Chapter 9

Conclusions and Perspectives

This chapter summarizes the work and presents an outlook.

9.1 Summary

This work deals with the development and implementation of flexible software for finite volume computations on unstructured grids. An object oriented framework has been created. A variety of different standard methods for the numerical solution of conservation equations have been integrated into this framework. Special attention has been paid to a standardized interface, which allows different physical problems to be easily tackled, using existing methods. To achieve this a number of dynamic data structures have been created. The framework has been coded in *C++* . A tradeoff has been found between computational efficiency on the one side and an extendible and flexible programming interface on the other side. Various CFD applications have been created, based upon this framework. A few verification tests are presented in chapter 8.
As part of this numerical framework an alternative meshing algorithm has been introduced. This algorithm uses scalar functions as input to describe geometrical features. Differing from other unstructured meshing algorithms, it starts from an equidistant Cartesian grid. This grid will then be triangulated in regions where geometrical features are present. One advantage is that no initial Delaunay triangulation has to be created. It has, so far, been used for two-dimensional problems. In this scope it proves to be an efficient tool to generate unstructured and hybrid grids. In particular it is able to easily generate layers of anisotropic elements. Furthermore it is able to handle moving and deforming geometries. Fuzzy input data, such as images, can also be dealt with.
A non-linear multi-grid, using the volume agglomeration technique, has been integrated into the framework. It has been tried to keep this FAS implementation as open to different equations as possible. Unfortunately it is fairly sensitive to the equations to be solved. Thus it possibly requires fine tuning of various parameters. The multi-grid implementation cannot be seen as a black box. In this work it has been used in conjunction with fluid dynamical problems on isotropic grids.

9.2 Critical Review of Design Criteria

The design objectives

- a reliable programming interface
- reasonable efficiency
- maximal reuse-ability of components
- extendible
- customizable to various problems

which have been postulated in chapter 7 shall be critically reviewed here.

9.2.1 Programming Interface and Flexibility

In the very beginning of this project, the programming interface changed a few times. In course of the project, however, it has become fairly reliable. The usage of an interpreted control language offers a high degree of flexibility. Several projects make use of the software at this point. The fact that no re-design had to be made for various new projects proves a certain reliability of the programming interface.

9.2.2 Efficiency

Efficiency is a crucial point for numerical simulation codes. Generally *C++* used to have a bad reputation in terms of efficiency for large-scale numerical algorithms. The object oriented features, which are offered by *C++* , can sometimes tempt a developer to create an inefficient software-design. A design that seems clear and obvious for many applications might not be suitable for a high performance application. *C++* is a hybrid language and not a pure object oriented one. Thus the encapsulation of computationally extensive algorithms into objects proved to be a good strategy. For instance the usage of a *list of elements* as an object, rather than a list of *element objects*, is preferable. Furthermore the template mechanism has been an ideal tool to combine a finer grained approach into a computationally effective one.
Most of the developments have been made, using the *GNU* compiler. Unfortunately this compiler does not offer as good an optimization as many specifically targeted *Fortran* compilers usually do. One option to overcome this problem would be to use a translator from *C++* to *C* . Then a platform-specific and highly optimizing *C* compiler could be used. Optimizing *C* compilers are more widely available than *C++* compilers. Furthermore many *C++* compilers do not fully support the *ANSI* standard, still. A good example for such a translator is the *KAI++*, which is especially targeted to numerical applications. Since *KAI++* has been recently bought by *Intel* a certain hope exists, that an efficient native compiler might become available in the future.
The performance has been generally observed to be comparable to *Fortran* codes, which

have been developed using the *GNU* platform as well. The code still holds a potential for performance optimization on the programming level. This has to be seen as an important task for the near future.

9.2.3 Reuseability

Reuse should be a major goal within every software design. The experiences that have been made during this project can be described as positive. The initial phase of the developments featured a few conceptual problems. Once these have been overcome, the software started to be used in various projects. This seems to be a fair prove of its reuse-ability.

9.3 Maintainability

A minor problem is that a certain discipline has to be kept within a workgroup that uses such a common library. This is a natural problem in software-design for large projects. In a research environment, however, a tendency exists to develop programs that are targeted at one particular problem. This development style can sometimes be preferable, since it might offer results quickly. Developers, however, should frequently take the time and clean-up such developments. It has to be decided what can be useful to others. These parts should hence be integrated into the framework. Sometimes this could pose a problem, especially in the presence of tight deadlines for various research projects. So far, however, it has been managed to coordinate the different developments. The software situation at the *Institut für Verbrennung und Gasdynamik* looks more homogeneous than before, and the maintainability seems to be better than with a large number of standalone codes. This is, however, not measurable in numbers and only a subjective estimation of the author.

9.4 Projects Using the Software

The main goal of this work has been to provide a flexible toolbox for numerical simulations. A number of researchers at the *Institut für Verbrennung und Gasdynamik* make use of this toolbox as basis for new developments. Apart from the developments which have been made for this work, the following projects exist:

1. simulation of ship-flow
2. simulation of particle loaded flow (especially targeted towards nano sized particles)
3. gas kinetic (Lattice Boltzmann) schemes on unstructured grids
4. tracking of discontinuous solutions for hyperbolic equations

A common software basis offers many advantages to the people using it. Experience in the last few years showed that this library eases the usage of unstructured grids for various problems.
Now the different projects, using this software, shall be briefly introduced.

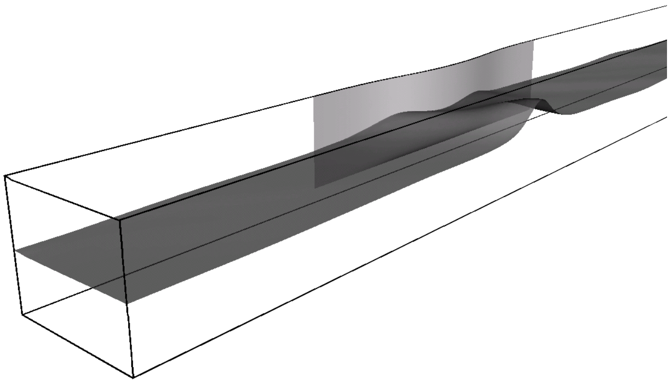

Figure 9.1: Water-surface.

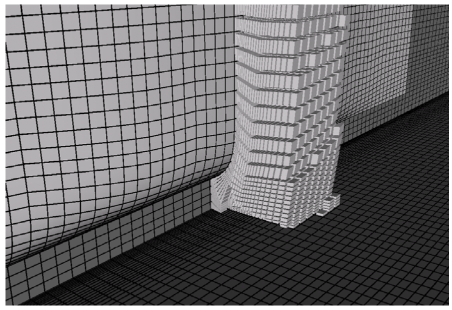

Figure 9.2: Mesh.

9.4.1 Simulation of Ship-Flow

As a cooperation between the *Institut für Verbrennung und Gasdynamik* and the *Institut für Schiffstechnik*, at the University of Duisburg, an algorithm for numerical simulations of free surface flow around ships has been developed [34],[35]. These developments originally used the method of artificial compressibility on structured grids. The free surface has been treated, using a level-set formulation. As the object oriented library became usable for large scale three dimensional computations, these developments have been integrated into this framework. This has been done by Norbert Stuntz, once again as cooperation with the institute for ship-technology. One computational example shall be shown here. It represents an inviscid and incompressible simulation, using a level-set to describe the free surface. The Froude number has been defined as

$$Fr = \frac{u_\infty}{\sqrt{gh}}, \tag{9.1}$$

where h is the depth of the water. It represents the ratio between the propagation speed of surface waves and a typical fluid-velocity. Please note that $\sqrt{gh}$ describes the propagation speed only in case of shallow-water problems.
This simulation has been made at the *Versuchsanstalt für Binnenschiffbau e.V. Duisburg* . It is part of a project, aiming at the development of a ship, with a wave resistance near zero. The goal is to design a catamaran for shallow water. The idea is that waves will only be produced in between the two hulls of the catamaran and not on the outside. These inner waves will completely eliminate each other in the wake of the ship. As a first approach towards this a monohull, traveling in a narrow channel, has been investigated. See [36] for details about this project, including a comparison between calculated and measured data.

Further Developments

For the near future it is planned to develop a ready to use application for free surface flow. Especially the interaction between a ship and the river or canal banks as well as the ground shall be investigated. Therefore a modeling of propellers becomes necessary. A fairly simple, but computationally effective method is currently being integrated by N.Stuntz, as part of his PhD thesis. This method describes the propeller as a momentum source (angular and translatory).

9.4.2 Nano Sized Particles

As part of a joint research project, a numerical approach to simulate nano-particle formation and growth is currently being developed by R. Kaiser. A number of different models for chemical reactions, transport of gas/particle mixtures, nucleation, coagulation and sintering of the particles is required. First results have been presented at the *Second International Conference on Computational Fluid Dynamics*, held in 2002 in Sydney, Australia. [37].

9.4.3 Unstructured Lattice Boltzmann Schemes

Quite recently U. Lantermann started to work on an unstructured Lattice Boltzmann scheme. Some early experiences and results have been shown at the *Third International Symposium on Finite Volumes For Complex Applications*, held in the summer of 2002 in Porquerolles, France [38]. As basis for this development he uses the library described in this work.

9.4.4 Tracking of Discontinuous Solutions

MOUSE also serves as basis for another project at the *Institut für Verbrennung und Gasdynamik* . This project deals with the treatment of discontinuous solutions for hyperbolic equations (e.g. shock-waves, flame-fronts, material-interfaces). The initial developments for this project, using a local level-set method on unstructured grids, have been made by Tran et.al. in 1998 [39]. Some more recent results have been presented at the *9th International Conference on Hyberpolic Problems* in 2002 [40]. The techniques, used for this project, are similar to the free-surface description, which is in use for ship flow computations. Developments are very ambitious, especially in terms of flexibility to solve a variety of different problems. Unfortunately they are not applicable to large scale simulations since they lack a support of distributed memory parallel machines, yet. This seems to be one of the major challenges for future developments. As stated above, this project relies on local methods and it uses many additional data-structures (e.g. local grids, mapping from local to global, ...). Thus a parallelization is not as straightforward as it is for other computations. Since ship-flow simulations require large grids, which can only be treated on parallel machines, a simpler (global) approach has been used for that project.

9.5 Future Perspectives

For the future it seems desirable to include more numerical methods, aimed at different physical problems. Another important point is to increase the robustness of the multi-grid implementation. Furthermore the multi-grid implementation should be extended to three dimensions, which seems to cause no major difficulties. The parallelization of the multi-grid structures and algorithms, however, is expected to be more complex. Another point would be to extend the level-set based meshing algorithm to three-dimensional grids. This seems to be an ambitious but also promising project. Last but not least the improvement of the compilation and installation process should be tackled. This would allow other researchers to download and use the software more easily. Right now the installation process has become rather complex. This is mainly due to the compiler requirements, as well as due to a number of software dependencies. For certain physical problems, such as the ship flow, the development of *end user* applications is planned.