Large-Eddy Simulation of Turbulent Flows
Application to Low Mach Number and Particle-Laden Flows

Bamdad Lessani
Ph.D. Thesis

March 2003
Promoter: Prof. Chris Lacor

Proefschrift ingediend tot het behalen van de graad van doctor in de Toegepaste Wetenschappen
Acknowledgment

I would like to thank several people who helped me to complete this thesis. I would especially like to thank my advisor, Professor Chris Lacor, for his enthusiasm about this research, continuous support, endless patience and unfailing encouragement that he has given to me.

I would also like to thank my reading committee members, Professors Charles Hirsch, Gregoire Winckelmans, Pierre Comte, Martine Baelsmans, Patrick Kool, Johan Deconinck and Jean Vereecken for their helpful comments on a draft of this dissertation.

This work has benefited greatly from discussion with many graduate students, among them Sergey Smirnov, Jan Ramboer, Tim Broeckhoven and Mark Brouns. Especial thanks to Sergey Smirnov who also provided me with the incompressible carrier flow solver for my particle-laden flow simulations.

This work would not be possible without constant help of Alain Wery, our system administrator, who was always willing to kindly solve my network related problems. Many thanks to Jenny D’Haes for her administrative assistance.

Financial support for this work was partially provided by the Flemish Science Foundation (FWO) under the grants G.0170.98. and G.0130.02. This support is gratefully acknowledged.
Abstract

Large-Eddy Simulation (LES) of low Mach number compressible turbulent flows has been investigated.

An efficient algorithm has been developed and successfully applied to two test cases such as the channel flow and cavity flow. An implicit time accurate approach combined with multigrid, residual smoothing, local time stepping and preconditioning was used. In general, due to the restriction imposed on the time step by the physics of the flow on one hand, and the stiffness of the equations at low Mach number on the other hand, the advantage of a fully implicit approach over an explicit one for LES is not obvious. It was shown that the present approach, for the test cases considered, allows an efficiency gain of a factor 4 to 7 compared to the use of a purely explicit approach.

Different multigrid strategies like V-cycle, W-cycle and their sawtooth counterparts have been tested and finally, V-sawtooth cycle was demonstrated to be the most robust. A residual smoothing which also takes into account the clustering of the mesh is used as an additional method for the convergence acceleration. The physical time derivative is discretized with a multi-step method. Both the trapezoidal method and the second order backward Euler method have been tested, and the advantage of the trapezoidal method over the backward Euler has been shown. A dual time stepping method in which a pseudo-time derivative is added to the equation, is used to solve the implicit equation. At each implicit time step, the steady state solution in pseudo-time gives the values of independent variables at the new time step.

Preconditioning is applied to the pseudo-time derivative to remove the stiffness of the equations. The preconditioning parameter should also take into account the unsteady effects. As a result of unsteadiness, a global preconditioning parameter has to be used in the entire flowfield.

Large-eddy simulation of particle-laden flows is another part of this thesis. A one-way coupling method, which neglects the effect of particles on the carrier flow, is used. An incompressible finite volume solver is used for the carrier flow computations.

Particles are supposed to be spherical and the particle motion is governed only by the drag force. To calculate the fluid velocity at particle locations different interpolation methods such as the first, second and third order interpolations were tested.

Lycopodium, glass and copper particles with different diameters have been considered. The results of the channel flow simulation are in good agreement with the DNS and other LES calculations found in the literature.
## Contents

1 LES of Low Mach Number flows ........................................ 3
   1.1 Introduction ................................................. 3
   1.2 Literature Survey ........................................... 5
   1.3 Filtered Navier-Stokes Equations ............................ 7
   1.4 Subgrid-Scale Modeling ....................................... 9
       1.4.1 Dynamic Smagorinsky Model ............................ 11

2 Convergence Acceleration Techniques .................................. 15
   2.1 Explicit Runge-Kutta Schemes ................................ 15
       2.1.1 Some of the Characteristics of the Runge-Kutta Schemes 16
       2.1.2 Dual Time Stepping With Explicit Runge-Kutta .......... 18
   2.2 Analysis of Temporal Discretization ......................... 20
   2.3 Multigrid ..................................................... 26
       2.3.1 Grid Transfer Operators ............................... 29
   2.4 Residual Smoothing ........................................... 33
   2.5 Preconditioning .............................................. 35
       2.5.1 Non-Conservative Preconditioned Equations ............ 36
       2.5.2 Conservative Preconditioned Equations ................. 37
       2.5.3 Dual Time Stepping .................................... 38
   2.6 Time Correction .............................................. 46

3 Comparative Study .................................................... 49
   3.1 Influence of Preconditioning on Multigrid .................... 49
   3.2 Influence of the Number of Inner Iterations ................. 50
   3.3 Influence of the Time Correction Technique ................... 53
   3.4 Relation Between the Time Integration and the Physical Time Step ......................................................... 54
   3.5 The Time Step and Preconditioning Parameters ............... 55

4 Final Results .................................................................. 59
   4.1 Channel Flow .................................................... 60
       4.1.1 Test Case Description and Turbulence Statistics ....... 60
4.1.2 One Dimensional Energy Spectra ..................... 65
4.2 Cavity Flow ........................................ 66
4.2.1 Test Case Description and Turbulence Statistics .... 66
4.2.2 The Mean Velocity Field ............................. 68
4.2.3 The Three-Dimensional Picture ...................... 72

5 Particle Laden Flows ................................. 77
  5.1 Introduction and Literature Survey .................... 77
  5.2 Incompressible Navier-Stokes Solver .................. 80
  5.3 Finite Volume Compact Scheme ......................... 82
    5.3.1 Convective Fluxes .............................. 83
    5.3.2 Viscous Fluxes .................................. 85
    5.3.3 Application to Channel Flow .................... 86
  5.4 Calculation of the Particle Trajectories ............. 90
    5.4.1 Interpolation .................................. 92
    5.4.2 Numerical Details ............................... 95
  5.5 Particle-Laden Channel Flow .......................... 98
    5.5.1 Lycopodium Particles ........................... 99
    5.5.2 Glass Particles ................................. 106
    5.5.3 Copper Particles ............................... 106

6 Conclusion and Future Work ............................ 113
  6.1 Compressible Low Mach Number Flows .................. 113
  6.2 Particle-Laden Flows ................................ 115
Chapter 1

LES of Low Mach Number flows

1.1 Introduction

Turbulent flows have an infinite variety ranging from the flow of blood in our body to the atmospheric flows. Everyday life gives us an intuitive knowledge of turbulence in fluids; during air travel, one often hears the word turbulence generally associated with the fastening of seat-belts. The flow passing an obstacle or an airfoil creates turbulence in the boundary layer and develops a turbulent wake which will generally increase the drag exerted by the flow on the obstacle. The majority of atmospheric or oceanic currents cannot be predicted accurately and fall into the category of turbulent flows, even in the large planetary scales. Galaxies look strikingly like the eddies which are observed in turbulent flows such as the mixing layer, and are, in a way of speaking, the eddies of a turbulent universe. Numerous other examples of turbulent flows arise in aeronautics, hydraulics, nuclear and chemical engineering, oceanography, meteorology, astrophysics and internal geophysics. A clear understanding of this physical phenomena is one of the most essential and important problems of applied science.

In the past, due to the complexity of the governing equations of the flow, analytical solutions of these equations were restricted to very simple cases and, as a result an experimental approach was the only method to tackle this problem. The experimental approach is mainly based on similitude so that measurements made on one system, which is generally in the laboratory, can be used to describe the behavior of other systems.

A large part of the fluid mechanics community are still involved in experimental analysis, but by the rapid growth of computer power and the reduction of their prices, experiments are becoming more and more expen-
sive compared to the numerical approaches. Furthermore, in many situations like hypersonic flows or combustion problems with high temperature, measurements are very difficult or even impossible.

In the numerical approach the governing partial differential equations of the flow, which after some simplifications are called Navier-Stokes equations, are discretized in space and a time marching method is used to deal with the time derivative term.

Three main procedures are now in use for the computation of turbulent flows, Direct Numerical Simulation (DNS), Large-Eddy Simulation (LES), and an statistical approach based on the Reynolds Averaged Navier-Stokes (RANS) equations.

The most accurate approach is DNS and the Navier-Stokes equations are solved without averaging or approximation other than numerical discretizations whose errors can be estimated and controlled. This method does not need any modeling and all the motions contained in the flow are resolved. The results of a DNS contain very detailed information about the flow at a large number of grid points. These results can be used to produce statistical information. DNS is used as a research tool for understanding the mechanisms of turbulence production, energy transfer, dissipation, noise production, drag reduction, and many other physical aspects of turbulence. In a direct numerical simulation all of the kinetic energy dissipation, which occurs on the smallest scales, is captured. The size of the grid must not be larger than a viscously determined scale, called the Kolmogorov scale, \( \eta \). If \( L \) is the characteristic length of the physical domain, the number of points in one direction should be of the order,

\[
N \sim \frac{L}{\eta}
\]

with the kinematic viscosity \( \nu (m^2/s) \) and the dissipation per unit mass \( \epsilon (m^2/s^3) \) the Kolmogorov length scale \( \eta = (\nu^3/\epsilon)^{1/4} \). The dissipation per unit mass \( \epsilon \) can be approximated as \( \epsilon \sim u^3/L \) where \( u \) is a characteristic velocity of the flow. Finally the number of points for a resolved DNS in three dimensions can be estimated as,

\[
N \sim \left( \frac{L}{\eta} \right)^3 \sim \left( \frac{uL}{\nu} \right)^{9/4} = Re^{9/4}
\]

For the high Reynolds numbers the number of points enormously increases and obviously DNS is restricted to relatively low Reynolds numbers.

A method to overcome this problem is called Large-Eddy Simulation (LES) in which the low-frequency modes in space (large eddies) are directly simulated and the energy exchange with the high-frequency modes (small
1.2 Literature Survey

Large-eddy simulation has become one of the major tools to investigate the physics of the turbulence. However, the majority of computations found in the literature are either incompressible ([32], [33], [50], [52], [13], [59], among others) or compressible with moderate or high Mach number ([80]), [108], [26], [67], [24], among others) and fewer articles can be found for LES of low Mach number compressible flows. We note that, if the Mach number in the entire flowfield is low, compressibility can be neglected and incompressible equations can be used. Many problems, however, contain some regions with very low Mach numbers while other regions are compressible so that the compressible equations must be used. If the size of low-speed regions is small they will have little effect on the convergence, but if the size of these regions
is large they completely dominate the convergence process. Low-speed flows with strong heat addition are another type of problems that are compressible because of density variations induced by heat addition. The most common examples are combustion problems and flows with heat transfer. For all of these low-speed problems a compressible code must be used and the problem of stiffness should be carefully addressed.

The LES of low Mach number with compressible codes has not been adequately discussed in the literature. To the author’s knowledge, the only LES of very low Mach number compressible channel flow is that of Wang et al. [116]. They studied a turbulent channel flow with significant heat transfer using a compressible code. In their article the numerical procedure is briefly described and they put emphasis on the physical results. Arad et al. [3] used an implicit time marching method combined with multigrid for LES of compressible flows with a minimum Mach number of 0.15 without using preconditioning. In the large-eddy simulation of Pierce [87], applied to reacting flows, a variable-density approach has been used to deal with the problem of compressibility at low Mach number. Chidambaram et al. [16] studied the isotropic decaying turbulence at low Mach number with a compressible preconditioned solver.

For a review of the implicit methods that preserve the possibility of unsteady applications at all speeds see Mary et al. [75], Shuen et al. [100], Buelow et al. [10] and Dailey et al. [22]. Dailey et al. [22] are among the few investigators who have reported the application of multigrid to preconditioned solvers, especially unsteady solvers. They demonstrated the effectiveness of this method for unsteady lid driven laminar cavity flow and pulsatile channel flow.

In this thesis we deal with the large-eddy simulation of very low Mach number flow with a compressible code combined with preconditioning and multigrid, with a focus on the numerical difficulties and the methods to overcome them. The LES code is an extension of the compressible Reynolds Averaged Navier-Stokes (RANS) solver EURANUS, Lacor et al. [60], [61], Alavilli et al. [1], Kang et al. [51], Rizzi et al. [92] and Zhu et al. [120]. Combined with preconditioning, EURANUS uses a unified computing method for all Mach numbers, ranging from low subsonic [37] to hypersonic regimes, [1], [120].

The preconditioning technique was applied successfully to different steady and unsteady RANS flows, Hirsch et al. [40]. EURANUS contains an efficient multigrid solver, Zhu et al. [120], which has also been used for LES Lessani et al. [70], based on the Full Approximation Storage (FAS) scheme, Brandt [9].

The LES formulation and the filtering procedure are presented in chapter
1. The Smagorinsky model is used for subgrid-scale modeling. Both the classical model and the dynamic model, Germano et al. [32], are available in the code. The detailed formulation of the time discretization such as the Runge-Kutta methods, preconditioning, multigrid, time extrapolation and time correction are explained in chapter 2. Dual time-stepping is used to update the equations in time and the physical time derivative is discretized either with a backward differencing method or with a trapezoidal scheme. The advantage of trapezoidal scheme over backward differencing, the relationships between preconditioning and multigrid, and other related issues are discussed in chapter 3. In chapter 4 the method is applied to channel and cavity flows. Chapter 5 is about particle-laden turbulent flows. And finally, in chapter 6 a brief summary of the results and conclusions is given.

1.3 Filtered Navier-Stokes Equations

In LES one computes the motion of large-scale structures, while modeling the nonlinear interactions with the small-scales. The governing equations for the large eddies are obtained after filtering. The filtering operation can be written in terms of a convolution integral:

\[ \tilde{f}(x) = \int_D G(x - x') f(x') dx' \]  

Where \( f \) is a turbulent field, \( G \) is some spatial filter and \( D \) is the flow domain. We note that the minimum width of the spatial filter could be equal to the grid spacing. For compressible flow the Favre averaging is used rather than the standard one. A Favre-filtered variable is defined as, \( \tilde{\phi} = \rho \bar{\phi} / \bar{\rho} \). The filtered Navier-Stokes equations are:

\[ \frac{\partial \tilde{\rho}}{\partial t} + \frac{\partial \tilde{\rho} \tilde{u}_i}{\partial x_i} = 0 \]  

\[ \frac{\partial \tilde{\rho} \tilde{u}_i}{\partial t} + \frac{\partial \tilde{\rho} \tilde{u}_i \tilde{u}_j}{\partial x_j} = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial \tilde{\sigma}_{ij}}{\partial x_j} \]  

\[ \frac{\partial \tilde{\rho} \tilde{E}}{\partial t} + \frac{\partial (\tilde{\rho} \tilde{E} \tilde{u}_j + \tilde{p} \tilde{u}_j)}{\partial x_j} = \frac{\partial \tilde{\sigma}_{ij} \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{\eta}_j}{\partial x_j} \]  

The momentum equation can be further worked out as:

\[ \frac{\partial \tilde{\rho} \tilde{u}_i}{\partial t} + \frac{\partial \tilde{p} \tilde{u}_i}{\partial x_i} = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial \tilde{\sigma}_{ij}}{\partial x_j} + \underbrace{\frac{\partial (\tilde{\sigma}_{ij} - \tilde{\sigma}_{ij})}{\partial x_j}}_A - \underbrace{\frac{\partial \tilde{\eta}_j}{\partial x_j}}_B \]
Two unclosed terms appear, term \( A \) is due to the nonlinearity of the viscous stresses, and term \( B \) is the divergence of the SGS stresses

\[
\tau_{ij} = \rho \left( \bar{u}_i \bar{u}_j - \tilde{u}_i \tilde{u}_j \right) \tag{1.5}
\]

\( \sigma_{ij} \) and \( \tilde{\sigma}_{ij} \) are defined as:

\[
\sigma_{ij} = \mu \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right)
\]

\[
\tilde{\sigma}_{ij} = \tilde{\mu} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right)
\]

Term \( A \) is always neglected, i.e., \( \sigma_{ij} = \tilde{\sigma}_{ij} \), while term \( B \) is modeled

\[
\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2\mu_t (\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{mm} \delta_{ij}) \tag{1.6}
\]

Where \( \tilde{S}_{ij} = \frac{1}{2}(\tilde{u}_{ij} + \tilde{u}_{ji}) \) is the magnitude of large-scale stress-rate tensor, the eddy viscosity is,

\[
\mu_t = \bar{\rho} C \Delta^2 |\tilde{S}| \tag{1.7}
\]

with \( \Delta = (\Delta_x \Delta_y \Delta_z)^{1/3} \) the filter width and \( |\tilde{S}| = (2\tilde{S}_{ij} \tilde{S}_{ij})^{1/2} \). The trace of the SGS stresses, \( \tau_{ii} \), can be either modeled, Moin et al. [80], or simply neglected, Erlebacher et al. [29]. In the present case it is neglected. The filtered energy equation (1.4),

\[
\frac{\partial \bar{E}}{\partial t} + \frac{\partial (\bar{E} \bar{u}_j + \bar{p} \bar{u}_j)}{\partial x_j} = \frac{\partial \sigma_{ij} \bar{u}_i}{\partial x_j} - \frac{\partial \tilde{\sigma}_{ij} \tilde{u}_i}{\partial x_j}
\]

\[
- \frac{\partial (\bar{p} \bar{E} \bar{u}_j - \bar{p} \tilde{E} \tilde{u}_j)}{\partial x_j} \quad \text{[C]} \quad \frac{\partial (\bar{p} \bar{u}_j - \bar{p} \tilde{u}_j)}{\partial x_j} \quad \text{[D]}
\]

\[
+ \frac{\partial (\sigma_{ij} \bar{u}_i - \tilde{\sigma}_{ij} \tilde{u}_i)}{\partial x_j} \quad \frac{\partial (\bar{q}_j - \tilde{q}_j)}{\partial x_j} \quad \text{[E]}
\]

\[
\frac{\partial (\bar{q}_j - \tilde{q}_j)}{\partial x_j} \quad \text{[F]}
\]

Four unclosed terms \( C, D, E \) and \( F \) appear, here:

\[
\bar{q}_j = -k \frac{\partial T}{\partial x_j} \quad \text{and} \quad \tilde{q}_j = -k \frac{\partial \tilde{T}}{\partial x_j}
\]

Term \( F \) is neglected i.e. \( \bar{q}_j = \tilde{q}_j \). Term \( E \) is the subgrid-scale contribution to the viscous dissipation. Vreman et al. [114] showed that at moderate or high Mach number modeling term \( E \) improves the results. In the present
1.4. SUBGRID-SCALE MODELING

calculations the Mach number is always very low and term \( E \) is neglected. The subgrid-scale heat-flux \( Q_i \) is modeled by using an eddy diffusivity SGS model.

\[ Q_i = \rho c_p (\tilde{T} u_i - \tilde{T} \tilde{u}_i) = -\frac{\mu_t c_p}{P_{rt}} \frac{\partial \tilde{T}}{\partial x_i} \]  

(1.9)

\( P_{rt} = 0.5 \) or is calculated dynamically using the dynamic procedure of Ger-
mano et al. [32]. Using the equation of state, term \( D \) can be closed

\[ p u_j - \tilde{p} \tilde{u}_j = p R (\tilde{T} u_j - \tilde{T} \tilde{u}_j) = \frac{\gamma - 1}{\gamma} Q_j \]

The last remaining term is the convective term \( C \), with \( E \) the total energy
for unit mass \( E = \epsilon + u_i u_i / 2 \):

\[ \rho (\tilde{u}_j E - \tilde{u}_j \tilde{E}) = \rho (\tilde{u}_j \epsilon - \tilde{u}_j \tilde{\epsilon}) + \rho \left( \frac{u_j \tilde{u}_i u_i}{2} - \tilde{u}_j \tilde{u}_i \tilde{u}_i \right) = \]

\[ \rho (\tilde{u}_j \tilde{\epsilon} - \tilde{u}_j \tilde{\epsilon}) + \rho \left( \frac{u_j \tilde{u}_i u_i}{2} - \frac{\tilde{u}_j \tilde{u}_i \tilde{u}_i}{2} \right) + \rho \left( \frac{\tilde{u}_j \tilde{u}_i \tilde{u}_i}{2} - \frac{\tilde{u}_j \tilde{u}_i \tilde{u}_i}{2} \right) = \]

\[ \frac{Q_j}{\gamma} - \tilde{u}_j \tau_{ii} / 2 \]  

(1.10)

It can be argued that \( \tau_{ii} / 2 \) is small compared to the thermodynamic pressure and it can be neglected. Term \( G \) is neglected. For the present calculations this approximation did not cause any problem, but we do not know how accurate this estimation is in a general case. Equation (1.10) may also be written in the following way,

\[ \rho (\tilde{u}_j E - \tilde{u}_j \tilde{E}) = \frac{Q_j}{\gamma} + D_j \]

with \( D_j = \rho \left( \frac{u_j \tilde{u}_i u_i}{2} - \tilde{u}_j \tilde{u}_i \tilde{u}_i \right) \) a turbulent transport term. Neglecting \( \tau_{ii} \) and \( G \) in equation (1.10) is equivalent to neglecting the turbulent transport term \( D_j \).

And finally, the filtered energy equation (1.8),

\[ \frac{\partial \rho \tilde{E}}{\partial t} + \frac{\partial (\rho \tilde{E} \tilde{u}_j + \tilde{p} \tilde{u}_j)}{\partial x_j} = \frac{\partial \tilde{\sigma}_{ij} \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{q}_j}{\partial x_j} - \frac{\partial Q_j}{\partial x_j} \]  

(1.11)

1.4 Subgrid-Scale Modeling

In the context of large-eddy simulation \( \tau_{ij} \) defined in (1.5) is called the subgrid-scale stress. The name “stress” comes from the way in which it
is treated rather than its physical nature. It is in fact the momentum flux caused by the action of the unresolved scales.

The most commonly used subgrid-scale (SGS) model is the one proposed by Smagorinsky [101]. It is based on the notion that the principal effects of the SGS stress are increased transport and dissipation. As these phenomena are due to the viscosity in laminar flows, it seems reasonable to have an eddy viscosity model which is modeled according to equation (1.6).

A variety of SGS models have been used by different researchers, such as two-point closures, scale-similar models, and one equation models, to name a few among others. Kraichnan [58], by using a two-point closure model for isotropic turbulence, computed the energy transfer from the resolved to the unresolved scales. He then defined an eddy-viscosity in wave space, which hampers its extension to finite-difference schemes and to complex geometries. To overcome this shortcoming, based on the theory of isotropic turbulence, Météis and Lesieur [84] derived the structure function model which is very similar to the Smagorinsky model, see also Ducros et al. [25]. Based on the assumption that the most active subgrid scales are those closer to the cutoff, and that the scales with which they interact most are those right above the cutoff, Bardina et al. [5] developed the scale-similar models. Another approach is the one-equation models, in which in a similar way to RANS, a transport equation is solved for the subgrid-scale energy to obtain the velocity scale. One-equation models have been used by Schumann [98], Horiuti [43] and Carati et al. [12], among others. However, as stated by Piomelli [88], based on the results obtained using one-equation models, the expense involved in solving an additional equation does not seem to be justified by improvements in the accuracy. They are more methods for the subgrid-scale modeling in the literature. The interested reader can refer to the book of Sagaut [96] for a complete review.

In the present thesis the Smagorinsky model is used and will be explained further in detail. The Smagorinsky model can be derived in a number of ways including with a sort of mixing-length assumption in which the eddy viscosity is assumed to be proportional to the SGS characteristic length scale (also called filter width) $\Delta$, and to a characteristic turbulent velocity $v_\Delta = \Delta |\tilde{S}|$. Where $|\tilde{S}| = (2\tilde{S}_{ij}\tilde{S}_{ij})^{1/2}$ is a typical velocity gradient at $\Delta$, determined with the aid of the large scale (filtered-field) deformation tensor $\tilde{S}_{ij}$. In the context of finite volume no explicit filtering is used, and the filter is implicit in the numerical grid which does not support scales smaller than the grid size. The filter width used in the Smagorinsky model is taken as the third root of the control volume. The eddy viscosity model can finally be written as,

\[ \nu_t = C[\Delta(1 - e^{-\frac{\Delta}{\nu_t}})]^2|\tilde{S}| \]

(1.12)

The coefficient $C$ is a constant which should be combined with a near wall
1.4. SUBGRID-SCALE MODELING

damping \((1 - e^{-\frac{y^+}{
u}})\) to vanish near the solid boundaries. The near wall
damping is a function of the wall coordinate \(y^+ = \frac{u_w y_w}{\nu}\). \(u_r = (\tau_w/\rho)^{1/2}\) is
the friction velocity, \(\tau_w\) is the wall shear stress, \(\rho\) is the density of the fluid
and \(y_w\) is the distance from closest the wall.

In the present thesis the filter width \(\Delta = (\Delta x \Delta y \Delta z)^{2/3}\), with \(\Delta x\), \(\Delta y\)
and \(\Delta z\) the dimensions of computational cell. However, generally speaking,
the width of the filter \(\Delta\) need not have anything to do with the grid size \(h\)
other than the obvious condition that \(\Delta > h\).

Another way is to be calculated using a dynamic procedure which is the
subject of the next section. Clearly, the Smagorinsky eddy viscosity is a
simple algebraic model, but since the small scales tend to be more homoge-
neous and isotropic than the large ones, it is hoped that even simple models
can describe their physics accurately. Moreover, since the SGS stresses only
account for a fraction of the total stresses, modeling errors should not af-
fect the overall accuracy of the results as much as in the Reynolds Averaged
Navier-Stokes (RANS) modeling approach.

For the temperature, one introduces an eddy diffusivity \(\alpha_t\) such that,
\[
\tilde{T}u_i - \tilde{T}\tilde{u}_i = -\alpha_t \frac{\partial \tilde{T}}{\partial x_i}
\]
The eddy diffusivity is related to the eddy viscosity via the turbulent Prandtl
number,
\[
Pr_t = \frac{\nu_t}{\alpha_t}
\]  
(1.13)
turbulent Prandtl number can be set to a constant \(.5 \leq Pr_t \leq 1\) or calculated
dynamically, Moin et al. [80].

1.4.1 Dynamic Smagorinsky Model

Originally, this model is due to Germano et al. [32]. Later, Moin et al. [80]
generalized the model for the large-eddy simulation of compressible flows.
Lilly [71] proposed using a least squares technique to minimize the difference
between the closure assumption and the resolved stresses. The formulations
of this section are based on the article of Moin et al. [80] combined with the
least square approach of Lilly [71].

The key element of the dynamic model concept is the utilization of the
spectral data contained in the resolved field. This information is obtained
by introducing a test filter with a larger filter width than the resolved grid
filter, which generates a second field with scales larger than the resolved field.
Assume that for a quantity \(\phi\), the spatially test-filtered quantity is denoted
by a caret as \(\hat{\phi}\). The width of the test filter is denoted by \(\hat{\Delta}\). The test-
filtered stresses \(T_{ij}\) are defined by direct analogy to subgrid-scale stresses \(\tau_{ij}\),
equation (1.5). Before giving the formulation of $T_{ij}$, the standard averaging is re-introduced in equation (1.5),

$$\tau_{ij} = \bar{\rho}u_i u_j - \bar{\rho}u_i \bar{\rho}u_j / \bar{\rho}$$

(1.14)

The test-filtered stresses $T_{ij}$ are defined in a similar way as $\tau_{ij}$,

$$T_{ij} = \bar{\rho}u_i u_j - \bar{\rho}u_i \bar{\rho}u_j$$

(1.15)

The Leonard stresses\(^1\) $L_{ij}$ are given by,

$$L_{ij} = \bar{\rho}\bar{u}_i \bar{u}_j - \frac{1}{3} \bar{\rho}\bar{u}_i \bar{\rho}\bar{u}_j$$

(1.16)

the Leonard stresses are completely computable from the resolved variables, and are related to equations (1.15) and (1.14) by the following algebraic relation,

$$L_{ij} = T_{ij} - \tau_{ij}$$

(1.17)

By applying the test filter to $\tau_{ij}$ defined in equation (1.6), the test-filtered values $\hat{\tau}_{ij}$ may be written as,

$$\hat{\tau}_{ij} - \frac{1}{3} \hat{\tau}_{kk} \delta_{ij} = -2\hat{\bar{C}}_{\alpha_{ij}}$$

(1.18)

with $\alpha_{ij}$ given as,

$$\alpha_{ij} = \Delta^2 \hat{\rho}\hat{S}|(\hat{\bar{S}}_{ij} - \frac{1}{3} \hat{\bar{S}}_{mm} \delta_{ij})$$

In order to continue modeling, we need to make the approximation; $\hat{\bar{C}}_{\alpha_{ij}} = C\hat{\alpha}_{ij}$ which is equivalent to considering that $C$ is constant over an interval at least equal to the test filter cutoff length,

$$\hat{\tau}_{ij} - \frac{1}{3} \hat{\tau}_{kk} \delta_{ij} = -2C\hat{\alpha}_{ij}$$

(1.19)

The same model of (1.6) is also used for the test field stresses,

$$T_{ij} - \frac{1}{3} T_{kk} \delta_{ij} = -2C\beta_{ij}$$

(1.20)

\(^1\)Originally, these stresses were introduced by Leonard [64] in the context of an incompressible flow. If the subgrid scale velocity $u'_i = u_i - \bar{u}_i$ is defined, the SGS stresses can be decomposed into three parts,

$$\tau_{ij} = u_i u_j - \bar{u}_i \bar{u}_j = (\bar{u}_i + u'_i)(\bar{u}_j + u'_j) - \bar{u}_i \bar{u}_j = L_{ij} + C_{ij} + R_{ij}$$

where $L_{ij} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j$ are the Leonard stresses, $C_{ij} = \bar{u}_i u'_j + u'_j \bar{u}_i$ are the cross terms and $R_{ij} = u'_j u'_i$ are the SGS Reynolds stresses.
1.4. **SUBGRID-SCALE MODELING**

with,

\[
\beta_{ij} = \hat{\rho} \Delta^2 |\hat{S}| (\hat{S}_{ij} - \frac{1}{3} \hat{S}_{mm} \delta_{ij})
\]

Now by inserting \(T_{ij}\) from (1.20) and \(\tilde{\tau}_{ij}\) from (1.19) into (1.17) we have,

\[
L_{ij} - \frac{1}{3} L_{kk} \delta_{ij} = -2C(\beta_{ij} - \alpha_{ij})
\]

(1.21)

with \(L_{kk} = T_{kk} - \tilde{\tau}_{kk}\).

To solve for \(C\), this expression is contracted with \(M_{ij} = \beta_{ij} - \hat{\alpha}_{ij}\), after the appropriate spatial averaging,

\[
C = -\frac{1}{2} \frac{\langle (L_{ij} - \frac{1}{3} L_{kk} \delta_{ij}) M_{ij} \rangle}{\langle M_{ij}^2 \rangle}
\]

(1.22)

\(< \cdots >\) denotes space averaging.

According to the formulation of \(M_{ij}\) one can easily check that \(M_{kk} = 0\), as a result, in the numerator of (1.22) the term \(\frac{1}{3} L_{kk} \delta_{ij} M_{ij} = \frac{1}{3} L_{kk} M_{ii} = 0\), and equation (1.22) could be further simplified as,

\[
C = -\frac{1}{2} \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{ij}^2 \rangle}
\]

(1.23)

For the energy equation, the subgrid scale turbulent Prandtl number \(Pr_t\) in equation (1.9) is to be determined. If the Favre averaging in (1.9) is replaced by the standard averaging,

\[
Q_i = c_p (\overline{\rho T u_i} - \overline{\rho T} \overline{\rho u_i})
\]

(1.24)

in the same way, \(\Phi_i\) the heat flux at the test filter scale is defined as,

\[
\Phi_i = c_p (\overline{\tilde{\rho T u_i}} - \overline{\tilde{\rho T}} \overline{\tilde{\rho u_i}})
\]

(1.25)

by subtracting (1.25) from (1.24) the value of \(\Theta_i = \Phi_i - \hat{Q}_i\) is directly computable,

\[
\Theta_i = c_p (\overline{\tilde{\rho u_i T}} - \frac{1}{\rho} \overline{\tilde{\rho u_i} \tilde{\rho T}})
\]

(1.26)

By applying the test filter to \(Q_i\) defined in (1.9), the test-filtered value may be written as,

\[
\hat{Q}_i = -C \frac{c_p}{Pr^*_i} \Delta^2 \overline{\tilde{S}} \frac{\partial \overline{\tilde{T}}}{\partial x_i}
\]

(1.27)
the same model is used for the test filter scale,

$$\Phi_i = -C \frac{c_p}{Pr_t} \hat{\rho} \hat{\Delta}^2 |\hat{S}| \frac{\partial \tilde{T}}{\partial x_i}$$

(1.28)

Now by taking $\Theta_i$ from (1.26), $\Phi_i$ from (1.28), and $\hat{Q}_i$ from (1.27), inserting them into $\Theta_i = \Phi_i - \hat{Q}_i$, and contracting the equation by $N_i$, we will finally have,

$$Pr_t = -C \frac{<N_i^2>}{<K_jN_j>}$$

(1.29)

with,

$$N_i = \hat{\rho} \hat{\Delta}^2 |\hat{S}| \frac{\partial \tilde{T}}{\partial x_i} - \Delta^2 \hat{\rho}|\tilde{S}| \frac{\partial \tilde{T}}{\partial x_i}$$

(1.30)

and,

$$K_i = \hat{\rho} \tilde{u}_i \tilde{T} - \frac{1}{\hat{\rho}} \hat{\rho} \tilde{u}_i \hat{\rho} \tilde{T}$$

(1.31)

For the channel flow calculation the test filter is only applied along the two homogeneous directions. To give an example, the test filter of a variable $\phi_{i,j,k}$ is defined as, (assuming $i$ and $k$ indices correspond to the homogeneous directions):

$$\hat{\phi}_{i,j,k} = \frac{\phi_{i,j,k}}{4} + (\phi_{i+1,j,k} + \phi_{i-1,j,k} + \phi_{i,j,k-1} + \phi_{i,j,k+1})/8$$

$$+(\phi_{i+1,j,k+1} + \phi_{i+1,j,k-1} + \phi_{i-1,j,k+1} + \phi_{i-1,j,k-1})/16$$

(1.32)
Chapter 2

Convergence Acceleration Techniques

2.1 Explicit Runge-Kutta Schemes

We are concerned with the numerical solution of:

\[ \frac{dU}{dt} = F(U) \]  \hspace{1cm} (2.1)

An s-stage Explicit Runge-Kutta (ERK) scheme from time level (n) to (n+1) is accomplished as,

\[ U^{n+1} = U^n + \Delta t \sum_{j=1}^{s} b_j F(U^j) \]  \hspace{1cm} (2.2)

In which,

\[ U^i = U^n + \Delta t \sum_{j=1}^{i-1} a_{ij} F(U^j) \]

As an example a three-stage scheme can be written,

\[ \begin{align*}
    U^1 &= U^n \\
    U^2 &= U^n + \Delta t (a_{21} F^1) \\
    U^3 &= U^n + \Delta t (a_{31} F^1 + a_{32} F^2) \\
    U^{n+1} &= U^n + \Delta t (b_1 F^1 + b_2 F^2 + b_3 F^3)
\end{align*} \]  \hspace{1cm} (2.3)

For the three-stage method there are \( \frac{3(3+1)}{2} = 6 \) coefficients available, in general for a basic s-stage Runge-Kutta scheme there are \( \frac{s(s+1)}{2} \) coefficients available, i.e., the degree of freedom (DOF) is \( \frac{s(s+1)}{2} \).
2.1.1 Some of the Characteristics of the Runge-Kutta Schemes

Order Conditions

Order conditions are nonlinear equations in the coefficients that must be satisfied to make the time discretization error of the method of order $O(\Delta t^n)$. The Mathematica [91] package can be used to find these nonlinear equations, the result is expressed in terms of unknown coefficients $a_{ij}$, $b_j$ and $c_i = \sum_{j=1}^{s} a_{ij}$.

The number of order conditions for each order, up to order 10, can be written as,

$$n_q = \{1, 1, 2, 4, 9, 20, 48, 115, 286, 719\} \quad (2.4)$$

It means that for a first order RK method we need one equation to be satisfied, for second order method 1+1=2 equations, for third order 1+1+2=4 equations, for fourth order 1+1+2+4=8 equations, and so on. For a first order $s$-stage RK method the only equation is

$$\tau_1^{(1)} = \sum_{i=1}^{s} b_i - 1 = 0 \quad (2.5)$$

for a second order $s$-stage these two equations are needed

$$\begin{align*}
\tau_1^{(1)} &= \sum_{i=1}^{s} b_i - 1 = 0 \\
\tau_1^{(2)} &= \sum_{i=1}^{s} b_i c_i - \frac{1}{2} = 0
\end{align*} \quad (2.6)$$

and for a third order $s$-stage Runge-Kutta method the following four equations have to be satisfied,

$$\begin{align*}
\tau_1^{(1)} &= \sum_{i=1}^{s} b_i - 1 = 0 \\
\tau_1^{(2)} &= \sum_{i=1}^{s} b_i c_i - \frac{1}{2} = 0 \\
\tau_1^{(3)} &= \frac{1}{2} \sum_{i=1}^{s} b_i c_i^2 - \frac{1}{3!} = 0 \\
\tau_2^{(3)} &= \sum_{i=1,j=1}^{s} b_i a_{ij} c_j - \frac{1}{3!} = 0
\end{align*} \quad (2.7)$$

$\tau_k^{(q)}$ is a shorthand notation denoting equation number $k$ of the equations that impose $q^{th}$-order accuracy in time.

Error

The error in a $q^{th}$-order explicit Runge-Kutta scheme may be quantified in a general way by taking the $L_2$ and $L_\infty$ principal error norm,

$$A^{(q+1)} = ||\tau^{(q+1)}||_2 = \sqrt{\sum_{j=1}^{n_q+1} (\tau_j^{(q+1)})^2} \quad (2.8)$$
2.1. EXPLICIT RUNGE-KUTTA SCHEMES

$A_{\infty}^{(q+1)} = ||\tau^{(q+1)}||_{\infty} = MAX\{|\tau_{j}^{(q+1)}|\}$  \hspace{1cm} (2.9)

For example for a 4$^{th}$-order method, according to eq. (2.4), there are nine equations related to 5$^{th}$-order accuracy which are not satisfied and the error will be:

$A^{(5)} = ||\tau^{(5)}||_{2} = \sqrt{\sum_{j=1}^{n_{j}=9} (\tau_{j}^{(5)})^{2}}$

Linear Stability

The linear stability function for an s-stage explicit Runge-Kutta method is given by:

$P(z) = 1 + \alpha_{1}z + \alpha_{2}z^{2} + \alpha_{3}z^{3} + \cdots + \alpha_{s}z^{s}$  \hspace{1cm} (2.10)

$\alpha_{i}$ coefficients depend on the values of $b_{j}$ and $a_{ij}$. For a value of $z$ such that $|P(z)| < 1$ the method is stable.

To provide an example, suppose the three-stage Runge-Kutta scheme given in (2.3) is used to discretize the following model problem,

$\frac{dy}{dt} = \lambda y \quad y(0) = y_{0}$  \hspace{1cm} (2.11)

here $\lambda$ is a complex constant. The amplification $\frac{y^{n+1}}{y^{n}}$ is equal to,

$\frac{y^{n+1}}{y^{n}} = p(z) = 1 + \alpha_{1}z + \alpha_{2}z^{2} + \alpha_{3}z^{3}$

with $\alpha_{1} = b_{1} + b_{2} + b_{3}$, $\alpha_{2} = b_{2}a_{21} + b_{3}a_{31} + b_{3}a_{32}$, $\alpha_{3} = b_{3}a_{32}a_{21}$, and $z = \lambda \Delta t$.

For more examples about model problems, see Moin [81].

Dispersion and Dissipation Error

The dispersion and dissipation errors of the Runge-Kutta method are calculated by comparing the phase and amplitude of the complex polynomial, $P(z)$, with the exponential function, $e^{z}$, along the imaginary axis.

Suppose $z = iy$, phase, $\phi$, and amplitude, $\alpha$, errors will be

$\phi = arg(e^{iy}) - arg(P(iy)) = iy - arg(P(iy))$  \hspace{1cm} (2.12)

$\alpha = \frac{|P(iy)|}{|e^{iy}|} = |P(iy)|$  \hspace{1cm} (2.13)
Efficiency

The relative efficiency of two Runge-Kutta schemes depends on their stability-limited time steps and their number of steps,

$$\eta^{(\text{stab})} = \frac{\Delta t_1/s_1}{\Delta t_2/s_2}$$  \hspace{1cm} (2.14)

Where $s$ is the number of stages and $\Delta t$ is the time step. Scheme 1 is more efficient for $\eta^{(\text{stab})} > 1$.

2.1.2 Dual Time Stepping With Explicit Runge-Kutta

For convenience and feasibility of analytical treatment, let us consider again the model problem,

$$\frac{dy}{dt} = \lambda y \quad y(0) = y_0$$  \hspace{1cm} (2.15)

From the analytical solution to this problem we have,

$$\frac{y(t + \Delta t)}{y(t)} = e^{\lambda \Delta t}$$

which means that, in order to have a bounded solution the real part of $\lambda$ should be negative.

For a linear convection-diffusion equation $u_t + au_x = \alpha u_{xx}$, the complex constant $\lambda$ is called the Fourier footprint, equation (2.31), which depends on the type of the spatial discretization, and the values of $a$ and $\alpha$.

If an explicit Runge-Kutta is used to solve (2.15), at stage $k$ we have,

$$y^{k+1} = y^0 + \alpha_k \left( \Delta t \lambda \right) \underbrace{z_0 y^k}_{y^k}$$  \hspace{1cm} (2.16)

to be stable, $z_o = \Delta t \lambda$ should lie inside the stability region of the Runge-Kutta scheme, or according to the linear stability condition of a Runge-Kutta method,

$$|P(\Delta t \lambda)| < 1$$

with $P(z)$ given by equation (2.10).

Now suppose a backward Euler method is used to discretize the time derivative $\frac{dy}{dt}$. A pseudo-time derivative $\frac{dy}{d\tau}$ is added to the equation to solve the implicit system,

$$\frac{dy}{d\tau} + \frac{y^{n+1} - y^n}{\Delta t} = \lambda y^{n+1}$$  \hspace{1cm} (2.17)
2.1. EXPLICIT RUNGE-KUTTA SCHEMES

An explicit Runge-Kutta can be used to march to a steady-state in pseudo-time. At stage $k$ we have,

$$y^{k+1} = y^0 + \alpha_k \Delta \tau (\lambda y^n - \frac{y^{n+1}}{\Delta t}) + \alpha_k \frac{y^n}{\Delta t} \Delta \tau$$  \hspace{1cm} (2.18)

The superscript $k$ counts pseudo-time steps and the index $n$ counts implicit (physical) time steps. The values at time level $n + 1$ are still unknown. The steady-state solution of equation (2.18) in pseudo-time $\tau$ gives the value of $y^{n+1}$. By replacing the unknown values at time level $n + 1$ by the known values at pseudo-time level $k$,

$$y^{k+1} = y^0 + \alpha_k \Delta \tau (\lambda y^k - \frac{y^k}{\Delta t}) + \alpha_k \frac{y^n}{\Delta t} \Delta \tau$$  \hspace{1cm} (2.19)

after further rearranging,

$$y^{k+1} = y^0 + \alpha_k (\Delta \tau \lambda - \frac{\Delta \tau}{\Delta t}) y^k + \alpha_k \frac{y^n}{\Delta t} \Delta \tau$$  \hspace{1cm} (2.20)

by comparison of (2.16) and (2.20) one can see,

$$z_e = z_o - \frac{\Delta \tau}{\Delta t}$$

which means that, when an explicit Runge-Kutta is used to solve an implicit system in pseudo-time, the Fourier footprint is shifted to the left along the real axis in a complex plane. For small values of the physical time step $\Delta t$, which is the case in a large-eddy simulation, this shift may cause $z_e$ to go beyond the stability limit of the Runge-Kutta method, i.e., $P(z_e) > 1$, and make the solution unstable. We note that in (2.20), $\alpha_k \frac{y^n}{\Delta t} \Delta \tau$ is constant during the inner-iterations (iterations in pseudo-time) and does not have any influence on the stability.

A method to overcome this problem is to treat the $\frac{y^n}{\Delta t}$ term in equation (2.19) implicitly within the Runge-Kutta cycle,

$$y^{k+1} = y^0 + \alpha_k \Delta \tau (\lambda y^k - \frac{y^{k+1}}{\Delta t}) + \alpha_k \frac{y^n}{\Delta t} \Delta \tau$$  \hspace{1cm} (2.21)

After further rearranging,

$$y^{k+1} = y^0 + \alpha_k \left( \frac{\Delta \tau \lambda}{1 + \alpha_k \frac{\Delta \tau}{\Delta t}} \right) y^k + \alpha_k \frac{\Delta \tau}{1 + \alpha_k \frac{\Delta \tau}{\Delta t}} (y^n - y^0)$$  \hspace{1cm} (2.22)
Now, for any value of $\Delta t$ or $\Delta \tau$ we have,

$$\frac{\Delta \tau \lambda}{1 + \alpha_k \frac{\Delta \tau}{\Delta T}} < \Delta \tau \lambda \rightarrow z_i < z_o$$

Thus, there is no shift to the left and a small value of $\Delta t$ would not cause any problem.

We also note that $\frac{\alpha_k \Delta x(y^n - y^0)}{1 + \alpha_k \frac{\Delta \tau}{\Delta T}}$ is constant during the inner-iterations and does not affect the stability.

In conclusion, according to what we have learned from the model problem, we know that in the case of the Navier-Stokes equations, the same procedure should be followed in order to stay in the Runge-Kutta stability region when the physical time step $\Delta t$ is small.

Special care should be taken to facilitate the inversion of matrices. This problem will be explained in section (2.5.3).

### 2.2 Analysis of Temporal Discretization

In an implicit approach to the Navier-Stokes equations, the physical time derivative is usually discretized with a multi-step method, like the trapezoidal method or the second order backward differencing method. The aim of this section is to observe the behavior of different time discretization schemes tested on a simple one dimensional linear equation and to choose the one which fits the best for LES calculations.

The one dimensional convection equation, $u_t + cu_x = 0$ is considered. The spatial derivative is approximated using a seven point, sixth order central difference scheme:

$$u_x = 1.5 \frac{u_{i+1} - u_{i-1}}{2\Delta x} - 0.6 \frac{u_{i+2} - u_{i-2}}{4\Delta x} + 0.1 \frac{u_{i+3} - u_{i-3}}{6\Delta x} \quad (2.23)$$

Suppose $u(x,t)$ is represented by a single Fourier mode, $u(x,t) = v(t)e^{ikx}$ and insert this value into equation (2.23),

$$\frac{\dot{v}(t)}{v(t)} + Ic(e^{-1.5sin(k\Delta x) - 0.3sin(2k\Delta x) + \frac{0.2}{6}sin(3k\Delta x)} \Delta x) = 0$$

which can be written as,

$$\frac{\dot{v}(t)}{v(t)} + Ick^* = 0 \quad (2.24)$$
2.2. ANALYSIS OF TEMPORAL DISCRETIZATION

Table 2.1: Coefficients of equation (2.26)

<table>
<thead>
<tr>
<th>Method</th>
<th>$\beta_1$</th>
<th>$\beta_0$</th>
<th>$\beta_{-1}$</th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second order backward</td>
<td>1.5</td>
<td>-2</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Trapezoidal method</td>
<td>1</td>
<td>-1</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

$k^*$ is called the effective wave number. The maximum value of $(k^* \Delta x)_{\text{max}}$ for this scheme is $(k^* \Delta x)_{\text{max}} = 1.585$. The space derivative is discretized with a high-order central method to minimize the error coming from the spatial discretization. For the temporal discretization, different methods have been used, such as the fourth order (for a linear equation) explicit Runge-Kutta method, second order backward Euler, the trapezoidal scheme, and two implicit second order Runge-Kutta methods. The second order backward Euler and the trapezoidal schemes can be written in a general form,

$$\frac{\beta_1 u_i^{n+1} + \beta_0 u_i^n + \beta_{-1} u_i^{n-1}}{\Delta t} + \gamma_1 (c u_x)^{n+1} + \gamma_2 (c u_x)^n = 0$$ (2.26)

with the values of $\beta_i$ and $\gamma_i$ given in table (2.1). The fourth order explicit Runge-Kutta scheme is given by equation (2.2) with $s = 4$ and,

$$a_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}, \quad b_j = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$ (2.27)

Two implicit Runge-Kutta schemes have also been tested. Their coefficients are set according to Marx [74], and they are called DIRK2.1 and DIRK2.2. They can be written as,

$$U^1 = U^n + \Delta t \beta_{11} F(U^1)$$
$$U^{n+1} = U^n + \Delta t (\beta_{20} F(U^n) + \beta_{21} F(U^1) + \beta_{22} F(U^{n+1}))$$ (2.28)

1It is also interesting to mention that the spatial discretization error changes the speed of propagation of the $k^{th}$ Fourier mode. By multiplying $v(t)e^{ikx}$ to both sides of equation (2.24),

$$\dot{v}(t)e^{ikx} + \frac{k^* \Delta x}{k \Delta x} v(t)I(ek)e^{ikx} = 0$$

The speed of propagation of a single Fourier mode is $e^{k^* \Delta x/k \Delta x}$ instead of $c$. For this special case (i.e., convection equation discretized with a pure central scheme) this error is equivalent to the dispersion error which causes some wiggles to appear on the left (or right depending on the ratio of $k^*/k$) of the propagating wave.
Figure 2.1: Left: 2nd order backward differencing, right: trapezoidal scheme.

with \( \beta_{11} = -0.2060, \beta_{20} = 0.2698, \beta_{21} = -0.9698 \) and \( \beta_{22} = -0.3 \) for DIRK2.1 and \( \beta_{11} = -0.2651, \beta_{20} = 0.0545, \beta_{21} = -0.7545 \) and \( \beta_{22} = -0.3 \) for DIRK2.2. These two implicit Runge-Kutta scheme, as well as the backward Euler and trapezoidal schemes are second order accurate in time and A-stable, i.e., stable in the left part of the complex plane including the axis.

Figures (2.1) and (2.2) show the effect of the time discretization on the propagation of a Gaussian wave. CFL = 1 for all cases and \( c = \Delta t = \Delta x = 1 \).

The results are compared with the exact solution which is the initial solution propagating with a convective velocity equal to one. From figure (2.1) one can see that the second order backward differencing leads to a larger decrease of amplitude compared to the trapezoidal scheme. This is in accordance with the well known fact that the backward differencing is more dissipative than the trapezoidal scheme which has no dissipation.

Figure (2.2) compares two second order implicit DIRK2.1, DIRK2.2 and a fourth order explicit Runge-Kutta. Obviously, for this test case the best result is obtained with the fourth order explicit method. However, for the compressible Navier-Stokes equations due to the restrictions imposed on the time step through the CFL condition, the explicit method becomes inefficient and an implicit method combined with dual time stepping has to be used.

In general, if the spatial derivative is approximated with a central scheme on a uniform mesh of spacing \( \Delta x \),

\[
\frac{\partial u}{\partial x} \bigg|_{i}^{n} = \frac{1}{\Delta x} \sum_{l=-N}^{N} q_{l} u_{i+l}^{n}
\]

(2.29)

Assume \( u_{i}^{n} \) is represented by a single Fourier mode,

\[
u_{i}^{n} = v^{n} e^{ik\Delta x}
\]

(2.30)
2.2. ANALYSIS OF TEMPORAL DISCRETIZATION

By inserting this Fourier mode into equation (2.29),

\[ \frac{\partial u}{\partial x} \bigg|_i^n = v^n \frac{1}{\Delta x} \sum_{l=-N}^N a_l e^{ik \Delta x (i+l)} \]

the Fourier footprint of the residual can be defined as,

\[ \mathcal{R} = \frac{\sum_{l=-N}^N a_l e^{ik \Delta x (i+l)}}{e^{ik \Delta x i}} = \sum_{l=-N}^N a_l e^{ik \Delta x l} \]  

(2.31)

In general the Fourier footprint is a complex number \( \mathcal{R} = x + iy \) and is directly related to the type of the spatial discretization. For example, if a convective term is discretized with a purely central method, the Fourier footprint would be purely imaginary with no real part. In contrast, a pure diffusion term has only negative real part without any imaginary value. The Fourier footprint of an upwind method, which has some hidden diffusion term contains both imaginary and real parts.

To calculate the dissipation and dispersion errors of the time discretization, the model problem \( \frac{dy}{dt} = \lambda y \) is again considered. If the model problem is discretized with a trapezoidal method the amplification factor \( P(\lambda \Delta t) = \frac{y^{n+1}}{y^n} \) is,

\[ P(\lambda \Delta t) = 1 + \frac{\lambda \Delta t}{2} \]  

(2.32)

For the implicit Runge-Kutta method given by equation (2.28) the amplification factor is equal to,

\[ P(\lambda \Delta t) = \frac{1 + \beta_{20} \lambda \Delta t + \beta_{21} \lambda \Delta t^2}{1 - \beta_{22} \lambda \Delta t} \]  

(2.33)

For the fourth-order explicit Runge-Kutta we have,

\[ P(\lambda \Delta t) = 1 + b_4 (\lambda \Delta t) + a_{43} b_4 (\lambda \Delta t)^2 + a_{43} a_{32} b_4 (\lambda \Delta t)^3 + a_{43} a_{32} a_{21} b_4 (\lambda \Delta t)^4 \]  

(2.34)
And finally, the amplification factor of the second order backward Euler is,

\[ P(\lambda \Delta t) = \frac{-\beta_0 \pm \sqrt{\beta_0^2 - 4\beta_1(\beta_1 - \gamma_1 \lambda \Delta t)}}{2(\beta_1 - \gamma_1 \lambda \Delta t)} \]  

(2.35)

Having more than one root is the key characteristic of multi-step methods. The second root is spurious. By putting the numerical values of \( \beta \) from table (2.1) into the above equation for the first root we will have,

\[ P(\lambda \Delta t) = \frac{2 + \sqrt{1 + 2(\lambda \Delta t)}}{3 - 2(\lambda \Delta t)} \]  

(2.36)

To separate the dissipation coming from the spatial discretization from the dissipation of the time stepping method, \( \lambda \) is supposed to be a pure imaginary complex number\(^3\). Now the dispersion and dissipation errors of the time discretization scheme is calculated by comparing the phase and amplitude of the complex polynomials \( P(\lambda \Delta t) \) calculated in equations (2.32), (2.33), (2.34), and (2.35) with the analytical amplification factor \( \frac{y(t+\Delta t)}{y(t)} = e^{\lambda \Delta t} \), obtained from the analytical solution of the model problem \( y' = \lambda y \). The dissipation and phase (dispersion) error of different schemes as a function of the imaginary variable \( \lambda \Delta t \) are shown in figure (2.3).

For a linear convection equation, the maximum value of \( \lambda \Delta t \) depends on the type of the spatial discretization, and the CFL number. For example, for the central scheme given in equation (2.23), the amplification factor of the semi-analytical\(^4\) equation is equal to,

\[ \frac{v(t+\Delta t)}{v(t)} = e^{-Ick \Delta t} \]

from the above equation one can see that the maximum value of the argument of the exponential function depends on the type of the discretization and the

\(^2\)To calculate the phase and amplitude of the square of a complex number, first its real and imaginary parts should be separated with the following formula,

\[ z = \sqrt{a+ib} = \sqrt{\frac{r+a}{2} + i \sqrt{\frac{r-a}{2}}} \quad b \geq 0 \]

with \( r = \sqrt{a^2+b^2} \).

\(^3\)This is in line with the analytical solution of the convection equation \( u_t + cu_x = 0 \). If \( u(x, t) \) is represented by a single Fourier mode \( u(x, t) = v(t)e^{ikx} \), the resulting equation for \( v(t) \) will be,

\[ \dot{v}(t) = -Ickv(t) \]

in the model problem, \( v(t) \) is represented by \( y \), and \(-Ick \) by \( \lambda \).

\(^4\)It is called semi-analytical because the spatial derivative is discretized but the time derivative is calculated analytically.
2.2. ANALYSIS OF TEMPORAL DISCRETIZATION

Dissipation of different schemes for central differencing.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Dissipation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRK2.1</td>
<td>0.00</td>
</tr>
<tr>
<td>DIRK2.2</td>
<td>1.00</td>
</tr>
<tr>
<td>TRAPEZOIDAL</td>
<td>2.00</td>
</tr>
<tr>
<td>Explicit RK 2nd</td>
<td>3.00</td>
</tr>
<tr>
<td>Order Euler</td>
<td>.50</td>
</tr>
<tr>
<td></td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
</tr>
</tbody>
</table>

Dispersion of different schemes for central differencing.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Dispersion</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRK2.1</td>
<td>-2.00</td>
</tr>
<tr>
<td>DIRK2.2</td>
<td>-1.20</td>
</tr>
<tr>
<td>TRAPEZOIDAL</td>
<td>-.40</td>
</tr>
<tr>
<td>Explicit RK 2nd</td>
<td>.40</td>
</tr>
<tr>
<td>Order Euler</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
</tr>
</tbody>
</table>

Figure 2.3: Dissipation and phase error of DIRK2.1, DIRK2.2, trapezoidal scheme and four-stage fourth order Runge-Kutta scheme

CFL number,

\[ c k^* \Delta t = \frac{c \Delta t}{\Delta x} k^* \Delta x = \text{CFL} k^* \Delta x \]

for the spatial discretization of equation (2.23) the maximum value of \( k^* \Delta x \) was 1.585, and the CFL number was set equal to 1. Thus, the imaginary variable \( \lambda \Delta t \) varies between 0 and 1.585 and the maximum amount of dissipation and dispersion can be obtained from figure (2.3) which is at \( \lambda \Delta t = 1.585 \). Another useful insight can be deduced from this analysis about the relationship between the CFL number and the accuracy of the spatial differencing. We know that the more accurate schemes have higher peak values for their effective wave number \( k^* \Delta x \). As a result, with the same CFL number, a higher order method will be more dissipated and dispersed by the errors of temporal discretization than a lower order method. That is why in general for the higher order spatial discretization a smaller CFL number is used.

According to figure (2.3) the explicit four-stage, fourth-order Runge-Kutta scheme (dot-dashed line) is dispersive for big \( \lambda \Delta t \), which is normal for an explicit scheme, but it has a very small dispersion error for \( \lambda \Delta t \) up to \( \simeq 1.5 \). The trapezoidal scheme (dotted line) is not dissipative but it has a larger dispersion error than DIRK2.1 (bold line) and DIRK2.2 (dashed line). The dissipation of DIRK2.1 is very small which makes it a good candidate for LES calculations. According to the plot, the worst time discretization is the second order backward Euler (dot-dot-dashed line) with the biggest dispersion and dissipation errors. The disadvantage of the DIRK2.1 and DIRK2.2 schemes is that they are two-stage implicit methods which makes their efficiency in terms of CPU time questionable. When a two-stage implicit scheme is used, one needs to obtain two steady-state solutions in pseudo-time to go
from the physical time level \( n \) to the time level \( n + 1 \). As a result, even though DIRK2.1 and DIRK2.2 perform nicely in a linear problem, their applicability for LES, where the computational time is an important parameter, is unclear.

In the following sections more comparisons will be made between the second order implicit multistage schemes (backward Euler and trapezoidal scheme) and the explicit Runge-Kutta scheme to assess the efficiency of these implicit approaches in the context of large-eddy simulation.

### 2.3 Multigrid

Multigrid can be used to improve the rate of convergence to a steady state. For unsteady calculations, a pseudo-time derivative is added to the set of equations and multigrid is used in pseudo-time during the inner iterations. The concept of acceleration by the introduction of multiple grids was first proposed by Fedorenko [30].

The idea of using multigrid emerges from the fact that most iterative methods eliminate the high frequency errors very fast, but fail to eliminate the low frequency errors at the same rate. In addition, the notion of high and low frequency error is related to the coarseness of the grid, in that any low frequency error on a certain grid will become a high frequency error on another sufficiently coarse grid. Therefore, the idea is to solve the equations on a sequence of grids ranging from fine to coarse, so that the complete spectrum of errors can be eliminated with a rate proper to the elimination of the high frequency errors.

First theories of multigrid methods were applied only to elliptic equations, later, extensions to hyperbolic systems were developed. There is by now a fairly well-developed theory of multigrid methods for elliptic equations based on the concept that the updating scheme (i.e., Runge-Kutta or any other time marching technique) acts as a smoothing operator on each grid [9], [36]. The smoothing is also possible for hyperbolic systems. It is possible to accelerate the evolution of a hyperbolic system to a steady state by using large time steps on coarse grids so that disturbances will be more rapidly expelled through the outer boundary. A number of effective multigrid solvers ([47], [46], [2], [83], among others) have been constructed for the Euler equations of gas dynamics, which are hyperbolic. Transonic and subsonic flows have been computed with these solvers. Some multigrid methods ([104], [73], [112], [1], [41], [11], [119], among others) have also been devised for the numerical solution of the compressible Navier-Stokes equations.

It is also interesting to mention that, when multigrid is used for solving non-linear systems of equations, the coarse grid equations are solved for an
approximation of the complete finer grid solution. In case of a linear system, however, the coarse grid equations are solved for the “correction” on the finer grid. Therefore, Brandt [9] calls his algorithm for solving non-linear systems of equations the Full Approximation Storage (FAS) scheme.

On a structured mesh, a sequence of independently generated coarser meshes can be generated by eliminating alternate points in each coordinate direction. In this section we will explain the V-sawtooth multigrid. For further readings, the interested reader can refer to Hackbush [36], or McCormick [95].

In order to give a precise description of the multigrid scheme, subscript $k$ may be used to indicate the grid. The finest grid level is on grid $k - 1$, and the next coarser levels are $k$ and $k + 1$. For example, with $k = 1$ these correspond to levels 0, 1 and 2, respectively. Suppose the general equation on the finest grid level $k - 1$ is represented as,

$$\frac{\partial U_0}{\partial t} = \text{Res}_0(U_0)$$  \hspace{1cm} (2.37)

$\text{Res}_0(U_0)$ is the residual of the finest grid. To explain the method, a three-level multigrid is assumed but the extension to more grid levels is straightforward. Before going to the next coarser level $k$, the solution vector on grid $k$ must be initialized as,

$$U_k^{(0)} = T_{k-1}^{k} U_{k-1}$$  \hspace{1cm} (2.38)

where $U_{k-1}$ is the current value on grid $k - 1$ (here is the finest grid and was called $U_0$), and $T_{k-1}^{k}$ is called a transfer (also called restriction) operator which brings the information of the fine grid to the coarse grid. Next it is necessary to transfer a residual forcing function such that the solution on grid $k$ is driven by the residuals calculated on grid $k - 1$. The forcing function is defined as,

$$F_k = Q_{k-1}^{k} \text{Res}_{k-1}(U_{k-1}) - \text{Res}_k(U_k^{(0)})$$  \hspace{1cm} (2.39)

where $Q_{k-1}^{k}$ is a transfer/restriction operator for the residuals. Then $\text{Res}_k(U_k)$ is replaced by $\text{Res}_k(U_k) + F_k$ in the time stepping scheme which can be a Runge-Kutta method. A $(q+1)$-stage scheme on grid level $k$ is reformulated as,

$$U_{k}^{(1)} = U_{k}^{(0)} + \alpha_1 \Delta t_k [\text{Res}_k(U_k^{(0)}) + F_k]$$

$$\cdots = \cdots$$

$$U_{k}^{(q+1)} = U_{k}^{(0)} + \alpha_{q+1} \Delta t_k [\text{Res}_k(U_k^{(q)}) + F_k]$$

The result of $U_k = U_k^{(q+1)}$ then provides the initial data for grid $k + 1$. We note that a number of smoothing sweeps (i.e., Runge-Kutta iterations) can
be applied before going to the next coarser grid. To go to the next coarser level $k + 1$, the solution vector on grid $k + 1$ is again initialized as,

$$U_{k+1}^{(0)} = T_{k}^{k+1} U_k$$

and the forcing function $F_{k+1}$ is,

$$F_{k+1} = Q_{k}^{k+1} (Res_k(U_k) + F_k) - Res_{k+1}(U_{k+1}^{(0)})$$

a $(q+1)$-stage scheme gives,

$$U_{k+1}^{(1)} = U_{k+1}^{(0)} + \alpha_1 \Delta t_{k+1}[Res_{k+1}(U_{k+1}^{(0)}) + F_{k+1}]$$

$$\cdots = \cdots$$

$$U_{k+1}^{(q+1)} = U_{k+1}^{(0)} + \alpha_{q+1} \Delta t_{k+1}[Res_{k+1}(U_{k+1}^{(q)}) + F_{k+1}]$$

Finally, the resulting solution on the coarsest level $U_{k+1} = U_{k+1}^{(q+1)}$ has to be transferred back to grid $k$ with the aid of a prolongation (also called interpolation) operator $T_{k}^{k+1}$,

$$U_{k}^{\text{new}} = U_k + T_{k+1}^{k}(U_{k+1} - T_{k}^{k+1} U_k)$$

the value of $U_{k}^{\text{new}}$ can either be smoothed again with a Runge-Kutta method or directly prolonged to the next finer grid $k - 1$. If it is directly prolonged without any smoothing, the multigrid cycle is called sawtooth cycle. In case of a sawtooth cycle the final value on grid level $k - 1$ is,

$$U_{k-1}^{\text{new}} = U_{k-1} + T_{k}^{k-1}(U_{k}^{\text{new}} - T_{k-1}^{k} U_{k-1})$$

If on grid level $k$ instead of going to the next finer level $k - 1$, the solution and residual are again restricted to the coarser level $k + 1$, the cycle is called a W-cycle. In figure (2.4), V-sawtooth and W-sawtooth cycles are schematically shown. The empty circles denote no smoothing.
2.3. MULTIGRID

2.3.1 Grid Transfer Operators

The intergrid transfer operators employed in the present multigrid method such as the restriction operator of residuals $Q_{\text{coarse}}^{\text{fine}}$, the restriction operator of variables $T_{\text{coarse}}^{\text{fine}}$ and the prolongation operator of variables $T_{\text{fine}}^{\text{coarse}}$ are discussed in this section. First and second order prolongations, as well as linear and quadratic restrictions are discussed. The order of the prolongation operator can be defined as the highest degree plus one of polynomials that are interpolated exactly. With this definition the piecewise constant prolongation and the linear prolongation, are first and second order, respectively.

The residual-restriction operator is frequently defined as the adjoint of the correction-prolongation operator, meaning that one operator is the transpose of the other, see Swanson et al. [105] for discussion of the adjoint property. According to this definition, the linear restriction is the adjoint of the first order prolongation and the quadratic restriction is the adjoint of the second order interpolation. Hackbush [36] proved that to have a mesh-independent rate of multigrid convergence, the sum of the order of the prolongation operator $m_p$, and the order of the restriction operator $m_r$, must exceed the order of the differential equation $2m$, being considered.

$$m_p + m_r > 2m$$

For example in Navier-Stokes equations $2m = 2$, and a choice could be the combination of a linear restriction $m_r = 1$ with a second order prolongation $m_p = 2$ to satisfy the above condition.

**Linear Restriction**

The linear restriction of the fine grid residuals $Res_F$ to a coarser mesh according to figure (2.5) may be written as,

$$Res_C = Q_f^{\text{coarse}}Res_F = \frac{\sum_{i=1}^{n} \Delta_i^{\text{a}} Res_i^{\text{a}}}{\Delta_C}$$

with $Res_i^{\text{a}}$, $Res_F^{\text{a}}$ the residuals and $\Delta_i^{\text{a}}$, $\Delta_C$ the length of the cells (a,1),(a,2). $\Delta_C$ is the length of the coarse mesh. In two and three dimensions $\Delta$ represents the surface and the volume of the cell, respectively.
CHAPTER 2. CONVERGENCE ACCELERATION TECHNIQUES

Figure 2.6: Two dimensional restriction operator.

In a similar way the restriction of the fine grid variables $U_F$ is defined as,

$$U_C = T^c_f U_F = \frac{\sum_{i=1}^{n} \Delta^a_i U^a_i}{\sum_{i=1}^{n} \Delta^a_i}$$  \hspace{1cm} (2.41)

For a one dimensional problem according to figure (2.5) $n = 2$ in equations (2.40) and (2.41), for a two dimensional $n = 4$, see figure (2.6), and for a three dimensional problem $n = 8$ (figure not shown). The cells with an $a$ index are the surrounding cells of point $C$. In one dimension there are two cells ($n=2$), in two dimension four cells ($n=4$), and in three dimension eight cells ($n=8$).

Quadratic Restriction

By increasing the number of cells involved in the restriction operator, a more accurate operator can be obtained. In 1D according to the figure (2.5) four cells are involved and the quadratic restriction of residual may be written as,

$$Res_C = Q^c_f Res_F = \frac{1}{\Delta C} \left( \frac{3}{4} \sum_{i=1}^{2} \Delta^a_i Res^a_i + \frac{1}{4} \sum_{i=1}^{2} \Delta^b_i Res^b_i \right)$$  \hspace{1cm} (2.42)

The restriction of the fine grid variables $U_F$ is defined as,

$$U_C = T^c_f U_F = \frac{3}{8} \sum_{i=1}^{2} \Delta^a_i U^a_i + \frac{1}{8} \sum_{i=1}^{2} \Delta^b_i U^b_i + \frac{1}{8} \sum_{i=1}^{2} \Delta^c_i U^c_i \hspace{1cm} (2.43)$$

In two dimensions according to figure (2.6) 16 neighboring cells are involved and the restriction of the residual is,

$$Res_C = \frac{1}{\Delta C} \left( \frac{9}{16} \sum_{i=1}^{4} \Delta^a_i Res^a_i + \frac{3}{16} \sum_{i=1}^{8} \Delta^b_i Res^b_i + \frac{1}{16} \sum_{i=1}^{4} \Delta^c_i Res^c_i \right) \hspace{1cm} (2.44)$$
whereas for the restriction of the variables,

\[ U_C = T^f C = \frac{9}{64} \sum_{i=1}^{4} \Delta_i U_i^a + \frac{3}{64} \sum_{i=1}^{8} \Delta_i U_i^b + \frac{1}{64} \sum_{i=1}^{4} \Delta_i U_i^c \]  \hspace{1cm} (2.45)

In three dimensions (figure not shown) 64 cells are involved and the residual restriction operator is,

\[ \text{Res}_C = \frac{1}{\Delta_C} \left( \frac{27}{512} \sum_{i=1}^{8} \Delta_i \text{Res}_i^a + \frac{9}{512} \sum_{i=1}^{24} \Delta_i \text{Res}_i^b + \frac{3}{512} \sum_{i=1}^{24} \Delta_i \text{Res}_i^c + \frac{1}{512} \sum_{i=1}^{8} \Delta_i \text{Res}_d^d \right) \] \hspace{1cm} (2.46)

and the restricted variable on the coarse mesh is,

\[ U_C = \frac{27}{512} \sum_{i=1}^{8} \Delta_i U_i^a + \frac{9}{512} \sum_{i=1}^{24} \Delta_i U_i^b + \frac{3}{512} \sum_{i=1}^{24} \Delta_i U_i^c + \frac{1}{512} \sum_{i=1}^{8} \Delta_i U_i^d \] \hspace{1cm} (2.47)

the point \( C \) is at the cell center of a coarser cube. The cubes with index \( a \) are the surrounding fine cubes of point \( C \), cubes \( b \) have a common face with cubes \( a \), cubes \( c \) have a common edge with cubes \( a \) and cubes \( d \) have only a common corner with cubes \( a \).

**First Order Prolongation**

The simplest prolongation is the piecewise constant prolongation which is a first order operator. For a 1D problem, as illustrated in figure (2.7), the correction of a fine cell \( \delta U_f \) is simply the correction of the coarse cell \( \delta U^a \),

\[ \delta U_f = T^F_C \delta U_C = \delta U^a \] \hspace{1cm} (2.48)

Note that cell \( i - 1 \) gets the same correction as cell \( i \) with this approach. In two and three dimensions the fine grid value is the value of the coarse grid in which the cell center of the fine element is located, for example figure (2.8) shows how the piecewise constant prolongation operates in two dimensions. The cell center of the fine element \( U_f \) is located in a coarse grid with a center at \( U^a_1 \), and the value of fine element simply equals,

\[ \delta U_f = T^F_C \delta U_C = \delta U^a_1 \] \hspace{1cm} (2.49)

**Second Order Prolongation**

To improve the accuracy, a linear prolongation can be used which is second-order accurate according to the definition. In one dimension, as illustrated
CHAPTER 2. CONVERGENCE ACCELERATION TECHNIQUES

Figure 2.7: One dimensional prolongation operator.

Figure 2.8: Two dimensional prolongation operator.
In figure (2.7) the linear operator is,

$$\delta U_f = T_C^T \delta U_C = \frac{3}{4} \delta U^a + \frac{1}{4} \delta U^b$$

(2.50)

In two dimensions, as shown in figure (2.8), the cell center of the fine element \( U_f \) is located in a bigger volume with four corners \( U^1, U^1, U^2, U^c \). These four corners are the cell centers of a coarser element. If a uniform mesh is assumed the bilinear prolongation in two dimensions may be written as,

$$\delta U_f = \frac{9}{16} \delta U^a + \frac{3}{16} (\delta U^b + \delta U_2^b) + \frac{1}{16} \delta U_1^c$$

(2.51)

In three dimension the cell centers of the coarser grid make a cube in which the cell center of the fine element is located, see figure (2.9). A trilinear prolongation in a uniform mesh gives,

$$\delta U_f = \frac{27}{64} \delta U^a + \frac{9}{64} (\delta U^b + \delta U^b_1 + \delta U_2^b) + \frac{3}{64} (\delta U^c + \delta U^c_1 + \delta U^c_2 + \delta U_3^c) + \frac{1}{64} \delta U_4^d$$

(2.52)

In order to account for the mesh non-uniformity a volume-weighted average can be used. Based upon the experience from a large number of numerical simulations, Zhu [122] and Eliasson [28], a constant linear prolongation works well for both uniform and non-uniform meshes. The weighted linear prolongation does not give significant improvement in convergence, but it is rather complicated and costly.

### 2.4 Residual Smoothing

The local stability range of the basic time-stepping scheme can be extended by applying a procedure called implicit residual smoothing. This technique

Figure 2.9: Three dimensional prolongation operator.
was first introduced by Lerat [65] and later devised by Jameson [45] for Runge-Kutta schemes. In a similar way to the other convergence acceleration techniques, for unsteady calculations a pseudo-time derivative is added to the set of equations and residual smoothing is used in pseudo-time during the inner iterations.

The smoothed residual can be calculated by using an elliptic operator. If $Res$ is the original residual and $\overline{Res}$ is the smoothed residual, in one dimension the residual smoothing may be written as, \(^5\)

$$\overline{Res}_i - Res_i = \epsilon \Delta^2_i \overline{Res}_i$$

with $\Delta^2_i$ defined as,

$$\Delta^2_i \overline{Res}_i = \overline{Res}_{i+1} - 2 \overline{Res}_i + \overline{Res}_{i-1}$$

The same formula is used in three dimensions, and to avoid expensive calculation, a factored form should be used as following:

$$(1 - \epsilon_i \Delta^2_i)(1 - \epsilon_j \Delta^2_j)(1 - \epsilon_k \Delta^2_k)\overline{Res}_{i,j,k} = Res_{i,j,k} \quad (2.53)$$

In one dimension by considering a linear convection equation

$$\frac{\partial w}{\partial t} + a \frac{\partial w}{\partial x} = 0$$

discretized with a central scheme, and using a Fourier analysis, see Swanson et al. [105], one can show that a sufficient condition for stability is,

$$\epsilon \geq \frac{1}{4} \left( \frac{CFL_s}{CFL} \right)^2 - 1$$

where $CFL_s$ and $CFL$ are the CFL numbers of the smoothed and unsmoothed time marching scheme, respectively.

Then, by considering the diffusion equation

$$\frac{\partial w}{\partial t} = \nu \frac{\partial^2 w}{\partial x^2}$$

and using the same procedure employed for the convection equation, the smoothing coefficient may be determined as,

$$\epsilon \geq \frac{1}{4} \left( \frac{D_s}{D} \right) - 1$$

\(^5\)To give a physical insight, one can look at the above equation as a 1D diffusion equation, discretized in time with a first order backward Euler with $\overline{Res}_i$ the value of the function at the new time step, and $\epsilon = \alpha \Delta t/\Delta x^2$. Obviously, after one time step, the high frequency components of $Res_i$ are quickly damped due to the diffusion.
with $D_s$ and $D$ the diffusion number of the smoothed and unsmoothed scheme, $D$ is defined as $D = \frac{\nu \Delta t}{\Delta x^2}$. For a three dimensional calculation different definitions of $\epsilon$ are available in the literature. For the test cases considered in this thesis, two types of residual smoothing, which take into account the effect of the mesh aspect ratio, are used. The first one is due to Swanson et al. [104] and the smoothing factor is defined as,

$$\epsilon_i = \frac{1}{4} \left[ \left( \frac{CFL_s}{CFL} \frac{1}{1 + 0.0625(\lambda_j/\lambda_i + \lambda_k/\lambda_i)} \right)^2 - 1 \right]$$

(2.54)

$\lambda_i$, $\lambda_j$ and $\lambda_k$ are the spectral radii scaled with the cell face area, in the three directions of $i$, $j$ and $k$,

$$\lambda_i = \vec{U} \cdot \vec{S}_i + c|\vec{S}_i|$$

and similarly for $\lambda_j$ and $\lambda_k$. The three vectors $\vec{S}_i$, $\vec{S}_j$ and $\vec{S}_k$ are the vectors normal to the cell face with a length equal to its surface. On a Cartesian mesh $\vec{S}_i$, $\vec{S}_j$ and $\vec{S}_k$ have non-zero components only in the $x$, $y$ and $z$ coordinates, respectively. But on a general mesh (non-Cartesian) each of them may have three components along $x$, $y$ and $z$ coordinates. $\vec{U}$ is the velocity vector at the cell face and $c$ is the speed of sound. The other formulation used is due to Radespiel et al. [90] and the smoothing coefficient is defined as,

$$\epsilon_i = \frac{1}{4} \left[ \left( \frac{CFL_s}{CFL} \frac{1}{1 + \max(\sqrt{\lambda_j/\lambda_i}, \sqrt{\lambda_k/\lambda_i})} \right)^2 - 1 \right]$$

(2.55)

Even though theoretically an explicit Runge-Kutta scheme can be made unconditionally stable with the implicit residual smoothing, there is a practical limit on the time step when solving the hyperbolic problem. A good practical value for $\frac{CFL_s}{CFL}$ in equations (2.54) and (2.55) is $\frac{CFL_s}{CFL} = 2$.

Note that apart from the central type residual smoothing, also upwind type, Blazek et al. [7], and forward-backward implementations, Zhu et al. [121] have been formulated.

## 2.5 Preconditioning

Preconditioning removes the stiffness of the compressible Navier-Stokes equations at low Mach number and, as a result, it allows the use of a unified method that is accurate and efficient for both compressible and incompressible flows. Using preconditioning has two advantages. First, it accelerates the convergence to a steady state. Second, by modifying the artificial dissipation it improves the accuracy of the steady state solution. However, for
LES there is no artificial dissipation in the momentum and energy equations and preconditioning would not have any effect on them in this respect. In addition, in the context of dual time stepping the real steady state in pseudo-time is never reached and the subiteration procedure stops as soon as the error of the residual drop is at the same order of magnitude as the other errors of calculation.

The first contribution to the development of preconditioning methods was the artificial compressibility method introduced by Chorin [18]. Turkel [110], [111], generalized the Chorin’s formulation by allowing artificial time derivatives in all the equations and not just the continuity equation. The Choi and Merkle [17] preconditioning system is closely related to Turkel’s system. The major difference is that the Choi-Merkle system was derived using temperature as a dependant variable in the energy equation. Van Leer et al. [62], developed an alternate preconditioning formulation that was based upon a characteristic re-scaling of the equations. Their analysis provided an interesting physical and geometrical interpretation of the acoustic and particle wave processes. However, the Van Leer system at low mach numbers is closely related to the Turkel and Choi-Merkle systems, for further reading, see Darmofal et al. [23].

2.5.1 Non-Conservative Preconditioned Equations

The preconditioning method used in the present calculations is very similar to the one introduced by Choi and Merkle [17] with temperature as a dependant variable in the energy equation. Furthermore, the extension of Turkel [109] to Chorin’s [18] artificial compressibility method, with a second parameter in the momentum equations, is used to keep the generality of the formulation. This method has been applied by Hakimi [37] to different test cases, including Reynolds averaged turbulent flows with different turbulence models as well as non-Newtonian fluids. The non-conservative preconditioned Navier-Stokes equations for steady-state flows take the following form,

\[
\begin{align*}
\frac{1}{\beta^2} p_t + (\rho u_j)_j &= 0 \\
\frac{\alpha_{\text{lin}}}{\beta^2} p_t + \rho u_i u_j + \rho u_{i,j}^u u_j &= -p_i + \tau_{ij,j} \\
\rho c_p (T_{t,i} + T_i u_i) - (p_{,i} + u_i p_{,i}) &= -q_{ii,i} + \tau_{ij} u_{i,j}
\end{align*}
\] (2.56)

The time derivative of density, \( \rho_{,t} \), in the continuity equation has been replaced by \( \frac{1}{\beta^2} p_t \) and \( \frac{\alpha_{\text{lin}}}{\beta^2} p_t \) has been added to the momentum equations. The energy equation keeps its original form and is represented in terms of temperature as dependant variable.
2.5. PRECONDITIONING

2.5.2 Conservative Preconditioned Equations

Starting with equations (2.56), the continuity equation is multiplied by \( u_i \) and is added to the momentum equations,

\[
\frac{(1 + \alpha)u_i}{\beta^2} p_{,t} + u_i (\rho u_j)_{,j} + \rho u_{i,t} + \rho u_{i,j} u_j = -p_{,i} + \tau_{ij,j}
\]

(2.57)

further rearranging gives the momentum equations in conservative form,

\[
\frac{(1 + \alpha)u_i}{\beta^2} p_{,t} + \rho u_{i,t} + (\rho u_j u_i)_{,j} = -p_{,i} + \tau_{ij,j}
\]

(2.58)

For the energy equation the momentum equations are multiplied by \( u_i \) and added to the energy equations,

\[
\frac{\alpha u_i^2}{\beta^2} - 1) p_{,t} + \rho H_{,t} + \rho u_j (\frac{u_i^2}{2})_{,j} + \rho c_p T_{,t} = -q_{,j} + (\tau_{ij} u_i)_{,j}
\]

(2.59)

We know that, \( H = c_p T + \frac{u_i^2}{2} \):

\[
\frac{\alpha u_i^2}{\beta^2} - 1) p_{,t} + \rho H_{,t} + \rho u_j (\frac{u_i^2}{2})_{,j} + \rho c_p T_{,t} = -q_{,j} + (\tau_{ij} u_i)_{,j}
\]

(2.60)

and finally the energy equation in conservative form can be written as,

\[
\frac{\alpha u_i^2 + H - \beta^2}{\beta^2} p_{,t} + \rho H_{,t} + ((\rho E + p) u_j)_{,j} = -q_{,j} + (\tau_{ij} u_i)_{,j}
\]

(2.61)
CHAPTER 2. CONVERGENCE ACCELERATION TECHNIQUES

The resulting equations of the preconditioned system with conservative fluxes take the form,

\[
\begin{align*}
\frac{1}{\beta^2} p_{t,i} + (\rho u_j)_{,j} &= 0 \\
\frac{(1+\alpha)u_i}{\beta^2} p_{t,i} + \rho u_{i,t} + (\rho u_j u_i)_{,j} &= -p_{,i} + \tau_{ij,j} \\
\frac{a \alpha^2 + H - \beta^2}{\beta^2} p_{t,i} + \rho H_{,i} + ((\rho E + p) u_j)_{,j} &= -q_{i,j} + (\tau_{ij} u_i)_{,j}
\end{align*}
\] (2.66)

By applying the finite volume method to these equations:

\[
\begin{align*}
\int \frac{1}{\beta^2} p_{t,i} dV + \int \rho u_j n_j dS &= 0 \\
\int \frac{(1+\alpha)u_i}{\beta^2} p_{t,i} dV + \int (\rho u_j u_i + p \delta_{ij} - \tau_{ij}) n_j dS &= 0 \\
\int \frac{a \alpha^2 + H - \beta^2}{\beta^2} p_{t,i} dV + \int ((\rho E + p) u_j + q_j - \tau_{ij} u_i) n_j dS &= 0
\end{align*}
\] (2.67)

or in matrix form:

\[
\Gamma^{-1} \frac{\partial Q}{\partial t} = Res
\] (2.68)

with,

\[
\begin{bmatrix}
\frac{1}{\beta^2} & 0 & 0 & 0 & 0 \\
\frac{(1+\alpha)u_i}{\beta^2} & \rho & 0 & 0 & 0 \\
\frac{(1+\alpha)u_i}{\beta^2} & 0 & \rho & 0 & 0 \\
\frac{a \alpha^2 + v^2 + w^2 + H - \beta^2}{\beta^2} & 0 & 0 & \rho
\end{bmatrix}
\] (2.69)

\[
Res = -
\begin{bmatrix}
\int \rho u_j n_j dS \\
\int (\rho u_j u_1 + p \delta_{1j} - \tau_{1j}) n_j dS \\
\int (\rho u_j u_2 + p \delta_{2j} - \tau_{2j}) n_j dS \\
\int (\rho u_j u_3 + p \delta_{3j} - \tau_{3j}) n_j dS \\
\int ((\rho E + p) u_j + q_j - \tau_{ij} u_i) n_j dS
\end{bmatrix}
\] (2.70)

\(\Omega\) is the volume of the cell and \(Q = [p, u, v, w, H]\).

2.5.3 Dual Time Stepping

The formulation presented in the previous sections is not time accurate and is only useful for steady state calculations. By introducing a pseudo-time, \(\tau\), the same procedure can be followed for a time accurate calculation and for unsteady solvers. The general equations can be written as following:

\[
\begin{align*}
\frac{1}{\beta^2} p_{,\tau} + \rho_{,\tau} + (\rho u_j)_{,j} &= 0 \\
\frac{a \alpha^2}{\beta^2} p_{,\tau} + \rho u_{i,\tau} + \rho u_{i,t} + \rho u_{i,j} u_j &= -p_{,i} + \tau_{ij,j} \\
\rho c_p T_{,\tau} - p_{,\tau} + \rho c_p (T_{,t} + T_{,i} u_i) - (p_{,t} + u_i p_{,i} u_i) &= -q_{i,t} + \tau_{ij} u_{i,j}
\end{align*}
\] (2.71)
2.5. PRECONDITIONING

or with the conservative fluxes:

\[
\frac{1}{\beta^2} p, \tau + \rho, t + (\rho u_j) = 0
\]

\[
\frac{(1+\alpha)u}{\beta^2} p, \tau + \rho u_i \tau + (\rho u_j u_i) = -p, \tau + \tau_{ij,j} \] (2.72)

By applying the finite volume method to these equations:

\[
\int \frac{1}{\beta^2} p, \tau dV + \int \rho, t dV + \int \rho u_j n_j dS = 0
\]

\[
\int (\frac{(1+\alpha)}{\beta^2} p, \tau + \rho u_i \tau + (\rho u_j u_i) dV + \int (\rho u_i \tau + \rho u_j u_i) n_j dS = 0
\]

In matrix form,

\[
\Gamma^{-1} \frac{\partial Q}{\partial \tau} + \frac{\partial U}{\partial t} = \text{Res} \] (2.74)

with \( U = [\rho, \rho u, \rho v, \rho w, \rho E] \), and \( \Gamma \) and \( \text{Res} \) given in equations (2.69) and (2.70), respectively. \( \beta \) and \( \alpha \) in equation (2.69) are the preconditioning parameters. \( \alpha \) is always taken as a constant around -1, and \( \beta \) can be defined locally as, Venkateswaran et al. [113],

\[
\beta = \text{Min}(K_1 \text{Max}(u_i u_i)^{1/2}, \frac{\nu}{\delta_{\text{min}}} \frac{l}{\pi \Delta t}, c) \] (2.75)

With \( \delta_{\text{min}} \) the smallest cell length, \( l \) the biggest characteristic dimension of the domain, \( \Delta t \) the physical time step and \( c \) the speed of sound. In the analytical studies of Venkateswaran et al. [113], \( K_1 = 1 \).

The physical time derivative is discretized with a multi-step scheme:

\[
\frac{\partial U}{\partial t} - \text{Res} = \frac{\beta_1 U^{n+1} + \beta_0 U^n + \beta_{-1} U^{n-1}}{\Delta t} - \gamma_1 \text{Res}(U^{n+1}) - \gamma_2 \text{Res}(U^n) \] (2.76)

The pseudo-time derivative can also be written in terms of \( U \) the conservative variables,

\[
P^{-1} \frac{\partial U}{\partial \tau} + \frac{\partial U}{\partial t} = \text{Res} \] (2.77)

with \( P^{-1} = \Gamma^{-1} \frac{\partial Q}{\partial U} \). The steady state solution of equation (2.77) in pseudo-time \( \tau \) gives the value of \( U^{n+1} \).

Now a Runge-Kutta method can be used to reach the steady state in pseudo-time. As explained in section (2.1.2), for stability reasons the \( \frac{\partial U}{\partial t} \) term should be treated implicitly within the Runge-Kutta cycle. For example at the stage \( i \),

\[
\frac{U^i - U^0}{\Delta \tau} + \alpha_i P \frac{\beta_1 U^i + \beta_0 U^n + \beta_{-1} U^{n-1}}{\Delta t} = P \alpha_i (\gamma_1 \text{Res}(U^{i-1}) + \gamma_2 \text{Res}(U^n)) \quad i = 1, q \] (2.78)
CHAPTER 2. CONVERGENCE ACCELERATION TECHNIQUES

By adding and subtracting $\beta_i U^0$, 
\[
\frac{U^i - U^0}{\Delta \tau} + \alpha_i P\left(\frac{\beta_1 U^i - \beta_1 U^0 + \beta_0 U^n + \beta_{-1} U^{n-1}}{\Delta t}\right) = P\alpha_i (\gamma_1 \text{Res}(U^{i-1}) + \gamma_2 \text{Res}(U^n)) \tag{2.79}
\]

\[
(I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P)(U^i - U^0) = P\alpha_i \Delta \tau (-\frac{\beta_1 U^0 + \beta_0 U^n + \beta_{-1} U^{n-1}}{\Delta t} + \gamma_1 \text{Res}(U^{i-1}) + \gamma_2 \text{Res}(U^n)) \tag{2.80}
\]

\[
U^i - U^0 = (I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P)^{-1} P\alpha_i \Delta \tau (-\frac{\beta_1 U^0 + \beta_0 U^n + \beta_{-1} U^{n-1}}{\Delta t} + \gamma_1 \text{Res}(U^{i-1}) + \gamma_2 \text{Res}(U^n)) \tag{2.81}
\]

the difference on the left hand side, $U^i - U^0$, is replaced by a difference of $(Q^i - Q^0)$, multiplied by a Jacobian, $\frac{\partial U}{\partial Q}$, 
\[
\frac{\partial U}{\partial Q}(Q^i - Q^0) = (I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P)^{-1} \frac{\partial U}{\partial Q} \Gamma \alpha_i \Delta \tau \times
\]

\[\tag{2.82}
\]

This is a second order approximation, however, when the steady-state is reached in pseudo-time the approximation will be exact. To check this, suppose a general case where $U = [u_1, u_2, \ldots, u_n]$ and $Q = [q_1, q_2, \ldots, q_n]$ are two vectors with $n$ elements. The elements of $U$, denoted by $u_l$ are a function of the elements of $Q$ denoted by $q_l$, i.e., 
\[
u_l = u_l(q_1, q_2, \ldots, q_n) \quad l = 1, n
\]

The Taylor series expansion of $u_l$ at time level $i$ around time level 0 is, 
\[
u_l^{(i)} = u_l^{(0)} + \sum_{j=1}^{n} (q_j^{(i)} - q_j^{(0)}) \frac{\partial u_l}{\partial q_j}^{(0)} + O(q_j^{(i)} - q_j^{(0)})^2 \quad l = 1, n
\]

where the superscript $i$ counts the time level in pseudo-time. At the steady state, $\tau \to \infty$, the leading truncation error approaches to zero, 
\[
\lim_{\tau \to \infty} \sum_{j=1}^{n} O(q_j^{(i)} - q_j^{(0)})^2 = 0
\]

and the following equality is correct, 
\[
u_l^{(i)} - u_l^{(0)} = \sum_{j=1}^{n} (q_j^{(i)} - q_j^{(0)}) \frac{\partial u_l}{\partial q_j}^{(0)} \quad l = 1, n
\]

in a matrix form we have, 
\[
U^i - U^0 = \frac{\partial U}{\partial Q}(Q^i - Q^0)
\]
2.5. PRECONDITIONING

\[
\frac{-\beta_1 U^0 + \beta_0 U^n + \beta_{-1} U^{n-1}}{\Delta t} + \gamma_1 \text{Res}(U^{i-1}) + \gamma_2 \text{Res}(U^n)
\] (2.82)

\[
Q^i - Q^0 = \frac{\partial Q}{\partial U} \left( I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P \right) \frac{\partial U}{\partial Q} \Gamma \alpha_i \Delta \tau \times \\
\frac{-\beta_1 U^i + \beta_0 U^n + \beta_{-1} U^{i-1}}{\Delta t} + \gamma_1 \text{Res}(U^{i-1}) + \gamma_2 \text{Res}(U^n)
\] (2.83)

\[U^0\] on the right side of equation (2.82) was replaced by \[U^{i-1}\] in equation (2.83) that is a more recent value.

The matrix \[\frac{\partial Q}{\partial \Gamma} \left( I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P \right)^{-1} \frac{\partial U}{\partial Q} \] should be calculated analytically. We note that if we started from equation (2.74) instead of equation (2.77), we would have to invert the matrix \[\left( I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P \right)^{-1} \frac{\partial U}{\partial Q} \] instead of \[\frac{\partial Q}{\partial \Gamma} \left( I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P \right)^{-1} \frac{\partial U}{\partial Q}\].

For the compressible case, if from the beginning the proper set of variables is not used, this matrix may be a full 5 \times 5 matrix and computationally expensive to invert. For instance to evaluate \[\left( I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} P \right)^{-1} \frac{\partial U}{\partial Q}\], first one needs to calculate \[P = \frac{\partial U}{\partial Q} \Gamma\] with,

\[
\frac{\partial U}{\partial Q} = \begin{bmatrix}
\frac{1}{R_T} & \frac{\rho u_1}{c_p T} & \frac{\rho u_2}{c_p T} & \frac{\rho u_3}{c_p T} & -\rho \\
\frac{u_1}{R_T} & \frac{\rho u_1 u_2}{c_p T} + \rho & \frac{\rho u_1 u_3}{c_p T} & \frac{\rho u_1 H}{c_p T} & -\rho u_1 \\
\frac{u_2}{R_T} & \frac{\rho u_2 u_3}{c_p T} & \frac{\rho u_2 H}{c_p T} & \frac{\rho u_3 H}{c_p T} & -\rho u_2 \\
\frac{u_3}{R_T} & \frac{\rho u_3 u_1}{c_p T} & \frac{\rho u_3 H}{c_p T} & \frac{\rho u_1 H}{c_p T} & -\rho u_3 \\
\frac{\rho}{R_T} - 1 & \frac{\rho u_1 H}{c_p T} & \frac{\rho u_2 H}{c_p T} & \frac{\rho u_3 H}{c_p T} & \frac{\rho H}{c_p T} + \rho
\end{bmatrix}
\] (2.84)

and,

\[
\Gamma = \begin{bmatrix}
\beta^2 & 0 & 0 & 0 & \frac{-\rho}{\rho + \beta^2 - \alpha V^2 - H} \\
\frac{-\rho}{\rho + \beta^2 - \alpha V^2 - H} & 1 & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & 0 & 0 \\
\frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & 0 & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} \\
\frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{-\rho}{\rho + \beta^2 - \alpha V^2 - H} & 0 \\
\frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{\rho}{\rho + \beta^2 - \alpha V^2 - H} & \frac{-\rho}{\rho + \beta^2 - \alpha V^2 - H}
\end{bmatrix}
\] (2.85)

\[\bar{V}^2 = u_1^2 + u_2^2 + u_3^2\]. Obviously, the matrix \[P = \frac{\partial U}{\partial Q} \Gamma\], is a full 5 \times 5 matrix. It
CHAPTER 2. CONVERGENCE ACCELERATION TECHNIQUES

is the same problem with $\frac{\partial U}{\partial Q} =$

$$
\begin{bmatrix}
\frac{\beta^2}{RT} & \frac{\beta^2 \rho u_1}{c_p T} & \frac{\beta^2 \rho u_2}{c_p T} & \frac{\beta^2 \rho u_3}{c_p T} & -\frac{\beta^2 \rho}{c_p T} \\
-\frac{\alpha u_1}{\rho RT} & -\frac{\alpha u_1^2}{c_p T} + 1 & -\frac{\alpha u_1 \rho u_1}{c_p T} & -\frac{\alpha u_1 \rho u_2}{c_p T} & \frac{\alpha u_1 \rho u_3}{c_p T} \\
-\frac{\alpha u_2}{\rho RT} & -\frac{\alpha u_2 u_2}{c_p T} & -\frac{\alpha u_2 \rho u_1}{c_p T} & \frac{\alpha u_2 \rho u_2}{c_p T} & -\frac{\alpha u_2 \rho u_3}{c_p T} \\
\frac{\beta^2 - \alpha \beta^2}{\rho RT} & -\frac{1}{\rho} \left( \frac{\beta^2 - \alpha \beta^2}{c_p T} \right) u_1 & -\frac{1}{c_p T} \left( \frac{\beta^2 - \alpha \beta^2}{c_p T} \right) u_2 & -\frac{1}{c_p T} \left( \frac{\beta^2 - \alpha \beta^2}{c_p T} \right) u_3 & -\frac{\beta^2 - \alpha \beta^2}{c_p T} + 1
\end{bmatrix}
$$

(2.86)

However, for the incompressible flow, the Jacobian $\frac{\partial U}{\partial Q}$ will have a very simple form of: $\frac{\partial U}{\partial Q} = \text{diag}(0, \rho, \rho, \rho, \rho)$ and the product of $\Gamma \frac{\partial U}{\partial Q}$ will also have a simple form of: $\Gamma \frac{\partial U}{\partial Q} = \text{diag}(0, 1, 1, 1, 1)$. As a result $(I + \Gamma \alpha q \Delta \tau \frac{\partial U}{\partial Q})^{-1}$ will be a diagonal matrix equal to:

$$(I + \alpha_i \Delta \tau \frac{\beta_1}{\Delta t} \Gamma \frac{\partial U}{\partial Q})^{-1} =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & \frac{\rho}{1 + \alpha_i \frac{\beta_1}{\Delta t}} & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & \frac{\rho}{1 + \alpha_i \frac{\beta_1}{\Delta t}} & 0 \\
0 & 0 & 0 & 0 & \frac{\rho}{1 + \alpha_i \frac{\beta_1}{\Delta t}}
\end{bmatrix}
$$

(2.87)

For the test cases considered in this thesis the Mach number is very low and no difference was observed between the results of compressible and incompressible Jacobians.

For the general compressible case, one should start from a proper set of variables using a non-conservative formulation to have a preconditioner which can be easily inverted.

The non-conservative formulation in equation (2.71) can be written in a matrix form as,

$$
\Gamma_w^{-1} \frac{\partial Q}{\partial \tau} + M \frac{\partial W}{\partial t} = A \frac{\partial W}{\partial x} + B \frac{\partial W}{\partial y} + C \frac{\partial W}{\partial z}
$$

(2.88)

with,

$$
\Gamma_w^{-1} =
\begin{bmatrix}
\frac{1}{\beta^2} & 0 & 0 & 0 & 0 \\
\frac{\alpha u_1}{\beta^2} & \rho & 0 & 0 & 0 \\
\frac{\alpha u_2}{\beta^2} & 0 & \rho & 0 & 0 \\
\frac{\alpha u_3}{\beta^2} & 0 & 0 & \rho & 0 \\
-1 & 0 & 0 & 0 & \rho c_p
\end{bmatrix}
$$

(2.89)
and,

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \rho & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & \rho & 0 \\ -TR & 0 & 0 & 0 & \rho(c_p - R) \end{bmatrix} \quad (2.90)$$

Both sides of equation (2.88) are multiplied by $M^{-1}$ and the pseudo-time derivative of $Q$ is written in terms of $W$, i.e., $\frac{\partial Q}{\partial \tau} = \frac{\partial Q}{\partial W} \frac{\partial W}{\partial \tau}$.

$$M^{-1}\Gamma_w^{-1} \frac{\partial Q}{\partial W} \frac{\partial W}{\partial \tau} + M^{-1}M \frac{\partial W}{\partial t} = M^{-1}A \frac{\partial W}{\partial x} + M^{-1}B \frac{\partial W}{\partial y} + M^{-1}C \frac{\partial W}{\partial z} \quad (2.91)$$

$$P_w^{-1} \frac{\partial W}{\partial \tau} + \frac{\partial W}{\partial t} = A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z} \quad (2.92)$$

with $P_w^{-1} = M^{-1}\Gamma_w^{-1} \frac{\partial Q}{\partial W}$, $A_w = M^{-1}A$, $B_w = M^{-1}B$ and $C_w = M^{-1}C$.

The Jacobian $\frac{\partial Q}{\partial W}$ equals,

$$\frac{\partial Q}{\partial W} = \begin{bmatrix} RT & 0 & 0 & 0 & R\rho \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -TR & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.93)$$

and the matrix $P_w^{-1}$ is equal to,

$$P_w^{-1} = \begin{bmatrix} \frac{RT}{\rho} & 0 & 0 & 0 & \frac{R\rho}{\rho} \\ \frac{RT\alpha u}{\rho} & 1 & 0 & 0 & \frac{R\alpha u}{\rho} \\ \frac{RT\beta}{\rho} & 0 & 1 & 0 & \frac{R\beta}{\rho} \\ \frac{RT\gamma}{\rho} & 0 & 0 & 1 & \frac{R\gamma}{\rho} \\ (RT)^2 - RT\beta^2 & 0 & 0 & 0 & \frac{RT^2}{\rho(c_p - R)\beta^2} \end{bmatrix} + 1 \quad (2.94)$$

A Runge-Kutta scheme is used in pseudo-time $\tau$ and first order implicit Euler\(^7\) is used in physical time $t$.

$$\frac{W^i - W^0}{\Delta \tau} + \alpha_i P_w \frac{W^{n+1} - W^n}{\Delta t} = \alpha_i P_w (A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z})^{n+1} \quad (2.95)$$

The superscript $i$ counts pseudo-time steps and the index $n$ counts physical time steps. The steady state solution of equation (2.95) in pseudo-time $\tau$

---

\(^7\)The use of any other multi-step method for the discretization of the physical time derivative is straightforward, but a first order implicit Euler scheme is chosen to simplify the algebraic manipulations.
CHAPTER 2. CONVERGENCE ACCELERATION TECHNIQUES

gives the value of $W^{n+1}$. The values at time level $n+1$ are unknown and are replaced by the known values at pseudo-time level $i-1$,

$$\frac{W^i - W^0}{\Delta \tau} + \alpha_i P_w \frac{W^{i-1} - W^n}{\Delta t} = \alpha_i P_w (A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z})^{i-1}$$  (2.96)

For stability reasons explained previously, the $\alpha_i P_w \frac{W^{i-1}}{\Delta \tau}$ term on the left hand side is treated implicitly in the Runge-Kutta cycle,

$$\frac{W^i - W^0}{\Delta \tau} + \alpha_i P_w \frac{W^i - W^n}{\Delta t} = \alpha_i P_w (A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z})^{i-1}$$  (2.97)

by adding $-\alpha_i \frac{\Delta \tau}{\Delta t} P_w W^0$ to both sides, and bringing $-\alpha_i P_w W^n$ from the left side to the right,

$$W^i - W^0 + \alpha_i \frac{\Delta \tau}{\Delta t} P_w (W^i - W^0) = P_w \Delta \tau \alpha_i (\frac{W^0 - W^n}{\Delta t} + (A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z})^{i-1})$$  (2.98)

$$(W^i - W^0)(I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w) = P_w \Delta \tau \alpha_i (\frac{W^0 - W^n}{\Delta t} + (A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z})^{i-1})$$  (2.99)

$$W^i - W^0 = (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} P_w \Delta \tau \alpha_i (\frac{W^0 - W^n}{\Delta t} + (A_w \frac{\partial W}{\partial x} + B_w \frac{\partial W}{\partial y} + C_w \frac{\partial W}{\partial z})^{i-1})$$  (2.100)

Now suppose $U = [\rho, \rho u, \rho v, \rho w, \rho E]$, is the conservative variables. In order to write the equations in terms of the conservative variables $U$, the transformation $\Delta W = \frac{\partial W}{\partial U} \Delta U$ is used,

$$\frac{\partial W}{\partial U} (U^i - U^0) = (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} P_w \Delta \tau \alpha_i (\frac{\partial W U^0}{\partial U} - \frac{\partial W U^n}{\partial U}) + (A_w \frac{\partial W \partial U}{\partial U \partial x} + B_w \frac{\partial W \partial U}{\partial U \partial y} + C_w \frac{\partial W \partial U}{\partial U \partial z})^{i-1}$$  (2.101)

multiplying both sides by $\frac{\partial U}{\partial W}$,

$$U^i - U^0 = \frac{\partial U}{\partial W} (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} P_w \frac{\partial W}{\partial U} \Delta \tau \alpha_i (\frac{U^0 - U^n}{\Delta t}) + (\frac{\partial U}{\partial W} A_w \frac{\partial W \partial U}{\partial U \partial x} + \frac{\partial U}{\partial W} B_w \frac{\partial W \partial U}{\partial U \partial y} + \frac{\partial U}{\partial W} C_w \frac{\partial W \partial U}{\partial U \partial z})^{i-1}$$  (2.102)
by using the simple algebraic equality, $\frac{\partial W}{\partial U} \frac{\partial U}{\partial W} = 1$,

$$\frac{\partial U}{\partial W} (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} \frac{\partial W}{\partial U} P_w \frac{\partial U}{\partial W} \Delta \tau \alpha_i \left( \frac{U^i - U^n}{\Delta t} \right) + (A_c \frac{\partial U}{\partial x} + B_c \frac{\partial U}{\partial y} + C_c \frac{\partial U}{\partial z})^{-1} \right) (2.103)$$

$A_c$, $B_c$ and $C_c$ are respectively equal to $A_c = \frac{\partial U}{\partial W} A_w \frac{\partial W}{\partial U}$, $B_c = \frac{\partial U}{\partial W} B_w \frac{\partial W}{\partial U}$, and $C_c = \frac{\partial U}{\partial W} C_w \frac{\partial W}{\partial U}$.

$$U^i - U^0 = \frac{\partial U}{\partial W} (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} \frac{\partial W}{\partial U} P_c \Delta \tau \alpha_i \left( \frac{U^i - U^n}{\Delta t} \right) + (A_c \frac{\partial U}{\partial x} + B_c \frac{\partial U}{\partial y} + C_c \frac{\partial U}{\partial z})^{-1} \right) (2.104)$$

Where $P_c = \frac{\partial U}{\partial W} P_w \frac{\partial W}{\partial U}$ is the preconditioner in $U$ variables. Now the conservative fluxes can be retrieved,

$$U^i - U^0 = \frac{\partial U}{\partial W} (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} \frac{\partial W}{\partial U} P_c \Delta \tau \alpha_i \left( \frac{U^i - U^n}{\Delta t} \right) + \left( \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial E}{\partial z} \right)^{-1} \right) \right) \right) (2.105)$$

with $\frac{\partial F}{\partial x} = A_c \frac{\partial U}{\partial x}$, $\frac{\partial G}{\partial y} = B_c \frac{\partial U}{\partial y}$, and $\frac{\partial E}{\partial z} = C_c \frac{\partial U}{\partial z}$. $U^0$ on the right hand side of equation (2.104) is replaced by $U^{i-1}$ in equation (2.105) that is a more recent value.

To work with another set of variables $Q$ instead of $U$, one simply needs to use the transformation $U^i - U^0 = \frac{\partial U}{\partial Q} (Q^i - Q^0)$ on the left hand side of equation (2.105),

$$Q^i - Q^0 = \frac{\partial Q}{\partial U} \frac{\partial U}{\partial W} (I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1} \frac{\partial W}{\partial U} P_c \Delta \tau \alpha_i \left( \frac{U^i - U^n}{\Delta t} \right) + \left( \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial E}{\partial z} \right)^{-1} \right) \right) \right) (2.106)$$

The advantage of this formulation is that for any set of variables the matrix which should be inverted, i.e., $(I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1}$, remains the same. And if from the beginning $P_w$ is chosen properly, then $(I + \alpha_i \frac{\Delta \tau}{\Delta t} P_w)^{-1}$ can always be easily calculated.

The matrix (2.94) has a good structure and is very easy to find its inverse $P_w$. The general matrix $A$ with a structure similar to $P_w$ is,

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & a_{15} \\ a_{21} & a_{22} & 0 & 0 & a_{25} \\ a_{31} & 0 & a_{33} & 0 & a_{35} \\ a_{41} & 0 & 0 & a_{44} & a_{45} \\ a_{51} & 0 & 0 & 0 & a_{55} \end{bmatrix}$$ (2.107)
and its inverse \( A^{-1} \),

\[
A^{-1} = \begin{bmatrix}
  b_{11} & 0 & 0 & b_{12} \\
  -a_{21}b_{11} - a_{31}b_{21} & 1/a_{22} & 0 & -a_{21}b_{12} - a_{22}b_{22} \\
  -a_{31}b_{11} - a_{41}b_{21} & 0 & 1/a_{33} & -a_{31}b_{12} - a_{33}b_{22} \\
  -a_{41}b_{11} - a_{45}b_{21} & 0 & 0 & 1/a_{44} \\
  b_{21} & 0 & 0 & b_{22}
\end{bmatrix}
\]  

(2.108)

with \( b_{ij} \) defined as,

\[
\begin{bmatrix}
  b_{11} & b_{12} \\
  b_{21} & b_{22}
\end{bmatrix} = \frac{1}{a_{11}a_{55} - a_{15}a_{51}} \begin{bmatrix}
  a_{55} & -a_{15} \\
  -a_{51} & a_{11}
\end{bmatrix}
\]  

(2.109)

The structure of \((I + \alpha_i \Delta \tau P_w)\) is also similar to \( A \) and the same method can be used to calculate its inverse.

### 2.6 Time Correction

The equations in pseudo time are always solved approximately and only a tentative solution is obtained. Hirt and Harlow [42] introduced a “General Corrective Procedure” that reduces the number of subiterations and maintains the required accuracy. The method has also been tested by Marx [74] for two-dimensional incompressible unsteady calculations, and showed to be effective in reducing the number of inner iterations. Weber et al. [117] applied it to their LES calculations and concluded that the use of this procedure improves the accuracy of the results when the given degree of convergence is “almost” enough to obtain a good solution.

In this section the general corrective procedure of Hirt and Harlow is briefly explained. Consider the following equation,

\[
\frac{\partial U}{\partial t} = Res(U)
\]  

(2.110)

suppose the time derivative is discretized implicitly with a trapezoidal scheme,

\[
\frac{U^{n+1} - U^n}{\Delta t} = \frac{1}{2}(Res(U^{n+1}) + Res(U^n))
\]

to solve the implicit system a pseudo-time \( \tau \) derivative is introduced,

\[
\frac{\partial U}{\partial \tau} + \frac{U^{n+1} - U^n}{\Delta \tau} = \frac{1}{2}(Res(U^{n+1}) + Res(U^n))
\]

the steady state solution of the above equation in pseudo-time \( \tau \) gives the value of \( U^{n+1} \). When the steady state is reached, the time derivative \( \frac{\partial U}{\partial \tau} \) is
“theoretically” equal to zero. However, in practice some convergence criterion stops the subiteration procedure and leaves a small error $\epsilon$ coming from the not fully converged iterative procedure. After the last inner iteration the system that is really solved is,

$$\frac{U_{n+1} - U^n}{\Delta t} = \frac{1}{2} (Res(U_{n+1}) + Res(U^n)) + \epsilon$$  \hspace{1cm} (2.111)$$

with $U_{n+1}$ an approximate solution at time level $n + 1$ and $\epsilon$ the error related to the drop of the residual during the inner iterations. This error can be easily suppressed by using a corrective procedure which put $\epsilon$ equal to zero in equation (2.111) and solve for an intermediate solution $\hat{U}_{n+1}$,

$$\hat{U}_{n+1} = \frac{1}{2} \Delta t (Res(U_{n+1}) + Res(U^n)) + U^n$$

the implicit system at the next time level is,

$$\frac{\partial U}{\partial \tau} + \frac{U^{n+2} - \hat{U}_{n+1}}{\Delta t} = \frac{1}{2} (Res(U^{n+2}) + Res(U^n))$$

which gives the value of $U^{n+2}$ after the computational steady state is reached,

$$\frac{U^{n+2} - \hat{U}_{n+1}}{\Delta t} = \frac{1}{2} (Res(U^{n+2}) + Res(U^n)) + \epsilon$$  \hspace{1cm} (2.112)$$

and again in the above equation $\epsilon$ is set to zero and $\hat{U}_{n+2}$ is calculated,

$$\hat{U}_{n+2} = \frac{1}{2} \Delta t (Res(U^{n+2}) + Res(U^{n+1})) + \hat{U}_{n+1}$$

This correction procedure reduces the error accumulation of time integration and might allow the use of a coarser convergence criteria. Nevertheless, it should be noted that this procedure does not completely eliminate the errors. In equation (2.112) the residuals are still calculated from $U$ and not from $\hat{U}$. It is also possible to calculate the residuals in function of $\hat{U}$. This variant has also been tested by Marx [74], but he concluded that the best results were obtained when using the procedure of Hirt and Harlow [42].
Chapter 3
Comparative Study

The efficiency of an implicit dual time-stepping method in comparison with a purely explicit approach depends on the number of inner iterations $N_i$ and the ratio of the physical time step to the time step of a purely explicit scheme $R = \frac{\Delta t}{\Delta \tau}$. The gain achieved will be, gain$= \frac{R}{N_i}$. As a result, the main goal is to choose the physical time step $\Delta t$ as large as possible and make the number of inner iterations $N_i$ as small as possible.

To be able to decrease the number of inner iterations, one has to use different convergence acceleration techniques and set the corresponding parameters properly. For example, to have a fast convergence in pseudo-time the optimum values of $\alpha$ and $\beta$ parameters in preconditioning should be used. Different types of multigrid cycle should be tested. The maximum value of the physical time step which also maintain the required temporal accuracy should be set. The necessary number of inner-iterations that is directly proportinal to the residual drop should be found. The relationship between multigrid and preconditioning should be studied and so on.

In the coming sections some aspects of this problem will be studied. The only aim is to create some feeling about the different parameters which are subsequently used for the final calculations on finer meshes.

3.1 Influence of Preconditioning on Multigrid

The success of multigrid depends on the ability of the smoother to remove the high frequency errors which cannot be supported on coarser meshes. In fact efficient smoothing can be obtained if one can insure that all error components containing high frequency modes are efficiently damped on finest grid since multigrid will smooth the lower frequencies.

Darmofal et al. [23] analysed the smoothing properties of a four stage Runge-Kutta method for preconditioned Euler and Navier-Stokes equations.
via the Fourier footprint of the discrete spatial operator. They showed preconditioning effectively removes the modes along the real axis\(^1\) and brings them to the regions with higher damping properties. As a result, for low Mach number flows preconditioning is necessary to remove the stiffness of the equations and to increase the high frequency damping ability of the smoother.

The use of Fourier footprint to analyse the smoothing abilities of preconditioned Euler and Navier-Stokes equations can also be found in other references, [63], [72], [37], and [34].

In order to show this dependence a test was carried out on a channel at \(Re_\tau = 180\) and \(M=0.06\), with a mesh of \(33 \times 33 \times 33\) points. The streamwise, normal and spanwise dimensions are \(4\pi\delta \times 2\delta \times 2\pi\delta\), with \(\delta\) the channel half-width. Figure (3.1) shows part of the convergence graphs of the continuity and first momentum equations for four different combinations of multigrid and preconditioning. The number of inner iterations is fixed to 50.

According to the figure, there is almost no difference between single grid and multigrid without preconditioning (dotted and dot-dashed lines). When preconditioning is activated multigrid becomes quite efficient (solid and dashed lines). When both preconditioning and multigrid are switched off, the residual drop of the first momentum equation is around one order of magnitude, and for the continuity equation less than two. However, when preconditioning and multigrid are combined together, the residual drops are more than four and three orders of magnitude for the first momentum and continuity equations, respectively.

Due to the addition of artificial dissipation, the multigrid works more efficiently for the continuity equation than for the first momentum equation in which the artificial dissipation is switched off\(^2\). For these calculations \(\alpha = -1\) and \(\beta\) is globally defined as \(\beta^2 = 3V_{\text{cent}}^2\), where \(V_{\text{cent}}\) is the velocity at the centerline of the channel. A three-level V-sawtooth multigrid is used.

### 3.2 Influence of the Number of Inner Iterations

The goal is to find the minimum number of inner iterations or the minimum drop of residual in pseudo time during inner iterations. The error related to the drop of the residual in inner the iterations should neither be bigger

---

\(^1\)According to Darmofal, these modes with a poor propagation property, are a direct result of the low Mach number stiffness and correspond to the low speed convection modes present in the system.

\(^2\)Adding artificial dissipation to momentum equations would immediately lead to the laminarization of the flow at this Reynolds number.
3.2. **INFLUENCE OF THE NUMBER OF INNER ITERATIONS**

nor smaller than the errors of spatial and temporal discretizations. A bigger residual drop error may lead to wrong results and a smaller residual drop error compared to other errors, unnecessarily increases the computational time.

Finding the proper amount of residual drop for unsteady calculations with a theoretical approach is very difficult, if not impossible. It strongly depends on the other errors coming from boundary conditions, spatial and temporal discretization, turbulence modeling etc...

In order to have a rough estimation about the minimum number of inner iterations the same channel flow has been tested at \( Re_\tau = 180 \). Preconditioning, multigrid and residual smoothing are all switched off. The time derivative is discretized using a second order backward differencing scheme with the physical time step \( \Delta t = 0.0025 \frac{\delta u}{\tau} \). Figure (3.2) shows the effect of the number of inner iterations on the results. Four different cases with different number of inner iterations of 12, 18, 25 and 120 have been considered. The initial solution is a fully developed and statistically steady simulation obtained with an explicit method. This solution is advanced in time for \( 1.5 \frac{\delta u}{\tau} \) seconds with an implicit method using the above mentioned number of inner iterations. In figure (3.2) the profiles of the normal fluctuations \( \sqrt{v'v'} \) and mean velocity are shown. The results obtained with 25 (dashed line) and 120 (bold line) inner iterations follow closely each other, but the deviation starts when the number of inner iterations is decreased to 18 (dotted line). With 12 inner iterations (dot-dashed line) the results are completely wrong. In figures (3.3) and (3.4) part of the residual history are plotted. With 25 inner iterations the residual drop of the continuity equation is around one

![Figure 3.1: The effect of preconditioning on multigrid for the channel flow calculation with a fixed number of inner iterations, left: continuity equation, right: first momentum equation](image-url)
CHAPTER 3. COMPARATIVE STUDY

Effect of the number of inner iterations versus $y/delta$

<table>
<thead>
<tr>
<th>Inner-it</th>
<th>Turbulence Intensity ($v_{rms}/u^*$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>-1.00</td>
</tr>
<tr>
<td>25</td>
<td>-0.67</td>
</tr>
<tr>
<td>18</td>
<td>-0.33</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Effect of number of inner iterations versus $y$ (log y)

<table>
<thead>
<tr>
<th>Inner-it</th>
<th>Mean Velocity Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>0.80</td>
</tr>
<tr>
<td>25</td>
<td>1.37</td>
</tr>
<tr>
<td>18</td>
<td>1.93</td>
</tr>
<tr>
<td>12</td>
<td>2.50</td>
</tr>
</tbody>
</table>

Figure 3.2: Left: $y$ components of turbulence intensity $\sqrt{v'/u'}$, right: mean velocity profile.

Figure 3.3: Drop of the residuals for 120 (left) and 25 (right) inner iterations.
3.3. Influence of the Time Correction Technique

The time correction technique explained in section (2.6) is tested on the same channel flow. From figure (3.2) one can see that with 18 inner iterations the error related to an insufficient drop of residual deteriorates the results. This order of magnitude and one can conclude that for the present calculation, to have a proper result, the drop of the continuity equation should be minimum one order of magnitude. However, this value is very marginal and for other calculations, to be on the safe side, a bigger residual drop is always used.
error can partly be eliminated by using the time correction technique. Figure (3.5) shows the influence of the time correction on the profile of mean velocity and the normal velocity fluctuations $\sqrt{v'v'}$ after the solution is advanced in time for $1.5 \frac{\delta}{u_\tau}$ seconds from a fully developed solution of an explicit scheme.

One can see from figure (3.5) that when the time correction method is used the velocity and turbulence intensity profiles of the implicit approach (bold line) follow more smoothly the profile of the explicit approach (dotted line) compared to the case where no time correction is used (dashed line).

However, it should be mentioned that by further decreasing the number of inner iterations the time correction technique will not be effective anymore. The reason is, that the procedure does not completely eliminate the error related to an insufficient residual drop, but somehow prevents the error accumulation.

The time correction method has also been applied to the case with 12 inner iterations (results not shown). With 12 inner iterations, the residual drop error was too big to be eliminated by the time correction technique and the results, even with the time correction, were not satisfactory. On the other hand, when the number of inner iterations is big enough and the residual drop error is small, there is no need to use the time correction procedure.

In summary, this procedure is only effective at some marginal point where the number of inner iterations is “almost” sufficient to get good results and by using the time correction those small errors may be eliminated.

### 3.4 Relation Between the Time Integration and the Physical Time Step

It is well known that the backward differencing scheme, unlike the trapezoidal scheme, has a dissipation error proportional to the time step, see Hirsch [39], Moin [81]. In spite of this, the backward differencing is much more popular than the trapezoidal scheme in the CFD community.

In order to see the influence of time discretization on the LES results, a channel flow at $Re_\tau = 180$ with a mesh of $33 \times 33 \times 65$ points in the $x$, $y$ and $z$ directions is considered. The streamwise, normal and spanwise dimensions are $4\pi\delta \times 2\delta \times 2\pi\delta$, with $\delta$ the channel half-width. The initial solution is a fully developed and statistically steady simulation obtained with an explicit method.

This solution is advanced in time for $1.5\delta/u_\tau$ seconds with an implicit method. The time derivative is discretized either with a backward differencing ($\beta_1 = 1.5$, $\beta_0 = -2$, $\beta_{-1} = 0.5$, $\gamma_1 = 1.$ and $\gamma_2 = 0.0$) or a trapezoidal
3.5. THE TIME STEP AND PRECONDITIONING PARAMETERS

In preconditioning, the most important parameter to be chosen is the $\beta$ parameter. In general, the optimum value of this parameter depends on the local speed of the flow, the viscosity, and for unsteady calculations on the magnitude of the physical time step. For steady calculations a local preconditioning parameter$^3$ has always been more successful than a global

$^3$A parameter which depends on the local speed of the flow rather than a global quantity like the free-stream velocity.
CHAPTER 3. COMPARATIVE STUDY

The preconditioning parameters $\alpha$ is always taken as a constant around -1, and $\beta$ is calculated as,

$$\beta = \text{Min}(K_1 \text{Max}((u_i u_i)^{1/2}, \frac{\nu}{\delta_{\text{min}}}, \frac{l}{\pi \Delta t}), c)$$  \hspace{1cm} (3.2)

In order to assess the applicability of this formula for unsteady viscous flows, a series of tests has been carried out. A two dimensional lid driven cavity flow at $Re = \frac{U h}{\nu} = 10000$ is considered. Artifical dissipation is added to all equations and the trapezoidal method is used to discretize the time derivative. A three-level V sawtooth multigrid is also used. Both the dimension $h$ of the square cavity, and the velocity $U$ of the lid are considered to be unity. A $65 \times 65$ mesh is used with a cosine distribution of the mesh points,

$$x, y = \frac{1 - \cos \theta}{2} \hspace{1cm} 0 < \theta < \pi$$

with $\delta_{\text{min}} / h = 6.045 \times 10^{-4}$ being the closest point to the wall. Three different time steps, $\Delta t_1 = 0.002 \frac{h}{U}$, $\Delta t_2 = 0.01 \frac{h}{U}$, and $\Delta t_3 = 0.1 \frac{h}{U}$ are considered. In equation (2.75) the first argument of the $\text{Max}$ operator $(u_i u_i)^{1/2}$ is called
3.5. THE TIME STEP AND PRECONDITIONING PARAMETERS

the *inviscid velocity scale*. The biggest value of the inviscid velocity scale is when the velocity is equal to the lid velocity $U = 1$,

$$\max(u_i u_i)^{1/2} = 1 \text{ (m/s)}$$

The second argument $\frac{\nu}{\delta_{min}}$ is called the *diffusion velocity scale* and for this test case its maximum value is at the first point away from the wall,

$$\max\left(\frac{\nu}{\delta_{min}}\right) = \frac{10^{-4}}{6.045 \times 10^{-4}} = 0.165 \text{ (m/s)}$$

The third argument $\frac{1}{\pi \Delta t}$ is called the *unsteady velocity scale*. For $\Delta t_1 = 0.002\frac{h}{U}$ it is equal to,

$$\frac{h}{\pi 0.002\frac{h}{U}} = \frac{1}{\pi 0.002\frac{h}{U}} = 159.23 \text{ (m/s)}$$

As a result, the unsteady velocity scale is the dominant term in equation (2.75) and $\beta$ will be constant in the whole field.

Figure (3.8) shows the residual history of the continuity, momentum and energy equations for $\Delta t_1 = 0.002\frac{h}{U}$ and for different $\beta$ parameters ranging from 1 to 500. It can be seen from the figure that the best convergence is for $\beta$ between 500 (dot dashed line) and 100 (dotted line). This is in accordance with what equation (2.75) suggests,

$$\beta = \text{Min}(K_1 \text{Max}(1, 0.165, 159.23), 300) = \text{Min}(K_1 159.23, 300)$$

depending on the values of $K_1$, which is normally around 2, the $\beta$ parameter is around 300.

Figure (3.9) shows the same residuals for $\Delta t_2 = 0.01\frac{h}{U}$. The optimum $\beta$ is between 50 (dotted line) and 100 (dot dashed line). The unsteady velocity scale is,

$$\frac{h}{\pi 0.01\frac{h}{U}} = \frac{1}{\pi 0.01\frac{h}{U}} = 31.84 \text{ (m/s)}$$

with an adjusting coefficient $K_1 \simeq 3$ the value of $\beta$ calculated from equation (2.75) will be in the same range of $\beta$ obtained from this numerical experience.

And finally, in figure (3.10) the residuals are shown for a higher value of the physical time step, $\Delta t_3 = 0.1\frac{h}{U}$, with an unsteady velocity scale equal to,

$$\frac{h}{\pi 0.1\frac{h}{U}} = \frac{1}{\pi 0.1\frac{h}{U}} = 3.184 \text{ (m/s)}$$

Based on the convergence histories of figure (3.10), the optimum $\beta$ is between 1 (dashed line) and 10 (dotted line), which is in agreement with the above velocity scale.
CHAPTER 3. COMPARATIVE STUDY

Effect of BETPAR, 65x65 clustered mesh, DT=0.002
2D cavity flow, unsteady calculation, with multigrid.

Effect of BETPAR, 65x65 clustered mesh, DT=0.002
2D cavity flow, unsteady calculation, with multigrid.

Effect of BETPAR, 65x65 clustered mesh, DT=0.002
2D cavity flow, unsteady calculation, with multigrid.

Figure 3.8: Convergence history for different \( \beta \)'s, \( \Delta t = 0.002 \), left: continuity, middle: first momentum, right: energy equations.

Effect of BETPAR, 65x65 clustered mesh, DT=0.01
2D cavity flow, unsteady calculation, with multigrid.

Effect of BETPAR, 65x65 clustered mesh, DT=0.01
2D cavity flow, unsteady calculation, with multigrid.

Figure 3.9: Convergence history for different \( \beta \)'s, \( \Delta t = 0.01 \), left: continuity, middle: first momentum, right: energy equations.

Effect of BETPAR, 65x65 clustered mesh, DT=0.1
2D cavity flow, unsteady calculation, with multigrid.

Effect of BETPAR, 65x65 clustered mesh, DT=0.1
2D cavity flow, unsteady calculation, with multigrid.

Figure 3.10: Convergence history for different \( \beta \)'s, \( \Delta t = 0.1 \), left: continuity, middle: first momentum, right: energy equations.
Chapter 4

Final Results

Channel and cavity flows are considered to test the ability of the method on finer meshes. Before going into the details of the specific test case, some points concerning the numerical method may be highlighted.

- The calculation of the viscous fluxes needs much more computational effort compared to the one of the convective fluxes. To have a faster computation the viscous fluxes are only calculated on the finest grid level and restricted to the coarser levels via the forcing function of the multigrid, see also Vatsa et al. [112], Lacor et al. [61]. This has another advantage; the subgrid-scale eddy viscosity which is a part of the viscous fluxes routine is evaluated only on the finest level. This is in agreement with the basic assumption of LES which needs a rather fine mesh to directly simulate the big vortices and model the smaller and more isotropic ones.

- Different multigrid strategies like V-cycle and W-cycle as well as their sawtooth counterparts have been tested and finally, the V-sawtooth cycle showed to be the most robust for the present calculations.

- First order prolongation and quadratic restriction are used as multigrid transfer operators.

- A trapezoidal method is used to discretize the physical time derivative.

- A five-stage explicit Runge-Kutta scheme, with three evaluations of dissipation, Jameson [48], is used as the smoother for the multigrid. For the equation,

\[
\frac{dU}{dt} = Res_c(U) + Res_d(U)
\]
with $Res_c$ and $Res_d$ the convective and diffusive residuals, the solution at an intermediate stage $k$ is updated as,

$$U^{k+1} = U^0 + \alpha_k(Res_c(U^k) + \beta_k Res_d(U^k) + (1 - \beta_k) Res_d(U^{k-1}))$$

$\alpha$ and $\beta$ are equal to, $\alpha = \{\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1\}$, $\beta = \{1, 0, 0.56, 0, 0.44\}$. This scheme has a good damping property and a large stability region, which allows to use a bigger time step.

- For the spatial discretization a central method with Jameson-Schmidt-Turkel (JST) type artificial dissipation, Jameson et al. [49], is used. The JST type dissipation adds a first order dissipation near discontinuities\(^1\), which are always absent in the present LES calculations, and a third order background dissipation. If the third order artificial dissipation is added to the momentum equations, it will lead to the immediate laminarization of the flow. To avoid this, the artificial dissipation is only added to the continuity equation but not to the momentum and energy equations. According to our experience this also suffices to guarantee a good overall convergence in the inner iterations.

### 4.1 Channel Flow

#### 4.1.1 Test Case Description and Turbulence Statistics

The geometry of the channel is shown in figure (4.2). A mesh of $65 \times 65 \times 65$ points is used in the $x$, $y$ and $z$ directions, the streamwise, normal and spanwise dimensions are $4\pi \delta \times 2\delta \times \frac{4}{3}\pi \delta$, with $\delta$ the channel half-width. Uniform meshes with spacing $\Delta x^+ = \frac{\Delta x u_\tau}{\nu} \simeq 35$ and $\Delta z^+ = \frac{\Delta z u_\tau}{\nu} \simeq 12$ are used in the streamwise and spanwise directions. A non uniform mesh with hyperbolic tangent distribution is used in the wall-normal direction. The first mesh point away from the wall is at $y^+_1 = \frac{\Delta y u_\tau}{\nu} \simeq 0.49$ and the maximum spacing (at the centerline of the channel) is 13.8 wall units. The following transformation gives the location of grid points in the $y$ direction,

$$y_i = \frac{1}{a} \tanh[\xi_i \arctanh(a)] \quad (4.1)$$

with,

$$\xi_i = -1 + 2(j - 1)/(N_y - 1) \quad (j = 1, 2, \cdots, N_y)$$

Here $a$ is the adjustable parameter of the transformation ($0 < a < 1$); a large value of $a$ distributes more points near the walls. In the present computations, $a = 0.98346$, $N_y = 65$.

---
\(^1\)Like shock waves.
4.1. CHANNEL FLOW

Figure 4.1: Variation of $C\Delta^2$ defined in Eq. (1.7) (left figure), and turbulent Prandtl number defined in Eq. (1.29) (right figure) with distance from the wall.

The Reynolds number based on the friction velocity is $Re_{\tau} = 180$ and the Mach number at the centerline is $M = 0.06$. A dynamic procedure, Germano et al. [32], is used to calculate the Smagorinsky coefficient $C$ and the turbulent Prandtl number $Pr_t$. In figure (4.1) the product $C\Delta^2$, (with $\Delta^2 = \Delta x \Delta y \Delta z)^{2/3}$) and the turbulent Prandtl number $Pr_t$ are plotted as a function of the wall coordinate $y^+$.

The preconditioning parameters $\alpha = -1$ and $\beta = 0.14 \frac{L_x}{\pi \Delta t}$. For this test case, $\Delta t = 0.01 \delta / u_{\tau}$ and $L_x / \delta = 4\pi$, and in equation (2.75) the third argument of the $\text{Max}$ operator equals, $\frac{L_x}{\pi \Delta t} = 400 u_{\tau}$. The smallest cell length is the $y$ coordinate of the first mesh point away from the wall and the second argument of the $\text{Max}$ operator in equation (2.75) equals, $\frac{\nu}{u_{\tau} \delta_{\min}} = 1.15 u_{\tau}$. The first argument of the $\text{Max}$ operator, $(u_i u_i)^{1/2}$ reaches its maximum value at the centerline of the channel which is equal to, $(u_i u_i)^{1/2} = 18 u_{\tau}$. As a result, everywhere in the channel, $\text{Max}((u_i u_i)^{1/2}, \frac{\nu}{\delta_{\min}} \frac{L_x}{\pi \Delta t}) = \frac{L_x}{\pi \Delta t}$ and when $\beta$ is calculated according to equation (2.75), it should be constant in the whole field. The $\beta$ parameter in terms of the velocity is equal to $\beta = 3.16 U_{\text{center}}$, with $U_{\text{center}}$ the centerline velocity.

The physical constraint requires $\Delta t$ to be less than the time scale of the smallest resolved scale of motion, i.e., $CFL_c = \frac{\Delta t U_c}{\Delta x} \leq 1$, with $U_c$ a convective velocity. For the channel flow the maximum $U_c$ is at the center of the channel. In order to check if the assumed time step satisfies this condition or not let us calculate the $CFL_c$ at the channel center. In the present calculation at the center of the channel $U_c / u_{\tau} = 18.58$ and $\Delta x = L_x / (N_x - 1) = 4\pi \delta / 64$. The $CFL_c$ may be written as,

$$CFL_c = \frac{\Delta t U_c}{\Delta x} = \frac{(0.01 \delta / u_{\tau})(18.58 u_{\tau})}{4\pi \delta / 64} = 0.946 < 1 \quad (4.2)$$
Therefore, the physical condition is satisfied.

For this test case the effect of residual smoothing was not clear. After using residual smoothing the convergence of the continuity equation was improved but the convergence of the momentum and energy equations were better without residual smoothing. The reason can be the absence of artificial viscosity in those equations. To be on the safe side, the residual smoothing was switched off for this test case.

The mean velocity profile and turbulence intensities are shown in figures (4.3) to (4.6). The results are compared with the DNS data of Kim et al. [52], and the LES of Morinishi [82]. The results of Morinishi are also obtained with a second order spatial discretization but on a coarser mesh of $32 \times 65 \times 32$. In the same figures the results of a coarse mesh simulation, i.e., $33 \times 33 \times 65$ points in $x, y$ and $z$ directions, are also plotted for comparison.

In figure (4.7), resolvable $< u'v' >$, viscous $< \nu \frac{\partial u}{\partial y} >$, modeled (SGS) $< \nu_l \frac{\partial u}{\partial y} >$ and total (resolvable+viscous+modeled) shear stresses are shown. As expected, the total shear stress is a straight line with a slope equal to one.

Fluid statistics were averaged for $11.5 \delta / u_\tau$. The physical time step is around 400 times bigger than the time step of a purely explicit method, the residual drop of the continuity equation is fixed to -2.5 and this requires almost 100 inner iterations. Increasing the number of inner iterations and the absolute value of the residual drop showed to be unnecessary as the results remained the same for higher number of inner iterations. The gain achieved compared with a purely explicit method is roughly equal to $\frac{400}{100} = 4$. 

Figure 4.2: Channel flow test case.
4.1. CHANNEL FLOW

Figure 4.3: Mean streamwise velocity of channel flow compared with DNS of Kim [52] and second order LES of Morinishi [82].

Figure 4.4: $\sqrt{u'u'}$ turbulence intensities for channel flow with fine $65^3$ and coarse $33 \times 33 \times 65$ meshes compared with DNS of Kim [52] and second order LES of Morinishi [82].
Figure 4.5: $\sqrt{\text{v}'\text{v}'}$ turbulence intensities for channel flow.

Figure 4.6: $\sqrt{\text{w}'\text{w}'}$ turbulence intensities for channel flow.
4.1. CHANNEL FLOW

Resolved modeled and total turbulence shear stress

\[ y/\Delta \]

-1.00  -0.80  -0.60  -0.40  -0.20  0.00

1 Viscous Stress
2 Modeled
3 Resolvable
(1+2+3) Total

Figure 4.7: Resolvable \( u'u' \), viscous \( \nu \partial u/\partial y \), modeled \( \nu \partial u/\partial y \) and total (resolvable+viscous+modeled) shear stress.

4.1.2 One Dimensional Energy Spectra

The energy spectra at two different locations of \( y^+ = 6.09 \) and \( y^+ = 144.54 \) compared with DNS of Kim et al. [52] are shown in figures (4.8) and (4.9). The mesh used for DNS is not the same as the mesh of the present calculation and in DNS the energy spectra are calculated at slightly different wall distances of \( y^+ = 5.39 \) and \( y^+ = 149.53 \).

We will briefly explain how the streamwise \( \langle E_{uu} \rangle \), \( \langle E_{vv} \rangle \), \( \langle E_{ww} \rangle \) and spanwise \( \langle E_{uu} \rangle \), \( \langle E_{vv} \rangle \), \( \langle E_{ww} \rangle \) energy spectra are calculated. As an example, it will be shown how streamwise energy spectra are calculated.

Suppose an \( xz \) plane parallel to the wall and at a distance of \( y = y_c \) from the wall. If \( \phi(x, y, z) \) denotes the velocities \( \phi = (u, v, w) \), the first step is to calculate, at an arbitrary spanwise location \( z_c \), the energy spectrum of the periodic function \( \phi(x, y_c, z_c) \) varying in the streamwise \( x \) direction.

If the number of cells (not points, because finite volume is used) in the streamwise \( x \) direction is \( N_x \), and \( L_x \) is the period of \( \phi(x, y_c, z_c) \), the complex spectrum of \( \phi(x, y_c, z_c) \) denoted by \( \hat{\phi}(k_x) \) is calculated as,

\[
\hat{\phi}(k_x) = \frac{1}{N_x} \sum_{j=0}^{N_x-1} \phi(x_j, y_c, z_c) e^{-ik_x x_j}, \quad k_x = \frac{2\pi}{L_x} k, \quad -\frac{N_x}{2} \leq k \leq \frac{N_x}{2} - 1
\]

with \( \phi(x_0, y_c, z_c) \) and \( \phi(x_{N_x-1}, y_c, z_c) \) the values of function at the first and last cells, respectively, and \( x_j = \frac{L_x j}{N_x} \).

Once \( \hat{\phi}(k_x) \) has been calculated, the energy spectrum \( E_{\phi\phi}(k_x) \) can be
written as,

\[ E_{\phi\phi}(k_x) = \hat{\phi}(k_x)\hat{\phi}^*(k_x) \]  

(4.4)

with \( \hat{\phi}^*(k_x) \) the complex conjugate of \( \hat{\phi}(k_x) \). The same procedure is applied to other streamwise lines of the \( xz \) plane. And finally, \( E_{\phi\phi}(k_x) \) are averaged over the spanwise direction.

For \( E_{\phi\phi}(k_z) \) the function is supposed to be periodic in the \( z \) direction and after the energy spectra are calculated, they are averaged in the streamwise direction.

### 4.2 Cavity Flow

#### 4.2.1 Test Case Description and Turbulence Statistics

A lid driven cavity flow is considered as another example to test the ability of the present approach. The motion of a Newtonian fluid within a lid-driven rectangular three-dimensional cavity is maintained by the continuous diffusion of kinetic energy from the moving wall. This energy is initially confined to a very thin viscous layer of fluid next to the moving boundary. The blocking action of the bounding walls and a great variety of hydrodynamic phenomena unevenly redistribute this energy over most of the cavity volume.

The geometry is shown in figure (4.10). The flow is driven by the top wall, \( z = h \), in the \( x \) direction moving with a velocity \( U = 1 \). In the vicinity of the lid the maximum Mach number of the flow is equal to \( M = 0.003 \), but according to the velocity distributions, figure (4.11), the average Mach number...
### 4.2. CAVITY FLOW

Figure 4.9: One dimensional energy spectra, left: streamwise, right: spanwise, at $y^+ = 144.54$ compared with DNS of Kim et al. [52] at $y^+ = 149.53$.

The number of the entire flow field is much smaller and roughly equal to $M = 0.001$. The Reynolds number is $Re = Uh/\nu = 10000$ where $h$ is the dimension of the cube. A $33^3$ mesh is used with a cosine distribution of the mesh points. Starting from a stagnant flow, the flow is fully developed after $32h/U$, after which statistics were collected during a period of $32h/U$.

The mean velocity profiles along the vertical and horizontal symmetry lines are shown in figure (4.11). The statistics of the fluctuating field are compared in figures (4.11) to (4.17). The agreement of the present simulation with the experiment of Prasad et al. [89], and the DNS data of Leriche et al. [66] is satisfactory.

A constant coefficient Smagorinsky model with $C = 0.0085$, see equation (1.12), combined with near wall damping is used for subgrid scale modeling. The preconditioning parameters $\alpha = -1$ and $\beta = 0.1\frac{L}{\pi\Delta t}$, with $\Delta t = 0.01h/U$. The $\beta$ parameter in terms of the velocity is $\beta = 3.16U$. By using residual smoothing a better convergence was obtained for all equations. The smoothing parameter is set according to Radespiel et al. [90], and the CFL number is increased by a factor of two, $\frac{CFL_{smooth}}{CFL} = 2$.

To have an estimation for the value of $CFL_c = \frac{\Delta tU_c}{\Delta x}$, which necessarily should be smaller than one to have a time accurate simulation, we calculate the $CFL_c$ number of a computational cell in the vicinity of the moving lid. We suppose that $U_c$ is roughly equal to the velocity of the lid $U_c = U$. The length of a computational cell away from the side walls and in the center of the top wall is around $\Delta x \simeq 0.05h$. The $CFL_c$ equals,

$$CFL_c = \frac{\Delta tU_c}{\Delta x} = \frac{(0.01h/U)(U)}{0.05h} = 0.25 < 1$$  \hspace{1cm} (4.5)
The physical time step $\Delta t$ is around 1000 times bigger than the maximum time step allowed by the stability condition of an explicit scheme in a compressible flow. To have a well resolved wall-layer near the moving lid, one needs a more clustered mesh near the wall compared to that of the channel flow. This explains why the explicit time step for the cavity flow is much smaller than that of the channel flow, and using an implicit approach is even more justified for the cavity flow calculation.

In almost 130 inner iterations the residual of the continuity equation drops four orders of magnitude. As a rough estimation, and without taking into account the extra work of the multigrid, the implicit approach is around $1000/130 \approx 7.5$ times faster than a purely explicit approach.

4.2.2 The Mean Velocity Field

The dominant feature of the mean flow is the large-scale recirculation which appears to span the cavity. Figure (4.18) shows the typical time-averaged flow structure on a central vertical plane (the symmetry plane). The flow consists of one primary eddy and three secondary eddies, see Kosef et al. ([56] and [57]) for experimental visualizations.

The eddies on the lower right and left corners of the cavity are called the “downstream secondary eddy” and the “upstream secondary eddy”, respectively. The eddy on the upper left corner is called the “upper secondary eddy”.

The contact of the rotating fluid mass with the side walls ($xz$ planes at $y = 0$ and $y = h$) leads to an axial (along $y$) flow towards the mid-plane ($xz$ plane at $y = h/2$). This can be explained in the following way; the fluid elements which rotate at a large distance from the side walls are in equilibrium under the influence of the centrifugal force which is balanced...
4.2. CAVITY FLOW

Figure 4.11: Mean velocity profile along two symmetry lines (ab and cd), lines are used for the simulation data and markers for experimental data.

Figure 4.12: Statistics of the fluctuating velocity $10^{\frac{u' u'}{\bar{U}^2}}$ along the $x$ axis, line $ab$, $y = z = h/2$, lines are used for the simulation data and markers for experimental data.
CHAPTER 4. FINAL RESULTS

Cavity flow, Re=10000

Figure 4.13: Statistics of the fluctuating velocity $10 \frac{\sqrt{\text{u}'\text{w}'}}{U_0}$ along the $x$ axis, line $ab$, $y = z = h/2$.

Figure 4.14: Statistics of the fluctuating velocity $100 \frac{\text{u}'\text{w}'}{U_0^2}$ along the $x$ axis, line $ab$, $y = z = h/2$
4.2. **CAVITY FLOW**

\[ z/h \]

\[ 10\sqrt{u'/u} \]

\[ 10^w/U \]

**Figure 4.15:** Statistics of the fluctuating velocity $10\sqrt{u'/u}$ along the $z$ axis, line $cd$, $(x = y = h/2)$, lines are used for the simulation data and markers for experimental data.

**Figure 4.16:** Statistics of the fluctuating velocity $10\sqrt{w'/w}$ along the $z$ axis, line $cd$, $(x = y = h/2)$. 
by a radial pressure gradient. The peripheral velocity of the elements near the side walls is reduced, thus decreasing the centrifugal force, whereas the pressure gradient in $y$ direction remain the same. This set of circumstances causes the fluid elements near the side walls to flow radially inwards (to the center of the side walls), and for reasons of continuity that motion must be compensated by an axial flow (along $y$) towards the mid-plane. According to Leriche et al. [66], the maximum axial velocity due to this mechanism is less than 1% of the lid velocity. For a complete discussion about axially symmetrical boundary layers, see Schlichting [97].

As a result of this axial ($y$) flow, there is not a single main eddie but two large eddies, one on each side of the symmetry plane at $y = h/2$, which have similar structure. The flow near the symmetry plane is a transition zone between the two systems.

### 4.2.3 The Three-Dimensional Picture

A fourth major secondary structure, which is not seen in $xz$ planes, called “corner vortex” has a significant influence on the flow and is the most evident manifestation of three-dimensionality in the flow. The corner vortex originates from the adjustment of the shear and pressure forces acting on the recirculating fluid. An example of a corner vortex is shown in figure (4.21, 

---

2 The same phenomenon is observed in a teacup! After stirring, the radial inward flow near the bottom brings the tea leaves to the center.
4.2. CAVITY FLOW

The sense of rotation of these vortices is consistent with the direction of the axial ($y$) flow towards the symmetry plane.

Koseff et al. [56] used a dye streak to visualize the vortical motion of the flow in the downstream secondary eddy (DSE) region. They showed that the DSE is divided into two parts and each part moves in two opposite directions (along the $y$ axis) towards the side walls ($xz$ planes at $y = 0$ and $y = h$). And then, due to the presence of the walls they change their directions and they continue moving in the opposite $x$ direction. This explanation is in accordance with the present simulation. In figure (4.20) one can see the two corner vortices, but in figure (4.20, left), which is in a plane further from the DSE, there are no vortices on the lower part of the cavity. However, in figure (4.20, left) one can see two other vortices on the upper part of the cavity. The presence of these two upper vortices was not reported on the experimental work of Koseff et al. ([56], [57]).

Figures (4.20, right) and (4.21, right) show the instantaneous vectorlines projected on two different cross sections. In both pictures the Taylor-Görtler-like (TGL) vortices are clearly visible. The longitudinal Taylor-Görtler-like (TGL) vortices are formed because the surface of separation between the primary eddy and the DSE is effectively a concave “wall”. The curved surface leads to the formation of TGL structure. In the experimental work of Koseff et al. [57] using the rheoscopic liquid technique, in which crystalline particles are placed in the flowfield and the flow is illuminated with a sheet of light, the TGL vortices have been visualized. The corner vortex, the DSE and the TGL vortices are all integrally inter-related. The length of the TGL vortices is smaller than the total length of the cavity. As can be seen in figures (4.19), no vortical motion can be observed in a plane far away from the DSE region.

Koseff [55] showed that the size and distribution of the TGL vortices is uneven and varies with time, and this behavior is a function of Reynolds number. At lower Reynolds numbers the TGL vortices are more uniform in size and evenly distributed. In the present calculation $Re=10000$ and the vortices meander in the $y$ direction. Thus, in figures (4.20, left) and (4.21, left) due to the time averaging, the time averaged vectorlines do not show the TGL vortices.
Figure 4.18: Time averaged vectorlines projected on the symmetry plane.

Figure 4.19: Streamlines at plane $x = 0.17/h$, left: time averaged, right: instantaneous.
4.2. CAVITY FLOW

Figure 4.20: Streamlines at plane $x = 0.38/h$, left: time averaged, right: instantaneous.

Figure 4.21: Streamlines at plane $x = 0.8/h$, left: time averaged vectorlines showing the presence of corner vortices. Right: instantaneous vectorlines showing the presence of Taylor-Görtler like vortices.
Chapter 5

Particle Laden Flows

5.1 Introduction and Literature Survey

Particle-laden turbulent flows are important in numerous industrial processes, such as coal combustion, dust deposition and removal in clean rooms, droplets deposition in gas-liquid flows, etc... Accurate prediction of particle-laden turbulence is important in order to gain a better understanding of particle transport by turbulence as well as ultimately improve the design of the engineering devices in which two-phase flows occur.

Traditional methods are usually based on the Reynolds-Averaged Navier-Stokes (RANS) equations in which the entire spectrum of velocity fluctuations is represented indirectly using various parametrizations, see for example Chang et al. [14] as well as Berlemont et al. [6]. A primary shortcoming of RANS methods for the prediction of particle-laden turbulent flows is related to deficiencies associated with the model used to predict properties of the Eulerian turbulence field. This problem can be solved by using a more sophisticated approach like DNS but the drawback is that it is only applicable to low or moderate Reynolds numbers. In between there is the large-eddy simulation (LES) which is less sensitive to modeling errors than in RANS calculation and less restricted to low Reynolds number than DNS, see for a review Yeung [118], Crowe et al. [21], Michaelides [78] and Peng et al. [86].

Concerning the effect of the particles on the carrier flow, two different approaches called one-way and two-way coupling can be used. In a one-way coupling model the presence of particles has negligible effect on the carrier flow but in a two-way coupling model the effects of particles is taken into account in the carrier flow.

Elghobashi [27] has proposed the map shown in figure (5.1) for the effect of particles on carrier-phase turbulence. For volume fractions\(^1\) less than 10\(^{-6}\)

\(^1\)The volume fraction is defined as the ratio of the total volume of the particle to the...
the presence of the particles would have no effects on turbulence. For volume fractions between $10^{-6}$ and $10^{-3}$, the particles can augment the turbulence, if the ratio of particle response time\(^2\) to the turnover time of a large eddy is greater than unity, or attenuate the turbulence, if the ratio is less than unity. Chen et al. [15] came to the same conclusion by conducting an experiment in a vertical wind tunnel. They denoted turbulence modulation, which generally decreases turbulence fluctuations, when the dispersed-phase motion accommodates to the continuous-phase motion, and turbulence generation, which generally increases turbulence fluctuations, when the continuous-phase velocity field is disturbed by particles wakes. Their studies show that turbulence generation (modulation) dominates turbulence modification, tending to increase (decrease) turbulence levels, when dispersed-phase elements have large (small) relaxation times compared to characteristic turbulence timescales.

For volume fractions greater than $10^{-3}$, particle-particle collisions become important, and the turbulence of the carrier phase can be affected by the oscillatory motion due to particle collisions. Elghobashi [27] identifies this effect as four-way coupling.

If two-way coupling effects are significant, then the subgrid-scale turbulence model might need modification. This would be especially important if the particulate phase couples strongly with the small-scale turbulence. Whether this is the situation or not may depend upon the matching of the total volume of the carrier flow.

\(^2\)The particle response time (also called relaxation time) is defined as, $\tau_p = \frac{\rho_p d^2}{18 \nu \rho_f}$, where $\rho_p$ and $\rho_f$ are the density of the particle and the flow, respectively. $d$ is the diameter of the particle and $\nu$ is the kinematic viscosity of the flow.
5.1. INTRODUCTION AND LITERATURE SURVEY

time-scale between the particles and the sub-grid-scale turbulence.

Armenio et al. [4] investigated the effect of the turbulent fluctuations on the particle motions by computing particle statistics obtained by the filtered DNS of turbulent channel flow at $Re_\tau = 175$. They used filtering to eliminate the effect of the subgrid scale fluctuations on the particles. Their minimum filter ratio was equal to two and the maximum was equal to four. Their results show that the subgrid fluctuations have a limited effect on particle statistics. The difference between DNS data and filtered results is more evident for particles initially located in the buffer layer, and decreases for particles released in the core of the channel.

They explain this behaviour as follows. In the core of the channel and away from the near-wall region, the particle motion is, to a large extent, governed by the large-scale, energy-carrying structures, and the effect of small-scale fluctuating velocity field is negligible. In the buffer layer and near the wall, on the other hand, the scale of the largest structures decreases, and the particle motion becomes more sensitive to fluctuations. However, the effect of filtering was only significant at the largest filter ratio.

Wang et al. [116] investigated the effect of subgrid scales on particle statistics in a channel flow at $Re_\tau = 640$ by solving a transport equation for the subgrid-scale (SGS) kinetic energy, which allowed them to model the fluctuating SGS velocities as a random process with Gaussian distribution and amplitude obtained from the SGS kinetic energy; these velocities were then added to the filtered ones. According to their results, the addition of SGS velocities has negligible effect on particle statistics.

Boivin et al. [8] conducted several a priori tests of subgrid-scale turbulence models utilizing results from direct numerical simulation (DNS) of a forced homogeneous isotropic turbulent flow using a two-way coupling and including the back effect of the particles. They stated that the dynamic subgrid model yields more accurate predictions than the conventional Smagorinsky model. This conclusion was further confirmed by a posteriori tests. They consider heavy particles with diameters very small compared to the grid spacing, but with relaxation times larger than the time scales of the subgrid motions and comparable to the time scales of the large eddies. They also pointed out that subgrid-scale modeling of the fluid turbulence in gas-solid flows may benefit from the fact that a large part of the total dissipation in the flow arises from fluid-particle interactions, especially at larger mass loadings. Consequently, modeling errors could have a smaller impact on LES predictions than in a single-phase flows, especially for large particle response times and at high mass loading.

Miller et al. [79] calculated a transitional droplet laden mixing layer with direct numerical simulation (DNS) and made a subgrid-scale analysis.
They defined a “subgrid Stokes number”: $St_{sg} = \frac{\tau_p}{\tau_{sg}}$, where $\tau_p$ is the particle relaxation time and $\tau_{sg}$ is the subgrid time scale based on the subgrid velocity variance and the filter width. They showed that subgrid effects on the droplet trajectories can only be neglected safely for a subgrid Stokes number substantially greater than unity.

The near wall behavior of particles has been studied by many researchers. To mention a few among others, Haarlem et al. [35] performed a DNS of a channel bounded by a no-slip and free-slip surface and they investigated the effect of wall boundary condition. The DNS of particle flow in the wall region of a horizontal channel was carried out by Pedinotti et al. [85].

In this chapter the numerical method to solve the incompressible Navier-Stokes is explained first. Then, the finite volume method is briefly explained as well as the use of compact schemes. An overview of the particle motion simulation including the calculation of particle trajectories, particle velocity statistics, different types of interpolation, initialization, time advancement and walk search algorithm is given in section (5.4). Later, the results of the particle-laden channel flow with different types of particles are represented. The effect of interpolation on the results is also investigated.

## 5.2 Incompressible Navier-Stokes Solver

The method used to solve the incompressible Navier-Stokes equations is a subset of the pressure correction method originally applied by Harlow and Welch [38] for the computation of free surface incompressible flows. It is called fractional step or projection method, developed independently by Chorin [19] and Temam ([106], [107]).

The fractional step method is slightly different from the original pressure correction method in a way that the pressure term is removed in the intermediate step. In this section we explain the fractional step method.

The governing equations for an incompressible flow are,

$$\frac{\partial u_i}{\partial x_i} = 0$$  \hspace{1cm} (5.1)

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}$$  \hspace{1cm} (5.2)

where $x_i$’s are the Cartesian coordinates, and $u_i$’s are the corresponding velocity components. The integration method is an explicit four stage Runge-Kutta method where $\tau_{sg} = \Delta_f / u'' u''_u^{1/2}$, where $\Delta_f$ is the filter width, and $u''$ is the fluctuating velocity with respect to the Favre density weighted filtered velocity $\bar{u}$, i.e., $u'' = u - \bar{u}$. 

\[3\]
Kutta scheme, equation (2.2), with \( s = 4 \) and,

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 \\
0 & 1/3 & 0 & 0 \\
0 & 0 & 1/2 & 0 \\
\end{bmatrix}
\begin{bmatrix}
a_{ij} \\
b_j
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 & 1
\end{bmatrix}
\]

The fractional step method, or time-split method, is used to solve the system of equations (5.2). To explain the fractional step method we assume, for the sake of clarity, that the time derivative is discretized with an explicit forward Euler method. The extension to a multi-stage Runge-Kutta method is straightforward as the same procedure is followed at every stage.

The discretized form of equation (5.2) may be written as,

\[
\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{\partial u_i^n u_j^n}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i^n}{\partial x_j \partial x_j} 
\]

In an incompressible flow the pressure cannot be updated from either the momentum, nor the continuity equation. However, with the following method a pressure field is calculated which satisfies the continuity equation.

First, by dropping the gradient of pressure from equation (5.3) an intermediate velocity \( u_i^* \) is obtained,

\[
\frac{u_i^* - u_i^n}{\Delta t} + \frac{\partial u_i^n u_j^n}{\partial x_j} = \nu \frac{\partial^2 u_i^n}{\partial x_j \partial x_j} 
\]

The velocity at the next time level \( u_i^{n+1} \) is equal to \( u_i^* \) plus a correction velocity \( u_i^{cor} \),

\[
u^{n+1} = u_i^* + u_i^{cor}
\]

The velocity field at time level \( n+1 \) must be divergence free, i.e.,

\[
\frac{\partial u_i^{n+1}}{\partial x_i} = 0 \rightarrow \frac{\partial u_i^*}{\partial x_i} = -\frac{\partial u_i^{cor}}{\partial x_i}
\]

By inserting the value of \( u_i^{n+1} \) from (5.5) into equation (5.3),

\[
\frac{(u_i^* + u_i^{cor}) - u_i^n}{\Delta t} + \frac{\partial u_i^n u_j^n}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i^n}{\partial x_j \partial x_j}
\]

after subtracting equation (5.4) from equation (5.7),

\[
\frac{u_i^{cor}}{\Delta t} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i}
\]
By applying the divergence operator to both sides of equation (5.8),

$$\frac{1}{\Delta t} \frac{\partial u^\text{cor}_i}{\partial x_i} = -\frac{1}{\rho} \frac{\partial^2 p}{\partial x_i \partial x_i}$$  \hspace{1cm} (5.9)

from (5.6) the value of $\frac{\partial u^\text{cor}_i}{\partial x_i}$ is known and is inserted into (5.9),

$$\frac{1}{\Delta t} \frac{\partial u^*_i}{\partial x_i} = \frac{1}{\rho} \frac{\partial^2 p}{\partial x_i \partial x_i}$$  \hspace{1cm} (5.10)

Once the pressure field, which satisfies the continuity equation, is calculated from equation (5.10), the velocity at time level $n + 1$ can be obtained from equation (5.5) and (5.8),

$$u^*_i = u^*_i - \Delta t \frac{\partial p}{\partial x_i}$$  \hspace{1cm} (5.11)

### 5.3 Finite Volume Compact Scheme

The space derivatives are discretized with a finite volume compact scheme using a cell-averaged approach. First, in order to highlight the advantage of a cell-averaged approach over a point-wise approach for high order unsteady calculations let us consider the following equation,

$$\frac{\partial U}{\partial t} + \nabla \cdot F = \nabla \cdot Q$$  \hspace{1cm} (5.12)

with $U$ the vector of independent variables, $F$ the vector of convective fluxes, and $Q$ the vector of viscous fluxes. The finite volume method can be applied to a volume of control $V$,

$$\int \frac{\partial U}{\partial t} dV + \int \nabla \cdot F dV = \int \nabla \cdot Q dV$$  \hspace{1cm} (5.13)

Let $\overline{U}$ denote the cell-averaged value of $U$,

$$\overline{U} = \frac{1}{V} \int U dV$$  \hspace{1cm} (5.14)

With a cell-averaged approach, the volume integral of the time derivative in equation (5.13) can be exactly calculated and this term therefore does not contribute to spatial errors,

$$\frac{\partial \overline{U}}{\partial t} V + \int \nabla \cdot F dV = \int \nabla \cdot Q dV$$  \hspace{1cm} (5.15)
5.3. FINITE VOLUME COMPACT SCHEME

However, if a point-wise approach was used, the time derivative would add a spatial second-order error to the equations.4

After this short introduction about the necessity of a cell-averaged approach for unsteady calculations, we proceed with the methods of discretizing the space derivatives with compact schemes. Using compact schemes in a finite volume context is part of the research conducted at the department. The interested reader can refer to Smirnov et al. [102], [103] and Lacor et al. [94] for further readings.

5.3.1 Convective Fluxes

Before going to three dimensions, a one dimensional case is considered to demonstrate the main aspects of the present approach. Suppose the cell-averaged values of the independent variable \( u \) are known, the first step is to calculate the values of \( u \) at the interfaces. In 1D the interfaces are points, but in 2D and 3D these are lines and surfaces, respectively, which require special attention. We will address this problem later in this section. A compact interpolation formula may be written as,

\[
\beta u_{i-3/2} + \alpha u_{i-1/2} + u_{i+1/2} + \alpha u_{i+3/2} + \beta u_{i+5/2} = \quad (5.16)
\]

The coefficients \( a, b, c \) and \( \alpha, \beta \) can be derived by developing all the variables in a Taylor series about point \( i + 1/2 \), and requiring all the coefficients of the resulting expansion to vanish up to some definite term. It is worthwhile to mention that the coefficients obtained for the cell-averaged approach would be different from the coefficients of a point-wise approach for the same order of accuracy. The reason is the way the Taylor expansions are calculated for the cell-averaged values. For example the Taylor expansion of \( \pi_{i+1} \) around point \( i + 1/2 \) would be,

\[
\pi_{i+1} = \frac{1}{x_{i+1/2} - x_{i+1/2}} \int_{x_{i+1/2}}^{x_{i+3/2}} u(x) dx = \frac{1}{x_{i+1/2} - x_{i+1/2}} \times \\
\int_{x_{i+1/2}}^{x_{i+3/2}} \left( u_{i+1/2} + (x - x_{i+1/2}) \frac{\partial u}{\partial x} |_{i+1/2} + \frac{1}{2}(x - x_{i+1/2})^2 \frac{\partial^2 u}{\partial x^2} |_{i+1/2} + \cdots \right) dx
\]

\footnote{It is easy to check that, \( \bar{U} = U_{\text{center}} + O(h^2) \)}

with \( \bar{U} \) the cell averaged value, \( U_{\text{center}} \) the value at the center of the cell, and \( h \) the grid spacing.
CHAPTER 5. PARTICLE LADEN FLOWS

The approximation (5.16) is fourth order accurate if the following relations are satisfied,

\[ a + b + c = 1 + 2\alpha + 2\beta \]
\[ a + 7b + 19c = 6(\alpha + 4\beta) \]

In the present study the following fourth order interpolation is used,

\[ \frac{1}{4}u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}} + \frac{1}{4}u_{i+\frac{3}{2}} = \frac{3}{4}(\bar{u}_{i+1} + \bar{u}_i) \tag{5.17} \]

The extension to two and three dimensions is straightforward. In 3D, for a Cartesian mesh, the formulation may be written as,

\[ \frac{1}{4}[u]_{i-\frac{1}{2},j,k} + [u]_{i+\frac{1}{2},j,k} + \frac{1}{4}[u]_{i+\frac{3}{2},j,k} = \frac{3}{4}(\bar{u}_{i+1,j,k} + \bar{u}_{i,j,k}) \tag{5.18} \]

\([\phi]\) denotes the interface averaged values of a primitive variables \(\phi_{i+\frac{1}{2},j,k}\). Once the values of the primitive variables at the interfaces have been calculated, the next step is to calculate the fluxes at the interfaces.

\[ \int \nabla \cdot F dV = \int F \cdot dS = \sum_{i=1}^{n_f} \bar{F}_i \Delta S_i \]

with \(\bar{F}_i\) the interface-averaged values of fluxes, \(\Delta S_i\) the surface area of each face and \(n_f\) the number of faces of the control volume. In 3D and 2D the interfaces are surfaces and lines, respectively.

To calculate the fluxes at the interface, the interface-averaged value of the product of primitive variables \([uv]\) is needed. Special care should be taken in calculating the interface-averaged of the products to keep the high order of accuracy of the method. The interface-averaged of the product of \(u\) and \(v\) is,

\[ [uv] = \frac{1}{y_A - y_B} \int_A^B uv dy \tag{5.19} \]

to calculate \([uv]\) the following approximation may be used,

\[ [uv] = [u][v] + O(\Delta y)^2 \tag{5.20} \]

with \(\Delta y = y_A - y_B\). The truncation error of equation (5.20) can be easily calculated by developing \(u\) and \(v\) in a Taylor series around point \(M\), figure (5.2). Kobayashi [54] proposed to recover 4th order of accuracy by approximating the leading truncation error of order 2 by finite-difference formula of second order accuracy,

\[ [uv] = [u][v] + \frac{\partial u}{\partial y} M \frac{\partial v}{\partial y} M \frac{\Delta y^2}{12} + O(\Delta y)^4 \tag{5.21} \]
5.3. FINITE VOLUME COMPACT SCHEME

The derivatives $\frac{\partial u}{\partial y}|_M$ and $\frac{\partial v}{\partial y}|_M$ are approximated by a classical second order finite-difference method,

$$\frac{\partial u}{\partial y}|_M = \frac{u_A - u_B}{\Delta y} + O(\Delta y)^2 \quad (5.22)$$

This method can also be generalized for the use on arbitrary meshes, Smirnov et al. [102].

5.3.2 Viscous Fluxes

In a finite volume context the viscous fluxes are evaluated as follows,

$$\int \nabla \cdot QdV = \int Q \cdot dS = \sum_{i=1}^{n_f} \overline{Q}_i \Delta S_i \quad (5.23)$$

with $\overline{Q}_i$ the interface-averaged values of viscous fluxes, $\Delta S_i$ the surface area of each face and $n_f$ the number of faces of the control volume. The calculation of viscous fluxes can be done in two steps,

- consider the rectangle $ABCD$ in figure (5.2). The interface-averaged values of primitive variables $[u], [v]$ at $AB, BC, CD$ and $DA$ are known. By applying the Gauss theorem the cell-averaged values of the velocity gradients $\frac{\partial u}{\partial x_i}$ and $\frac{\partial v}{\partial x_i}$ are calculated.

- in a similar way to equation (5.18), the interface-averaged values of gradients, $([\frac{\partial u}{\partial x_i}], [\frac{\partial v}{\partial x_i}])$, are evaluated from their cell-averaged $([\frac{\partial u}{\partial x_i}], [\frac{\partial v}{\partial x_i}])$ counterparts.

Once the interface-averaged values of gradients are known, the calculation of $\sum_{i=1}^{n_f} \overline{Q}_i \Delta S_i$ in equation (5.23) is straightforward.
5.3.3 Application to Channel Flow

The compact scheme is applied to the channel flow test case at $Re_\tau = 180$. However, to save some computational time, the compact scheme is only used for the convective fluxes in the streamwise and spanwise directions. The viscous and convective fluxes in the normal directions are calculated with a standard second order method. The viscous fluxes in the spanwise and streamwise directions are obtained with a classical (not compact) fourth order method. The overall accuracy of the method is then $O(\Delta x)^4 + O(\Delta z)^4 + O(\Delta y)^2$.

Convective Fluxes

The convective flux in the normal direction ($xz$ planes) is calculated with a standard second order difference. The values at the interface of two cells $(i, j, k)$ and $(i, j + 1, k)$ are obtained with a linear interpolation $\phi = (u, v, w)$,

$$\phi|_{i,j+\frac{1}{2},k} = \frac{\Delta y_1 \bar{\phi}_{i,j+1,k} + \Delta y_2 \bar{\phi}_{i,j,k}}{\Delta y_1 + \Delta y_2} \quad (5.24)$$

with $\Delta y_1, \Delta y_2$ the distances between the cell centers of $(i, j, k), (i, j + 1, k)$ and the center of the interface $(i, j + \frac{1}{2}, k)$.

In the streamwise direction ($yz$ planes) the interface-averaged values of the primitive variables are calculated according to equation (5.18). Then, the interface-averaged of the product of the primitive variables is, $\psi, \phi = (u, v, w)$,

$$[\phi \psi] = [\phi][\psi] + \frac{\partial \phi}{\partial z}|_M \frac{\partial \psi}{\partial z}|_M \frac{\Delta z^2}{12} + O(\Delta y)^2$$

with $M$ a point in the middle of the interface. The derivatives $\frac{\partial \phi}{\partial z}|_M$ and $\frac{\partial \psi}{\partial z}|_M$ are approximated by a classical second order finite-difference method.

In the spanwise direction ($xy$ planes) the same procedure is applied,

$$\frac{1}{4}[\phi]_{i,j,k-\frac{1}{2}} + [\phi]_{i,j,k+\frac{1}{2}} + \frac{1}{4}[\phi]_{i,j,k+\frac{1}{2}} = \frac{3}{4}(\bar{\phi}_{i,j,k+1} + \bar{\phi}_{i,j,k}) \quad (5.25)$$

the products may be written as,

$$[\phi \psi] = [\phi][\psi] + \frac{\partial \phi}{\partial x}|_M \frac{\partial \psi}{\partial x}|_M \frac{\Delta x^2}{12} + O(\Delta y)^2$$

with a second order approximation for $\frac{\partial \phi}{\partial x}|_M$ and $\frac{\partial \psi}{\partial x}|_M$. 

5.3. **FINITE VOLUME COMPACT SCHEME**

**Viscous Fluxes**

In the normal direction (xz planes) the following difference formulation is used to calculate the gradients of \( \phi = (u, v, w) \),

\[
\frac{\partial \phi}{\partial y}_{i,j+\frac{1}{2},k} = \frac{\overline{\phi}_{i,j+1,k} - \overline{\phi}_{i,j,k}}{\Delta y} \quad (5.26)
\]

with \( \Delta y \) the distance between the cell centers.

In the streamwise and spanwise directions a fourth order difference method is used. In the streamwise direction (x direction) it may be written,

\[
\left[ \frac{\partial \phi}{\partial x} \right]_{i+\frac{1}{2},j,k} = \frac{5}{4} \frac{\overline{\phi}_{i+1,j,k} - \overline{\phi}_{i,j,k}}{\Delta x} - \frac{1}{4} \frac{\overline{\phi}_{i+2,j,k} - \overline{\phi}_{i-1,j,k}}{3\Delta x}
\]

and similarly for the spanwise direction (z direction),

\[
\left[ \frac{\partial \phi}{\partial z} \right]_{i,j,k+\frac{1}{2}} = \frac{5}{4} \frac{\overline{\phi}_{i,j,k+1} - \overline{\phi}_{i,j,k}}{\Delta z} - \frac{1}{4} \frac{\overline{\phi}_{i,j,k+2} - \overline{\phi}_{i,j,k-1}}{3\Delta z}
\]

**Poisson Equation**

As explained in section (5.2) a Poisson equation for the pressure (equation (5.9)) should be solved to satisfy the continuity equation. In the present calculation a collocated grid is used, which means that the same control volume is used for the momentum and continuity equations. By integrating equation (5.9) over a volume \( V \) and using the Gauss theorem,

\[
\int u_{i}^{\text{cor}} n_{i} dS = \int \frac{\partial p}{\partial x_{i}} n_{i} dS \quad (5.27)
\]

where \( n_{i} \) is the normal unit vector of the faces. To evaluate the right hand side, the interface-averaged values of the pressure gradient \( [\frac{\partial p}{\partial x_{i}}] \) are needed.

The way to calculate these gradients, is as explained in section (5.3.2).

Concerning the boundary conditions for the pressure, periodic boundary conditions are imposed for the pressure in the homogeneous directions with a driving source term in the streamwise momentum equation. The gradient of pressure normal to the wall is set to zero.

**Channel Flow Calculation**

Before doing a simulation with particles, it was necessary to test the ability of the code in solving the carrier flow.

Therefore, large-eddy simulations were performed at a Reynolds number, based on friction velocity and channel half-width, of \( Re_{\tau} = \frac{u_{\tau}}{\nu} = 180 \). The
CHAPTER 5. PARTICLE LADEN FLOWS

Figure 5.3: Mean streamwise velocity of channel flow compared with DNS of Kim [52].

Figure 5.4: Turbulence intensities for channel flow compared with DNS of Kim [52].
5.3. FINITE VOLUME COMPACT SCHEME

Figure 5.5: One dimensional energy spectra, left: streamwise, right: spanwise, at $y^+ = 6.09$ compared with DNS of Kim et al. [52] at $y^+ = 5.39$.

Figure 5.6: One dimensional energy spectra, left: streamwise, right: spanwise, at $y^+ = 144.54$ compared with DNS of Kim et al. [52] at $y^+ = 149.53$. 
flow was resolved using $65 \times 65 \times 65$ grid points in the $x$, $y$ and $z$ directions. The streamwise, normal and spanwise dimensions are $4\pi\delta \times 2\delta \times \frac{4}{3}\pi\delta$, with $\delta$ the channel half-width. The grid spacing in wall coordinates in the $x$ and $z$ directions is $\Delta x^+ = \frac{\Delta_x u^+}{\nu} \simeq 35$ and $\Delta z^+ = \frac{\Delta_z u^+}{\nu} \simeq 12$. A non-uniform mesh with hyperbolic tangent distribution is used in the wall-normal direction. The first mesh point away from the wall is at $y^+ = \frac{\Delta_y u^+}{\tau} \simeq 0.49$ and the maximum spacing (at the centerline of the channel) is 13.8 wall units. A Smagorinsky model with damping near the wall is used for the subgrid scale modeling. The Smagorinsky constant is equal to $C = 0.007$, (see equation (1.7)). Fluid statistics were averaged for $6\delta/u_T$. The mean velocity profile and turbulence intensities are shown in figures (5.3), and (5.4). The results are compared with the DNS data of Kim et al. [52]. The energy spectra at two different locations of $y^+ = 6.09$ and $y^+ = 144.54$ compared with DNS of Kim et al. [52] are also shown in figures (5.5) and (5.6).

5.4 Calculation of the Particle Trajectories

The particle tracking routines can be added to any LES solver without major modification of the original routines. The particle tracking routines should be called at each time step immediately after the carrier flow velocities have been updated. The time step used for the particles must be equal to the time step of the carrier flow. The three components of the particles velocities as well as their $x,y$ and $z$ coordinates are updated at every iteration.

The particle equation of motion used in the simulations describes the motion of particles with densities substantially larger than that of the surrounding fluid and diameters small compared to the Kolmogorov scale. The total (drag+body) force $F_i$ of a single particle in a uniform flow field can be generally expressed by,

$$ F_i = C_D A_p \rho_f \frac{1}{2} |\mathbf{v} - \mathbf{u}| (u_i - v_i) + g \delta_{i1} (\rho_p V_p) $$

(5.28)

Where $A_p$ is the exposed frontal area of the particle to the direction of the incoming flow, and $V_p$ is the volume of the particle. For a spherical particle $A_p = \pi d^2$ and $V_p = \frac{4}{3} \pi \left(\frac{d}{2}\right)^3$, with $d$ the particle diameter. $v_i$ is the velocity of the particle and $u_i$ is the velocity of the fluid at the particle position. The body force $g \delta_{i1} (\rho_p V_p)$ acts along the positive streamwise direction corresponding to flow in a vertical channel flow. The fluid and particle densities are denoted $\rho_f$ and $\rho_p$, respectively. The particle acceleration $\frac{dv_i}{dt} = \frac{F_i}{\rho_p V_p}$ can finally be written as,

$$ \frac{dv_i}{dt} = \frac{\rho_f}{\rho_p} \frac{3}{4} \frac{C_D}{d} |\mathbf{v} - \mathbf{u}| (u_i - v_i) + g \delta_{i1} $$

(5.29)
Equation (5.29) may also be written in the following form,

\[
\frac{dv_i}{dt} = \frac{1}{\tau_p}(u_i - v_i) + g\delta_{1i} \tag{5.30}
\]

Where \( \frac{1}{\tau_p} = \frac{\rho_f}{\rho_p} \frac{2}{3} \frac{C_D}{d} |v - u| \) is the particle response or relaxation time. If the drag coefficient is defined with the Stokes’s drag law \( C_D = \frac{24}{Re_p} \), which is valid for small particle Reynolds number \( Re_p = |v - u|d/\nu \), the Stokes relaxation time \( \tau_p \) may be written as,

\[
\tau_p = \frac{\rho_p d^2}{18\nu\rho_f} \tag{5.31}
\]

Previous computations of particle-laden turbulent channel flow have shown that the particle Reynolds number \( Re_p \) does not necessarily remain small (e.g., see Rouson et al. [93]). Therefore, an empirical relation for \( C_D \) from Clift et al. [20] valid for particle Reynolds number up to about 40 was employed,

\[
C_D = \frac{24}{Re_p}(1 + 0.15Re_p^{0.687}) \tag{5.32}
\]

The particle Reynolds number of the present calculations never exceeds this maximum value. It should also be noted that (5.29) is appropriate for describing the motion of smooth rigid spheres and neglects the influence of virtual mass, buoyancy, and the Basset history force on particle motion. For particles with material densities large compared to the fluid these forces are negligible compared to the drag. The effect of lift force, while relevant to problems of particle deposition, is less significant to this work and therefore the effect of shear-induced lift in the equation of motion has been neglected. Finally, the volume fraction of particles is assumed small enough such that particle-particle interactions are negligible.

Once the velocities of the particles at the new time level are calculated with equation (5.29), their positions will be obtained by solving the following equation:

\[
\frac{dx_{i,p}}{dt} = v_i \quad i = 1, 2, 3 \tag{5.33}
\]

where \( x_{i,p} \) is the particle position. Equations (5.29) and (5.33) are integrated in time using a second-order Adams-Bashforth method. In general, for the equation,

\[
\frac{d\Phi}{dt} = Re_s(\Phi)
\]

the Adams-Bashforth method is,

\[
\Phi^{n+1} = \Phi^n + \Delta t(\frac{3}{2}Re_s(\Phi^n) - \frac{1}{2}Re_s(\Phi^{n-1}))
\]
Sometimes, it is more convenient to work with the nondimensionalized equations. Let the superscript ‘+’ denote variables nondimensionalized by inner scales, i.e., \( u^+ = u/u_\tau, \; x^+ = xu_\tau/\nu, \; t^+ = tu_\tau^2/\nu \) and \( g^+ = g\nu/u_\tau^3 \), the nondimensional forms of equations (5.29) and (5.33) may be written as,

\[
\frac{dv_i^+}{dt^+} = \frac{\rho_f 3 C_D}{\rho_p 4 d^+} |v^+ - u^+| (u_i^+ - v_i^+) + g^+ \delta_i_1 \quad i = 1, 2, 3
\]

\[
\frac{dx_{i,p}^+}{dt^+} = v_i^+ \quad i = 1, 2, 3
\]

with the Reynolds number defined as,

\[
Re_p = |v^+ - u^+| d^+
\]

If one wants to take the effect of particles on the carrier flow also into account, a two-way coupling method may be used.

According to equation (5.29), the force exerted on the particle of mass \( m_p \) is equal to,

\[
f_i = \frac{dv_i}{dt} m_p
\]

the same force is exerted on the fluid from the particle and the acceleration of the fluid \( S_i \) from this force is equal to,

\[
S_i = -\frac{f_i}{\rho V}
\]

with \( V \) the volume of the computational cell. If the cell contains \( N_{parc} \) particles the total force exerted on the fluid element is,

\[
F_i = \sum_{s=1}^{N_{parc}} (f_i)_s
\]

For the two-way coupling calculation the source term, \( \frac{F_i}{\rho V} \), should be added to the momentum equations to take into account the effects of the particles on the carrier flow. The momentum equations are then written as,

\[
\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{F_i}{\rho V} \quad (5.34)
\]

### 5.4.1 Interpolation

To calculate the drag force exerted on the particles, the fluid velocity at the particle locations is needed.

Three types of interpolations are explained, the first one is the simplest one and is the first order interpolation, then the second and third order interpolations are explained.
5.4. **CALCULATION OF THE PARTICLE TRAJECTORIES**

**First Order Interpolation**

For the first order interpolation if the particle is located in cell \(i\), and the cell center velocity of the carrier flow, or the cell-averaged velocity of the carrier flow in cell \(i\) is equal to \(U\), then the velocity at particle location is also supposed to be \(U\).

**Second Order Interpolation**

In a second order approximation, first, the values of the carrier flow velocity are supposed to be stored at the center of the cells, then a quadratic Lagrange interpolation\(^5\) is used to calculate the velocity of the fluid at the particle locations.

For a quadratic interpolation, in one dimension three cells are involved, in two dimensions nine cells and in three dimensions 27 cells. The Lagrange interpolation for a 1D problem is written as,

\[
\begin{align*}
\quad f(x) &= F_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + F_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + F_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} \\
&\quad (5.35)
\end{align*}
\]

Where \(F_1, F_2, \text{ and } F_3\) are the values of the function at the points \(x_1, x_2, \text{ and } x_3\), respectively. In this case \(x_1, x_2, x_3\) are the coordinates of the cell centers, and \(F_1, F_2, F_3\) are the cell center values. For three dimensional problems the above mentioned formula is used consecutively in three directions.

**Third Order Interpolation**

In the present code, a cell averaged approach is used for the higher (more than two) order spatial approximation. In a cell averaged approach, unlike the pointwise approach, the values in some specific points (the center of the cell for example) are not specified and the Lagrange formulation cannot be directly used.

To deal with this problem, suppose a 1D mesh with \(x_1, x_2, x_3, x_4\) the mesh coordinates and \(\bar{U}_1\) the cell-averaged value of the function in the interval \([x_1, x_2]\) and \(\bar{U}_2\) the cell-averaged value of the function in the interval \([x_2, x_3]\) and \(\bar{U}_3\) the cell-averaged value of the function in the interval \([x_3, x_4]\).

Suppose the variation of the function \(U(x)\) is approximated with a polynomial \(L(x)\) which is not known yet, i.e., \(U(x) \approx L(x)\). All we know is that three cells are involved and \(L(x)\) should be a second order polynomial. \(^5\)

\(^5\)We note that, if a quadratic interpolation is used, then the accuracy of the interpolation is third order. But for the current calculations where a cell-averaged approach is used for the carrier flow, the assumption that the flow velocities are stored at the center of the cells brings a second order error into simulations.
Eventhough the coefficients of the second order polynomial $L(x)$ are unknown, we know its integral in the interval $[x_1, x_2]$ is equal to the cell-averaged value of $U$ multiplied by the length of the cell. By defining a function $\phi(x) = \int_{x_1}^{x} L(x)dx$ we have,

$$\int_{x_1}^{x_2} L(x)dx = (x_2 - x_1)U_1 = \phi(x_2)$$

similarly integrating from $x_1$ to $x_3$ leads,

$$\int_{x_1}^{x_3} L(x)dx = \int_{x_1}^{x_2} L(x)dx + \int_{x_2}^{x_3} L(x)dx = (x_2 - x_1)U_1 + (x_3 - x_2)U_2 = \phi(x_3)$$

and in the same way after integrating from $x_1$ to $x_4$,

$$\int_{x_1}^{x_4} L(x)dx = (x_2 - x_1)U_1 + (x_3 - x_2)U_2 + (x_4 - x_3)U_3 = \phi(x_4)$$

Or in general one can write,

$$\int_{x_1}^{x} L(x)dx = \phi(x) \rightarrow \frac{d}{dx} \int_{x_1}^{x} L(x)dx = \phi'(x) \rightarrow L(x) = \phi'(x)$$

Now to calculate $L(x)$ one first has to calculate $\phi(x)$ and then the derivative of $\phi(x)$ will be equal to $L(x)$. With four points at $(x_1, \phi(x_1) = 0), (x_2, \phi(x_2)), (x_1, \phi(x_3))$ and $(x_1, \phi(x_4))$, the function $\phi(x)$ can be estimated with a third order Lagrange interpolation,

$$\phi(x) = 0 \times \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} + \phi(x_2) \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)} + \phi(x_3) \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)} + \phi(x_4) \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)}$$

Finally, $L(x)$ is calculated by taking the derivative of $\phi(x)$, and it is a second order polynomial.

$$L(x) = \phi'(x)$$
5.4.2 Numerical Details

Particle Velocity Statistics

To be able to draw the profiles of the particle velocities, the cell-averaged values of the particles are needed. The instantaneous cell-averaged velocity vector of the particles is,

$$\mathbf{\bar{u}}_p = \frac{\sum_{s=1}^{N_{parc}} \mathbf{u}_s}{N_{parc}}$$ (5.38)

where $\mathbf{\bar{u}}_p$ is the cell averaged velocity vector of the particles, $\mathbf{u}_s$ is the velocity vector of a single particle inside the cell, and $N_{parc}$ is the total number of particles in the cell.

Once $\mathbf{\bar{u}}_p$ is obtained the time or/and space averaging can be performed and, in the same way that the turbulence statistics of the carrier flow are calculated, the particle fluctuations can be calculated. The particle fluctuating velocity vector, $\mathbf{v}'_p$, can be defined as,

$$\mathbf{v}'_p = <\mathbf{\bar{u}}_p> - \mathbf{\bar{u}}_p$$ (5.39)

Where $<\cdots>$ denotes the time or/and space averaging. For example, if we denote the three components of the cell averaged velocity vector of the particles as, $\mathbf{\bar{u}}_p = (\mathbf{\bar{u}}_{1p}, \mathbf{\bar{u}}_{2p}, \mathbf{\bar{u}}_{3p})$ and the fluctuating part as $\mathbf{v}'_p = (v'_1p, v'_2p, v'_3p)$ then $<v'_{1p}v'_{2p}>$ is calculated as,

$$<v'_{1p}v'_{2p}> = <\mathbf{\bar{u}}_{1p}\mathbf{\bar{u}}_{2p}> - <\mathbf{\bar{u}}_{1p}><\mathbf{\bar{u}}_{2p}>$$ (5.40)

Memory Consideration

If the total number of particles is $N_{par}$ and the number of cells is $N_{cell}$, the extra memory needed for the particle tracking routines will be $15N_{par} + 13N_{cell}$.

The extra 15 arrays with the length of $N_{par}$ are as follows; three arrays for the three components of velocity, three arrays for the coordinates of the particles, three arrays for the indices of the cells where the particles are located, and six arrays to store the particles acceleration and velocity at time level $n-1$ used in the Adams-Bashforth method.

The extra 13 arrays with the length $N_{cell}$ are three arrays for the particles cell averaged velocities, one array for the number of particles in cells and nine arrays for the time averaged values of the three components of velocity and their products.
Boundary Conditions

If a particle crosses the boundary, its velocity and position should be modified according to the type of the boundary condition. There are two types of boundary conditions in the channel flow, the wall boundary condition and the periodic boundary condition.

- **Wall boundary condition:** if \((x_{p1}, y_{p1}, z_{p1})\) is the coordinates of the particle, \((u_{p1}, v_{p1}, w_{p1})\) the particle velocity, \((y_{min}, y_{max})\) the \(y\) coordinates of the wall and \(r_p\) the radius of the particle. The particle is assumed to hit the wall as soon as its center is one radius from the wall, i.e., \((y_{p1} < y_{min} + r_p \text{ or } y_{p1} > y_{max} - r_p)\).

The wall boundary condition for the velocity, assuming that the wall is a \(xz\) plane is,

\[
\begin{align*}
  u_{p2} &= u_{p1} \\
  v_{p2} &= -v_{p1} \\
  w_{p2} &= w_{p1}
\end{align*}
\]  

(5.41)

Where state 2 is when the boundary condition has been applied. From the above condition it is obvious that a fully elastic collision is assumed. For the coordinates of the particle,

\[
\begin{align*}
  x_{p2} &= x_{p1} \\
  y_{p2} &= 2(y_{max} - r_p) - y_{p1} \\
  z_{p2} &= z_{p1}
\end{align*}
\]  

(5.42)

Figure (5.7) is a schematic representation of an elastic collision from the lower wall.

- **Periodic boundary condition:** in the channel flow the periodicity is applied in the streamwise and spanwise directions. In the periodic
boundary condition the velocities of the particles are not changed and only the position is modified. For example suppose a particle has left the domain in the positive streamwise direction, i.e. \( x_{p1} > x_{\text{max}} \), then the periodicity is applied as follows,

\[
\begin{align*}
x_{p2} &= x_{p1} - (x_{\text{max}} - x_{\text{min}}) \\
y_{p2} &= y_{p1} \\
z_{p2} &= z_{p1}
\end{align*}
\] (5.43)

If the particle leaves the domain in the negative streamwise direction, i.e. \( x_{p1} < x_{\text{min}} \) then the boundary condition will be,

\[
\begin{align*}
x_{p2} &= x_{p1} + (x_{\text{max}} - x_{\text{min}}) \\
y_{p2} &= y_{p1} \\
z_{p2} &= z_{p1}
\end{align*}
\] (5.44)

A similar rule is valid in the \( z \) direction.

**Walk Search Algorithm**

To calculate the drag force, one needs to find the fluid velocity at the particle location, and it is necessary to know the indices of the cell in which the particle is located. A very simple algorithm called *walk search* \(^6\) can be used. In 1D suppose \( x_p \) is the coordinate of the particle and \( x_i \) is the mesh coordinates and \( x_p > x_i \), the walk search algorithm is,

- **Step 1:** if \((x_p > x_i)\) And \(x_p < x_{i+1}\) the particle is in the cell, otherwise go to the Step 2.
- **Step 2:** \(i = i + 1\) Go to Step 1.

If \( x_p < x_i \) the algorithm is similar but now one sweeps through the mesh in the opposite direction, i.e., the \( i \) index is decreased.

In 3D one has to apply sequentially the same algorithm in three directions. In the worst case the number of operations is \((N_i + N_j + N_k)\), with \( N_i, N_j \) and \( N_k \) the number of cells in \( i, j \) and \( k \) directions. Once the indices of a particle are obtained, they are saved and at the next time step, searching starts from the previous location of the particle to find its new location, which requires only a few operations.

\(^6\)It is interesting to mention that walk search algorithm is also used in the context of unstructured grid generation, Mavriplis [76], as a way to find the cell where a newly generated point is located.
For a curvilinear mesh, first an arbitrary cell is chosen, then the dot products of the outward normal unit vectors of the cell faces with the vectors connecting the particle position to the cell face centers are calculated. Depending on the values of these dot products, the next appropriate neighboring cell is chosen. The process continues until all dot products are negative, or in other words when the point is located inside the cell.

For the time being, we are dealing with a Cartesian mesh, and the walk search algorithm can be used in its simplified form.

5.5 Particle-Laden Channel Flow

Once a time-averaged steady state solution has been obtained for the Eulerian velocity field, i.e. the carrier flow, the particles are assigned random locations throughout the channel. This can be done in two ways. First, one may consider the whole volume of the channel and distribute the particles randomly. With this approach some near wall cells, which have a very small thickness, may not receive any particle. A second way, to have a more uniform distribution and make sure that the cells receive about the same number of particles, is to go through the cells one by one and randomly put a particle inside each cell. With this approach, the initial volume fraction of the particles will be higher close to the walls where the mesh cells are clustered. The initial particle velocity was assumed to be the same as the fluid velocity at the particle locations.

Generally, at one time step, the carrier flow calculation is more time consuming than the calculation of particle trajectories. Therefore, another way to obtain a closer initial solution to the fully developed results is to calculate the particle trajectories in a “frozen” turbulent field, and use the output of this calculation as an initial solution for a realistic simulation.

Starting with this initial solution, the flow and particles are advanced in time simultaneously until a time-averaged steady state is reached for the particles. The development time, i.e., the time required for particles to become independent of their initial conditions, depends on the particle response time. In the calculation of Wang et al. [116] the development times of the particles range from $0.5\delta/u_r$ for the Lycopodium particles which are the lightest particles, to $6\delta/u_r$ for the copper which are the heaviest.

In the present calculations, the development time is set to $6\delta/u_r$ for all particles. After an equilibrium condition had been reached, particle statistics were accumulated for another $6\delta/u_r$.

\footnote{At least for an incompressible calculation where at each stage the Poisson equation should be calculated.}
Table 5.1: Parameters of Lycopodium particles

<table>
<thead>
<tr>
<th>$\tau_{p/\delta}$</th>
<th>$\tau_{p/\delta}$</th>
<th>$\tau_{p/\delta}$</th>
<th>$\rho_p/\rho_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4.73 \times 10^{-2}$</td>
<td>8.53</td>
<td>$1.4 \times 10^{-3}$</td>
<td>0.2</td>
</tr>
</tbody>
</table>

5.5.1 Lycopodium Particles

The particle response time $\tau_p = \frac{\rho_p d^2}{18 \mu \rho_f}$, the radius of the particle $r_p$, and the particle to fluid density ratio $\frac{\rho_p}{\rho_f}$ are shown in table 5.1. The values are expressed in terms of both channel variables ($\delta$ and $u_\tau$) and in wall units. These data are exactly the same as the data of Wang et al. [116] with which the present results are compared.

In the present calculation the channel half width $\delta = 0.01m$, and the carrier fluid is air with a density of $\rho_f = 1.16 Kg/m^3$.

The Effect of the Order of Interpolation

Fukagata et al. [31] studied the effect of the order of interpolation on the rms particle velocity fluctuations of glass and copper particles. They examined three different types of interpolation such as the Nearest Grid Point (first order), linear interpolation (second order) and a sixth order Lagrange interpolation.

Wang et al. [115] found, that linear interpolation causes large errors in rms fluid velocity fluctuations at the particle locations. Fukagata et al. [31] showed, however, that these errors do not have a harmful effect on the calculation of particle velocity fluctuations.

The results presented by Fukagata et al. [31] for the mean velocity and turbulence intensities of glass and copper particles show that for heavy particles, the interpolation errors have negligible effect on the time averaged statistics. They argue that this might be due to the large Stokes relaxation time of glass and copper particles, which implies that the particles are less sensitive to the small fluctuations in fluid velocity which may be smoothed out by a low order interpolation scheme.

In order to check the argument of Fukagata et al. [31], the effect of interpolation on Lycopodium particles, which are lighter than glass and copper particles, is examined in the present study. The density of Lycopodium is $700 kg/m^3$ compared to 2500 and 8500 for glass and copper, respectively.

---

8The Nearest Grid Point method is the one which we explained previously, see section (5.4.1), and called first order interpolation.
A channel flow at $Re_\tau = 180$ with a rather coarse mesh of $33 \times 65 \times 33$ points in the $x$, $y$ and $z$ directions is considered. Uniform meshes with spacing $\Delta x^+ = \frac{\Delta x u^+}{\nu} \simeq 70.6$ and $\Delta z^+ = \frac{\Delta z u^+}{\nu} \simeq 23.6$ are used in the streamwise and spanwise directions. A non uniform mesh with hyperbolic tangent distribution (see equation (4.1)) is used in the wall-normal direction. The first mesh point away from the wall is at $y^+ = \frac{\Delta y u^+}{\nu} \simeq 0.49$ and the maximum spacing (at the centerline of the channel) is 13.7 wall units. The development time and averaging time for particle statistics are both equal to $6\delta/u_\tau$ seconds.

Figures (5.8) and (5.9) show the mean velocity profiles and the turbulence intensities of particles for different types of interpolation. From figure (5.9) one can easily see that the particle turbulence intensities have a higher magnitude when a higher order of interpolation is used. This may be explained in the way, that by increasing the degree of interpolation, the small scale fluctuations are captured more precisely, which results in a better representation of the particle fluctuations. Also shown in the figures, are the fluid turbulence intensities. One can see from the figures that the streamwise values of the particle turbulence intensities are higher than that of the fluid, but the spanwise and normal values are smaller. The same behavior was observed in the simulation of Wang et al. [116]. We will discuss more about the particle velocity fluctuations in the coming sections.

The shape of the mean velocity profile of the particles, figure (5.8), also changes slightly. From the same figure one can also notice that by increasing the degree of interpolation the magnitude of particle velocities slightly decreases.

Even though the effect of interpolation is not very significant, and for heavier particles like glass particles it may even be negligible, to be on the safe side, a third order interpolation is always used for the rest of the calculations.

**Fine Mesh Calculation**

For the final simulation of Lycopodium a finer mesh with more points in the spanwise direction is used. The new grid parameters in wall units are, $\Delta x^+ = 70.6$, $\Delta z^+ = 11.8$, $\Delta y^+_{\text{min}} = 0.49$ and $\Delta y^+_{\text{max}} = 13.7$, and the number of points in the $x$, $y$, and $z$ directions is $33 \times 65 \times 65$.

The results are shown in figures (5.10) to (5.13). The mean velocity profile of particles (figure (5.10)) shows that the particles, apart from a small region around the buffer layer $y^+ \simeq 30$, closely track the fluid flow. The same behavior was observed in the LES of Wang et al. [116]. In the direct numerical simulations of Rouson et al. [93] (results not shown), the mean velocity profile of particles, slightly lags that of the fluid in the near wall region ($y^+ < 10$). Rouson et al. [93] attribute the lag in the Lycopodium
5.5. PARTICLE-LADEN CHANNEL FLOW

Figure 5.8: Effect of the order of interpolation on the mean streamwise velocity of the particles.

Figure 5.9: Effect of the order of interpolation on the rms velocity fluctuations of the particles.
profile to preferential concentration of Lycopodium particles in the low-speed streaks. The discrepancy between LES and DNS results may be related to the fact that the near-wall structures are less well resolved in the LES and, consequently, preferential concentration of particles in low-speed regions is less significant than in the DNS, resulting in an over-prediction of the mean velocity of the Lycopodium particles.

Comparisons of rms particle velocity fluctuations from the present calculations with both the LES results of Wang et al. [116] and the DNS data of Rouson et al. [93] are shown in figures (5.11), (5.12) and (5.13).

Apart from some small differences in the streamwise particle velocity fluctuations, the general agreement between the present LES and the LES of Wang et al. [116] is quite satisfactory. In figure (5.11) the predicted particle streamwise turbulence intensities (bold line) are somewhat higher than the one of Wang et al. (crosses). The reason can be the different type of the spatial discretization and/or the subgrid-scale model that they use in their carrier flow calculation. Wang et al. used a finer mesh with $65 \times 65 \times 65$ grid points, and for the subgrid scale modeling, they used the Lagrangian dynamic approach of Meneveau et al. [77].

As mentioned before, one can see from the figures that the streamwise values of the particle turbulence intensities are higher than that of the fluid, but the spanwise and normal values are smaller.

It is difficult to speculate on the causes of the difference between LES predictions and DNS results. Wang et al. [116] investigated the errors caused by neglecting the particle transport by subgrid-scale (SGS) velocities. It is well known that in LES the smallest scales of motion are not resolved. Thus, only the large-scale velocity field is directly available in an LES computation for determining particle motion. Wang et al. [116] investigated the effect of SGS velocity field on particle transport by adding SGS fluctuations to the fluid velocity used in the particle equation of motion (5.29). The magnitude of the SGS fluctuations $u''_i$ was determined by solving a transport equation\(^9\) for SGS kinetic energy, $q^2 = u''^2_i/2$ as proposed by Schuman [99]. The component fluctuations $u''_i$ were then scaled by a random number sampled from a Gaussian distribution and added to $\bar{u}_i$ at the particle locations. The component fluctuations $u''_i$ were then scaled by a random number sampled from a Gaussian distribution and added to $\bar{u}_i$ at the particle locations. The component fluctuations $u''_i$ were then scaled by a random number sampled from a Gaussian distribution and added to $\bar{u}_i$ at the particle locations. The component fluctuations $u''_i$ were then scaled by a random number sampled from a Gaussian distribution and added to $\bar{u}_i$ at the particle locations.

\[^9\]The transport equation can be written as,

$$\frac{\partial q^2}{\partial t} + \bar{u}_j \frac{\partial q^2}{\partial x_j} = 2 \nu \frac{|\bar{S}|^2}{\Delta} + \frac{\partial}{\partial x_j} \left( \frac{1}{3} \sqrt{q^2} \frac{\partial q^2}{\partial x_j} \right) + \nu \frac{\partial^2 q^2}{\partial x_j \partial x_j} - \sqrt{1/2} \epsilon \frac{q^3}{l_\Delta}$$

where

$$c_\epsilon = \pi \left( \frac{2}{3 k_0} \right)^{3/2}, \quad l_\Delta = \text{min}(\bar{\Delta}, c_\epsilon y)$$

with the Kolmogorov constant $k_0 = 1.6$. \[\]
Table 5.2: Parameters of glass particles

<table>
<thead>
<tr>
<th>$\frac{\nu}{\delta/u_r}$</th>
<th>$\tau_p^+$</th>
<th>$\tau_p^+$</th>
<th>$\rho_p^+$</th>
<th>$\rho_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$54 \times 10^{-2}$</td>
<td>97.2</td>
<td>$2.5 \times 10^{-3}$</td>
<td>0.45</td>
<td>2155</td>
</tr>
</tbody>
</table>

complete velocity, i.e., $\bar{u}_i + u''_i$, was subsequently used in (5.29) to determine the particle velocity.

Their findings showed that for the 28 $\mu$m Lycopodium, there is a negligible effect of the SGS fluid velocity on particle fluctuations and the change in the particle velocities due to the addition of the SGS fluid velocity is less than 1%. This conclusion is also valid for larger or heavier particles, as the inertia makes the particle response less sensitive to the small-scale, fluctuating, velocity field. For particles with material densities large compared to the carrier flow the response of particles to the frequency spectrum of turbulence can be shown to be proportional to $1/(\tau_p w)$ ($w$ is the frequency). Thus, for increasing values of the relaxation time $\tau_p$ and/or frequency the filtering of high frequency motions by particle inertia is significant.

A possible explanation for the discrepancies between the LES and DNS results may be suggested by considering the coherent structures near the wall. From the experiments of Kline et al. [53], it has been well known that streaks of respectively low and high longitudinal velocity exist close to the wall, between $y^+ = 5$ and $y^+ = 40 \sim 50$. Several numerical and experimental studies have shown that the fluid elements located above the low-speed streaks are ejected away from the wall. Conversely, the high speed fluid elements correspond to the sweep events, i.e., pumping of the high speed flow of the outer region to the near wall region.

There are still a lot of controversies about the interpretation of the low- and high-speed streaks, and how they are related to the sweep and ejection events. But what is obvious, is the important role of these structures on the transport of parcels of fluid as well as the particles. In LES the inadequacy of the computational grid resolution to resolve the streaks at their proper scale, may be a reason for the difference between the LES and DNS results. Furthermore, this problem could be more serious for the particles than for the fluid elements, due to their higher inertia and density compared to the fluid which helps them to keep better their inertia when they are transported.
CHAPTER 5. PARTICLE LADEN FLOWS

Figure 5.10: Mean streamwise velocity of Lycopodium particles in turbulent channel flow.

Figure 5.11: Root-mean-square velocity fluctuations \((u'u')^{1/2}\) of Lycopodium particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].
5.5. PARTICLE-LADEN CHANNEL FLOW

Figure 5.12: Root-mean-square velocity fluctuations \((u'v')^{1/2}\) of Lycopodium particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].

Figure 5.13: Root-mean-square velocity fluctuations \((w'w')^{1/2}\) of Lycopodium particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].
5.5.2 Glass Particles

The results for glass particles, which have a bigger diameter compared to the Lycopodium, are presented in figures (5.14) to (5.17). The parameters of glass particles are shown in table (5.2). They also have a higher relaxation time, Stokes number, and density than the Lycopodium particles.

In figure (5.14) the particle mean velocity profile is compared with the DNS of Rouson et al. [93], and LES of Wang et al. [116]. LES predictions of the mean velocity are in good agreement with the DNS and the other LES calculations. Unlike the Lycopodium particles, that closely track the fluid, the velocity of glass particles near the wall is higher than that of the fluid. This is a result of the transport of high velocity particles from the outer region to the near-wall region. The glass particles have a higher inertia than Lycopodium particles and they can memorize better their velocities.

Particle turbulence intensities are shown in figures (5.15), (5.16), and (5.17). We note that the streamwise rms velocities are again smaller than the DNS values while the wall normal and spanwise rms velocities in the LES are slightly greater than the corresponding values in the DNS.

When compared with the Lycopodium, it is clear from the figures that near the wall, the streamwise fluctuation levels increase for a higher value of Stokes number while the wall normal and spanwise fluctuations are reduced. This anisotropy can be measured by defining the following parameter,

\[
\beta_i = \frac{<f_i^2>}{<f_1f_1> + <f_2f_2> + <f_3f_3>} \quad \text{(no sum)}
\]

where \( f_i \) is either the fluid or particle velocity fluctuations. If the fluctuations are isotropic, \( \beta_i \)'s are equal to 1/3 and the deviation from this value is a measure of the anisotropy in fluctuations.

Shown in figures (5.21), (5.22), and (5.23) is a comparison of the anisotropy for Lycopodium, glass, and copper particles with the carrier flow. It is evident that the anisotropy of the particle fluctuations is larger than the fluid, and this anisotropy increases with the particle Stokes number. Copper particles transport is discussed in the next section.

5.5.3 Copper Particles

The parameters of copper particles, which have a bigger diameter and density than the glass particles, are given in table (5.3). The results are shown in figures (5.18) to (5.20). For the particle mean velocity, there is a discrepancy between the present calculation and the LES of Wang et al. [116]. The number of grid points and the grid spacing in wall coordinates are the same as the one of Wang et al. [116]. However, Wang et al. [116] used a different
5.5. PARTICLE-LADEN CHANNEL FLOW

Figure 5.14: Mean streamwise velocity of glass particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].

Figure 5.15: Root-mean-square velocity fluctuations \( (u'u')^{1/2} \) of glass particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].

Table 5.3: Parameters of copper particles

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{\tau_p}{\delta/u_c} )</td>
<td>4.50</td>
</tr>
<tr>
<td>( \tau_p^+ )</td>
<td>811.40</td>
</tr>
<tr>
<td>( \frac{\rho_p}{\rho} )</td>
<td>( 3.94 \times 10^{-3} )</td>
</tr>
<tr>
<td>( \rho_p^+ )</td>
<td>0.71</td>
</tr>
<tr>
<td>( \rho_f^+ )</td>
<td>7241.4</td>
</tr>
</tbody>
</table>
Figure 5.16: Root-mean-square velocity fluctuations \((v'v')^{1/2}\) of glass particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].

Figure 5.17: Root-mean-square velocity fluctuations \((w'w')^{1/2}\) of glass particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].
5.5. PARTICLE-LADEN CHANNEL FLOW

Figure 5.18: Mean streamwise velocity of copper particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].

type of subgrid-scale modeling and spatial discretization. Note also that all calculations exhibit a slight plateau near the wall. The plateau may result from the transport of high velocity particles from the outer region of the channel to the near-wall region and this feature appears to be somewhat more pronounced in the DNS.

Comparisons of \( \text{rms} \) particle velocity fluctuations from the present calculation to both DNS and LES results are shown in figures (5.19) and (5.20). As may be observed there is a general good agreement between the present predictions and the DNS and LES.
Figure 5.19: Root-mean-square velocity fluctuations ($u'u'$)$^{1/2}$ of copper particles in turbulent channel flow compared with DNS of Rouson et al. [93] and LES of Wang et al. [116].

Figure 5.20: Root-mean-square velocity fluctuations ($v'v'$)$^{1/2}$ of copper particles in turbulent channel flow compared with LES of Wang et al. [116].
5.5. PARTICLE-LADEN CHANNEL FLOW

Figure 5.21: Anisotropy comparison of the fluid and the Lycopodium particles.

Figure 5.22: Anisotropy comparison of the fluid and the glass particles.
Figure 5.23: Anisotropy comparison of the fluid and the copper particles.
Chapter 6

Conclusion and Future Work

6.1 Compressible Low Mach Number Flows

The numerical problems of large-eddy simulation of compressible low Mach number flows have been studied, Lessani et al.[68], [69]. The main numerical aspects may be itemized as follows:

- For stability reasons, as explained in section (2.1.2), a semi-implicit Runge-Kutta method should be used during the inner iterations in pseudo-time. A way to simplify the matrix inversion, in the case of the Navier-Stokes equations, has been explained in section (2.5.3).

- The advantage of the trapezoidal method over the commonly used second-order backward differencing for the time derivative discretization has been shown. The trapezoidal method allows the use of a bigger time-step compared to the second-order backward differencing method, section (3.4).

- The necessity of using preconditioning along with multigrid in order to have an effective damping ability has been demonstrated, section (3.1). In the absence of preconditioning, as a result of the high frequency errors which cannot be damped properly on fine mesh, the multigrid becomes quite ineffective.

- In order to prevent the laminarization of the flow on one hand, and to have a good damping property\(^1\) on the other hand, the third order Jameson type artificial dissipation is only added to the continuity equation and not to the momentum and energy equations. Numerical experience has shown that this suffices to guarantee a good overall convergence in the inner iterations.

\(^1\)Specially for the continuity equation where there is no physical dissipation.
• To have a faster calculation, the viscous fluxes are only calculated on the finest grid. This has another advantage, that is in agreement with the basic assumption of LES which needs rather fine mesh.

• V-sawtooth multigrid was the most robust for the present calculations. The analytical reason is not clear.

• The effect of the residual smoothing was not clear. For the channel flow it did not bring any improvement, but for the cavity flow, helped to reduce the number of inner iterations.

• Time correction technique was not a big help to reduce the number of inner iterations. However, for some particular number of inner iterations, using the time correction technique leads to better results, section (2.6).

• A fully implicit approach is used and compared with a purely explicit one. It was shown that the implicit approach, depending on the test case considered, is 4 to 7 times faster than a purely explicit method.

One may argue against comparing the efficiency of a fully implicit approach with a purely explicit one, and ask for a comparison between different implicit approaches, or at least semi-implicit approaches. A common semi-implicit approach, which is especially used for incompressible calculations, is to discretize the viscous part implicitly\(^2\), and the non-linear convective part explicitly. An approximate factorization method like Alternating Direction Implicit (ADI) is generally used to deal with the implicit part. The error of the factorization is of the same order as the error of the time discretization scheme and is second order in time. This method is quite useful in the framework of incompressible flows, but it does not bring any improvement for the compressible low Mach number flows where the time step is mainly restricted by the convective CFL condition based on the biggest eigenvalue of the system.

In a fully implicit approach where both convective and viscous terms are calculated implicitly, one first has to linearize the non-linear convective terms, and then an implicit solver can be used to march in time. The linearization brings another second order error into equations.

If the equations are discretized with a finite difference method and the time accuracy is not the main issue, the linearization and factorization errors approach to zero in the steady state. But if time accuracy is important, one cannot easily neglect these errors, even though they are always second

\(^2\)It is almost always done with a trapezoidal method, which is also called Crank-Nicolson when applied to the diffusion term.
6.2. PARTICLE-LADEN FLOWS

order in time. To solve this problem, for the unsteady calculations one has to always add a pseudo-time to reach the steady state in pseudo-time and minimize the above mentioned errors.

Unfortunately, the use of an ADI solver in the context of a finite volume method is not straightforward. In a finite volume method the fluxes are directly calculated and the method is supposed to be applicable to any kind of complex mesh including structured curvilinear or even unstructured grid. However, by incorporating some of the features of the finite difference method into the finite volume approach such as the coordinate transformation which in the finite volume context should be applied to the fluxes, the use of the ADI in combination with the finite volume method could be again possible.

Recently, Hsu and Jameson [44] used a hybrid scheme to calculate complex unsteady compressible flows with a finite volume method. Their approach is a two step method which combines ADI and dual time stepping. At the first step, the linearized implicit equations are solved with an ADI method which provides a formal second order accuracy in time. At the second step, a pseudo-time is introduced and with a number of inner iterations the errors due to both linearization and factorization are reduced. They tested a two dimensional pitching airfoil and showed that this hybrid approach is 40 percent cheaper than a pure dual time stepping scheme.

The approach of Hsu and Jameson seems to be an interesting approach and it is worthwhile to be further investigated in the future in an LES context.

The second interesting method, which has not been studied in this thesis, is the use of an implicit approach in pseudo-time and to march in time with an ADI method. The implicit treatment brings in the complexity of coordinate transformation which is necessary for an ADI method, but the overall speed up may compensate for the extra computational cost.

6.2 Particle-Laden Flows

Large-eddy simulation of particle-laden channel flow was carried out at $Re_{\tau} = 180$. An incompressible finite volume solver, based on a cell-averaged approach, was used for the carrier flow calculations. The methodology of calculating the space derivatives with a finite volume compact scheme was briefly described.

The velocity and location of the particles were calculated using a second order Adams-Bashforth formula. The drag force of the particles is calculated assuming the particles are smooth rigid spheres and the influence of virtual mass, buoyancy, shear-induced lift, and Basset history force were neglected on particles motion. The empirical relation used for drag coefficient $C_D$ is valid for particle Reynolds number up to 40 and during the calculation, the
Different interpolation methods such as first, second and third order interpolations were tested. In the final calculations, a third order Lagrange interpolation was used.

The Lycopodium, glass, and copper particles were considered. A one-way coupling method, which neglects the effect of particles on the carrier flow, is used. The agreement between the present calculations and the existing DNS and LES data in the literature is satisfactory. The anisotropy of particle fluctuations were compared. It was demonstrated that the anisotropy of the particle fluctuations increases with the Stokes number.

Two-way coupling of carrier flow and particles will be considered in the future. For the incompressible flow, after the addition of particle source terms in the momentum equations, special care should be taken in solving the Poisson equation to keep the stability and efficiency of the code.
Bibliography


BIBLIOGRAPHY


