CHALMERS UNIVERSITY OF TECHNOLOGY

THE FULL MULTIGRID METHOD APPLIED TO TURBULENT FLOW IN VENTILATED ENCLOSURES USING STRUCTURED AND UNSTRUCTURED GRIDS

by

Peter Emvin

Thesis submitted for the degree of Doctor of Philosophy at
the School of Mechanical and Vehicular Engineering,
Chalmers University of Technology, Göteborg, Sweden

Faculty opponent: Adjunct Professor Per Lötstedt,
SAAB AB, Linköping, Sweden

Department of Thermo and Fluid Dynamics
Chalmers University of Technology
412 96 Göteborg, Sweden

Göteborg, April 1997

ISBN 91-7197-460-1

This thesis is based on the following papers

Paper I P. Johansson[1] and L. Davidson, A full multigrid method applied to turbulent flow using the SIMPLEC algorithm together with a collocated arrangement, *Multigrid Methods IV*, Birkhäuser Verlag", pp 245-256", 1994

Paper II P. Johansson[1] and L. Davidson, Modified collocated SIMPLEC algorithm applied to buoyancy-affected turbulent flow using a multigrid solution procedure, *Num. Heat Transfer, Part B*, Vol. 28, pp 39-57, 1995

Paper III P. Emvin and L. Davidson, A local mesh refinement algorithm applied to turbulent flow, *Int. J. Num. Meth. in Fluids*, Vol 24, No. 5, pp 519-530, 1997

Paper IV P. Emvin and L. Davidson, A numerical comparison of three inlet approximations of the diffuser in case E1 Annex20, *5th Int. Conf. on Air Distributions in Rooms, ROOMVENT'96"*, Vol 1, pp 219-226, 1996

Paper V P. Emvin and L. Davidson, Development and implementation of a fast implicit multigrid method for large eddy simulations, *Submitted for Journal publication*

Paper VI P. Emvin and L. Davidson, An unstructured finite element method with a full multigrid SIMPLE solver for turbulent flow, *Submitted for Journal publication*

[1] Peter Emvin changed his name from Peter Johansson in September 1995

Contents

Abstract

Computational fluid dynamics require large computer resources and it is thus of paramount interest to have as efficient an algorithm as possible. This means that both an accurate discretization and a fast solver are required.

In this work, we have used the full multigrid concept, which is a general and optimal solver. It has been applied to subsonic flows in ventilated enclosures, and as these flows are usually turbulent, we have used several different turbulence models, such as the $k-\epsilon$ model, the $k-\omega$ model and large eddy simulation methods. We have investigated the effects of such models on the performance of the multigrid method and present some guidelines to improve both the speed and the robustness of the multigrid method in combination with turbulence models.

We have employed and developed this multigrid technique for structured one-block grids and locally refined grids, as well as for completely unstructured grids. Both the local mesh refinement method and the adaptivity showed to be very efficient and substantial reductions in both RAM memory usage and CPU-time usage were observed. All of these discretizations are at least second-order accurate, which has been confirmed on laminar benchmark problems. The performance of the multigrid has in these model cases been found to be high, i.e. convergence within work comparable with 100 residual evaluations. However, more important is that the same performance was also most often shown for turbulent flows on both structured and unstructured grids.

Keywords: Multigrid, Finite Volume, Finite Element, $k-\epsilon$, $k-\omega$, Large Eddy Simulation, Ventilated Enclosure, Local Mesh Refinement, Adaptivity

Acknowledgements

I would first like to express my gratitude to Professor Erik Olsson, who initiated this work, for his interest and support.

I am particularly grateful to my supervisor, Professor Lars Davidson, for his contribution, support and encouragement during my work.

Valuable discussions with Professor Lars-Erik Eriksson and Licenciate of Technology Thomas Hellström have provided a great deal of help and new insight, and they are very much appreciated.

I also wish to express my gratitude to all my friends and colleagues at the Department of Thermo and Fluid Dynamics and at Volvo Aero Corporation for their help and support. Special thanks to Jan Hansson and Hans Mårtensson for their support and practical arrangements.

The support of the Swedish Research Council for Engineering Sciences (TFR) is also gratefully acknowledged.

Finally, I am very grateful for all the support of my wife Lisbeth.

1 Introduction

The nature of fluid dynamics is very complex and many issues are still not fully understood, such as turbulence. Nevertheless, the performance of a large number of applications depends on the fluid flow in and around them, such as cars, aircrafts, engines, pipes and pumps. Since the analytical solution is accessible for only a few very special flow situations, we are limited to experiments or numerical methods. Experiments usually predict global effects well, while local effects are very difficult to measure. Experiments are also quite expensive, and they must be designed carefully to give reliable results. In most cases global constraints are given which only can verified in a full scale experiments. However, to use such a trial and error method is very expensive and time consuming whereby numerical predictions of the fluid flow today has shown to be necessary. These numerical predictions provide both local and global information within a short time and they therefore significantly reduce the development costs of a product.

A numerical method is based on a local information exchange, and thus both local and global information is obtained automatically in a simulation in contrast to a measurement. Therefore, when the nature of turbulence is studied for small Reynolds number flows, direct simulations are most often used instead of experiments, as they more accurately provide the desired local information. A CFD code today has the primary purpose of predicting a fluid flow rapidly and accurately even when the geometry is complex. It is no small task to develop such a tool, and thus a commercial CFD package is typically modularized into three separate parts, a pre-processor, a post-processor and flow solver. The pre-processor models the geometry and creates a grid for the flow solver, while the post-processor visualizes the predictions made in the flow solver. The most general pre-processors of today automatically provide a grid from a set of CAD surfaces with a reasonable amount of user input, while a commercial post-processor provides both plots and animations. The main part of the CFD code, the flow solver, simulates the fluid flow and typically consists of a model, a discretization and a solver.

This thesis discuss the design of a flow solver for low Mach numbers, typically $Ma = 0.001 - 0.01$. We have thus focused on the issues relating incompressible flow (at least ρ independent of p) and only briefly on shocks or other transonic phenomena. We have also disregarded all phenomena concerning combustion, discrete phase interactions and non-Newtonian fluids. We would like to point out that the object of this thesis has not been to construct a turbulence model, a discretization or a solver. Our intentions have instead been to investigate and modify the discretization and the solver combined with turbulence models to achieve an as efficient flow solver as possible. In the three following sections, we briefly present the cornerstones of a flow solver: the model, the discretization and the solver. We review some basic concepts together with some "state of the art material" for each topic. The intention is to present some of the basics that are used or referred to in our papers so that a non-specialist can enjoy the thesis without looking up all references given.

2 The model

This section presents a short review of some of the basics in turbulence modeling, in particular issues concerning our choices of models.

2.1 The Navier Stokes Equations

We first assume that we have a calorically perfect non-reacting Newtonian fluid flow without radiation. The equations governing the motion of such a fluid are the conservation laws of mass, momentum and energy, together with an equation of state:

$$\partial_t \rho + \partial_j(\rho u_j) = 0 \tag{1}$$

$$\partial_t(\rho u_i) + \partial_j(\rho u_j u_i) = \rho g_i + \partial_j P_{ji} \tag{2}$$

$$\partial_t(\rho h - p) + \partial_j(\rho u_j h - u_j p) = P_{ji}\partial_j u_i - \partial_j q_j \tag{3}$$

$$P_{ij} = -p\delta_{ij} - \frac{2}{3}\mu\delta_{ij}\partial_k u_k + \mu(\partial_j u_i + \partial_i u_j) \tag{4}$$

$$q_j = -\lambda\partial_j T \tag{5}$$

$$p = \rho RT \tag{6}$$

$$h = C_p T \tag{7}$$

where u_i=velocity, p=pressure, ρ=density, T=temperature, μ=dynamic viscosity, λ=thermal conductivity, C_p=specific heat at constant pressure, $R = 8314/M$, M=mass gas per mole and $g_i = [0, 0, -g]$. These are the basic governing equations, but they are usually simplified for specific applications. Common simplifications are incompressibility, inviscidity, stationarity, 2D, axisymmetry, irrotationallity etc. Many of these simplifications are restrictive, but it is almost always impossible to handle the full model with the computer capacity of today.

A way to reduce the size of the problem is to seek the time-averaged unstationary solution via the Reynolds-averaged Navier-Stokes equations (the time-averaged Navier-Stokes equations). However, when these equations are time-averaged, their non-linearity produces the unknown Reynolds stresses. One way to estimate the Reynolds stresses is to manipulate the Navier-Stokes equations such that a new set of differential equations for the Reynolds stresses is achieved. Unfortunately, they include even more unresolved quantities that must be modeled. These quantities are usually modeled by dimensional analysis, realizability constraints and curve-fitting to some typical flows, resulting in an RSM model [1].

A simpler and more commonly used method is to model the Reynolds stresses in terms of the mean strain rate tensor (and sometimes also the mean vorticity tensor) [2, 3, 4, 5, 6, 7, 8]. The closure parameters are then given by an expression of the turbulent kinetic energy and a quantity representing the local turbulent length scale. A differential equation for the turbulent kinetic energy and the turbulent

length scale is then derived by manipulating the Navier-Stokes equations. This type of model is usually referred to as an eddy viscosity model and is used in both research codes and commercial codes. An even simpler class of models is the algebraic models, but they are not considered in this work.

Another way to handle the turbulence is to approximate the Navier-Stokes in the sense of Large Eddy Simulations. The underlying idea is to resolve all large scales that are non-isotropic and contain most of the boundary interaction, while the small isotropic small scales are modeled. That is performed via the introduction of a filter function, separating the resolvable part from the sub-grid part of the field variables. This filter is applied to the Navier-Stokes equations, and the resulting equations govern the resolvable scales. The unknown terms caused by the unresolvable scales are then modeled, most often by a gradient diffusion assumption.

2.2 The Reynolds-averaged Navier Stokes Equations

We would first like to state that there exists a large number of turbulence models with modifications for different flow situations, but this thesis presents only the models we have used, together with some comments.

However, we first present some approximations that have proven to be valid for our applications [9]. Assume that the equation of state and the entalphy equation is independent of the pressure variations in both time and space, and the effects of both compressibility and viscous dissipation are negligible in the entalphy equation. Incompressibility together with the Boussinesq approximation is shown to be valid [9], but we skip only the dilatation term and the cross derivative term in the stress tensor P_{ji}. We also assume the density fluctuations to be negligible except in the buoyancy term. Under the assumptions given, the Reynolds-averaged equations with a linear two-equation eddy viscosity model may read:

$$\partial_j(\rho u_j) = 0 \tag{8}$$

$$\partial_j(\rho u_j u_i) = -(\rho - \rho_o)g\delta_{i3} + \partial_j((\mu + \mu_t)\partial_j u_i) \tag{9}$$

$$\partial_j(\rho u_j T) = \partial_j((\mu Pr^{-1} + \mu_t \sigma_t^{-1})\partial_j T) \tag{10}$$

$$p_o = \rho RT \tag{11}$$

$$\partial_j(\rho u_j k) = \partial_j((\mu + \mu_t \sigma_k^{-1})\partial_j k) + P_k + G_k - \rho\epsilon \tag{12}$$

$$\partial_j(\rho u_j Z) = \partial_j((\mu + \mu_t \sigma_Z^{-1})\partial_j Z) + Zk^{-1}(c_1(P_k + c_3 G_k) - c_2\rho\epsilon) \tag{13}$$

$$P_k = \mu_t \partial_j u_i (\partial_j u_i + \partial_i u_j) \tag{14}$$

$$G_k = -\mu_t g (\sigma_t T_o)^{-1} \partial_3 T \tag{15}$$

where p_o is the operating pressure, ρ_o is the operating density in the domain and $Pr = \mu C_p / \lambda$.

In Papers I-IV, we used the $k - \epsilon$ model, which implies that $c_1 = 1.44$, $c_3 = tanh|u_3/(max(|u_1|, |u_2|)|$, $c_2 = 1.92$, $\epsilon = Z$ and $\mu_t = 0.09\rho k^2/Z$.

The term G_k appears in buoyant flows and represents the buoyant production of turbulence owing to density and velocity fluctuations in a gravity field. The coefficient c_2 was recommended in [10] to be 1 in vertical boundary layers and 0 in horizontal layers, and that is satisfied by our choice based on a 2D analogy in [9]

As the $k - \epsilon$ equations are not consistent with the law of the wall, we must modify them near the wall. A popular and simple way is to simply represent the whole boundary layer by a log-law based expression in the first cell layer normal to the wall, which is required to be located at $(30 < y^+ < 200)$ [2]. This gives a very practical model as we need not resolve the boundary layer, reducing the size of the computation by a factor of two or so. The stiffness of the algebraic equation system associated with the highly stretched cells in a resolved boundary layer is also avoided when using wall-functions.

However, the simplicity of the wall functions unfortunately has a penalty in the accuracy of the predictions. Thus an improvement is to replace the wall function by a near-wall modified $k - \epsilon$ model, often referred to as low-Reynolds model [11]. Unfortunately, these are usually more unstable than wall functions, but a model that does not sacrifice robustness and still achieves most of the improvements of resolving the boundary layer was presented in [12]. They suggest the standard $k - \epsilon$ model to be used in the whole domain except in the boundary layer, where ϵ and ν_t are prescribed as:

$$\epsilon = (k\sqrt{C_\mu})^{1.5} \left(\kappa y (1 - e^{-\frac{y\sqrt{k}\, C_\mu^{0.75}}{2\nu\kappa}}) \right)^{-1} \tag{16}$$

$$\nu_t = \sqrt{k\sqrt{C_\mu}} \kappa y \left(1 - e^{-\frac{y\sqrt{k}}{70\nu}} \right). \tag{17}$$

The matching point between the two different regions in which ϵ and μ_t are solved and specified, respectively, is recommended to be set at $R_n := y\sqrt{k}/\nu > 200$ [12]. This is in fact the near-wall model we used in Papers I-IV.

Most low-Reynolds models, as well as Eq.s 16-17, use the normal wall distance, which may be troublesome to evaluate on an unstructured grid. To avoid the need of these wall distances, we use the $k - \omega$ model in combination with structured grids, which is consistent with the boundary layer and consequently does not need to be modified near the wall. Therefore, in Paper VI, we used the $k - \omega$ model reading: $c_1 = 5/9$, $c_2 = 3/40$, $\epsilon = 0.09kZ$ and $\mu_t = \rho k/Z$. However, the ω

equation requires a special treatment at the wall, as the ω equation is found to have a singular behaviour at the wall [3]. We approximated the singular boundary condition $\lim_{y\to 0}\omega = 6\mu/\rho y^2$ by replacing y with the size of the element adjacent to the wall, $\overline{y}$, which is similar to what is suggested in [3]. Note that, also for the $k-\omega$ model, wall functions can be used to approximate the whole boundary layer, reducing the number of nodes.

2.3 Large Eddy Simulations

LES is today a rapidly growing field as computer capacity and algorithms have improved [13]. We apply a spatial filter function with the size Δ_f to Eq. 1-7 and assume isothermal incompressible flow which results in the following equations.

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{u}_i\bar{u}_j\right) = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{18}$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{19}$$

where

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -\mu_t \bar{S}_{ij} \text{ and } \bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \tag{20}$$

and over-bar means filtered quantity. This is also a non-closed set of equations having a set of unknown stresses, τ_{ij}, that must be approximated analogously as in the case of Reynolds averaging. However, the underlying idea is that the unresolved turbulence becomes more local as the filter size is reduced. If then a cheap model with sufficiently good accuracy is used, it may hopefully be more efficient and robust than an eddy-viscosity model or an RSM model.

As in the case of Reynolds averaging, there is a number of LES models based on different concepts having different behaviour. The most commonly used methods employ some kind of gradient diffusion approximation of the sub-grid stresses, and they consequently differ in their way of determining the eddy viscosity. The simplest eddy viscosity approximation is presented in [14], in which $\mu_t = \rho(C\Delta_f)^2|S_{ij}|$ with $C \sim 0.06 - 0.21$.

In the dynamic model, employed in Paper V the unknown diffusion coefficient is estimated locally using two filter sizes [15, 16, 17]. Another method is to use a one-equation model similar to the k equation in the $k-\epsilon$ model and to set $\mu_t = C\sqrt{k}\Delta_f$ [18, 19]. However, LES with these types of sub-grid models are unfortunately relatively expensive, and [20] estimates the required number of nodes and time steps for a channel flow with $Re = 230000$ to be 10^8 and 10^5, respectively. The resources enabling such calculations are not admissible, and therefore coarser grids with stretching and longer time steps are used and a reduction in accuracy is accepted. Because of that, LES today is mainly a research tool, although we expect its usage to increase as computer capacity, algorithms and models are further developed.

2.4 Comments on turbulence models

A question is how well the Reynolds-averaged models represent reality. The answer is unfortunately that they sometimes predict the real fluid flow very well and sometimes do not. If we turn for a moment to more simple models, such as heat conduction or linear elasticity in homogeneous isotropic materials with well posed boundaries, it is quite easy to prove the existence of a stable and unique solution [21, 22]. For the stationary Navier-Stokes, it is possible only to prove the stability for small Reynolds numbers, while the stability is lost for larger Reynolds numbers and the existence of multiple solutions is possible [23].

The major purpose of the turbulence model is to reproduce the time-averaged, time-dependent (turbulent) solution, implying that the solution to the turbulence-modeled RNS equations should have a solution exactly the same as the time-averaged turbulent solution. This is obviously usually not true, but worse still, no error bounds of the discrepancy are available. Today, that uncertainty is usually managed by basing predictions on some *post-dictions* of similar flow situations and on engineering experience. This might also explain why a commercial code can be attractive even if it has a poor discretization and a slow solver as long as it offers a large number of turbulence and combustion models.

Another question is whether these complex equations are stable. On the basis of our experience, we believe that sometimes they are and sometimes they are not, as, to the author's knowledge, there does not exist any proof of stability or existence of any RNS turbulence model (except when simply increasing the molecular viscosity). Nevertheless, models such as $k-\epsilon$ and $k-\omega$ are today indispensable and we are forced to use them with and without error estimates.

For completeness, we would like to point out that the RNS concept works in almost all cases when using standard models such as $k-\epsilon$ with wall functions, at least when it is accepted that the solution retains some quasi-transients in some small regions.

However, it turns out to be difficult to incorporate turbulence models in a second-order accurate discretization and a fast solver, and we believe that this thesis will contribute some insight and improvements in that area.

3 The discretization

With discretization we mean the approximation of a continuous model (such as a differential equation or an integral equation) by a system of algebraic equations. Because the analytical solution is normally not in reach, such a system of algebraic equations is attractive, as it may be solved by a computer.

First we split the most commonly used methods into two classes: Finite difference methods (FDM) and Petrov-Galerkin methods. Some special methods have been extracted, such as the spectral method, the finite volume method (FVM) and the finite element method (FEM), from the Petrov-Galerkin method. In practice, an FVM can often be interpreted as an FEM and vice versa, but they are most often treated as independent concepts. We will in the remaining part of this section present the basics of the FDM, the FVM and the FEM with emphasis on the methods we have used.

The FDM is applied directly to the PDE and was the pioneer method in CFD, mainly owing to its simplicity and because much of the first theoretical work was performed using the FDM. The basic idea behind the FDM is to introduce gitter points and to use Taylor expansions of the differential operator in these points to obtain a difference stencil in each gitter point. The basic FDMs are not necessarily conservative and are thus not suitable for non-elliptic problems where special care must be taken. FDMs have also geometrical restrictions which must be considered. Lack of conservation, even if it is of the size of the discretization error, will, if accumulated, have a dramatic effect on the accuracy in predicting conservation laws [24, 25, 26]. Many finite difference schemes have thus been modified to be conservative. In the spirit of the demand for conservative schemes, use of the FVM grew rapidly during the 1970's to become the most frequently used method today. In about 1980, there were still several problems that were unsolved for both the FVM and the FEM, such as retaining stability and accuracy and handling complex geometries. Today many of the problems present in 1980 have been partially solved, but improvements are still proposed.

Until just a few years ago, almost all commercial packages used structured grids based on a cell-centered FVM, but unstructured versions have now appeared. Unfortunately, they often show a significant degeneration in accuracy compared with the corresponding structured grid formulations [27]. The FEM (or the FVM using the dual grid) is used in most unstructured research codes, and it has most often been shown to retain the same high accuracy as in the structured grid case [28, 29, 30]. Before proceeding with a description of the FVM and the FEM, we note that both seek a weak solution to the PDE (Petrov-Galerkins method) which in fact is the basic form in which a PDE usually is derived. It is therefore in fact the weak solution we seek, which is also achievable even if it is irregular.

In order to have control over our predictions, we need an error estimate for the discrete problem, and this estimate requires a stability estimate and an approximation estimate. The latter is not difficult, while ensuring good stability is difficult if the accuracy is not to degenerate.

3.1 The Petrov-Galerkin method

The Petrov-Galerkin method seeks a weak solution to the differential equation, i.e. a solution to the corresponding variational equation.

Consider an abstract problem:

$$L\Phi = s. \tag{21}$$

We have a variational form if we multiply the PDE with a test function, w, integrate over the domain Ω with the boundary Γ and vary w over a sufficiently large class of functions. The resulting variational form reads:

Find $\Phi \in H$ such that

$$(L\Phi, w)_\Omega = (s, w)_\Omega \quad \forall w \in V \tag{22}$$

where $(a, b)_\Omega = \int_\Omega abd\Omega$ and V, H are suitable Sobolev spaces. Different choices in approximating V, H by V^h H^h will now give different numerical methods. Some have been used more than others and have been given separate names. In the spectral method, V^h and H^h consist of trigonometric polynomials, V^h consists of piecewise constant functions in the standard FVM and $V^h = H^h$ consists of continuous piecewise polynomials in the standard FEM.

Many other choices of V^h and H^h can be made, and it is simply a matter of taste whether to call them an FVM or an FEM. It is, however, quite a large difference to implement the methods on a computer, and their physical interpretation is often different. For these reasons, we distinguish between the FVM and the FEM.

3.2 The grid

We have not as yet specified exactly how we approximate V, H by their discrete counterparts V^h and H^h. Gridless methods are very attractive as they are general, but they are too expensive today and developments are still required for them to be competitive with grid-based methods [31]. If Ω is subdivided into cells, Ω_j, and if local polynomials, which are almost orthogonal, are used as a basis for V^h and H^h, the matrices become sparse. There exists a number of candidates, and many approaches have been proposed during the evolution of CFD. Nevertheless, there are some strategies that are more commonly used than others, and we will present some of them.

One simple choice is to use a Cartesian structured grid, which consists of uniform isotropic orthogonal hexagons having exactly coinciding faces. In this case, the cell-to-cell connectivity is inherently built into the formulation, making it easy to construct a computer code, but putting severe restrictions on the applicability to general geometries. One improvement of the geometry restriction is to use a grid consisting of non-orthogonal hexagons and use a local mapping of each hexagon onto a unit cube. A second improvement is to use a domain decomposition technique, partioning the grid into structured blocks, and then using some kind of

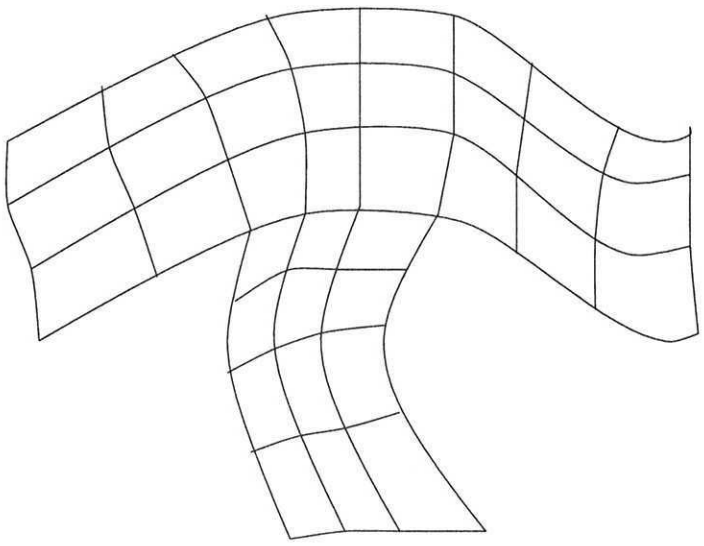


Figure 1: An example of a multiblock grid.

interface treatment between the blocks. The simplest choice of interface is to have exactly coinciding cell faces, usually referred to as a multi-block method, see Fig. 1. This is a kind of standard today in both commercial and research CFD codes. This type of method has the restriction that an opposite boundary should have the same cell face topology, making it tedious and time-consuming to create grids in complex geometries.

The topological restriction may be relieved by a more general interface treatment that does not require coinciding faces on the interfaces between the sub-domains, see Fig. 2. To preserve the flexibility of free interfaces but simplify the grid generation, overlapping grids (chimera grids) may be used, in which the interface between the sub-domains is replaced by an overlap region, see Fig. 3. The grid may then be created independently in each region, making it easier to generate the grid in each sub-domain.

Starting from a one-block structured grid where no topological information was stored and proceeding to the sub-domain methods where the block-to-block topology was stored, we now come to the completely unstructured grid formulation where all topology is stored, see Fig. 4. However, it turns out that, if tetrahedrons are used instead of hexagons, the topological requirement present for a block-structured grid disappears, which makes it much easier to construct an automatic grid generator for complex geometry. On unstructured grids, a natural extension is to use adaptive refinements of the grid, as all topology is stored. Using adaptivity, the grid can be modified such that regions in which rapid changes in Φ occur are provided with a finer grid than smooth regions in order to reduce computer costs. The adaptivity can either be based on clustering and stretching of already

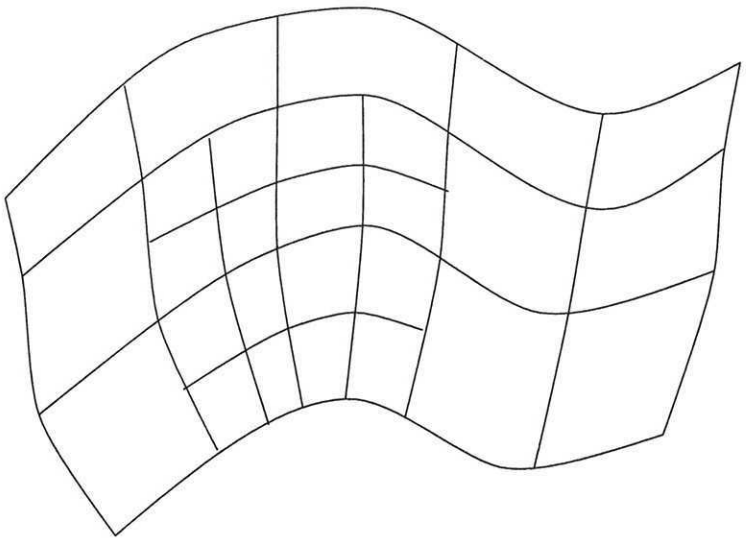


Figure 2: An example of a local grid refinement.

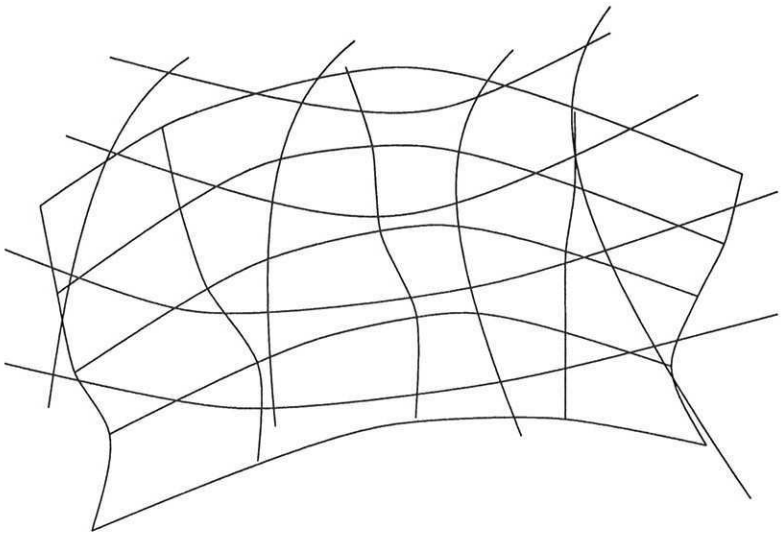


Figure 3: An example of an overlapping grid.

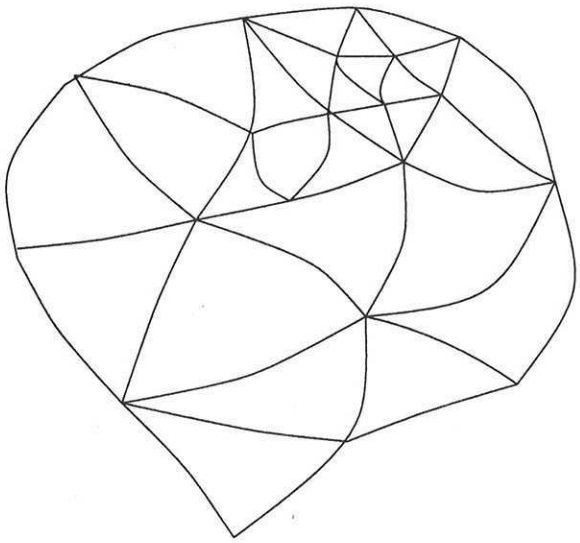


Figure 4: An example of an unstructured grid.

inserted cells or new cells can be inserted. However, estimating the required local grid size requires an error estimate, which will be treated in the "Error estimates" subsection.

The use of unstructured grids is rapidly growing as a result of their ability to approximate complex geometries, and most unstructured methods of today are based on tetrahedrons. However, methods using a mix of prisms, tetrahedrons and hexagons are also appearing and may take advantage of the different features of each element type.

3.3 The finite volume method

The finite volume method was developed for conservation laws and is per definition conservative. If we assume for the sake of simplicity that our abstract weak form Eq. 22 is a scalar conservation law with the flux vector $f_i(\Phi)$, we obtain our weak form as:

Find $\Phi \in H$ such that

$$(\partial_i f_i, w)_\Omega = (s, w)_\Omega \ \ \forall w \in V. \tag{23}$$

If we then split Ω into N finite volumes, Ω_j, and let w^h be a linear combination of w_j^h, $j = 1, ..., N$ where $w_j^h = 1$ if $x \in \Omega_j$ and $w_j^h = 0$ elsewhere, we find the following method:

Find $\Phi^h \in H^h$ such that

$$(\partial_i f_i^h, w^h)_\Omega = (s^h, w^h)_\Omega \ \ \forall w^h \in V^h. \tag{24}$$

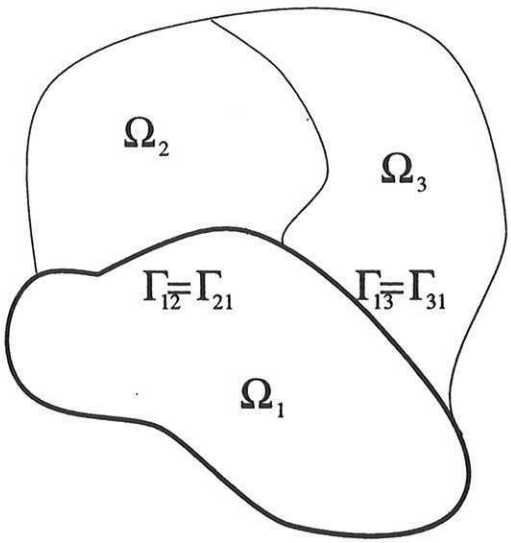


Figure 5: The definition of the internal interface Γ_{jp}.

$\Leftrightarrow$

Find $\Phi^h \in H^h$ such that

$$(\partial_i f_i^h, 1)_{\Omega_j} = (s^h, 1)_{\Omega_j} \quad \forall j = 1, ..., N. \tag{25}$$

If we then apply the Gauss Theorem on Ω_j with the boundary Γ_j with the outward normal n_i, together with a flux conservation assumption at Γ_j , we obtain our FVM:

Find $\Phi^h \in H^h$ such that

$$[f_i^h, 1]_{\Gamma_j} = (s^h, 1)_{\Omega_j} \quad \forall j = 1, ..., N \tag{26}$$

where $[a_i, b]_\Gamma = \int_\Gamma a_i b n_i d\Gamma$. Next, we subdive Γ_j such that Eq. 26 may be evaluated more easily. Owing to the conservation condition on Eq. 26, it is suitable to subdivide Γ_j into parts Γ_{pj}, which are connected to only one neighboring cell, Ω_p, see Fig. 5. To avoid duplicate operations and ensure the conservation property in a computer code, it is suitable to define a flux F_{jp} such as:

$$-F_{jp} = F_{pj} := [f_i^h, 1]_{\Gamma_{pj}} \tag{27}$$

and thus

$$[f_i^h, 1]_{\Gamma_j} = \sum_p [f_i^h, 1]_{\Gamma_{jp}} = \sum_p F_{jp} = \sum_p -F_{pj}. \tag{28}$$

When evaluating the fluxes, F_{pj}, we preferably use numerical integration by quadrature and also often a mapping of Ω_j onto a unit element. Using numerical

quadrature, we weight with W_q the value of the integrand, f_i^h, in the preselected quadrature points q as:

$$[f_i^h, n_i]_{\Gamma_{jp}} = \sum_q W_q \overline{f_{iq}^h n_i} \tag{29}$$

We must now define our degrees of freedom (d.o.f's) and express $\overline{f_{iq}^h}$ in terms of the chosen d.o.f.'s. The two most common choices are either to use the average of Φ over a cell Ω_j or to use the value of Φ in the center of Ω_j as d.o.f.'s. To express the fluxes in the quadrature points, $\overline{f_{iq}^h}$, in terms of our d.o.f's, we may suitably reconstruct Φ in terms of our d.o.f.'s in the neighbourhood of Γ_{jp} and then evaluate $\overline{f_{iq}^h}$. One commonly used approach is to reconstruct Φ in both Ω_j and Ω_p independently, whereby Φ is discontinuous at Γ_{jp}, and to use an approximative Riemann solver [32, 33] to determine the fluxes $\overline{f_{iq}^h}$.

Another commonly used method is to reconstruct Φ with a function continuous over Γ_{jp}, giving the fluxes $\overline{f_{iq}^h}$. However, to also achieve the necessary stability, which is hard to realize, some special care must be taken. An odd derivative discretized centrally turns out to have low stability and produce spurious oscillations. These oscillations appear when the grid size is larger than the local scale of the flow, such as in boundary layers or shocks. These oscillations may contaminate the whole solution but, even worse, it will be difficult to select a good adaption criterion if the solution is oscillating. Another problem is that most iterative methods show poor behaviour if the underlying discretization has a low stability.

A symmetric diffusive operator is usually stabilizing, and a large class of methods has been based on adding various diffusion operators. A closer investigation shows that the added diffusion operator, in combination with the convection operator, gives a stencil with higher weights on the upwind side than on the downwind side. This then reflects the sensible idea that information should be collected from the upwind direction. To use more cells on the upwind side of a reconstruction is therefore another means to improve the stability. When using cell averages as d.o.f's, it is suitable for the Riemann solver to be stabilizing.

If we like to have a higher order FVM ($O(h^{p+1})$) with $p \geq 1$, we first need an $O(p+1)$ order mapping, roughly $2p-1$ quadrature points (in 3D) and a stable $O(p+1)$ order reconstruction for the integrand in the quadrature points [34, 35, 36, 37]. Therefore, in a first and a second-order method, only one quadrature point, linear mapping and reconstruction are necessary. If in addition we restrict ourselves to smooth, structured almost orthogonal grids which vary as $O(h)$, we do not need to be as strict in the mapping, and stability estimates on Cartesian grids have often proven valid. Such stability estimates may be performed by expanding the solution in Fourier series. On a structured grid can the discretization of a convective term on a 3D problem be reduced to three independent 1D discretizations of the mapped operator, which makes it much easier to implement on a computer.

3.3.1 First and second order FVMs

Developing a stencil with a second-order truncation error is quite simple, as linear approximations do. However, to also retain stability is more difficult, and a stencil arising from a second-order accurate central discretization of an odd derivative on a Cartesian grid lacks stability of the highest frequency. This results in large coefficients to the neighbours and a small coefficient to the center node, often referred to as odd-even decoupling of a stencil. This means that a perturbation in the solution with the highest wave number goes almost unnoticed by the stencil. Consequently, the part of the source term corresponding to that wave number will produce a large output, i.e. there will be spurious oscillation. The absolute need of having stability has also resulted in its being given different names, such as convective sensibility and h-ellipticity [24, 38].

To stabilize via adding diffusion operators may suitably be implemented as undivided differences, implying the second-order diffusion operator to be scaled by h and the fourth-order diffusion operator to be scaled by h^3. Stencils based on this idea are used in both compressible flows and incompressible flows using structured and unstructured grids [28, 30, 39, 40].

A second class of methods is the upwinded reconstructions, which have been widely used for incompressible flows and structured grids, resulting in schemes such as first and second-order upwind and QUICK, to mention a few [24, 41, 42, 43], although upwinding techniques have also been used for compressible flows [44, 45]. However, only a first-order linear scheme obeys a discrete maximum principle, and we consequently need a non-linear scheme if this is required. It will then locally be only first order, but it will hopefully not have a great effect on the global behaviour. Several concepts for (almost) enforcing a discrete maximum principle have been proposed. This has been realized via flux limiters (almost) ensuring a Total Variation Diminishing (TVD) property [34, 46, 47, 48, 49, 50, 51, 52], and via special reconstructions said to be Essentially Non Oscillating (ENO) which satisfy TVD up to $O(h^{p+1})$ [35, 36, 37, 53]. In [54] it is shown that several of the most common TVD limiters do not preserve second order accuracy on either structured or unstructured grids. They often degenerate to first order even for smooth flows.

We have in Papers I-IV consequently used the QUICK scheme for U_i, T, while we used either an upwind scheme or a TVD-MUSCL scheme for k, ϵ to avoid undershoots which may lead to rapid divergence.

When simulating incompressible flows, an instability on the pressure is usually present. A common way to stabilize the pressure, which preserves the second-order accuracy, is to add $(h^3|U|^{-1}\partial_i^4 P)$ to the continuity equation [55]. This method is usually referred to as the Rhie and Chow interpolation or PWIM (Pressure Weighted Interpolation Method), and has been used and discussed by many [56, 57, 58, 59, 60]. This issue is treated in Paper II, where we found that the iterative method (SIMPLEC) became unstable if the scaling of the dissipation stencil was assumed to vary over the support of the stencil. We should also point out that the stability of the pressure may also be ensured via a non-symmetric stencil of

the continuity and pressure [61], but we have found the dissipation concept to be the most robust.

When simulating incompressible flows 15 years ago, however, it was most common to locate the velocity d.o.f.'s on each Γ_{pj}, i.e. the MAC stencil or the staggered grid, but this had the drawback that a new set of finite volumes had to be created for each velocity d.o.f. The major reason for doing so was to stabilize the pressure. However, the extra complications induced by these three extra sets of volumes are great for structured nonorthogonal grids and even greater for unstructured grids. This method is thus less used today, and almost all large research codes and all commercial FVM packages use cell-centered d.o.f.'s. However, the choice of cells varies and, in general, commercial codes use the cells provided by the mesh generator (tetrahedrons) while research codes use the dual cells (the Voronoi cells).

However, the discussion so far has been concerned with structured and unstructured methods. An alternative with higher flexibility than a structured grid but that avoids the difficulties related to an unstructed grid is to use structured sub-grids with interfaces without coinciding cell faces [62, 63, 64, 65, 66, 67, 68]. This concept requires the interface to be treated separately, thus requiring special stencils in calculating F_p on the interface. The concept of totally free distributions on each side of the interface is implemented in [69] and is based on the investigations in [65]. In Papers III-IV we used the system that a face of one cell on the coarse side exactly coincides with the faces of four cells on the fine side. Such an arrangement is suitable in combination with a multigrid solver.

Another approach is to use overlapping grids in which the two regions are coupled by the fluxes achieved on each sub-domain boundary using information from the region overlapping this sub-domain boundary [70, 71, 72]. Note that these fluxes are introduced into only one cell, whereby we have lost the conservation property. Conservation is important, and is often globally fixed via prescribing zero mass accumulation in the overlap region.

3.4 The finite element method

The Petrov-Galerkin method Eq. 22, with H_h consisting of piecewise polynomials and V_h not consisting of piecewise constant polynomials is here referred to as the FEM. It has been used in many variants [21, 73, 74]. If we chose V_h and H_h to consist of piecewise linear functions, it corresponds to a centered scheme, and all problems present in a centered FDM or FVM are also present here, implying a low stability of the discrete equations. For incompressible flows in combination with a turbulence model, the boundary layers must be resolved to achieve good accuracy, and thus it is only the pressure that is most important to stabilize. This can be done by using lower order elements for the pressure [75, 76, 77].

However, it is unnecessarily complex to implement that strategy, and it is well suited for neither compressible flows nor multigrid. Stabilized methods using equal order elements are thus required, and two classes of methods have often

been used. The first method adds linear or non-linear artificial viscosity terms of the form $(h\partial_i^2 + h^3\partial_i^4)$ to the PDE and then applies the standard Galerkin method (V_h and H_h, consisting of piecewise linear polynomials) [28, 30, 40]. These methods, which are approximately second-order accurate, can also be regarded as cell-centered FVMs on the dual grid and is most often referred to as cell-vertex FVMs [40, 78]. Another method is the SUPG method, which modifies the test function with the residual (or a part of the residual) [79, 80, 81] and has been successfully applied to scalar equations [82, 83, 84], incompressible flows [85, 86, 87, 88, 89] and compressible flows [34, 90, 91, 92].

The lack of high-frequency stability of the convection operator may, as in finite volumes, easily be cured by adding artificial viscosity at the cost of accuracy. If instead of simply adding viscosity, we add an artificial diffusion term acting along the streamlines, we obtain a stabilization. This is analogous to the skew upwind scheme used in finite volumes [41], but unfortunately is only first-order accurate. However, in SUPG, we add $hU \cdot \nabla w$ to the test function (h is the grid size). This method is consistent and corresponds to a skew upwind scheme with a compensation in the source term, such that the second-order accuracy is retained. For the scalar convection equation, this SUPG method reads:

$$(U_i^h \partial_i^h \Phi^h - s^h, w^h + \delta U_i^h \partial_i^h w^h) = 0 \quad \forall w^h \in V^h \tag{30}$$

and, if integrated by parts and under sufficient regularity assumptions (and $\partial_i u_i = 0$), we see that it corresponds approximately to

$$(1 - \delta u_i \partial_i)(u_j \partial_j \Phi - s) = 0 \tag{31}$$

explaining what we mean by compensating for the skew-upwind diffusion-term $\delta u_i \partial_i (u_j \partial_j \Phi)$ in the source term by $\delta u_i \partial_i s$.

The instability on the pressure in incompressible flows can be cured as in the FVM by adding pressure diffusion to the continuity equation [93]. However, if we add a perturbed least square problem to the Galerkin method, we obtain a consistent stabilization without sacrificing much accuracy. The Galerkin/Least squares FEM incorporates a consistent velocity and pressure diffusion for the Navier-Stokes equations and, when applied to Eq. 23, it then reads:

$$(R(\Phi^h), w^h + \delta_R R(w^h)) = 0 \quad \forall w^h \in V^h \tag{32}$$

where $R(\Phi^h) = L\Phi^h - f$ and $\delta_R = O(L^{-1})$, and it can be shown that it provides most of the necessary stability. If a still higher stability is required, a non-linear artificial viscosity $\mu^* \sim h^\alpha R(\Phi)$ may be added that (almost) enforces a discrete maximum principle [80, 82], which may be important in turbulent flow simulations.

As the FEM is transparent with the weak form, it makes energy error estimates straightforward. However, they show that the SUPG is not optimal ($O(h^{p+0.5})$) for a convection-dominated flow, although this non-optimality is usually not observed in practice unless the grid not is severely distorted [94]. In Paper VI, we

used the SUPG-FEM on unstructured grids and the non-optimality did not appear in our calculations. Note that if, for example, a convection diffusion operator is discretized with SUPG on a 2D Cartesian grid split into triangles with linear polynomials, and with the resulting stencil expanded in Taylor series, we clearly obtain a second-order nodal truncation error.

As the stability and error analysis is based on energy estimates, it does not matter whether the grid is structured or unstructured, as long as the grid is not degenerated. Thus, no additional problems in the discretization arise when switching from structured to unstructured grids, as is often the case when extending a structured FVM into an unstructed FVM. In an FEM, like in an FVM, it is possible to have a special interface treatment between structured sub-grids, although this is not often used in an FEM.

3.5 A postteori error estimates

To save computational costs, we wish to distribute our *d.o.f.'s* such that the discretization error is minimized, and thus we need a strategy to quantify the discretization error. The solution error can be estimated in terms of the exact solution via an a priori estimate having the advantage that the information of the method can be extracted before any calculation is done, i.e. a priori. An a priori error estimate of the error $e_h = \Phi - \Phi^h$ may then be done if we assume that L_h is stable, i.e. invertible,

$$\tau^h := (L^h\Phi - f) = L^h(\Phi - \Phi^h) = L^h e^h \;\Rightarrow e^h = L^{-h}\tau^h. \tag{33}$$

We see that the origin of the error, e^h, is the truncation error, τ^h, and thus it is in regions of high τ^h that the grid should be refined. Considering Eq. 33, we see that regions of high τ^h are not necessarily the same as regions of high e^h, as L^h acts on e^h. In fluid flow that means that errors may be transported far from their origin of high τ^h. Therefore if the grid is adapted according to the magnitude of e^h instead of τ^h it will thus most likely give too fine a grid downstream a region of hig τ^h, slowing down the calculations. However, as the exact solution is unknown, this a priori estimate does not provide any local information. In an a postteori estimate, the error is expressed in an already computed quantity and thus provides the required local information. A common way to achieve a rough a postteori error estimate is to model the unknown truncation error, τ^h, in an a priori estimate. This may be done by expressing τ^h in higher order derivatives of the exact solution, $D^\alpha\Phi$ with $\alpha \sim 3-5$, and it is a widely used adaption criterion to state that $D^\alpha\Phi^h \sim D^\beta\Phi$ with $\beta \sim 1-2$ [28, 62, 95].

Another strategy for estimating τ^h is presented in [38, 96], where τ^h is estimated in terms of Φ^h when projected onto a twice as coarse grid, and has been used by several authors [97, 98, 99].

This error estimate reads:

$$\tau^h = 2^{-(p+1)}\tau^{2h} + O(h^{(q+1)}) = -2^{-(p+1)}L^{2h}(\Phi^{2h} - \Phi) + O(h^{(q+1)} =$$
$$-2^{-(p+1)}L^{2h}(\Phi^{2h} - (2^{(p+1)} - 1)^{-1}(2^{(p+1)}I_h^{2h}\Phi^h - \Phi^{2h}) + O(h^{(q+1)})) =$$
$$(2^{p+1} - 1)^{-1}(f - L^{2h}(I_h^{2h}\Phi^h)) + O(h^{q-1}) \tag{34}$$

Before proceeding, note that even if I_h^{2h} is sufficiently accurate, the assumption is at most of the same order as τ^h but, according to Brandt [38], τ^h should be dominant. When τ^h is estimated, an ansatz of the behaviour of τ^h is made, and the grid points are distributed to minimize $\|\tau^h\|$.

We used this approach in Paper III to obtain a rough estimate of where to refine the grid, and we used $I_h^{2h} \sim O(h^2)$, which clearly dominates over τ^h, on fine grids. As the leading term in the error of $I_h^{2h}\Phi^h$ is a second derivative, however, we might have adopted the grid to the second derivatives instead. Because higher derivatives of Φ are the leading terms in τ^h, it might explain why good results can be obtained using this technique for grid refinements. If we had used cubical interpolation in I_h^{2h}, then $L^{2h}(I_h^{2h}) \sim O(h^2)$ but, as it is expensive and difficult to implement on non-Cartesian grids, we disregarded this option.

However, we would appreciate a simpler and more accurate a postteori estimate for grid adaption than those presented above. Thus, rather than an "a posteorized" a priori estimate, we instead present a second technique that gives a sharp bound of the error in terms of Φ^h [21, 22, 80, 81, 100]. It actually admits an estimate of the absolute size of the error in an approximation, Φ^h, and is a method developed for FEMs, based on the residual of the computed solution $R(\Phi^h)$. Basically, such an a postteori estimate may be constructed as follows. Consider an abstract problem:

$$R(\Phi) := L\Phi - s = 0 \tag{35}$$

with the variational form

$$\int_\Omega (L\Phi - f)w dx = 0 \ \ \forall w \in V \tag{36}$$

and the corresponding finite element method

$$(L\Phi^h, w^h) - (f, w^h) = 0 \ \ \forall w^h \in V_h \subset V. \tag{37}$$

Because $V_h \subset V$ we obtain from Eq. 36-37 the so-called Galerkin orthogonality:

$$(Le^h, w^h) := (L(\Phi - \Phi^h), w^h) = 0 \ \ \forall w^h \in V_h. \tag{38}$$

We will need the continuous dual problem

$$L^T\chi = e^h/\|e^h\| \ \text{ where } \ \|\cdot\| = sqrt(\cdot,\cdot) \tag{39}$$

supplied with a strong stability estimate

$$\|D^\alpha\chi\| \leq C_2\|e/\|e\|\| = C_2. \tag{40}$$

We also need an interpolation estimate (with π^h as the nodal interpolant)

$$\|h^{-\alpha}(\chi - \pi^h\chi)\| \leq C_1\|D^\alpha\chi\|. \tag{41}$$

Combining Eq. 38-41, we obtain

$$\begin{gathered}\|e^h\| = (e^h, e^h/\|e^h\|) = (e^h, L^T\chi) = (Le^h, \chi) = (Le^h, \chi - \pi^h\chi) = \\ (L(\Phi - \Phi^h), \chi - \pi^h\chi) = (s - L\Phi^h, \chi - \pi^h\chi) = \\ -(R(\Phi^h), \chi - \pi^h\chi) \leq C_3\|h^\alpha R(\Phi^h)\|.\end{gathered} \tag{42}$$

Thus we have an error estimate of a computed approximation, Φ^h, in terms of the easily computable quantity, $R(\Phi^h)$. The constant C_3 is in fact computable, and thus the absolute size of the error in an approximation can be obtained. However, for complicated cases, C_3 may be too large and Eq. 42 will then not make sense. Still, is it possible to use Eq. 42 as a sensor to refine the grid in regions of high $h^\alpha R(\Phi)$, giving a better approximation on this new grid. It may be made by an equidistribution of the nodes, where $h^\alpha \int_{\Omega_j} R(\Phi)d\Omega_j$ should be equal in the whole domain, i.e. all elements should contribute equally to the error.

4 The solver

To be able to solve the linear or non-linear set of algebraic equations arising from the discretization, we need some kind of matrix solver. It is an easy task to find a method inverting L^h (if L^h is non-singular), but to do that in $O(N)$ (N=number of d.o.f.'s) operations has been found to be very difficult. Before proceeding, we will note that matrix L^h is sparse, is banded if the grid is structured and is symmetric if the operator and the discretization are symmetric.

Most matrix solvers may be classified as direct, relaxation, minimization or multigrid methods. The direct methods invert matrix L^h, while the others are iterative. The multigrid method in fact uses a relaxation or minimization method in combination with a sequence of subproblems.

The efficiency of different methods varies greatly and direct methods are in practice useless as they require far too much memory and CPU time. The relaxation methods are the most frequently used solvers today, as they are simple, robust, have a low memory requirement and are easily used in combination with turbulence and combustion models . However, an increasing use of minimization methods has appeared [29, 89]. One advantage such methods offer is that they exist as black-box solvers on public domain sites, but they unfortunately use quite alot of RAM.

Finally, the multigrid method, which in fact is optimal $O(N)$ in contrast to other methods, is usually a factor of ten faster than any of the other methods and does not require a great deal of extra memory. Note that, if multigrid is implemented such as it runs with the optimal speed, it would usually be a further factor of ten faster. In the next subsection, will we describe some of the most commonly used matrix solvers and we therefore define an iterative method such as:

$$\Phi^h_{n+1} = M(s - L^h\Phi^h_n) + \Phi^h_n = -Mr^h_n + \Phi^h_n = \Delta\Phi^h_n + \Phi^h_n \quad n = 0, 1, 2, ... \tag{43}$$

4.1 Direct methods

We immediately see that if $M = L^{-h}$ then $\Phi = \Phi^h_1 = L^{-h}s$, which is referred to as a direct method. There are several different techniques to evaluate L^{-h}, such as Gauss Elimination or LU decomposition. However, they taking only partially advantage of the sparsity of L^h and therefore require large CPU and RAM resources. Even if there exist somewhat more efficient direct methods, such as nested dissection, these methods become prohibitively expensive for CFD purposes in terms of both CPU time and RAM usage. For non-linear problems such as CFD, it is also necessary to iteratively update the non-linearity and, even when using a Newton linearization, ~ 20 iterations are usually required when using a standard turbulence model such as $k - \epsilon$.

Our experience is that it is not possible on a workstation to run computations of more than ~ 10000 nodes with a direct method, as the memory usage exceeds

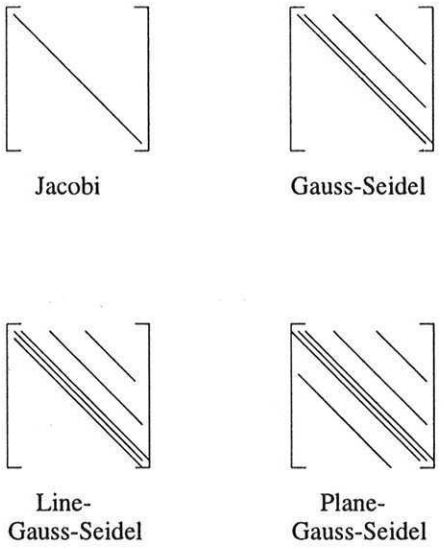


Figure 6: The structure of M^{-1} for the most commonly used relaxation methods for a scalar equation on a structured grid.

500 Mbytes and the CPU time is greater than 10 hours. We might point out that at least 100 000 nodes are needed in a 3D calculation, and direct methods are therefore used only in small research codes today.

4.2 Relaxation methods

The relaxation methods may be described in terms of splittings of L^h, i.e $L^h = B - C$ and $M = B^{-1}$. The behaviour of the iterative method is related to the choice of B and, loosely speaking, we can say that the more of L^h left in B, the better the behaviour of the iterative method. Fig. 6 shows the choice of B in some of the most commonly used iterative methods. We see that the Jacobi and the Gauss-Seidel (GS) methods are easily implemented as recursive substitutions on both structured and unstructured grids. The line and plane GS methods are implemented on structured grids as recursive substitutions of lines and planes, respectively. There exist renumbering algorithms such that a set of lexographically numbered points become located on a line [29], a so called snake, and thus admit the use of line GS on unstructured grids as well.

As these basic iterative methods are implemented as recursive substitutions, they are also referred to as local averaging methods. Many variants are constructed from these basic concepts, one example being the black and white GS, where even numbers are relaxed first and then odd. Another variant of GS is the symmetric GS in which first a standard GS sweep is performed, followed by a GS sweep with lexographically decreasing numbers. Symmetric and "black and white" variants of

the line and plane GS also exist. In addition, the Runge-Kutta family can also be regarded as a sequence of scaled Jacobi methods.

Fig. 6 shows that the changes in Φ_n^h may become large if M_{ii} is small (at least for Jacobi and Gauss-Seidel). However, as we can modify M as we like (since it does not affect the accuracy), we may use an under-relaxation of M, decreasing the changes in Φ_n^h and consequently increasing the robustness but probably also slowing down the iteration process. With under-relaxation we mean either to scale $diag(M)$ with a number $\alpha > 1$ or to add a diagonal mass matrix, $I^h \delta V/\beta$, where β can be regarded as a time step.

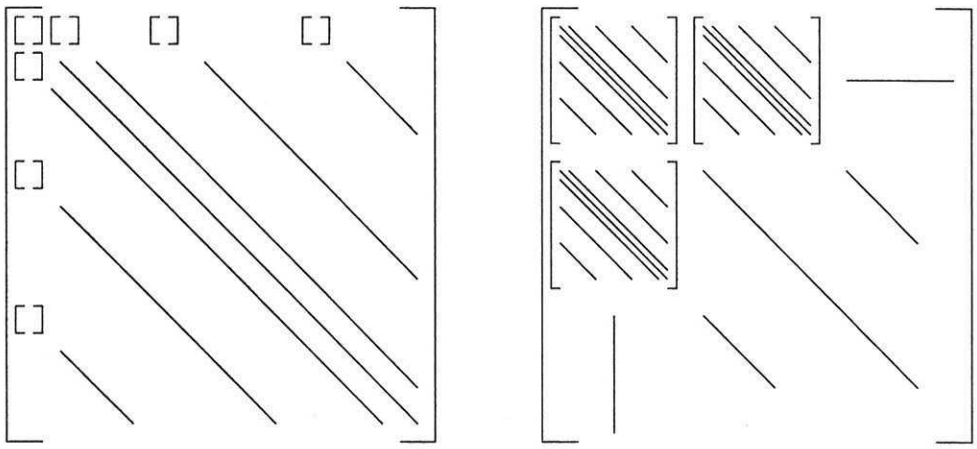
It can be shown that GS converges for an M matrix, i.e a matrix with the property $M_{ij} \leq 0$ if $i \neq j$ and $M_{ii} \geq -\sum_j M_{ij}$. On structured grids, a second-order central difference approximation of a diffusion term and a first-order upwind approximation of a convective term yields an M matrix. This is not necessarily the case on unstructured grids, but a strategy to extract a M-matrix on general grids is presented in [101]: if for $i \neq j$ then $M_{ij} > a > 0$, $M_{ij} \leftarrow 0$, $M_{ii} \leftarrow M_{ii} + a$, $M_{ji} \leftarrow M_{ji} - a$ and $M_{jj} \leftarrow M_{jj} + a$. For the Navier-Stokes equations, however, we still need to use an under-relaxation, $\alpha \sim 0.7$, even if we have an M matrix, probably because of the non-linearity in the convection term and the feedback from the pressure correction equation.

This concept for suitably modifying M has been employed in Papers I-IV and Paper VI and can be regarded as a deferred correction method (with $i_0 = 1$):

$$
\begin{aligned}
&do\ j = 1, j_0 \\
&do\ i = 1, i_0 \\
&\Phi^h_{i+1,j} = M((s - L^h\Phi^h_{0,j} + \overline{L^h}\Phi^h_{0,j}) - \overline{L^h}\ \Phi^h_{i,j}) + \Phi^h_{i,j} = M(s_{0,j} - \overline{L^h}\Phi^h_{i,j}) + \Phi^h_{i,j} \\
&end\ do \\
&end\ do
\end{aligned}
\tag{44}
$$

where $\overline{L^h}$ is an M-matrix approximation of L^h. If we choose to have i_0 as large as Eq. 44 converges for all j, then it is shown in [102, 103, 104, 105, 106] that $j_0 = 2 - 10$ is sufficient to retain second-order convergence for smooth problems while $j_0 \sim 10-50$ is necessary for unsmooth problems. Note that if only j_0 is small, it is not necessary for L^h to be stable, but if i_0 is chosen to be small (whereby j_0 must be large), the stability of L^h is necessary [38, 106].

There is a system of equations in fluid dynamics, the resulting stiffness matrices of which are usually treated in two ways. In the first concept, all equations for a node are assembled into a collection and the matrix structure is the same as for a scalar problem. Then, instead of a scalar coefficient, we have a sub-matrix of the size $d \times d$, where d is the number of differential equations, see the coupled method Fig. 7. Methods using this structure are the collective GS and the approximative factorization technique, which are sometimes used in compressible flows. The second class of methods number the $d.o.f's$ as corresponding to each PDE in turn, see the decoupled method Fig. 7. However, to be able to use a simple relaxation method, it turns out to be necessary to use some special techniques for



Coupled method Decoupled method

Figure 7: The numbering of the d.o.f's in a coupled or a decoupled method for a system of equations on a structured grid and a stencil with a one-node support.

some specific systems of PDEs, such as the incompressible Navier-Stokes. This is because the equation for the pressure does not explicitly contain the pressure, as this is determined from an extra condition on the velocities. This means that some of the diagonal entities of L^h are zero and may render a relaxation method unstable. One way to circumvent this problem is to relax on the least-square problem $L^h L^{h*}$ where L^{h*} is the adjoint of L^h, but this turns out to be inefficient. Another strategy is to introduce a distribution matrix, B, and a ghost-variable β as

$$L^h \Phi^h = (L^h B B^{-1}) \Phi^h = (L^h B) B^{-1} \Phi^h = (L^h B) \beta^h = f. \tag{45}$$

B is then chosen such that a splitting of $(L^h B)$ according to Fig. 6 is stable. Finally, the corrections in β^h are transformed into corrections in Φ^h as

$$\Delta \Phi^h = B \Delta \beta^h. \tag{46}$$

This class of methods has been widely used for incompressible flows, and many variants of B have appeared. Among the most frequently used methods are the SIMPLE and the Distributive Gauss-Seidel (DGS) method. The SIMPLE [107] method first appeared for an isotropic structured staggered grid, where the discrete equations for the velocities were inserted into the discrete continuity equation. This equation was then truncated such that only the pressure terms remainded and were used in a deferred correction manner with the continuity equation. The resulting equation is referred to as the pressure correction equation. Having relaxed the pressure correction equation, the pressure and the velocities are changed with respect to the pressure corrections.

For incompressible flows, if disregarding the stabilization, the stiffness matrix, L^h, reads:

$$L^h = \begin{bmatrix} Q & G \\ D & 0 \end{bmatrix} \tag{47}$$

and, if the SIMPLE method is identified with Eq. 45, we obtain:

$$B = \begin{bmatrix} I & -Q^{-1}G \\ 0 & I \end{bmatrix} \tag{48}$$

and

$$L^h B = \begin{bmatrix} Q & 0 \\ D & -DQ^{-1}G \end{bmatrix} \tag{49}$$

where Q is the convection-diffusion operator, G the gradient operator and D the divergence operator.

The relaxation of the blocks in $L^h B$ is in practice performed by first assembling the momentum residual $r_u^h = QU^h - s_u$, relaxing $\overline{Q}\Delta U^h = -r_u^h$ and updating U^h with ΔU^h. One must then assemble the continuity residual $r_p = DU^h - s_p$, relax the pressure correction equation, $DG(\mathrm{diag(Q)}^{-1})\Delta P^h = r_p^h$, and update the pressure, P^h, with ΔP^h and the velocities, U^h, with $-\mathrm{diag(Q)}^{-1}G\Delta P$. Note that Q^{-1} is lumped into $\mathrm{diag(Q)}^{-1}$ in the pressure correction equation, DG is the standard Laplacian, and that $\overline{Q}$ is a first-order M matrix approximation of Q. We have used the SIMPLE method in Papers I-IV and Paper VI, but have modified it somewhat to better suit our purposes. These modifications are presented in Paper II, where we suggested a somewhat modified SIMPLE method for buoyant flows using a second order structured FVM, and in Paper VI, where a SIMPLE method for a third-order unstructured FEM is developed.

The second distributive method addressed earlier is the DGS, in which the basic idea is to relax the subproblems according to each PDE such that the residuals in the other subproblems are not magnified [38]. In DGS, B is defined as:

$$B = \begin{bmatrix} I & -G \\ 0 & I \end{bmatrix} \tag{50}$$

and

$$L^h B = \begin{bmatrix} Q & 0 \\ D & -DG \end{bmatrix}. \tag{51}$$

From these pioneering distributive relaxations methods many similar concepts are presented, such as the SIMPLEC, SIMPLER, SIMPLEM, PISO [107, 108] and the ILU-based methods as in [109]. The SIMPLE method without approximations is in fact equivalent to the exact Uzawa algorithm [110]. During the past ten years has inexact versions of the Uzawa method been investigated where $-DQ^{-1}G$ and Q^{-1} has been approximated differently (preconditioned).

The use of a multigrid in evaluating Q^{-1} was presented in [111] and compared to a multigrid with a distributive relaxation method, and the latter was seen to be

several times faster even on moderate sized problems. A couple of preconditioners for $-DQ^{-1}G$ were presented in [112, 113, 114], and they found (not surprisingly) a preconditioner very similar to the standard evaluation of the pressure correction equation in the SIMPLE method to be most efficient.

The inexact Uzawa method can in fact be viewed as a SIMPLE method without updating the velocities by means of the pressure corrections. We tried this approach for some cases using our unstructured FEM-SIMPLE solver and found that if not the updating the velocities we only found that the convergence rate were reduced somewhat. However, as the Uzawa algorithm is most often used with a conjugate gradient method, it is thus usually only applied to Stokes flow or Navier-Stokes in Lagrangian coordinates, and it is thus not considered further in the reminder of this thesis.

However, all distributive relaxation methods usually behave similarly and converge with approximately the same speed. In fact, the distributive relaxation methods are approximately as fast as the fastest minimization methods [115], even on such complicated flows as swirling flows [116]. Our experience is that these kinds of solvers are robust and reliable as long as the underlying models and the discretization are stable. The SIMPLE-like methods are very popular today and are implemented in most research codes and almost all commercial codes.

4.3 Minimization methods

A class of minimization methods are the so-called Krylov-methods. They minimize the quantity $||r_0^h + L^h \Delta\Phi^h||_2$ by seeking $\Delta\Phi^h$ in the Krylov sub-space K_k which is defined as: $K_k = \text{span}(r_0^h, L^h r_0^h, (L^h)^2 r_0^h, ..., (L^h)^{k-1} r_0^h)$.

One Krylov-method is the conjugate gradient method (CGM), which unfortunately works only for symmetric matrices. It can be implemented with only a three-vector recursion, making it very efficient, and has in fact been shown to be competitive with multigrid on coarse grids in conjunction with a suitable preconditioner. The conjugate gradient method is sometimes used in the SIMPLE method on the pressure correction equation when there is a preference to solve and not just relax it within each SIMPLE iteration. This is because it sometimes leads to an increased robustness and speed if solving the pressure correction equation instead of just relaxing it. However, these improvements are, to the author's knowledge, usually not substantial. The conjugate gradient method is also often used when a Pressure Poisson equation is used instead of the continuity equation.

A second Krylov method is the RGMRES method, applicable also to non-symmetric problems. It uses the Arnoldi method to select a basis v^i in K_k such that the resulting matrix obtains an upper Hessenberg form. In this method, one must either store matrix L^h or to compute it in every instance that it is needed (k times per RGMRES iteration). If we have a laminar compressible $3D$ problem and a second-order method, the storage requirement of L^h is roughly 400 words of RAM. However, it turns out that the RGMRES method must be preconditioned to work satisfactorily and it seems that the best choice (beside multigrid [117]) is

MILU [29, 118, 119, 120, 121]. A precondition matrix must then also be stored, whereby another 400 words must be stored. In addition, we need to store the search directions in the RGMRES, which may be about 30, thus requiring another 150 words. The rest of the code requires at least 50 words and thus a total of 1000 words of RAM requirements, which is too much for many applications.

Most often, the preconditioned RGMRES is used in combination with a Newton method, and thus the number of evaluations of the preconditioner is reduced. In the RGMRES method the work is basically the evaluation of the search directions. Thus the work in one restart with 30 search directions of the RGMRES is of the size of 30 residual evaluations. In practice, 30-100 restarts are typically required which correspond to work comparable to 900-3000 iterations with a distributive relaxation method [29, 101, 115, 122].

4.4 The multigrid method

The final class of methods is the multigrid concept, [38, 96, 123] which has appeared in a large number of variants and has been applied to both structured grids [61, 71, 98, 124, 125, 126, 127, 128, 129, 130] and unstructured grids [28, 30, 40, 131, 132, 133, 134, 135]

Multigrid is based on the idea that, when using a relaxation method (or a minimization method [115, 117]), high-frequency errors are usually reduced rapidly and independently of N, so that the error is smooth after a few sweeps. Smooth means that the components with a wave length comparable to the grid size are small compared with other components. It is also usually recommended that the smooth part of the error not being amplified during the relaxation, even though this is not necessary.

This smooth error corresponds to a smooth residual, r_i^h, and continuing to relax on a smooth residual makes sense only if it can be done efficiently. However, as all local averaging methods have a very slow convergence for the smooth part of the residual, they are not well suited for that. In fact, they usually have a rate of smooth error reduction of $\rho_s \sim 1 - O(h^2)$, where h is the grid size, while they perform well in reducing the non-smooth part of r_n^h, often with an error reduction rate of $0.2 - 0.7$. If we more precisely define a non-smooth error as an error with a wave length of less than $4h$ we see that, on a set of structured grids successively coarsened by a factor of two, all error components will be viewed as non-smooth on one of the grids. The drawback, however, is the aliasing phenomenon. This means that a non-smooth error, if projected on a grid that is twice as coarse, will be aliased as a smooth error, since the non-smooth component is not representable on this coarse grid.

The multigrid idea is therefore to relax each error on the grid where it is considered non-smooth. To spectrally decompose the residual would be both expensive and difficult to generalize, and a recursive strategy is thus used instead. One must first start on a fine grid Ω_h and perform a few relaxations, which removes the part of the residual considered non-smooth on this grid. Then the remaining

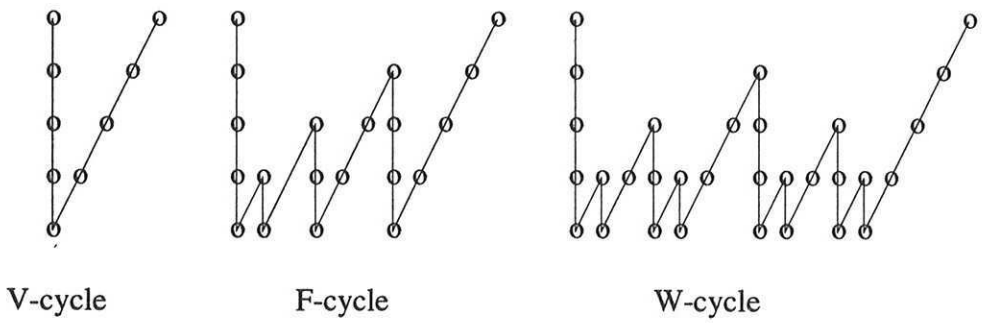


Figure 8: Three different cycles often used.

part of the residual is projected on the grid that is twice as coarse, Ω_{2h}. Because the residual was smooth on Ω_h, the aliasing problem is avoided. Some of the components that were considered smooth on Ω_h are considered non-smooth on Ω_{2h}, and are consequently efficiently removed by relaxation. The remaining part of r^{2h}, which is considered smooth on Ω_{2h}, is now projected onto the next coarser grid, Ω_{4h}, and so on.

We now have a set of corrections, $\Delta\Phi^{2^k h}$, appearing on a sequence of grids, Ω_h,Ω_{2h}, Ω_{4h},..., $\Omega_{(2^{k-2})h}$, $\Omega_{(2^{k-1})h}$, $\Omega_{2^k h}$, and we must project them back onto the original grid, Ω_h. Starting on $\Omega_{2^k h}$, projecting $\Delta\Phi^{2^k h}$ onto $\Omega_{(2^{k-1})h}$, we will eventually introduce some non-smooth components, but they are efficiently removed by relaxation. The total correction on $\Omega_{(2^{k-1})h}$ is then projected and added to the old corrections on $\Omega_{(2^{k-2})h}$, where non-smooth components resulting from the projection are removed via relaxation. This procedure is referred to as a $V(\nu_1, \nu_2)$-cycle, where ν_1 represents, the number of pre-relaxation sweeps, i.e. the number of relaxation sweeps on a grid before the residual is transferred to a coarser grid, and ν_2 represents the number of post-relaxation sweeps before the corrections obtained on this grid are projected onto a finer grid. At the coarsest grid, a sufficiently large amount of relaxation sweeps or a direct inversion is performed. However, other choices of cycles exist, such as the W-cycle and the F-cycle, which are also frequently used, see Fig. 8.

Owing to the fact that L^h is approximated differently on the different grids and to the incompleteness of the projections, the error is not removed within one cycle, whereby the cycle is repeated until convergence is achieved. Note that we are now actually performing a small number of relaxation sweeps on each of the grids and, as $\dim(V_{2^k h}/V_h) = 2^{-3k}$, the relaxation work on coarse grids becomes almost negligible. Table 1 shows the total amount of relaxation work on all grids transformed into an equivalent number of relaxation sweeps on the finest grid (:=WU). Note that a fixed number of relaxation sweeps, $\nu = \nu_1 + \nu_2$, is used on all grids.

The multigrid method is slightly non-optimal, $work \sim O(NlogN)$, however,

	2D	3D
V-cycle	1.33 ν	1.14 ν
F-cycle	1.79 ν	1.31 ν
W-cycle	2.00 ν	1.33 ν

Table 1: The amount of work per cycle in terms of WU. (Isotropic global refinement by a factor of two)

and thus the concept of full multigrid (FMG) is introduced, which in fact is optimal, *work* $\sim O(N)$. The FMG method starts at a coarse grid, $\Omega^{2^k h}$, and when $\|r_i^{2^k h}\| < \|\tau^{2^k h}\|$, the approximation is projected onto the next finer grid, $\Omega_{2^{k-1}h}$, and a two-level cycle is used until $\|r_i^{2^{k-1}h}\| < \|\tau^{2^{k-1}h}\|$. $\Phi^{2^{k-1}h}$ is then projected onto the next finer grid, $\Omega_{2^{k-2}h}$ and a three-level cycle is performed.

To achieve the crucial smoothing the design of the relaxation method is of paramount importance when developing a multigrid method because, as has already been discussed, the relaxation must, at the same time as being cheap, have a good grid-independent, non-smooth error reduction rate, ρ_u, i.e good smoothing properties. That is a difficult task and has been discussed in detail in [38, 96, 123].

To now, we have not made any mention about of the projections. Define the projection from a fine to a coarse grid as the restriction I_h^{2h}, a projection from a coarse to a fine grid as prolongation I_{2h}^h and the projection from a coarse to a fine grid in the FMG as the FMG prolongation. Linear projections are most widely used for the Navier-Stokes equations, and our experience is that they work best for fluid flow computations.

Note that in this work we only consider isotropic refinements, even though it in some situations can be more efficient to use directional refinements or coarsening [136].

4.4.1 Some details in the development of an FMG

This subsection presents some basic ideas and recommendations about when to construct an FMG method [38, 96, 106, 123, 137]. These are often based on some heuristic arguments combined with a more detailed explanation or proof of a simplified problem. However, more rigorous theory employing arbitrary grids and full representation of boundaries can be found in [138, 139]. We will in the remainder of this subsection restrict our discussion to second-order elliptic operators, such as the Poisson equation and the incompressible Navier-Stokes equations.

First consider the Laplace equation discretized on a uniform 2D Cartesian grid and use the GS relaxation method, in which, the non-smooth errors are reduced by a factor of 0.5 each relaxation sweep. This means that, after two relaxation sweeps (2 WU), only 25 % of the original high-frequency error remains. If

we assume cheap and sufficiently good projections and use a V-cycle, we will need a further (0.67WU) to reduce the smooth error by 25 %.

We see above that it is the non-smooth part of the error that is most expensive to remove, (2WU) compared to (0.66WU), and thus we need a prolongation operator such that the projection of smooth corrections does not produce non-smooth projection errors of a similar size as the smooth corrections. This means that at least a linear prolongation is needed, while injection is sufficient for the restriction operator. However, linear restriction is recommended for more general problems because, if r_h is somewhat non-smooth on the fine grid, the most of the non-smooth part is then filtered out and left on the fine grid. When doing so, most of the non-smooth part of the fine grid residual is not aliased into smooth parts on the coarse grid, which would wrongly produce smooth corrections on the coarse grid.

The use of higher-order projections becomes very expensive, and thus it is most flexible and efficient to use a linear prolongation and to perform a small amount of post-relaxations, removing the non-smooth residuals owing to the prolongation. In addition, approximations in the coarse grid operator make it difficult to realize the high error reduction implied by high ν, and small values (1-5) of ν_1, ν_2 are thus usually selected with linear projections. Furthermore, simple projections are much easier to implement, and simplifications may also be admitted near boundaries etc.

The FMG-prolongation, however, operates on the full approximation, U^h, and not the correction. Since the residual should not be dominated by the projection errors of the smooth part of U^h, the FMG-projection must be in the order of $O(h^{p+\alpha})$, i.e. cubical interpolation for a second-order, five-point central difference stencil of the Laplace equation ($p = \alpha = 2$). Note that the error in an initial approximation prolongated from the coarser grid onto this finer grid is both smooth and non-smooth, and it is thus not sufficient only to relax, but it is necessary to use cycles.

To sum up, we can roughly say that we should be able to solve a PDE within work comparable to six to seven work units (WU), where a WU is work comparable to only a few flops per d.o.f. To realize that, a cheap relaxation method with good smoothing properties is required. Finding such a relaxation method has proven to be very difficult for the general case, while, for the Laplace equation, most relaxation methods smooth with an acceptable rate, resulting in an FMG convergence in about 25 WU. However, for the stationary incompressible Navier-Stokes equations, there is a large hyperbolic influence, a mixed system, non-linearities and turbulence etc., making the design of relaxation methods difficult. In addition, in practice, the preference is to be able to use highly stretched or unstructured grids to resolve layers or complex geometries, increasing the complexity.

The non-linearities are most efficiently handled by Picard iteration (principal linearization) and, since we perform a large number of relaxation sweeps (mainly on coarse grids), we have the possibility to frequently update the non-linearity. That is realized if we also restrict the approximation, Φ^h, from the finest grid and

use it as a starting point, $\overline{\Phi^{2h}}$, on the coarse grid, whereby the changes $\Phi^{2h} - \overline{\Phi^{2h}}$ represent the corrections produced on the coarse grid, Ω^{2h}. This also allows us to use the same stencil as on the finest grid when evaluating L^{2h} for a non-linear problem on a coarse grid. This technique is referred to as the Full Approximation Scheme, (FAS) [38, 96], and is today used in almost all multigrid formulations.

The layers and the hyperbolic influence of the Navier-Stokes equations were discussed in Section 2. We noted that a good stability is needed to prevent spurious oscillations. Thus, when improving the stability, the odd-even decoupling decreases and the ellipticity on the scale h increases, resulting in an increased smoothing rate of a relaxation scheme [38]. Distributive relaxation is, as mentioned earlier, well suited for some systems of PDE's. Such a relaxation is constructed with the purpose of giving a triangular L^hB, and it can be shown that, if each diagonal block of L^hB smooth well, L^hB smooth well [38, 109, 140, 141]. This is ensured in Papers I-IV and Paper VI by replacing L^hB with $\overline{L^hB}$, in which the convective terms for the velocities are discretized via upwind and a scaled Laplacian is used for the pressure.

4.4.2 Multigrid in practice

When constructing a multigrid method, we also need to specify a grid hierarchy. Conformal nodes are usually used in an FDM multigrid on structured grids. This is a natural concept, as FDMs are based on gitter points. However, in an FVM, the underlying idea is local flux conservation over each cell, and conformal cells thus seems more natural in an FVM. In an FVM can the residual be regarded as flux imbalance or simply a source term integrated over a cell, Ω_j, and a conservative residual restriction is therefore suitable. In the case of conformal cells, this may be accomplished by a summation of the fine grid residuals into the corresponding coarse grid residual. The prolongation would then, according to our idea of conservation, be injection. However, as it is of too low order, linear prolongation is chosen, which is also conservative (in terms of Φ). In FEM, the L_2 projection is the natural choice [138, 139, 142], and thus the prolongation of Φ^{2h} reads:

Find $\Phi^h \in V^h$ such that

$$(\Phi^h - \Phi^{2h}, w^h) = 0 \;\; \forall w^h \in V^h, \tag{52}$$

or, in matrix notation,

$$E^h(\Phi^h - P\Phi^{2h}) = E^h\Phi^h - C\Phi^{2h} = 0. \tag{53}$$

In the case of conformal cells, $V^{2h} \subset V^h$, whereby P is simply the nodal interpolant. For the restriction, we first note that the residual, r^h, can be viewed as an L_2 projected residual field b^h, as:

$$E^h b^h = r^h \tag{54}$$

Analogously, as in the finite volume method, it is this residual field that should be projected onto Ω_{2h}. That means:

$$E^{2h}b^{2h} - C^T b^h = 0 \Rightarrow r^{2h} = C^T E^{-h} r^h = P^T r^h \tag{55}$$

Thus the restriction operator is the transpose of the prolongation operator which, in the case of conformal cells, is the transpose of the nodal interpolant. We have this far only discussed symmetric projections; however, for hyperbolic problems, there might be a gain in having upwind-downwind directed transfer operators as suggested in [106]. Later on in [143] for hyper-sonic flows it turned out to be significantly more robust to use upwinded transfer operators. However, we have not considered this approach in this work.

While conformal nodes or cells seem natural, they have the drawback that the boundary resolution and the grid quality degenerates. We believe that these problems can be avoided by special boundary adaptivity, smoothing and edge-swapping [134] (having $V^{2h} \subset V^h$ only approximatively), but it has still forced the development of other methods. One choice is to use a set of independently generated grids [30, 143, 144], but this has the drawback that it requires complex search routines for the transfer operators. It also makes the adaptivity more connected to the grid generation, slowing down the solution process, and makes it more difficult to modularize a code.

Another approach is to use an agglomeration technique [28, 135] where the coarser grids are constructed from the finest grid. This method is suited for a finite volume method on a first-order hyperbolic system, while diffusion terms are difficult to realize. However, a fix presented in [135] improved the performance for the diffusion terms, although is still not as good as for conformal grids [28]. The major advantage of this method is that the problems of the degeneration of boundary resolution and grid quality are avoided; it has a penalty, however, in increased algorithmic complexity and poorer performance for diffusion operators.

The most general method is the algebraic multigrid in which the sequence of subproblems is generated directly from the stiffness matrix. Unfortunately, it has a severe penalty either in robustness or CPU and RAM efficiency, and is usually applied only to such small scalar equations as the Poisson equation [137, 145].

However, for complicated problems, such as the turbulence-modeled Navier-Stokes equations, it is difficult to achieve the ultimate goal of six to seven WU, even if such performance has been demonstrated in some cases for both compressible and incompressible Navier-Stokes [141, 146, 147]. These results are based on algorithms which, to the author's knowledge, have been performed only in some special simple flow situations.

As it is difficult to extend these concepts for more general situations, a more realistic goal is convergence within 30 WU instead. This is because the required cubical FMG-prolongation is difficult to implement in the general case and is most often replaced by a linear projection. However, FMG-projection then introduces errors that are visible in the residuals as $O(\max(\mu U, U^2 h))$. Our experience is that, on a practical fine grid, we need about 10 to 20 relaxations to reduce the

non-smooth errors induced by the linear projection. Furthermore, the artificial ellipticity added to obtain stability on the scale h reduces the two-level rate of convergence to a maximum of 0.5 per cycle, independent of the choice of ν, instead of the 0.06 obtained for the Laplace equation with RBGS ($\alpha = 0.25$) and $\nu = 2$.

In [140] it is shown that the SIMPLE method have a smoothing rate of 0.7 when $Pe := Uh/\nu \sim 10$ (staggered grid and O(h) discretization). When using an eddy viscosity turbulence model Pe may be estimated as:

$$Pe = \frac{Uh}{\nu_t} = \frac{Uh\epsilon}{C_\mu k^2} \sim Uh\frac{0.16(0.1U)^3}{C_\mu(0.1U)^4 l} = 18\frac{h}{l} \sim 5 - 10 \tag{56}$$

This means that the SIMPLE solver should have a sufficient smoothing rate. However, in [146, 147] it is stated that for low-order cycles the coarse grid corrections degenerate, in the limit of infinite number of grid-levels when $Pe \to \infty$. For $Pe \sim 10$ and a reasonable number of grid-levels (~ 5) we do not believe that this problem is significant but as we on the other other hand have we noticed the F-cycle to be somewhat more efficient for some problems this effect might have a small influence.

To sum up, we mean that the goal of reaching convergence within six to seven optimal WU may be replaced by a convergence within three to five cycles, whereby, in practice. We will then most likely need about 30 WU when solving a CFD problem below the level of the truncation error.

In the literature, the work required to reach convergence below truncation errors of different implementations of the multigrid method usually varies from ~ 10 WU up to almost no speed-up at all. The performance of about 100 WU is usually presented for sub-sonic flows in simple geometries with Reynolds numbers of $Re < 0.1Re_c$, where Re_c is the Re where the flow becomes turbulent, while, for $Re \sim 0.9Re_c$, the performance usually varies from 100 up to 1000 WU [61, 68, 98, 125, 126]. These results are usually done with a distributive relaxation method (DGS smoother). In sub-sonic flows for the Euler or the Navier-Stokes equations, a multigrid with a Runge-Kutta smoother most often needs 1000 WU or more [28, 30, 132, 133].

In the few cases that present the application of the multigrid to the turbulence-modeled Navier-Stokes equations and/or on unstructured grids, they typically convergence within $1000WU$ [124, 127, 148].

In summary, the multigrid method in practice seems most often to be 10 times more efficient than any other method.

An approach often used is to only relax instead of cycling on each level. This method can wrongly be mistaken to have a similar performance as the multigrid method. In the following example, we show that there is no asymptotic speed-up but, in practice, a CPU reduction of a factor of two or three can be obtained as compared with starting from scratch on the finest grid. As we wish to solve the algebraic error just below the level of the discretization error, the error in the initial approximation on the finest grid must be reduced by a factor of four (for an $O(h^2)$ discretization) instead of a factor ϵ. Thus we obtain a speed-up of only

a factor $\log(\epsilon/4)$ when using the FMG-prolongation and relaxation as compared with only using relaxation.

In practice, we prefer to have an error reduction, ϵ, of the size of 10^{-3}, resulting in a speed-up of 2.4. This typically means in practical CFD that we have reduced the work from $10000WU$ to $4000WU$ and, compared with our goal of $30WU$ for a multigrid method, the factor of 2.4 is almost negligible. A factor of two is of course still an improvement, but it is probably more important that the FMG-prolongation may increase the robustness since a rough estimate is obtained on a coarse grid, which has more dissipation and is usually more stable.

5 On the choice of CFD method

The choice of model, discretization and solver is an open question and will probably remain so some time. However, we would like to offer our opinions based on our experience of our preferred methods and why we prefer them.

The choice of model is closely related to the application and, if only a few applications are considered, an optimized two-equation model or RSM is most often found. Recently, it also seems that the problems with the unphysical, normal strain production of turbulence in a two-equation model may be avoided with a non-linear strain-rate Reynolds-stress coupling, still retaining most of the stability and simplicity of a two-equation model [5, 7].

Selecting a discretization is difficult and depends on the purpose of the code. If a very simple code is desired and only simple geometries are considered, a structured grid is an obvious choice. If a mapping and a multiblock discretization are used, a larger class of applications may be accessed, while general geometries require an unstructured grid. A great amount of work has been performed for both compressible and incompressible flows in developing stable $\sim O(h^2)$ accurate schemes for smooth grids, but higher order methods seem difficult to realize in an FVM formulation. In an FEM, an higher order method $O(h^p)$, $p \geq 3$ is more easily achieved. In addition, FEM quite easily accounts for boundary curvature.

Paper VI compares the QUICK-FVM on a Cartesian uniform structured grid, with a second-order SUPG FEM on a homogeneous isotropic, unstructured grid with equally many nodes. We achieved slightly better results with the FEM, even though QUICK is formally third-order accurate in one dimension and even though a quadrilateral has a lower approximation error than a triangle. We would thus be very surprised if a more detailed comparison of the SUPG-FEM and the most common FVM schemes, would show the FVM schemes to be superior to SUPG-FEM on mapped structured grids in incompressible flows. In [149], it was shown for potential flows that the FEM is superior to the FVM (especially on the boundaries).

In incompressible flows, the need for stabilization of the velocities is not as strong as in compressible flows, and it is unclear whether the FEM-SUPG is competitive with FVM on structured grids and compressible flows. We have compared the results of SUPG with FVM publications on unstructured grids and incompressible flows. We have not yet seen any results that are comparable with SUPG, and the discrepancies are often large, as when compared with the results in [27, 150, 151]. For unstructured grids and compressible flows, almost all work is performed with FVMs on the dual grid (or standard FEM + artificial viscosity). In [34], second and third-order FVMs with linear mapping based on three or six-node triangles were presented. These methods still had simplifications near boundaries and was not able to account for curved boundaries, and they were no more than only reasonably complex to extend into 3D. They were compared to the $O(h^2)$ SUPG-FEM method presented in [152], and the $O(h^2)$ FVM performed somewhat more poorly than the FEM. The $O(h^3)$ FVM performed somewhat better, but

the $O(h^3)$ FVM was too expensive and not competitive in terms of CPU time. It should be mentioned that the FEM was roughly implemented, and a better choice of the perturbation parameter would probably show better results and remove the small spurious oscillations shown in [34]. They also stated that the SUPG works surprisingly well, despite the fact that it is so simple and straightforward.

On the basis of this and our experience with incompressible flows, we believe that SUPG is the most simple and most efficient discretization for both incompressible and compressible flows on unstructured grids. However, we also believe that improvements are expected in the choice of δ [153].

The choice of solver seems simple, as the only general optimal method is the full multigrid. In practice, it also seems that multigrid most often is a magnitude or two faster than any other method, and our experience is that multigrid is the most efficient solver. To support this, however, we will review some results presented in the literature. For compressible Euler [29] presented a method based on Newton linearization and a MILU preconditioned restarted GMRES method, claiming that in one application it was only a factor of three slower and used only about 600 words more RAM than the multigrid by Mavripilis [28]. In addition, this minimization method was claimed to be a magnitude faster than an explicit time-stepping relaxation method. The same experience was reported in [45].

For low-Ma flows, the distributive relaxation methods appear to be about a magnitude faster than the time-stepping methods (even with preconditioning [154]). For subsonic flows a distributive relaxation method will use the same number of iterations as a minization method ($\sim$ 1000). On a structured grid the evaluation of the stiffness matrix is as expensive as the relaxation scheme. Thus is it no significant gain in storing the stiffness matrix.

However, on an unstructured grid, we found in Paper VI that the major part of the work was typically dedicated to assembling the stiffness matrix and, thus, if the stiffness matrix is stored, a speed-up of a factor of approximately five is achieved. This means that, if a preconditioned RGMRES method stores the preconditioner and the stiffness matrix ($\sim$ 1000 words RAM), it must on unstructured grids be compared to a relaxation method storing the stiffness matrix ($\sim$ 400 words RAM). Doing so, it appears as though both minimization and relaxation methods have about the same speed, i.e. at least 10-100 times slower than a multigrid method (also storing the stiffness matrix). Implementing GMRES in a matrix-free fashion [122] showed a penalty of a factor two in comparison with the standard method on structured grids, and we expect the penalty to be significantly higher if unstructured grids are used.

There has been much discussion as to the amount of work incorporated in a specific discretization method. In particular, an extension of an $O(h^2)$ method to an $O(h^3)$ one increased CPU time by a factor of two for our FEM method in Paper VI and in the FVM [34] by a factor of eight. However, when implemented in a deferred correction manner with a lumped first-order upwind based an as simple a discretization as possible, and not updating the defect more than, say, every 10-100 iterations, this penalty is removed. This technique may also possibly

be implemented in a multigrid method, updating the defect, say, every fifth cycle.

If implemented in such a manner, all methods with sufficient stability, second, third, and perhaps even fourth-order accurate FVM or FEM, should use approximately the same amount of CPU time. Thus it is mainly the accuracy and stability of a discretization that is of importance, which has been verified in explicit tests on a higher order FVM on both 1D and 2D model problems [155]

6 Summary of papers

This work was primary initiated to develop a local mesh refinement method and a multigrid method for turbulent flows on collocated structured grids. A similar study was made in [156] using staggered Cartesian grids but, in order to handle complex geometries, we employed the collocated grid formulation. Thus, in the first paper, we developed a multigrid formulation for turbulent flows and collocated grid using a preliminary study [157].

However, during the development of the local mesh refinement method, we found the method in [158] to be unstable for buoyant flows. This pathological behaviour resulted in the work leading to Paper II, in which we presented some modifications improving the performance of a collocated SIMPLE method applied to buoyant flows. At this stage, we found it too complicated to work further with the CALC-BFC code [158] and wrote a new code PEC-FLTBB [159] from the multigrid structure in Papers I-II, and implemented an interface treatment for the local mesh refinements. The local mesh refinement algorithm was then presented in Paper III and could optionally be used with or without the multigrid method. In Paper IV we used PEC-FLTBB to analyze some different inlet approximations and found that such approximations may have quite a large influence, on the flow in the whole domain.

As we were satisfied with the performance of this multigrid concept (typically convergence within $\sim$ 100 WU), we decided to develop an unstructured version. In particular, we chose to work with a second and a third-order accurate adaptive FEM, and this was also observed to converge typically in $\sim$ 100 WU and we present this work in Paper VI.

During the work in Paper VI, an explicit fractional step LES method was developed [160] but this turned out to suffer from slow convergence. That initiated a study on developing an implicit fractional step method and a multigrid solver for the pressure Poisson equation. This method was found to be 20 times faster than the method in [160] on a 1 000 000-node mesh. The work done in this study is presented in Paper V.

The remainder of this section summarizes Papers I-VI. They are presented with some notes on their purpose and background, the most important results of the work are briefly presented and some general comments or corrections are finally given.

6.1 Paper I

6.1.1 Purpose and background

A numerical simulation of a ventilated enclosure often requires large 3D domains with a large number of nodes. It then becomes very expensive to use such standard solvers as the SIMPLE method or time-stepping, as they roughly require $O(h^{-2})$ WU to reach convergence (h is the grid size).

However, the multigrid method shows optimal $O(1)$ behaviour and uses only a small amount of extra memory. In a series of publications [38, 98, 126], we found that the multigrid method had been successfully applied to the laminar incompressible Navier-Stokes equations with staggered grids. These authors used conservative velocity projections between the different grids. For collocated grids, an extension of that concept via an introduction of ghost fluxes was used in [124, 125, 127, 157, 161]. Such ghost fluxes are difficult to implement, however, and also difficult to generalize to more complex grids.

At the time of this work, we found only a very few turbulent simulations that used the multigrid method. In [127, 161], poor performance was shown, although some useful guidelines for the multigrid formulation were given. Good performance was reported in [124, 129] but no details of the implementations were given.

6.1.2 The work and results

We extended the CALC-BFC code [158] with a multigrid solver. This is based on a consistent multigrid formulation relying on the definition of the multigrid method presented in [96]. It implies a general straightforward multigrid formulation of the continuity equation which is often used in compressible flows. This circumvents the need for the complex conservative interpolations or ghost fluxes.

We found that special care was needed with respect to the under-relaxation of the turbulent viscosity and the scaling of the pressure dissipation. These were recursively implemented in the standard collaced SIMPLE algorithm [56, 158]. In [127], a modification to the multigrid formulation of the k and ϵ equations was given, which was a non-linear damper of negative corrections on $k - \epsilon$, which smoothly prevents $k - \epsilon$ from being negative. However, we found that to be insufficient and therefore also separated the locally negative multigrid source terms in $k - \epsilon$ from the positive ones, in a manner similar to that of the standard implementation of the turbulent production and dissipation. This improved the robustness significantly, and we obtained a multigrid convergence rate for turbulent flows similar to that for laminar flows, i.e convergence within ~ 100 WU

With FMG, we achieved for both the laminar and turbulent flows approximately grid-independent convergence typically 100 times faster than the corresponding single-grid SIMPLEC solver. We saw no decreased robustness but rather an increased robustness using the FMG, compared with the use of the single-grid SIMPLEC solver. Approximately the same speed and robustness were achieved for both a first and a second-order discretization.

6.1.3 Comments

Our separation of the turbulent multigrid source terms results in a decreased dissipation during the first iterations on a coarse grid, whereby a somewhat higher turbulent viscosity will appear during these first iteration, which may explain the increased robustness. However, the continuation process in the FMG is also

stabilizing. If, on the other hand, we sorted all parts of the multigrid source term into negative and positive contributions, we nearly kill the iteration process on the coarse grid, resulting in poorer convergence. Another technique was later presented in [69, 148], where the slope of $\partial_t(k, \epsilon)$ was limited. This method is closely related to our non-linear damper. However, [148] also fixed μ_t on coarser grids and showed quite poor speed, (2000 V-cycles on a flat plate $Ma = 0.2$). When we tried to use a fixed μ_t, we found only a decreased speed and lower stability.

In this paper we also recommend the use of inlet profiles with the same mass flux on different grids. This has later been found not to be necessary but, as a boundary condition is used in FAS to evaluate the stiffness matrix, we should not use completely different profiles on the different grids. Our multigrid formulation of the continuity equation was mentioned in [162], and they claim without explanation that it has a built-in inconsistency of 2 %. This is of course incorrect, and it should be mentioned that our formulation has proven to be consistent to double machine precision, see Paper I. However, if special care is not taken for the pressure dissipation, we also obtained an inconsistency of about 1%.

6.2 Paper II

6.2.1 Purpose and background

In ventilation applications, we wish to predict buoyancy affected flows. It was found to be very difficult, however, to achieve convergence with the code CALC-BFC on buoyancy-driven flow. Buoyancy-affected flows are usually considered to be more difficult than isothermal flows [27, 163], but not as difficult as we experienced.

In [163], some modifications were presented that were based on a global initiation of the pressure, or the introduction of dissipation of the density in the continuity equation. We tried these modifications without any improvement.

6.2.2 The work and results

The essence of our paper is that we found the standard way [56, 158] to implement the necessary pressure dissipation in combination with a SIMPLE may be unstable for buoyant flows. This is probably because, in the standard implementation, the scaling of the dissipation stencil were evaluated locally for each term in the stencil. That actually results in a modification of the weights, and the stabilization property may then be lost.

We thus simply used the same scaling of all terms in the stencil, which resulted in a more stable behaviour, and this in fact is the original method suggested in [55]. In addition, we removed the recurrence in the dissipation which, required special treatment in the multigrid, see Paper I. We also removed the inconsistency in the original scaling of the dissipation with respect to the under-relaxation factor. Finally, we simulated a couple of cases both with and without the multigrid with good performance and accuracy.

6.2.3 Comments

Most of the problems that still may occur are related to the G_k term in the $k-\epsilon$ equations. When that is switched off, the robustness is increased and the CPU-time reduced significantly . That is also what is recommended in [27] when problems occur.

The paper listed two further modifications. The first was to update the defect in the continuity residual while relaxing the pressure correction equation. The positive results indicated in the paper have later been shown to be case-dependent and, in some cases, a negative effect has even been found. The second modification is simply another scaling in the pressure correction equation, as the SIMPLEC. As expected, we did not see any significant differences in the convergence rate, as we pointed out in the paper.

6.3 Paper III

6.3.1 Purpose and background

In ventilation, there are usually inlets that are small in comparison with the domain. In addition, they are located in the middle of a wall, which causes difficulties when constructing grids, and one way to solve this problem is to use some kind of local grid refinement. Local grid refinements use structured grids within each part of the grid and refine some regions. These refinements can be based on overlapping grids, internal boundary conditions or special discretization stencils.

With the use of overlapping grids, the flexibility in constructing grids is great but it is difficult to retain accuracy and conservation in the overlap region. Matched interfaces, on the other hand, can preserve conservation and accuracy but are not as flexible as overlapping grids. In this work, we chose to work with internal interfaces, which may be implemented as internal boundary conditions or flux stencils. Internal boundary conditions are usually used when using staggered grids, ensuring conservation of mass and accuracy [38, 98, 99]. However, for collocated grids, the crucial conservation in addition with stability is more difficult to achieve, and special stencils over the interfaces are used instead.

6.3.2 The work and results

We used the same concept as in [65] to achieve a stable interface discretization. We introduced the refinement by splitting the cells in a region by a factor of two and introducing fine ghost cells on the coarse side of the interface, using them in the stencil for the fine cell face fluxes. These fluxes are then conservatively assembled into the coarse cell's residuals.

The variables in the fine ghost cells are obtained via linear interpolation from corresponding coarse cells. We evaluated the QUICK and Hybrid-Upwind schemes with Lagrangian polynomials, while the dissipation was evaluated in a stencil accounting for the sudden jump in grid size of a factor two at the interface.

If this sudden expansion of the grid is not accounted for in the stencils of QUICK and the pressure dissipation, it may give rise to oscillations in the pressure near the interface, as was seen in [151].

The solver is the FMG-SIMPLEC method described in Papers I-II. However, the grid hierarchy was chosen such that all grids cover the whole domain, which means that regions refined only a few times will be smoothed at all finer levels. Such extra smoothing will not pay back in fewer iterations, but we estimated the slowdown to be a factor of two at most in practice.

This choice of grid hierarchy provides smoothing over the interfaces, which increases the robustness which is important for complex flows. As a by-product of the multigrid, we obtain an estimate of the local truncation τ^h indicating the regions to be refined. However, as we use an order of the projections that is too low, we most likely monitor second derivatives instead. On the backward-facing step, we showed the power of the concept of local grid refinements in combination with a multigrid solver, i.e. a significantly reduced number of nodes and convergence within $\sim$ 100 WU. We also used the local grid refinement technique to resolve a complex inlet diffuser in a buoyant, turbulent, ventilated enclosure.

6.3.3 Comments

The inlet diffusor consisted of 84 jets and was geometrically representable only on the finest two levels. The approximative inlets on the coarse grids most likely gave too poor a quality of the corrections, and thus the speed-up of the multigrid was small.

The CALC-BFC code [158] with the multigrid extension presented in Paper I-II was based on a standard 3D-stencil with a complex BFC part, and it turned out to be faster to start from scratch than to extend CALC-BFC with local grid refinements.

The new PEC-FLTBB code [159] employs a second-order 1D flux calculation stencil (for the convective and the orthogonal diffusion terms), and these fluxes are distributed to cell residuals, used in a deferred correction manner with a first-order upwind scheme. This strategy allows a very simple coefficient evaluation over the interface without sacrificing accuracy, stability, robustness or speed.

6.4 Paper IV

6.4.1 Purpose and background

This work focused on the validity of the different approximations of the inlet diffusor in annex20. The purpose of these approximations were to reduce the complexity of the inlet (84 nozzles) and to avoid the required resolution, which is expensive even with local grid refinements.

The three inlet approximations we analyzed are suggested in [164] and are described briefly as:

IA1 Full resolution of all jets.

IA2 The basic model, in which all jets are collected into the linear center of the diffusor plate.

IA3 The momentum method, in which, the jets are replaced by a decoupled flux boundary condition, i.e. the total mass and momentum flux are distributed over the diffusor plate.

6.4.2 The work and results

We compared simulations of a calculation using the full resolution method and a calculation using the basic inlet model. The results differ significantly and, by a simple global analysis, we present an indication of the reason for the differences. Briefly, the indication states that the collection of the jets reduces the diffusion and the jets are consequently preserved longer.

However, more sophisticated global analysis of wall jets correlates more poorly with our differences than does our simple model. We thus believe that our good correlations were somewhat lucky and that the analogy of a free jet is not good in our application [165]. We also gave motivation as to why the momentum method is inconsistent and why it almost never can be used.

6.4.3 Comments

Other approximative inlet boundary conditions were used in [164]. For example, inlet jets were collected into groups, which are easier to resolve. Such a method will of course be better the more groups that are chosen.

In [164], another inlet approximation choice is given. There the whole domain is then simulated except the inlet region, where data along this fictive boundary are measured instead. This requires measurements of the inlet, and the applicability will greatly depend on the feedback from the rest of the room, how the conditions in the simulated room correspond to the conditions in the measured one and the quality of the measurements. Because of these limitations, we have not considered this inlet condition in this work.

6.5 Paper V

6.5.1 Purpose and background

Large eddy simulations represent a promising concept in turbulence modeling as they are consistent with the Navier-Stokes equations in the limit of zero grid size [13, 20]. We have in this work used the dynamic subgrid model [15, 16], and a spatial discretization is presented and validated in [160]. The purpose of this work has been to develop as fast a solver as possible for stretched grids.

This means that we have not considered any issues concerning the validity of this model and space discretization, such as that both aliasing and discretization

errors are probably larger than the whole subgrid model, as we use filter size and spatial discretization of the same size [166, 167]. In addition, no attention is paid to analyzing what kind of grid is suitable for these applications. That means that we used the grid-distributions given in [160]. These grids have a maximal aspect ratio of about 100, which must be accounted for in constructing the solver. We have thus in this work focused on developing as efficient a solver as possible for this class of problems, as they are typical for LES applications [168].

We must have sufficient resolution, and we need to run the simulation long enough to obtain a stable average while we simultaneously have a sufficient resolution in time. It seems that, with the subgrid models of today, the space and time resolution required to achieve better results with LES than with eddy viscosity models are significant. When using a fully explicit time-accurate formulation, the time step becomes very small. Even when using an implicit pressure equation, the time step may become a factor of $\sim 3 - 10$ smaller than is required by accuracy [169].

Thus, if using an fully implicit method or only an implicit pressure equation, the efficiency of the matrix solver is crucial if one should benefit from the longer time steps admitted.

6.5.2 The work and results

We briefly investigated the influence on the mean flow and turbulence with respect to the time step for our applications in which we use stretched grids near walls. As this near-wall stretching put severe restrictions on the time step for an explicit method, we attempted to speed up the calculation by using longer time steps. In fact, it was found that we could increase the CFL number from 0.3 to 2.0 with retained accuracy.

We used a pressure Poisson equation in order to avoid variable coefficients when using a preconditioned conjugate gradient method [160]. In the present work, we constructed a geometric multigrid solver with an alternating plane Gauss-Seidel smoother for this pressure Poisson equation. The multigrid method for the Poisson equation has been described in general terms in [137, 170] for finite difference discretizations of the Dirichlet problem on a unit cube. We reformulated that concept for an FVM on the Neumann problem, and compared it with an incomplete Cholesky preconditioned conjugate gradient method (MIC-CG) and an algebraic multigrid method (AMG) [137]. The geometric multigrid (GMG) turned out to be about 10 times faster than the (MIC-CG) and four times faster than the AMG on a 1000 000-node grid. The (MIC-CG) and (GMG) use only a small amount of extra memory, while the AMG uses at least 100 words per node of extra memory, which is more than the whole LES code uses even when including any of the other two pressure solvers.

Altogether, we showed that the implicit formulation of the geometric multigrid solver was, compared with the explicit velocity formulation with an MIC-CG pressure solver, about 20 times faster on a 1000 000-node grid.

6.5.3 Comments

This paper used the dynamic subgrid model, but this implicit multigrid solver has later been used in combination with both the Smagorinsky and a dynamic one-equation model, and the improvements in performance are equally good [19, 171]. If a more detailed study of the influence of the local CFL number were performed, we may be able to use even longer time steps, increasing the speed-up a bit more, at least on grids with such high aspect ratios. Note that we could also probably gain effectiveness using an unstructured formulation with adaptivity, reducing the number of nodes in smooth regions.

6.6 Paper VI

6.6.1 Purpose and background

As indicated in the previous papers, it is important to have a sensible grid distribution and to manage complex geometry efficiently and accurately. A geometry can be always be handled with structured grids, but it becomes very tedious when the geometry is complex. Overlapping grids, local grid refinements, can improve the flexibility of structured grids but, for a truly complex geometry, they will also be too expensive. We cannot use self-adaptive refinements on structured grids if we wish to use the same data structure in at least partions of the grid. If unstructured grids using tetrahedrons are instead used, general geometry and adaptivity can be handled in a natural way.

We have in our previous work on structured grids been satisfied with the FMG-SIMPLE solver and we therefore chose to extend that concept to unstructured grids. In comparison of two earlier studies at our department [101, 151], the SUPG-FEM was found to be superior to the FVM, and we therefore chose the SUPG finite element method [79, 80, 81].

6.6.2 The work and results

We used an isoparametric mapping and both $P1/P1$ and $P2/P2$ elements, which showed second and third-order convergence, respectively. We wrote the SIMPLE algorithm on operator form [109], in contrast to the standard notation in [107, 158], and reformulated the pressure correction equation in variational form. It was also necessary to make a modification of the SIMPLE method to simulate zero traction outlets, and this modification gave the SIMPLE method the same performance on multiple free outlets (zero traction outlets) as for a single Dirichlet outlet.

The adaptivity is based on equidistribution of the error estimate presented in [81], and it proved to work nicely, reducing the number of nodes significantly. This FMG applied to the unstructured SUPG-FEM is based on the same basic concepts as our FMG, presented for FVM, but uses piecewise linear or quadratic test-functions instead of piecewise constants.

Finally, we implemented the $k-\omega$ turbulence model in the same manner as the $k-\epsilon$ model in Paper I, to be able to simulate high Reynolds number flows with high accuracy using this FEM-FMG method. This choice circumvents the need for wall distances, which are used in most low-Reynolds $k-\epsilon$ models. This concept proved to have the same high accuracy and speed for unstructured grids as our FMG-QUICK-FVM solver had for structured grids, and in particular it means that in all cases presented we achieved grid-independent convergence within about $100-200$ WU.

6.6.3 Comments

We started from a 2D public domain code Femlab [172], which is a 2D code for the Poisson equation with $P1/P1$ elements and a direct solver. Although we were able to use only about 10 % of the code in our work, it gave us several hints and useful data structures. In addition, we have in all applications used the grid generator in Femlab. We greatly appreciated the use of Femlab.

If in the higher order scheme $P2/P2$ were to be implemented in a deferred correction manner with a linear scheme, another CPU reduction of a factor of two for the $P2/P2$ elements would be obtained. We will most likely implement that in the near future and thus the $P2/P2$ elements will be equally cheap as the $P1/P1$ elements.

Throughout this work, we avoid storing the full matrix and a WU is thus dominated, to 75% for the $P1/P1$ elements and 85% for the $P2/P2$ elements by stiffness matrix evaluation. Thus, if storing the matrix, we could speedup the convergence rate by a factor of four to six at the price of about 75 words of RAM per node (in 2D). In 3D, the speed-up will be a bit higher, but there is an extra memory usage of 250 words of RAM per node instead. Note that this method still uses only $30-50$% of a ILU(0)-preconditioned RGMRES method.

The work in this paper was restricted to 2D, but the extension to 3D is straightforward and will probably be done in the near future.

7 Conclusions

* The multigrid method has been shown to be robust and most often converges very fast with a grid-independent speed, i.e. convergence within $\sim 100WU$. That is often 100 times faster than other methods.

* It is necessary to damp the multigrid method for the turbulent quantities to retain robustness. Doing so, the same high performance as for laminar flows is usually achieved.

* The complex inlet boundary conditions in ventilated rooms must be approximated carefully in order to preserve the accuracy.

* The use in LES of a fully implicit discretization equipped with a multigrid solver seems to be 10-100 times faster than other methods.

* Second-order accuracy can be retained for both local mesh refinements as well as unstructured grids without any significant degeneration of the performance of the multigrid solver. With preserved multigrid performance and third-order accuracy we also implemented a third-order adaptive unstructured FEM.

* The accuracy of the second-order SUPG-FEM P1/P1 method using an unstructured grid is as accurate as the second-order QUICK-FVM using a structured grid with an equal number of degrees of freedom. The third order SUPG-FEM P2/P2 method were found superior to both second-order methods.

8 Recommendations for future research

* An investigation of the behaviour of multigrid in combustion. More precisely, the effects from the heat release are interesting as is the treatment of a discrete face on coarse meshes.

* A detailed investigation of the perturbation parameter δ in SUPG, especially for turbulence models, reaction models and compressible Euler. Special emphasis on the coupling to turbulence modeling is recommended.

* An implementation of an adaptive unstructured higher order (third or fourth order) accurate method in a deferred correction manner into a full multigrid method. The use of hybrid grids and semi-coarsening in boundary layers would admit full multigrid speed even with a point-wise smoother.

References

[1] B.E. Launder, G.J. Reece, and W. Rodi. Progress in the development of a Reynolds-stress turbulence closure. *J. of Fluid Mech.*, 68:537–566, 1975.

[2] B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Comp. Meth. in Appl. Mech. Eng.*, 3:269–289, 1974.

[3] D.C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA J.*, 26(11):1299–1310, 1988.

[4] S. Zeierman and M. Wolfshtein. Turbulent time scale for turbulent-flow calculations. *AIAA J.*, 24(10):1606–1610, 1986.

[5] J. Larsson. Numerical simulation of turbine blade heat transfer. Thesis for the degree of Licentiate of Engineering, report 96/4, Dept. of Thermo and Fluid Dynamics, Chalmers Univ. Techn., Göteborg, 1996.

[6] F.S. Lien and M.A. Leschziner. Assessment of turbulence-transport models including non-linear RNG eddy-viscosity formulation and second-moment closure for flow over a backwards-facing step. *Comp. and Fluids*, 23(8):983–1004, 1994.

[7] T.J. Craft, B.E. Launder, and K. Suga. Development and application of a cubic eddy-viscosity model of turbulence. *Int. J. Heat and Fluid flow*, 17:108–115, 1996.

[8] S. Orzag, V. Yakhot, W. Flannery, F. Boysan, D. Choudhury, J. Maruzewski, and B. Patel. Renormalisation group modeling and turbulence simulations. In R.M.C. So, C.G. Speziale, and B.E. Launder, editors, *Near-Wall Turbulent Flows*, pages 1031–1047, 1993.

[9] R.A.W.M. Henkes. *Natural-Convection Boundary Layers.* PhD thesis, Delft University, Delft, The Netherlands, 1990.

[10] W. Rodi. Turbulence models and their applications in hydraulics, a state of the art review. Technical report, Int. Ass. for Hydraul. Research , Delft, 1980.

[11] V.C. Patel, W. Rodi, and G. Scheuerer. Turbulence models for near-wall and low Reynolds number flows: A review. *AIAA J.*, (September 1985):1308–1319, 1985.

[12] H.C. Chen and V.C. Patel. Practical near-wall turbulence models for complex flows including separation. AIAA Paper 87-1300, Reno, USA, 1987.

[13] M. Lesieur and O. Métais. New trends in large-eddy simulations of turbulence. *Ann. Rev. Fluid. Mech.*, 28:45–82,, 1996.

[14] L. Smagorinsky. General circulation experiments with the primitive equations I. The basic experiments. *Month. Weath. Rev.*, 91:99–164, 1963.

[15] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A*, 3:1760–1765, 1991.

[16] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. Erratum. *Phys. Fluids A*, 3:3128, 1991.

[17] L. Fuchs. Turbulence modelling as a multi-level approach. In *Multigrid Methods IV*, pages 35–42. Birkhäuser Verlag, 1994.

[18] D.K. Lilly. On the application of the eddy viscosity concept in the inertial subrange of turbulence. NCAR Manuscript 123, 1966.

[19] L. Davidson. Large eddy simulation: A dynamic one-equation subgrid model for three-dimensional recirculating flow. *To. be subitted for journal publication*, 1997.

[20] D.C. Wilcox. *Turbulence modeling for CFD.* DCW Industries Inc., La Canada , California, 1993.

[21] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Studentlitteratur, Lund , Sweden, 1987.

[22] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations.* Studentlitteratur, Lund, Sweden, 1996.

[23] C. Johnson. Partial differential equations-nonlinear problems. Lecture notes, Dept. of Math., Chalmers Univ. Techn., Göteborg, 1990.

[24] B.P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comp. Meth. in Appl. Mech. Eng.*, 19:59–98, 1979.

[25] P. Hansbo. Aspects of conservation in finite element flow computations. *Comp. Meth. Appl. Mech Eng.*, 117:423–437, 1994.

[26] E. Pärt-Enander and B. Sjögren. Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems. *Computers and Fluids*, 23:551–574, 1993.

[27] User's Guide for FLUENT/UNS & RAMPANT Release 4.0. Centerra Resource Park , 10 Cavendish Court Lebanon, NH 0376, Fluent Inc., 1996.

[28] D.J. Mavripilis. Multigrid techniques for unstructured meshes. Computational fluid dynamics lecture notes at von-Kármán Institute, 1995.

[29] V. Venkatakrishnan. Implicit schemes and parallel computing on unstructured grid CFD. Computational fluid dynamics lecture notes at von-Kármán Institute, 1995.

[30] J. Peraire, J. Peiro, and K. Morgan. Multigrid solution of the 3-D compressible Euler equations on unstructured grids. *Int. J. for Num. Meth. in Eng.*, 36:1029–1044, 1993.

[31] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Comp. Meth. in Appl. Mech. and Eng.*, 139:3–47, 1996.

[32] J.J. Quirk. A contribution to the great Riemann solver debate. *Int. J. for Num. Meth. in Fluids*, 18:555–574, 1994.

[33] W. Rider. A review of approximate Riemann solvers with Gudonov's method in Lagrangian coordinates. *Comp. and Fluids*, 23(2):397–413, 1994.

[34] T.J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA Paper 93-0668, Reno, USA, 1993.

[35] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation. *J. of Comp. Phys.*, 114:45–58, 1994.

[36] J. Casper and H.L. Atkins. A finite-volume high-order ENO scheme for two-dimensional hyperbolic systems. *J. of Comp. Phys.*, 106:62–76, 1993.

[37] J. Casper. Finite-volume implementation of high-order essentially nonoscillatory schemes in two dimensions. *AIAA J.*, 30(12):2829–2835, 1992.

[38] A. Brandt. Multigrid techniques: 1984 guide with applications to fluid dynamics. Technical report, Von-Kármán Institute, 1984.

[39] A. Jameson. Multigrid algorithms for compressible flow calculations. In *Multigrid Methods II*, pages 167–203. Springer Verlag, 1985.

[40] H. Guillard. Mixed element volume methods in computational fluid dynamics. Computational fluid dynamics lecture notes at von-Kármán Institute, 1995.

[41] G.D. Raithby. A critical evaluation of upstream differencing applied to problems involving fluid flow. *Comp. Meth. in Appl. Mech. and Eng.*, 9:75–103, 1976.

[42] P. Tamamidis and D. Assanis. Evaluation of various high-order-accuracy schemes with and without flux limiters. *Int. J. for Num. Meth. in Fluids*, 16:931–948, 1993.

[43] T. Hayse, J.A.C. Humphrey, and R. Grief. A consistently formulated QUICK scheme for fast and stable convergence using finite-volume iterative calculation procedures. *J. of Comp. Phys.*, 98:108–118, 1992.

[44] D.W. Levy, K.G. Powell, and B. van Lerr. Use of a rotated Riemann solver for the two-dimensional Euler equations. *J. of Comp. Phys.*, 106:201–214, 1993.

[45] H. Paillere and E. van der Weide. Multidimensional upwind methods for inviscid and viscous compressible flows. Computational fluid dynamics lecture notes at von-Kármán Institute, 1995.

[46] P. Arminjon and A. Dervieux. Construction of TVD-like artificial viscosities on two-dimensional arbitrary FEM grids. *J. of Comp. Phys.*, 106:176–198, 1993.

[47] B. van Leer. Towards the ultimate conservative difference scheme. Monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics*, 14:361–370, 1974.

[48] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Gudonov's method. *J. of Comp. Phys.*, 32:101–136, 1979.

[49] B.P. Leonard. Simple high-accuracy resolution program for convective modelling of discontinuities. *Int. J. for Num. Meth. in Fluids*, 8:1291–1318, 1988.

[50] P.R.M. Lyra, K. Morgan, J. Peraire, and J. Peiro. TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *Int. J. for Num. Meth. in Fluids*, 19:827–847, 1994.

[51] M. Zijlema. On the construction of a third-order accurate monotone convection scheme with application to turbulent flows in general domains. *Int. J. for Num. Meth. in Fluids*, 22:619–641, 1996.

[52] J. Cabello, K. Morgan, and L. Löhner. A comparison of higher order schemes used in a finite volume solver for unstructured grids. AIAA Paper 94-2293, Colorado Springs, USA, 1994.

[53] P. Vankeirsbilck and H. Deconinck. Solution of the compressible Euler equations with higher order ENO-schemes on general unstructured meshes. In *Computational Fluid Dynamics 92, Vol 2*, pages 843–850. Elsevier Science Publisher, 1992.

[54] M. Aftosmis, D. Gaitonde, and S. Tavares. On the accuracy, stability and monotonicity of various reconstruction algorithms for unstructured meshes. AIAA Paper 94-0415, Reno, USA, 1994.

[55] C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.*, 21(11):1525–1532, 1983.

[56] M. Perić, R. Kessler, and G. Scheuerer. Comparison of finite-volume numerical methods with staggered and colocated grids. *Computers and Fluids*, 16(4):389–403, 1988.

[57] S. Majumdar, W. Rodi, and S.P. Vanka. On the use of non-staggered pressure-velocity arrangement for numerical solution of incompressible flows. Rep. SFB 210/t/35, Sonderforschungsbereich, Universität Karlsruhe, 1987.

[58] M.C. Melaanen. Calculation of fluid flows with staggered and nonstaggered curvilinear nonorthogonal grids-The theory. *Num. Heat. Transf., Part B.*, 21:1–19, 1992.

[59] T.F. Miller and F.W. Schmidt. Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a nonstaggered grid system. *Num. Heat. Transf., Part B.*, 14:213–233, 1988.

[60] S.K. Choi, H.Y. Nam, and M. Cho. Use of the momentum interpolation method for numerical solution of incompressible flows in complex geometries: Choosing cell face velocities. *Num. Heat Transf. Part B*, 23:21–41, 1993.

[61] L. Fuchs and H.S. Zhao. Solution of three-dimensional viscous incompressible flows by a multi-grid method. *Int. J. for Num. Meth. in Fluids*, 4:539–555, 1984.

[62] K. Srinivasan and S.G. Rubin. Segmented multigrid domain decomposition solutions for three dimensional viscous recirculating flows. AIAA Paper 93-3344, USA, 1993.

[63] T. Schönfeld and M. Rudgyard. A cell-vertex approach to local mesh refinement for the 3-D Euler equations. AIAA Paper 94-0318, Reno, USA, 1994.

[64] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. of Comp. Phys.*, 82:64–84, 1989.

[65] L.-E. Eriksson and M.M. Rai. A stability analysis of various patched grid interface conditions for hyperbolic equations. Report 144, FFA, 1985.

[66] Y. Jiang, Y.G. Lai, S.Y. Ho, and A.J. Przekwas. 3D simulations of complex flows with an implicit multi-domain approach. AIAA Paper 93-3124, USA, 1993.

[67] R.L. Davis and J.F. Dannenhoffer. 3-D adaptive grid-embedding Euler technique. AIAA Paper 93-0330, Reno, USA, 1993.

[68] L. Fuchs. A local mesh-refinement technique for incompressible flows. *Comp. and Fluids*, 14(1):69–81, 1986.

[69] R. Rydén. VolSol V2.6 Users Guide. Report 9970-1875, Volvo Aero Corporation, 1996.

[70] L. Fuchs. Solution of 3-D problem using overlapping grids and multi-grid methods. In *Multigrid Methods III*, pages 167–178. Birkhäuser Verlag, 1991.

[71] J.Y. Tu and L. Fuchs. Overlapping grids and multigrid methods for three-dimensional unsteady flow calculations in IC engines. *Int. J. for Num. Meth. in Fluids*, 15:693–714, 1992.

[72] Y. Zang and R.L. Street. A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *Int. J. for Num. Meth. in Fluids*, 20:341–361, 1995.

[73] R. Glowinski and O. Pironneau. Finite element methods for Navier-Stokes equations. *Ann. Rev. Fluid Mech.*, 24:167–204, 1982.

[74] T.J.R. Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 7:1261–1275, 1987.

[75] D. Pelletier and F. Ilinca. Adaptive remeshing for the $k-\epsilon$ model of turbulence. AIAA Paper 94-0818, Reno, USA, 1994.

[76] D. Pelletier, L. Ignat, and F. Ilinca. An adaptive finite element method for conjugate heat transfer. AIAA Paper 95-0637, Reno, USA, 1995.

[77] J.P. Chabard. Modelling of turbulent complex flows on unstructured meshes using a finite element method. In *Comp. Fluid Dyn. '94*, pages 210–219, Barcelona, 1994.

[78] S.R. Idelsohn and E. Onate. Finite volumes and finite elements: Two good friends. *Int. J. for Num. Meth. in Eng.*, 37:3323–3341, 1994.

[79] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. in Appl. Mech. and Eng.*, 32:199–259, 1982.

[80] C. Johnson. Finite element methods for flow problems. Report, Dept. of Math., Chalmers Univ. Techn., Göteborg, 1993.

[81] P. Hansbo and C. Johnson. Streamline diffusion finite element methods for fluid flow. Computational fluid dynamics lecture notes at von-Kármán Institute, 1995.

[82] T.J.R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics II: Beyond SUPG. *Comp. Meth. in Appl. Mech. and Eng.*, 54:341–355, 1986.

[83] S. Idelsohn, N.Nigro, M. Storti, and G. Buscaglia. A Petrov-Galerkin formulation for advection-reaction-diffusion problems. *Comp. Meth. in Appl. Mech. and Eng.*, 136:27–46, 1996.

[84] I. Harari and T.J.R. Hughes. Stabilized finite element methods for steady advection-diffusion with production. *Comp. Meth. in Appl. Mech. and Eng.*, 115:165–191, 1994.

[85] P. Hansbo and A. Szepessy. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech Eng.*, 84:175–192, 1990.

[86] P. Hansbo. The characteristic streamline diffusion method for the time-dependent incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech Eng.*, 99:171–186, 1992.

[87] T. de Mulder, H. Deconinck, and J. M. Buchlin. SUPG/PSPG finite element simulation of the air-droplet flow in the inlet section of a cooling tower. In *Comp. Fluid Dyn.'96*, pages 697–703, 1996.

[88] L.P. Franca and S.L. Frey. Stabilized finite element methods:II. The incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 99:209–233, 1992.

[89] M. Behr and T.E. Tezduyar. Finite element solution strategies for large-scale flow simulations. *Comp. Meth. Appl. Mech. Eng.*, 112:3–24, 1994.

[90] G.J. Le Beau, S.K. Aliabadi, S.E. Ray, and T.E. Tezduyar. SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. *Comp. Meth. in Appl. Mech. and Eng.*, 104:397–422, 1993.

[91] J.-C. Carette and H. Deconinck. Unstructured mesh adaptivity for SUPG formulations based on residual decomposition of the Euler equations. Computational fluid dynamics lecture notes at von-Kármán Institute, 1995.

[92] G. Hauke and T.J.R. Hughes. A unified approach to compressible and incompressible flows. *Comp. Meth. in Appl. Mech. and Eng.*, 113:389–395, 1994.

[93] T.J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element method for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accomodating equal-order interpolations. *Comp. Meth. in Appl. Mech. and Eng.*, 59:85–99, 1986.

[94] T. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Num. Anal.*, 28(1):133–140, 1991.

[95] J. Peraire, J. Peiro, and K. Morgan. Adaptive remeshing for three-dimensional compressible flow computations. *J. of Comp. Phys.*, 103(2):269–285, 1992.

[96] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. of comput*, 31(138):333–390, 1977.

[97] L. Fuchs. An adaptive multi-grid scheme for simulation of flows. In *Multigrid Methods II*, pages 123–135. Springer Verlag, 1985.

[98] M. C. Thompson and J.H. Ferziger. An adaptive multigrid technique for the incompressible Navier-Stokes equations. *Int. J. Comp. Phys.*, 82:94–121, 1989.

[99] S.C. Caruso, J.H. Ferziger, and J. Oliger. Adaptive grid techniques for elliptic fluid-flow problems. Report no. TF-23, Thermosciences Division Department of Mechanical Engineering, Stanford University, 1985.

[100] C. Johnson. Adaptive finite element methods for diffusion and convection problems. *Comp. Meth. in Appl. Mech. and Eng.*, 82:301–395, 1990.

[101] T. Hellström. Numerical simulation of turbulent reactive boundary layer flows on unstructured meshes. Thesis for the degree of Licentiate of Engineering, report 96/1, Dept. of Thermo and Fluid Dynamics, Chalmers Univ. Techn., Göteborg, 1996.

[102] P.W. Hemker. Defect correction and higher order schemes for the multigrid solution of the steady Euler equations. In *Multigrid Methods II*, pages 150–166. Springer Verlag, 1985.

[103] B. Koren. Multigrid and defect correction for the steady Navier-Stokes equations. *J. of Comp. Phys.*, 87:25–46, 1990.

[104] B. Koren. Low-diffusion rotated upwind schemes, multigrid and defect correction for steady, multi-dimensional Euler flows. In *Multigrid Methods III*, pages 265–276. Birkhäuser Verlag, 1991.

[105] J.A. Michelsen. Mesh-adaptive solution of the Navier-Stokes equations. In *Multigrid Methods III*, pages 301–312. Birkhäuser Verlag, 1991.

[106] A. Brandt. Multigrid solvers for non-elliptic and singular-pertubation steady-state problems. Technical report, Dept. of Appl. Math., The Weizmann Institute of Science, Rehovot, Israel, 1981.

[107] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, New York, 1980.

[108] J.P. van Doormaal and G.D. Raithby. Enhancement of the SIMPLE method for predicting incompressible fluid flows. *Num. Heat. Transf.*, 7:147–163, 1984.

[109] S. Zeng and P. Wesseling. An ILU smoother for the incompressible Navier-Stokes equations in general co-ordinates. *Int. J. for Num. Meth. in Fluids*, 20:59–74, 1995.

[110] J. B. Perot. An analysis of the fractional step method. *J. of Comp. Phys.*, 108:51–58, 1993.

[111] H.C. Elman. Multigrid and Krylov subspace methods for the discrete Stokes equations. *Int. J. for Num. Meth. in Fluids*, 22:755–770, 1996.

[112] J. Cahouet and J.-P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *Int. J. for Num. Meth. in Fluids*, 8:869–895, 1988.

[113] E.C. Elman and G.H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31(6):1645–1661, 1994.

[114] P. Carriere and D. Jeandel. A 3D finite element method for the simulation of thermoconvective flows and its performances on a vector-parallel computer. *Int. J. for Num. Meth. in Fluids*, 12:929–946, 1991.

[115] W.K. Anderson, R.D. Rausch, and D.L. Bonhaus. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. *J. of Comp. Phys.*, 128:391–408, 1996.

[116] R.F. Hanby, D.J. Silvester, and J.W. Chew. A comparison of coupled and segregated iterative solution techniques for incompressible swirling flow. *Int. J. for Num. Meth. in Fluids*, 22:353–373, 1996.

[117] P. Lötstedt. Improved convergence to the steady state of the Euler equations by enhanced wave propagation. *J. of Comp. Phys.*, 114:34–44, 1994.

[118] C. Vuik. Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. *Int. J. for Num. Meth. in Fluids*, 16:507–523, 1993.

[119] C. Vuik. Fast iterative solvers for the discretized incompressible Navier-Stokes equations. *Int. J. for Num. Meth. in Fluids*, 22:195–210, 1996.

[120] P.R. McHugh and D.A. Knoll. Fully coupled finite volume solutions of the incompressible Navier-Stokes and energy equations using an inexact Newton method. *Int. J. for Num. Meth. in Fluids*, 19:439–455, 1994.

[121] S.S. Clift and P.A. Forsyth. Linear and non-linear iterative methods for the incompressible Navier-Stokes equations. *Int. J. for Num. Meth. in Fluids*, 18:229–256, 1994.

[122] R. McHugh and D.A Knoll. Comparison of standard and matrix-free implementations of several Newton-Krylov solvers. *AIAA J.*, 32(12):2394–2400, 1994.

[123] P. Wesseling. *An introduction to multigrid methods.* John Wiley and Sons Ltd., West Sussex P019 1UD, England, 1992.

[124] M. Perić, M. Rüger, and G. Scheuerer. A finite volume multigrid method for calculating turbulent flows. In *Proc. 7th Symp. on Turb. Shear Flows*, pages 7.3.1–7.3.6, Stanford, USA, 1989.

[125] M. Hortmann and M. Perić. Finite volume multigrid prediction of laminar natural convection: Bench-mark solutions. *Int J. for Num. Meth. in Fluids*, 11:189–207, 1990.

[126] D.S. Joshi and S.P. Vanka. Multigrid calculation procedure for internal flows in complex geometries. *Num. Heat Transf., Part B*, 20:61–80, 1991.

[127] F.S Lien and M.A. Leschziner. Multigrid convergence accelleration for complex flow including turbulence. In *Multigrid Methods III*, pages 277–288. Birkhäuser Verlag, 1991.

[128] X.S. Bai and L. Fuchs. A multi-grid method for calculation of turbulence and combustion. In *Multigrid Methods IV*, pages 131–142. Birkhäuser Verlag, 1994.

[129] Y. Li and L. Fuchs S. Holmberg. An evaluation of a computer code for predicting indoor airflow and heat transfer. In *12th AIVC Conf.*, pages 123–136, Ottawa, Canada, 1991.

[130] D. Dick and J. Linden. A multigrid method for steady incompressible Navier-Stokes equations based on flux difference splitting. *Int. J. for Numer. Meth. in Fluids*, 14:1311–1323, 1992.

[131] R. Löhner and K. Morgan. An unstructured multigrid method for elliptic problems. *Int. J. for Num. Meth. in Eng.*, 24:201–115, 1987.

[132] V. Parthasarathy and Y. Kallinderis. New multigrid approach for three-dimensional unstructured, adaptive grids. *AIAA J.*, 32(5):956–963, 1994.

[133] V. Parthasarathy, Y. Kallinderis, and K. Nakajima. Hybrid adaption method and directional viscous multigrid with prismatic-tetrahedral meshes. AIAA Paper 95-0670, Reno, USA, 1995.

[134] T.J. Barth. Randomized multigrid. AIAA Paper 95-0207, Reno, USA, 1995.

[135] B. Koobus, M.-H. Lallemand, and A. Dervieux. Unstructured volume-agglomeration MG: Solution of the Poisson equation. *Int. J. for Num. Meth. in Fluids*, 18:27–42, 1994.

[136] Y. Li and L. Fuchs. An anisotropic local grid refinement method for fluid flow simulation. *Num. Heat Transf., Part B*, 30:195–215, 1996.

[137] A. Brandt, W. Joppich, J. Linden, G. Lonsdale, A. Schüller, B. Steckel, and K. Stüben. Arbeitspapiere der GMD 690. Technical report, Gesellschaft für mathematik und datenverarbeitung, 1992.

[138] V.V. Shaidurov. *Multigrid methods for finite elements.* Kluwer Academic Publishers, Dordrecht , Netherlands, 1995.

[139] J.H. Bramble. Multigrid methods. Lecture notes, Dept. of Math., Cornell Univ. Techn., 1992.

[140] G.J. Shaw and S. Sivaloganathan. On the smoothing properties of the SIMPLE pressure-correction algorithm. *Int. J. for Num. Meth. in Fliuds*, 8:441–461, 1988.

[141] S. Ta'asan. Optimal multigrid method for inviscid flows. In *Multigrid Methods IV*, pages 309–320. Birkhäuser Verlag, 1994.

[142] P. Hansbo. Finite elements in practice. Lecture notes, Dept. of Math., Chalmers Univ. Techn., Göteborg, 1994.

[143] M.P. Leclercq and B. Stoufflet. Characteristic multigrid method application to solve the Euler equations with unstructured and unnested grids. *J. of Comp. Phys.*, 104:3–47, 1993.

[144] D.J. Mavripilis. Multigrid solution of the two-dimensional Euler equation on unstructured triangular meshes. *AIAA J.*, 26(7):824–831, 1987.

[145] R. Webster. An algebraic multigrid solver for Navier-Stokes problems. *Int. J. for Num. Meth. in Fluids*, 18:761–780, 1994.

[146] A. Brandt and I. Yavneh. On multigrid solution of high-Reynolds incompresible entering flows. *J. of Comp. Phys.*, 101(1):151–164, 1993.

[147] A. Brandt and I. Yavneh. Accellerated multigrid convergence and high-Reynolds recurculating flows. *SIAM J. Sci. Comp.*, 14(3):607–626, 1993.

[148] T. Siikonen. An application of Roe's flux-difference splitting for $k - \epsilon$ turbulence model. *Int. J. for Num. Meth. in Fluids*, 21:1017–1039, 1995.

[149] H. Lou, J.D. Baum, and R. Löhner. An improved finite volume scheme for compressible flows on unstructured grids. AIAA Paper 95-0348, Reno, USA, 1995.

[150] M. Thomadakis and M. Leschziner. A pressure-correction method for the solution of incompressible viscous flows on unstructured grids. *Int. J. for Num. Meth. in Fluids*, 22:581–801, 1996.

[151] L. Davidson. A pressure correction method for unstructured meshes with arbitrary control volumes. *Int. J. Numer. Meth. Fluids*, 22:265–281, 1996.

[152] P. Hansbo and C. Johnson. Adaptive streamline diffusion methods for compressible flow using conservation variables. *Comp. Meth. in Appl. Mech. and Eng.*, 87:267–280, 1991.

[153] T.J.R. Hughes. Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models. bubbles and the origins of stabilized methods. *Comp. meth. Appl. Mech. Eng.*, 127:387–401, 1995.

[154] L.-E. Eriksson. Preconditioning of Navier-Stokes equations. Slides, Volvo Aero Corp., 1996.

[155] L.-E. Eriksson. Private communications. 1997.

[156] Y. Li. *Simulations of Flow and Heat Transfer in Ventilated Rooms.* PhD thesis, Royal Inst. of Techn., Stockholm, Sweden, 1992.

[157] P. Johansson. A three-dimensional laminar multigrid method applied to the SIMPLEC algorithm. Diploma thesis. report 92/5, Thermo and Fluid Dynamics, Chalmers University of Technology, Gothenburg, 1992.

[158] L. Davidson and B. Farhanieh. CALC-BFC: A finite-volume code employing collocated variable arrangement and Cartesian velocity components for computation of fluid flow and heat transfer in complex three-dimensional geometries. Report 92/4, Thermo and Fluid Dynamics, Chalmers University of Technology, Gothenburg, 1992.

[159] P. Emvin. PEC-FLTBB, a CFD code for turbulent buoyant flow based on a local mesh refinement full multigrid strategy using boundary fitted coordinates. Report 96/7, Dept. of Thermo and Fluid Dynamics, Chalmers University of Technology, Gothenburg, 1996.

[160] L. Davidson and P. Nielsen. Large eddy simulations of the flow in a three-dimensional ventilated room. In S. Murakami, editor, *5th Int. Conf. on Air Distributions in Rooms, ROOMVENT'96*, volume 2, pages 161–68, Yokohama, Japan, 1996.

[161] F.-S. Lien. *Computational Modelling of 3D Flow in Complex Ducts and Passages.* PhD thesis, UMIST, Dept. of Mech. Eng., Manchester, UK, 1992.

[162] K.M. Smith, W.K. Cope, and S.P. Vanka. A multigrid procedure for three-dimensional flows on non-orthogonal collocated grids. *Int. J. for Num. Meth. in Fluids*, 17:887–904, 1993.

[163] S. Vasić and K. Hanjalić. Turbulent natural convection in a square cavity. In *Turbulent Natural Convection in Enclosures*, pages 133–144, Editions Européennes Thermique et Industrie, 1993.

[164] ANNEX 20, room air and flow contaminant flow, evaluation of computational methods. Report 92/4, TNO Building and Construction Research , Delft, 1993.

[165] P.V. Nielsen. Lecture notes on mixing ventilation. Report, ISSN 0902-8005 U9513, Dept. of Building Technology and Structural Engineering,, Aalborg Universitetscenter, Aalborg, 1995.

[166] S. Ghosal and P. Moin. The basic equations for the large eddy simulation of turbulent flows in complex geometry. *J. of Comp. Phys.*, 118:24–37, 1995.

[167] S. Ghosal. An analysis of numerical errors in large eddy simulations of turbulence. *J. of Comp. Phys.*, 125:187–206, 1996.

[168] L. Davidson. Private communications. 1997.

[169] H. Choi and P. Moin. Effects of the computational time step on numerical solutions of turbulent flow. *J. of Comp. Phys.*, 113:1–4, 1994.

[170] C.A: Thole and U. Trottenberg. Basic smoothing procedures for the multigrid treatment of elliptic 3D-operators. *Notes on Num. Fluid Mech.*, 11:102–111, 1994.

[171] A. Sohankar and L. Davidson. Private communications. 1997.

[172] P. Hansbo and M. Levenstam. Femlab. Public domain software available on internet , (http.www.math.chalmers.se/Research/Femlab), Dept. of Mathematics, Chalmers University of Technology, Göteborg, 1995.