf{fuel}}$ is the heating value of the fuel and η is a combustion efficiency parameter.

Radiation heat transfer can simply be accounted by using the radiative fraction approach, which changes (Equ. 30) to the following form

$$\mathbf{Q} = \dot{\mathbf{m}}_{\mathbf{fuel}} \mathbf{H}_{\mathbf{fuel}} \eta (1 - \chi_{\mathbf{R}}) \qquad \text{(Equ. 31)}$$

A fixed fraction $\chi_{\mathbf{R}}$ of the total heat released is assumed to be lost to the surroundings. Thermal radiation in the participating medium is ignored. As stated by [5] this fraction lies between 0.2 and 0.4 for non-premixed flames.

Currently ICE supports only a fixed amount of smoke and energy released during the calculation. In Version 2.0 it will be possible to use pre-defined or user-defined smoke and energy emission charts like the one in Fig. 3 [8].

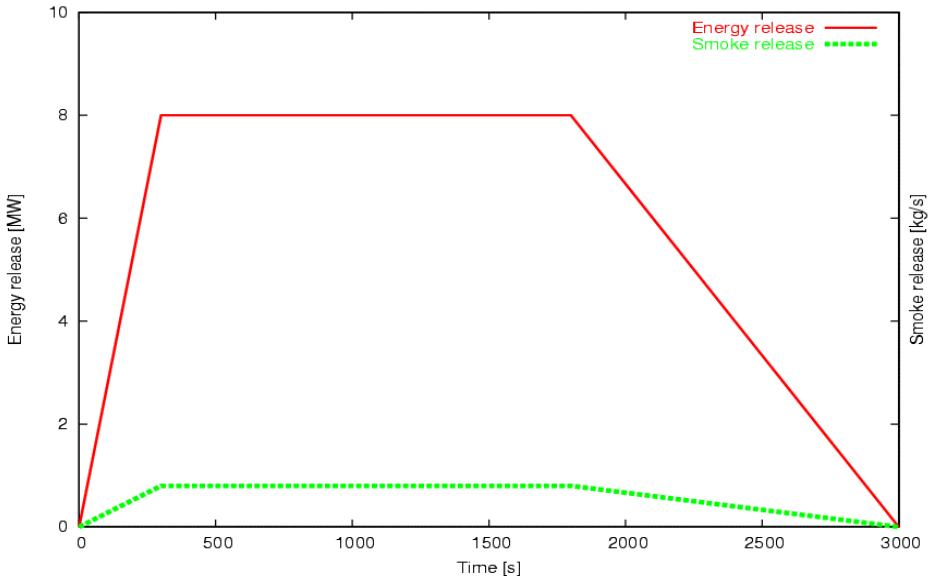

Figure 3: *Energy and smoke release rates*

The evolution of smoke is governed by the transport equation for a scalar species with source terms in pre-defined regions. The convective transport is calculated using the velocity of the carrier fluid.

If the energy equation is treated as an active scalar equation with the energy source region coinciding with the smoke source region the transport equations of energy and smoke become similar. Due to the difference in boundary conditions in general both equations would not be identical. Assuming a unity Lewis number and the walls to be adiabatic (which may be a first approximation in the early stage of fire development) the transport equation for energy and smoke are linearly dependent and only one of them has to be considered.

5. LATTICE BOLTZMANN METHOD

The Lattice Boltzmann Method (LBM) is a recent tool for simulating fluid flow problems based on kinetic theory. The Lattice Boltzmann Equation (LBE) is a finite difference approximation to the discrete Boltzmann equation using a Bhatnagar-Gross-Krook (BGK) model for the collision term.

For the derivation and theoretical background the interested reader is referred to the excellent reviews published by Luo [9] and Chen and Doolen [10]. A lot of information can also be found in the recent book by Succi [11].

In what follows the notation introduced by Qian [12] is used. The term d**X**q**Y** means a model in X-dimensional space using Y discrete lattice vectors. Currently **ICE** supports 2 dimensional model using 9 lattice vectors and 3 dimensional model using 15 or 19 lattice vectors.

The evolution equation for a particle distribution function on a lattice consisting of discrete lattice vectors is given by

$$\mathbf{f_i}(\mathrm{x}+\mathrm{e_i},\mathbf{t}+1)-\mathbf{f_i}(\mathrm{x},\mathbf{t})=\frac{1}{\tau}\left[\mathbf{f_i^{eq}}(\mathrm{x},\mathbf{t})-\mathbf{f_i}(\mathrm{x},\mathbf{t})\right]+\mathbf{Q} \qquad \text{(Equ. 32)}$$

where $\mathbf{f_i}$ is the particle distribution function related to the lattice vector $\mathbf{e_i}$, **eq** denotes the equilibrium particle distribution, τ is the relaxation parameter related to the transport coefficient, i.e. viscosity $\upsilon=\left(\frac{\tau}{3}-\frac{1}{6}\right)$, and **Q** is a general volume source term. ´

Note that the equation above is in dimensionless form. The lattice structure for the D3Q19 model used in ICE is depicted in Fig. 4 [13].

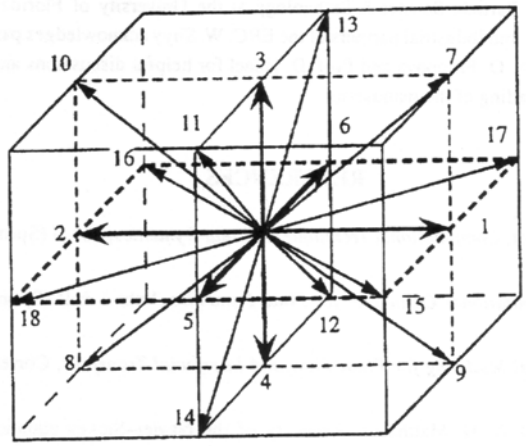

Figure 4: *D3Q19 model*

As mentioned above the BGK approximation was used for the collision term occurring in the Boltzmann equation. The use of the BGK model requires that the conservation laws are to be fulfilled on basis of the local equilibrium.

The equilibrium distribution function can be seen as a Taylor expanded Maxwellian where terms of third order and higher are neglected (low Mach number expansion).

$$\mathbf{f}_i^{eq} = \mathbf{t}_i \left\{ \mathbf{p} + \mathbf{p}_0 \left(\frac{\mathbf{e}_{i\alpha}\mathbf{u}_\alpha}{\mathbf{c}_s^2} + \frac{\mathbf{u}_\alpha \mathbf{u}_\beta}{2\mathbf{c}_s^2} \left(\frac{\mathbf{e}_{i\alpha}\mathbf{e}_{i\beta}}{\mathbf{c}_s^2} - \delta_{\alpha\beta} \right) \right) \right\} \quad \text{(Equ. 33)}$$

$\mathbf{t}_i$ are model dependent coefficients and $\mathbf{c}_s$ is the speed of sound. Their values for the d3q19-model are given in table 1.

The hydrodynamic variables can be obtained by taking the moments of the distribution functions for each cell

$$\mathbf{p} = \rho_o \mathbf{c}_s^2 \sum_i \mathbf{f}_i \quad \text{(Equ. 34)}$$

$$\mathbf{u} = \sum_i \mathbf{f}_i \mathbf{e}_i \quad \text{(Equ. 35)}$$

$\mathbf{t}_0$	$1/3$
$\mathbf{t}_{1-6}$	$1/18$
$\mathbf{t}_{7-18}$	$1/36$
$\mathbf{c}_s$	$1/\sqrt{3}$

Table 1: *Coefficients for the d3q19 model*

5.1 TRANSPORT EQUATIONS FOR SCALAR QUANTITIES IN THE LBGK FRAMEWORK

The transport equations for scalar quantities, e.g. smoke, are solved within the framework of the LBGK method by introducing an additional distribution function for each quantity.

The particle distribution functions for the scalar quantities evolve on the same lattice than the pressure distribution function for the fluid. The distribution function is given by

$$\mathbf{g}_i^{eq} = \mathbf{t}_i \phi \left(1 + \frac{\mathbf{e}_{i\alpha}\mathbf{u}_\alpha}{\mathbf{c}_s^2} + \frac{\mathbf{u}_\alpha \mathbf{u}_\beta}{2\mathbf{c}_s^2} \left(\frac{\mathbf{e}_{i\alpha}\mathbf{e}_{i\alpha}}{\mathbf{c}_s^2} - \delta_{\alpha\beta} \right) \right) \quad \text{(Equ. 36)}$$

In principle it is possible to neglect terms quadratic in velocity and to use only a lattice consisting of the Cartesian lattice vectors, i.e. only 6 lattice vectors instead of 19 in the standard model. As relatively high flow velocities are expected to occur in most cases it is not advisable to use this reduced model as it introduces a compressibility error, which increases quadratically with increasing Mach number. In addition for high Peclet numbers the numerical error can be substantially.

5.2 LIMITATIONS OF THE LATTICE BOLTZMANN METHOD

There are some serious limitations of the Lattice Boltzmann method in general and the LBGK model in specific.

Probably the most serious drawback of the Lattice Boltzmann method is the limitation to nearly incompressible flows.

For the LBGK method a strong limitation arises from the use of one single relaxation parameter. This implies that all modes decay at the same time rate, which can easily lead to numerical instabilities. Mass, momentum and heat transfer take place at the same time rate, which does not allow correct modelling of the energy equation for non-unity Prandtl numbers. This can partially be overcome by using multi relaxation time (MRT) schemes [14] in combination with finite difference methods for the energy equation [15]. These MRT models are very attractive and seem promising, unfortunately especially for turbulent flows they are almost unexplored.

6. INITIAL CONDITIONS

For initialising a simulation run values of velocity, pressure and concentration are required for each cell. These values are read from the .INI file.

The equilibrium distributions calculated from the initial values are assigned to each cell prior to the start of the simulation.

In principle it is possible to include the first order deviations from the equilibrium state into the initialisation. The spatial and temporal derivatives of the hydrodynamic variables have to be known. It is assumed that for the temporal evolution of tunnel fires the non-equilibrium deviations are not important in the initialisation process.

If the .INI file is not present the simulation is initialised with zero velocity, unit pressure and zero concentration for each cell. A warning is issued to the user.

7. BOUNDARY CONDITIONS

REMARK

Inlet and outlet conditions are assigned to areas rather to cells, i.e. a certain prescribed hydrodynamic variable is assigned to all cells within the specified boundary area. If the boundary values are required to vary over a certain area this area has to be subdivided into an appropriate number of boundary areas.

For all boundary conditions the equilibrium distributions are calculated based on the prescribed or extrapolated hydrodynamic variables.

7.1 INLET BOUNDARY CONDITION

7.1.1 VELOCITY INLET

The velocity inlet condition is used to prescribe a certain velocity at an inlet opening. The pressure is extrapolated from the corresponding neighbour cells. For stability reasons this extrapolation is of first order.

7.1.2 VELOCITY INLET ACCORDING TO LADD

Ladd [16] proposed a velocity boundary condition based on the bounce back concept. The velocity and the pressure at the inlet have to be specified by the user.

7.1.3 MASSFLUX INLET

The mass flux inlet condition is used to prescribe a certain mass flux at an inlet opening. The velocity is calculated using the fluid density according to the inlet temperature. If the calculated velocity is above a critical limit (Ma > 0.3) the velocity is set to this limit and a warning is issued to the user.

7.2 OUTLET BOUNDARY CONDITIONS

7.2.1 OUTFLOW

For the outflow condition both the pressure and the velocity are obtained by an appropriate extrapolating from the corresponding neighbour cells. For flows with higher Mach number this boundary condition is subject to severe mass conservation errors. Therefore the outflow boundary conditions should be used with caution.

7.2.2 PRESSURE OUTLET

The pressure outlet condition uses a pressure specified by the user at the outlet. Again the velocity is obtained by an extrapolation procedure from the neighbour cells. Generally the pressure outlet condition leads to a higher numerical stability than the outflow boundary condition.

7.3 FIXED WALL BOUNDARY CONDITIONS

At fixed walls the bounce back rule is applied, which means that the momentum of an incoming particle distribution is reversed at the boundary node and the particle distribution is translated back to the node it comes from in the next time step. This results in a zero velocity at the wall, which is assumed to be located at the half distance between the last active fluid node and the first wall node.

Fig. 5 depicts the “Bounce Back” rule. The red arrows show the state before and subsequent to the bounce back.

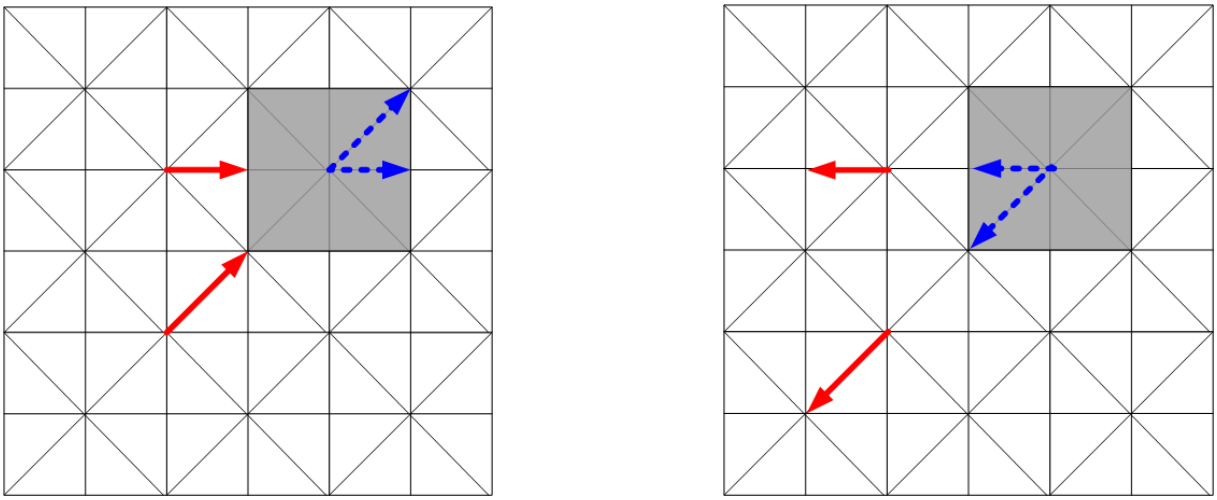

Figure 5: *Schematic representation of the bounce back rule*

For scalar quantities the fixed wall boundary conditions are accounted for by assigning the reference value specified in the .CAS file to the wall nodes.

7.4 SYMMETRY BOUNDARY

A symmetry boundary is an imaginary wall that defines a plane of flow symmetry. This means that neither flow nor scalar flux cross the boundary. However unlike a fixed wall the fluid behaviour near the symmetry boundary on one side is a mirror image of the fluid behaviour on the other side. The normal velocities are zero at symmetry boundaries.

Symmetry boundaries can be specified only for whole covering surfaces.

7.5 PERIODIC BOUNDARIES

At periodic or cyclic boundaries an exact copy of all variables of the corresponding boundary surface is assigned to the cells, i.e. the flow leaving the computational domain at one boundary re-enters the domain at the corresponding periodic boundary.

Periodic boundary conditions can be specified for the X-, Y- and Z-direction. Therefore both covering surfaces in the corresponding direction are set to periodic boundaries.

7.6 POTENTIAL PITFALLS REGARDING BOUNDARY CONDITIONS

As the fluid flow inside the computational domain is driven by the boundary conditions it is very important to specify physically meaningful boundary conditions. Inappropriate selection of boundary conditions may be the cause of failing of the simulation run.

Care has to be taken in selecting multiple outlet boundaries. It is not possible to combine an outflow condition with a constant pressure outlet as this problem is under specified.

For symmetry boundary conditions it may be important to note that symmetry of the computational domain does not imply that the flow possesses the same symmetry.

8. PROGRAM STRUCTURE OF THE SERIAL VERSION OF ICE

In what follows an overview over the program structure is given by a pseudo code. The communication between ICE and CoVise is highlighted in chapter 9.

```
Program ICE_Serial

!---Read geometry data and perform the domain decomposition
            OPEN .GEO file
            READ geometry

!---Read case file, boundary definitions and initial data
            OPEN .CAS, .BND and .INI file
            READ case and data

!---Perform the LBGK simulation
solve_equations : DO time = 1, number_of_timesteps

!---Fluid flow
                  Translation of the particle distribution functions along
                  their corresponding lattice vectors

                  Impose boundary conditions at boundary points.

                  Relaxation including turbulence modelling

!---Temperature and smoke propagation
                  Solve equations for thermal energy and smoke transport

!---At distinct times the data should be sent to the data base
                  dump_results : IF(time==write_result_time) THEN
!---The data are sent to the data base via a Unix socket
                        CALL SENT_DATA_TO_DATABASE
                  end dump_results : ENDIF(time==write_result_time)

end solve_equations END DO

end program ICE_Serial
```

Fig. 6 presents a graphical representation of the algorithm.

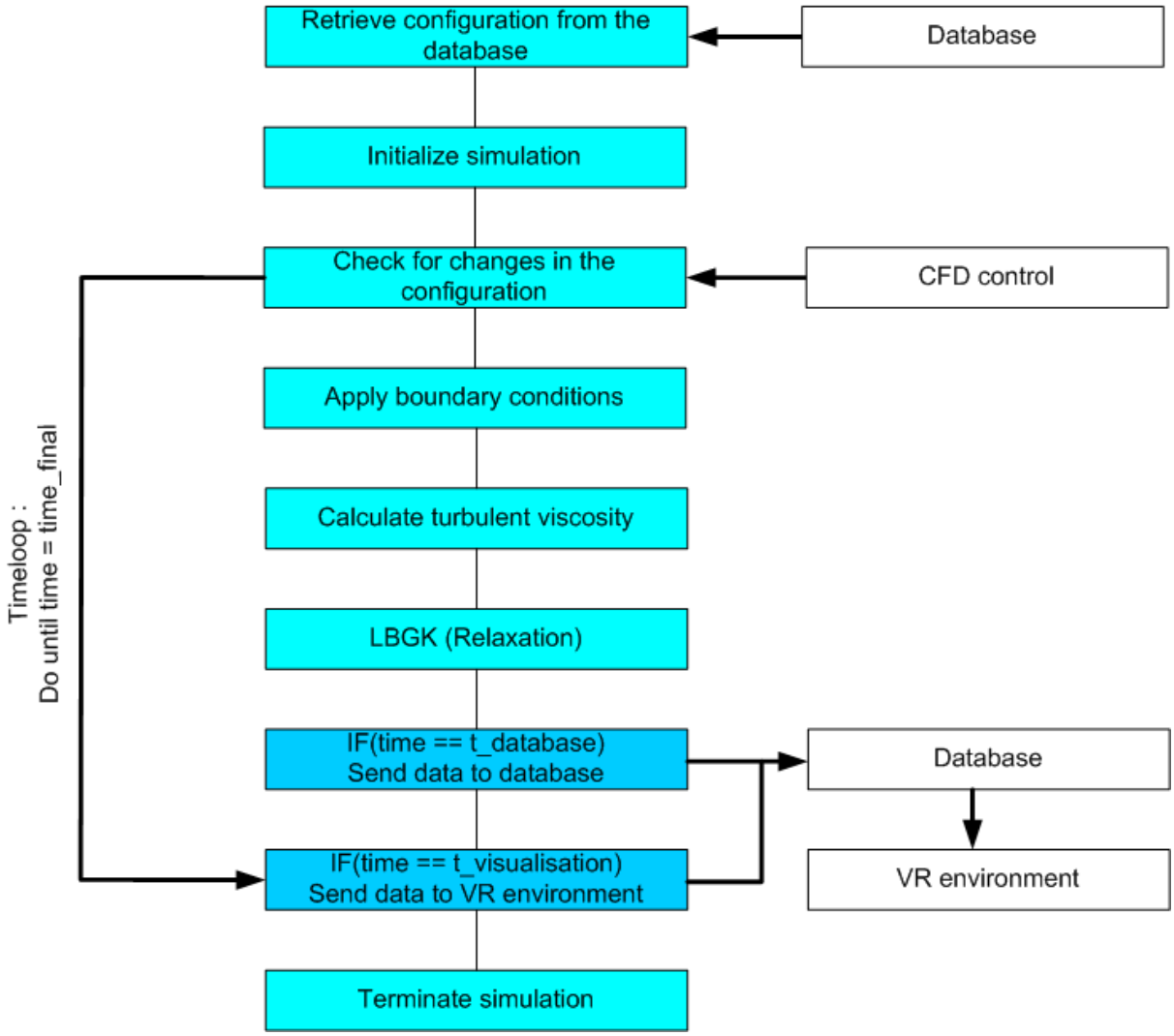

Figure 6: *Algorithm of the simulator*

9. COUPLING ICE TO VR ENVIRONMENT

This chapter is devoted to the description of the coupling between ICE and the virtual reality environment.

For the communication between ICE and the VR system (CAVE, HMD) the commercial visualisation software CoVise is used as it provides appropriate communication libraries with Fortran90 language bindings [17].

9.1 COMMUNICATION

For the communication between ICE and CoVise the capabilities of CoVise are used. CoVise provides several functions, which can be called from within the simulation program. To gain access to these functions it is necessary to include the file **coSimLib.INC** and to link the library **coSimClient.o** to the ICE main program.

To be in line with the development of ICE the syntax of **coSimLib.INC** was changed to Fortran 90 and it was included as a MODULE. The current version of **coSimLib.f90** is given below :

```
!    ++++++++++++++++++++++++++++++++++
!    +                                +
!    +    Covise SimLib include file  +
!    +                                +
!    ++++++++++++++++++++++++++++++++++

MODULE coSimLib

! --- commando
!---COVINI : Initialise COVISE
!---CONOCO : Check the connection to COVISE
      INTEGER CONOCO,COVINI,COFINI,COEXEC

! --- Data Object Creation
!---COSU1D : Send an unstructured 1D data object
!---COSU3D : Send an unstructured 3D data object
      INTEGER COSU1D,COSU3D

! --- Parameter Requests
!---COGPFL : Get a float parameter from COVISE
!---COGPSL :
!---COGPIN : Get an integer parameter from COVISE
!---COGPCH : Get a character parameter from COVISE
!---COGPBO : Get a Boolean parameter from COVISE
      INTEGER COGPFL,COGPSL,COGPIN,COGPCH,COGPBO,COGPTX

! --- binary reading/writing
      INTEGER CORECV,COSEND

! --- request verbose level
      INTEGER COVERB

! --- attach an attribute
      INTEGER COATTR
```

```
! --- parallel stuff
      INTEGER COPAIN,COPAPO,COPACM,COPAVM,COPANO

END MODULE coSimLib
```

As it can be seen from the include-File Covise currently supports only unstructured data object. Therefore the topological information of the structured grid has to be sent to Covise. The grid construction for visualisation is done by Covise internally.

The following piece of code lines out the communication between ICE and CoVise. The type definition section and definition of the lattice structure is left out as it is not important for the principal understanding.

```
!---Interaction with the VR environment using Covise
PROGRAM CovICE

!---Covise include file
      USE coSimLib

!---Define lattice structure
      DEFINE Grid

!---Initialise Covise
      INIT_STAT = COVINI()
!---Check initialisation
      IF(INIT_STAT /= 0) WRITE 'Cannot initialise Covise'; STOP

!---Check if Covise is connected
      CONNECT_STAT = CONOCO()
!---Check on connection
      IF(CONNECT_STAT /= 0) WRITE 'Covise not connected'; STOP

!---Sent grid information as 1D data
      COSU1D('grid', Number_of_cells, GRID)
      COSU1D('links', Number_of_links, LINK)

!---Ask Covise if the velocity data should be sent
      IF(COGPBO('send_velocity', Send_Vel) /= 0) THEN
            WRITE 'No answer received'
      ELSE
            IF(Sent_Vel == 0)THEN
                  IF(COSU1D('Velocity', Number_of_cells, Velocity) /=0)THEN
                        WRITE 'Lost connection to Covise'
                  ELSE
                        WRITE Number_of_cells, 'items successfully sent!'
            END
      END

!---Finalise Covise
      COFINI()

END PROGRAM CovICE
```

Of course this is only a rudimentary fragment of the implementation of communication. The grid information has to be sent only once at the beginning of the simulation, whereas the request for hydrodynamic variables must be checked every time step.

10. SPECIFICATION OF THE SIMULATION BASE NAME AND FILE NAME CONVENTIONS

10.1 ICE_BASENAME

For a simulation run all file names are constructed using a common name called ICE_BASENAME. The simulation program searches all its necessary input files in the directory it has been started.

The first file read is **ICE_INPUT**. It simply contains the ICE_BASENAME consisting of exactly six letters. The ICE_BASENAME must be the first entry in **ICE_INPUT**.

10.2 FILE NAME CONVENTIONS

All file names are based on ICE_BASENAME. In what follows the file name conventions are highlighted:

<ICE_BASENAME>.CAS	Case file
<ICE_BASENAME>.GEO	Geometry description
<ICE_BASENAME>.INI	File containing the initial values
<ICE_BASENAME>.LOG	File containing log information and error messages
<ICE_BASENAME>.RES	Restart file containing velocity, pressure and concentrations
<ICE_BASENAME>.<DAT>	Result file containing velocity and pressure in data format <DAT>
<ICE_BASENAME>_SPEC<ID>.<DAT>	Result file containing concentration of species <ID> in data format <DAT>

The result file names are built similar and are described in chapter 14.

Fig. 7 shows the file structure used by ICE.

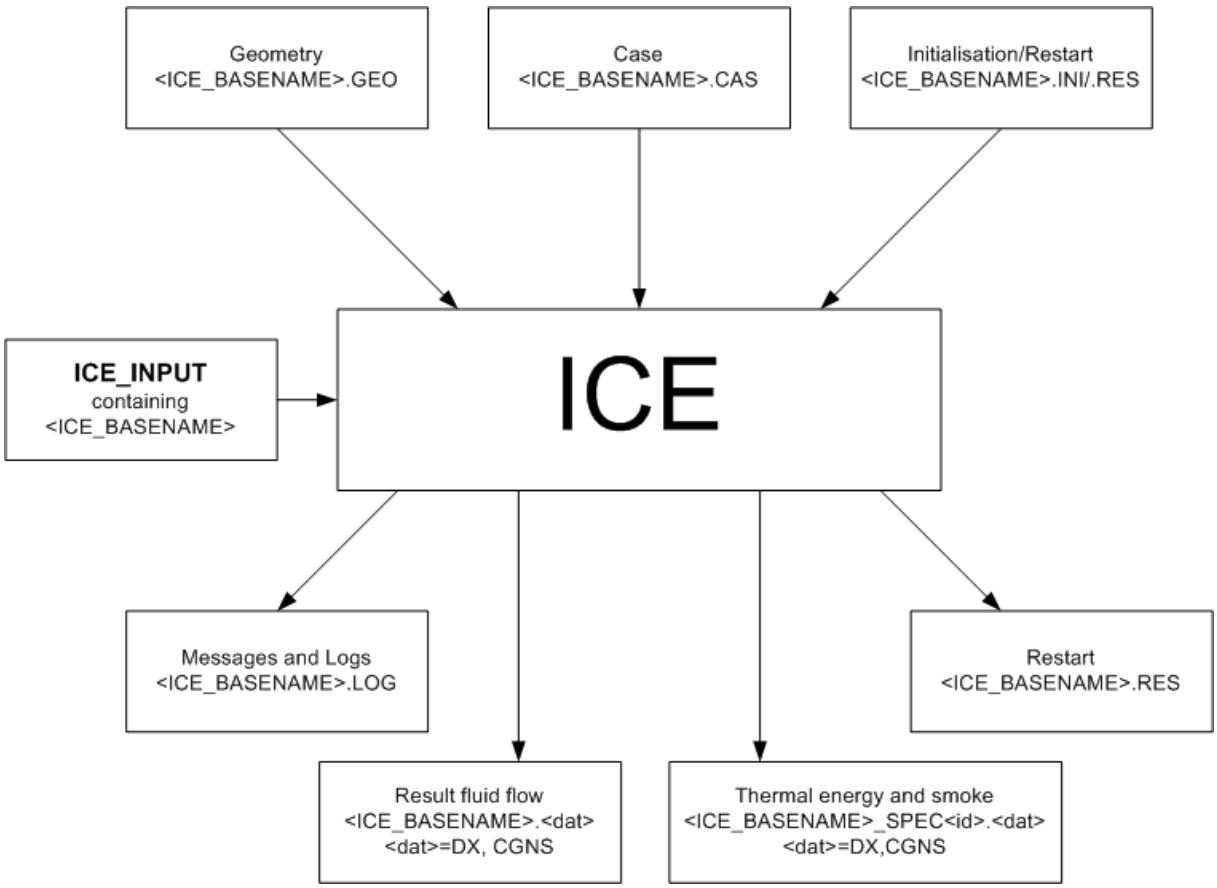

Figure 7: *File structure used by ICE*

11. SPECIFICATION OF THE CASE FILE <ICE_BASENAME>.CAS

The file <ICE_BASENAME>.CAS contains all information regarding the fluid and species properties, models, turbulence parameter and time management.

11.1 STRUCTURE

MODEL
NDIM[INT] NDIR_FLUID[INT] NDIR[SPECIES]
NCOMP
SOLVE_FLUID_FLOW[BOOLEAN]
NCOMP[INT] NSPECIES[INT]
LTURB
USE_LES[BOOLEAN]
SMAGORINSKY[FLOAT]
FLUI[id]
DENSITY[FLOAT] VISCOSITY[FLOAT]
SPEC[id]
REF_QUANTITY[FLOAT] DIFFUSIVITY[FLOAT]
GRAVITY
USE_GRAVITY[BOOLEAN]
GRAV_X[FLOAT] GRAV_Y[FLOAT] GRAV_Z[FLOAT]
TIMEM
NTIMESTEPS[INT] NCHECK[INT] NRESTART[FLOAT]
LRESTART
READ_FLOW[BOOL] READ_SPECIES[BOOL]
WRITE_SOLUTION[BOOL]
BND[id]
TYPE[INT]
VEL_X[FLOAT] VEL_Y[FLOAT] VEL_Z[FLOAT]
REF_DENSITY[FLOAT]
REF_QUANTITY_1[FLOAT] .... REF_QUANTITY_[nspecies][FLOAT]
#EOF#

Remarks : **No blank lines allowed!**

Keywords are in bold face. They may be left out in case the default values are used.

11.2 VARIABLE NAMES

MODEL
NDIM Number of spatial dimensions.
Input value : 2 3
Default input : 3

NDIR_FLUID Number of discrete lattice vectors
Input value : NDIM = 2 : 9
NDIM = 3 : 15, 19
Default : NDIR_FLUID = 19
Attention : The number of lattice vectors must correspond to the spatial dimensions.

NDIR_SPECIES Number of discrete lattice vectors for the species transport
Input value : NDIM = 2 : 4, 9
NDIM = 3 : 6, 15, 19
Default : NDIM = 3 : 19
Attention : The number of lattice vectors must correspond to the spatial dimensions.

NCOMP

SOLVE_FLUID_FLOW Specifies if the equation for the flow field is to be solve.
Input values : T F
Default : T

NCOMP Number of (fluid) components
Input values : 1
Default : 1
Currently only a single fluid is considered.

NSPECIES Number of species
Input values : 0 or positive integer
Default : 0
In principle the number of species is limited only by memory issues.

LTURB

USE_LES Use a Smagorinsky type SGS – LES model
Input values : T F
Default : F

SMAGORINSKY Smagorinsky constant
Input values : Positive float
Default : 0.18
Theoretical value is 0.18, usual values are between 0.10 and 0.30.

FLUI[id]

DENSITY Fluid density [kg/m³]
Input value : Positive float
Default : 1.0

VISCOSITY Fluid viscosity [Pa s]
Input value : Positive float
Default : 0.02

SPEC[id]

REF_CONCENTRATION — Reference concentration
Input value : Positive float
Default : 1.0

DIFFUSIVITY — Species diffusion constant
Input value : Positive float
Default : 0.02

GRAVITY

USE_GRAVITY — Specifies if gravity is to be considered in the simulation
Input values : T F
Default : F

GRAV_X
GRAV_Y
GRAV_Z — Gravity vector
Input values : Float Float Float
Default : 0.0 0.0 0.0

TIMEM

NTIMESTEPS — Number of calculated time steps
Input value : Positive integer
Default : 0

NCHECK — Interval for convergence check
Input value : Positive integer
Default : 1000

NRESTART — Interval for dumping a restart file
Input value : Positive integer
Default : 10000

LRESTART

READ_FLOW — Read the flow field from ICE_BASENAME.RES
Input value : Boolean
Default : F

READ_SPECIES — Read the species field from ICE_BASENAME_SPEC[id].RES
Input value : Boolean
Default : F

WRITE_SOLUTION — Dump a restart file ICE_BASENAME.RES
Input value : Boolean
Default : F

BND[id]

TYPE	Specifies the type of the boundary		
	Input value :	1	Velocity inlet
		2	Massflow inlet
		3	Inflow according to Ladd
		101	Outflow
		102	Pressure outlet
		401	Energy source region
	Default :	501	Inactive

12. SPECIFICATION OF THE GEOMETRY FILE <ICE_BASENAME>.GEO

12.1 STRUCTURE

DIMENSIONS
NI[INT] NJ[INT] NK[INT]
SYMMETRY
SYMMETRY_BND[INT, DIMENSION(2*NDIMENSION)]
PERIODIC
PERIODIC_BND[INT, DIMENSION(NDIMENSION)]
NBOUND
NBOUNDARY[INT] NBOUNDARYCELLS[INT]
BNDIDX
BOUNDARY_SID[INT, DIMENSION(NBOUNDARY)]
BOUNDARY_EID[INT, DIMENSION(NBOUNDARY)]
BNDCOORD
BOUNDARY_CELL[INT, DIMENSION(NBOUNDARYCELLS)]
BOUNDARY_NBC[INT, DIMENSION(NBOUNDARYCELLS)]
WALLFLAGS
CELL_ID(OBSTACLE(1)) [INT] WALLFLAG(OBSTACLE(1)) [INT]
.
.
.
CELL_ID(OBSTACLE(NOBS)) [INT] WALLFLAG(OBSTACLE(NOBS)) [INT]
NOBSTACLES
NOBST[INT] NOBST_INNER[INT]
#EOF#

Remarks : **No blank lines allowed!**

Keywords are in bold face. Except of **DIMENSION** and **#EOF#** they may be left out if not required. The order of their appearance is insignificant.

12.2 VARIABLE NAMES

DIMENSION
NI Number of cells in X-direction
NJ Number of cells in Y-direction
NK Number of cells in Z-direction

SYMMETRY

Symmetry boundaries may be specified for all enclosing surfaces of the computational domain.

SYMMETRY_BND(+X, -X, +Y, -Y, +Z, -Z)

A integer flag for each surface has to be specified. A number greater zero means that the corresponding surface is defined as symmetry boundary.

PERIODIC

Periodic boundary condition may be specified in each direction.

PERIODIC_BND(X-direction, Y-direction, Z-direction)

A integer flag for each direction has to be specified. A number greater zero means that the corresponding surfaces (WEST-EAST, NORTH-SOUTH, HIGH-LOW) are defined as symmetry boundaries.

BNDIDX

The start and end indices specifies which cells belong to a distinct boundary. Therefore it is important to keep a strict ordering of the cell co-ordinates.

BOUNDARY_SID	Boundary start index
BOUNDARY_EID	Boundary end index

BNDCOORD

As mentioned above the boundary cell co-ordinates MUST be in a specific order corresponding to the boundary start and end indices.

BOUNDARY_CELL	Cell index
BOUNDARY_NBC	Index of the neighbour cell

WALLFLAG

The cell indices are only given for occupied cells (wall cells)

CELL_ID	Cell index of the wall cell
WALLFLAG	Flag

NOBSTACLE

NOBST	Total number of obstacles
NOBST_INNER	Number of obstacles in the domain [(2,NI-1), (2,NJ-1), (2,NK-1)]

13. SPECIFICATION OF THE FILE <ICE_BASENAME>.INI

The file <ICE_BASENAME>.INI contains the initial values for all hydrodynamic variables.

13.1 FILE STRUCTURE

The structure is as follows :

PRES<id>
PRESSURE_<id>(1:NIJK)[FLOAT]
VELO<id>
VEL_X_<id>(1:NIJK)[FLOAT]
VEL_Y_<id>(1:NIJK)[FLOAT]
VEL_Z_<id>(1 :NIJK)[FLOAT]
CONC<id>
CONCENTRATION_<id>(1:NIJK)[FLOAT]

Remark: No blank lines are allowed.

Keywords are in bold face and can appear in any order. Each component or species requires its own data set within the <ICE_BASENAME>.INI file.

13.2 VARIABLE NAMES

PRES<id>

The pressure has to be specified for each component in each cell as an array of reals.

VELO<id>

The fluid velocity for each component has to be specified in each cell. For each spatial direction one array of real is required.

CONC<id>

The concentration of each species has to be specified for each cell as an array of reals.

13.3 EXAMPLE <ICE_BASENAME>.INI FILE

Example 1 : Homogenous flow field

The following file is used to initialise a homogenous flow field of 0.1 [LU/LT] and uniform pressure. The concentration field of the first species is set equal zero everywhere, whereas the concentration of species 2 is set equal 1.0 throughout the whole computational domain.

PRES1
300000*1.0
VELO1
300000*0.1
300000*0.0
300000*0.0
CONC1
300000*0.0
CONC2
300000*1.0

Example 2 : Non-homogenous flow field

The following file initialises a flow field in a 2*2*2 domain with non-homogenous values for pressure, velocity and concentrations. Note that this example does not contain very meaningful values but it should only be considered as a demonstration.

PRES1
1.002 1.002 1.003 1.001 1.003 0.998 0.998 1.000
VELO1
0.100 0.101 0.102 0.103 0.104 0.105 0.106 0.017
0.001 0.001 0.003 0.002 0.002 0.001 0.003 0.003
0.003 0.003 0.001 0.004 0.003 0.002 0.008 0.001
CONC1
8*0.0
CONC2
0.987 0.888 0.999 0.987 0.987 0.876 0.998 0.988

14. RESULT FILE FORMAT

Currently two output file formats are supported by ICE.

14.1 OPEN DATA EXPLORER

Open DX is the open source version of IBM's Visualization Data Explorer [18].

Velocity, pressure and geometry information are contained in one data file. For each scalar quantity a single data file is written.

The file name convention is as follows :

<ICE_BASENAME>.DAT for the fluid variables

<ICE_BASENAME>_SPEC<ID>.DAT for species <ID>

For each data file a corresponding header file is created. The file name convention for the header files is as follows :

<ICE_BASENAME>.general for the fluid variable data file

<ICE_BASENAME>_SPEC<ID>.general for the data file of species <ID>

14.2 CGNS

CGNS (CFD General Notation System) [19] is an effort to standardize CFD data including grid, flow solution, boundary conditions, etc.

One single CGNS file is created for the output. It contains a single zone structured grid together with the corresponding flow solution as described in the CGNS User's guide.

The file name convention for CGNS result files is as follows :

<ICE_BASENAME>.CGNS

14.3 USER DEFINED OUTPUT FILE FORMAT

It is possible to specify a user defined output file format within the subroutine ice_usr_out.f90, which has the following header specification :

```
SUBROUTINE ICE_USR_OUT(ICE_BASENAME, NIJK, NI, NJ, NK, VEL_X, VEL_Y, VEL_Z,
PRESSURE, CONCENTRATION, GRID, NCOMP, NSPEC)
```

```
!---Simulation base name
CHARACTER(LEN=*), INTENT(IN) :: ICE_BASENAME
!---Number of cells
INTEGER, INTENT(IN) :: NIJK
!---Number of cells in X-, Y- and Z-direction
INTEGER, INTENT(IN) :: NI, NJ, NK
!---Fluid velocity
REAL(HIGH), DIMENSION(:), INTENT(IN) :: VEL_X, VEL_Y, VEL_Z
!---Pressure
REAL(HIGH), DIMENSION(:), INTENT(IN) ::  PRESSURE
!---Species concentration
REAL(HIGH), DIMENSION(:), INTENT(IN) :: CONCENTRATION
!---Cell information flag
INTEGER, DIMENSION(:), INTENT(IN) :: GRID
!---Number of phases and species
INTEGER, INTENT(IN) :: NCOMP, NSPEC
```

To access the different species it is necessary to add a shift variable to the actual cell number. This shift parameter is calculated within ICE_USER_OUT by the following command :

```
!---GET CELL SHIFT
CELLSHIFT = NIJK*(ISPEC - 1)
```

where ISPEC is the actual species ID.

The actual concentration of a species in cell NCELL can be accessed by

```
CONCENTRATION(NCELL + CELLSHIFT)
```

15. DESCRIPTION OF A REFERENCE CASE

The configuration shown in Fig. 8 consists of a section of a tunnel containing two exhaust air outlets located at the ceiling. This corresponds to a usual installation in a modern full cross ventilated tunnel [20]. A certain underpressure is assigned to the extraction openings in order to model the removal of hot gases. The effects of natural ventilation are taken into account by applying a pressure difference along the tunnel axis. In reality this longitudinal velocity is induced by differences in altitude of the two tunnel portals or wind outside the tunnel.

Two heavy goods vehicles (HGV) are placed within the tunnel. One trailer is defined as fire zone and set on fire. The maximum values of the smoke and energy release curve shown in Fig. 8 are used.

The ventilation system is working at full extraction capacity. As can be seen from Fig. 9 a big fraction of hot gases are extracted by the ventilation opening in front of the burning truck. The isosurface is shown at 373 [K]. The velocity vectors in the cutting plane are overscaled to accentuate the extraction openings.

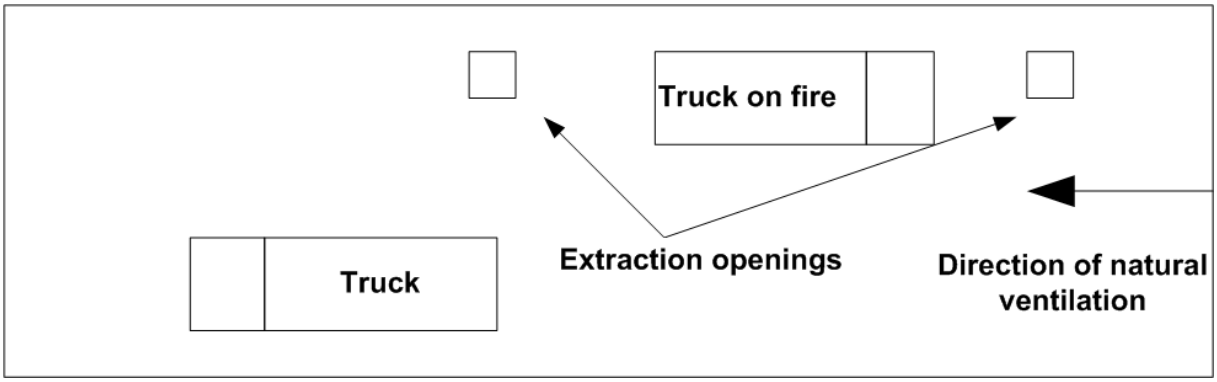

Figure 8: *Scenario*

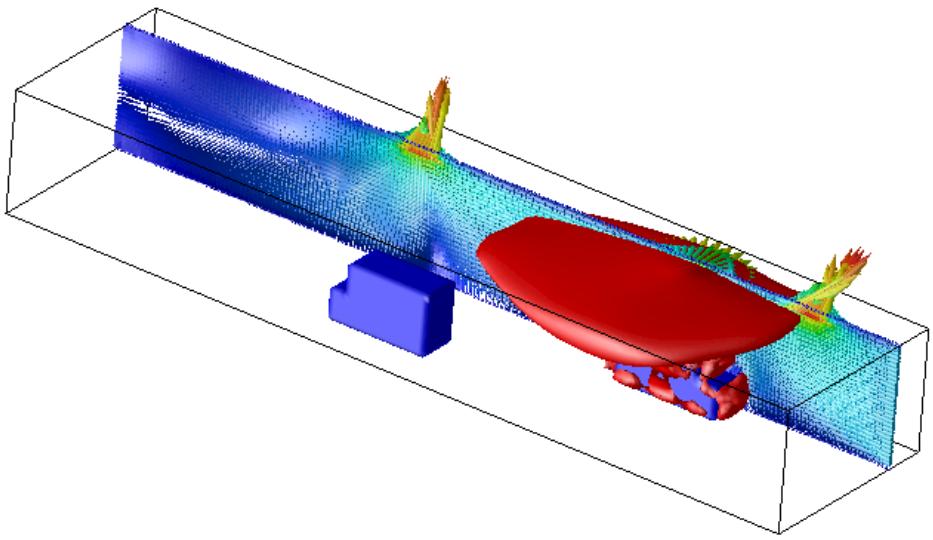

Figure 9: *Temperature isosurface (T=373 [K]) and velocity vectors*

16. FILES USED FOR THE EXAMPLE SIMULATION „TRUCKS“

16.1 ICE_INPUT

The ICE_INPUT file contains only the ICE_BASENAME:

```
Trucks
```

16.2 Trucks.CAS

The following file is the case file for the simulation of a tunnel fire. Note that currently all physical quantities are specified in dimensionless form, i.e. the velocity is given in lattice units per lattice timestep [LU/LT]. The scaling to real world dimensions can be done by using the dimensionless groups given in Appendix A.

```
# 3 spatial dimensions
# d3q19 model for both the carrier and the species
      MODEL
      3 19 19
# Solve the fluid flow equation
# 1 fluid, 1 species
      NCOMP
      T
      1 1
# Turbulence parameter
# Use LES turbulence model with Cs = 0.1
      LTURB
      t
      0.1
# Parameter for Species 1
# Reference value and diffusivity
      SPEC1
      293.0 0.01
# Fluid properties
# Reference density and viscosity
      FLUI1
      1.0 0.0075
# Consider buoyancy
# Definition of the gravity vector
      GRAVITY
      T
      0.0 0.0 0.001
# Time management data
      TIMEM
      1000 200 200
# Restart data
# LREAD LRPSC LWRITE
# Read the velocity and the pressure from the .RES file
# Read the concentration from the .INI file
# Write a .RES file at the end of the simulation
      LRESTART
      t f t
#
```

```
#Boundary specifications
# Tunnel portal
# Specifies a longitudinal velocity of u=(0.01,0.0,0.0)
# The inlet temperature is 293.0 [K]
      BND1
      1
      0.01 0.0 0.0
      1.0
      293.0
#
# Definition of the energy source region
# Values for the velocity and the pressure must be specified, but they
# are meaningless in the simulation.
# Note that the heat release rate is in dimensionless units and corresponds
# to a real heat release rate of about 5 [MW].
      BND2
      401
      0.0 0.0 0.0
      1.0
      3000.0 0.0 0.0
# Specifies a outflow boundary
# Some values for the flow variables must be given, but they are
# not used in the simulation
      BND3
      102
      0.0 0.0 0.0
      0.99
      293.0
# Definition of the extraction openings
# A certain underpressure is specified
      BND4
      102
      0.0 0.0 0.0
      0.98
      293.0
      BND5
      102
      0.0 0.0 0.0
      0.98
      293.0
# Definition of the fresh air inlet
# For the emergency mode the inlet is deactivated
      BND6
      501
      0.0 0.0 -0.05
      1.0
      293.0
#EOF#
```

16.3 Trucks.GEO

In what follows the geometry file Trucks.GEO is shown. As it contains about 600 pages of text only a few exemplary lines of the geometry definitions are given.

```
DIMENSIONS
         200           50           30
SYMMETRY
           0            0            0            0            0            0
PERIODIC
           0            0            0
NBOUND
           6         3088
BNDIDX
           1         1345         1445         2789         2879         2969
        1344         1444         2788         2878         2968         3068
...
```

…Only the first three lines of the boundary coordinates are shown

```
…
BNDCOORD
       10002        10003        10004        10005        10006        10007
       10008        10009        10010        10011        10012        10013
       10014        10015        10016        10017        10018        10019
…
```

… The corresponding neighbour nodes to the boundary nodes above

```
…
       10052        10053        10054        10055        10056        10057
       10058        10059        10060        10061        10062        10063
       10064        10065        10066        10067        10068        10069
…
```

…Only the first three lines of the cell flags are shown

```
…
WALLFLAGS
           1 1
           2 1
           3 1
...
...
...
NOBSTACLES
       37431         6511
NACTIVECELLS
      262569       259601
#EOF#
```

16.4 Trucks.INI

```
PRES1
300000*1.0
VELO1
300000*0.05
300000*0.0
300000*0.0
CONC1
300000*273.0
```

17. APPENDIX A: DIMENSIONLESS GROUPS

This chapter is devoted to the most important dimensionless numbers used to characterise fluid flow problems.

17.1 REYNOLDS NUMBER

Probably the most famous dimensionless number related to fluid dynamics is the Reynolds number. The basic idea is that flows in a certain geometry are similar if their Reynolds numbers are identical.

The Reynolds number relates the inertia forces to the viscous forces as follows

$$\mathrm{Re} = \frac{\rho u \mathbf{L}}{\mu} \qquad \text{(Equ. 37)}$$

17.2 MACH NUMBER

The Mach number relates the flow velocity to the speed of sound of the fluid and reads

$$\mathbf{Ma} = \frac{\mathbf{u}}{\mathbf{c_s}} \qquad \text{(Equ. 38)}$$

In general a flow can be considered incompressible if the Mach number is below 0.3.

17.3 PRANDTL NUMBER

In contrary to the Reynolds number and the Mach number the Prandtl number does neither depend on the flow domain nor on the flow velocity but contains only fluid properties.

The Prandtl number is the ratio of viscosity and heat transfer coefficient :

$$\mathrm{Pr} = \frac{\nu}{\mathbf{a}} = \frac{\mu \mathbf{C_p}}{\lambda} \qquad \text{(Equ. 39)}$$

The Prandtl number of air is about 0.71 at standard conditions. For Pr = 1 the velocity boundary layer and the temperature boundary layer coincide.

17.4 PECLET NUMBER

The Peclet number relates the advective heat transport to the heat transport by diffusion and is given by

$$\mathbf{Pe} = \frac{\mathbf{uL}}{\mathbf{a}} \qquad \text{(Equ. 40)}$$

17.5 SCHMIDT NUMBER

The Schmidt number is defined similar to the Prandtl number except that the heat transfer coefficient is replaced by the diffusion coefficient of a species.

$$\mathbf{Sc} = \frac{\nu}{\mathbf{D}} \qquad \text{(Equ. 41)}$$

17.6 LEWIS NUMBER

The Lewis number relates the mass diffusivity and the thermal diffusivity. In many real combustion applications the Lewis number is roughly unity.

$$\mathbf{Le} = \frac{\mathbf{D}}{\mathbf{a}} = \frac{\mathbf{D}\rho\mathbf{C}_p}{\lambda} \qquad \text{(Equ. 42)}$$

18. REFERENCES

[1] Wilcox, D. C.
Turbulence modelling for CFD
DCW Industries, 2000.

[2] McComb, W. D.
The Physics of Fluid Turbulence
Brady, J. M., Cullen, A. L., Jones, T. V., Van Bladel, J., Woods, L. C., Wroth, C. P. (eds.)
Oxford Engineering Science Series, Claredon Press (Oxford), 25, 1990.

[3] Pope, S. B.
Turbulent flows
Cambridge University Press, 2000.

[4] Orlandi, P.
Fluid Flow Phenomena
Fluid Mechanics and its Applications, Kluwer Academic Publishers, 2001.

[5] Karki, C., Patankar, S. V., Rosenbluth, E. M., Levy, S. S.
CFD Model for Jet Fan Ventilation Systems
Proc. 10th Int. Symposium on Aerodyn. and Ventilation of Vehicle Tunnels
Boston USA, November 1st – 3rd, 2000.

[6] Novozhilov, V.
Computational fluid dynamics modelling of compartment fires
Prog. Eng. Comb. Sci. **27** (2001), 611 – 666.

[7] Xue, H., Ho, J. C., Cheng, Y. M.
Comparison of Different Combustion Models in Enclosure Fire Simulation
Fire Safety Journal, **36** (2001), 37 – 54.

[8] Centre d'Etudes des Tunnels
Les Etudes Specifiques des Danger (ESD) pour les Tunnels de Reseau Routier
Ministére de l'Equipment, des Transports et du Logement, Direction des Routes, 2001.

[9] Luo, L.-S.
The Lattice Gas and Lattice Boltzmann Methods: Past, Present, and Future
Proceedings Int. Conf. Appl. Comp. Fluid Dyn., Beijing, China, October 17 – 20, 2000.

[10] Chen, S., Doolen, G. D.
Lattice Boltzmann Method for Fluid Flows
Annu. Rev. Fluid Mech. 30, 329 – 364 (1998).

[11] Succi, S.
The Lattice Boltzmann Equation for Fluid Dynamics and Beyond
Numerical Mathematics and Scientific Computation, Oxford University Press, 2001.

[12] Qian, Y. H., D'Humieres, D., Lallemand, P.
Lattice BGK models for the Navier-Stokes equations
Europhy. Lett, **17** (1992), 479 – 484.

[13] Mei, R., Shyy, W., Yu, D., Luo, L.-S.
Lattice Boltzmann Method for 3-D Flows with Curved Boundaries
J. Comp. Phys. **161** (2000), 680 – 699.

[14] D'Humieres, D., Ginzburg, I., Krafczyk, M., Lallemand, P., Luo, L.-S.
Multiple-Relaxation-Time Lattice Boltzmann Models in Three Dimensions
Phil. Trans. R. Soc. Lond. A **360** (2002), 437 – 451.

[15] Lallemand, P., Luo, L.-S.
Hybrid Finite-Difference Thermal Lattice Boltzmann Equation
submitted to Int. J. Mod. Phy. B (2002).

[16] Ladd, A. J. C.
Numerical Simulation of Particulate Suspensions via a Discretized Boltzmann Equation
J. Fluid Mech., (1994) 271 – 285.

[17] VirCinity, http://www.vircinity.com

[18] OpenDx, http://www.opendx.org

[19] CGNS, http://www.cgns.org

[20] Pucher, K.
Konzepte der Tunnel-Brandrauchentlüftung, Längs-, Quer- oder Vollquerlüftung?
Sicherheit im Tunnel – Austroschutz 99, Österreich (1999).