

DESIGN AND APPLICATION OF DISCRETE EXPLICIT  
FILTERS FOR LARGE EDDY SIMULATION OF  
COMPRESSIBLE TURBULENT FLOWS

by

Willem Deconinck

A thesis submitted in conformity with the requirements  
for the degree of Masters of Applied Science  
Graduate Department of Aerospace Engineering  
University of Toronto

Copyright © 2008 by Willem Deconinck

## **Abstract**

# Design and Application of Discrete Explicit Filters for Large Eddy Simulation of Compressible Turbulent Flows

Willem Deconinck

Masters of Applied Science

Graduate Department of Aerospace Engineering

University of Toronto

2008

In the context of Large Eddy Simulation (LES) of turbulent flows, there is a current need to compare and evaluate different proposed subfilter-scale models. In order to carefully compare subfilter-scale models and compare LES predictions to Direct Numerical Simulation (DNS) results (the latter would be helpful in the comparison and validation of models), there is a real need for a “grid-independent” LES capability and explicit filtering methods offer one means by which this may be achieved. Advantages of explicit filtering are that it provides a means for eliminating aliasing errors, allows for the direct control of commutation errors, and most importantly allows a decoupling between the mesh spacing and the filter width which is the primary reason why there are difficulties in comparing LES solutions obtained on different grids. This thesis considers the design and assessment of discrete explicit filters and their application to isotropic turbulence prediction.

## Acknowledgements

The present work was performed at the University of Toronto Institute for Aerospace Studies. The author gratefully acknowledges the financial support from the University of Toronto.

I wish to thank the head of the CFD and Propulsion Group at UTIAS, Prof. Dr. Clinton P. T. Groth for the opportunity to perform this Masters thesis under his supervision. I'm grateful for the proposed subject, which turned out to be very interesting, and for the perfect guidance offered into this research area.

I would like to thank my colleagues from the CFD-lab, which have become good friends. Thanks Francisco for the patient hours teaching and explaining the details of turbulence.

Finally, my main gratitude is to my family, and to my girlfriend Cristiana and her family. Without my family's support and funding, this Masters research abroad would not have taken place. Cristiana, thank you for the moral support you gave me when I came home after a less successful day at work and for the way you were able to give me the necessary courage to continue this work until the end. Many thanks to Cristiana's family for welcoming me and giving me a good atmosphere at home.

*Toronto, August 2008*

*Willem Deconinck*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Motivation . . . . .	2
1.3	Objectives . . . . .	2
1.4	Overview of Thesis . . . . .	3
<b>2</b>	<b>Large Eddy Simulation of Turbulent Flows</b>	<b>4</b>
2.1	Turbulent Flow . . . . .	4
2.1.1	Description . . . . .	4
2.1.2	The Energy Cascade . . . . .	5
2.1.3	Spectral representation . . . . .	7
2.2	Numerical Solution of Turbulent Flows . . . . .	8
2.2.1	Direct Numerical Simulation . . . . .	9
2.2.2	Reynolds Averaged Navier-Stokes . . . . .	9
2.2.3	Large Eddy Simulation . . . . .	10
2.3	Large Eddy Simulation Techniques . . . . .	11
2.3.1	Favre-Filtered Navier-Stokes Equations . . . . .	11
2.3.2	Subfilter-Scale Modelling . . . . .	14
2.3.3	Closed Favre-Filtered Navier-Stokes Equations . . . . .	17

2.3.4	Implicit versus Explicit Filtering . . . . .	18
2.3.5	Aliasing Errors . . . . .	18
2.3.6	Commutation Errors . . . . .	19
<b>3</b>	<b>Spatial Filtering Techniques</b>	<b>20</b>
3.1	Implicit Spatial Filtering . . . . .	20
3.1.1	Smagorinsky Filter . . . . .	22
3.2	Explicit Spatial Filtering . . . . .	24
3.2.1	Properties . . . . .	24
3.2.2	Commutation Error in One Dimension . . . . .	25
3.2.3	Approaches to Explicit Spatial Filtering . . . . .	27
3.3	Explicit Analytical Filters . . . . .	29
3.3.1	Three Classical Analytical Filters . . . . .	29
3.3.2	A Higher Order Commuting Analytical Filter . . . . .	32
3.4	Explicit Discrete Filters . . . . .	33
3.4.1	Linear Constraints Filter . . . . .	36
3.4.2	Least Squares Reconstruction Filter . . . . .	38
3.4.3	Polynomial Interpolation Filter . . . . .	40
3.4.4	Discrete Approximation of Convolution Filters . . . . .	40
3.5	Approximate Deconvolution Approach . . . . .	42
<b>4</b>	<b>Design and Assessment of Discrete Explicit Filters</b>	<b>43</b>
4.1	Linear Constraints Filter . . . . .	43
4.1.1	Design . . . . .	43
4.1.2	Assessment . . . . .	51
4.2	Least-Squares Reconstruction Filter . . . . .	55

4.2.1	Design . . . . .	55
4.2.2	Assessment . . . . .	63
<b>5</b>	<b>Parallel Finite-Volume Scheme</b>	<b>66</b>
5.1	Favre-Filtered Navier-Stokes Equations in Matrix-Vector Notation . . . . .	66
5.2	Godunov Finite-Volume Method . . . . .	67
5.2.1	Finite-Volume Formulation of Governing Equations . . . . .	67
5.2.2	Inviscid Flux Evaluation . . . . .	68
5.2.3	Viscous Flux Evaluation . . . . .	69
5.2.4	Time Integration Scheme . . . . .	70
5.3	Multi-Block Structured Mesh . . . . .	70
<b>6</b>	<b>Implementation and Application of Discrete Explicit Filters</b>	<b>72</b>
6.1	Implementation of Discrete Explicit Filters . . . . .	72
6.1.1	Computational Framework for Discrete Explicit Filtering . . . . .	72
6.1.2	Analysis of Approaches to Explicit Filters . . . . .	74
6.2	Application of Discrete Explicit Filtering to Homogeneous Isotropic Turbulence .	76
6.2.1	Synthetic Turbulent Flow Field . . . . .	76
6.2.2	Application of Discrete Explicit Filter to Model Spectrum . . . . .	77
6.2.3	Application to Isotropic Turbulence Decay . . . . .	78
<b>7</b>	<b>Conclusions and Future Research</b>	<b>86</b>
7.1	Concluding Remarks . . . . .	86
7.2	Contributions . . . . .	87
7.3	Future Research . . . . .	87
<b>References</b>		<b>92</b>

# List of Tables

3.1	Filter moments for the top-hat filter and the Gaussian filter up to the 8 <sup>th</sup> order.	32
4.1	Filter weights for basic constraints . . . . .	46
4.2	Filter weights for the fourth-order linear constraints filter using additional constraints on derivatives of $\hat{G}(\kappa_\Delta)$ , with FGR = 2 (DC = number of derivative constraints). . . . .	47
4.3	Filter weights for the fourth-order linear constraints filter using additional for least-squares constraints, with target FGR = 2. (LS = number of least-squares constraints) . . . . .	49
4.4	Number of unknowns using the least-squares reconstruction filter for different degrees in one, two, and three dimensions. . . . .	56
6.1	Specifics of the Model Spectrum . . . . .	77

# List of Figures

2.1	A large variation of length scales in the experimental turbulent mixing layer between helium (upper) and nitrogen (lower), (Brown and Roshko [5]) . . . . .	5
2.2	Visualization of turbulent eddies in an isotropic turbulent flow . . . . .	6
2.3	Schematic of the energy cascade at high Reynolds number (logarithmic scale). . . . .	8
2.4	Energy-spectrum function for isotropic turbulence (logarithmic scales). . . . .	9
2.5	Velocity fluctuations as a function of time . . . . .	10
2.6	The resolved and subfilter range in the isotropic turbulence spectrum. $\kappa_{\bar{\Delta}}$ represents the filter cut-off wavenumber. . . . .	11
2.7	Schematic of the resolved Reynolds stress $\mathcal{L}_{ij}$ used in the dynamic Smagorinsky model. . . . .	16
3.1	Modified wavenumber for various schemes . . . . .	21
3.2	Transfer function of the Smagorinsky filter . . . . .	23
3.3	Smagorinsky filter applied to the kinetic energyspectrum function . . . . .	23
3.4	Sharp Fourier cut-off filter. Convolution kernel $G$ in physical space. . . . .	30
3.5	Top-hat filter. Convolution kernel $G$ in physical space normalized by $\bar{\Delta}$ . . . . .	31
3.6	Gaussian filter. Convolution kernel $G$ in physical space normalized by $\bar{\Delta}$ . . . . .	31
3.7	Associated transfer functions to the Fourier cut-off, top-hat and Gaussian filters. .	31
3.8	Associated transfer functions to the Van der Ven filters with $m = 1, \dots, 4$ . . . . .	33
3.9	Riemann summation of the cell centred values is an exact integration-scheme for a piece-wise linear finite volume scheme. . . . .	34

4.1	Filter transfer functions of the linear constraints filter using no additional constraints, for different commutation orders. . . . .	46
4.2	Filter transfer functions of the fourth-order linear constraints filter using additional for constraints on derivatives of $\hat{G}(\kappa_{\Delta})$ , with FGR = 2. (DC = number of derivative constraints) . . . . .	48
4.3	Filter transfer functions of fourth-order linear constraints filter using additional for least-squares constraints, with target FGR = 2. (LS = number of least-squares constraints) . . . . .	49
4.4	Comparison of the filter transfer functions from Figures 4.2 and 4.3. . . . .	50
4.5	Three-dimensional filter transfer function of fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2. The iso-surface shown denotes $\hat{G}(\kappa_{\bar{\Delta}})$ . . . . .	51
4.6	Diagonals of a three-dimensional filter transfer function of fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2. . . . .	52
4.7	Three-dimensional spectrum of the application to a uniform spectrum of a fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2. The iso-surface shown represents $\hat{G}(\kappa_{\bar{\Delta}})$ . . . . .	53
4.8	Diagonals of a three-dimensional spectrum of the application to a uniform spectrum of a fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2. . . . .	53
4.9	Demonstration of commutation of filters of order 2, 4, 6, 8. A finite differencing scheme of order 10 is used. Shown is the L2-norm of the commutation error, and the L2-norm of the truncation error. . . . .	54
4.10	Effect of the relaxation factor $w_0$ on the diagonals of a third-order least-squares reconstruction filter transfer function obtained using three rings of solution values.	57
4.11	Effect of the different weighting functions on the diagonals of a least-squares reconstruction filter transfer function of third degree with three rings. . . . .	58
4.12	Effect of the different weighting factors in the Gaussian distribution on the diagonals of a third-order least-squares reconstruction filter transfer function with six rings. . . . .	59

4.13	Effect of the commutation order on the diagonals of a weighted least-squares reconstruction filter transfer function with a Gaussian weighting ( $D = 3.0 \Delta$ ) . . . . .	60
4.14	Filter transfer function of a third-order weighted least-squares filter with $D = 3.5 \Delta$ , using a stencil with 1 neighbouring ring. . . . .	61
4.15	Diagonals of the filter transfer function of a third-order weighted least-squares filter with $D = 3.5 \Delta$ , using a stencil with 1 neighbouring ring. . . . .	61
4.16	Filter transfer function of a third-order weighted least-squares filter of with $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings. . . . .	62
4.17	Diagonals of the filter transfer function of a third-order weighted least-squares filter with $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings. . . . .	62
4.18	Three-dimensional spectrum of the application to a uniform spectrum of a third-order weighted least-squares filter with $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings. The shown iso-surface denotes $\hat{G}(\kappa_{\bar{\Delta}})$ . . . . .	63
4.19	Diagonals of a three-dimensional spectrum of the application to a uniform spectrum of a of a third-order weighted least-squares filter with $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings. . . . .	64
4.20	Demonstration of commutation of filters of degree 2, 3, 4, 5. Shown is the L2-norm of the commutation error. . . . .	65
5.1	Hybrid average gradient-diamond-path approach for a two-dimensional grid. . . . .	70
6.1	The Explicit_Filters class which provides the interface for the filtering framework.	73
6.2	The currently implemented inheritance relations. In blue the abstract classes, in purple the different implemented filters. Solid-lined arrows represent a “inherits from” relation, while dashed arrows represent a “makes use of” relation. . . . .	74
6.3	Application of explicit filtering to the Model Spectrum. . . . .	77
6.4	Q-criterion of filtered and unfiltered solution of the Model Spectrum. . . . .	78
6.5	Initial velocity field. Contour values show the velocity component normal to the faces, arrows show the velocity vectors on the faces. . . . .	80
6.6	Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms using explicit filtering on a $64 \times 64 \times 64$ mesh and a $128 \times 128 \times 128$ mesh. . . . .	81

6.7	Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms on a $128 \times 128 \times 128$ mesh using explicit filtering and implicit filtering. . . .	81
6.8	Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms on a $128 \times 128 \times 128$ mesh using explicit filtering and implicit filtering. . . .	82
6.9	Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms on a $128 \times 128 \times 128$ mesh using explicit filtering and implicit filtering. . . .	83
6.10	Coherent turbulence structures for application of isotropic turbulence decay after 50ms, for both explicit and implicit filtering, on a $64 \times 64 \times 64$ and a $128 \times 128 \times 128$ mesh. Visualization occurs through a positive isovalue of the Q-criterion. . . .	85

# List of Symbols

## Alphanumeric Symbols

$\check{q}_T$	subfilter-scale heat flux
$\mathbf{u}$	velocity vector
$\mathcal{L}_{ij}$	resolved Reynolds stress tensor
$T_{ij}$	test filter-scale Reynolds stress tensor
$E$	total energy
$G$	spatial filter function
$K$	specific total turbulent kinetic energy
$k$	specific turbulent kinetic energy
$k_c$	heat conductivity coefficient
$l$	length scale
$l_0$	characteristic length scale of largest eddies
$l_{\text{DI}}$	length scale dividing the dissipation and the inertial subrange
$l_{\text{EI}}$	length scale dividing the energy-containing range and the inertial subrange
$l_{11}$	integral length scale
$p$	pressure
$Pr_T$	turbulent Prandtl number
$q$	heat flux
$S_{ij}$	strain rate tensor
$T$	temperature
$u$	velocity in $x$ -direction
$u_0$	characteristic velocity of the largest eddies

$u_\eta$	Kolmogorov velocity scale
$u_i$	velocity component $i$
$v$	velocity in $y$ -direction
$w$	velocity in $z$ -direction
FGR	filter-grid ratio $\bar{\Delta}/\Delta$
$M^k$	$k^{\text{th}}$ spatial filter moment
SFS	subfilter-scale
$w_l$	discrete filter weight

### Greek Symbols

$\bar{\Delta}$	spatial filter width
$\Delta$	mesh spacing
$\eta$	Kolmogorov length scale
$\kappa$	wavenumber
$\kappa'$	modified wavenumber
$\kappa_\Delta$	grid cut-off wavenumber
$\kappa_{\text{DI}}$	wavenumber dividing the dissipation and the inertial subrange
$\kappa_{\text{EI}}$	wavenumber dividing the energy-containing range and the inertial subrange
$\kappa_{\bar{\Delta}}$	filter cut-off wavenumber
$\mu_T$	dynamic eddy viscosity
$\nu$	kinematic viscosity
$\nu_T$	kinematic eddy viscosity
$\rho$	density
$\sigma_{ij}$	viscous stress tensor
$\tau_\eta$	Kolmogorov time scale
$\tau_{ij}$	subfilter-scale Reynolds stress tensor
$\varepsilon$	dissipation rate

# Chapter 1

## Introduction

### 1.1 Introduction

Fluid Dynamics is the study of fluids in motion. The physics of any fluid flow are governed by three fundamental principles: conservation of mass, conservation of momentum, and conservation of energy. These fundamental principles can be expressed in terms of basic mathematical equations, which in their most general form are either integral or partial differential equations. Computational Fluid Dynamics (CFD) is the science of solving these equations numerically at discrete points in time and/or space. Such numerical simulations help us understand, model and later control the basic physical mechanisms that govern the particular fluid flow of interest.

There is an increasing demand for high fidelity, unsteady CFD capabilities for applications ranging from turbulence and transition modelling, to flow control, and to aero-acoustics, and to combustion system dynamics. For the best possible precision, the simulation has to take into account all the relevant space-time scales affecting the flow dynamics. When the range of scales is very large, as it is in turbulent flows, for example, the problem is stiff, in the sense that the ratio between the largest and smallest scales becomes very large. Direct Numerical Simulation (DNS) is therefore currently not suitable for industrial applications, [1], as it is too computationally expensive. Conventional Reynolds Averaged Navier Stokes (RANS) solvers based on various turbulence models, [2], often fail to capture unsteady flow physics accurately due in a large part to the lack of a clear separation of scales between the time scales associated with the mean flow variation and the turbulent fluctuations. Alternative methods are needed for unsteady CFD analyses in industrial applications. As computer power becomes more affordable, Large Eddy Simulation (LES) has emerged as a viable and powerful alternative technique in turbulence computations, [3, 4]. In this technique, the unsteady large scale motions are resolved,

while the more universal small scales are modelled. The separation of the small scales from the large scales is performed through a low-pass spatial filtering operation.

## 1.2 Motivation

Most practical LES methods to date have relied on the low-pass filtering properties of the discrete mesh and the spatial discretization scheme based on the assumption that the discrete numerics act to filter the solution. It is unclear however at which wavenumber this *implicit* filter acts to cut off the high-wavenumber solution content. This implicit approach has many problems associated with it: aliasing errors are not explicitly controlled, the filter width is virtually impossible to determine precisely, commutation between the filter and the differentiation operator is not considered, and the filter width is inherently linked with the mesh width. In the context of LES, there is a current need to compare and evaluate different proposed subfilter-scale models in order to assess their suitability for various applications. In order to carefully compare subfilter-scale models and compare LES predictions to DNS results (the latter would be helpful in the comparison and validation of models), there is a real need for a “grid-independent” LES capability. Currently however, it is difficult to perform LES that are independent of the grid since the grid and discretization procedure act together as the filter. This makes it hard to compare one subfilter-scale model to another as the numerical errors are not necessarily reduced to a sufficient level. This also means that with an adaptive grid the subfilter-scale models will act differently on the finer mesh than on the coarser mesh. Use of explicit spatial filtering with LES provides several advantages over implicit filtering and allows all of the problems described above to be addressed. It provides control of aliasing errors, commutation errors, and most importantly allows a decoupling between the mesh spacing and the filter width. The latter is the reason why it is difficult to directly compare LES solutions obtained using different grids.

## 1.3 Objectives

This thesis research seeks to establish a computational framework for performing grid-independent LES using a high-order finite-volume scheme and adaptive mesh refinement. Explicit filtering will be used to decouple the filter width from the mesh spacing, to control commutation errors on grids of varying mesh width, and thereby to achieve truly grid-independent solutions of the filtered Navier-Stokes equations.

The study, design and assessment of explicit filters will be the primary objective of this thesis

research. Several explicit filters will be studied. The explicit filters have to be constructed in such way that the order of the commutation error will be at least equal to the order of accuracy of the numerical scheme. Commutation errors arise when the filter width is not uniform throughout the grid. The commutation errors will also be studied and assessed for non-uniform grid types.

## 1.4 Overview of Thesis

An outline of the remainder of this thesis is as follows. In Chapter 2, a brief introduction to turbulent flows is given. The concept of LES is introduced, including the derivation of the filtered Navier-Stokes equations, and the possible sources of numerical errors when using LES. In Chapter 3, an extensive study is made of existing LES filtering techniques. Implicit filtering and several explicit filters (both analytical and discrete) are described, as well as possible approaches to the application of explicit filtering. Control of commutation errors is also discussed. In Chapter 4, two forms discrete explicit filters are studied in greater detail and assessed. The focus is on the optimal design of their transfer functions. In Chapter 5, the finite-volume scheme used here to perform LES of compressible turbulent flows is described. In Chapter 6, implementation of the developed explicit filtering framework with this research code is briefly discussed. It is then assessed by applying it to homogeneous isotropic turbulence. The thesis concludes with a summary of findings in Chapter 7.

## Chapter 2

# Large Eddy Simulation of Turbulent Flows

This chapter presents a general introduction to turbulent flows, and a summary of current techniques for obtaining numerical solutions of turbulent flows. LES methods are described and emphasized. The governing filtered equations for LES are systematically developed from the compressible form of the Navier-Stokes equations. Finally, issues related to implicit filtering for LES are explained.

### 2.1 Turbulent Flow

#### 2.1.1 Description

Turbulent flow is unsteady, irregular, seemingly random and chaotic, and inherently three dimensional. An essential feature of turbulent flows is that the velocity field varies significantly and irregularly in both position and time. Virtually all flows of practical engineering interest are turbulent: Flow past vehicles such as rockets, airplanes, ships and automobiles is always turbulent. Turbulence equally dominates in geophysical applications such as river currents and motion of clouds.

Another key feature of turbulence is that it increases the rate at which a fluid is stirred and/or mixed. Stirring is a process in which parcels of fluid with differing concentrations and/or properties are brought into contact. The actual final mixing is accomplished by molecular diffusion. Nonetheless the overall process is often called turbulent diffusion.

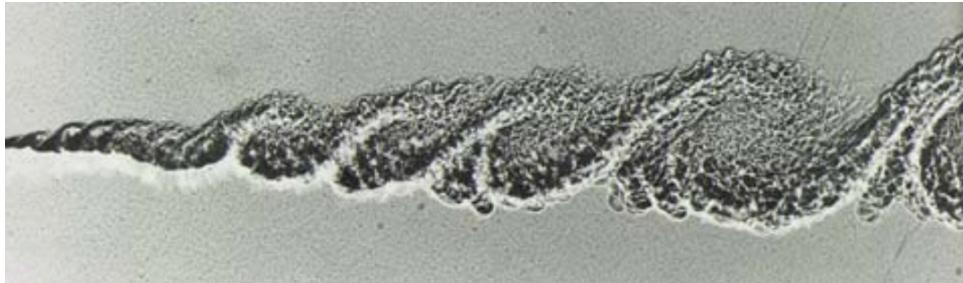


Figure 2.1: A large variation of length scales in the experimental turbulent mixing layer between helium (upper) and nitrogen (lower), (Brown and Roshko [5])

It is well established that in free shear flows that the turbulent motions range in size from the width of the mixing layer to much smaller scales, which become progressively smaller as the Reynolds number increases. Figure 2.1 shows a turbulent mixing layer between helium and nitrogen. It demonstrates that there exists a very wide range of scales in a turbulent flow from the large scales down to the smallest turbulent scale called the Kolmogorov scale,  $\eta$  (see further). In describing turbulence, many researches refer to turbulent eddies, which are local swirling motions where the vorticity can be very intense. In Figure 2.2, the structure of these turbulent eddies is depicted for an isotropic homogeneous flow field.

Perhaps the most important feature of turbulence, from an engineering point of view is its enhanced diffusivity. Turbulent diffusion greatly enhances the transfer of mass, momentum and energy. Apparent stresses in turbulent flows are often several orders of magnitude larger than in corresponding laminar flows, [2].

### 2.1.2 The Energy Cascade

In 1922, Richardson introduced the concept of the energy cascade [6]. The idea is that kinetic energy enters the turbulence at the largest scales of motion. This energy is then transferred by inviscid processes to smaller and smaller scales until at the smallest scales, the energy is dissipated by viscous processes. In 1941, Kolmogorov identified the smallest scales of turbulence to be those that now bear his name [6]. In this concept, the turbulence can be considered to be composed of eddies of different sizes that overlap in space, large ones carrying smaller ones. The largest eddies are characterized by the length scale  $l_0$  which is comparable to the flow scale, and a characteristic velocity  $u_0$  which is on the order of the turbulence intensity. The Reynolds number of these eddies  $Re_0 \equiv \frac{u_0 l_0}{\nu}$  is therefore large and the effects of viscosity are negligibly small. A comparable length scale is the integral length scale  $l_{11}$  which is the length scale for

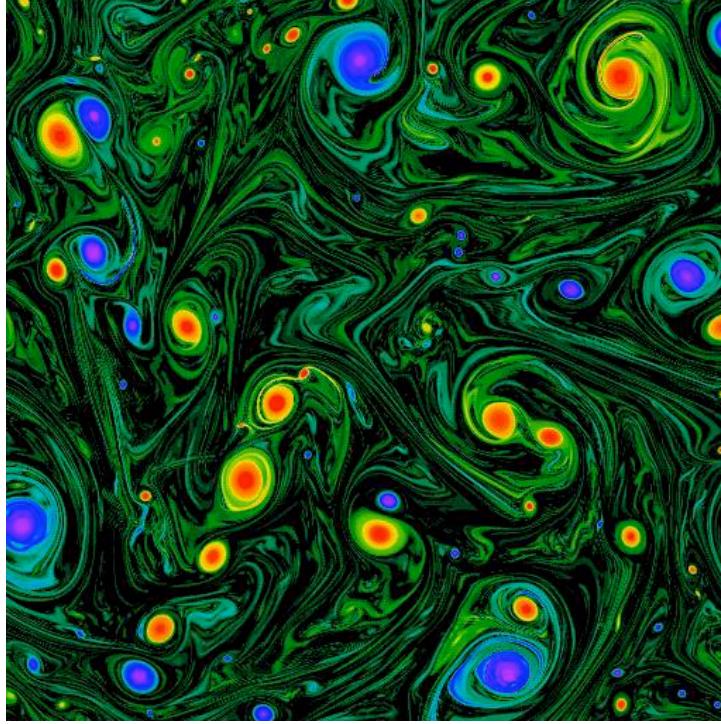


Figure 2.2: Visualization of turbulent eddies in an isotropic turbulent flow

which eddies remain well correlated. For homogeneous isotropic turbulence this is proportional with  $l_0$  as  $l_{11} \approx 0.43 l_0$ .

Richardson noted that the large eddies are unstable and break up, transferring their energy to smaller eddies, which in turn undergo a similar process. At the smallest scales the Reynolds number of these small eddies will be sufficiently small that the eddy motion is stable and molecular viscosity is effective in dissipating the kinetic energy. The importance is that dissipation is only really significant in the last process in the cascade. The rate of dissipation  $\varepsilon$  is therefore determined by the first process in the cascade which is the transfer of energy from the largest eddies. If  $K$  is the total specific turbulent kinetic energy defined by  $K = \frac{3}{2} u_{\text{rms}}^2$ , then

$$\varepsilon \equiv \frac{K^{3/2}}{l_0}. \quad (2.1)$$

**Kolmogorov's hypothesis of local isotropy** is that in every turbulent flow at sufficiently high Reynolds number, the small-scale turbulent motions ( $l \ll l_0$ ) are statistically isotropic (see Figure 2.1). One can introduce a length scale  $l_{EI}$  which separates the anisotropic large eddies from the isotropic small eddies.

**Kolmogorov's first similarity hypothesis** is that the small scale turbulent motions have a universal form that is uniquely determined by  $\nu$  and  $\varepsilon$ . The Kolmogorov length, velocity and

time scales are thus defined:

$$\eta \equiv (\nu^3/\varepsilon)^{1/4} \quad (2.2)$$

$$u_\eta \equiv (\varepsilon\nu)^{1/4} \quad (2.3)$$

$$\tau_\eta \equiv (\nu/\varepsilon)^{1/2}. \quad (2.4)$$

Kolmogorov argued that all information about the geometry of the large eddies which is determined by the mean flow field and boundary conditions, is also lost. As a consequence, the statistics of the small-scale motions ( $l < l_{EI}$ ) are in a sense universal. In this range the time scales are small compared with  $l_0/u_0$ , so that the small eddies can adapt quickly to maintain a dynamic equilibrium with the energy-transfer rate imposed by the large eddies. For this reason this range is called the *universal equilibrium range*. Defining a length scale  $l_{DI}$ , one can split the universal equilibrium range into two distinct subranges: the *inertial subrange* ( $l_{EI} > l > l_{DI}$ ) and the *dissipation range* ( $l < l_{DI}$ ).

**Kolmogorov's second similarity hypothesis** is that in the inertial subrange the motions have a universal form that is uniquely determined by  $\varepsilon$ , independent of  $\nu$ . So all the motions in the inertial subrange are determined by inertial effects, viscous effects being negligible. Only motions in the dissipation range experience significant viscous effects, and are responsible for essentially all of the dissipation. The bulk of the energy is contained in the large eddies in the size range  $l > l_{EI}$ . This size range is therefore called the *energy containing range*. Figure 2.3 depicts the various length scales and ranges. Through the energy cascade, the energy produced in the large scales is transferred through the inertial subrange and dissipated in the small scales.

### 2.1.3 Spectral representation

Further insights can be gained by examining the distribution of turbulent energy for isotropic turbulence in wavenumber space. If  $E(\kappa)d\kappa$  is the specific turbulence kinetic energy contained between wavenumbers  $\kappa$  and  $\kappa + d\kappa$ , we can write the total specific turbulent kinetic energy as

$$K = \int_0^\infty E(\kappa)d\kappa \quad (2.5)$$

According to Kolmogorov's first similarity hypothesis, the velocity statistics corresponding to the universal equilibrium range have a universal form that is uniquely determined by  $\varepsilon$  and  $\nu$ . Dimensional analysis shows that  $E(\kappa)$  is then a universal function of  $\kappa$ ,  $\varepsilon$  and  $\nu$  and has the form

$$E(\kappa) = \varepsilon^{2/3}\kappa^{-5/3}\psi(\kappa\eta), \quad (2.6)$$

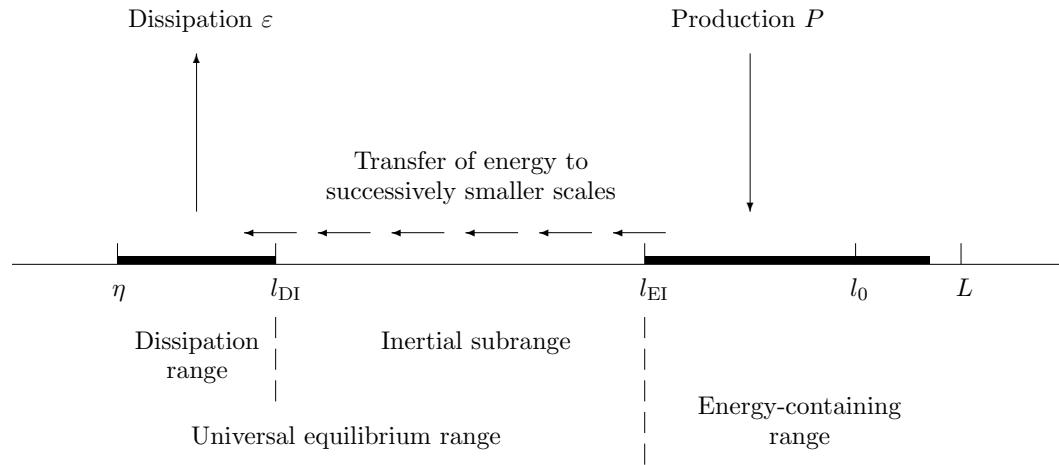


Figure 2.3: Schematic of the energy cascade at high Reynolds number (logarithmic scale).

where  $\psi(\kappa\eta)$  is a universal non-dimensional function. Note that the influence of viscosity,  $\nu$ , is incorporated in  $\eta$ . In the inertial subrange Kolmogorov's second similarity hypothesis says that  $E(\kappa)$  has a universal form independent of  $\nu$ . Since  $\nu$  enters through  $\eta$  this means that  $\psi(\kappa\eta)$  will become a constant, and the energy-spectrum function in the inertial subrange becomes:

$$E(\kappa) = C\varepsilon^{2/3}\kappa^{-5/3}. \quad (2.7)$$

This is the famous Kolmogorov  $-\frac{5}{3}$  spectrum, and  $C$  is a universal Kolmogorov constant ( $C = 1.5$ ). Figure 2.4 illustrates this function and shows the  $-\frac{5}{3}$  law in the inertial subrange. These features of the spectrum for isotropic turbulence have also been observed in both experiments and numerical simulations.

## 2.2 Numerical Solution of Turbulent Flows

Turbulence modelling is one of the three key elements in CFD. Very precise mathematical theories have evolved for the other two key elements: grid generation and algorithm development. In creating a mathematical model that approximates the physical behavior of turbulent flows, far less precision has been achieved. This is not really surprising since the objective involves approximating an extremely complicated phenomenon. Three main techniques or approaches can be distinguished: Direct Numerical Simulation (DNS), Reynolds Averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES).

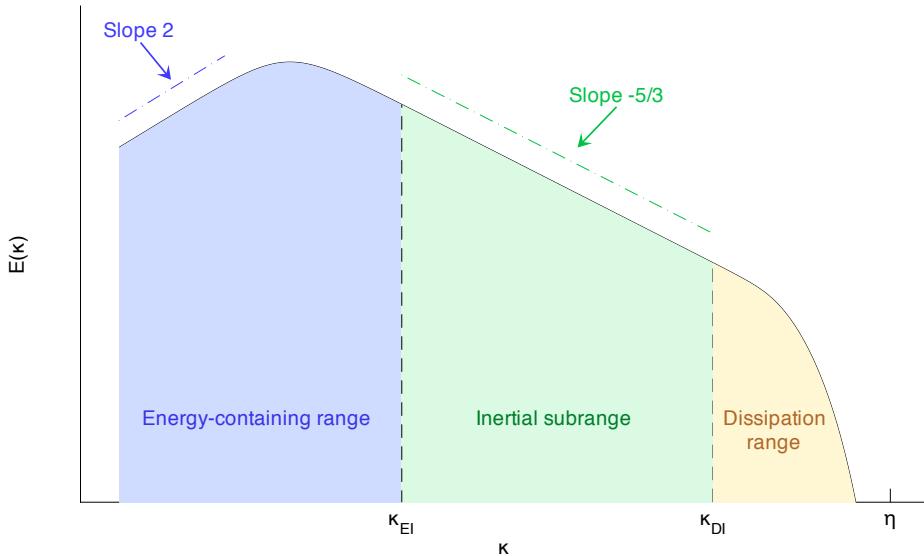


Figure 2.4: Energy-spectrum function for isotropic turbulence (logarithmic scales).

### 2.2.1 Direct Numerical Simulation

DNS involves the solution of the equations of conservation of mass, momentum and energy down to the smallest physical scale being the Kolmogorov scale [1]. This technique involves no modelling since the smallest physical scales of the energy spectrum are fully resolved. However in order to resolve the smallest scales, the computational mesh has to be very fine to capture these scales. Typically at least 4 grid points in each direction are needed to resolve the smallest eddies. Also time steps of such simulations must be of the order of the Kolmogorov time scale,  $\tau_\eta$ . The value of such simulations is obvious: they give in principle the proper solution to the turbulence problem, as no modelling is involved. It is the most accurate method for the treatment of turbulent flows, but it is also the most computationally expensive method. Therefore, its use is impractical for most engineering problems involving complex geometries. Its main use is to validate other turbulence models, or it could also be viewed as an additional source of experimental data. This can be the case for many of the essentially unmeasurable properties.

### 2.2.2 Reynolds Averaged Navier-Stokes

RANS is an approach where the Navier-Stokes equations are time-averaged, effectively removing the turbulent fluctuations from the flow solution. Their influence on the mean flow solution must

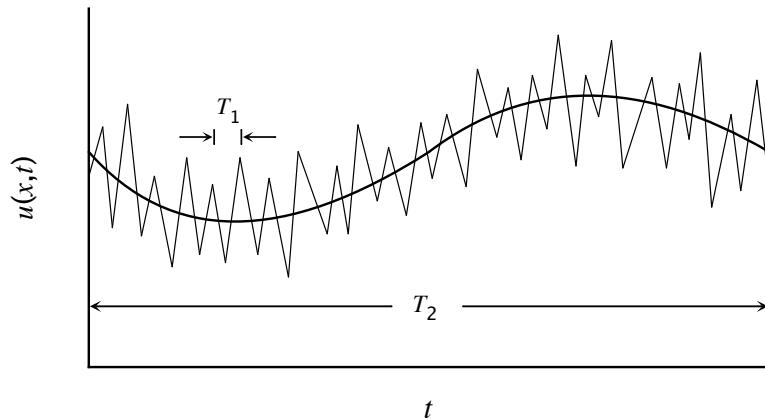


Figure 2.5: Velocity fluctuations as a function of time

be modelled. Since the fast fluctuations are time-averaged, the grid requirements are far less than those for DNS.

In order to use RANS for unsteady flows it must be possible to separate the time scales of the mean flow and the components of the turbulent fluctuations. By time-averaging one loses upstream history of the flow. And it is observed that the large eddies have a very long lifetime migrating across the flow. RANS has therefore a hard time representing the effect of these large eddies migrating across the flow. An example of such separation of time scales is shown in Figure 2.5.

The time-averaging procedure introduces some unclosed terms which need to be modelled, [2]. Several models have been proposed in the past such as algebraic models (Baldwin - Lomax 1978), one-equation models (Spalart - Allmaras 1992), two-equation models ( $k-\varepsilon$  by Launder 1972,  $k-\omega$  by Wilcox 1988) and stress-transport models (Launder - Reece - Rodi 1975). In general, RANS techniques are computationally inexpensive enough to be used in practical engineering applications

### 2.2.3 Large Eddy Simulation

In LES, the larger three-dimensional unsteady turbulent motions are computed explicitly while the effect of smaller-scale motions are modelled. The application and validity of this approach is supported by the observed similarity of the small scales (Kolmogorov's similarity hypothesis, Section 2.1.2), which tend to only transmit energy to smaller scales (energy cascade), and the fact that the dissipation is therefore dictated by the energy containing large scales. A spatial

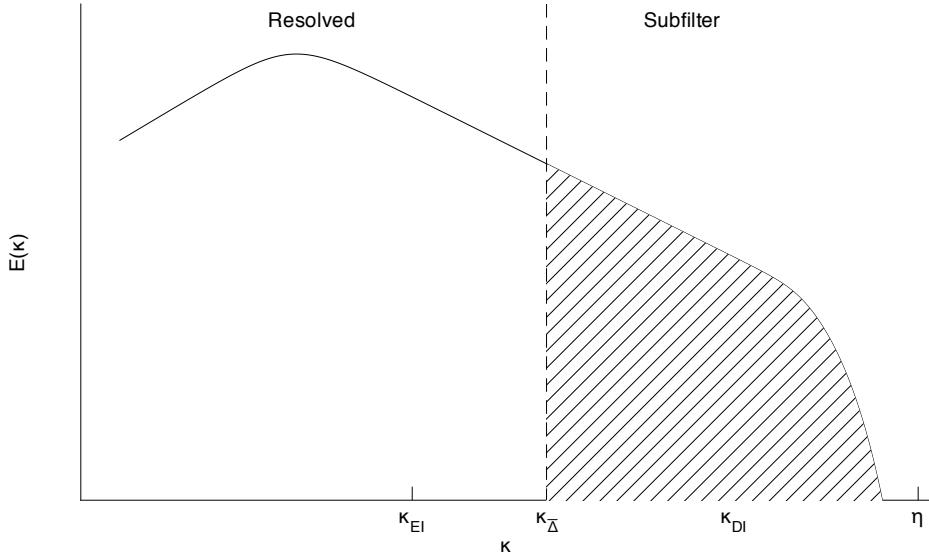


Figure 2.6: The resolved and subfilter range in the isotropic turbulence spectrum.  $\kappa_{\bar{\Delta}}$  represents the filter cut-off wavenumber.

filtering operation divides the turbulence field into a filtered (or resolved) component and a residual or subfilter-scale (SFS) component. Typically, the spatial filter width is chosen such that 80% of the total turbulent kinetic energy,  $K$ , is resolved and the filter cut-off wavenumber,  $\kappa_{\bar{\Delta}}$  then lies somewhere in the inertial subrange. Figure 2.6 illustrates the LES concept and spatially filtering procedure in the case of an ideal spectral cut-off filter.

A SFS model is required to represent the influence of the small scales on the resolved turbulence. The main role of the SFS model is to remove energy from the explicitly computed scales, mimicking the drain that is usually associated with the energy cascade. Hence it is not necessary for a SFS model to represent the “exact” SFS stresses at each point in time and space, but only to account for their global effect [3, 4]. In terms of computational expense, LES lies somewhere between RANS and DNS and its development and application is motivated by the limitations and/or drawbacks of these other approaches.

## 2.3 Large Eddy Simulation Techniques

### 2.3.1 Favre-Filtered Navier-Stokes Equations

The compressible Navier-Stokes equations govern the fluid dynamics for Newtonian fluids. The Navier-Stokes equations for a compressible gas includes the continuity, momentum and energy

equations which, using tensor notation, can be written as follows:

Mass Conservation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0 \quad (2.8)$$

Momentum Conservation:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0 \quad (2.9)$$

Total Energy Conservation:

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial((\rho E + p)u_j)}{\partial x_j} - \frac{\partial(\sigma_{ij}u_j)}{\partial x_j} + \frac{\partial q_j}{\partial x_j} = 0 \quad (2.10)$$

In these equations,  $p$  is defined as the pressure,  $\rho$  the density,  $u_i$  the velocity vector,  $\sigma_{ij}$  the viscous stress tensor,  $E$  the total energy per unit volume, and  $q_j$  the heat flux with the following definitions:

$$\sigma_{ij} = 2\rho\nu \left( S_{ij} - \frac{\delta_{ij}}{3}S_{kk} \right), \quad (2.11)$$

$$q_j = -k_c \frac{\partial T}{\partial x_j}, \quad (2.12)$$

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u_i u_i, \quad (2.13)$$

where  $S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$  is the strain rate tensor,  $k_c$  is the thermal conductivity, and  $\nu$  is the kinematic viscosity. The latter are generally functions of the temperature,  $T$ .

In LES, these governing equations are filtered by a spatial filtering operation. The spatial filtering procedure is defined as

$$\bar{\phi}(\mathbf{x}) = \int_D \phi(\mathbf{x}') G(\mathbf{x}, \mathbf{x}'; \bar{\Delta}) d\mathbf{x}', \quad (2.14)$$

where the filtered variable is denoted by an overbar.  $G(\mathbf{x}, \mathbf{x}'; \bar{\Delta})$  is the filter function and  $\bar{\Delta}$  the filter width. This procedure will be further discussed in Chapter 3. In order to reduce the number of additional terms appearing when applying this filtering operation to the compressible Navier-Stokes equations, a Favre-filtering operation is introduced. Favre-filtering is essentially a density-weighted filtering procedure defined as

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\overline{\rho}}, \quad (2.15)$$

where tilde represents the Favre-filtered variable. The Favre-filtered form of the Navier-Stokes equations for compressible flow [3, 7], is then given by

Mass Conservation:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial(\bar{\rho} \bar{u}_i)}{\partial x_i} = 0 \quad (2.16)$$

Momentum Conservation:

$$\frac{\partial(\bar{\rho} \bar{u}_i)}{\partial t} + \frac{\partial(\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} - \underbrace{\frac{\partial \check{\sigma}_{ij}}{\partial x_j}}_{\text{I}} + \frac{\partial \bar{p}}{\partial x_i} = - \underbrace{\frac{\partial \tau_{ij}}{\partial x_j}}_{\text{II}} + \underbrace{\frac{\partial(\bar{\sigma}_{ij} - \check{\sigma}_{ij})}{\partial x_j}}_{\text{III}} \quad (2.17)$$

Total Energy Conservation:

$$\begin{aligned} \frac{\partial(\bar{\rho} \bar{E})}{\partial t} + \frac{\partial((\bar{\rho} \bar{E} + \bar{p}) \tilde{u}_j)}{\partial x_j} - \frac{\partial(\check{\sigma}_{ij} \tilde{u}_j)}{\partial x_j} + \frac{\partial \check{q}_j}{\partial x_j} &= - \underbrace{\frac{1}{\gamma - 1} \frac{\partial(\bar{p} \bar{u}_j - \bar{p} \tilde{u}_j)}{\partial x_j}}_{\text{III}} - \underbrace{\frac{\partial(\bar{p} \bar{u}_j - \bar{p} \tilde{u}_j)}{\partial x_j}}_{\text{IV}} \\ &+ \underbrace{\frac{\partial(\bar{\sigma}_{ij} \tilde{u}_i - \check{\sigma}_{ij} \tilde{u}_i)}{\partial x_j}}_{\text{V}} + \underbrace{\frac{\partial(\bar{u}_i \bar{\sigma}_{ij} - \tilde{u}_i \bar{\sigma}_{ij})}{\partial x_j}}_{\text{VI}} - \underbrace{\frac{1}{2} \frac{\partial(\bar{\rho} \bar{u}_j \bar{u}_i \bar{u}_i - \bar{\rho} \bar{u}_i \bar{u}_i \tilde{u}_j)}{\partial x_j}}_{\text{VII}} + \underbrace{\frac{\partial(\bar{q}_j - \check{q}_j)}{\partial x_j}}_{\text{VIII}} \end{aligned} \quad (2.18)$$

In these equations, the SFS stress tensor,  $\tau_{ij}$ , is defined by

$$\tau_{ij} = \bar{\rho} \bar{u}_i \bar{u}_j - \bar{\rho} \tilde{u}_i \tilde{u}_j, \quad (2.19)$$

and the breve represents the variables in terms of filtered variables [8, 7]:

$$\check{\sigma}_{ij} = \sigma_{ij}(\tilde{\mathbf{u}}, \tilde{T}) \quad (2.20)$$

$$\check{q}_j = q_j(\tilde{T}) \quad (2.21)$$

In these equations, the terms on the right-hand side are known as the subfilter terms, which represent the effect of the unresolved scales on the resolved solution content. Unlike the terms on the left-hand side of the equations, these terms cannot be expressed in terms of the filtered flow variables. Since a Favre-filtering is used here, no subfilter-terms appear in the continuity equation. A summary of the subfilter-terms, introduced by the Favre-filtering of the Navier-Stokes equations, is as follows:

- Term I results from the non-linearity of the convective term and needs modelling.
- Term II results from the non-linearity of the viscous term and the fact that the Favre filter and partial derivatives do not commute. Term II may be neglected under the assumption that  $\bar{\sigma}_{ij} - \check{\sigma}_{ij} = 0$ . A priori tests confirm that it is an order of magnitude smaller than term I [7].
- Term III and IV are the pressure-velocity subfilter terms, representing the effect of the SFS turbulence on the conduction of heat in the resolved scales. They have to be modelled.

- Term V and VIII are created by the non-linearities of respectively the viscous stresses and the heat flux. With similar assumptions as for term II, these terms can be neglected.
- Term VI represents the SFS viscous diffusion and is generally much smaller than the other terms that require modelling, and can thus be neglected [9].
- Term VII is the SFS turbulent diffusion and has to be modelled.

One can derive a definition for the specific SFS turbulent kinetic energy,  $k$ , by applying the Favre-filter to (2.13) to yield

$$\tilde{E} = \frac{\bar{p}}{\gamma - 1} + \frac{1}{2}\bar{\rho}\widetilde{u_i u_i}. \quad (2.22)$$

The second term on the right hand side represents kinetic energy and can be written as

$$\begin{aligned} \frac{1}{2}\bar{\rho}\widetilde{u_i u_i} &= \frac{1}{2}\bar{\rho}\widetilde{u_i u_i} + \frac{1}{2}\bar{\rho}\underbrace{(\widetilde{u_i u_i} - \widetilde{u_i u_i})}_k \\ &= \frac{1}{2}\bar{\rho}\widetilde{u_i u_i} + \frac{1}{2}\bar{\rho}k. \end{aligned} \quad (2.23)$$

From (2.23) and (2.19), it follows that  $k$  is related to the trace of the SFS stress and given by

$$\begin{aligned} \tau_{kk} &= \bar{\rho}(\widetilde{u_k u_k} - \widetilde{u_k u_k}) \\ &= 2\bar{\rho}k \end{aligned} \quad (2.24)$$

### 2.3.2 Subfilter-Scale Modelling

As mentioned above, the unresolved filtered scales have an effect on the resolved scales. To account for these effects terms I, III, IV and VII of (2.17) - (2.18) have to be modelled, and terms II, V, VI and VIII can be neglected.

#### Subfilter-Scale Stress

The SFS stress is represented by term I. The model for this term involves most consideration as it is primarily responsible for the mimicking of the energy cascade. Several SFS stress models have been postulated of which the Smagorinsky model is the progenitor of all such models [10]

##### *The Smagorinsky Model*

The SFS stress model proposed by Smagorinsky [11] models the SFS stress tensor in a fashion similar to the Bousinesq approximation, and is used to model the Reynolds-stress tensor: a SFS eddy viscosity is defined and is used to relate the SFS stress to the filtered strain rate. This

modelling approach is based on the assumption that the SFS stress tensor behaves similarly to the viscous stress tensor, which is also proportional to the strain rate tensor. In this model, the proportionality factor is called the SFS eddy viscosity,  $\nu_T$ . All SFS eddy viscosity models are based on the notion that the principle effects of the SFS stress on the filtered flow quantities are increased transport and dissipation. Smagorinsky postulated that the SFS eddy viscosity has the form

$$\nu_T = (C_s \bar{\Delta})^2 |\check{S}|, \quad (2.25)$$

where  $|\check{S}|$  is the magnitude of the filtered strain rate tensor given by

$$|\check{S}| = \sqrt{2\check{S}_{ij}\check{S}_{ij}}, \quad (2.26)$$

and  $C_s$  the Smagorinsky coefficient, which can be determined from isotropic turbulence decay, and lies in the range between 0.1 and 0.24. In (2.25), the factor  $\bar{\Delta}$  can be seen as the mixing length and  $C_s^2 \bar{\Delta} |\check{S}|$  as the mixing velocity. The SFS stress is then given by

$$\tau_{ij} = -2\bar{\rho}\nu_T \left( \check{S}_{ij} - \frac{\delta_{ij}}{3} \check{S}_{kk} \right) + \frac{\delta_{ij}}{3} \tau_{kk}. \quad (2.27)$$

The trace of the SFS stress tensor  $\tau_{kk}$  is related to  $k$  as in (2.24). Deardorff [12] proposed a model for  $k$  which was further studied by Yoshizawa et al. [13]:

$$k = \frac{\nu_T^2}{(C_v \bar{\Delta})^2}, \quad (2.28)$$

where the constant  $C_v = 0.094$ .

### *The Dynamic Smagorinsky Model*

The dynamic Smagorinsky model is based on the Smagorinsky model. The main concern with the Smagorinsky model is that the Smagorinsky coefficient is only valid for the particular flow regime for which the coefficient  $C_s$  has been optimized. This means that in some regimes of the flow the SFS model may be too dissipative or vice versa. The dynamic Smagorinsky model tries to solve this problem by using a variable Smagorinsky coefficient [14]. This coefficient is calculated by means of a test filter with a filter width typically twice that of the LES filter. The test filtering operation is denoted with a hat. If  $T_{ij}$  is the test filter-scale stress tensor and  $\hat{\tau}_{ij}$  the test filtered SFS stress tensor, one can take the difference which is called the resolved stress tensor [15]:

$$\mathcal{L}_{ij} \equiv T_{ij} - \hat{\tau}_{ij} \quad (2.29)$$

$$= \left( \widehat{\rho u_i u_j} - \widehat{\rho u_i} \widehat{\rho u_j} / \hat{\rho} \right) - \left( \widehat{\rho u_i u_j} - (\overline{\rho u_i} \overline{\rho u_j} / \bar{\rho}) \widehat{} \right) \quad (2.30)$$

$$= (\overline{\rho u_i} \overline{\rho u_j} / \bar{\rho}) \widehat{} - \widehat{\rho u_i} \widehat{\rho u_j} / \hat{\rho} \quad (2.31)$$

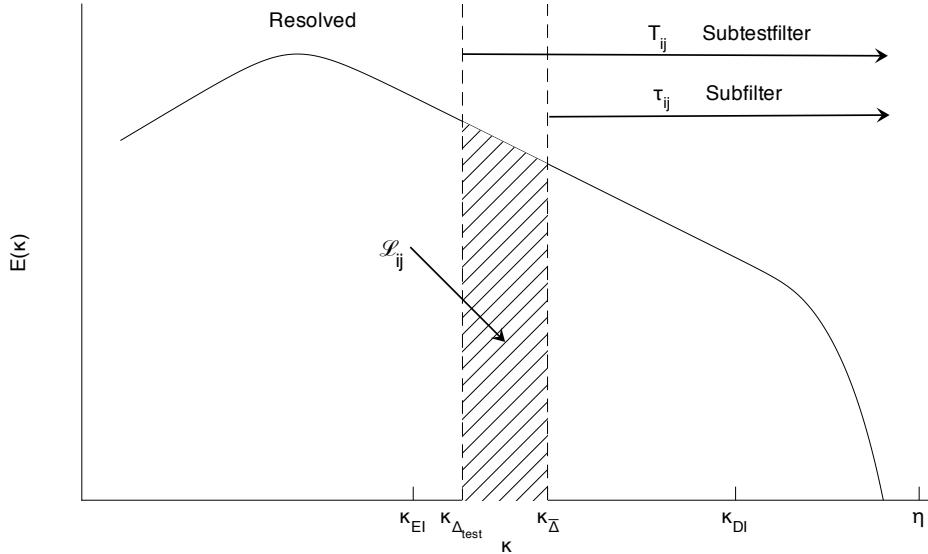


Figure 2.7: Schematic of the resolved Reynolds stress  $\mathcal{L}_{ij}$  used in the dynamic Smagorinsky model.

Equation (2.29) is called the Germano identity [14]. This is schematically represented in Figure 2.7.

Simplified models for  $\tau_{ij}$  and  $T_{ij}$  are then introduced in to (2.29). It is assumed that  $\tau_{ij}$  and  $T_{ij}$  have the forms  $\tau_{ij} = -2C\alpha_{ij}$  and  $T_{ij} = -2C\beta_{ij}$  with  $C = C_s^2$ . Substituting this in to (2.29) one gets:

$$\begin{aligned}\mathcal{L}_{ij} &= -2C(\beta_{ij} - \hat{\alpha}_{ij}) \\ &= -2CM_{ij}\end{aligned}\tag{2.32}$$

Lilly [16] proposed using a least squares procedure to minimize the error  $e_{ij} = \mathcal{L}_{ij} + 2CM_{ij}$  to get a value for  $C$  and thus  $C_s$ .

### *The k-equation Model*

The  $k$ -equation model was proposed by Yoshizawa et al. [13], and improved by Menon et al. [17]. In this model a scalar transport equation for the SFS kinetic energy  $k$  is solved along with the Favre-filtered Navier-Stokes equations:

$$\frac{\partial \bar{\rho}k}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{k}}{\partial x_i} = \underbrace{-\tau_{ij} \check{S}_{ij}}_I - \underbrace{C_\varepsilon \frac{\bar{\rho} \tilde{k}^{3/2}}{\Delta}}_II + \frac{\partial}{\partial x_k} \left( \bar{\rho} \frac{\nu_T}{Pr_T} \frac{\partial \tilde{k}}{\partial x_k} \right).\tag{2.33}$$

Term I and II in this equation are respectively the production and dissipation of SFS turbulent kinetic energy,  $k$ .  $C_\varepsilon$  is a constant taken to be  $C_\varepsilon = 0.845$ , [17]. This model is similar to the

dynamic approach in that the SFS eddy viscosity,  $\nu_T$ , is determined from the solution. The model offers the advantages that non-equilibrium effects at the subfilter-scale level are taken into account and the eddy viscosity does vanish for a resolved laminar flow, i.e., when  $\tilde{k} \rightarrow 0$ .

Several other SFS stress models exist such as two-point closures, scale-similar models, mixed models. For a summary the reader is referred to [4].

### Subfilter-Scale Heat Conduction

The SFS heat conduction is being represented by term III and IV. These 2 terms can be modelled together using a gradient or diffusion approximation similar to Fourier's law for heat conduction [18]:

$$\text{III + IV} = \frac{\partial}{\partial x_j} \left( -\frac{\mu_T}{Pr_T} \frac{\partial \check{T}}{\partial x_j} \right) \quad (2.34)$$

where typically  $Pr_T \approx 0.7 - 0.9$ .

### Subfilter-Scale Turbulent Diffusion

The SFS turbulent diffusion is represented by term VII and may be modelled as [19]:

$$\text{VII} = \frac{\partial}{\partial x_j} (\tilde{u}_i \tau_{ji}) \quad (2.35)$$

### 2.3.3 Closed Favre-Filtered Navier-Stokes Equations

Using the SFS modelling described above, the Favre-filtered form of the Navier-Stokes equations for compressible flow can be written as

Mass Conservation:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial(\bar{\rho} \bar{u}_i)}{\partial x_i} = 0 \quad (2.36)$$

Momentum Conservation:

$$\frac{\partial(\bar{\rho} \bar{u}_i)}{\partial t} + \frac{\partial(\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} - \frac{\partial \check{\sigma}_{ij}}{\partial x_j} + \frac{\partial \bar{p}}{\partial x_i} = - \frac{\partial \tau_{ij}}{\partial x_j} \quad (2.37)$$

Total Energy Conservation:

$$\frac{\partial(\bar{\rho} \bar{E})}{\partial t} + \frac{\partial((\bar{\rho} \bar{E} + \bar{p}) \tilde{u}_j)}{\partial x_j} - \frac{\partial(\check{\sigma}_{ij} \tilde{u}_j)}{\partial x_j} + \frac{\partial \check{q}_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left( -\frac{\mu_T}{Pr_T} \frac{\partial \check{T}}{\partial x_j} \right) - \frac{\partial}{\partial x_j} (\tilde{u}_i \tau_{ji}) \quad (2.38)$$

### 2.3.4 Implicit versus Explicit Filtering

In traditional LES, the filtering operation described in Section 2.3.1 is not explicitly described. Instead the filtering happens through the combined effects of the computational grid and the discretization operators. The wavenumber-dependence of the dissipative properties of spatial discretization schemes act as an effective “implicit” filter. Partial support for this argument is based on the following equivalence between a finite-difference operator and the exact derivative of a filtered variable [20] :

$$\left. \frac{\delta u}{\delta x} \right|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} = \frac{d}{dx} \int_{x_{i-1}}^{x_{i+1}} u(x') dx' = \left. \frac{d\bar{u}}{dx} \right|_i \quad (2.39)$$

This implies that the filter width  $\bar{\Delta}$  is inherently related to the mesh spacing. For an adaptive mesh this means that as the mesh is refined the filtering procedure is applied with a narrower filter width and the filter will act differently on the solution. It is difficult to then draw conclusions on the computed solution. In particular if the mesh is refined, additional length scales are added each time the mesh is refined, and thus the process converges eventually to a DNS rather than a LES. This makes it difficult to distinguish between the effects of reduced numerical error and the increase in the range of resolved length scales.

The filter width,  $\bar{\Delta}$ , used in implicit filtering is coupled with the mesh-spacing, making it difficult to compare the results obtained in an adaptive mesh. With explicit filtering,  $\bar{\Delta}$  can be held fixed throughout the entire grid. Applying identical spatial filtering procedure to results obtained with DNS also allows one to compare an LES solution with reference DNS results.

Another advantage of explicit filtering is that it allows for the use of a filter where the order of the commutation errors can be controlled and maintained to the same order of the truncation errors (see further in Section 2.3.6 below for further discussion). Finally explicit filtering also allows to control the aliasing errors that arise using implicit filtering (see Section 2.3.5 below).

### 2.3.5 Aliasing Errors

Favre filtering of the convective term in the Navier-Stokes equation results in the non-linear correlation

$$\frac{\partial}{\partial x_j} (\bar{\rho} u_i u_j) , \quad (2.40)$$

which creates closure problems. In LES, this term is typically treated by computing the product of the closed filtered velocities and modelling the remainder which essentially transfers the

closure problem to the right-hand side of the filtered Navier-Stokes equation:

$$\overline{\rho u_i u_j} = \bar{\rho} \tilde{u}_i \tilde{u}_j + \underbrace{(\overline{\rho u_i u_j} - \bar{\rho} \tilde{u}_i \tilde{u}_j)}_{\tau_{ij}}, \quad (2.41)$$

where  $\tau_{ij}$  is the SFS stress tensor. The problem with this decomposition is that the non-linear product  $\tilde{u}_i \tilde{u}_j$  generates frequencies beyond the characteristic frequency that defines  $\tilde{u}_i$ . These frequencies alias back as resolved ones and therefore act as fictitious stresses [21]. The errors caused by this aliasing are difficult to eliminate and control if implicit filtering is used.

### 2.3.6 Commutation Errors

For inhomogeneous turbulent flows, the minimum size of eddies that need to be resolved is different in different regions of the flow. In general, the closer to solid walls the smaller the eddies become due to the damping effects of the wall. This implies that the mesh should generally become smaller to resolve these scales in the near-wall region. In order to ensure that the structure of the filtered equations remains the same before and after filtering, the filtering operation should commute with the differential operation as follows:

$$\overline{\frac{d\phi}{dx}} = \frac{d\bar{\phi}}{dx}. \quad (2.42)$$

In general, filters do not commute when a variable filter width is used [22]. To be acceptable, the errors associated with the commutation properties of the filter should be of the same order as the truncation errors associated with the numerical scheme, i.e.,

$$\left[ \frac{d\phi}{dx} \right] \equiv \left| \overline{\frac{d\phi}{dx}} - \frac{d\bar{\phi}}{dx} \right| = \mathcal{O}(\Delta^n), \quad (2.43)$$

where  $n$  is the order of the filter, which should be at least equal to the order of accuracy of the spatial discretization scheme.

## Chapter 3

# Spatial Filtering Techniques

This chapter provides a general overview of the current spatial filtering techniques and approaches. Firstly, implicit filtering is reviewed. Secondly, a number of existing explicit filters are described, where a distinction is made between analytical explicit filters and discrete explicit filters. Finally an explanation is given as to how commutation errors may be controlled.

### 3.1 Implicit Spatial Filtering

Implicit spatial filtering was briefly described in Section 2.3.4. This section will go deeper into of how filtering of solution content occurs without actually defining an explicit filter.

Consider a solution variable  $f(x)$  defined in physical space as follows

$$f(x) = \sum_{j=1}^N e^{i\kappa_j x}. \quad (3.1)$$

This expression indicates that  $f(x)$  can be represented by a Fourier series consisting of  $N$  modes with wavenumbers  $\kappa_j$  with  $j = 1 \cdots N$ . The analytical differentiation of  $f(x)$  is of course given by

$$\frac{\partial f(x)}{\partial x} = \sum_{j=1}^N i\kappa_j e^{i\kappa_j x}. \quad (3.2)$$

However numerical differentiation gives

$$\frac{\delta f(x)}{\delta x} = \sum_{j=1}^N i\kappa'_j e^{i\kappa_j x}, \quad (3.3)$$

where  $\kappa'$  is often referred to as the modified wavenumber. This modified wavenumber depends on the particular spatial discretization scheme and depending on the scheme and it can contain

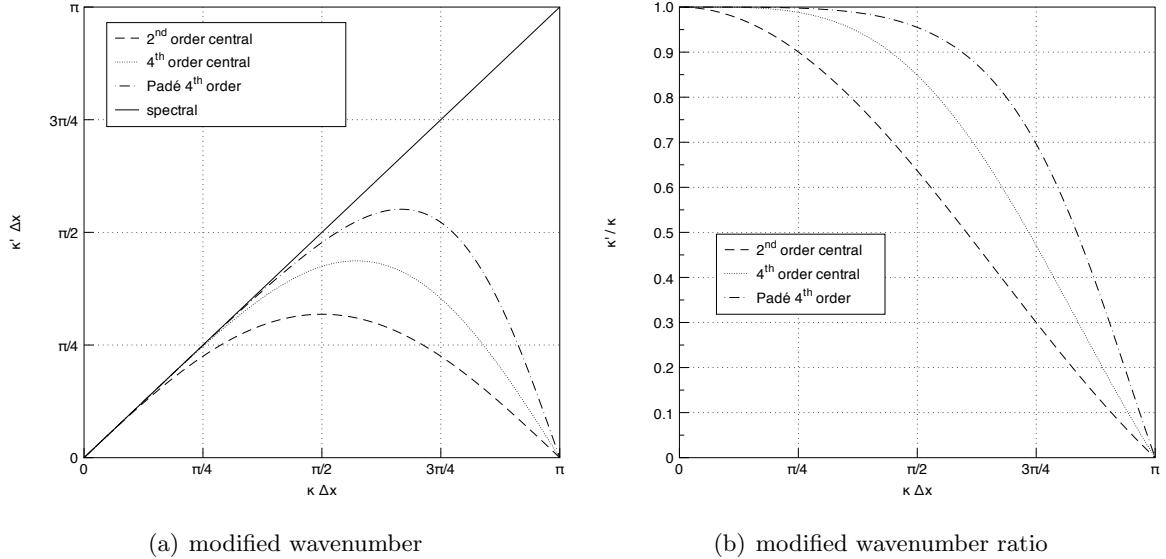


Figure 3.1: Modified wavenumber for various schemes

both real and imaginary components. In the case of a second-order central finite-difference scheme the modified wavenumber is defined as

$$\kappa' = \frac{\sin(\kappa\Delta x)}{\Delta x}. \quad (3.4)$$

The real part of the modified wavenumber is shown in Figure 3.1 for various spatial discretization schemes. Its real part represents the dispersion of the computed derivative compared to the actual one, whereas a non-zero imaginary part of  $\kappa'$  indicates that dissipation errors are introduced in the derivative. Figure 3.1 shows that the numerical derivative is quite accurate for small wavenumbers; high wavenumbers fluctuations, however, become poorly resolved as the wavenumber increases.

If one wants to resolve higher wavenumbers more accurately this implies that either low-order schemes must be used on very fine meshes, or that higher-order schemes are required on coarser meshes. High-order schemes are more expensive, in terms of computational cost than lower-order ones, but the increase in accuracy and possible savings in computer storage they afford, for a given mesh, often justifies their use.

Consider the numerical solution of the Burgers equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \quad (3.5)$$

with the velocity  $u(x, t) > 0$  and  $\nu$  the viscosity. By using a crude two-point first-order upwind finite-differencing scheme for the first-order derivative, and a three-point central finite-differencing scheme for the second-order derivative, one actually solves a modified equation of

the form [6]:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + (\nu + \nu_{\text{num}}) \frac{\partial^2 u}{\partial x^2} = \mathcal{O}(\Delta^2), \quad (3.6)$$

In the above equation,  $\nu_{\text{num}} = \frac{1}{2}u\Delta$  is called the numerical viscosity. The spatial discretization scheme essentially acts as a lowpass filter, dependent on the wavenumber and the mesh spacing. Since the spatial discretization scheme is dependent on the grid, the implicit filter does not allow for grid independent LES.

With this approach to spatial filtering there is another issue of consistency. While it is true that numerical differentiation acts as a low-pass filter, the associated filter acts only in the spatial direction in which the derivative is taken. There is no way to derive the discrete equations through the application of a single three-dimensional filter [23]. Considering this ambiguity in the definition of the filter, it is nearly impossible to make detailed comparisons of LES results with filtered experimental data.

### 3.1.1 Smagorinsky Filter

It is shown from equations (2.17) and (2.25)-(2.27) that the Smagorinsky model equally acts as an extra viscosity which as pointed out above acts as a low pass filter. Simplifying the Smagorinsky model, one can write (2.25) as:

$$\nu_T = l_s^2 \left\langle \bar{S}^2 \right\rangle^{1/2}. \quad (3.7)$$

This is a non-random and uniform eddy viscosity so that it can be added to the viscosity. Using the definition for the Kolmogorov scale (2.2), one can now write an effective Kolmogorov length scale:

$$\bar{\eta} \equiv \left( \frac{(\nu + \nu_T)^3}{\varepsilon} \right)^{1/4} \quad (3.8)$$

Pope [6] proposed a model for the kinetic energy spectrum of homogeneous isotropic turbulence of the form

$$E(\kappa) = C\varepsilon^{2/3}\kappa^{-5/3}f_L(\kappa L)f_\eta(\kappa\eta). \quad (3.9)$$

See Pope [6] for its specifications. When applying the effective Kolmogorov length scale to this model one gets a model for the *Smagorinsky filtered* energy spectrum function:

$$\bar{E}(\kappa) = C\varepsilon^{2/3}\kappa^{-5/3}f_L(\kappa L)f_\eta(\kappa\bar{\eta}). \quad (3.10)$$

The transfer function of this Smagorinsky filter is then defined as:

$$\hat{G}(\kappa) = \left( \frac{\bar{E}(\kappa)}{E(\kappa)} \right)^{1/2}. \quad (3.11)$$

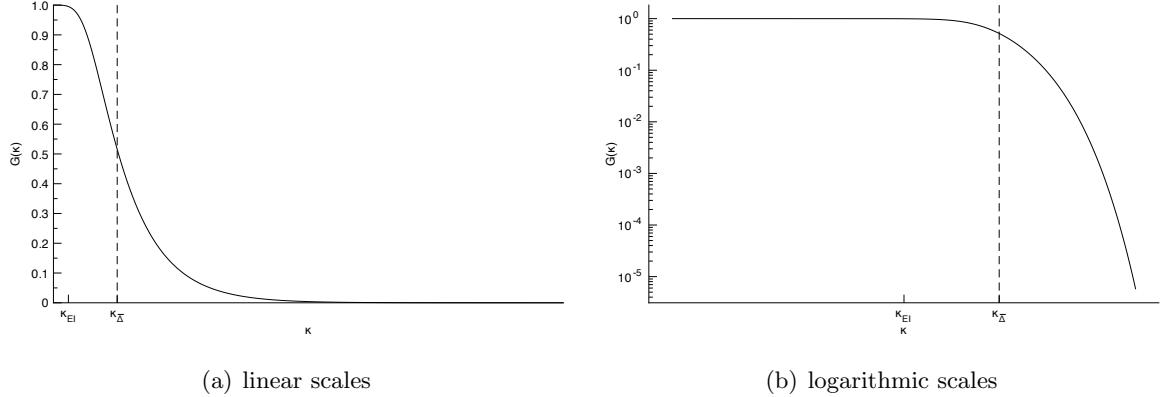


Figure 3.2: Transfer function of the Smagorinsky filter

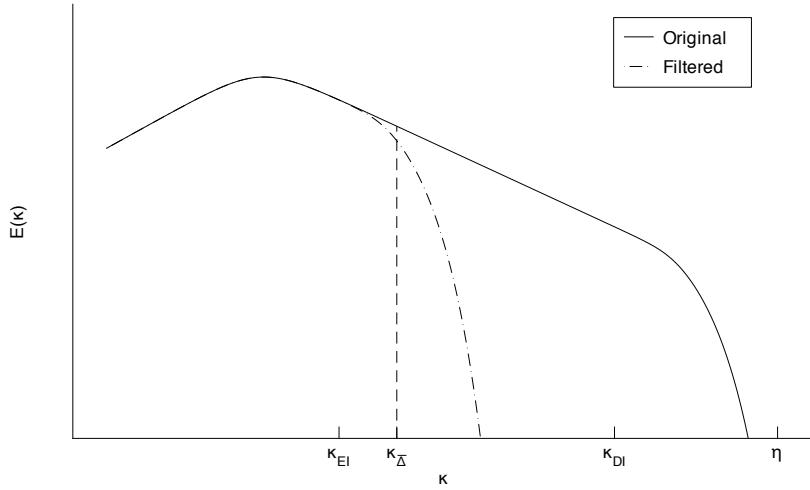


Figure 3.3: Smagorinsky filter applied to the kinetic energyspectrum function

Taking the filter in the inertial subrange such that the mixing length  $l_s$  is much larger than the Kolmogorov scale ( $l_s \approx \bar{\eta} \gg \eta$ ), the implied Smagorinsky filter then has the transfer function [6]

$$\hat{G}(\kappa) = \exp\left(-\frac{1}{2}\beta\left([(kl_s)^4 + c_\eta^4]^{1/4} - c_\eta\right)\right), \quad (3.12)$$

with the coefficients defined by Pope in [6]. This function is shown in Figure 3.2. The filtered energy spectrum function is shown in Figure 3.3.

This shows that by applying a Smagorinsky based model and without the definition of an explicit filter one still has some sort of control over the filter cut-off wavenumber by changing the length scale in the Smagorinsky model.

## 3.2 Explicit Spatial Filtering

The difficulty associated with implicit filtering can be alleviated by performing an explicit filtering operation as a part of the solution process. With explicit filtering the energy in the high frequency portion of the spectrum is damped and it becomes possible to reduce or eliminate the various sources of numerical error that dominate this frequency range [23]. Explicit filtering reduces the effective grid resolution of the simulation but allows the filter size to be chosen independently from the mesh spacing. The form of the filter is also known exactly. This allows the direct comparison of the LES solutions to experimental or DNS data.

### 3.2.1 Properties

An explicit filter can be defined in physical space characterized by a cut-off length or filter width,  $\bar{\Delta}$ , and a filter kernel  $G$ . A convolution filter is defined in physical space symbolically as

$$\bar{\phi}(\mathbf{x}) = [G \star \phi](\mathbf{x}). \quad (3.13)$$

In spectral space, the convolution operator  $\star$  transforms into a multiplication:

$$\hat{\bar{\phi}}(\boldsymbol{\kappa}) = \hat{G}(\boldsymbol{\kappa}) \cdot \hat{\phi}(\boldsymbol{\kappa}) \quad (3.14)$$

The above equation shows that the Fourier transform of the filter kernel  $\hat{G}$  represents the transfer function of the filter, as it is the ratio  $\frac{\hat{\bar{\phi}}(\boldsymbol{\kappa})}{\hat{\phi}(\boldsymbol{\kappa})}$ . While it is very straightforward to filter in spectral space (via multiplication), in physical space a convolution is more elaborate to calculate:

$$\bar{\phi}(\mathbf{x}) = \int_{-\infty}^{+\infty} \phi(\mathbf{x}') G(\mathbf{x} - \mathbf{x}'; \bar{\Delta}) d\mathbf{x}' \quad (3.15)$$

The filter kernel  $G(\mathbf{x} - \mathbf{x}'; \bar{\Delta})$  can be defined fully three-dimensional or as a product of three one-dimensional filter kernels:

$$G(\mathbf{x} - \mathbf{x}'; \bar{\Delta}) = \prod_{i=1}^3 G_i(x_i - x'_i; \bar{\Delta}_i) \quad (3.16)$$

This makes it easy to construct a filter in one dimension and extending it to multiple dimensions.

In order to be able to manipulate the Navier-Stokes equations after applying a filter, we require that filters possess the following three properties [24]:

1. Preserves constant values:

$$\bar{a} = a \quad (3.17)$$

Since a constant has only content of the frequency  $\kappa = 0$ , this means that the value of the transfer function must satisfy

$$\hat{G}(0) = 1 \quad (3.18)$$

or represented in physical space:

$$\int_{-\infty}^{+\infty} G(\mathbf{x} - \mathbf{x}'; \bar{\Delta}) d\mathbf{x}' = 1 \quad (3.19)$$

2. Commutes with addition:

$$\overline{\phi + \psi} = \overline{\phi} + \overline{\psi} \quad (3.20)$$

3. Commutes with differentiation:

$$\overline{\frac{d\phi}{dx_i}} = \frac{d\overline{\phi}}{dx_i} \quad (3.21)$$

Property 1 is no problem as the filter can easily be constructed to satisfy this property. Property 2 is automatically satisfied independently of the choice of the filter kernel  $G$ . Property 3 is generally not satisfied, but as will be described in Section 3.2.2, it is possible to construct the filter kernel so that this property is satisfied to a desired order of accuracy.

### 3.2.2 Commutation Error in One Dimension

In Chapter 2, explicit filtering was briefly mentioned as a possible solution to many problems such as aliasing errors and commutation errors. The objective of this research is to consider the use of explicit filters for controlling commutation errors.

The definition of the commutation error was defined in (2.43). Marsden [25] proposed a means to formulate the commutation error as a function of the filter function and filter width. First one takes the definition of the filter operation given in (3.15) but written specifically for the one-dimensional case:

$$\bar{\phi}(x) = \frac{1}{\bar{\Delta}(x)} \int_a^b \phi(x') G\left(\frac{x - x'}{\bar{\Delta}(x)}\right) dx'. \quad (3.22)$$

With the change of variables  $\eta = \frac{x - x'}{\bar{\Delta}(x)}$ , Eq. (3.22) becomes

$$\bar{\phi}(x) = \int_{\frac{x-b}{\bar{\Delta}(x)}}^{\frac{x-a}{\bar{\Delta}(x)}} \phi(x - \bar{\Delta}(x)\eta) G(\eta) d\eta. \quad (3.23)$$

Then taking the Taylor series expansion of  $\phi(x - \bar{\Delta}(x)\eta)$  about  $x$  yields

$$\phi(x - \bar{\Delta}(x)\eta) = \phi(x) + \sum_{l=1}^{\infty} \frac{d^l \phi}{dx^l} \Big|_x \frac{(x - \bar{\Delta}(x)\eta - x)^l}{l!} \quad (3.24)$$

$$= \phi(x) + \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \bar{\Delta}(x)^l \eta^l \frac{d^l \phi}{dx^l} \Big|_x. \quad (3.25)$$

Substituting (3.25) into (3.23) gives

$$\bar{\phi}(x) = \int_{\frac{x-b}{\Delta(x)}}^{\frac{x-a}{\Delta(x)}} \phi(x) G(\eta) d\eta + \int_{\frac{x-b}{\Delta(x)}}^{\frac{x-a}{\Delta(x)}} \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \bar{\Delta}(x)^l \eta^l \left. \frac{d^l \phi}{dx^l} \right|_x G(\eta) d\eta. \quad (3.26)$$

With the filter property that a filtered constant remains the same constant (see Eq. (3.17)), and defining the  $k^{\text{th}}$  filter moment as

$$M^k(x) = \int_{\frac{x-b}{\Delta(x)}}^{\frac{x-a}{\Delta(x)}} \eta^k G(\eta) d\eta, \quad (3.27)$$

equation (3.26) can be written as:

$$\bar{\phi}(x) = \phi(x) + \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \bar{\Delta}(x)^l \left. \frac{d^l \phi}{dx^l} \right|_x M^l(x). \quad (3.28)$$

Taking the derivative of this equation leads to the second term in the definition of the commutation error given in (2.43):

$$\frac{d\bar{\phi}(x)}{dx} = \frac{d\phi(x)}{dx} + \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \frac{d}{dx} \left( \bar{\Delta}(x)^l \left. \frac{d^l \phi}{dx^l} \right|_x M^l(x) \right). \quad (3.29)$$

An expression for the first term of (2.43) can be obtained by a similar approach by taking a Taylor series expansion of  $\frac{d\phi}{dx}$  and applying the filtering operation. This term yields

$$\overline{\frac{d\phi(x)}{dx}} = \frac{d\phi(x)}{dx} + \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \bar{\Delta}(x)^l \left. \frac{d^{l+1} \phi}{dx^{l+1}} \right|_x M^l(x). \quad (3.30)$$

The local commutation error can then be calculated by combining (3.29) and (3.30) to obtain

$$\left[ \frac{d\phi}{dx} \right] = \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \frac{d}{dx} \left( \bar{\Delta}(x)^l M^l(x) \right) \left. \frac{d^l \phi}{dx^l} \right|_x. \quad (3.31)$$

This result is very important as it illustrates a number of important things. It indicates that on a uniform grid with a uniform filter width the term  $(\bar{\Delta}(x)^l M^l(x))$  is a constant such that  $\frac{d}{dx} (\bar{\Delta}(x)^l M^l(x)) = 0$  and the commutation error vanishes. However, as is indicated earlier in the thesis, for inhomogeneous turbulent flows, the minimum size of eddies that needs to be resolved varies throughout the flow. Thus the filtering operation should in general be performed with a variable filter width and the commutation error will therefore not be zero. Most importantly (3.31) shows us a means to construct explicit filters that commute to a desired order of accuracy  $n$ . This is possible by choosing the filter moments  $M^l(x)$  in the following way:

$$M^l(x) = \begin{cases} 1, & l = 0; \\ 0, & l = 1, \dots, n-1; \\ \text{exists}, & l \geq n. \end{cases} \quad (3.32)$$

The first line of (3.32) is nothing less than the filter property that preserves a constant value. The second line assures that  $\frac{d}{dx} (\bar{\Delta}(x)^l M^l(x)) = 0$  for values of  $l < n$  in Eq. (3.31) such that the only remaining terms in the commutation error are of order  $\bar{\Delta}(x)^n$ :

$$\left[ \frac{d\phi}{dx} \right] = \mathcal{O}(\bar{\Delta}(x)^n), \quad (3.33)$$

where  $n$  is the commutation order of filter.

Extension of the preceding analysis to three dimensions is quite straightforward, [25]. The local commutation error has the form

$$\left[ \frac{d\phi}{dx_1} \right] = \sum_{l=1}^{\infty} \sum_{i+j+k=l} \frac{(-1)^l}{l!} \alpha_{ijk}^l \frac{d}{dx_1} \left( \bar{\Delta}_1(\mathbf{x})^i \bar{\Delta}_2(\mathbf{x})^j \bar{\Delta}_3(\mathbf{x})^k M^{ijk}(\mathbf{x}) \right) \mathcal{D}_{x_1}^i \mathcal{D}_{x_2}^j \mathcal{D}_{x_3}^k \phi(\mathbf{x}). \quad (3.34)$$

In this equation  $\mathcal{D}_{x_1}^i$  is the operator  $\frac{d^i}{dx_1^i}$ ,  $\alpha_{ijk}^l$  is the trinomial coefficients of the expansion given by

$$(a+b+c)^l = \sum_{i+j+k=l} \alpha_{ijk}^l a^i b^j c^k, \quad (3.35)$$

and  $M^{ijk}(\mathbf{x})$  are the three-dimensional filter moments defined in a similar manner as above

$$M^{ijk} = \int_{\Psi} \eta_1^i \eta_2^j \eta_3^k G(\boldsymbol{\eta}) d^3 \boldsymbol{\eta}. \quad (3.36)$$

The desired properties of the filter moments are as in the one-dimensional case

$$M^{ijk}(x) = \begin{cases} 1, & i, j, k = 0; \\ 0, & i + j + k = 1, \dots, n - 1; \\ \text{exists}, & i + j + k \geq n. \end{cases} \quad (3.37)$$

The local commutation error in three dimensions is then given by

$$\left[ \frac{d\phi}{dx_1} \right] = O(\bar{\Delta}_1(\mathbf{x})^i \bar{\Delta}_2(\mathbf{x})^j \bar{\Delta}_3(\mathbf{x})^k), \quad i + j + k = n. \quad (3.38)$$

### 3.2.3 Approaches to Explicit Spatial Filtering

It is interesting to compare different approaches to explicit filtering. The main idea in explicit filtering is to perform filtering with smooth filters in such way that excessive filtering of the results from previous time levels is avoided. This can be clarified by applying one step of an explicit Euler time step to the momentum equation (2.37):

$$\overline{\rho u_i}^{n+1} = \overline{\rho u_i}^n + \Delta t^n \left( -\frac{\partial(\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} + \frac{\partial \check{s}_{ij}}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} \right) \quad (3.39)$$

$$= \overline{\rho u_i}^n + \Delta t^n \bar{R}^n, \quad (3.40)$$

where the superscript refers to time levels,  $\Delta t^n$  the time step and  $\bar{R}^n$  the rate of change of the flowfield on the  $n^{\text{th}}$  time level. If at every time step  $\overline{\rho u_i}^{n+1}$  would be filtered explicitly, then the terms  $\overline{\rho u_i}^n$  and  $\bar{R}^n$  will be filtered multiple times over time. This leads to excessive damping of the filtered flow solution, [26]:

$$\overline{\rho u_i}^{n+1} = \overline{\rho u_i}^n + \Delta t^n \overline{\bar{R}^n} \quad (3.41)$$

$$= \overline{\overline{\rho u_i}^{n-1}} + \Delta t^{n-1} \overline{\overline{\bar{R}^{n-1}}} + \Delta t^n \overline{\bar{R}^n}. \quad (3.42)$$

In general, repeated application of the same filter implies an increased filter width and thus a severe loss of spectral information of the velocity field.

A first approach to explicit filtering is to limit the generation of higher frequencies in  $\overline{\rho u_i}^{n+1}$  by filtering the residual  $\bar{R}^n$  every time step. This means that both convective and dissipative terms in (3.39) are filtered. As an alternative approach, the idea is to selectively limit the introduction of high frequency content for the terms in (3.39). The only terms that inject higher frequency content in the solution are the non-linear convective terms and the SFS terms. Lund [26] suggested that by filtering only these terms no additional frequency content is added. The non-linear term from (3.39) can be decomposed by applying an explicit spatial filtering procedure to obtain

$$\overline{\rho u_i u_j} = \overline{\rho \tilde{u}_i \tilde{u}_j} + \underbrace{(\overline{\rho u_i u_j} - \overline{\rho \tilde{u}_i \tilde{u}_j})}_{\tau'_{ij}} \quad (3.43)$$

Equation (3.39) then becomes

$$\overline{\rho u_i}^{n+1} = \overline{\rho u_i}^n + \Delta t^n \left( -\frac{\partial (\overline{\rho \tilde{u}_i \tilde{u}_j})}{\partial x_j} + \frac{\partial \check{\sigma}_{ij}}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau'_{ij}}{\partial x_j} \right). \quad (3.44)$$

It is noted that additional filtering is only applied to the non-linear terms in the expression above and an appropriate modelling for the modified SFS stress,  $\tau'_{ij}$ , is required to achieve an explicit filtering of the velocity field for all time. In a recent study, Brandt [27] found that when the Smagorinsky SFS-model is applied, filtering  $\tau_{ij}$  to obtain  $\tau'_{ij}$  has no effect on the results; but if scale-similarity and mixed models are applied, this SFS term could also introduce additional higher frequencies to the solution [28] and would require explicit filtering.

Brandt [29] has made a comparison between both approaches to explicit filtering (filtering the residual vs. filtering the non-linear terms as described above). He looked at the one-dimensional streamwise energy spectrum of a fully developed turbulent channel flow between two infinite parallel walls. He found that both approaches to explicit filtering only result in small differences, and thus the filtering of the non-linear term, which generates the high frequency components in the resolved flowfield, seems to be the essential part of explicit filtering. More information on the implementation of explicit filtering techniques is given in Section 6.1.2.

While the decomposition used in (3.43) has several desirable properties from the point of view of explicit filtering, there is also a deleterious side effect involved. The resulting equation (3.44) is no longer Galilean invariant in general [26]. The error is proportional to the difference between the singly and doubly filtered velocity. For a Fourier cut-off filter, this difference will be zero and Galilean invariance will be preserved. In the general case however this difference will not vanish. Lund [26] formulated that the spectral content of this error is proportional to  $\hat{G}(\kappa)(1 - \hat{G}(\kappa))$ . This fact implies that it is possible to minimize the error by constructing the explicit filter to have a transfer function as close as possible to a Fourier cut-off filter.

### 3.3 Explicit Analytical Filters

Several explicit filters can be defined. The most widely used or known explicit filters are the top-hat filter, the Gaussian filter and the spectral or sharp Fourier cut-off filter. They are convolution filters characterized by a cut-off length or filter width  $\bar{\Delta}$ . In what follows, the mathematical properties of these classical analytical filters are now briefly summarized.

#### 3.3.1 Three Classical Analytical Filters

##### Spectral Fourier cut-off

The spectral Fourier cut-off filter is easily defined in spectral space. It essentially retains all solution frequency content up to a wavenumber  $\kappa_{\bar{\Delta}} = \frac{\pi}{\bar{\Delta}}$ . All the frequency content above this wavenumber is removed and/or set to zero:

$$\hat{G}(\kappa) = \begin{cases} 1 & \text{if } |\kappa| \leq \kappa_{\bar{\Delta}} \\ 0 & \text{otherwise} \end{cases} \quad (3.45)$$

In physical space the transfer function translates to the convolution kernel

$$G(x) = \frac{\sin(x\kappa_{\bar{\Delta}})}{x\kappa_{\bar{\Delta}}} \quad (3.46)$$

The convolution kernel  $G$  and the transfer function  $\hat{G}$  are represented in Figure 3.4 and 3.7 respectively.

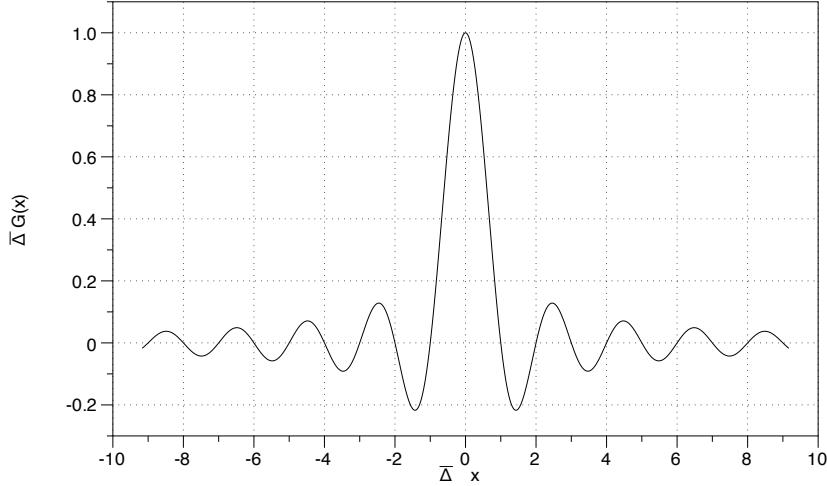


Figure 3.4: Sharp Fourier cut-off filter. Convolution kernel  $G$  in physical space.

### Top-Hat filter

The top-hat filter can be seen as taking an average value over the domain bounded by  $\bar{\Delta}/2$ :

$$G(x) = \begin{cases} 1/\bar{\Delta} & \text{if } |x| \leq \bar{\Delta}/2 \\ 0 & \text{otherwise} \end{cases} \quad (3.47)$$

In the spectral domain, the transfer function is then given by

$$\hat{G}(\kappa) = \frac{\sin(\kappa\bar{\Delta}/2)}{\kappa\bar{\Delta}/2} \quad (3.48)$$

Graphs of the convolution kernel  $G$  and the transfer function  $\hat{G}$  are depicted in Figure 3.5 and 3.7, respectively.

### Gaussian Filter

The convolution kernel for the Gaussian filter has the form of a Gaussian function defined by

$$G(x) = \sqrt{\frac{\gamma}{\pi\bar{\Delta}^2}} \exp\left(-\frac{\gamma x^2}{\bar{\Delta}^2}\right), \quad (3.49)$$

in which  $\gamma$  is a constant generally taken to be equal to 6. In the spectral domain, the transfer function of this filter is given by

$$\hat{G}(\kappa) = \exp\left(-\frac{\bar{\Delta}^2\kappa^2}{4\gamma}\right). \quad (3.50)$$

The convolution kernel  $G$  and the transfer function  $\hat{G}$  are represented in Figure 3.6 and 3.7 respectively.

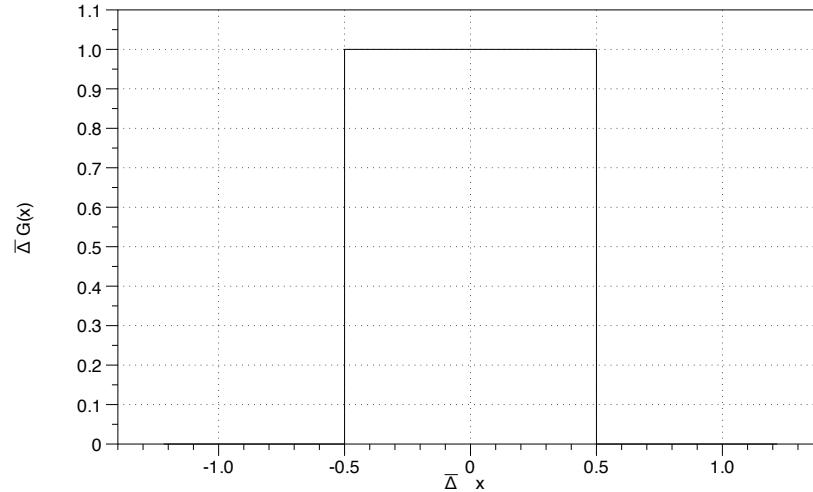


Figure 3.5: Top-hat filter. Convolution kernel  $G$  in physical space normalized by  $\bar{\Delta}$ .

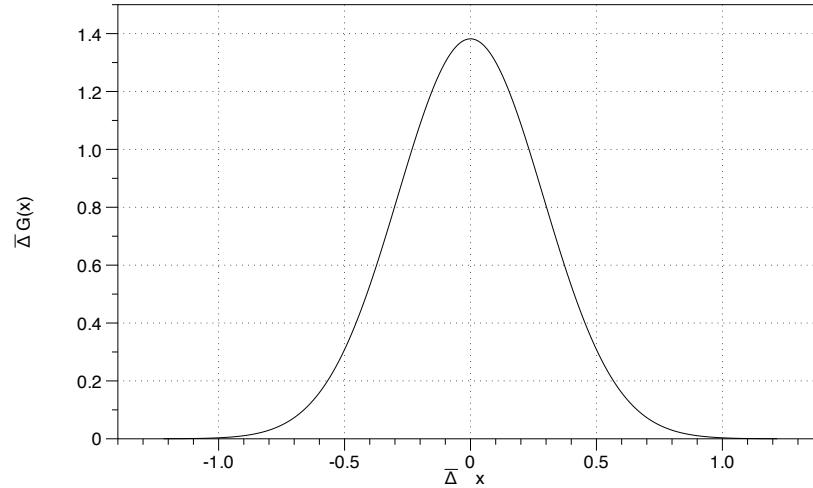


Figure 3.6: Gaussian filter. Convolution kernel  $G$  in physical space normalized by  $\bar{\Delta}$ .

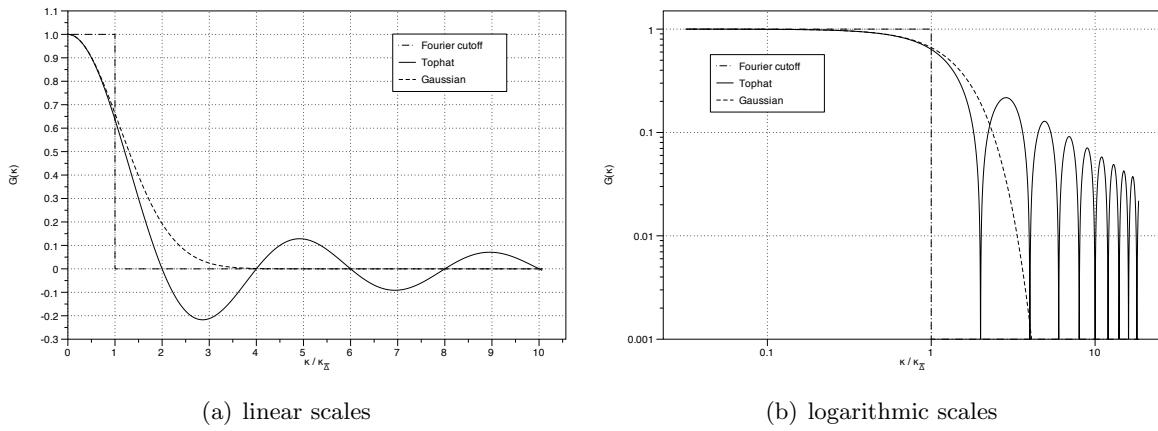


Figure 3.7: Associated transfer functions to the Fourier cut-off, top-hat and Gaussian filters.

### Analysis of the Three Classical Analytical Filters

Application of the sharp Fourier cut-off filter is very straightforward in the spectral domain and this filtering approach is well suited to a LES solution method based on a spectral discretization procedure. In practice, this is very difficult due to the need to deal with complex geometries and boundaries. This means that the convolution will generally have to be performed in physical space. The more localized the filter kernel is, the better because then the integration limits for the convolution integral can be reduced to ranges where  $G \neq 0$ . The top-hat filter is the most compact. It requires only an integration range from  $-\frac{\bar{\Delta}}{2}$  to  $\frac{\bar{\Delta}}{2}$ , as can also be observed in Figure 3.5. Figure 3.6 shows that a range of  $-\bar{\Delta}$  to  $\bar{\Delta}$  is sufficient for the Gaussian filter, making it half as compact as the top-hat filter. Coming back to the sharp Fourier cut-off, it is now clear through Figure 3.4 that this filter is not compact at all in comparison with the top-hat filter and the Gaussian filter.

It is now interesting to check the order of the commutation error for these filters. In the light of Section 3.2.2, this comes down to calculating the filter moments of the convolution kernel. Due to the non-localized nature of the convolution kernel of the sharp Fourier cut-off filter, the integral in (3.27) is not convergent and (3.28) is a divergent series. For the other two analytical filters the filter moments can be computed and are given in Table 3.1.

$M^n$	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$
Top-hat	1	0	$\frac{\bar{\Delta}^2}{12}$	0	$\frac{\bar{\Delta}^4}{80}$	0	$\frac{\bar{\Delta}^6}{448}$	0	$\frac{\bar{\Delta}^8}{2304}$
Gaussian	1	0	$\frac{\bar{\Delta}^2}{12}$	0	$\frac{\bar{\Delta}^4}{48}$	0	$\frac{5\bar{\Delta}^6}{576}$	0	$\frac{35\bar{\Delta}^8}{6912}$

Table 3.1: Filter moments for the top-hat filter and the Gaussian filter up to the 8<sup>th</sup> order.

The first non-zero moment with  $n > 0$  for both filters is of second-order. Using the methodology from Section 3.2.2 reveals that both filters commute only to second-order.

#### 3.3.2 A Higher Order Commuting Analytical Filter

Van der Ven [30] constructed a class of explicit filters that commute up to a desired order by analyzing the commutation error in a different way than described in 3.2.2. He found that the filter convolution kernel has to satisfy the following property for the commutation error to be of order  $n$ :

$$G + \eta G' = a G^{(n)}, \quad n > 1. \quad (3.51)$$

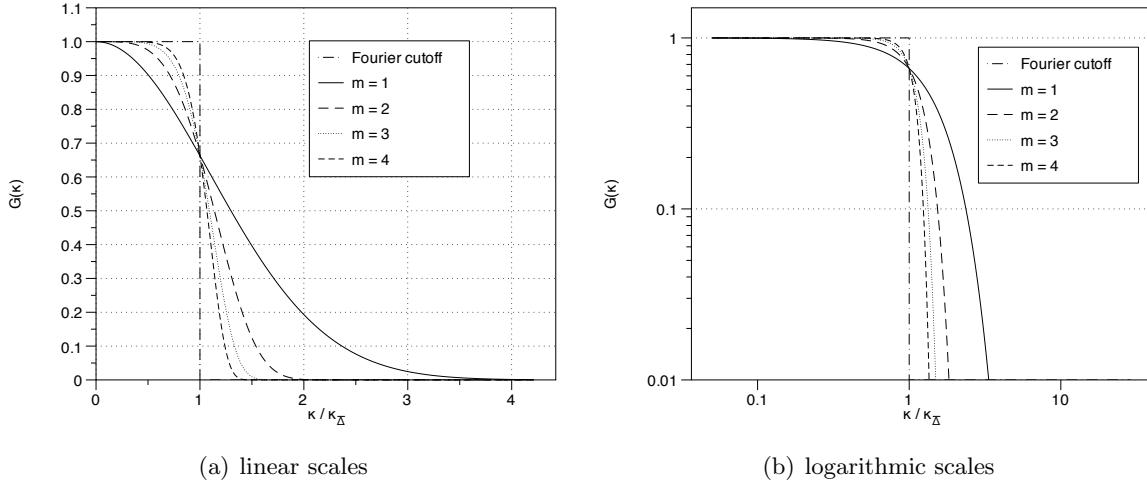


Figure 3.8: Associated transfer functions to the Van der Ven filters with  $m = 1, \dots, 4$ .

In the equation above,  $a$  is a real constant and  $G^{(n)}$  the  $n^{\text{th}}$  derivative of the kernel  $G$ . The Fourier transform of this problem is simple and the transfer function  $\hat{G}$  is given by

$$\hat{G}(\kappa) = \exp\left(-\frac{a\kappa^n}{n}\kappa^n\right). \quad (3.52)$$

The symmetry of  $\hat{G}$  requires that  $n = 2m$  is even:

$$\hat{G}(\kappa) = \exp\left(-\frac{a}{2m}\kappa^{2m}\right). \quad (3.53)$$

One can now solve (3.53) for  $a$  so that  $\hat{G}(\kappa_{\Delta}) = \exp\left(-\frac{\pi^2}{4\gamma}\right)$  as is the case for the Gaussian filter:

$$a = \frac{\pi^2}{4\gamma} 2m \kappa_{\Delta}^{-2m} \quad (3.54)$$

The transfer functions associated with this filter is shown in Figure 3.8 for several values of  $m$ . Note that this filter with  $m = 1$  is exactly the Gaussian filter. By increasing  $m$ , the order of commutation order increases and the filter transfer function resembles closer to the sharp Fourier cut-off filter.

It is important to note that this analysis is valid only for infinite domains, because when the boundaries of the fluid domain are included they bring out additional error terms with which it is no longer possible to be sure of the order of the commutation error [30].

### 3.4 Explicit Discrete Filters

Due to the lack of a straightforward and robust filtering procedure for inhomogeneous flows, most LES performed to date have not made use of explicit filters. The analytical filters described

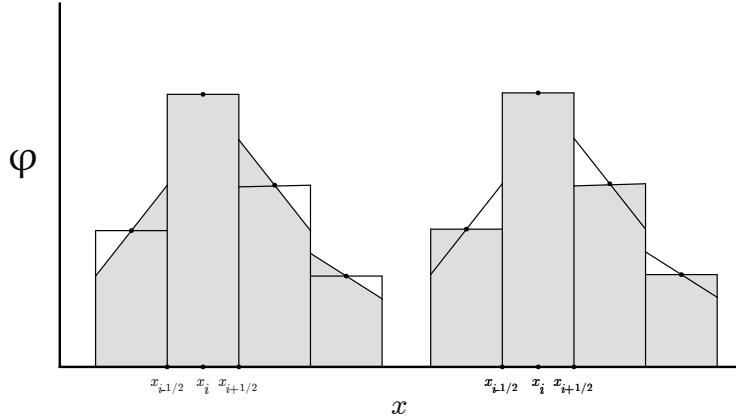


Figure 3.9: Riemann summation of the cell centred values is an exact integration-scheme for a piece-wise linear finite volume scheme.

above require a numerical integration procedure to compute the convolution integral. For a second-order finite-volume scheme, the solution is seen as piece-wise linear in each cell and the cell centre thus corresponds to the cell average. Performing the convolution integral for the top-hat filter can then in principle be calculated by approximating the integral by a Riemann summation of the cell-centred values on a compact bounded domain since the filter kernel  $G$  is constant within this domain:

$$\bar{\phi}(x) = \frac{1}{\Delta} \int_{x-\frac{\Delta}{2}}^{x-\frac{\Delta}{2}} \phi(x') dx' \quad (3.55)$$

$$= \frac{1}{\Delta} \sum_i \phi(x_i) (x_{i+1/2} - x_{i-1/2}) \quad \forall i \text{ for which } x_i \in \left[ x - \frac{\Delta}{2}, x + \frac{\Delta}{2} \right], \quad (3.56)$$

where  $x_{i+1/2}$  and  $x_{i-1/2}$  are respectively the right and left cell interface. Figure 3.9 schematically represents how for a second-order finite-volume scheme the integral in (3.55) can be easily calculated. In (3.56) every cell-centred value is essentially multiplied with an appropriate weight defined by the cell geometry.

This is also the idea for discrete filters since calculating the convolution integral numerically is very computationally expensive. The objective, however, is to obtain a formulation that is independent of the finite-volume scheme and allows for the specification of the desired order of the commutation error. Vasilyev et al. [31] proposed a form for the filter convolution kernel  $G$ , so that instead of calculating a convolution integral, a simple summation similar to (3.56) suffices. In one dimension, this filter kernel can be defined as

$$G^i(x_i - x') = \sum_{l=-K_i}^{L_i} w_l^i \delta(x' - x_{i+l}), \quad (3.57)$$

where the superscript  $i$  denotes that  $G$  and the weight factors  $w_l$  are specific for the index  $i$ ,  $x_i$  is the location of the filtered quantity, and  $\delta(x)$  is the dirac-function with the property

$$\int_a^b \phi(x')\delta(x' - x)dx' = \phi(x), \quad \text{with } x \in [a, b]. \quad (3.58)$$

Substituting (3.57) in (3.22) gives the following definition for a discrete filter with  $\phi_i = \phi(x_i)$ :

$$\bar{\phi}_i = \sum_{l=-K_i}^{L_i} w_l^i \phi_{i+l}. \quad (3.59)$$

To obtain the transfer function of the discrete filter one simply takes the Fourier transform of (3.57):

$$\hat{G}^i(\kappa) = \sum_{l=-K_i}^{L_i} w_l^i e^{-i\kappa\Delta x_{il}}, \quad (3.60)$$

with  $\Delta x_{il} = (x_l - x_i)$ . The filter moments of a discrete filter easily follow through substitution of (3.57) in (3.27):

$$M^k(x)|_i = \sum_{l=-K_i}^{L_i} x^k w_l^i. \quad (3.61)$$

In three dimensions, this is easily extended and (3.59) becomes

$$\bar{\phi}_{ijk} = \sum_{l=-K_i}^{L_i} \sum_{m=-K_j}^{L_j} \sum_{n=-K_k}^{L_k} w_{lmn}^{ijk} \phi_{i+l,j+m,k+n}, \quad (3.62)$$

Eq. (3.60) then becomes

$$\hat{G}^{ijk}(\kappa) = \sum_{l=-K_i}^{L_i} \sum_{m=-K_j}^{L_j} \sum_{n=-K_k}^{L_k} w_{lmn}^{ijk} e^{-i\kappa\Delta r_{lmn}^{ijk}}, \quad (3.63)$$

with  $\Delta r_{lmn}^{ijk} = (\mathbf{x}_{lmn} - \mathbf{x}_{ijk})$ ; and (3.61) has the form

$$M^{qrs}(\mathbf{x})|_{ijk} = \sum_{l=-K_i}^{L_i} \sum_{m=-K_j}^{L_j} \sum_{n=-K_k}^{L_k} x^q y^r z^s w_{lmn}^{ijk} \quad (3.64)$$

Lund [26] discusses many ways to define the filter width for a variety of spatial filters. One way is to assume that the filter width is equal to the location of the first zero crossing of the filter kernel  $G$ . Another way is to use the second filter moment  $M^2(x)$ . However since this is not guaranteed to be different from zero and since it is shown that the filter moments have to be zero to obtain a higher order of the commutation error, this method is rejected herein. The

third and preferred method is looking at the transfer function of the filter and taking the filter width to correspond to the value where  $G(\kappa_{\bar{\Delta}}) = 0.5$ . The filter width is then given by  $\frac{2\pi}{\kappa_{\bar{\Delta}}}$ . Note that previously for the analytical filters  $\kappa_{\bar{\Delta}}$  was defined such that  $G(\kappa_{\bar{\Delta}}) = 0.6628$  with  $\gamma = 6$ . One could argue to now change either  $\gamma$  for analytical filters or the definition of the filter width for discrete filters so that both give the same value of  $G(\kappa_{\bar{\Delta}})$ . Taking into consideration that some analytical filters such as the top-hat filter might not have a dependence on  $\gamma$ , the latter is suggested here.

### 3.4.1 Linear Constraints Filter

Vasilyev [31] introduced a class of commutative filters that offers both control over the shape of the filter as well as the filter width by solving a linear system of equations for nonuniform structured grids. The filter is a one-dimensional filter which can easily be extended to three dimensions using (3.16). The filter also allows for the use of asymmetric stencils. A major drawback of this class of filters is that the filtering occurs in computational space. This implies that there is a mapping of the nonuniform grid to a uniform grid with spacings  $\Delta_1, \Delta_2, \Delta_3$  in the corresponding directions. The transformation is given by  $\boldsymbol{\xi} = \mathbf{f}(\mathbf{x})$ , where  $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^T$ . This generally restricts the application of this class of discrete filters to structured meshes.

Given an arbitrary function  $\phi(\mathbf{x})$  one obtains through the transformation a function  $\psi(\boldsymbol{\xi}) = \phi(\mathbf{x})$ . Using (3.16) the filtering operation in computational space in three dimensions is defined by

$$\bar{\psi}(\boldsymbol{\xi}) = \int \prod_{i=1}^3 G(\xi_i - \eta_i; \Delta_i) \psi(\boldsymbol{\eta}) d\boldsymbol{\eta}. \quad (3.65)$$

The filtering operation in physical space can be written as

$$\bar{\phi}(\mathbf{x}) = \int \prod_{i=1}^3 G(f_i(\mathbf{x}) - f_i(\mathbf{y}); \Delta_i) \phi(\mathbf{y}) J(\mathbf{y}) d\mathbf{y}, \quad (3.66)$$

with  $J(\mathbf{x})$  the determinant of the Jacobian  $\left. \frac{\partial(\xi_1, \xi_2, \xi_3)}{\partial(x_1, x_2, x_3)} \right|_{\mathbf{x}}$ . It should be noted that the filter kernel in physical space is the same as in computational space, but that the transformation occurs through the Jacobian:  $d\boldsymbol{\eta} = J(\mathbf{y})d\mathbf{y}$ . Since  $\bar{\psi}(\boldsymbol{\xi}) = \bar{\phi}(\mathbf{x})$ , it is more practical to use (3.66) for construction and application of the filter. This is because the structured grid is characterized by indices  $i, j, k$  which can be viewed as the uniform grid with grid spacing  $\Delta = 1$ . It is then straightforward to apply (3.66).

Through application of three one-dimensional filters the weight factor  $w_{lmn}$  in (3.62) and (3.63) can be found by multiplication of the one-dimensional weight factors:

$$w_{lmn} = w_l w_m w_n. \quad (3.67)$$

It now suffices to construct each one-dimensional discrete filter in computational space. The idea is to solve a linear system of equations with the weight factors  $w_l$  as unknowns. Eq. (3.61) is linear in these weight factors and could be used in this system. This allows to specify the order of commutation error (see Section 3.2.2). The weight factors should then satisfy

$$\sum_{l=-K_i}^{L_i} w_l^i = 1, \quad (3.68)$$

$$\sum_{l=-K_i}^{L_i} l^m w_l^i = 0, \quad m = 1, \dots, N-1. \quad (3.69)$$

This gives  $N$  constraints on  $w^l$  with  $N$  being the order of the commutation error. Eq. (3.68) already sets  $\hat{G}(0) = 1$ . A desirable property is that  $\hat{G}(\kappa_\Delta) = 0$ . This can be achieved by using (3.60) and  $\Delta x_{il} = l\Delta$  and  $\kappa_\Delta = \frac{\pi}{\Delta}$ . This requirement can then be written as a linear equation in terms of  $w_l$  as

$$\sum_{l=-K_i}^{L_i} (-1)^l w_l^i = 0. \quad (3.70)$$

If this requirement is not met, then the transfer function might not go to zero and the filter will not behave as a low pass filter. Equations (3.68)-(3.70) represent the minimum number of constraints which should be imposed on the filter. This system is only solvable if  $L_i + K_i + 1 = N + 1$ . In other words, a one-dimensional discrete filter requires minimum  $N+1$  points. This however doesn't allow any control over filter width or shape of the filter transfer function. Additional linear constraints can be added for more control over the shape of the transfer function and filter width. Each additional constraint requires an additional point in the stencil. Several possibilities and/or combinations can be formulated and are further discussed in Section 4.1.1.

With the above defined constraints and the chosen additional constraints, a linear system of equations is constructed and solved for the unknowns  $w_l$ . This procedure is carried out for each direction  $i$ ,  $j$  and  $k$ . With the discrete definition for  $G$  in computational space, the filtered variable is then defined as

$$\phi_{ijk} = \sum_{l=-K_i}^{L_i} \sum_{m=-K_j}^{L_j} \sum_{n=-K_k}^{L_k} w_{l,m,n}^{i,j,k} \phi_{i+l,j+m,k+n}, \quad (3.71)$$

The reason why this discrete filter is constructed from one-dimensional filters is that otherwise the unknowns are directly  $w_{l,m,n}$  which are many more in number than the sum of the three one-dimensional unknowns. This would require many more equations to solve. Another alternative is solving a non-linear system of equations for the one-dimensional unknowns, but this is computationally inefficient and does not guarantee the desired behaviour.

### 3.4.2 Least Squares Reconstruction Filter

The linear constraints filter described above can only be used on structured grids. Haselbacher [32] constructed a commutative discrete filter on unstructured grids based on least-squares techniques. As with the linear constraints filter, the key concept in this filter is that the filter weights exhibit vanishing moments. The larger the number of vanishing moments, the higher the order of commutation error. The advantage of this filter is that it is truly constructed in three dimensions and no transformation from physical to computational space is required.

The construction of this filter is very similar to a least-squares gradient reconstruction. A first-order least-squares gradient reconstruction can be defined as follows:

$$\phi_i = \phi_0 + (\nabla\phi)_0 \cdot \Delta\mathbf{r}_{0i}, \quad i = 1, \dots, d_0. \quad (3.72)$$

In this equation  $d_0$  is the number of neighbours of the point  $\mathbf{x}_0$  used in the least-squares gradient reconstruction and  $(\nabla\phi)_0$  is the unknown, sought by the least-squares procedure. The value  $\phi_0$  is the known value of the solution at location  $\mathbf{x}_0$ . Note that  $i$  loops over all the neighbours defined in three dimensions and is not a one-dimensional index. The order of accuracy of (3.72) can be increased by including higher order terms in the Taylor expansion. If  $q$  denotes the degree of the highest derivative included in the Taylor expansion, then the least-squares gradient-reconstruction method gives an order of accuracy  $q - m + 1$  for derivatives of degree  $m \leq q$ . For the gradient the sought derivative is of degree  $m = 1$ , and the order of accuracy equals  $q$ . Hence  $q$  can be called the order of accuracy of the gradient-reconstruction. On symmetric stencils cancellation may result in higher orders of accuracy.

The gradient reconstruction can be turned into a filtering method by modifying (3.72) so that  $\phi_0$  is no longer a point value, but represents an interpolated or filtered value  $\bar{\phi}_0$ , which is also considered to be an unknown.

$$\phi_i = \bar{\phi}_0 + (\nabla\phi)_0 \cdot \Delta\mathbf{r}_{0i}, \quad i = 1, \dots, d_0. \quad (3.73)$$

If  $l$  denotes the degree of the highest derivative included in the Taylor series expansion for this filter, then the order of accuracy for a derivative of degree  $m$  is defined in the same manner as above to be  $l - m + 1$ . For the filter the sought derivative is of order  $m = 0$ , so that the order of accuracy of the filter is  $p = l + 1$ . When using the same Taylor expansion as for the gradient-reconstruction, the order of accuracy of the filter is  $p = q + 1$ .

Equation (3.73) can be written as an overdetermined system of linear equations

$$\mathbf{Ax} = \mathbf{b}, \quad (3.74)$$

where  $\mathbf{A}$ ,  $\mathbf{x}$  and  $\mathbf{b}$  are defined as:

$$\begin{pmatrix} 1 & (x_1 - x_0) & (y_1 - y_0) & (z_1 - z_0) \\ 1 & (x_2 - x_0) & (y_2 - y_0) & (z_2 - z_0) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (x_{d_0} - x_0) & (y_{d_0} - y_0) & (z_{d_0} - z_0) \end{pmatrix} \cdot \begin{pmatrix} \bar{\phi}_0 \\ (\nabla_x \phi)_0 \\ (\nabla_y \phi)_0 \\ (\nabla_z \phi)_0 \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{d_0} \end{pmatrix}. \quad (3.75)$$

Calling  $\mathbf{Z}$  the pseudo-inverse of the matrix  $\mathbf{A}$ ,

$$\mathbf{Z} = \mathbf{A}^\dagger, \quad (3.76)$$

and taking the discrete filter weights to be the elements of the first row of  $\mathbf{Z}$ , then  $\bar{\phi}_0$  can be written in terms of  $\phi_i$ :

$$\bar{\phi}_0 = \sum_{i=1}^{d_0} w_i \phi_i \quad (3.77)$$

This casts this discrete filter in the same form as in (3.59), so that all defined operations for a discrete filter can be applied.

Marsden [25] has shown that discrete filters based on an  $(N-1)^{\text{th}}$  order polynomial construction automatically guarantee an  $N^{\text{th}}$  order commutation error. Since this least-squares procedure is nothing less than a polynomial construction, the order of commutation error can be calculated as

$$N = \min(p-1, q) \quad (3.78)$$

with  $p-1 = l$  and  $q$  the degrees of the highest derivative in the Taylor expansions of respectively the filtering-reconstruction (3.73) and the gradient-reconstruction (3.72). This is one order lower than the prediction by Marsden [25], because the filter weights  $w_i \propto \Delta r_{0i}^{-1}$  [32]. Thus using higher derivatives in (3.72) and (3.73) allows one to achieve higher orders of the commutation error.

The main disadvantage with this filter is that there is no straightforward way to control the filter width and shape of the transfer function. Haselbacher [32] suggests that the spectral behaviour can be improved by adding an extra coefficient which will from this point on be referred to as the relaxation coefficient  $w_0$  :

$$\bar{\phi}_0 = w_0 \phi_0 + (1 - w_0) \sum_{i=1}^{d_0} w_i \phi_i \quad (3.79)$$

This adds a weight to the original  $\phi_0$  and appears to have a big influence on the spectral behaviour (see Section 4.2.1). It is however not possible to define this value as it is different

for every used stencil. To influence the filter width, some success has been achieved by using a weighted least-squares technique. In the case of a weighted least-squares,

$$\mathbf{Z} = (\mathbf{W}\mathbf{A})^+ \mathbf{W}, \quad (3.80)$$

with  $\mathbf{W}$  a diagonal matrix defining the least-squares weights. This will be further discussed in Section 4.2.1.

The minimum number of neighbours used in the least-squares technique is given by [32]:

$$d_{0,\min} = \frac{1}{D!} \prod_{i=1}^D (q + i) \quad (3.81)$$

where  $q$  denotes the degree of the highest derivative included in  $\mathbf{x}$  (for gradient reconstruction).

### 3.4.3 Polynomial Interpolation Filter

A commuting polynomial interpolation filter for unstructured grids has also been introduced by Marsden [25]. The above least-squares reconstruction-based discrete filter is however an effort to simplify this filter. A method is described to select the neighbours used in the discrete filter, as well as a method to control the filter width. The disadvantage of this filter is the difficulty of selecting the neighbours and extention to an arbitrary order of commutation error compared to the least-squares reconstruction-based discrete filter. The advantage is the possibility of using a more compact stencil and a more rigorous method to define the filter width. For a detailed explanation and implementation of this filter, the reader is referred to [25, 33].

### 3.4.4 Discrete Approximation of Convolution Filters

Integrating the convolution integral for the convolution filters presented in Section 3.3 is a possibility, but numerically very inefficient and non-local, requiring a lot of neighbouring cells. Sagaut [34] suggests two ways to approximate the convolution integral discretely with a compact stencil.

#### Polynomial Truncation

By introducing a Taylor expansion for  $\phi(x)$ , substituting this expansion into (3.15) and following some re-arrangement of terms, one arrives at the following definition for the filtered quantity [34]

$$\bar{\phi}(x) = \phi(x) + \sum_{l=1}^{\infty} \frac{M^l}{l!} \frac{\partial^l \phi(x)}{\partial x^l}. \quad (3.82)$$

Truncation of this series leads to an approximation for the filter. The top-hat and the Gaussian filter moments have been calculated in Table 3.1 so that the filter operations for truncation after the 8<sup>th</sup> order terms are

- for the top-hat filter

$$\bar{\phi}(x) = \phi(x) + \frac{\bar{\Delta}^2}{24} \frac{\partial^2 \phi(x)}{\partial x^2} + \frac{\bar{\Delta}^4}{1920} \frac{\partial^4 \phi(x)}{\partial x^4} + \frac{\bar{\Delta}^6}{322560} \frac{\partial^6 \phi(x)}{\partial x^6} + \frac{\bar{\Delta}^8}{92897280} \frac{\partial^8 \phi(x)}{\partial x^8} + O(\bar{\Delta}^{10}) \quad (3.83)$$

- for the Gaussian filter

$$\bar{\phi}(x) = \phi(x) + \frac{\bar{\Delta}^2}{24} \frac{\partial^2 \phi(x)}{\partial x^2} + \frac{\bar{\Delta}^4}{1152} \frac{\partial^4 \phi(x)}{\partial x^4} + \frac{\bar{\Delta}^6}{82944} \frac{\partial^6 \phi(x)}{\partial x^6} + \frac{\bar{\Delta}^8}{7962624} \frac{\partial^8 \phi(x)}{\partial x^8} + O(\bar{\Delta}^{10}) \quad (3.84)$$

The derivatives can then be calculated discretely. The derivatives have to be calculated to the same order of accuracy as the order of accuracy of the truncated Taylor expansion. A filtering operation in terms of the neighbours used in the discrete differentiation then follows.

### Optimized Approximation

Here the discrete filtering operator is defined in such a way that the the associated transfer function fits as well as possible with that of the target filter. Whereas the previous method was based on the equivalence up to a given order, the present one relies on an extension of the equivalence for a given wavenumber. This is done in a least-squares sense similar to (4.8). The target filter is chosen to be one of the convolution filters such as the top-hat or the Gaussian filter. The number of degrees of freedom to approximate the target filter will define the required number of neighbours. As an example, the functional corresponding to the 5-point approximation of the Gaussian filter is [34]

$$I_5 = \int_0^\pi \left[ e^{-\kappa^2 \bar{\Delta}^2 / 24} - 2 \left( w_2 \cos \left( \frac{2\kappa \bar{\Delta}}{FGR} \right) + w_1 \cos \left( \frac{\kappa \bar{\Delta}}{FGR} \right) + \frac{w_0}{2} \right) \right]^2 d(\kappa \bar{\Delta}), \quad (3.85)$$

where FGR is the filter-grid ratio defined as  $FGR = \frac{\bar{\Delta}}{\Delta}$ . It should be noted that the symmetry of the discrete filter (i.e.  $w_{-2} = w_2$  and  $w_{-1} = w_1$ ) is accounted for in this relation.

These discrete filters are easy to construct but do not allow to control the order of the commutation error. They are a good alternative however on uniform grids where the commutation error doesn't play a role. For assessment and implementation of these filters the reader is referred to [34].

### 3.5 Approximate Deconvolution Approach

For completeness another LES technique is mentioned. Stolz [35, 36] presented a discrete approach based on approximate deconvolution. In the Favre-filtered Navier-Stokes equations, the SFS stress tensor  $\tau_{ij}$  is defined as

$$\tau_{ij} = \overline{\rho u_i u_j} - \overline{\rho} \tilde{u}_i \tilde{u}_j . \quad (3.86)$$

The second term on the right hand side can easily be calculated since the filtered quantities  $\tilde{u}_i$  are solved for every time step. The first term however requires closure. By trying to recover the unfiltered quantity  $u_i$  however, one could easily perform the multiplication  $\rho u_i u_j$  and perform the filtering operation on this product. This is exactly the idea of the approach suggested by Stolz.

If  $G$  is the filter kernel of the convolution filter then the filtered quantity can be written as

$$\bar{\phi} = G \star \phi . \quad (3.87)$$

The unfiltered quantity can then be approximated by doing an inverse filtering operation:

$$\phi^* \approx G^{-1} \star \bar{\phi} , \quad (3.88)$$

Equation (3.86) then becomes

$$\tau_{ij} \approx \overline{\rho^* u_i^* u_j^*} - \overline{\rho} \tilde{u}_i \tilde{u}_j . \quad (3.89)$$

The scales smaller than the grid size can not be recovered with this operation, and an extra “relaxation term” is used to mimick the energy drain to the small scales. For more details the reader is referred to [36]. Mathew further refined this technique in [37, 38].

## Chapter 4

# Design and Assessment of Discrete Explicit Filters

This chapter describes two of the discrete explicit filters, introduced in Chapter 3 in further depth. One is the described discrete explicit filter based on solving a linear system of constraints, the other is the described discrete explicit filter based on a least-squares reconstruction. The design of these filters is studied, and their properties are assessed by showing their numerical application and order of commutation.

### 4.1 Linear Constraints Filter

#### 4.1.1 Design

The linear constraints filter is constructed in one space dimension and can then be easily extended to the three-dimensional case as described in Section 3.4. The filter is defined in computational space which dramatically simplifies its construction.

#### Linear Constraints

The linear constraints filter in one dimension is based on imposing  $N$  constraints on the filter moments, where  $N$  is the order of commutation, plus additional constraints that define the filter width and shape. The basic constraints were introduced earlier in Section 3.4.1:

$$\text{Filter moment } M^0 = 1 : \quad \sum_{l=-K_i}^{L_i} w_l^i = 1 , \quad (4.1)$$

$$\text{Filter moment } M^m = 0 : \quad \sum_{l=-K_i}^{L_i} l^m w_l^i = 0 , \quad m = 1, \dots, N-1 , \quad (4.2)$$

$$\hat{G}(\kappa_\Delta) = 0 : \quad \sum_{l=-K_i}^{L_i} (-1)^l w_l^i = 0 . \quad (4.3)$$

Using the above constraints does not guarantee a correct filter width or filter shape. Additional constraints can be added to obtain a higher quality filter:

- *Specifying the filter width.* This is equivalent to imposing the constraint

$$\hat{G}(\kappa_{\bar{\Delta}}) = 0.5 \quad (4.4)$$

or by using (3.60):

$$\sum_{l=-K_i}^{L_i} w_l^i e^{i\kappa_{\bar{\Delta}} l \Delta} = 0.5 . \quad (4.5)$$

This method could be used to specify other values of interest.

- *Specifying derivatives of  $\hat{G}(\kappa_\Delta)$  to zero.* One can set the derivatives of  $\hat{G}(\kappa_\Delta)$  to zero in order to construct a discrete filter transfer function that closely resembles a Fourier cut-off filter by forcing the transfer function close to the grid cut-off to be as close to zero as possible. This is equivalent to setting

$$\hat{G}(\kappa_\Delta)^{(p)} = 0 , \quad (4.6)$$

where superscript  $p$  represents the order of the derivative. The constraints corresponding to this condition are

$$\sum_{l=-K_i}^{L_i} (-i\kappa_\Delta l \Delta)^p w_l^i e^{i\kappa_{\bar{\Delta}} l \Delta} = 0 . \quad (4.7)$$

This can be enforced for an increasing number of derivatives  $p$  to get a better approximation to the Fourier cut-off filter. This method can also be used for other values of interest.

- *Approximating a target transfer function by use of linear least-squares.* In this method a target transfer function is approximated by using a linear least squares method with any number of remaining degrees of freedom. In [31], a least-squares method is described. However, in this thesis an alternate easier to implement procedure is presented. Defining

$\hat{G}_t(\kappa)$  to be the target transfer function, and  $w_p$  a remaining degree of freedom ( $p \in [-K_i, L_i]$ ). The condition corresponding to the least-squares optimization of this degree of freedom can then be written as

$$\frac{\partial}{\partial w_p} \left( \frac{1}{2} \int_0^{\kappa_\Delta} (\hat{G}(\kappa) - \hat{G}_t(\kappa))^2 d\kappa \right) = 0 \quad (4.8)$$

Using the definition for  $\hat{G}(\kappa)$  in (3.60) this can be further simplified:

$$\int_0^{\kappa_\Delta} \left( \sum_{l=-K_i}^{L_i} w_l^i e^{-\imath \kappa l \Delta} - \hat{G}_t(\kappa) \right) e^{-\imath \kappa p \Delta} d\kappa = 0 \quad (4.9)$$

or

$$\sum_{l=-K_i}^{L_i} w_l^i \int_0^{\kappa_\Delta} e^{-\imath \kappa (l-p) \Delta} d\kappa = \int_0^{\kappa_\Delta} \hat{G}_t(\kappa) e^{-\imath \kappa p \Delta} d\kappa \quad (4.10)$$

As  $\hat{G}(\kappa)$  is generally not purely real, the functional has to be re-expressed as

$$\frac{1}{2} \int_0^{\kappa_\Delta} (\mathcal{R}\{\hat{G}(\kappa) - \hat{G}_t(\kappa)\})^2 d\kappa + \frac{1}{2} \int_0^{\kappa_\Delta} (\mathcal{I}\{\hat{G}(\kappa) - \hat{G}_t(\kappa)\})^2 d\kappa \quad (4.11)$$

so that (4.10) has to be rewritten as

$$\begin{aligned} \sum_{l=-K_i}^{L_i} w_l^i & \left( \int_0^{\kappa_\Delta} \mathcal{R}\left\{e^{-\imath \kappa (l-p) \Delta}\right\} d\kappa + \int_0^{\kappa_\Delta} \mathcal{I}\left\{e^{-\imath \kappa (l-p) \Delta}\right\} d\kappa \right) = \\ & \int_0^{\kappa_\Delta} \mathcal{R}\left\{\hat{G}_t(\kappa) e^{-\imath \kappa p \Delta}\right\} d\kappa + \int_0^{\kappa_\Delta} \mathcal{I}\left\{\hat{G}_t(\kappa) e^{-\imath \kappa p \Delta}\right\} d\kappa \end{aligned} \quad (4.12)$$

### Effect of Order of Commutation Error

The selected order of the commutation error for the discrete filter has a substantial effect on the sharpness of its transfer function. This can be demonstrated by using only the basic constraints for varying filter orders. Using these basic constraints, the filter weights can be calculated. Table 4.1 shows the filter weights associated with each chosen commutation order. The corresponding filter transfer functions are shown in Figure 4.1. The transfer functions are plotted using the analytical expression in terms of the discrete filter weights as is described in (3.60). In this figure, FGR stands for filter-grid ratio. This is the ratio of the grid cut-off wavenumber and the filter cut-off wavenumber:  $FGR = \kappa_\Delta / \kappa_{\bar{\Delta}} = \bar{\Delta} / \Delta$ . The shown FGR is calculated by a root-finding method for  $\kappa_{\bar{\Delta}}$ . Desired values for this FGR are typically two and four (see Section 6.1.2 below).

A first observation from the results depicted in Figure 4.1 is that the higher the commutation order (or just order) of the filter, the sharper the filter transfer function. A second observation

order	w <sub>0</sub>	w <sub>±1</sub>	w <sub>±2</sub>	w <sub>±3</sub>	w <sub>±4</sub>
2	5.000000e-01	2.500000e-01			
4	6.250000e-01	2.500000e-01	-6.250000e-02		
6	6.875000e-01	2.343750e-01	-9.375000e-02	1.562500e-02	
8	7.265625e-01	2.187500e-01	-1.093750e-01	3.125000e-02	-3.906250e-03

Table 4.1: Filter weights for basic constraints

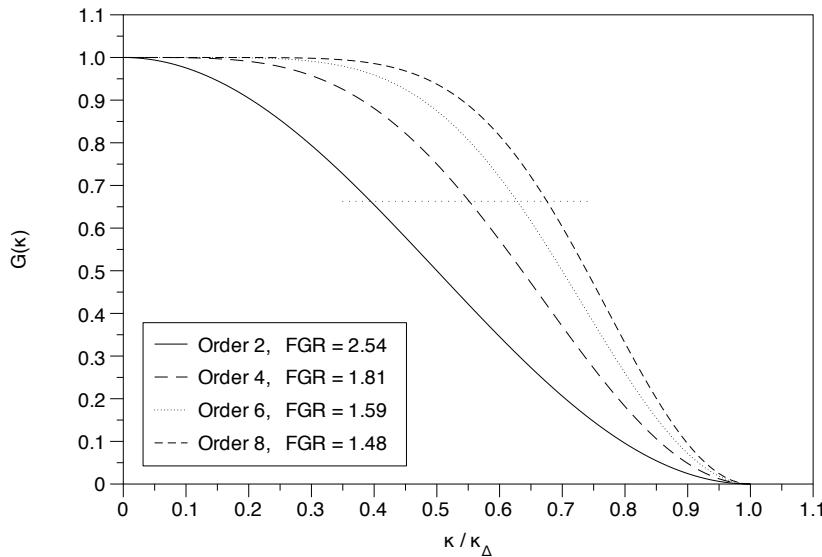


Figure 4.1: Filter transfer functions of the linear constraints filter using no additional constraints, for different commutation orders.

DC	w <sub>0</sub>	w <sub>±1</sub>	w <sub>±2</sub>	w <sub>±3</sub>	w <sub>±4</sub>	w <sub>±5</sub>
1	5.814161e-01	2.608960e-01	-4.070803e-02	-1.089598e-02		
3	5.610620e-01	2.812500e-01	-4.070803e-02	-3.125000e-02	1.017701e-02	
5	5.610620e-01	3.031458e-01	-4.070803e-02	-6.409364e-02	1.017701e-02	1.094788e-02

Table 4.2: Filter weights for the fourth-order linear constraints filter using additional constraints on derivatives of  $\hat{G}(\kappa_\Delta)$ , with FGR = 2 (DC = number of derivative constraints).

that can be made relates to the increase of the spectral filter width and thus a decrease of the FGR. This clearly indicates that higher-order filters are inherently sharper, but have a lower FGR. The number of points used in the stencil represents the minimum number required for a filter with that order of commutation. That means that for a filter of even order  $N$ , the minimum required number of rings around the central point equals  $N/2$ , where a ring represents the extension of the stencil in a symmetric way, by adding a neighbouring cell in each direction.

### Design with Constraints on Derivatives

Using the discrete filters described above as a basis, one can add additional rings to the stencil and add two extra constraints for each additional ring. In this section the effect of the additional constraints based on derivatives of the transfer function at the grid cut-off wavenumber will be studied, as imposed using (4.7). The base filter is a fourth-order commuting filter with 2 rings. Adding one ring allows to dictate the filter width (FGR = 2), and set the first-order derivative to zero. Adding one more ring allows to set also the second and third-order derivative to zero. The filter weights are tabulated in Table 4.2, and the transfer functions associated with these weights are given in Figure 4.2.

Figure 4.2 shows that the more constraints on the derivative are set, the sharper the transfer function. This is a nice property, but comes at a high cost; additional rings in the stencil are required. This approach will guarantee a zero complex part of the transfer function, as the filter weights are symmetric when a symmetric stencil is used.

### Design with Linear Least-Squares Constraints

Instead of adding constraints on the derivative of  $\hat{G}(\kappa_\Delta)$ , one can also try to minimize the difference between the filter transfer function and a target transfer function, as represented by (4.12) above. Special care has to be taken in the choice of the target filter. Taking a target filter that is too sharp, will lead to large undesirable overshoots in the filter transfer function.

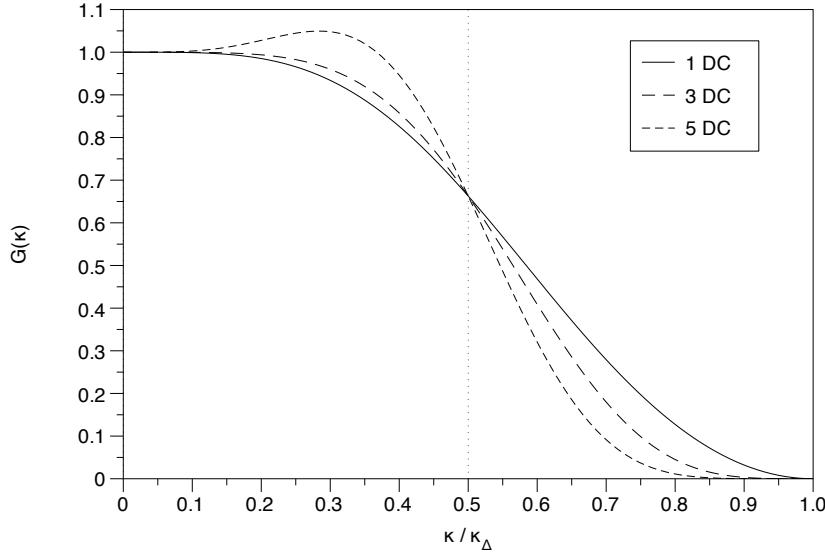


Figure 4.2: Filter transfer functions of the fourth-order linear constraints filter using additional constraints on derivatives of  $\hat{G}(\kappa_\Delta)$ , with FGR = 2. (DC = number of derivative constraints)

This consideration is different for every order of the filter: a second-order filter is inherently less sharp than a sixth-order filter; thus taking the same (sharp) target filter for the unsharp second-order filter as the sharper sixth order filter will lead to overshoots. Similarly taking an unsharp target filter for a sharp sixth-order filter will defeat the already inherent sharpness of the original filter. On the other hand, increasing the stencil and adding more least-squares constraints similarly allows to use sharper target filters, as more degrees of freedom are available to optimize the filter's transfer function.

The current choice of the target filter is the higher-order commuting analytical filter described in Section 3.3.2. This analytical expression for the target filter has the nice property that it becomes sharper as the commutation order increases and it allows for the control of the filter width. This last property is useful because the filter width can now be approached in a least-squares sense to allow for an optimal shape, or be specified as before. Adding one ring to the stencil, in addition to the stencil of the basic constraints filter, allows either for one least-squares constraint and one filter width constraint, or two least-squares constraints and only approaching the filter width. Each additional ring then adds two more least-squares constraints. This approach doesn't preserve the symmetry of the filter weights, even with a symmetric stencil. Therefore a complex part of the transfer function will be introduced. Eq. (4.12) minimizes the complex part however, so that this generally does not play an important role. The transfer function should be checked nevertheless in case the complex part becomes too large.

In this section, the same fourth-order commutative filter will be considered as the base filter.

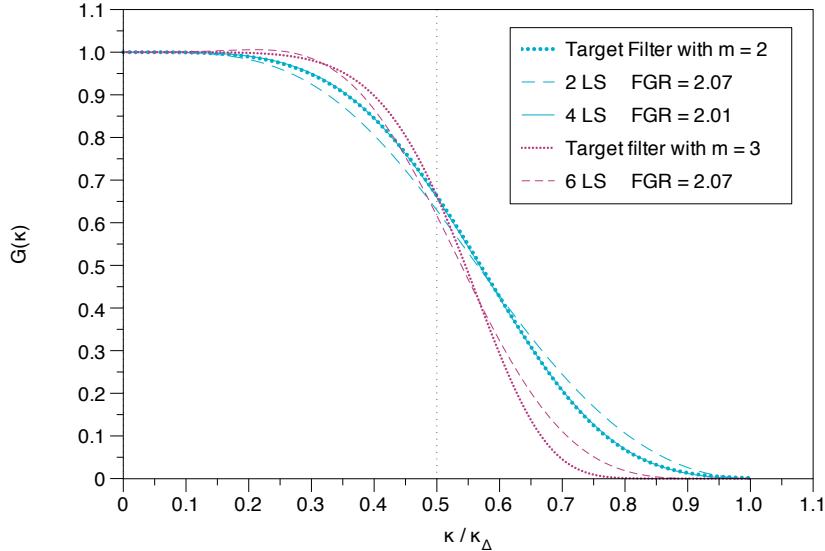


Figure 4.3: Filter transfer functions of fourth-order linear constraints filter using additional for least-squares constraints, with target FGR = 2. (LS = number of least-squares constraints)

Here the case will be considered where the filter width is approached by the least-square optimization procedure. When adding one and two rings, the target filter will be defined with  $m = 2$  in (3.53), and  $m = 3$  with three additional rings. These results are given in Table 4.3 and Figure 4.3.

These results show that the least-squares approach is capable of accurately approaching the target filter width. For the case with two additional constraints, the target filter shape is well approximated by the discrete filter. With four additional constraints, the target filter is almost exactly reproduced (the target filter with  $m = 2$ , shown in Figure 4.3 lies under this transfer function, denoted by “4 LS”). Since this target transfer function is almost exactly reproduced, adding two more additional constraints won’t affect the filter transfer function. Therefore, with six additional constraints the order of the target transfer function is set higher, to obtain a sharper target filter. As these results show, this target transfer function with  $m = 4$  is well approximated by the filter with six additional constraints.

LS	w <sub>-5</sub>	w <sub>-4</sub>	w <sub>-3</sub>	w <sub>-2</sub>	w <sub>-1</sub>	w <sub>0</sub>	w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>	w <sub>4</sub>	w <sub>5</sub>
2				-1.11e-02	-4.81e-02	2.84e-01	5.64e-01	2.45e-01	-1.65e-02	-1.90e-02	
4		6.65e-03	-2.44e-02	-4.03e-02	2.75e-01	5.65e-01	2.75e-01	-3.90e-02	-2.59e-02	7.14e-03	
6	-4.44e-04	2.55e-02	-6.34e-02	-2.88e-02	2.95e-01	5.42e-01	2.90e-01	-2.93e-02	-3.40e-02	-9.97e-03	1.14e-02

Table 4.3: Filter weights for the fourth-order linear constraints filter using additional for least-squares constraints, with target FGR = 2. (LS = number of least-squares constraints)

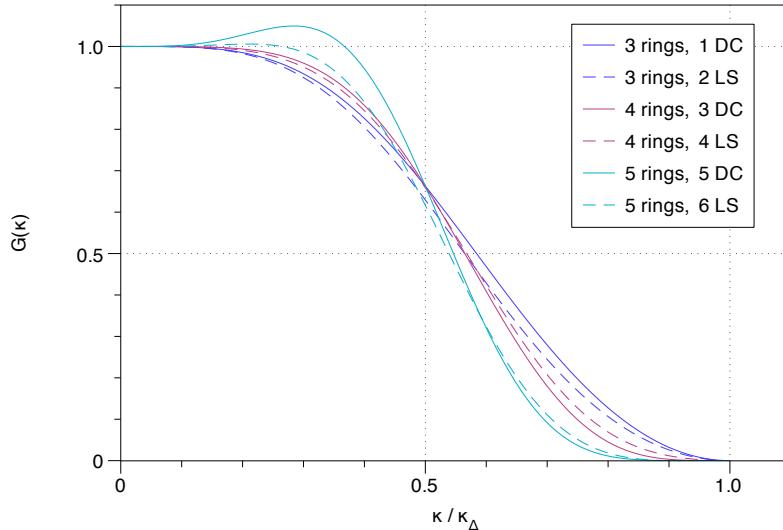


Figure 4.4: Comparison of the filter transfer functions from Figures 4.2 and 4.3.

### Design with Combination of Constraints

A comparison of the two types of constraints is given in Figure 4.4. The filter is of course not limited by one type of constraint. It is possible to combine these constraints to obtain the desired shape. Generally, when a high FGR is desired the least-squares constraints perform better, while for a FGR close to 2 the constraints on the derivative of  $\hat{G}(\kappa_\Delta)$  perform better. The user should make a careful study of the different transfer functions obtained by applying different constraints, when designing a linear constraints filter.

### Extension to Three Dimensions

With the design complete in one space dimension, extension to the three-dimensions case is rather straightforward using (3.67) and (3.71). The transfer function of a linear constraints filter with five additional least-squares constraints and a constraint for the filter width is calculated and shown in Figures 4.5 and 4.6. The transfer functions are plotted using the analytical expression in terms of the discrete filter weights, as is described in (3.63). In this figure, “1D” denotes the filter in the direction aligned with the spectral coordinate frame, “diagonal 2D” denotes the filter in the diagonal of the  $(\kappa_1, \kappa_2)$ -plane of the spectral coordinate frame, and “diagonal 3D” denotes the diagonal direction of the full 3D spectral coordinate frame. From this point on this notation, is used in other figures. As can be seen from Figures 4.5 and 4.6, the transfer function is not perfectly spherical. This is inherent to the way the filter is constructed: as a sequence of three one-dimensional filters. This means that the FGR is only correct in the

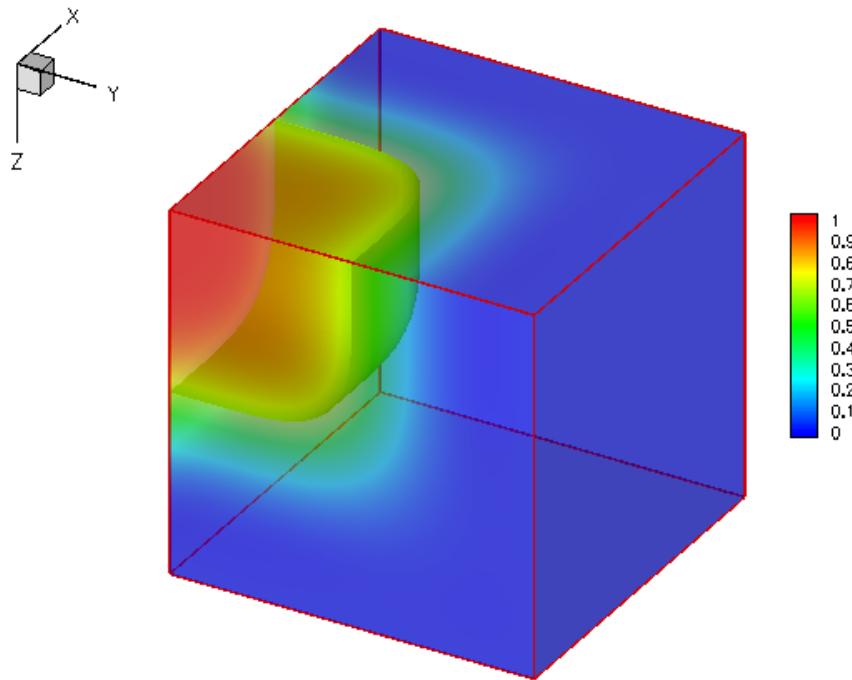


Figure 4.5: Three-dimensional filter transfer function of fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2. The iso-surface shown denotes  $\hat{G}(\kappa_{\bar{\Delta}})$ .

directions aligned with the coordinate frame.

#### 4.1.2 Assessment

##### Application to a Uniform Spectrum

After the shape of the transfer function has been optimized to the design criteria, it can be applied in practice. For this assessment, a uniform grid with  $64 \times 64 \times 64$  cells is used, with a uniform spectrum. This means that the Fourier-transformation of the unfiltered solution is assigned such that the average amplitude of each absolute value of every wavenumber represented on the grid is unity. The number of modes  $N$  represented on the grid in one direction is then  $N = \frac{64}{2} + 1 = 33$ . The solution is obtained from an inverse Fourier-transform, and numerically filtered in physical space. It is transformed back to the spectral domain to study the filtered uniform spectrum, providing a numerical estimation of the transfer function of this explicit discrete filter applied to a discrete mesh. This can then be compared to the theoretical transfer function obtained with (3.63). Using a finer mesh would provide for a more accurate

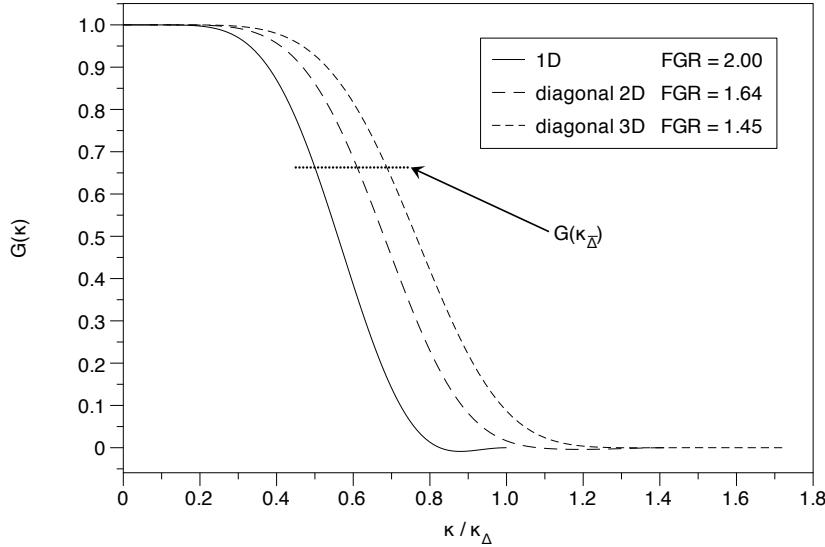


Figure 4.6: Diagonals of a three-dimensional filter transfer function of fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2.

and detailed estimate of the actual transfer function.

The application of the same fourth-order linear constraints filter as in Figures 4.5 and 4.6 is shown in Figures 4.7 and 4.8. In the latter, the averaged spectrum is also given to give an idea of the *actual* transfer function on the mesh. Comparing Figure 4.7 with Figure 4.5 shows good agreement. The non-spherical aspect, and filter-grid ratio's agree with the designed transfer function, as can be observed from the diagonals of the transfer function represented in Figure 4.8. The transfer function from the *domain average* shows that even though the filter is constructed with FGR = 2, the average FGR is well below this value. It would seem like a good approach to perform some trial and error adjustments to the constraints to obtain an average transfer function with the desired FGR.

### Demonstration of Commutation

To show the use of a commuting filter, an actual numerical calculation is performed on a stretched mesh with varying number of cells. The numeric calculations consists of filtering a sine function with a chosen commuting filter. A tenth-order finite-differencing scheme is used to obtain  $\frac{\partial \bar{\phi}}{\partial x}$ . The sine function is equally differentiated with the same scheme, and filtered next to obtain  $\frac{\partial \bar{\phi}}{\partial x}$ . The L2-norm of the absolute difference of these terms is taken over the entire grid. Figure 4.9 shows these values as a function of the number of cells in one direction on the grid. The order of the numerical finite-differencing scheme is shown as well.

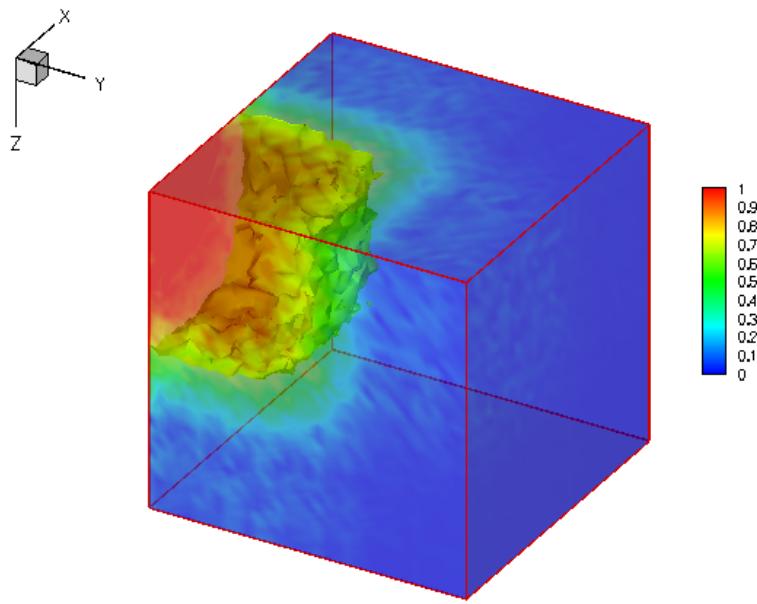


Figure 4.7: Three-dimensional spectrum of the application to a uniform spectrum of a fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2. The iso-surface shown represents  $\hat{G}(\kappa_{\bar{\Delta}})$ .

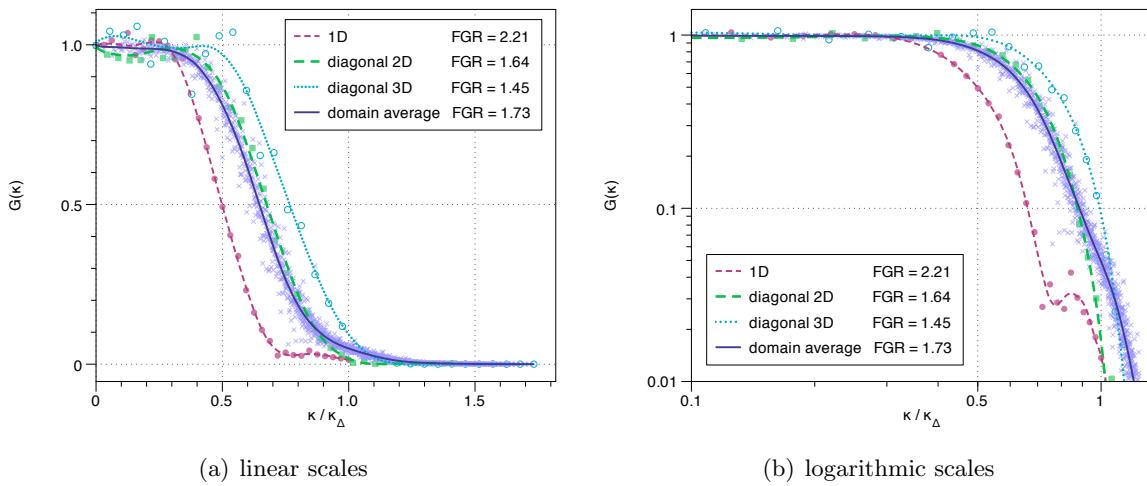


Figure 4.8: Diagonals of a three-dimensional spectrum of the application to a uniform spectrum of a fourth-order linear constraints filter using five additional least-squares constraints, and a constraint for FGR = 2.

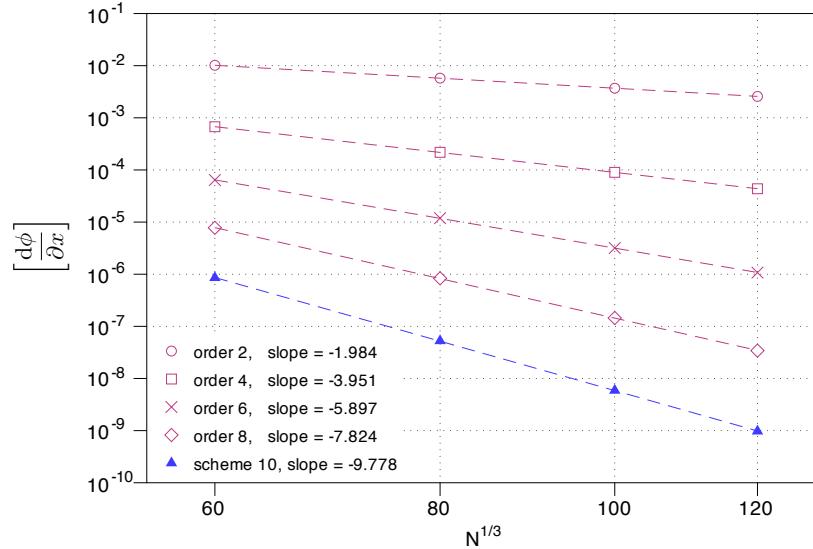


Figure 4.9: Demonstration of commutation of filters of order 2, 4, 6, 8. A finite differencing scheme of order 10 is used. Shown is the L2-norm of the commutation error, and the L2-norm of the truncation error.

The results of Figure 4.9 demonstrate that when using a high order numerical scheme (here order 10), the order of the filter will dictate the order of accuracy of the solution. It is now obvious that the commutation error may not be neglected and that control over the filter is essential to acquire the correct accuracy of the solution on non-uniform grids.

## 4.2 Least-Squares Reconstruction Filter

### 4.2.1 Design

Unlike the linear constraints filter, the design of least-squares reconstruction filters described in Section 3.4.2 are carried out fully in three dimensions. This filter is ideally suited for unstructured grids, for the same reasons least-squares gradient reconstruction is applicable to unstructured grids.

The general reconstruction equation for a filter with highest degree of included derivative,  $l$  (for brevity, we shall refer to this as a filter of degree  $l$ ) , can be written as

$$\phi_i = \bar{\phi}_0 + \sum_{k=1}^l \left[ \sum_{n_1+n_2+n_3=k} \frac{a^{n_1,n_2,n_3}}{k!} \frac{\partial^k \phi_0}{\partial x^{n_1} \partial y^{n_2} \partial z^{n_3}} \Delta x_{0i}^{n_1} \Delta y_{0i}^{n_2} \Delta z_{0i}^{n_3} \right], \quad i = 1, \dots, d_0, \quad (4.13)$$

where  $a^{n_1,n_2,n_3}$  are the trinomial coefficients defined by

$$a^{n_1,n_2,n_3} = \frac{(n_1 + n_2 + n_3)!}{n_1! n_2! n_3!} \quad (4.14)$$

In one dimension, a third-order filter would require a Taylor expansion of the form

$$\phi_i = \bar{\phi}_0 + \Delta x_{0i} \frac{\partial \phi}{\partial x} + \frac{\Delta x_{0i}^2}{2!} \frac{\partial^2 \phi}{\partial x^2} + \frac{\Delta x_{0i}^3}{3!} \frac{\partial^3 \phi}{\partial x^3}, \quad i = 1, \dots, d_0. \quad (4.15)$$

In three dimensions the number of terms in the Taylor expansion for this filter grows rapidly and is given by

$$\begin{aligned} \phi_i = & \bar{\phi}_0 + \frac{1}{1!} \frac{\partial \phi_0}{\partial x} \Delta x_{0i} + \frac{1}{1!} \frac{\partial \phi_0}{\partial y} \Delta y_{0i} + \frac{1}{1!} \frac{\partial \phi_0}{\partial z} \Delta z_{0i} \\ & + \frac{1}{2!} \frac{\partial^2 \phi_0}{\partial x^2} \Delta x_{0i}^2 + \frac{2}{2!} \frac{\partial^2 \phi_0}{\partial x \partial y} \Delta x_{0i} \Delta y_{0i} + \frac{1}{2!} \frac{\partial^2 \phi_0}{\partial y^2} \Delta y_{0i}^2 \\ & + \frac{2}{2!} \frac{\partial^2 \phi_0}{\partial x \partial z} \Delta x_{0i} \Delta z_{0i} + \frac{2}{2!} \frac{\partial^2 \phi_0}{\partial y \partial z} \Delta y_{0i} \Delta z_{0i} + \frac{1}{2!} \frac{\partial^2 \phi_0}{\partial z^2} \Delta z_{0i}^2 \\ & + \frac{1}{3!} \frac{\partial^3 \phi_0}{\partial x^3} \Delta x_{0i}^3 + \frac{3}{3!} \frac{\partial^3 \phi_0}{\partial x^2 \partial y} \Delta x_{0i}^2 \Delta y_{0i} + \frac{3}{3!} \frac{\partial^3 \phi_0}{\partial x \partial y^2} \Delta x_{0i} \Delta y_{0i}^2 \\ & + \frac{1}{3!} \frac{\partial^3 \phi_0}{\partial y^3} \Delta y_{0i}^3 + \frac{3}{3!} \frac{\partial^3 \phi_0}{\partial x^2 \partial z} \Delta x_{0i}^2 \Delta z_{0i} + \frac{6}{3!} \frac{\partial^3 \phi_0}{\partial x \partial y \partial z} \Delta x_{0i} \Delta y_{0i} \Delta z_{0i} \\ & + \frac{3}{3!} \frac{\partial^3 \phi_0}{\partial y^2 \partial z} \Delta y_{0i}^2 \Delta z_{0i} + \frac{3}{3!} \frac{\partial^3 \phi_0}{\partial x \partial z^2} \Delta x_{0i} \Delta z_{0i}^2 + \frac{3}{3!} \frac{\partial^3 \phi_0}{\partial y \partial z^2} \Delta y_{0i} \Delta z_{0i}^2 \\ & + \frac{1}{3!} \frac{\partial^3 \phi_0}{\partial z^3} \Delta z_{0i}^3, \quad i = 1, \dots, d_0. \end{aligned} \quad (4.16)$$

The number of unknowns in the least-squares reconstruction thus increases rapidly as well. The number of terms and unknowns  $n$  can be calculated using

$$n = \sum_{k=0}^l \frac{(d+k-1)!}{k! (d-1)!}, \quad (4.17)$$

where  $d$  is the number of dimensions. Table 4.4 shows the number of unknowns for different orders of commutation, in one, two, and three dimensions.

degree	1	2	3	4	5	6	7	8
1D	2	3	4	5	6	7	8	9
2D	3	6	10	15	21	28	36	45
3D	4	10	20	35	56	84	120	165

Table 4.4: Number of unknowns using the least-squares reconstruction filter for different degrees in one, two, and three dimensions.

### Effect of the Relaxation Factor

The relaxation factor is the parameter that weights the central value, as described in (3.79). In this section the influence of this factor on the discrete filter will be discussed. For visualization purposes, the study will be made on the three-dimensional diagonal of the transfer function, under the assumption that the filter is spherical. The specific discrete filter considered here is constructed with a uniform stencil with three rings and was selected to be of third degree ( $l = 3$ ). The least-squares reconstruction involves no weighting of the constraints.

The effect of the relaxation factor,  $w_0$ , on the transfer function of this third-order discrete filter is shown in Figure 4.10. As can be seen, the relaxation factor doesn't change the actual shape of the transfer function, but scales the transfer function changing  $\hat{G}(\kappa_\Delta)$ , keeping  $\hat{G}(0) = 0$ .

It can be useful to force the transfer function to zero at the largest wavenumber represented on the grid. The relaxation factor that is capable of forcing this, is dependent on the number of points used in the stencil, as well as the degree of the filter, and is thus very unpredictable. A root-finding method (Ridder's method) is used to find the value of  $w_0$  such that  $\hat{G}(\kappa_\Delta) = 0$ ; note that  $|\kappa_\Delta| = \sqrt{3} \kappa_\Delta$ , with  $\kappa_\Delta$  the one-dimensional grid cut-off wavenumber. In the following least-squares filter studies, the relaxation factor will be optimized in this fashion.

### Effect of Weighting

To have finer control over the transfer function shape, weighted least-squares is introduced, as in (3.80). Several weighting functions can be used:

- *Inverse distance.* A weighting to the cells in the reconstruction is applied proportional to

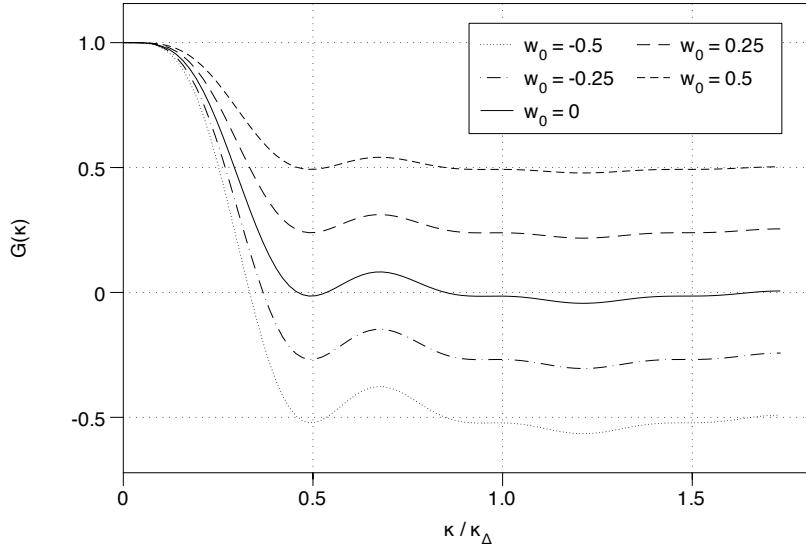


Figure 4.10: Effect of the relaxation factor  $w_0$  on the diagonals of a third-order least-squares reconstruction filter transfer function obtained using three rings of solution values.

the inverse distance to the centre cell:

$$W_i = \frac{1}{|\Delta\mathbf{r}_{0i}|} \quad (4.18)$$

- *Inverse squared distance.* A weighting to the cells in the reconstruction is applied proportional to the inverse of the squared distance to the centre cell:

$$W_i = \frac{1}{|\Delta\mathbf{r}_{0i}|^2} \quad (4.19)$$

- *Gaussian distribution.* A weighting to the cells in the reconstruction is applied according to a Gaussian distribution as function of the distance to the centre cell:

$$W_i = \sqrt{\frac{6}{\pi D^2}} \exp\left(-\frac{6|\Delta\mathbf{r}_{0i}|^2}{D^2}\right). \quad (4.20)$$

This weighting has a parameter  $D$  that can be tuned. This will prove useful in the control of the filter width.

Figure 4.11 shows the effect of the change in weighting functions to the diagonals of the filter transfer function.

It is observed that the weighting actually increases the filter width. This is a direct effect of reducing the importance of cells further away from the centre. The sharpness of the filter is decreased by weighting. The Gaussian weighting has the rather nice property that the transfer

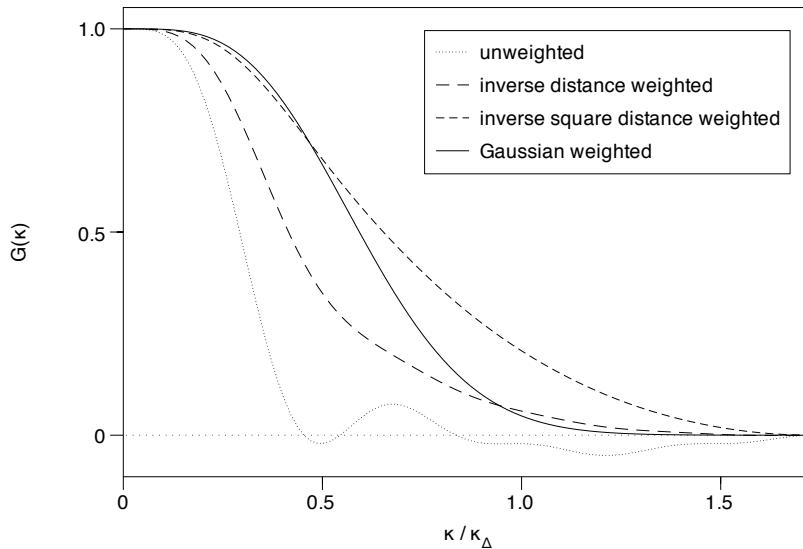


Figure 4.11: Effect of the different weighting functions on the diagonals of a least-squares reconstruction filter transfer function of third degree with three rings.

function gains some of the nice properties of the analytical Gaussian function, but with none of its disadvantages. The filter width is somewhat controllable through the parameter  $D$ , and the transfer function does not exhibit oscillations, as is the case with the unweighted transfer function. The filter still retains its high commutation order, unlike the analytical Gaussian filter. In the further studies of the least-squares filters below, the Gaussian distribution will be used as the weighting function.

### Effect of the Weighting Factor

In this section the effect of the weighting factor or the parameter  $D$  in the Gaussian distribution for the weighted least-squares reconstruction will be studied. For the purpose of maximum effectiveness, the used stencil uses six neighbouring rings to achieve a third-order filter. The weighting factor is taken to be proportional to the mesh spacing,  $\Delta$ , in order to be scalable to the mesh. Figure 4.12 demonstrates the effect of this weighting factor on the diagonal of the transfer function. Starting from the weighting factor  $D = 5 \Delta$ , the Gaussian distribution envelops all cells in the stencil. As the weighting factor decreases, outer cells become less important and don't contribute as much to the reconstruction, effectively decreasing the FGR. The difference between  $D = 5 \Delta$  and  $D = 4 \Delta$  is not as big as for  $D = 3 \Delta$  and  $D = 2 \Delta$ . This shows that a maximum weighting factor exists. This is because the Gaussian function becomes so wide that all the points in the stencil get almost equal importance in the reconstruction. The

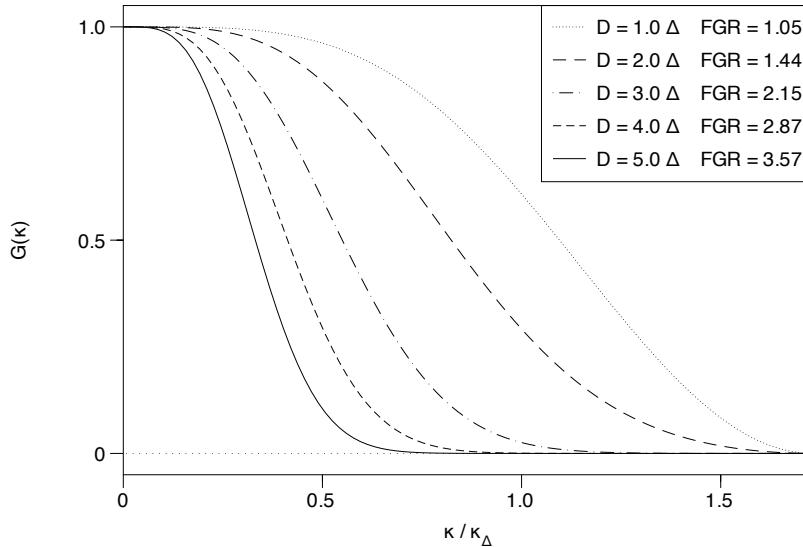


Figure 4.12: Effect of the different weighting factors in the Gaussian distribution on the diagonals of a third-order least-squares reconstruction filter transfer function with six rings.

stencil then has to be expanded to allow for significant difference in importance of the cells. So the more rings are included in the stencil, the greater the FGR can become. This is why for this demonstration, six rings were included in the stencil.

### Effect of the Order of the Commutation Error

As the commutation order increases, and thus the required degree in the Taylor series expansion for the reconstruction, the number of unknowns to be solved in the linear system described by (3.75) increases rapidly as can be seen in Table 4.4. It is evident that more rings in the stencil will be required to maintain an overdetermined linear system.

To allow for an objective comparison of the effect of the highest degree used in the reconstruction, the reconstruction uses again six neighbouring rings and a Gaussian weighting with a fixed weighting factor ( $D = 3 \Delta$ ). The first degree reconstruction with 2 neighbouring rings should then give the same transfer function as when 6 neighbouring rings are used. Figure 4.13 demonstrates the transfer functions with different degrees.

It is apparent from this figure that the increase of the degree (and thus the commutation order) is similar to previous observations for the one-dimensional case given in Figure 4.1. As the degree increases, the transfer function becomes sharper and wider, decreasing the FGR. This makes it inherently harder to obtain a high FGR for filters of high commutation order. As

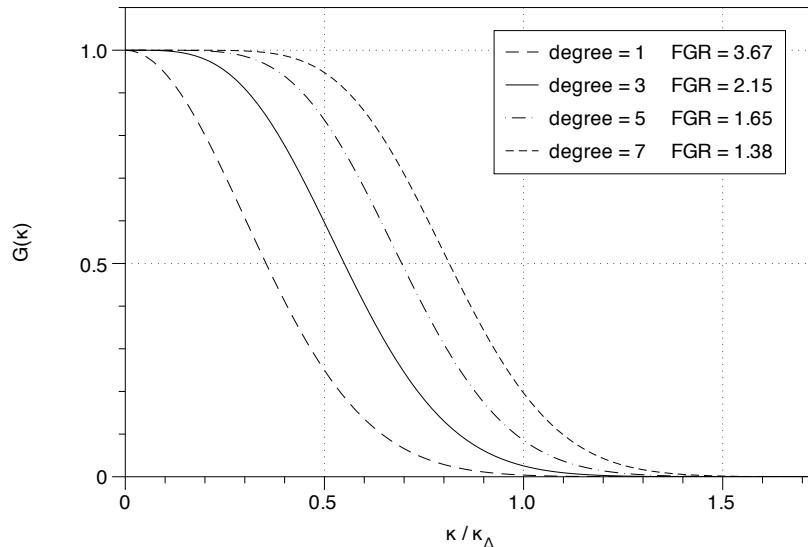


Figure 4.13: Effect of the commutation order on the diagonals of a weighted least-squares reconstruction filter transfer function with a Gaussian weighting ( $D = 3.0 \Delta$ )

observed from the study of the effect of the weighting factor above, the stencil will have to be larger to obtain a higher FGR; more so for filters of high commutation order.

Because of the three-dimensional nature of the least-squares reconstruction, the transfer function can be constructed to be more spherical than the linear constraints filter, as was assumed in the previous sections. This is however not always the case. With two rings around the central point, the total number of discrete points in the stencil is 125 ( $d_0 = 125$ ). Theoretically a degree of 7 can then be used in the least-squares reconstruction (see Table 4.4). In practice however the filter becomes highly non-spherical when using a higher degree than the number of rings. To demonstrate this, a third-order filter with Gaussian weighting factor  $D = 3.5$  will be studied using different stencils. Figures 4.14 and 4.15 show that a stencil with one neighbouring ring is insufficient for this filter of third degree. The transfer function is not spherical at all, and the one-dimensional direction is not filtered at all. Figures 4.16 and 4.17 show that by increasing the stencil to 4 neighbouring rings the transfer function becomes perfectly spherical. A stencil with 2 and 3 neighbouring rings give approximate spherical transfer functions. The more rings, the better the omni-directional nature of the transfer function.

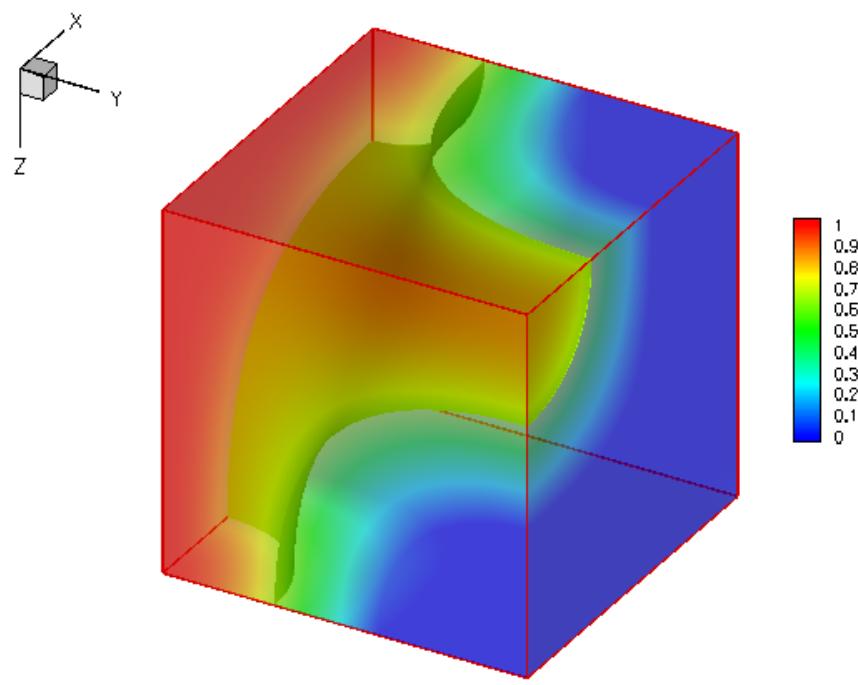


Figure 4.14: Filter transfer function of a third-order weighted least-squares filter with  $D = 3.5\Delta$ , using a stencil with 1 neighbouring ring.

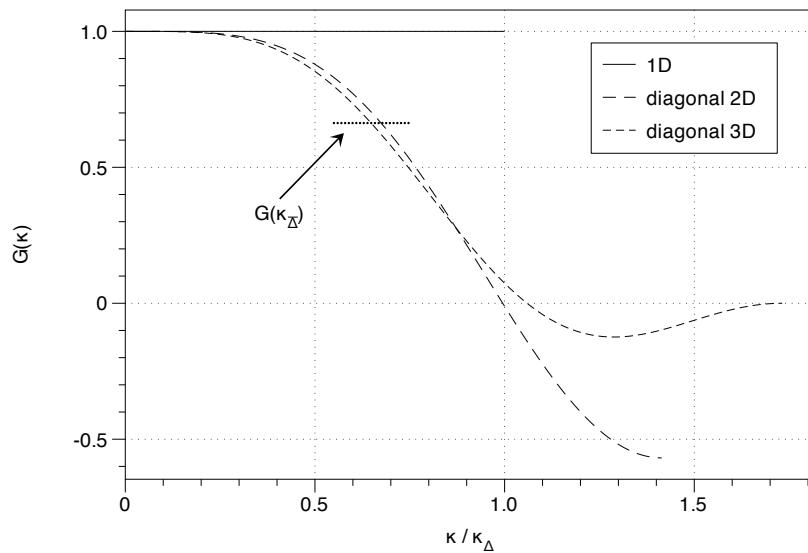


Figure 4.15: Diagonals of the filter transfer function of a third-order weighted least-squares filter with  $D = 3.5\Delta$ , using a stencil with 1 neighbouring ring.

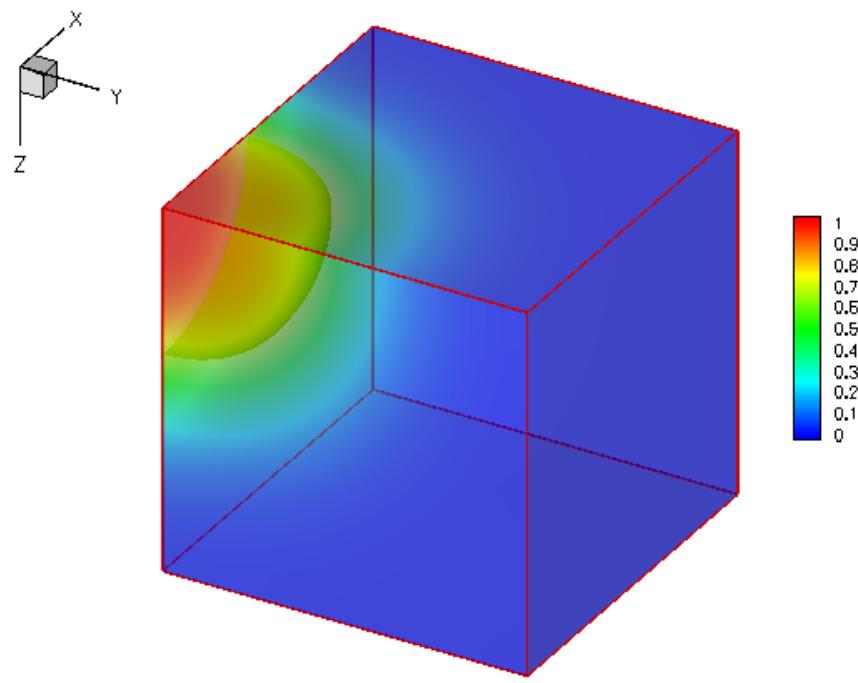


Figure 4.16: Filter transfer function of a third-order weighted least-squares filter of with  $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings.

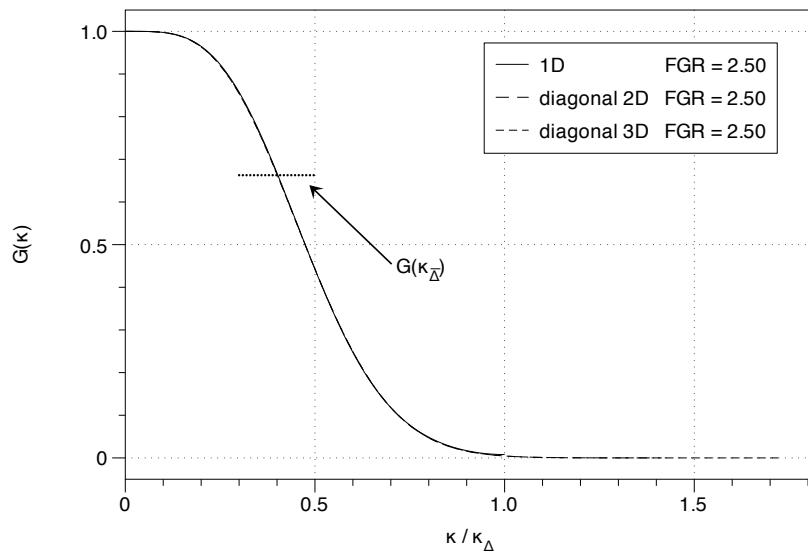


Figure 4.17: Diagonals of the filter transfer function of a third-order weighted least-squares filter with  $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings.

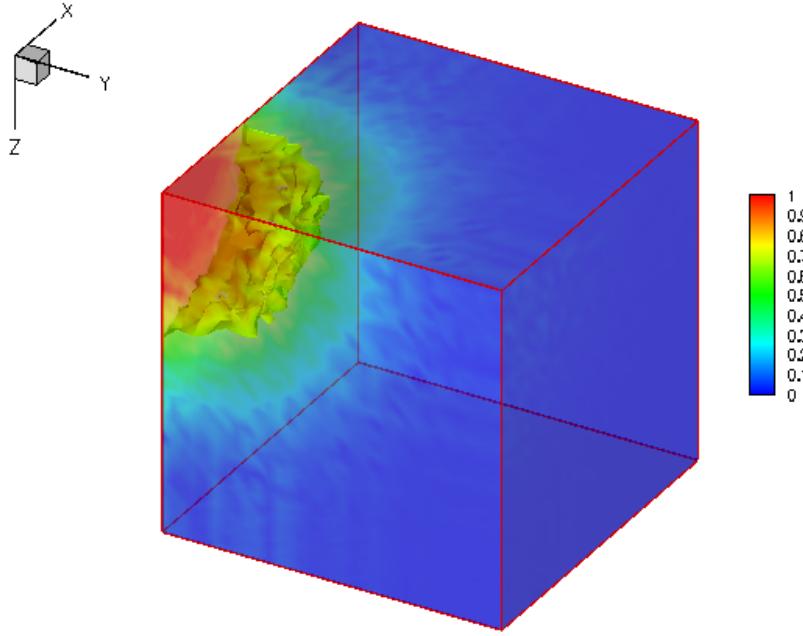


Figure 4.18: Three-dimensional spectrum of the application to a uniform spectrum of a third-order weighted least-squares filter with  $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings. The shown iso-surface denotes  $\hat{G}(\kappa_{\bar{\Delta}})$ .

#### 4.2.2 Assessment

##### Application to a Uniform Spectrum

The same grid and uniform spectrum as in Section 4.1.2 is reconsidered in order to assess the application of the discrete least-squares filter described above and whose properties are shown in Figures 4.16 and 4.17. This is a third-order weighted least-squares reconstruction filter with 4 neighbouring rings, with weighting factor  $D = 3.5 \Delta$ . The application of this filter is shown in Figures 4.18 and 4.19. In the latter, the averaged spectrum is also given to get an idea of the mean transfer function.

Comparing Figure 4.18, where the actual transfer function is numerically calculated through application to a uniform spectrum, with Figure 4.16, where the transfer function is analytically calculated using (3.63), shows good agreement. As can be noted from the diagonals represented in Figure 4.19, the spherical aspect is maintained quite well, with exception of the one-dimensional direction aligned with the coordinate frame. The transfer function from the “domain average” shows that the average FGR is the same as the designed one. This is in contrast to the finding in the application of the linear constraints filter described earlier in the

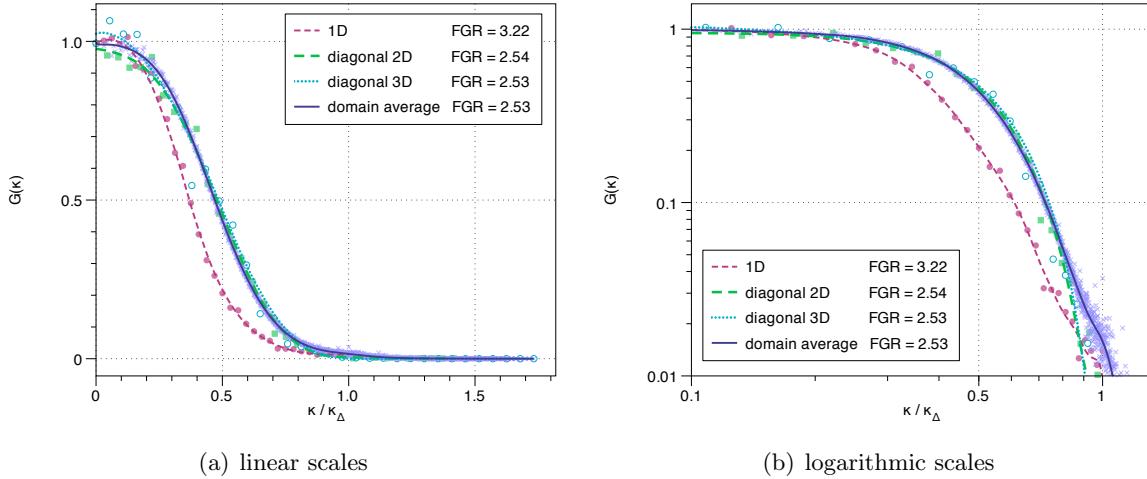


Figure 4.19: Diagonals of a three-dimensional spectrum of the application to a uniform spectrum of a of a third-order weighted least-squares filter with  $D = 3.5 \Delta$ , using a stencil with 4 neighbouring rings.

Chapter.

### Demonstration of Commutation

Similar to the demonstration of commutation for the linear constraints filter (Section 4.1.2), an actual numerical calculation is performed on non-uniform grids with varying number of cells to demonstrate the use of commuting filters. The numerical calculations consist of filtering a sine function with a chosen commuting filter. A tenth-order finite-differencing scheme is used to obtain  $\frac{\partial \phi}{\partial x}$ . The sine function is equally differentiated with the same scheme, and filtered next to obtain  $\frac{\partial \phi}{\partial x}$ . The L2-norm of the absolute difference of these terms is taken over the entire grid.

When using the same stretched mesh as for the linear constraints filter, the commutation error was zero for every order. When using a curved mesh however, commutation errors become significant. Figure 4.20 shows the L2-norms of the commutation errors as a function of the number of cells on a curved mesh.

The order of commutation error for a filter of second and third degree is both 3. This is because cancellation occurs due to the ring-approach in defining the stencil, leading to a higher order than predicted for the second degree filter. It is therefore advantageous and much less computationally expensive to construct and apply the filter of second degree than it is for the filter of third degree. The same reasoning explains the agreement between the commutation

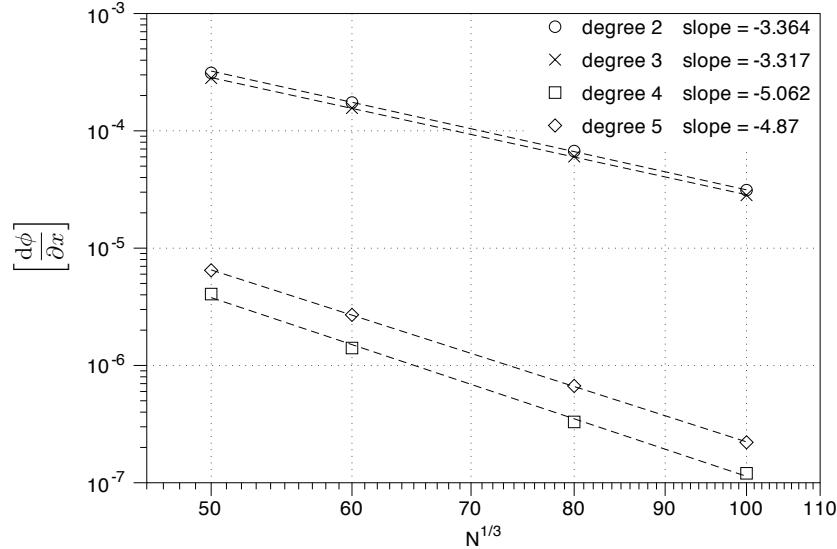


Figure 4.20: Demonstration of commutation of filters of degree 2, 3, 4, 5. Shown is the L2-norm of the commutation error.

orders of the filter of fourth and fifth degree.

Figure 4.20 also clearly demonstrates that when using a high-order spatial discretization scheme, the order of the filter will dictate the order of accuracy of the solution. It is now obvious that the commutation error may not be neglected, and that control over the filter is essential to acquire the correct accuracy of filtered solutions on non-uniform grids.

# Chapter 5

## Parallel Finite-Volume Scheme

This chapter provides a brief summary of the numerical scheme and solver used here to obtain LES solutions of compressible turbulent flows. A second-order Godunov finite-volume method is used to solve the Favre-filtered Navier-Stokes equations as they are described in Section 2.3.3. This finite-volume method has already successfully been applied to LES of turbulent premixed combusting flows [39].

In what follows, the finite-volume method is described for application to the solution of the Favre-filtered Navier-Stokes equations on a multi-block structured mesh.

### 5.1 Favre-Filtered Navier-Stokes Equations in Matrix-Vector Notation

The Navier-Stokes equations described in Section 2.3.1 and the closed form of the Favre-filtered Navier-Stokes Equations described in Section 2.3.3 have a similar structure and can therefore be cast using matrix-vector notation in the following form:

$$\frac{d\mathbf{U}}{dt} + \nabla \cdot \vec{\mathbf{F}} = \boldsymbol{\xi}, \quad (5.1)$$

where  $\mathbf{U}$  represents the vector of conserved variables,  $\vec{\mathbf{F}}$  is the flux dyad, and  $\boldsymbol{\xi}$  is the source term. The flux dyad  $\vec{\mathbf{F}}$  can be split into an inviscid part  $\mathbf{F}$  and a viscous part  $\mathbf{F}_\nu$ , and can be written as

$$\begin{aligned} \vec{\mathbf{F}} &= (\vec{\mathbf{F}}_x, \vec{\mathbf{F}}_y, \vec{\mathbf{F}}_z) \\ &= (\mathbf{F} - \mathbf{F}_\nu, \mathbf{G} - \mathbf{G}_\nu, \mathbf{H} - \mathbf{H}_\nu) \end{aligned} \quad (5.2)$$

In case of the Navier-Stokes equations (see eq. (2.8)-(2.10)), the vectors  $\mathbf{U}$ ,  $\mathbf{F}$  and  $\mathbf{F}_\nu$  can be expressed as

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \quad \mathbf{F}_\nu = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ u\sigma_{xx} + v\sigma_{xy} + w\sigma_{xz} - q_x \end{pmatrix}, \quad (5.3)$$

where the velocity vector was split in components  $u, v, w$ .

For the Favre-filtered Navier-Stokes equations, the vector  $\mathbf{U}$  now contains the Favre-filtered conserved variables, and the fluxes are now constructed to include the extra terms as they appear in equations (2.36)–(2.38):

$$\mathbf{U} = \begin{pmatrix} \bar{\rho} \\ \bar{\rho}u \\ \bar{\rho}v \\ \bar{\rho}w \\ \bar{\rho}E \end{pmatrix}, \quad (5.4)$$

$$\mathbf{F} = \begin{pmatrix} \bar{\rho}u \\ \bar{\rho}\tilde{u}^2 + \tilde{p} \\ \bar{\rho}\tilde{u}\tilde{v} \\ \bar{\rho}\tilde{u}\tilde{w} \\ (\bar{\rho}E + \bar{p})\tilde{u} \end{pmatrix}, \quad \mathbf{F}_\nu = \begin{pmatrix} 0 \\ \check{\sigma}_{xx} - \tau_{xx} \\ \check{\sigma}_{xy} - \tau_{xy} \\ \check{\sigma}_{xz} - \tau_{xz} \\ u(\check{\sigma}_{xx} - \tau_{xx}) + v(\check{\sigma}_{xy} - \tau_{xy}) + w(\check{\sigma}_{xz} - \tau_{xz}) - \check{q}_x - \check{q}_{T_x} \end{pmatrix},$$

in which  $\check{q}_T$  is the SFS heat conduction given by (2.34). Similar expressions can be derived for  $\vec{\mathbf{F}}_y = \mathbf{G} - \mathbf{G}_\nu$  and  $\vec{\mathbf{F}}_z = \mathbf{H} - \mathbf{H}_\nu$ .

## 5.2 Godunov Finite-Volume Method

### 5.2.1 Finite-Volume Formulation of Governing Equations

In the finite-volume method used herein, the spatial discretization of the governing Favre-filtered Navier-Stokes equations is carried out by considering their integral form for a finite control volume which is taken to be the computational cell of interest. This integral form can be written as

$$\int_V \left[ \frac{d\mathbf{U}}{dt} + \nabla \cdot \vec{\mathbf{F}} \right] dV = \int_V \boldsymbol{\xi} dV, \quad (5.5)$$

where  $V$  represents the control volume. The finite-volume method is based on the integral form of these conservation laws. The average of  $\mathbf{U}$  and  $\boldsymbol{\xi}$  over the control volume can be defined as:

$$\mathbf{U} = \frac{1}{V} \int_V \mathbf{U} \, dV, \quad (5.6)$$

$$\mathbf{S} = \frac{1}{V} \int_V \boldsymbol{\xi} \, dV. \quad (5.7)$$

Eq. (5.5) can be rewritten using (5.6)–(5.7) and the Gauss-Ostrogradsky or divergence theorem as

$$\frac{d\mathbf{U}}{dt} = -\frac{1}{V} \oint_S \mathbf{n} \cdot \vec{\mathbf{F}} \, d\mathcal{S} + \mathbf{S}, \quad (5.8)$$

where  $S$  is the closed surface of the control volume  $V$ . When applied to a three-dimensional control volume in a structured multi-block mesh consisting of hexahedral cells, (5.8) can be rewritten for cell  $(i,j,k)$  as

$$\frac{d\mathbf{U}_{i,j,k}}{dt} = -\frac{1}{V_{i,j,k}} \sum_{l=1}^{N_f} \left[ \mathbf{n}_l \cdot \vec{\mathbf{F}}_l \, A_l \right]_{i,j,k} + \mathbf{S}_{i,j,k}, \quad (5.9)$$

where  $N_f$  denotes the number of cell faces, and  $A_l$  the surface area of face  $l$ . Spatial discretization leads to a coupled non-linear system of ordinary differential equations.

### 5.2.2 Inviscid Flux Evaluation

The finite-volume scheme defined above requires the evaluation of the solution fluxes at the cell faces, as can be understood from (5.9). Eq. (5.3) shows that the inviscid (hyperbolic) flux is function of the conserved variables. In a finite-volume scheme, these conserved variables are cell averages and are thus different for every cell. The Riemann problem is thus a one-dimensional initial value problem with initial data given by

$$\mathbf{U} = \begin{cases} \mathbf{U}_{\text{left}} & \text{if } x < x_0 \\ \mathbf{U}_{\text{right}} & \text{if } x \geq x_0 \end{cases} \quad (5.10)$$

For hyperbolic systems, this problem will have a self-similar solution and results in the generation of waves separating piecewise constant states.

A Riemann solver is used to determine the solution of the Riemann problem at the cell interfaces. Exact Riemann solvers, such as the one proposed by Gottlieb and Groth [40] can be used for the Euler equations. The  $x$ -direction inviscid flux  $\mathbf{F}$  between cell  $(i,j,k)$  and cell  $(i+1,j,k)$  is then given by

$$\mathbf{F}_{i+\frac{1}{2},j,k} = \mathbf{F}(\mathcal{R}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i+1,j,k})) . \quad (5.11)$$

More efficient approximate Riemann solvers such as the Roe [41], HLLE [42] and AUSM<sup>+</sup>-up [43] approximate Riemann solvers can be used for more complicated systems, where the approximation doesn't decrease the accuracy of the overall solution scheme. Solving the Riemann problem provides natural upwinding. In this thesis research, the AUSM<sup>+</sup>-up approximate Riemann solver proposed by Liou [43] is used, as it is valid for a wide range of flow speeds and Mach numbers encountered in the turbulent flows of interest here.

### Limited Reconstruction

In order to obtain a solution that is first-order accurate in space, flux evaluations are carried out by solving Riemann problems using the cell averages as the left and right states. For higher-order accuracy however, sub-cell reconstruction is required. A linear least-squares reconstruction technique is used to determine the solution slopes within each cell based on information from neighbouring cells [44]. The fluxes are then calculated by solving a Riemann problem based on the reconstructed solution states at the midpoint of each cell interface as follows:

$$\begin{aligned}\mathbf{U}_{i+1/2,j,k}^{\text{left}} &= \mathbf{U}_{i,j,k} + \phi_{i,j,k} \Delta\mathbf{r}_{i,j,k} \nabla\mathbf{U}_{i,j,k}, \\ \mathbf{U}_{i+1/2,j,k}^{\text{right}} &= \mathbf{U}_{i+1,j,k} + \phi_{i+1,j,k} \Delta\mathbf{r}_{i+1,j,k} \nabla\mathbf{U}_{i+1,j,k},\end{aligned}\quad (5.12)$$

where  $\Delta\mathbf{r}$  is the vector containing the distance between the midpoint of the cell interface and the cell centre, and  $\phi$  is a vector of slope limiters. This reconstruction assumes a piecewise linear solution within each cell, and thus increasing the order of the spatial discretization scheme to second-order. Limiters must be used in order to maintain monotonicity of the solution near shocks and discontinuities. An extensive analysis of limiters is made in [45]. The limiter used in this thesis research is the one proposed by Barth and Jespersen [46].

#### 5.2.3 Viscous Flux Evaluation

The viscous (elliptic) fluxes are not hyperbolic and can thus not be found through a Riemann solver. The viscous flux is instead calculated on the cell face using the conserved variables and its gradient on the cell face:

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F} \left( \mathbf{U}_{i+\frac{1}{2},j,k}, \nabla\mathbf{U}_{i+\frac{1}{2},j,k} \right) \quad (5.13)$$

In the current finite-volume approach, the gradient  $\nabla\mathbf{U}_{i+\frac{1}{2},j,k}$  is found using the hybrid average gradient-diamond-path approach proposed by Mathur and Murthy [47, 48]. In this approach, the face gradient is evaluated using

$$\nabla\mathbf{U}_{i+\frac{1}{2},j,k} = \frac{\mathbf{U}_{i+1,j,k} - \mathbf{U}_{i,j,k}}{ds} \frac{\mathbf{n}}{\mathbf{n} \cdot \mathbf{e}_s} + \left( \overline{\nabla\mathbf{U}} - \overline{\nabla\mathbf{U}} \cdot \mathbf{e}_s \frac{\mathbf{n}}{\mathbf{n} \cdot \mathbf{e}_s} \right), \quad (5.14)$$

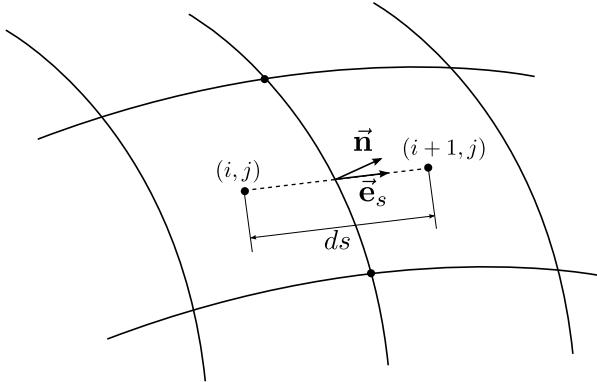


Figure 5.1: Hybrid average gradient-diamond-path approach for a two-dimensional grid.

with

$$\bar{\nabla} \mathbf{U} = \alpha \nabla \mathbf{U}_{i,j,k} + (1 - \alpha) \nabla \mathbf{U}_{i+1,j,k}, \quad (5.15)$$

$$\alpha = \frac{V_{i,j,k}}{V_{i,j,k} + V_{i+1,j,k}}, \quad (5.16)$$

and  $\mathbf{n}$  the norm of the face,  $ds$  the distance between the adjacent cell centres, and  $\mathbf{e}_s$  the unity vector along this direction. These parameters and viscous flux evaluation procedure are shown schematically in Figure 5.1 in two dimensions for visualization purposes.

#### 5.2.4 Time Integration Scheme

The non-linear system of ordinary differential equations resulting from the spatial discretization procedure given by (5.9) can be solved by applying a time-marching scheme to advance the solution forward in time. Currently the spatial discretization scheme uses linear reconstruction and is thus second-order accurate. To maintain a consistent scheme in terms of accuracy, the time-marching scheme should have the same order of accuracy. A wide variety of time-marching methods, such as explicit methods (Euler,  $n^{\text{th}}$  order Adams-Moulton, Runge-Kutta, etc.) and implicit methods (Euler, Trapezoidal, multi-step Runge-Kutta, etc.) can be used for different purposes. Reader is referred to consult CFD textbooks [49, 50, 51] on this subject. In this thesis research, an explicit two-stage Runge-Kutta time-marching scheme is used.

### 5.3 Multi-Block Structured Mesh

A flexible block-based hierarchical data structure has been developed for the mesh and is used in conjunction with the finite-volume scheme described above. The solution is calculated as

averages of hexahedrel computational cells which are embedded in structured blocks consisting of  $N_x \times N_y \times N_z$  cells, where  $N_x$ ,  $N_y$  and  $N_z$  are even but not necessarily equal integers. To allow data to be exchanged between adjacent blocks, having common interfaces, each block has an additional two (or more) layers of overlapping “ghost” cells, which contain solution information from neighbouring blocks. This happens by exchanging information of the “ghost” cells between adjacent blocks after each time step.

Since the mesh is block-based, the algorithm can run on a parallel computing cluster, where ideally each solution block gets assigned to one processor. After each time step the processors then communicate the information in the “ghost” cells to each other, so the solution can be updated on the different solution blocks in a simultaneous fashion.

# Chapter 6

# Implementation and Application of Discrete Explicit Filters

This chapter describes the implementation of the discrete explicit filters considered in Chapter 4, in the LES research code described in Chapter 5. The overall computational framework developed for applying the discrete explicit filtering is presented. This chapter describes the application of the discrete explicit filtering framework toLES prediction for homogeneous isotropic turbulence decay.

## 6.1 Implementation of Discrete Explicit Filters

### 6.1.1 Computational Framework for Discrete Explicit Filtering

Current implementation of the discrete explicit spatial filtering method has endeavoured to be a modular plug-in, that can be used where necessary, but at the same time preserve the parallel properties of the C++ object oriented research code and flow solver being developed within the CFD and Propulsion Group at UTIAS. From an object oriented programming perspective, the filters are defined as objects which are declared at the very base level, and initialized with two base level objects, which serve as data containers to organize and simplify data handling within the solution algorithm. The first base level object contains the non-solution specific data, and the second base level object comprises the solution specific data. Pointers to these base level objects are stored within the filter object so that the entire filtering-framework has access to these base level objects without using additional memory. This object class is shown in Figure 6.1. It provides the interface to the discrete explicit filters.

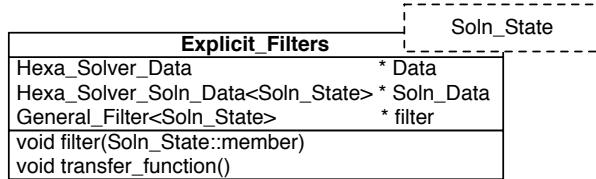


Figure 6.1: The `Explicit_Filters` class which provides the interface for the filtering framework.

The function “`void filter(Soln.State::member)`” has an argument indicating the type of the solution that is to be filtered. This could be a single variable, or one or more members of the solution vector. An external adaptor deals with the analysis of this argument to designate the proper values to  $\phi$  in (3.15) according to the argument with respect to the grid, where  $\phi$  is the solution content that is to be filtered. This function loops over the computational cells of each block in the multi-block grid (see Section 5.3) and orders the object “`General_Filter`” to filter the appropriate cell solution content.

The class “`General_Filter`” is constructed as an abstract class. This means that only a pointer to this class can be constructed, but that the instance must be created from inheriting classes. This makes it possible to use the same object, but referring to different filter types according to the user input. This is called polymorphism. Two classes inherit from this class, which are in turn also abstract: “`Discrete_Filters`” and “`Analytic_Filters`”. The same reasoning as for “`General_Filters`” also counts for these abstract classes as well. The inheriting classes of “`Discrete_Filters`” are non-abstract and describe the actual discrete filters. The same counts for the analytical filters. These relations are represented schematically in Figure 6.2.

Enough similarities exist between the different discrete filters and between the analytical filters to construct an abstract parent class for the discrete filters, and one for the analytical filters. The main difference in the discrete filters consists in the way the filter weights are constructed. Other properties, such as calculation of transfer function, filter-grid ratio, stencil gathering, etc., are the same and are bundled in the abstract parent class.

For discrete filters a separate class “`Neighbours`” is used to calculate the filtering-stencil for each cell. The above mentioned adaptor will then assign the values of  $\phi$  for each cell of the stencil. Finally these values are multiplied with the calculated filter weights and a sum is performed to return the filtered value  $\bar{\phi}$  for the cell.

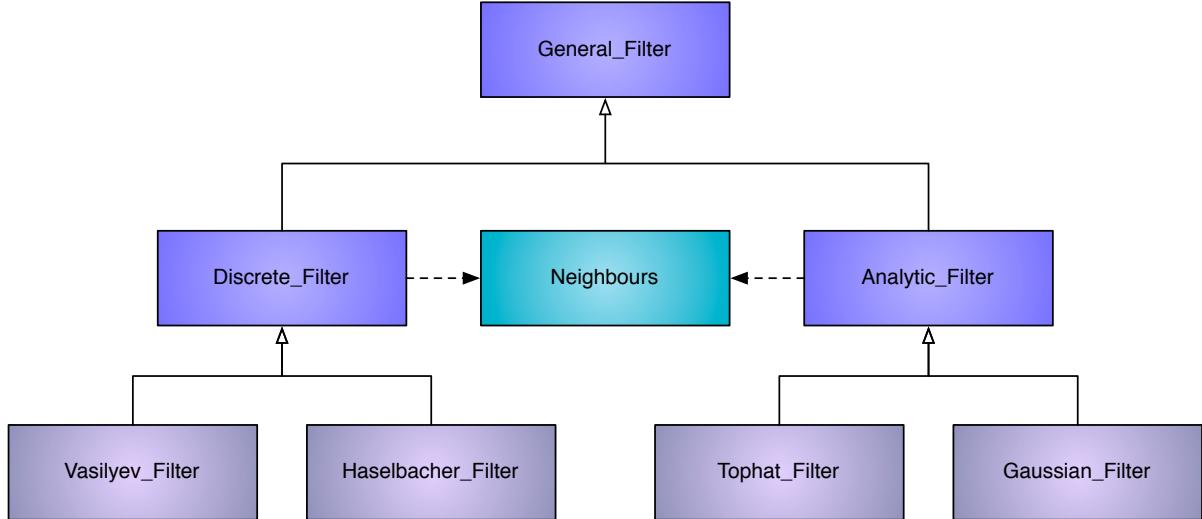


Figure 6.2: The currently implemented inheritance relations. In blue the abstract classes, in purple the different implemented filters. Solid-lined arrows represent a “inherits from” relation, while dashed arrows represent a “makes use of” relation.

### 6.1.2 Analysis of Approaches to Explicit Filters

The approaches to applying explicit filtering were described and summarized in Section 3.2.3. In this light a choice has to be made. To address this issue a summary of the requirements for each approach is made in what follows below.

#### Filtering the Non-Linear Terms

In this approach only selected terms, that will inject higher frequencies if left unfiltered, have to be filtered. Nevertheless, the number of terms to be filtered can be rather large. To illustrate this feature, the  $x$ -direction fluxes in (5.4) are repeated with the non-linear terms circled:

$$\mathbf{F} = \begin{pmatrix} \bar{\rho}\bar{u} \\ (\bar{\rho}\tilde{u}^2) + \tilde{p} \\ (\bar{\rho}\tilde{u}\tilde{v}) \\ (\bar{\rho}\tilde{u}\tilde{w}) \\ ((\rho\bar{E} + \bar{p})\bar{u}) \end{pmatrix}, \quad \mathbf{F}_\nu = \begin{pmatrix} 0 \\ \check{\sigma}_{xx} - (\tau_{xx}) \\ \check{\sigma}_{xy} - (\tau_{xy}) \\ \check{\sigma}_{xz} - (\tau_{xz}) \\ u(\check{\sigma}_{xx} - \tau_{xx}) + v(\check{\sigma}_{xy} - \tau_{xy}) + w(\check{\sigma}_{xz} - \tau_{xz}) - \check{q}_x - \check{q}_{T_x} \end{pmatrix}. \quad (6.1)$$

For the  $x$ -direction fluxes, there are 8 terms, with an additional 10 terms total from the fluxes in the  $y$ - and  $z$ -direction; this leads to a total of 18 non-linear terms that must be filtered. It follows from the finite-volume scheme, that these fluxes have to be evaluated at the cell inter-

faces, which means that a staggered grid would be required for the evaluation of these terms. The memory requirements for this approach are also tremendous, if of all 18 terms are filtered simultaneously (i.e., the terms are stored in a row vector and filtered as a whole). Note that the SFS stresses need no filtering when the Smagorinsky SFS model is applied [27]. Then 12 terms remain to be filtered.

### Filtering the Residuals

Alternatively, the full solution residuals can be filtered directly, which are calculated in the cell centres. The filtering of the solution update will limit high frequencies to enter the solution. The first element of the residual will not require filtering since there are no non-linear terms in the mass-conservation equation (2.36). This reduces the total number of terms to be filtered to 4. The memory requirements for this are not as much as for the above approach, and the filter can use the cell centred values in evaluating the filtered residuals.

It is clear that for the purpose of ease of implementation and efficiency, this second approach is the more logical and economical choice. However, as was apparent from applications, this method introduces some high frequency content over time in wavenumbers higher than the grid cut-off wavenumber. This is probably due to the fact that the explicit filter is not perfect (not a perfectly spherical transfer function or a not perfectly zero  $\hat{G}(\kappa_{\Delta})$ ). A simple solution to this problem is to apply an explicit filter to the actual fully updated solution after a certain number of time steps, with a larger FGR, so that the frequency content of the solution is not affected below the filter cut-off wavenumber of the explicit filter used to filter the residual.

### Choice of Filter-Grid Ratio

One of the open questions in the application of discrete explicit filters is the choice of the filter-grid ratio (FGR). In [52], Ghosal made an analysis on numerical errors in LES of turbulence, as did Chow et al. in [53] for more realistic turbulence kinetic energy spectra. They found through applying an explicit filter to a DNS calculation the *exact* SFS-force. Ghosal's results indicate that for a second-order finite-differencing scheme, the truncation error will overshadow the contribution of the SFS-force for  $\text{FGR} = 2$ . In order for the SFS-force to be significant enough, the FGR has to be increased to  $\text{FGR} \geq 4$ . For a fourth-order spatial discretization scheme, a filter-grid ratio of  $\text{FGR} = 2$  suffices. In practice, many LES codes use second-order spatial discretization schemes with a filter-grid ratio of unity or at best two. Some of the results of these LES codes are likely contaminated by significant numerical errors.

### Choice of CFL number

Using the second-order finite-volume scheme described in Chapter 5, with an explicit two-stage Runge-Kutta time integration scheme, and applying it to the decay of isotropic turbulence (in Section 6.2.3 below), allows for the use a CFL number of 0.5 in the case of implicit filtering. The limiting factor comes from the acoustic waves. Increasing the CFL number leads to instabilities originating at high frequencies. It is observed that this instability does not occur when using explicit filtering, as the high frequencies are eliminated by the filtering procedure. The limit of  $\text{CFL} = 1$  has been tested with explicit filtering (see Section 6.2.3 below), and agrees with the case of  $\text{CFL} = 0.5$ . This effect effectively eliminates a large part of the computational expenses associated with the explicit filtering, and might even lead to a slight speedup of the LES calculations.

## 6.2 Application of Discrete Explicit Filtering to Homogeneous Isotropic Turbulence

### 6.2.1 Synthetic Turbulent Flow Field

A homogeneous isotropic turbulence field is obtained by initializing a solution domain with Rogallo's random procedure [54], using the model spectrum proposed by Pope [6]. This model spectrum has already been introduced in Section 3.1.1, with the specifications described by Pope in [6].

$$E(\kappa) = C\varepsilon^{2/3}\kappa^{-5/3}f_L(\kappa L)f_\eta(\kappa\eta) \quad (6.2)$$

This spectrum has been initialized on a grid of  $128 \times 128 \times 128$  cells with the domain dimensions  $2\pi \times 2\pi \times 2\pi$  m. The latter might seem somewhat large for the current application, but the turbulent velocity field is scaled accordingly. These dimensions allow for a better comprehension of the spectral results since the one-dimensional grid cut-off wavenumber is then a whole number defined as

$$\kappa_\Delta = \frac{N}{2} \frac{2\pi}{D} = \frac{N}{2}, \quad (6.3)$$

with  $N$  an even number of cells in one direction, and  $D$  the size of the domain in the same direction. The spectrum is constructed so that the specific total kinetic energy  $K = 15000$ , and the characteristic length scale of the largest eddies  $l_0 = \frac{D}{3.2}$ . This fixes the other spectrum parameters, which are listed in Table 6.1. This table lists that the Kolmogorov wavenumber for this spectrum is of the order  $10^5$  m $^{-1}$ . If one wants to resolve this scale, the mesh requirements would obviously be enormous.

parameter	value	parameter	value
$K$	15000 $\text{m}^2/\text{s}^2$	$l_{11}$	0.85 m
$u_{\text{rms}}$	100 m/s	$\varepsilon$	$9.36 \cdot 10^3 \text{ m}^2/\text{s}^3$
$l_0$	1.96 m	$\eta$	$7.73 \cdot 10^{-6} \text{ m}$
$\kappa_0$	$3.2 \text{ m}^{-1}$	$\kappa_\eta$	$8.13 \cdot 10^5 \text{ m}^{-1}$

Table 6.1: Specifics of the Model Spectrum

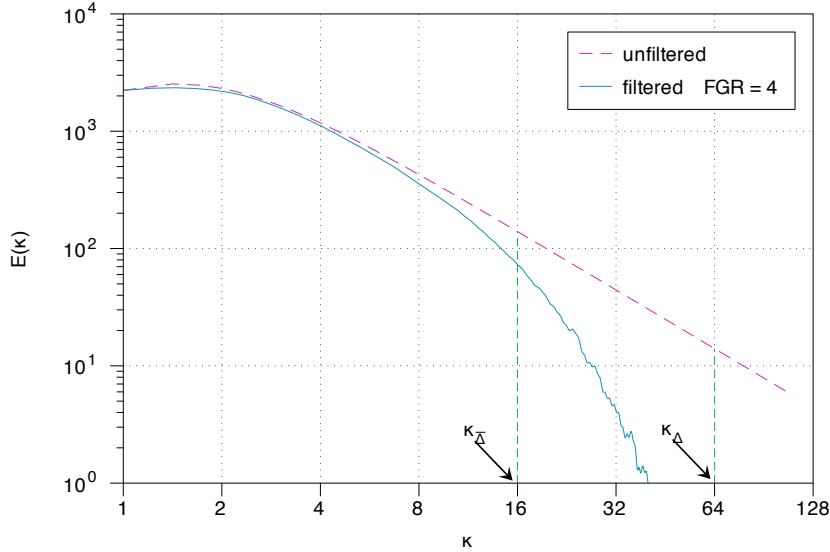


Figure 6.3: Application of explicit filtering to the Model Spectrum.

### 6.2.2 Application of Discrete Explicit Filter to Model Spectrum

The discrete explicit filtering operation is first applied to the solution provided by the synthetic turbulent field on the grid. The applied filter is a second-order linear constraints filter described in Chapters 3 and 4, with an additional constraint on the derivative of  $\hat{G}(\kappa_\Delta)$  and an additional constraint  $\text{FGR} = 4$ . Figure 6.3 shows the initial model spectrum and the application of this filter to this spectrum. The latter was obtained using discrete Fourier-analysis (Fast-Fourier Transforms) of the velocity field in the domain.

The identification of coherent vortices plays an important role in the analysis of turbulent flows. Most common definitions are associated with vortical motion (see [55] for a review) and turbulent structures are often exhibited by showing a positive iso-value of the criterion  $Q$  [56]:

$$Q = \frac{1}{2}(\Omega_{ij}\Omega_{ij} - S_{ij}S_{ij}) = \frac{1}{4}(|\vec{\omega}|^2 - 2S_{ij}S_{ij}), \quad (6.4)$$

with  $\vec{\omega}$  the vorticity vector, and  $S_{ij}$  and  $\Omega_{ij}$  the symmetric and antisymmetric parts of the

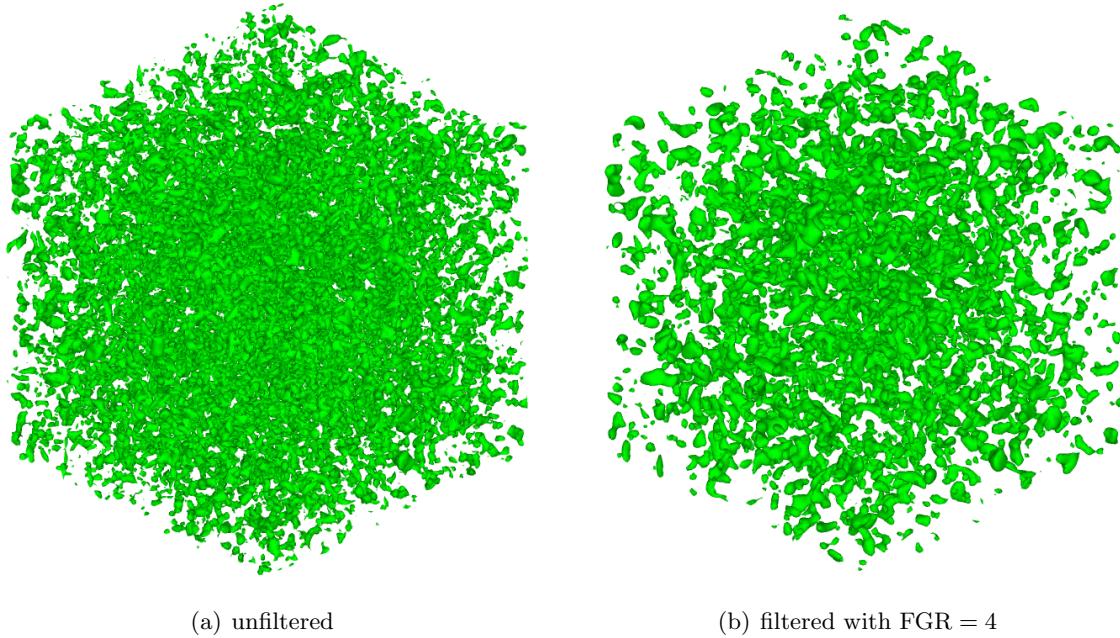


Figure 6.4: Q-criterion of filtered and unfiltered solution of the Model Spectrum.

velocity-gradient tensor  $\partial u_i / \partial x_j$  :

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad \Omega_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (6.5)$$

A positive iso-value of this Q-criterion is used to show the filtering effect on the model spectrum. Figure 6.4 shows the coherent turbulent structures for the unfiltered and the filtered Model Spectrum, with a FGR = 4. It is apparent from this figure that by applying the explicit filtering procedure the very small scale turbulent structures have been removed, while the larger turbulent eddies remain.

### 6.2.3 Application to Isotropic Turbulence Decay

No LES solution is valid if it cannot reproduce the correct behaviour of isotropic turbulence decay. Two main criteria are used to validate the LES solution.

- *Kolmogorov's -5/3 law.* If the turbulent kinetic energy spectrum  $E(\kappa)$  does not have a region that follows  $\kappa^{-5/3}$ , means that the LES simulation is not valid. This result is so important, that the coefficient  $C_s$  in the Smagorinsky model has been optimized to obtain this law behaviour.
- *Turbulence decay rate.* Experiments and DNS results agree that the rate at which isotropic

turbulence decays is proportional to  $t^{-1.25}$ . Convergence of the decay rate from the LES calculations to this value indicates the proper behaviour.

In this section, a study will be made of isotropic turbulence decay, with four different setups. They are as follows:

- Explicit filtering, coarse mesh ( $64 \times 64 \times 64$  cells), FGR = 2.
- Implicit filtering, coarse mesh ( $64 \times 64 \times 64$  cells), FGR = 2.
- Explicit filtering, fine mesh ( $128 \times 128 \times 128$  cells), FGR = 4.
- Implicit filtering, fine mesh ( $128 \times 128 \times 128$  cells), FGR = 4.

The domain for each setup is the same as used in Section 6.2.1 ( $2\pi \times 2\pi \times 2\pi$  m), with periodic boundary conditions. The fine mesh has half the mesh spacing of the coarse mesh, but the FGR for the fine mesh is chosen twice the mesh spacing of the coarse mesh, so that the filter width  $\bar{\Delta}$  remains constant for both the fine and the coarse mesh. For the explicit filtering procedure, a second-order linear constraints filter is used with one additional constraint on the derivative of  $\hat{G}(\kappa_{\bar{\Delta}})$  and an additional constraint on the FGR. For the SFS stress, the Smagorinsky model is used. It is assumed that this model equally acts as a filter as is described in Section 3.1.1. The spatial discretization scheme and time-marching scheme, used in the LES calculation are second-order, as outlined in Chapter 5.

The initial velocity field is initialized in the same manner as in the previous section. The difference is that now another energy spectrum is used for the initialization, which is comparable to an already explicitly filtered spectrum, so that every setup starts with exactly the same velocity field. The initial synthetic turbulent kinetic energy spectrum has been proposed by Haworth and Poinsot [57] and has the form

$$E(\kappa) = \frac{32}{2} \sqrt{\frac{2}{\pi}} \frac{u_{\text{rms}}^2}{\kappa_0} \left( \frac{\kappa}{\kappa_0} \right)^4 \exp \left( -2 \left( \frac{\kappa}{\kappa_0} \right)^2 \right). \quad (6.6)$$

This spectrum starts off with an asymptotic limit proportional to  $\kappa^4$  in contrast with the model spectrum above which starts off with an asymptotic limit proportional to  $\kappa^2$ . This slope is important as the spectrum decays differently for each of these cases, as is observed in [58, 59, 60]. The spectrum is initialized with  $u_{\text{rms}} = 100$  m/s and  $\kappa_0 = 3.2$  m $^{-1}$ , so that the filter cut-off wavenumber is greater than the wavenumber corresponding to the peak of this energy spectrum ( $\kappa_0$ ). The velocity field corresponding with this initial spectrum is illustrated in Figure 6.5.

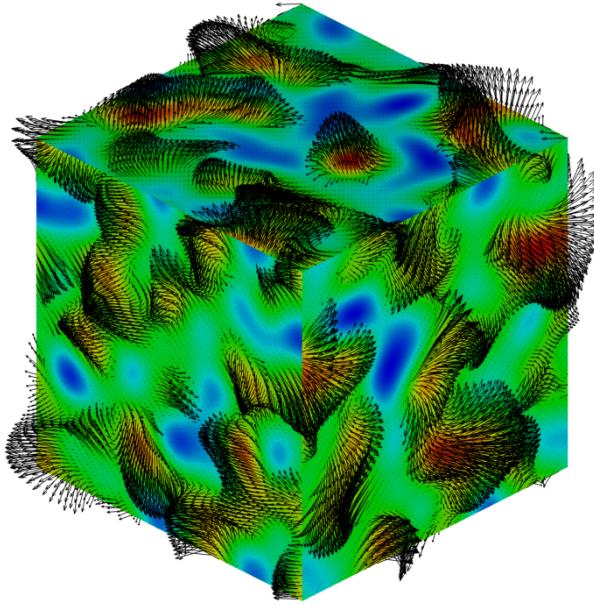


Figure 6.5: Initial velocity field. Contour values show the velocity component normal to the faces, arrows show the velocity vectors on the faces.

### Turbulent Kinetic Energy Spectrum

The initial turbulent kinetic energy spectrum as well as the turbulent kinetic energy spectra of the solution of these cases after 20 ms and 50 ms are depicted in Figures 6.6 and 6.7. The first figure shows the comparison for the explicit filtering cases between the fine and the coarse mesh. The second figure shows a comparison for the fine mesh between the explicit filtering case and the implicit filtering case. These energy spectra all show that the trend at which it decays is analogous to results in [58, 59, 60].

A first observation that can be made from Figure 6.6, is that as the time progresses, the initial spectrum decays and forms almost instantly the Kolmogorov's -5/3 law. This region grows larger as time progresses. After 50 ms this region is extended to  $(2 \lesssim \kappa \lesssim 8)$  for the finest mesh. For larger  $\kappa$ , the explicit filter and the SFS model start showing their effect by cutting off the spectrum. The fact that even though the filter width is the same for both meshes, the coarse mesh has more difficulty in keeping Kolmogorov's -5/3 law than the fine mesh. This is most likely related to the choice of the FGR, as discussed in Section 6.1.2, which states that the FGR for a second-order accurate scheme should be greater than four, so that the truncation errors of the numerical scheme don't overshadow the subfilter-terms. It remains to be observed if this still occurs using a fourth-order accurate scheme, as currently the research code is limited to second-order spatial accuracy.

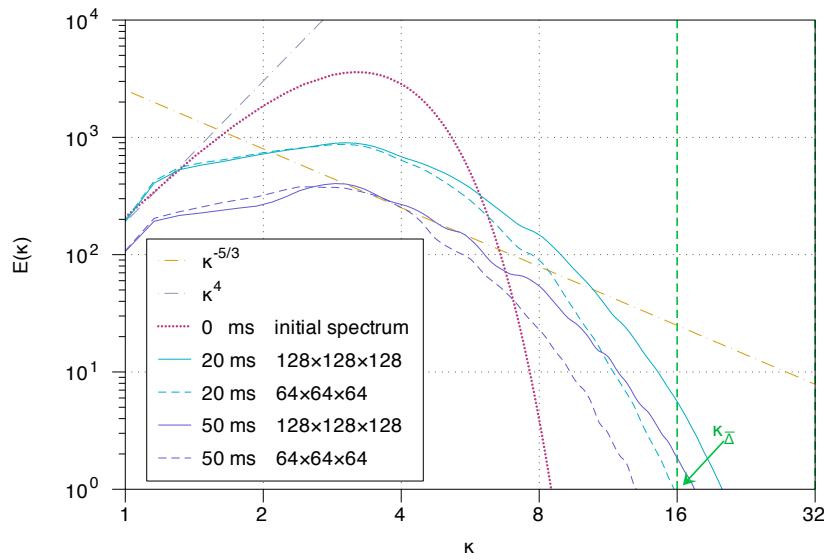


Figure 6.6: Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms using explicit filtering on a  $64 \times 64 \times 64$  mesh and a  $128 \times 128 \times 128$  mesh.

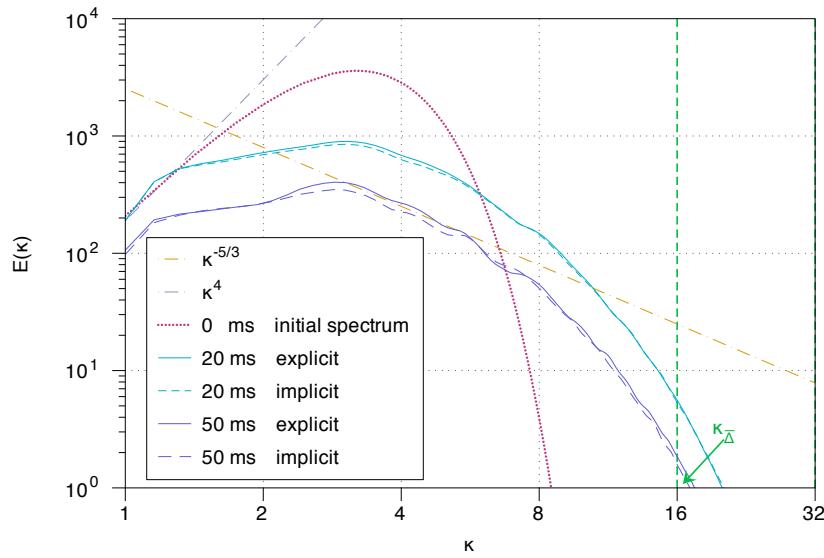


Figure 6.7: Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms on a  $128 \times 128 \times 128$  mesh using explicit filtering and implicit filtering.

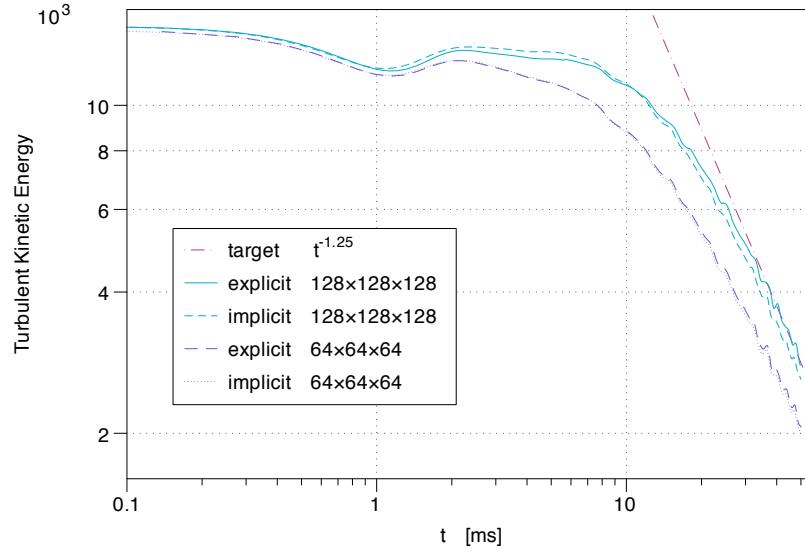


Figure 6.8: Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms on a  $128 \times 128 \times 128$  mesh using explicit filtering and implicit filtering.

A second observation can be made from Figure 6.7, where now the energy spectra obtained with implicit filtering is also shown. A comparison of the spectra obtained using explicit filtering and implicit filtering shows excellent agreement. This illustrates that the application of the explicit filter does not pollute the results obtained through implicit filtering. Furthermore this shows that implicit filtering equally cuts off at a FGR = 4, as was described in Section 3.1.1 for the Smagorinsky model. This is not surprising as an implicit filter is a second-order filter, and the *Smagorinsky filter* even acts like a second-order Gaussian filter (see Section 3.1.1 and [6]). The explicit filter used here is also of second-order and study of transfer functions showed that the shape of the filter is dictated mainly by this order (see Chapter 4).

### Turbulence Decay Rate

The turbulent kinetic energy can be plotted in time to observe the rate of its decay. Figure 6.8 shows the total turbulent kinetic energy, consisting of the sum of the resolved turbulent kinetic energy and the modelled subfilter turbulent kinetic energy, for all four cases. It follows from this figure that the coarse mesh transitions more rapidly to the correct asymptotic decay rate than the fine mesh. For the coarse mesh, the asymptotic decay rate is reached after 20 ms, while for the fine mesh, this decay rate is achieved after 40 ms. Again, as is the case when comparing the turbulent kinetic energy spectra, the implicit filtering cases and explicit filtering cases do not exhibit significant differences (as should be expected).

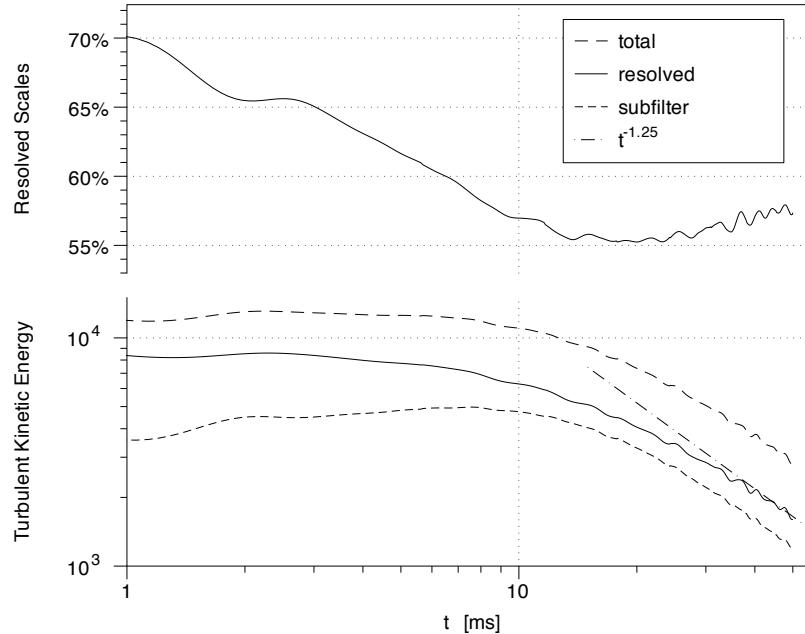


Figure 6.9: Turbulent kinetic energy spectra of isotropic turbulence decay at 20 ms and 50 ms on a  $128 \times 128 \times 128$  mesh using explicit filtering and implicit filtering.

Other interesting aspects that can be learnt by looking at the components of the total turbulent kinetic energy  $K$ . In Figure 6.9, the total, the resolved, and the SFS turbulent kinetic energy,  $k$ , are plotted as functions of simulated time, as well as the percentage of the turbulent kinetic energy that is resolved on the grid. What can be observed is that total turbulent kinetic energy remains relatively constant in the beginning, while the SFS turbulent kinetic energy increases and the resolved turbulent kinetic energy decreases (there is a transfer of energy from resolved to SFS-scales). This is because the initial spectrum already cuts off before filter cut-off wavenumber. The total kinetic energy thus migrates from larger scales to smaller scales, decreasing the resolved turbulent kinetic energy and increasing the SFS-kinetic energy. After about 10 ms, the SFS model starts dissipating the turbulent kinetic energy, and both components start decreasing. This figure also depicts the percentage of turbulent kinetic energy that is resolved on the grid. Before 10 ms, the resolved kinetic energy decreases. After 10 ms, the percentage of turbulent kinetic energy that is resolved remains fairly constant. This is expected, as the energy cascade dictates that what comes in, must go out (see Section 2.1.2).

### Solution Content

A final look at the turbulent structures for these four cases will indicate whether the LES solutions are not just the same in terms of the turbulence decay rate and the turbulent kinetic energy spectrum, but also in terms of the predicted solution content. Figure 6.10 shows the turbulent structures for both implicit and explicit filtering on both the coarse and the fine mesh, using positive iso-values of the Q-criterion. From this figure, it would seem that all four solutions are in fairly good agreement. The fine mesh shows finer structures, most likely because of the above mentioned issue of the FGR of two on a second-order discretization scheme, and because of the coarser mesh resolution.

Typical for isotropic turbulence decay is the phenomenon called vortex stretching. This can also be observed via the Q-criterion. Figure 6.10 clearly shows these elongated or stretched turbulent structures, as was expected from both DNS and experimental results [3].



Figure 6.10: Coherent turbulence structures for application of isotropic turbulence decay after 50ms, for both explicit and implicit filtering, on a  $64 \times 64 \times 64$  and a  $128 \times 128 \times 128$  mesh. Visualization occurs through a positive isovalue of the Q-criterion.

## Chapter 7

# Conclusions and Future Research

### 7.1 Concluding Remarks

This thesis research has attempted to address the issues associated with filtering errors that may arise when using LES with implicit filtering, by considering an explicit filtering approach. Explicit filtering can be used to decouple the filter width from the mesh spacing, to control commutation errors on grids of varying mesh width, and thereby to achieve grid-independent solutions of the filtered Navier-Stokes equations. In this thesis, a study of different explicit filters has been made. The emphasis was laid on discrete explicit filters, as they are more computationally efficient than analytical filters for general grid topologies.

Both linear constraints and least-squares reconstruction filters were studied in detail, assessed, and implemented in a code for performing LES of compressible turbulent flows as described in Chapter 5. The first part of the study was concerned with the transfer functions of these filters, as described in Chapter 4. Procedures were outlined to allow the construction of filters with the desired behaviour. The filters were then assessed to verify this desired behaviour is translated correctly to the actual behaviour. Finally the filtering procedure has been implemented for LES of compressible turbulent flows.

Promising results have been obtained using the explicit filtering procedure for the decay of isotropic homogeneous turbulence on a cubic domain with periodic boundaries. It has been demonstrated that explicit filtering does not affect results on a uniform mesh, showing that no errors are introduced with this procedure. A surprising element of the numerical results is also that, although explicit filtering increases the computational cost per time step, larger time steps can be taken without leading to instabilities, thus offsetting a large part of the disadvantages

of the incurred higher computational costs of explicit filtering methods.

## 7.2 Contributions

In the context of this thesis research, a computational framework for designing and performing explicit filtering has been constructed in a modular way, allowing the user to readily implement other explicit filters (see Chapter 6). The framework has been built using an easy to use interface, so that implementation in future research will be straightforward.

Another framework has been constructed to generate the initial turbulent velocity field and analyze its turbulent kinetic energy spectrum. This is another essential ingredient in the assessment of solutions of turbulence flows.

## 7.3 Future Research

In Chapter 6, explicit filtering has been applied to LES of a compressible turbulent flow on a uniform mesh, using a second-order accurate scheme. In order to obtain a clear advantage of explicit filtering over implicit filtering, higher-order schemes are required, on non-uniform meshes, as commutation errors will then play an important role. Currently, higher-order schemes have not been implemented in the current version of the research code, but the explicit filtering framework can be readily applied when higher-order schemes become available. Another aspect that remains to be studied is the use of explicit filtering with an adaptive mesh refinement technique that is available in the research code described in Chapter 5.

# References

- [1] Moin, P. and Mahesh, K., “DIRECT NUMERICAL SIMULATION: A Tool in Turbulence Research,” *Annual Reviews in Fluid Mechanics*, Jan 1998.
- [2] Wilcox, D. C., *Turbulence Modeling for CFD*, DCW Industries, Inc., 3rd ed., 2006.
- [3] Piomelli, U., “Large-eddy simulation: achievements and challenges,” *Progress in Aerospace Sciences*, Jan 1999.
- [4] Piomelli, U., “Large-Eddy and Direct Simulation of Turbulent Flows,” *von Karman Institute for Fluid Dynamics Lecture Series*, Aug 2000, pp. 74.
- [5] Brown, G. and Roshko, A., “On density effects and large structure in turbulent mixing layers,” *Journal of Fluid Mechanics Digital Archive*, Jan 1974.
- [6] Pope, S. B., *Turbulent Flows*, Cambridge University Press, 2000.
- [7] Vreman, B., Geurts, B., and Kuerten, H., “A priori tests of large eddy simulation of the compressible plane mixing layer,” *Journal of Engineering Mathematics*, Jan 1995.
- [8] Vreman, A., Geurts, B., and Kuerten, J., “Subgrid-modelling in LES of Compressible Flow,” *Direct and Large-Eddy Simulation I: Selected Papers from the ...*, Jan 1994.
- [9] Martin, M. P., Piomelli, U., and Candler, G. V., “Subgrid-scale models for compressible large eddy simulations,” *Theoretical and Computational Fluid Dynamics*, Vol. 13, 2000, pp. 361–376.
- [10] Carati, D., Winckelmans, G., and Jeanmart, H., “On the modelling of the subgrid-scale and filtered-scale stress tensors in large-eddy simulation,” *J. Fluid Mech.*, Jan 2001.
- [11] Smagorinsky, J., “General Circulation Experiments with the Primitive Equations,” *Monthly Weather Review*, Vol. 91, No. 3, 1963, pp. 99–164.

- [12] Deardorff, J., "A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers," *J. Fluid Mech.*, Vol. 41, Jan 1970, pp. 453–480.
- [13] Yoshizawa, A. and Horiuti, K., "A Statistically-Derived Subgrid-Scale Kinetic Energy Model for the Large-Eddy Simulation of Turbulent Flows," *Journal of the Physical Society of Japan*, Jan 1985.
- [14] Germano, M., Piomelli, U., Moin, P., and Cabot, W., "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A: Fluid Dynamics*, Jan 1991.
- [15] Martín, M. P., Piomelli, U., and Candler, G. V., "Subgrid-Scale Models for Compressible Large-Eddy Simulations," *Theoretical and Computational Fluid Dynamics*, 2000.
- [16] Lilly, D., "A proposed modification of the Germano subgrid-scale closure method," *Physics of Fluids A: Fluid Dynamics*, Jan 1992.
- [17] Menon, S. and Kim, W., "A New Dynamic One-Equation Subgrid Model for Large-Eddy-Simulations," *AIAA Paper*, 1995.
- [18] Lenormand, E. and Sagaut, P., "Subgrid-Scale Model for Large-Eddy Simulations of Compressible Wall Bounded Flows," *AIAA Journal*, Vol. 38, No. 8, 2000.
- [19] Knight, D., Zhou, G., Okong'o, N., and Shukla, V., AIAA, 1998.
- [20] Rogallo, R. and Moin, P., "Numerical Simulation of Turbulent Flow," *Ann. Rev. Fluid Mech.*, Vol. 16, 1984, pp. 99–137.
- [21] Lund, T. S., "On the use of discrete filters for large eddy simulation," Center for Turbulence Research, Annual Research Briefs, 1997.
- [22] Berselli, L., Grisanti, C., and John, V., "On commutation errors in the derivation of the space averaged Navier-Stokes equations," *Preprint*, 2004.
- [23] Vasilyev, O. and Lund, T., "A general theory of discrete filtering for LES in complex geometry," *Annual Research Briefs*, Jan 1997.
- [24] Sagaut, P. and Meneveau, C., *Large Eddy Simulation for Incompressible Flows*, Springer-Verlag Berlin Heidelberg, 1998.
- [25] Marsden, A., "Construction of Commutative Filters for LES on Unstructured Meshes," *Journal of Computational Physics*, Vol. 175, No. 2, Jan 2002, pp. 584–603.
- [26] Lund, T., "On the use of discrete filters for large eddy simulation," *Center for Turbulence Research*, 1997.

- [27] Brandt, T., "Comparison between Approaches to Explicit Filtering in Large Eddy Simulation," *Journal of Structural Mechanics*, Jan 2005.
- [28] Gullbrand, J., "Explicit filtering and subgrid-scale models in turbulent channel flow," *Center for Turbulence Research*, Jan 2001.
- [29] Brandt, T., "Study of Large Eddy Simulation and Smagorinsky Model using Explicit Filtering," *36th AIAA Fluid Dynamics Conference and Exhibit*, Jan 2006.
- [30] van der Ven, H., "A family of large eddy simulation (LES) filters with nonuniform filter widths," *Physics of Fluids*, Jan 1995.
- [31] Vasilyev, O., Lund, T., and Moin, P., "A general class of commutative filters for LES in complex geometries," *Journal of Computational Physics*, Jan 1998.
- [32] Haselbacher, A., Moser, R., and Constantinescu, G., "Toward optimal LES on unstructured meshes," *Proceedings of the Summer Program*, Jan 2002.
- [33] Ciardi, M., Sagaut, P., Klein, M., and Dawes, W., "A dynamic finite volume scheme for large-eddy simulation on unstructured grids," *Journal of Computational Physics*, Vol. 210, No. 2, Dec 2005, pp. 632–655.
- [34] Sagaut, P. and Grohens, R., "Discrete filters for large eddy simulation," *International Journal for Numerical Methods in Fluids*, Jan 1999.
- [35] Stolz, S. and Adams, N., "An approximate deconvolution procedure for large-eddy simulation," *Physics of Fluids*, Jan 1999.
- [36] Stolz, S., Adams, N. A., and Kleiser, L., "An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows," *Physics of Fluids*, Vol. 13, No. 4, Jan 2001, pp. 997.
- [37] Mathew, J., "An explicit filtering method for large eddy simulation of compressible flows," *Physics of Fluids*, Jan 2003.
- [38] Mathew, J., Foysi, H., and Friedrich, R., "A new approach to LES based on explicit filtering," *International Journal of Heat and Fluid Flow*, Mar 2006, pp. 9.
- [39] Lin, W., Hernández-Pérez, F. E., Groth, C. P. T., and Gülder, Ö. L., "Comparison of Subfilter Scale Models for LES of Turbulent Premixed Flames," *AIAA Paper*, 2008.
- [40] Gottlieb, J. and Groth, C., "Assessment of Riemann solvers for unsteady one-dimensional inviscid flows for perfect gases," *Journal of Computational Physics*, Jan 1988.

- [41] Roe, P., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, Oct 1981, pp. 357–372.
- [42] Einfeldt, B., "On Godunov-Type Methods for Gas Dynamics," *SIAM Journal on Numerical Analysis*, Jan 1988.
- [43] Liou, M., "A sequel to AUSM, Part II: AUSM+-up for all speeds," *Journal of Computational Physics*, Jan 2006.
- [44] Barth, T., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," *AIAA Paper*, Jan 1993.
- [45] Waterson, N. and Deconinck, H., "Design principles for bounded higher-order convection schemes—a unified approach," *Journal of Computational Physics*, Jan 2007.
- [46] Barth, T. J., Jespersen, J., and Dennis, C., "The design and application of upwind schemes on unstructured meshes," *AIAA Paper*, 1989.
- [47] Mathur, S. and Murthy, J., "A Pressure-Based Method for Unstructured Meshes," *Numerical Heat Transfer*, Jan 1997.
- [48] Gao, X., *Parallel Solution-Adaptive Method for Turbulent Combusting Flows*, Phd thesis, University of Toronto, 2008.
- [49] Lomax, H., Pulliam, T., and Zingg, D., *Fundamentals of Computational Fluid Dynamics*, Springer-Verlag Berlin Heidelberg, 2001.
- [50] Hirsch, C., *Numerical Computation of Internal and External Flows, Volume 1*, John Wiley & Sons, Toronto, 1989.
- [51] Hirsch, C., *Numerical Computation of Internal and External Flows, Volume 2*, John Wiley & Sons, Toronto, 1990.
- [52] Ghosal, S., "An Analysis of Numerical Errors in Large-Eddy Simulations of Turbulence," *Journal of Computational Physics*, Jan 1996.
- [53] Chow, F. and Moin, P., "A further study of numerical errors in large-eddy simulations," *Journal of Computational Physics*, Jan 2003.
- [54] Rogallo, R., "Numerical Experiments in Homogeneous Turbulence," *NASA Technical Memorandum*, Jan 1981.
- [55] Dubief, Y., "On coherent-vortex identification in turbulence," *J. of Turbulence*, Jan 2000.

- [56] Hunt, J., Kaimal, J., and Gaynor, J., “Eddy structure in the convective boundary layer: new measurements and new concepts,” *Quarterly Journal of the Royal Meteorological Society*, Jan 1988.
- [57] Haworth, D. C. and Poinsot, T. J., “Numerical simulations of Lewis number effects in turbulent premixed flames,” *J. Fluid Mech.*, Jan 1992.
- [58] Vreman, A., Geurts, B., Kuerten, J., and Zandbergen, P., “A finite volume approach to large eddy simulation of compressible, homogeneous, isotropic, decaying turbulence,” *International Journal for Numerical Methods in Fluids*, Jan 1992.
- [59] Chasnov, J., “Similarity states of passive scalar transport in isotropic turbulence,” *Physics of Fluids*, Jan 1994.
- [60] Yu, H., Girimaji, S., and Luo, L., “DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method,” *Journal of Computational Physics*, Jan 2005.