

Computational Steering in Civil Engineering

J. Tölke, T. Fahrig, B. Nachtwey, M. Krafczyk
Institut für Computeranwendungen im Bauingenieurwesen (CAB)
Pockelsstraße 3, 38106 Braunschweig
TU Braunschweig
toelke@cab.bau.tu-bs.de

Abstract

We present a computational steering based system for fluid flow problems in civil engineering. It is based on three software components, a modeler, a simulation kernel and a post processing toolkit.

Introduction

Computational Steering provides methods for the integration of modeling, simulation, visualization, data analysis and post processing (Mulder, 1999). The user has full control over a running simulation and the possibility to modify objects (geometry and other properties), boundary conditions and other parameters of the system interactively. The objective of such a system is to explore the effects of changes made immediately and thus to optimize the target problem interactively. The outline of the paper is as follows: First, we briefly sketch the modeler, then the Lattice-Boltzmann method as a new technique for the computation of fluid flow problems is introduced, a section about grid generation and parallelization follows. After a short overview over the postprocessor, we discuss the integration of the different components. Finally a Prototype-Application, the optimization of a heat-conditioning, ventilation, air-conditioning system is introduced.

The modeler

As basis for the modeler we use the CAD-system AutoCAD (AutoCAD, 2003) for the following reasons:

- It is a powerful modeling system.
- It offers a powerful programming interface allowing an efficient access to the geometric data.
- It posses convenient manipulators for geometric objects.
- It is a Quasi-Standard in the civil engineering community
- Any valid volume model can be transformed to a special Brep-model (facets model), which consists only of plain triangles as constituent surfaces. This is important for the fast grid generation to be discussed below.

First the user has to construct or to import a initial geometrical configuration, which is a volume model, then he has to define the initial boundary conditions for the different objects.

Lattice Boltzmann Methods and Numerical simulation

The simulation kernel is a Lattice-Boltzmann (LB) solver for the Navier-Stokes equations, which is especially suitable for instationary flows in complex geometries.

This paragraph briefly sketches the ingredients of the method: During the last fifteen years, Lattice-Boltzmann methods have been developed and applied to simulate fluid flows for a variety of problems. The interested reader is referred to (Succi, 2001) or (Krafczyk, 2001) and the references therein. Instead of approximately solving the Navier-Stokes equations, many fluid flow problems can be simulated by solving the so-called Lattice-Boltzmann equations,

$$f_a(t + \Delta t, \bar{x} + \bar{e}_a \Delta t) - f_a(t, \bar{x}) = \Omega_a(t, \bar{x}), \quad a = 0, \dots, N$$

which can be interpreted as a discrete evolution equation originating from a Lattice-Gas system. The left hand side of the equation describes the advection of particles f_a moving with discrete velocities \bar{e}_a , whereas the right hand side models the interaction of the particles. Using an appropriate lattice and collision operator Ω_a it can be shown, that these equations are a second order scheme for the Navier-Stokes equations,.

$$\frac{\partial p}{\partial t} + \nabla(\rho \bar{u}) = 0, \quad \frac{\partial \rho \bar{u}}{\partial t} + \nabla(\rho \bar{u} \bar{u}) = -\nabla p + \nabla \bar{\tau}$$

which describe the flow of weakly compressible fluid.

For HVAC systems, flow is usually turbulent. It turns out, that LB-models can be easily extended for turbulent flows by borrowing ideas from Navier-Stokes based turbulence models, e.g. by using an algebraic Smagorinsky Ansatz for Large-Eddy simulations (LES, see Hou, 1996). Several LB extensions for thermal flows have been proposed. We use an approach proposed in (Lallemand, 2003), where the energy transport equation is additionally solved by a Finite Difference scheme.

CFD simulations are known to be computationally extremely expensive, since the numerical grid has to represent a wide range of length and time scales. The number of grid points for a direct numerical simulation can be estimated to $\text{Re}^{9/4}$ (Reynolds number), while the number of time steps is approximately given by $\text{Re}^{3/4}$. Thus, the computational effort for the simulation of turbulent flows is proportional to $\text{Re}^{11/4}$. Analyzing indoor air flow phenomena of technical relevance thus requires the use of sophisticated turbulence models and an efficiently parallelized simulation kernel.

Grid Generation, Parallelization and dynamic load balancing

The generation of unstructured meshes and the mapping of arbitrary CAD-data to computational grids in three dimensions still requires tedious user interaction. Lattice-Boltzmann solvers typically use a regular lattice, which can be created automatically and fast by the following procedure:

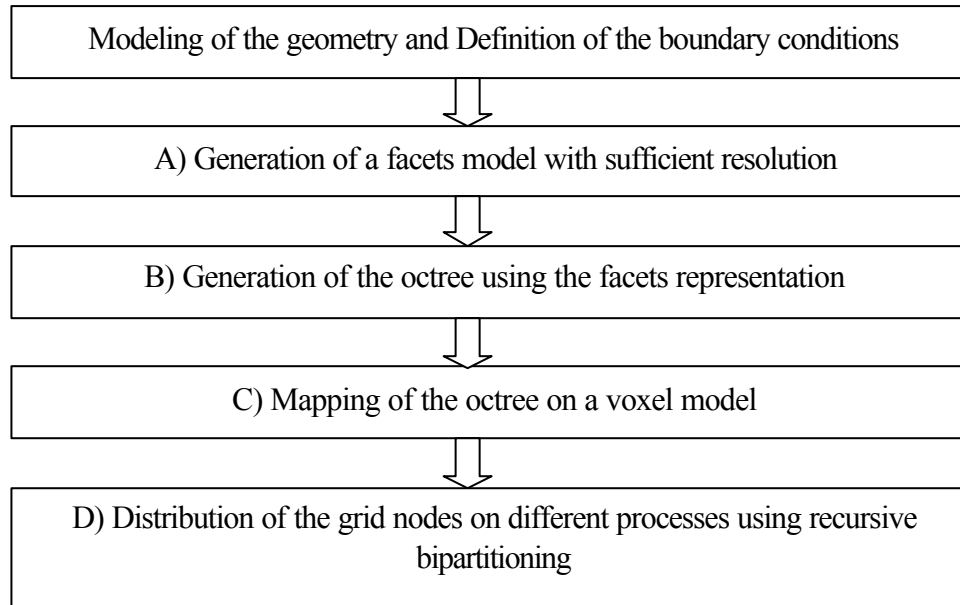


Figure 1: From the geometrical model to the distributed grid nodes.

Grid Generation:

First the user creates a model of the flow geometry and defines the boundary conditions. The flow volume has to be defined as a consistent volume model, i.e. a Brep model without gaps and overlaps.

Facets model: This is a special Brep-model, which consists only of plain triangles as constituent surfaces. It is almost the same as the standard file format STL, where the shape of the object is defined by a mesh of tiny triangles laid over the surfaces. The triangles must meet up exactly with each other, without gaps or overlaps. The Facets model is generated by an AutoCAD function, but the boundary conditions have to be assigned correctly to the facets.

Octree: For the representation of a solid by an octree the space is divided into cells. Each cell has one parent and up to eight sons and an attribute indicating whether the cell is completely inside, outside or intersecting the solid. A cell is recursively subdivided as long as it has an intersection with the boundary of the solid or the level of the tree has reached a maximal depth. The advantage of defining the solid by a facets model is that the intersection of a cube with a plain triangle can be easily calculated and results therefore in a fast generation of the octree for the flow simulation. This is done by an octree based grid generator (Jaksch, 2001), creating a grid via recursive subdivision.

Finally the octree is mapped on a discrete voxel model representing the geometrical data for the LB-simulation kernel.

Parallelization:

For the simulator, which runs on a Linux cluster using MPI (MPI, 2003), the grid nodes are distributed over the different processors using an recursive bipartitioning algorithm (Gun 2003). This works as follows:

- The voxel matrix is enclosed by a bounding box

- The box is intersected with a 'test' plane, which moves along the three orthogonal axes.
- For each position of the plan the load-imbalance and the amount of communication is calculated. The amount of communication is simply calculated by counting the number of grid nodes in the two volumes, whereas the load-imbalance is calculated by the number of cutted links. So we obtain two discrete functions depending on the location of the plane.
- The user has the possibility to weight the influence of these two factors and one obtains one function to be minimized. The cube is then intersected with that plane which minimizes the function.
- The algorithm is now recursively applied to the new two sub-boxes.

Finally one comes up with a domain partitioned into different boxes. This geometric domain decomposer is very fast and the resource requirement is rather small compared to other methods like METIS. It needs only 56000 bytes of memory to decompose $3 \cdot 10^7$ nodes. The results obtained with respect to the load-imbalance and the amount of communication are almost as good as with the decomposer METIS (Schulz, 2002,2003).

If the user modifies the system either by moving an object or by changing a boundary condition within the CAD-System, the whole procedure from A) to D) in figure 1 is repeated. This results in a different distribution of the grid nodes on the different processes, so one has to communicate the relevant information between the processes. Since the partitioning algorithm produces boxes, which can be located in space by two points, one can easily calculate the overlapping area and transmit the required information to couple the appropriate subdomains.

Visualization and post processing

For the visualization and postprocessing we use the software tool AVS (AVS, 2003), which provides a powerful programming interface and allows the efficient visualization and manipulation of scalar and vector fields even for millions of grid points.

AVS provides for scalar fields:

- Isolines
- Isosurfaces
- Isovolumes

AVS provides for vector fields:

- Vectorplots in three dimensions or in a slice plane
- Streamlines in three dimensions or in a slice plane

The geometry of the CAD system is directly imported to AVS in a vector format.

Integration of the components

The three components (modeler, simulation kernel and postprocessor) are coupled by two communication modules and three interfaces as depicted in figure 2. Interface 1 is responsible for the transformation of the modified system for the simulation kernel, interface 2 is

responsible for the proper preparation of the simulation data whereas interface 3 transforms the data from the modeler into a format suitable for the visualization system. The overall system is synchronized by the two communication modules.

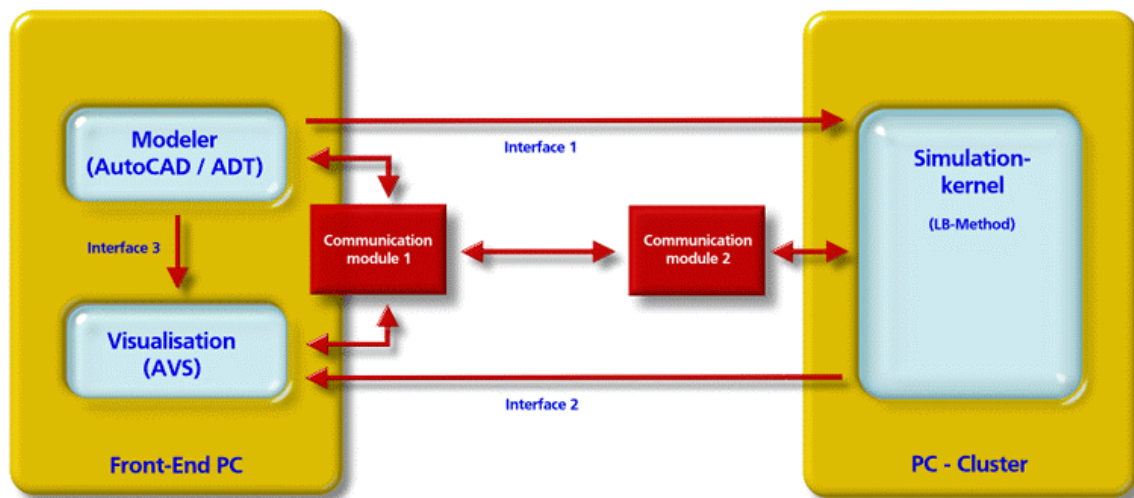


Figure 2: Integration of different software components for a computational steering system

Prototype-Application:

Optimization of an HVAC (Heat-conditioning, Ventilation, Air-conditioning) installation by interactive simulation (Kühner, 2002,2003). For an HVAC engineer, the primary variables like temperature, density and velocity are not of primary interest, but serve as the basis for the computation of human wellness and comfort criteria being relevant for dimensioning an HVAC installation. In the following, a brief review of the comfort criteria is given. For a detailed description including the corresponding equations, see e.g. Ritschel, 1994.

- PMV-Index (Predicted Mean Vote): This index is an ISO certified criterion to evaluate human thermal comfort. Basically, this index is derived from the human comfort equation, describing a combination of ambient temperature, averaged radiant temperature, air velocity and relative humidity being comfortable for a considered person at his current physical activity (metabolic rate) and clothing. The PMV-Index quantifies deviations from this optimal state by a seven point scale: hot (+3), warm (+2), slightly warm (+1), neutral (0), slightly cool (-1), cool (-2), cold (-3).
- PPD-Index (Predicted Percentage of Dissatisfied): This index specifies the percentage rate of persons being unsatisfied with a given indoor climate. The PPD-Index is derived experimentally from the PMV-Index.
- Draught risk DR: The DR indicates the risk of people feeling draught at the considered flow conditions. The important parameters are local temperature, local air velocity and the degree of turbulence (i.e. fluctuations in the transient flow velocity field).

The PVM-Index is computed from values given by the user (like the metabolic rate or humidity) and values computed from the simulation (like the temperature and the air velocity). The HVAC engineer has now the possibility to visualize the PVM-Index using slice planes or isosurfaces and to modify the boundary conditions (the in- and outlets of the ventilations) or the geometry by moving objects (tables, chairs and separating walls) in the modeler. The answer of the flow field and consequently the PVM-Index is computed online

and is available almost immediately (the time period is depending on the hardware and grid resolution), so the engineer can analyse the impact of the changes and optimize the indoor climat interactively.

Conclusions and outlook

The intention of the presented computational steering environment is to provide a tool for the evaluation and optimization of HVAC system, whereas the interactivity of the tool reduces the time for the iterative optimization process dramatically. Also an HVAC engineer will be able to predict and present effects of design variations requested by architects or building owners in the design phase. A future extension to the approach discussed here by collaborative engineering would allow engineers to dynamically attach to the interactive simulation from remote locations.

References:

- AutoCAD (2003), <http://www.autodesk.com>.
- AVS (2003), <http://www.avs.com>.
- A. Gun (2002): *Ein Gebietszerleger für parallele Lattice Boltzmann Applikationen auf der Basis von rekursiv-bipartiven Algorithmen für equidistante teilgefüllte 3D-Gitter*, diploma thesis, Lehrstuhl für Bauinformatik, TU München.
- S. Hou et al.(1996): *A Lattice-Boltzmann subgrid model for high Reynolds number flows*, Fields Inst. Comm., 6, pp 151-165.
- S. Jaksch (2001): *Facettierung dreidimensionaler Gebiete und Gittergenerierung unter Verwendung von Octree-Datenstrukturen*, diploma thesis, Lehrstuhl für Bauinformatik, TU München.
- M. Krafczyk (2001): *Gitter-Boltzmann Methoden: Von der Theorie zur Anwendung*, Professoral Dissertation, LS Bauinformatik, TU München.
- S. Kühner, M. Krafczyk, J. Tölke (2002), *Towards interactive comfort optimization of Indoor Flows using Virtual Reality based analysis of Large-Eddy simulation results*, in proceedings of ICCCBE-IX, The 9th International Conference on Computing in Civil and Building Engineering, Taipei, Taiwan, pp. 843-848.
- S. Kühner (2003), *Virtual Reality basierte Analyse und interaktive Steuerung von Strömungssimulationen im Bauingenieurwesen*, PhD thesis, LS Bauinformatik, TU München.
- P. Lallemand and L-S. Lou (2003), *Hybrid Finite-Difference thermal Lattice Boltzmann Equation*, International Journal of Modern Physics B, Vol. 17, Nos. 1 & 2 (2003) 41-47.
- MPI (2003), <http://www.mpi-forum.org>.
- J. Mulder, J. van Wijk, R. van Liere (1999): *A survey of computational steering environments*, Future generation computer systems, Vol. 15, nr. 2.
- Ritschel (1994): *Raumklimatechnik*, Band 1, Springer Verlag, Berlin, pp. 125-153.
- M. Schulz, M. Krafczyk, J. Tölke, E. Rank (2002), *Parallelization Strategies and Efficiency of CFD Computations in complex geometries using Lattice-Blotzmann methods on High-Performance Computers*, Breuer, M.; Durst, F.; Zenger, C., (Eds.): High-Performance Scientific and Engineering Computing, Proceedings of the 3rd International FORTWIHR Conference on HPSEC, Erlangen, March 12-14, 2001, pp. 115-122.
- M. Schulz (2003), *Ein anwendungsorientierter, paralleler Strömungssimulator auf Basis der Lattice-Boltzmann Methode*, PhD thesis, LS Bauinformatik, TU München.
- S. Succi (2001): *The Lattice Boltzmann equation for fluid dynamics and beyond*, Oxford University Press.