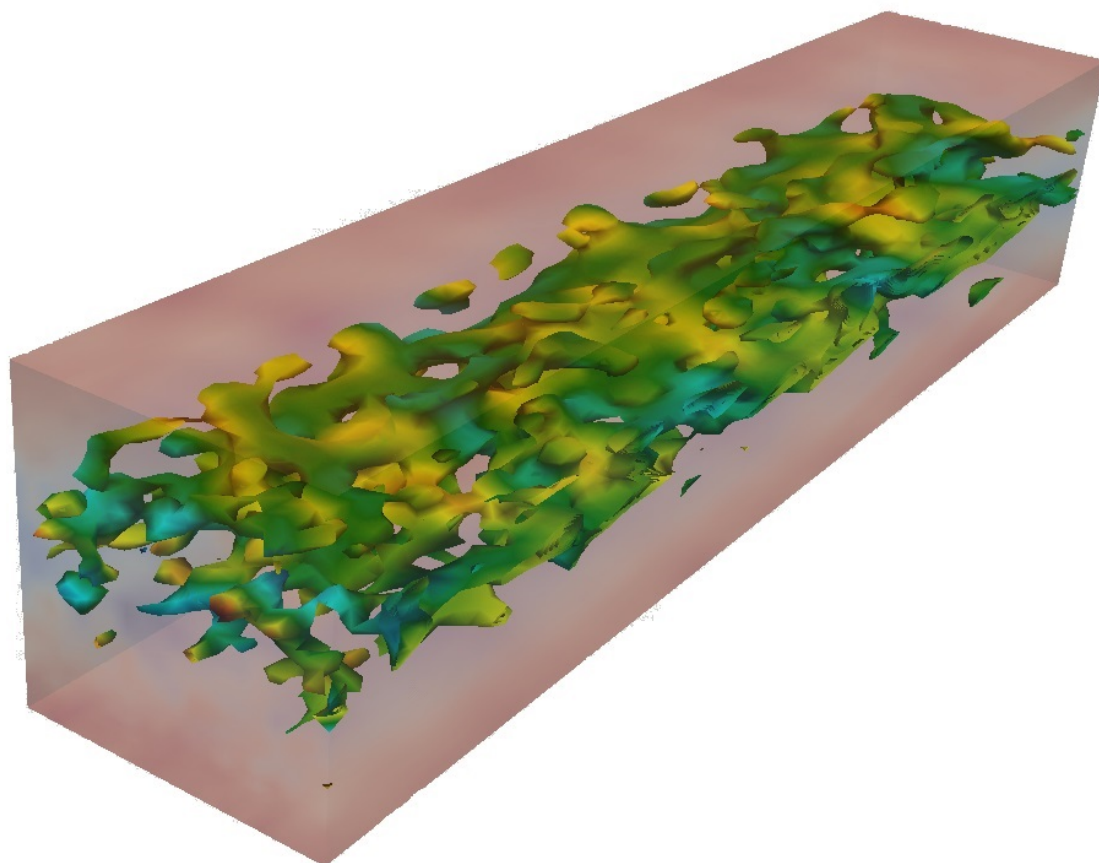




CHALMERS



Synthetic Inlet Boundary Conditions for LES

Master's thesis in Engineering Mathematics and Computational Science

JIM PERSSON

MASTER'S THESIS IN ENGINEERING MATHEMATICS AND COMPUTATIONAL SCIENCE

Synthetic Inlet Boundary Conditions for LES

JIM PERSSON

Department of Applied Mechanics
Division of Fluid Dynamics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

Synthetic Inlet Boundary Conditions for LES
JIM PERSSON

© JIM PERSSON, 2015

Master's thesis 2015:04
ISSN 1652-8557
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Synthetically generated turbulence in a channel.

Chalmers Reproservice
Göteborg, Sweden 2015

Synthetic Inlet Boundary Conditions for LES
Master's thesis in Engineering Mathematics and Computational Science
JIM PERSSON
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology

ABSTRACT

This thesis describes the analysis of four different methods for generating synthetic turbulence, and the implementation of the methods into HYDRA CFD Code. These methods are the synthetic eddy method (SEM) by Jarrin, divergence free synthetic eddy method (DFSEM) by Poletto, synthetic fluctuations by Davidson, and a spectral method for generating fluctuations by Batten. The SEM and the DFSEM are stochastic algorithms that use the view of turbulence as a superposition of eddies, whereas the last two methods use the Fourier space to describe turbulence in terms of wavenumbers.

The motivation of this work is the growing interest of using unsteady simulations, in particular large eddy simulations (LES), in engineering applications, and the computational issues that come with it. In order to use LES, instantaneous inlet velocities are needed as turbulent inflow boundary conditions. These boundary conditions will be given by synthetic turbulence, generated by the four methods mentioned. The object is to trigger the equations to start resolve turbulence.

The methods are first implemented in C and investigated in terms of correlation in space and time for the generated turbulence. It is shown that all of the methods generated fluctuations with proper correlation in time and space. Next, the SEM, the DFSEM, and the synthetic fluctuations method are implemented into the CFD programming language HYDRA CFD Code. Last, the implementation of the SEM and the DFSEM is further investigated through a channel flow simulation in HYDRA, where the two methods are used for generating inlet conditions. The results from these simulations show that the SEM and the DFSEM produces fluctuations that remain throughout the channel.

Keywords: LES, SEM, DFSEM, synthetic fluctuations, inlet boundary conditions, turbulence, HYDRA CFD Code

PREFACE

The present thesis has been made as the final part for the Master of Science in Mathematics at Chalmers University of Technology. The work was carried out during spring and summer 2015 at Rolls-Royce Plc in Derby, United Kingdom. I would like to thank my supervisor Luigi Capone at Rolls-Royce Plc for all the support, discussions and encouragement throughout the work. I would also like to thank Rob Watson at Cambridge for his support and guidance, as well as giving feedback on the implemented methods. Special thanks also to my examiner at Chalmers, Lars Davidson for the help and advice during the project.

NOMENCLATURE

Acronyms

CFD	Computational Fluid Dynamics
DFSEM	Divergence Free Synthetic Eddy Method
DNS	Direct Numerical Simulation
EARSM	Explicit Algebraic Reynolds Stress Model
KEP	Kinetic Energy Perserving
LES	Large Eddy Simulation
MUSCL	Monotone Upwind Schemes for Scalar Conservation Laws
RANS	Reynolds-averaged Navier-Stokes
SEM	Synthetic Eddy Method
SGS	Sub-Grid Scale

Greek symbols

α	Constant
α^n	Random angle
α^k	Intensity
β	Coefficient
Γ	Length scale ratio
δ_{ij}	Kronecker's delta
ε	Dissipation rate
ϵ_{ijk}	Levi-Civita symbol
θ	Angle
κ	Wave number
μ	Dynamic viscosity
ν	Kinematic viscosity
ν_t	Turbulent kinematic viscosity
ν_{SGS}	Sub-Grid Scales turbulent viscosity
ξ	Random coefficient
ρ	Density
σ_i	Length scale, i th direction
σ^n	Unit vector
τ	Viscous stress
τ	Time scale
φ	Random angle
ω	Frequency
Δ	Grid spacing
Ψ	Phase
Ω	Vorticity tensor

Roman symbols

a_{ij}	Cholesky decomposition of the Reynolds stress tensor
b_{ij}	Reynolds stress anisotropy tensor
dx_i	Grid spacing in x_i -direction
dt	Time step
f_i	Body forces, i th component
k	Turbulent kinetic energy
L_t	Turbulent length scale
p	Pressure
S_{ij}	Strain rate tensor
t	Time
u_i	Velocity vector, i th component
x_i	Spatial coordinate, i th component
\mathcal{T}	Inlet time scale
\mathcal{U}'	Time filtered velocity in x_1 -direction
\mathcal{V}'	Time filtered velocity in x_2 -direction
\mathcal{W}'	Time filtered velocity in x_3 -direction

CONTENTS

Abstract	i
Preface	iii
Nomenclature	v
Contents	vii
1 Introduction	1
1.1 Background	1
1.2 Purpose	2
1.3 Limitations	2
1.4 Method	2
2 Theory	3
2.1 Governing equations	3
2.1.1 Large Eddy Simulation	3
2.2 Spectral method	4
2.3 Synthetic fluctuations	5
2.4 Synthetic eddy method	9
2.5 Divergence free synthetic eddy method	10
2.6 Calculating the Reynolds stress tensor	12
2.7 General method for generating synthetic turbulence	14
2.8 Turbulent statistics	14
2.8.1 Two-point correlation	14
2.8.2 Autocorrelation	15
2.9 HYDRA	15
3 Numerical method	17
3.1 Setup for computing correlations	17
3.2 Channel flow	17
4 Results	19
4.1 Correlations	19
4.2 Channel flow	19
5 Conclusions	26
References	27
A Correlations	29

1 Introduction

Describing the motion of fluids, such as air or water, mathematically is of great importance in many areas, both in academics as well as in the industry. However, even though the importance of describing flow correctly is great, the Navier-Stokes equations that are used to describe the flow are not generally analytically solvable. Hence, numerical methods are used to simulate the flow in order to get a good understanding of how the flow behaves in a specific situation.

1.1 Background

The most commonly used method for describing flows in industrial applications, is Computational Fluid Dynamics (CFD), where the finite volume method is used in order to numerically solve the Navier-Stokes equations. However, even though the powers of computers are steadily increasing, solving Navier-Stokes equations directly, a so called direct numerical simulation (DNS), is still far too computationally heavy for most cases. Instead of solving the equations directly, there are different methods for modelling the Navier-Stokes equations, such that computers are able to solve the equations in a reasonable amount of time, of which two are of most interest in this thesis. The first one is Reynolds-averaged Navier-Stokes (RANS), which relies only on the mean motion and uses models to describe turbulence. The second model is Large Eddy Simulation (LES), which resolves large turbulent eddies while the smaller ones are simulated.

Compared to DNS and LES, RANS is quick and gives good enough results for it to be widely used in industrial applications. However, with increasing computer power, LES is used more and more to simulate spatially developing flows. Using LES to simulate spatially developing flows requires the specification of instantaneous turbulent inlet boundary conditions for triggering the equations to start resolve turbulence. Hence, in order to use LES in industrial cases, there is a need for methods to generate realistic inflow conditions.

One method to achieve proper inlet conditions, is to obtain inflow data from a precursor simulation. However, as this is computationally heavy, more efficient methods are sought for. A second alternative method is to use recycling methods which have been used successfully by Kaltenbach et al. in [1] and Aider and Danet in [2], amongst others. A third alternative to achieve proper inlet conditions, is to generate synthetic turbulence. There are different methods for generating synthetic turbulence, with the same goal of generating turbulence that resembles the actual flow by matching a reduced set of statistics. It should be noted that using ordinary white noise as turbulent inflow is not satisfactory, since the fluctuations from a white noise is totally uncorrelated in space and time, which will make the fluctuations disappear too soon.

The present thesis focuses on the implementation of four methods for generation of synthetic turbulence. The first method was presented by Batten et al. in [3], where the turbulence is described in Fourier space, using the notion of wavenumbers to describe the fluctuations. The second method also uses the Fourier space to describe the turbulence, and is presented by Davidson in [4]. The last two methods are the synthetic eddy method (SEM) by Jarrin and the divergence free synthetic eddy method (DFSEM) by Poletto, presented in [5] and [6] respectively. These methods are very similar and view turbulence as a superposition of eddies.

The thesis presents these four methods and how the accuracy of each of the methods is examined. This is first done through investigating the correlations of the generated turbulence in space and time, using autocorrelation in time and two-point correlation in space. It is shown in [7] that

examining correlations is the preferable way of measuring the generated turbulence compared to other methods, such as inspecting the energy spectrum of the generated velocity signal. Next, the methods are integrated into HYDRA CFD Code, which is a CFD code first developed by University of Oxford and customised and further developed by Rolls-Royce Plc for turbo-machines applications. HYDRA is designed as a coupled solver, since the aim is mainly to solve for high speed flows in ideal gases. A coupled solver is considered suitable for those kinds of flows.

In HYDRA, the SEM and the DFSEM are further investigated and used to produce inlet boundary conditions in a channel flow simulation. It should be noted that the methods could be used in the interface between the RANS region and the LES region in a hybrid LES-RANS simulation as well [8], even though this is not investigated in the present thesis.

Even though this thesis focuses on four methods for generating synthetic turbulence, there are other methods that have been used successfully. In [9], Hanna et al. generated one-dimensional time series of inflow data based on an exponential correlation function to simulate flows over an array of cubes using LES. The efficiency of this method was shown to be very high, however, the accuracy was limited due to that there were no spatial correlation imposed at the inlet. In [10], Klein et al. developed a technique generating synthetic velocities as inflow data for jet flows, which reproduced first and second order one-point statistics as well as locally given correlations. The technique is based on the knowledge that for late stage homogeneous turbulence the correlation function takes a Gaussian form.

1.2 Purpose

The purpose of the present project is to implement four different methods for generating turbulent flow into the programming language C, and then integrate three of the methods into HYDRA CFD Code. In HYDRA, the SEM and the DFSEM will be tested on a simple test case, in form of a channel flow. This test case is used for its simplicity, and the challenges that arise with the constant influence from the walls.

1.3 Limitations

There are some limitations that need to be addressed. As there were some time constraints to the project, only the methods by Davidson, Jarrin, and Poletto will be integrated into HYDRA, and only the SEM and the DFSEM will be used in a channel flow simulation. The methods will not be tested on complex geometries, but only the simple test case already mentioned. This will keep the test work to a minimum, as well as giving a clear view of how well the methods perform. When implementing the models, the flow will be considered to be incompressible. Many flows for which the methods will be used are almost incompressible, hence this might not be a huge limitation but it deserves to be mentioned.

1.4 Method

The project work began with literature studies in order to determine which methods to implement, and how this was to be done. Next, the methods were implemented into the programming language C, where input to the methods was taken from a RANS simulation. The plotting and result analysis was done in Matlab. The methods were finally integrated into HYDRA CFD Code, where the SEM and the DFSEM were used in an LES simulation of a channel flow.

2 Theory

This section presents the most important theoretical aspects of the work done in the present project. Starting with a short presentation of the flow equations of interest, followed by a description of the four different methods for generating synthetic turbulence, and how these methods are implemented and analysed.

2.1 Governing equations

The motion of fluids are described by Navier-Stokes equations. The first equation that is part of the Navier-Stokes equations is given by the continuity equation:

$$\frac{d\rho}{dt} + \rho \frac{\partial u_i}{\partial x_i} = 0, \quad (2.1)$$

where ρ is the density of the fluid and u_i is the velocity vector. In the present project, the flow is assumed to be incompressible, that is $\rho = \text{constant}$, which means that Equation (2.1) can be rewritten as:

$$\frac{\partial u_i}{\partial x_i} = 0. \quad (2.2)$$

The next equation is the momentum equation which is given by:

$$\rho \frac{du_i}{dt} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(2\mu S_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) + \rho f_i, \quad (2.3)$$

where μ denotes the dynamic viscosity, p is the pressure, and u_i denotes the velocity vector. In the second step, the viscous stress τ_{ij} was rewritten as:

$$\tau_{ij} = \left(2\mu S_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \right), \quad (2.4)$$

where the strain rate tensor S_{ij} is defined as:

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (2.5)$$

and the second term in Equation (2.4) is zero due to Equation (2.2).

Solving Equation (2.2) and Equation (2.3) directly, that is using a DNS, is a tedious work and not applicable in most industrial applications due to the turbulent behaviour of the flow. Turbulent flow can be viewed upon as flow consisting of eddies of different scales of velocity, length, and time. Solving all scales directly demands a very fine mesh resolution, which in most cases results in too demanding computations for any wider application. Hence, the flow is usually modelled by using a turbulence model.

2.1.1 Large Eddy Simulation

Modelling all turbulent scales, as done in RANS, gives results that, to some extent, resembles the actual flow. Further, modelling no turbulence at all and solve the Navier-Stokes equations with DNS is computationally too heavy. While RANS models all of the turbulent scales, LES models only the smaller scales, while the larger scales of the turbulence are resolved. As it is the smaller scales that are the heaviest scales to compute, this will make LES less computationally heavy than DNS, but

still heavier than RANS.

When formulating the equations used in RANS, the velocity and the pressure terms are decomposed into mean and fluctuating values, and the Navier-Stokes equations are time averaged. When using LES the Navier-Stokes equations are instead volume averaged, which results in equations that are dependent on both space and time. The equations become:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(\bar{u}_i \bar{u}_j) = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left((\nu + \nu_{sgs}) \frac{\partial \bar{u}_i}{\partial x_j} \right), \quad (2.6)$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0. \quad (2.7)$$

The large, time dependent turbulence is part of the filtered velocity and pressure terms \bar{u}_i and \bar{p} . The term $\frac{\partial \tau_{ij}}{\partial x_j}$ includes the Reynolds stresses of the small eddies, which are called sub-grid stresses (SGS). This term is modelled with an SGS-turbulent viscosity ν_{sgs} in the second step. ν_{sgs} includes only the effects of small eddies.

In order to use LES, proper inlet data is required. Using synthetic turbulence as inflow was investigated by Davidson and Billson in [11], where it is shown that the imposed fluctuations considerably improve the results in a fully developed channel flow. There are different methods for generating synthetic turbulence, where this project focuses on four different methods. A spectral method developed by Batten, presented in [3], synthetic turbulence, presented by Davidson in [4], the SEM presented by Jarrin in [5], and the DFSEM, presented by Poletto et.al in [6].

2.2 Spectral method

It is a well known fact that every turbulent signal can be expressed as a series of sines and cosines [12]. In general, any periodic function, g , with a period of $2L$ can be written as a Fourier series as:

$$g(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(\kappa_n x) + b_n \sin(\kappa_n x)), \quad (2.8)$$

where x is the spatial coordinate and κ_n is the wave number, given by

$$\kappa_n = \frac{n\pi}{L}. \quad (2.9)$$

The Fourier coefficients are given by

$$a_n = \frac{1}{L} \int_{-L}^L g(x) \cos(\kappa_n x) dx, \quad (2.10)$$

$$b_n = \frac{1}{L} \int_{-L}^L g(x) \sin(\kappa_n x) dx. \quad (2.11)$$

Working with a Fourier decomposition in order to generate synthetic turbulence was first explored by Kraichnan in [13], which was further developed by Batten et al. in [3]. The method developed by Batten et al. was formed in such a way, that the velocity signal is specified in terms of the input parameters, mean velocity, Reynolds stress tensor, and dissipation rate. This report follows the method presented by Batten et al. The fluctuations are computed as:

$$u'_k(x_j, t) = a_{ki} \sqrt{\frac{2}{N}} \sum_{n=1}^N \left[p_i^n \cos(\hat{d}_j^n \hat{x}_j + \omega^n \hat{t}) + q_i^n \sin(\hat{d}_j^n \hat{x}_j + \omega^n \hat{t}) \right], \quad (2.12)$$

where a_{ki} is the Cholesky decomposition of the Reynold stress tensor, N is the total number of modes, p_i^n and q_i^n are amplitudes, \hat{d}_j^n is the modified wave numbers, ω^n is the frequency, \hat{x}_j and \hat{t} are spectral coordinates given by

$$\hat{x}_j = \frac{2\pi x_j}{L_t}, \quad \hat{t} = \frac{2\pi t}{\tau_t}, \quad (2.13)$$

where $L_t = \frac{k^{3/2}}{\varepsilon}$ is the local turbulent length scale and $\tau_t = \frac{k}{\varepsilon}$ is the local turbulent time scale, and k is the turbulent kinetic energy and ε is the dissipation rate. The translation of the spatial and temporal variables ensures correlation in space and time. The frequency ω^n is generated from a normal distribution with mean equal 1 and standard deviation equal 1, the wave numbers d_j^n are generated from a normal distribution with mean equal 0 and standard deviation equal 1/2. The amplitudes p_i^n and q_i^n are computed as

$$p_i^n = \epsilon_{ijk} \eta_j^n d_k^n, \quad q_i^n = \epsilon_{ijk} \xi_j^n d_k^n, \quad (2.14)$$

where the values of η_j^n and ξ_j^n are taken from a normal distribution with mean equal 1 and standard deviation equal 1, and ϵ_{ijk} is the Levi-Civita tensor, denoting a cross product. The modified wave numbers \hat{d}_j^n are computed as:

$$\hat{d}_j^n = d_j^n \frac{V_t}{c^n}, \quad (2.15)$$

where $V_t = \frac{L_t}{\tau_t}$ is the turbulent velocity scale and the coefficient c^n is given by

$$c^n = \sqrt{\frac{3}{2} R_{lm} \frac{d_l^m d_m^n}{d_k^n d_k^n}},$$

where R_{lm} is the Reynolds stress tensor. The Cholesky decomposition of the Reynolds stress tensor a_{ki} , that is introduced to ensure turbulence anisotropy, is defined according to Lund et al. in [14]:

$$a_{ki} = \begin{pmatrix} \sqrt{R_{11}} & 0 & 0 \\ R_{21}/a_{11} & \sqrt{R_{22} - a_{21}^2} & 0 \\ R_{31}/a_{11} & (R_{32} - a_{21}a_{31})/a_{22} & \sqrt{R_{33} - a_{31}^2 - a_{32}^2} \end{pmatrix}. \quad (2.16)$$

The Reynolds stress tensor used is computed locally in every cell, hence a_{ki} is computed locally in every cell. The variables are computed at every mode and the synthesised turbulent velocities are computed at every mode at every time step.

2.3 Synthetic fluctuations

Another method for generating synthetic turbulence, and the second method used in the present project, is presented by Davidson in [4], for isotropic fluctuations. The method bears some resemblance with the method presented by Batten et al., in the way that both methods are decomposing the turbulent signal in the Fourier domain. However, using the method proposed by Davidson, the turbulent velocity field is computed as:

$$u'_i(x_j) = 2 \sum_{n=1}^N \hat{u}^n \cos(\kappa_j^n x_j + \Psi^n) \sigma_i^n, \quad (2.17)$$

where \hat{u}^n is the amplitude, Ψ^n is the phase, σ_i^n is the direction of Fourier mode n , and κ_j^n is the wave number for mode n . This formula follows from first considering the decomposition of a signal into the Fourier domain, given in Equation (2.8). This can be rewritten as:

$$a_n \cos(\kappa_n x) + b_n \sin(\kappa_n x) = c_n \cos(\alpha_n) \cos(\kappa_n x) + c_n \cos(\alpha_n) \sin(\kappa_n x) = c_n \cos(\kappa_n x - \alpha_n), \quad (2.18)$$

$p(\varphi^n) = 1/(2\pi)$	$0 \leq \varphi^n \leq 2\pi$
$p(\Psi^n) = 1/(2\pi)$	$0 \leq \Psi^n \leq 2\pi$
$p(\theta^n) = 1/2 \sin(\theta)$	$0 \leq \theta^n \leq \pi$
$p(\alpha^n) = 1/(2\pi)$	$0 \leq \alpha^n \leq 2\pi$

Table 2.1: Probability distributions of the random angles φ^n , α^n , and θ^n , and the random phase Ψ^n .

$\kappa_{x_1}^n$	$\sin(\theta^n) \cos(\varphi^n)$
$\kappa_{x_2}^n$	$\sin(\theta^n) \sin(\varphi^n)$
$\kappa_{x_3}^n$	$\cos(\theta^n)$

Table 2.2: Components of the wave number vector κ_j^n .

where c_n and the phase angle α_n , are related to a_n and b_n as

$$c_n = (a_n + b_n)^{1/2}, \quad \alpha_n = \arctan\left(\frac{b_n}{a_n}\right). \quad (2.19)$$

This leads to the general form presented in Equation (2.17).

The procedure for computing the turbulence is as follows:

1. At every Fourier mode, generate random angles $\varphi^n, \alpha^n, \theta^n$ and phase Ψ^n with the probability distributions given in Table 2.1. Figure 2.1 and Figure 2.2 give a visual explanation of the different angles. ξ_i^n denotes the principal axes coordinate system.
2. Define the highest wave number for the used mesh κ_{\max} , the lowest wave number κ_l , and κ_e as:

$$\kappa_{\max} = \frac{2\pi}{2\Delta}, \quad \kappa_l = \frac{\kappa_e}{p}, \quad \kappa_e = \alpha \frac{9\pi}{55L_t}, \quad (2.20)$$

where Δ is the grid spacing, $\alpha = 1.453$, and L_t is the turbulent length scale. The factor p is chosen to $p = 2$ to make the largest scales larger than those corresponding to κ_e .

3. Divide the wave number space, obtained by $\kappa_{\max} - \kappa_l$, into N modes with equal size $\Delta\kappa$.
4. From Figure 2.1, compute the components of the wave number vector, as shown in Table 2.2.
5. Calculate components of unit vector σ_i^n . This is done by looking at Figure 2.2. Continuity requires that σ_i^n and κ_j^n are orthogonal for each wave number n . σ_3^n is chosen to be parallel with κ_3^n , and the direction of σ_i^n in the $\xi_1^n - \xi_2^n$ plane is chosen randomly through the randomisation of α^n . The resulting expressions for σ_i^n are presented in Table 2.3.

$\sigma_{x_1}^n$	$\cos(\varphi^n) \cos(\theta^n) \cos(\alpha^n) - \sin(\varphi^n) \sin(\alpha^n)$
$\sigma_{x_2}^n$	$\sin(\varphi^n) \cos(\theta^n) \cos(\alpha^n) + \cos(\varphi^n) \sin(\alpha^n)$
$\sigma_{x_3}^n$	$-\sin(\theta^n) \cos(\alpha^n)$

Table 2.3: Components of the unit vector σ_i^n .

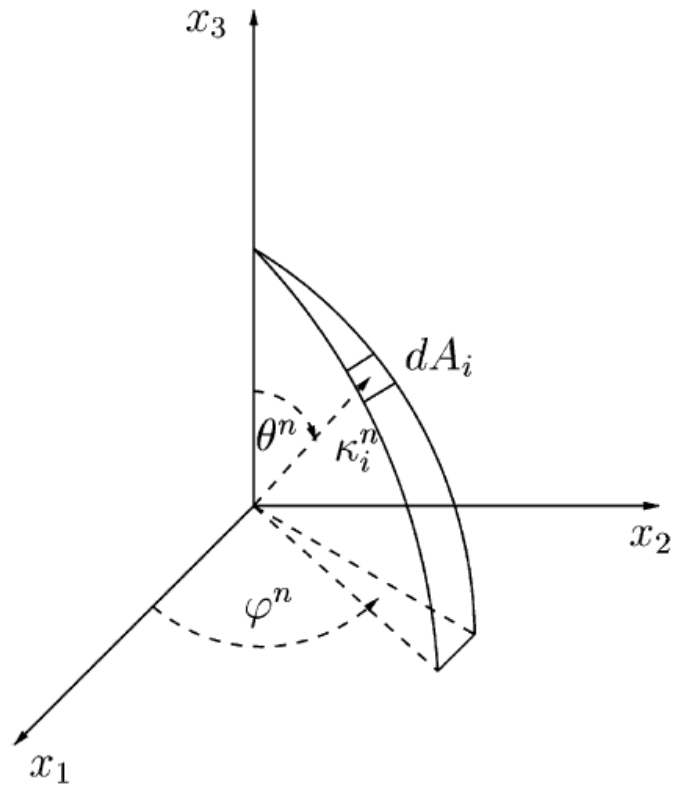


Figure 2.1: Visualisation of the randomised angles θ^n and φ^n , and their relation to the wave number vector κ_i^n . The figure is taken from [4].

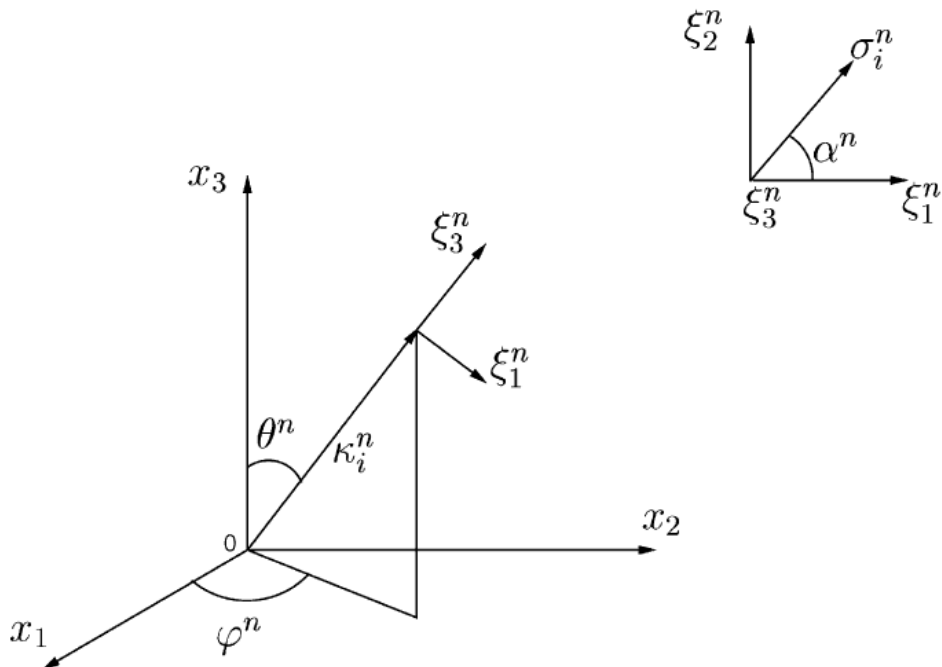


Figure 2.2: Visualisation of the randomised angles α^n , θ^n , and φ^n , and their relation to the wave number vector κ_i^n and the velocity unit vector σ_i^n . The figure is taken from [4].

6. Compute the amplitude as:

$$\hat{u}^n = (E(|\kappa_j^n|) \Delta\kappa)^{1/2}, \quad (2.21)$$

where the energy $E(\kappa)$ is chosen to describe a modified von Kármán spectrum by the equation:

$$E(\kappa) = \alpha \frac{u_{rms}^2}{\kappa_e} \frac{(\kappa/\kappa_e)^4}{[1 + (\kappa/\kappa_e)^2]^{17/6}} e^{-2(\kappa/\kappa_\eta)^2}, \quad (2.22)$$

with $\kappa = (\kappa_i \kappa_i)^{1/2}$, $\kappa_\eta = \varepsilon^{1/4} \nu^{-3/4}$, and u_{rms} is the root mean square of the velocity, computed as $u_{rms} = (\frac{2}{3}k)^{1/2}$. The spectrum can be seen in Figure 2.3.

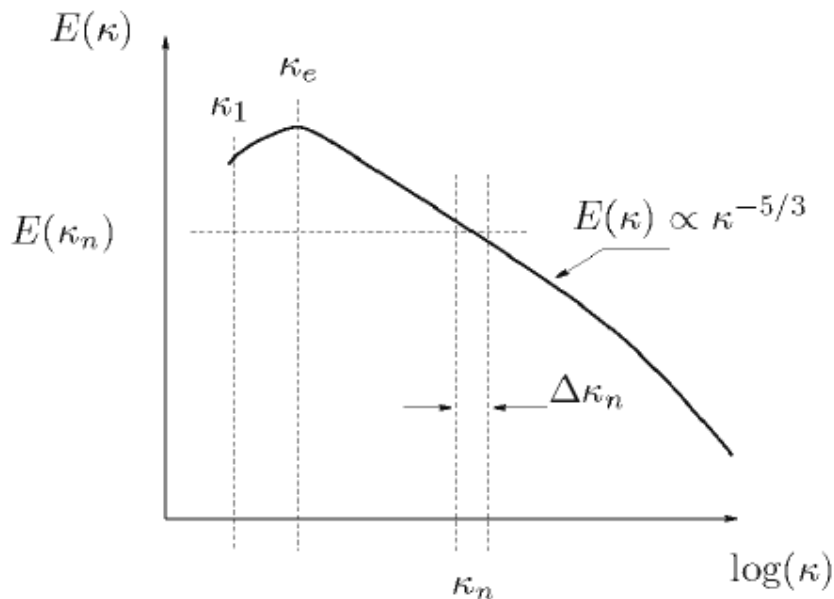


Figure 2.3: *Modified von Kármán spectrum. The figure is taken from [4].*

7. The velocity field is generated for a series of time steps, where the randomised quantities are determined for every time step. However, the generation of a velocity field using this method, will lead to independent time steps. To introduce correlation in time, an asymmetric time filter is used. The new velocities are computed for time step m as:

$$\begin{aligned} (\mathcal{U}')^m &= a(\mathcal{U}')^{m-1} + b(u'_1)^m, \\ (\mathcal{V}')^m &= a(\mathcal{V}')^{m-1} + b(u'_2)^m, \\ (\mathcal{W}')^m &= a(\mathcal{W}')^{m-1} + b(u'_3)^m, \end{aligned} \quad (2.23)$$

where $a = \exp(-\Delta t/\mathcal{T})$ and $b = (1 - a^2)^{1/2}$, where \mathcal{T} is the inlet time scale, and Δt is the time step. This will ensure a time correlation equal to $\exp(-\Delta t/\mathcal{T})$.

The method outlined above generates isotropic fluctuations. However, the same approach was made for the generation of non-isotropic fluctuations by Davidson and Billson in [11]. The procedure for generating the non-isotropic fluctuations is very similar to the method for generating isotropic fluctuations, with some small additions as follows:

1. The Reynolds stress tensor for the flow is supplied, as well as a turbulent length scale. In the present thesis, these quantities were obtained from a RANS simulation.

2. The eigenvalues of the Reynolds stress tensor, that is the normal Reynolds stresses in the principal coordinate system, are computed, as are the principal coordinate directions.
3. Isotropic fluctuations in the principal coordinate system are generated, as described above, and rescaled.
4. The non-isotropic fluctuations length scales σ_i and the wavenumbers κ_j are modified so as to make the fluctuations anisotropic and to ensure that the fluctuations satisfy continuity.
5. The non-isotropic fluctuations are transformed back to the original coordinate system, where the Reynolds stresses of the synthetic fluctuations are equal to the Reynolds stresses supplied in step 1..

2.4 Synthetic eddy method

The third method for generating synthetic turbulence discussed here, is the synthetic eddy method, presented by Jarrin et al. in [5]. The procedure for generating fluctuations using this method is as follows:

1. Define and calculate input data, consisting of the mean velocity U_i , the Reynolds stress tensor R_{ij} , and the length scales σ_i .
2. Define a box on which eddies are generated. This box is defined as:

$$[-\sigma_{x_1}, \sigma_{x_1}; -\sigma_{x_2}, \sigma_{x_2}; -\sigma_{x_3}, \sigma_{x_3}], \quad (2.24)$$

where σ_i is the length scale in the different directions. A visualisation of the generated box of eddies, can be seen in Figure 2.4. The solid lines make the LES domain, and the cross hatched lines make the box in which the eddies are generated. The grey area is the inlet to the LES domain and the black dots represent the randomised eddies.

3. Generate a random position \mathbf{x}^k and intensity ε_i^k for every eddy, inside the previously defined box.
4. The eddies are convected through the box with a reference velocity scale U_0 . Using Taylor's frozen turbulence hypothesis, saying that the advection of turbulence past a fixed point is due to the mean flow, the new position for the eddy is given as

$$x_i(t + dt) = x_i(t) + U_0 dt. \quad (2.25)$$

When $x_i > \sigma_i$, that is when the eddy reaches the end of the box, regenerate the eddy at $x_i = -\sigma_i$, at a random position in the two other directions, with a new random intensity.

5. The fluctuations are computed as:

$$u'_i(x) = \frac{1}{\sqrt{N}} \sum_{k=1}^N a_{ij} \varepsilon_j^k f_{\sigma(x)}(\mathbf{x} - \mathbf{x}^k), \quad (2.26)$$

where N is the number of eddies, a_{ij} is the Cholesky decomposition of the Reynolds stress tensor, computed locally in every cell as in Equation (2.16). The intensities ε_i^k are randomly generated as $\varepsilon_i^k \in \{-1, 1\}$, \mathbf{x} is the position in the mesh and \mathbf{x}^k is the position of eddy i , and $f_{\sigma(x)}$ is a shape function taken as:

$$f_{\sigma(x)}(\mathbf{x} - \mathbf{x}^k) = \frac{\sqrt{V_B}}{\sigma_{x_1} \sigma_{x_2} \sigma_{x_3}} f\left(\frac{x_1 - x_1^k}{\sigma_{x_1}}\right) f\left(\frac{x_2 - x_2^k}{\sigma_{x_2}}\right) f\left(\frac{x_3 - x_3^k}{\sigma_{x_3}}\right),$$

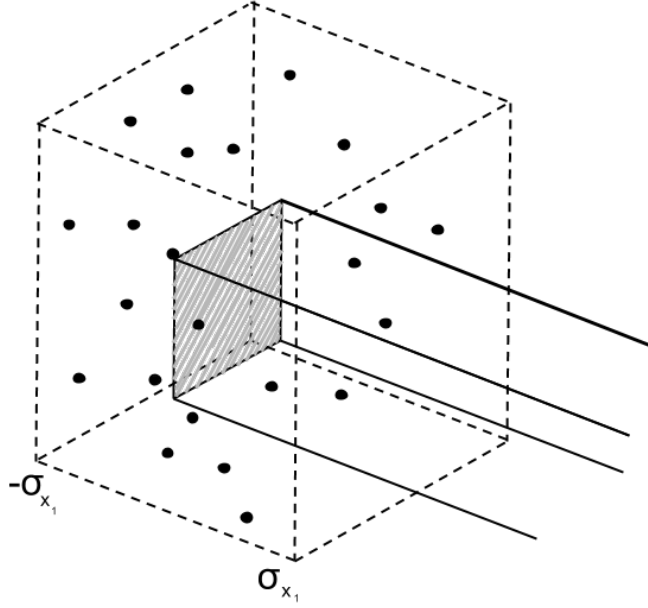


Figure 2.4: *Box of eddies, generated when using SEM and DFSEM.*

where V_B is the volume of the box and $f(x)$ in the present project is taken as a hat function [15]:

$$f(x) = \begin{cases} \sqrt{\frac{3}{2}}(1 - |x|), & \text{if } x < 1, \\ 0, & \text{else} \end{cases}.$$

2.5 Divergence free synthetic eddy method

The last method for generating synthetic turbulence discussed here, is the divergence free synthetic eddy method, presented by Poletto et al. in [6]. The method was originally developed as the SEM, however, the SEM did not fulfil the requirement of continuity, hence the method was further developed into a divergence free synthetic eddy method. The two methods are very similar and differ mainly in the expression for the fluctuations. The procedure for generating the turbulent velocities is as follows:

1. Define and calculate input data as in the SEM.
2. Define the same box as in the SEM.
3. Generate a random position x^k and intensity α^k for every eddy, inside the previously defined box.
4. The eddies are convected through the box in the same way as in the SEM.
5. The fluctuations are now computed as:

$$\mathbf{u}'(\mathbf{x}) = \sqrt{\frac{1}{N}} \sum_{k=1}^N \frac{g_\sigma(|\mathbf{r}^k|)}{|\mathbf{r}^k|^3} \mathbf{r}^k \times \boldsymbol{\alpha}^k, \quad (2.27)$$

where N is the number of eddies, $r_i^k = \frac{x_i - x_i^k}{\sigma_i^k}$, α_i^k are the intensities of the eddies, and $q_\sigma(|\mathbf{r}^k|)$ is a shape function, and the summation is taken over the k number of eddies.

The fluctuations generated by Equation (2.27) do not reproduce proper turbulence anisotropy. In order to introduce anisotropy, Equation (2.27) is rewritten with a shape function that both satisfies the divergence free requirement as well as introducing turbulent anisotropy. This shape function is taken as:

$$q_i = \begin{cases} \sigma_i [1 - (d^k)^2], & \text{if } d^k < 1, \\ 0, & \text{else} \end{cases}, \quad (2.28)$$

where $d^k = \sqrt{(r_j^k)^2}$. The fluctuations are now computed as:

$$u'_\beta(x) = \sqrt{\frac{1}{N}} \sum_{k=1}^N \sigma_\beta^k [1 - (d^k)^2] \epsilon_{\beta j l} r_j^k \alpha_l^k, \quad (2.29)$$

where r_j^k is computed as $r_i^k = \frac{x_i - x_i^k}{\sigma_i^k}$ and $\epsilon_{\beta j l}$ is the Levi-Civita tensor. To prevent the shear stresses from becoming zero, the fluctuations in the global coordinate system are computed using a rotation transformation of the eddies generated in the local principal axes coordinate system as:

$$u_i^G(x) = C_1 R_{im}^{P \rightarrow G} u_m^P, \quad (2.30)$$

where $R_{im}^{P \rightarrow G}$ is the rotational matrix from the principal coordinate system to the global coordinate system, u_m^P is the fluctuations in the principal axes coordinate system, and C_1 is a normalisation coefficient given by

$$C_1 = \frac{\sqrt{10V_0} \sum_{i=1}^3 \frac{\sigma_i}{3}}{\sqrt{N} \prod_{i=1}^3 \sigma_i} \min\{\sigma_i\}, \quad (2.31)$$

where V_0 is the eddy box volume. In order to ensure that the method will return the desired Reynolds stress statistics, over a range of anisotropy levels, the length scale ratios denoted by $\Gamma = \frac{\sigma_x}{\sigma_y} = \frac{\sigma_x}{\sigma_z}$, need to be chosen, together with the proper intensities, given by

$$\langle (\alpha_\beta^k)^2 \rangle = \frac{\lambda_j / \sigma_j^2 - 2\lambda_\beta / \sigma_\beta^2}{2C_2}, \quad (2.32)$$

where λ_j are the normal stresses in the local principal reference system, and C_2 is a constant coefficient. The different anisotropy levels of the flow can be visualised using a Lumley triangle. The idea of this triangle is to map the Reynolds stress anisotropy tensor, b_{ij} , through using the eigenvalues of b_{ij} . The stress anisotropy tensor is given by

$$b_{ij} = \frac{\langle u'_i u'_j \rangle}{\langle u'_n u'_n \rangle} - \frac{\delta_{ij}}{3}. \quad (2.33)$$

Since the right hand side of Equation (2.32) needs to be positive, there are, for every eigenvalue, some permitted values on the length scales. Hence, by choosing a series of values on the length scale ratio Γ , it is possible to map different parts of the Lumley triangle. A series of values on Γ are chosen, following the values in [6], and are presented in Table 2.4, together with corresponding values on coefficient C_2 . The Lumley triangle with the mapped parts is presented in Figure 2.5, where the axes are described as:

$$\eta^2 = \frac{b_{ij} b_{ji}}{6}, \quad \xi^3 = \frac{b_{ij} b_{jn} b_{ni}}{6}. \quad (2.34)$$

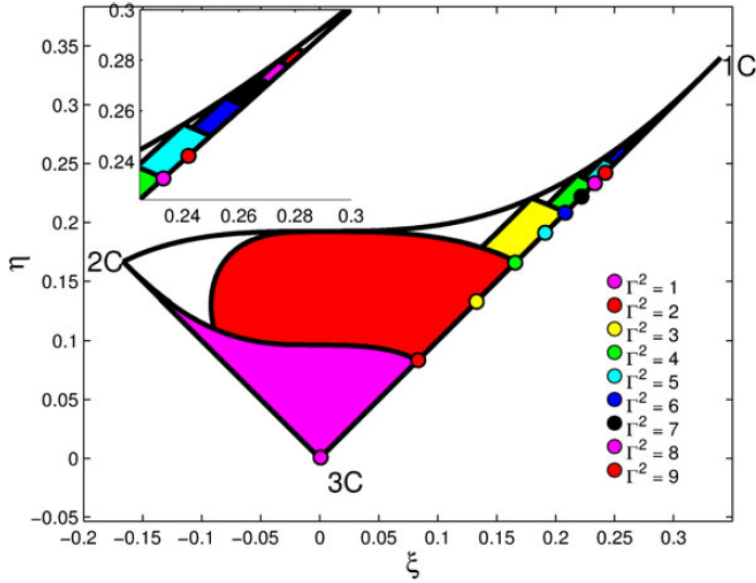


Figure 2.5: *Lumley triangle used for visualising different anisotropic regions of the Reynolds stress tensor. The figure is taken from [6].*

Γ	1	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{4}$	$\sqrt{5}$	$\sqrt{6}$	$\sqrt{7}$	$\sqrt{8}$
C_2	2.0	1.875	1.737	1.75	0.91	0.825	0.806	1.5

Table 2.4: Length scale ratios Γ with corresponding values on coefficient C_2 .

The actual length scale magnitude is taken as:

$$\sigma_{\text{avg}} = \min \left(\frac{k^{3/2}}{\varepsilon}, \kappa\delta, \max(\Delta x_1, \Delta x_2, \Delta x_3) \right), \quad (2.35)$$

where $\frac{k^{3/2}}{\varepsilon}$ is the local length scale taken from RANS data, δ is the channel half-height, and $\kappa = 0.41$ is the Von Kármán constant.

2.6 Calculating the Reynolds stress tensor

The methods described in previous chapters all demand proper input in order to be able to generate accurate synthetic turbulence. One of the quantities that must be provided, and that needs to be computed, is the Reynolds stress tensor of the flow. In the present project, the Reynolds stresses were obtained from the results of computing a 1D RANS simulation of the flow, using the explicit algebraic Reynolds stress model (EARSM), developed by Wallin and Johansson in [16]. A short summary of the model used in this project is presented here, for a more thorough description the reader is referred to [16]. The Reynolds stresses in a two-dimensional flow may be written on the form

$$\langle u_i u_j \rangle = k \left(\frac{2}{3} \delta_{ij} - 2C_\mu^{\text{eff}} S_{ij} + a_{ij} \right), \quad (2.36)$$

where k is the kinetic energy, S_{ij} is the strain rate tensor, δ_{ij} is Kronecker's delta, and

$$C_\mu^{\text{eff}} = -\frac{1}{2} f_1 \beta_1. \quad (2.37)$$

The anisotropy tensor b_{ij} is, for the two-dimensional case, given by

$$b_{ij} = \beta_1 S_{ij} + \beta_2 \left(S_{ik} S_{kj} - \frac{1}{3} S_{kl} S_{kl} \delta_{ij} \right) + \beta_4 (S_{ik} \Omega_{kj} - \Omega_{ik} S_{kj}), \quad (2.38)$$

where Ω_{ij} is the vorticity tensor, defined as:

$$\Omega_{ij} = \frac{1}{2} \tau \left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right), \quad (2.39)$$

where the time scale $\tau = \frac{k}{\varepsilon}$. The β - coefficients are given by

$$\beta_1 = -\frac{6}{5} \frac{N}{N^2 - 2II_\Omega}, \quad (2.40)$$

$$\beta_2 = 0, \quad (2.41)$$

$$\beta_4 = \frac{\beta_1}{N}, \quad (2.42)$$

where II_Ω is given by

$$II_\Omega = \Omega_{kl} \Omega_{lk}. \quad (2.43)$$

N is obtained from the solution of a cubic equation

$$N = \frac{A'_3}{3} + \left(P_1 + \sqrt{P_2} \right)^{1/3} + \text{sign} \left(P_1 - \sqrt{P_2} \right) \left| P_1 - \sqrt{P_2} \right|^{1/3}, \quad \text{for } P_2 \geq 0, \quad (2.44)$$

and

$$N = \frac{A'_3}{3} + 2 \left(P_1^2 + P_2 \right)^{1/6} \cos \left[\frac{1}{3} \cos^{-1} \left(\frac{P_1}{\sqrt{P_1^2 - P_2}} \right) \right], \quad \text{for } P_2 < 0, \quad (2.45)$$

where

$$P_1 = A'_3 \left[\frac{A_3'^2}{27} + \frac{9}{20} II_S - \frac{2}{3} II_\Omega \right], \quad (2.46)$$

$$P_2 = P_1^2 - \left[\frac{A_3'^2}{9} + \frac{9}{10} II_S + \frac{2}{3} II_\Omega \right]^3, \quad (2.47)$$

with

$$II_S = S_{kl} S_{lk}, \quad (2.48)$$

and A'_3 is chosen to

$$A'_3 = \frac{9}{4} (1.8 - 1). \quad (2.49)$$

For low Reynolds numbers, the model needs to be slightly modified. The turbulence time scale is chosen to

$$\tau = \max \left(\frac{k}{\varepsilon}, C_\tau \sqrt{\frac{\nu}{\varepsilon}} \right), \quad (2.50)$$

where k is the kinetic energy, ε is the dissipation rate, ν is the viscosity, and C_τ is a constant. The non-dimensional shear stress is expressed as

$$\sigma = \frac{1}{2} \frac{k}{\varepsilon} \frac{dU}{dy}. \quad (2.51)$$

A wall damping function is chosen to

$$f_1 = 1 - \exp\left(-\frac{y^+}{A^+}\right), \quad (2.52)$$

where A^+ is set to $A^+ = 26$. The β -coefficients are now computed as

$$\beta_{1,low} = f_1\beta_1, \quad (2.53)$$

$$\beta_{4,low} = f_1^2\beta_4 - (1 - f_1^2)\frac{B_2}{4\sigma^2}, \quad (2.54)$$

where B_2 is chosen to $B_2 = 1.8$, and

$$\beta_{2,low} = (1 - f_1^2)\frac{3B_2 - 4}{2\sigma^2}. \quad (2.55)$$

2.7 General method for generating synthetic turbulence

The general way of generating synthetic turbulence, regardless of which method that is used, can be described by the steps presented below, following the same methodology as in [17].

1. A 1D precursor RANS simulation is performed.
2. The Reynolds stress tensor is computed using the EARSM by Wallin and Johansson.
3. The Reynolds stress tensor is used as input for generating the anisotropic synthetic fluctuations.
4. The Reynolds stresses are taken in one point of the domain, where the magnitude of the turbulent shear stress is largest.
5. The synthetic fluctuations are scaled with the local value of the Reynolds stress

$$\left(\frac{|\langle u'_1 u'_2 \rangle|}{|\langle u'_1 u'_2 \rangle|_{\max}}\right)_{\text{RANS}}^{1/2}, \quad (2.56)$$

which is taken from the RANS simulation.

2.8 Turbulent statistics

In order to verify that the generated turbulence is satisfying and that the behaviour of the turbulence is physically correct, some statistics of the turbulence will be examined. The generated turbulence must follow the same physical behaviour as naturally existing flow, both in time and in space. Hence, for the generated flow, there is a need to examine the two-point correlation in space and the autocorrelation in time.

2.8.1 Two-point correlation

In order to examine if the generated velocity field behaves physically correct in space, two-point correlation is used in order to find out if there is a correlation between two points in space. The two-point correlation can be described as in [12]. First, pick two points along the x_1 -axis x_1^A and x_1^C . Sample the fluctuating velocity in one direction, for example the x_1 -direction. The correlation of u'_1 at the two chosen points is given by:

$$B_{11}(x_1^A, \hat{x}_1) = \overline{u'_1(x_1^A)u'_1(x_1^A + \hat{x}_1)}, \quad (2.57)$$

where $\hat{x}_1 = x_1^C - x_1^A$ is the distance between the two points. From Equation (2.57), it follows that when the distance between the two points decreases, the two-point correlation increases. When the distance between the points instead increases, the two-point correlation goes to zero. The two-point correlation is usually normalised so that it varies between -1 and 1 . The normalised two-point correlation is given by:

$$B_{11}^{norm}(x_1^A, \hat{x}_1) = \frac{1}{u_{1,rms}(x_1^A)u_{1,rms}(x_1^A + \hat{x}_1)} \overline{u_1'(x_1^A)u_1'(x_1^A + \hat{x}_1)}, \quad (2.58)$$

where

$$u_{1,rms} = \left(\overline{u_1'^2}\right)^{1/2} \quad (2.59)$$

is the root mean square of u_1' . The generated fluctuations are expected to have a two-point correlation that goes towards zero as the distance between the two measuring points increases, and equal zero where the distance is of the order of the turbulent length scale. An integral length scale can be computed as:

$$L_{int}(x_1) = \int_0^\infty \frac{B_{11}(x_1, \hat{x}_1)}{u_{1,rms}^A u_{1,rms}^C} d\hat{x}_1. \quad (2.60)$$

2.8.2 Autocorrelation

Autocorrelation can be thought of in the same way as two-point correlation, but instead of choosing two points in space, now choose two points in time t^A and t^C , separated by $\hat{t} = t^C - t^A$. The autocorrelation of u_1' for the two time points is given by

$$B_{11}(t^A, \hat{t}) = \overline{u_1'(t^A)u_1'(t^A + \hat{t})}. \quad (2.61)$$

In the same way as with the two-point correlation, the autocorrelation is usually normalised and the equation becomes:

$$B_{11}^{norm}(\hat{t}) = \frac{1}{u_{1,rms}^2} \overline{u_1'(t^A)u_1'(t^A + \hat{t})}. \quad (2.62)$$

Similar to the two-point correlation, the autocorrelation is expected to go to zero when the time difference between the measuring points increases, and equal zero when the difference is of the order of the turbulent time scale of the flow. An integral time scale can be computed as:

$$T_{int} = \int_0^\infty B_{11}^{norm}(\hat{t}) d\hat{t}. \quad (2.63)$$

2.9 HYDRA

Here follows a short presentation of the CFD solver HYDRA. The information presented in this section can be found in [18].

HYDRA is a CFD code which is mainly used for high speed flows in aero engine and industrial applications with air as the working fluid. Hence it is written as a coupled solver and the code is designed for solving flows of compressible ideal gases. All fluids are treated as compressible and to avoid compressibility, it is necessary to use low Mach-numbers as fluids at low Mach-numbers are deemed compressible. In HYDRA, this is made possible by using the feature Low Mach Pre-Conditioning. HYDRA is aimed for solving fully submerged flows, hence there is no capability for free surface solutions.

Fluxes are divided into two parts, one inviscid and one viscous part, where the viscous part is nonlinear. This decomposition of the flux originates from the coupled eigen system of mass-momentum and energy equation. To increase the convergence rate, a multigrid is implemented, consisting of four grid levels. The first is the original mesh imported into HYDRA and the three other grid levels are coarser grid levels created in HYDRA by removing edges from the previous grid. The idea is that at each iteration all equations are solved on every grid level. Starting at the coarsest grid and moving to finer and finer grids, where the solution is obtained at the finest grid. To decrease the convergence time, a pre-conditioner is used.

3 Numerical method

This section describes the numerical setup for the testing of the methods. First, the setup for computing the autocorrelation and two-point correlation is described followed by the setup used for a channel flow simulation.

3.1 Setup for computing correlations

All three methods were examined with regard to autocorrelation in time and two-point correlation in space. This was done through using the methods to create turbulence on a small mesh, consisting of $1 \times 1 \times 62$ cells, with the directions streamwise, wall normal, and spanwise respectively. The input for the methods was obtained from a 1D precursor RANS simulation, as described in Section 2.6. The number of time steps used was set to 10 000. The length scale and time scale were obtained through the turbulent kinetic energy k and the dissipation rate ε taken from the RANS simulation.

It was found that in order to provide smooth results, the method developed by Batten needed to use more Fourier modes than was needed when using the method developed by Davidson and was much slower. The number of modes when using the two methods was set to 1000 for the method by Batten and 150 for the method by Davidson. Regarding the DFSEM, it was pointed out in [6] that a large value of eddies produces a more accurate result, at the cost of being more computationally heavy. When testing the methods, a number of 2000 eddies were used for the DFSEM and SEM.

3.2 Channel flow

A second order monotone upwind scheme for scalar conservation laws (MUSCL) was used for discretisation in space, together with an explicit time scheme. A kinetic energy preserving conservative scheme (KEP), presented in [19], was used in order to ensure that the global discrete kinetic energy evolves in a manner that corresponds to the true equation for kinetic energy. The global conservation property is obtained by proper construction of the interface flux between each pair of neighbouring cells. More explicitly, the flux between each pair of neighbouring cells is averaged, where the average of any quantity between its values at o and p is defined by

$$\bar{q} = \frac{1}{2}(q_p + q_o). \quad (3.1)$$

The kinetic energy conservation law is given by:

$$\frac{\partial k}{\partial t} = p \frac{\partial u_j}{\partial x_j} - \tau_{ij} \frac{\partial u_i}{\partial x_j}, \quad (3.2)$$

where k is the kinetic energy, p is the pressure, and τ is the viscous stress. The discretisation scheme used is presented by:

$$\sum_o vol_o \frac{dk^o}{dt} = \sum_b S_i^o \left\{ u_i^o \left(p^o + \rho^o \frac{u_i^{o2}}{2} \right) - u_i^o \sigma_{ij}^o \right\} + \sum_o \left(p^o \sum_p \bar{u}_i S_i^{op} - \sigma_{ij}^o \sum_p \bar{u}_j S_i^{op} \right), \quad (3.3)$$

where index o represents an interior cell, index p represents the surrounding cells, and index b represents the boundaries. vol_o is the volume of the cell, S_i^o is the outer face area, S_i^{op} is the face area between the two cells, ρ is the density, and \bar{u}_i is the mean value of the velocity between the points o

and p . The first sum on the right hand side represents the flux through boundaries and the second sum is the finite volume discretisation of

$$\int_D \left(p \frac{\partial u_i}{\partial x_i} - \tau_{ij} \frac{\partial u_j}{\partial x_i} \right) dV. \quad (3.4)$$

The channel consisted of $800 \times 20 \times 40$ cells in streamwise, wall normal, and spanwise direction respectively. There are periodic boundary conditions in the spanwise direction and viscous walls at the top and bottom. The Mach number was set to 0.15, which is low enough for using the assumption of incompressible flow.

4 Results

The four methods were first examined through the use of autocorrelation and two-point correlation. Next, the synthetic fluctuations method, the SEM, and the DFSEM were implemented into HYDRA, where the SEM and the DFSEM were used for generating inflow in a channel flow simulation.

4.1 Correlations

The four different methods were tested regarding the autocorrelation and the two-point correlation. The normalised autocorrelation of the generated turbulence using the different methods are presented in Figure 4.1, and the normalised two-point correlations are presented in Figure 4.2. The correlations are computed in the spanwise direction of the flow. The figures show the correlations for the streamwise components of the velocity, for the correlation of the other components see Appendix A.

All the methods seem to yield turbulence with physically expected correlations. Correlations that are converging towards zero when the distance, in time and in space, increases. Looking at Figure 4.1b, showing the autocorrelation using the method by Davidson, this autocorrelation behaves precisely as an exponential function, just as expected when using the time filter. Further, by inspecting Figure 4.1 for the autocorrelations, it is seen that the time scale is of order 0.15 since this is the value of the time difference for when the autocorrelation becomes zero. In Figure 4.2, it is seen that the two-point correlation goes to zero when the distance between the measure points is close to 0.05, which is the order of the length scale.

4.2 Channel flow

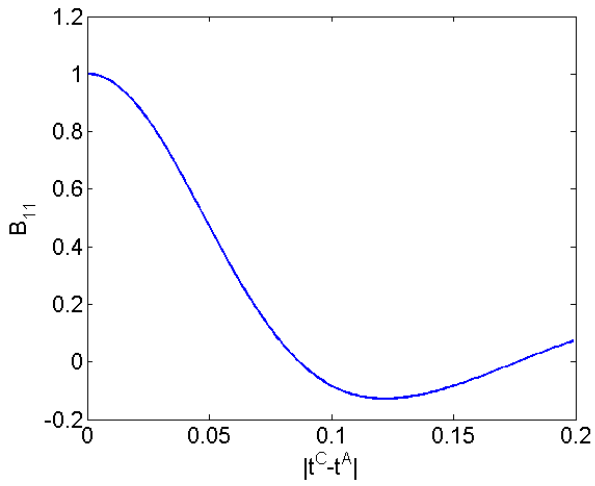
It was chosen to implement the synthetic fluctuations, SEM, and DFSEM, into HYDRA CFD Code. These methods were chosen over the method by Batten since they, according to the documentation of the methods as well as the results from testing the correlations, seemed more promising. Further, the SEM and the DFSEM were used for generating turbulent inlet conditions in a channel flow simulation.

Figure 4.3 shows the result obtained from the channel flow using the SEM and Figure 4.4 shows the result obtained when using the DFSEM. In Figure 4.3a and Figure 4.4a the relative velocity in the streamwise direction is shown for the two methods. It is clear from the figures that there are indeed some fluctuations throughout the channel in both cases. Figure 4.3b and Figure 4.4b shows the density in the channel, which seems to not vary much all through the channel. It is not fully constant but the variations are small enough to not ruin the assumption of an incompressible flow.

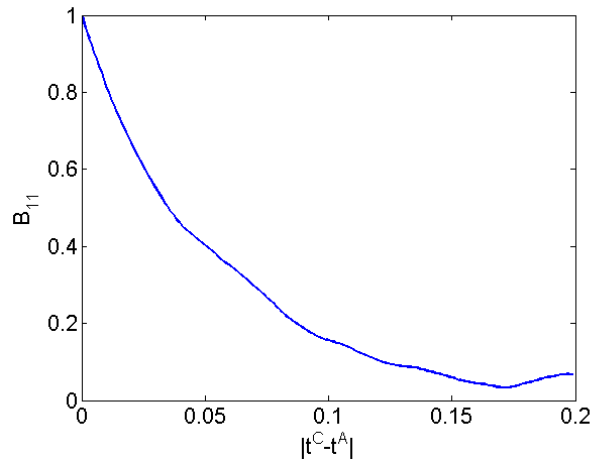
The autocorrelation for the velocity in the wall normal direction is for the two methods shown in Figure 4.5. The convergence towards zero when the time between two measuring points increases, makes it possible to assume that the correlation of the generated fluctuations behaves as expected and that the methods have been implemented into HYDRA correctly.

As a qualitative presentation of the generated fluctuations, the turbulence is shown for a small channel flow in Figure 4.3. The setup is a $100 \times 20 \times 40$ channel with periodic boundary conditions in the spanwise direction and viscous walls at the top and bottom. Figure 4.6a shows the turbulence when using SEM for the generation of synthetic fluctuations at the inflow. As the turbulence is convected downstream, the finer scales tend to dissipate. However, since the turbulence is properly correlated the turbulence persists all the way to the outflow. As a comparison for what happens when using uncorrelated turbulence at the inflow, Figure 4.6b shows the resulting flow when describing

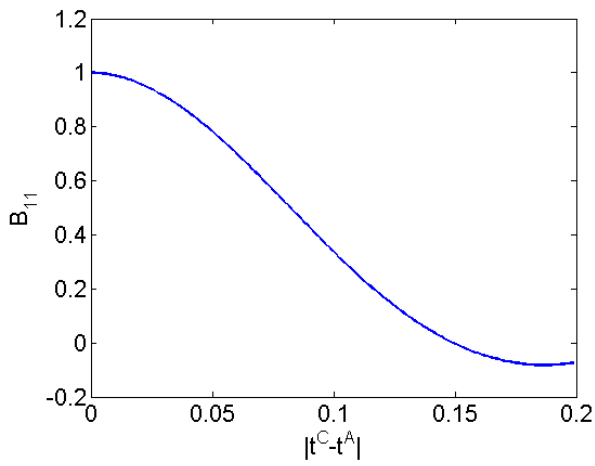
the inlet fluctuations as simple white noise. It is seen from the figure that, when the fluctuations are uncorrelated, the turbulence does not persist all through the flow, but instead disappears almost instantly.



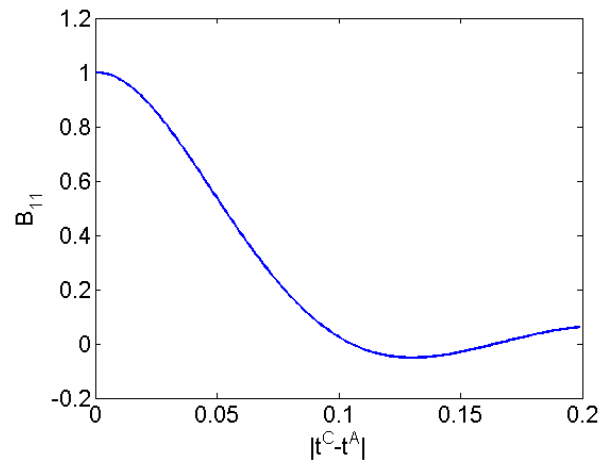
(a) Autocorrelation using the method by Batten.



(b) Autocorrelation using the method by Davidson.

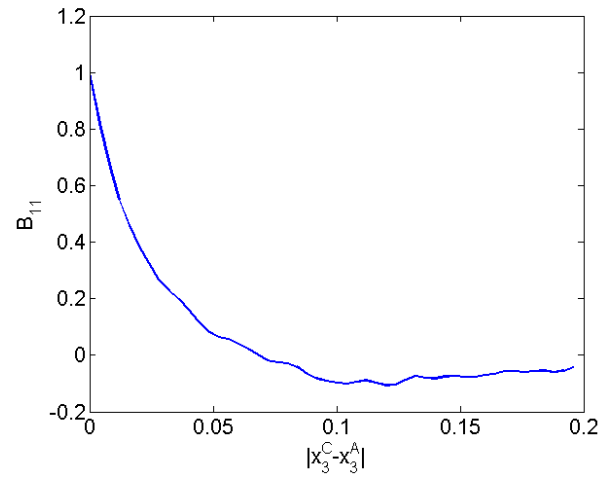
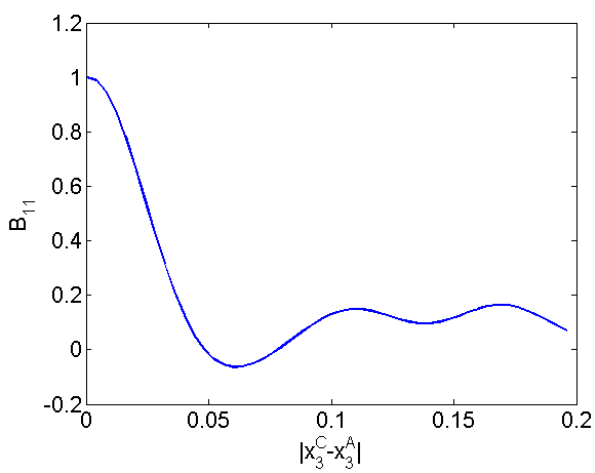


(c) Autocorrelation using SEM.

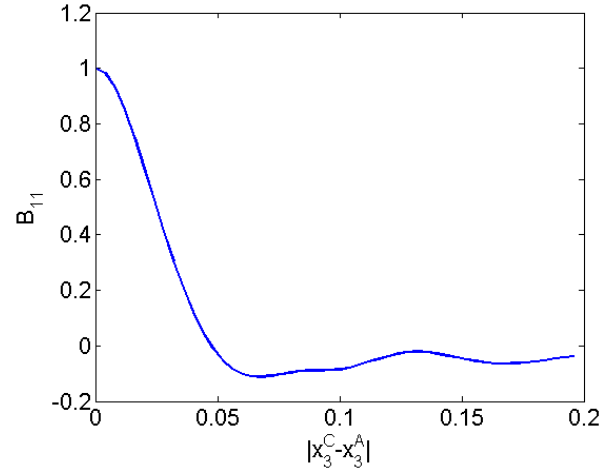
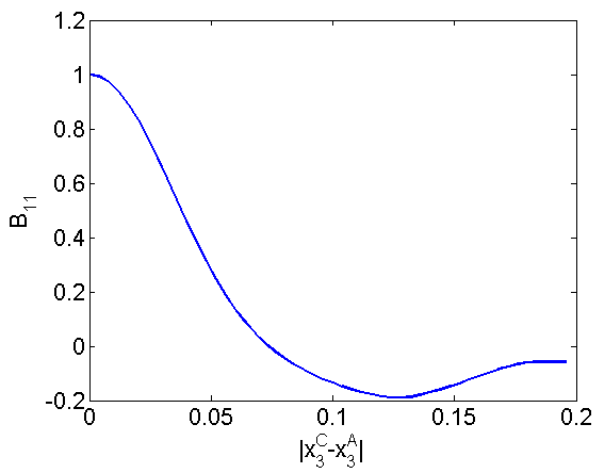


(d) Autocorrelation using DFSEM.

Figure 4.1: Autocorrelation in the x_3 -direction.



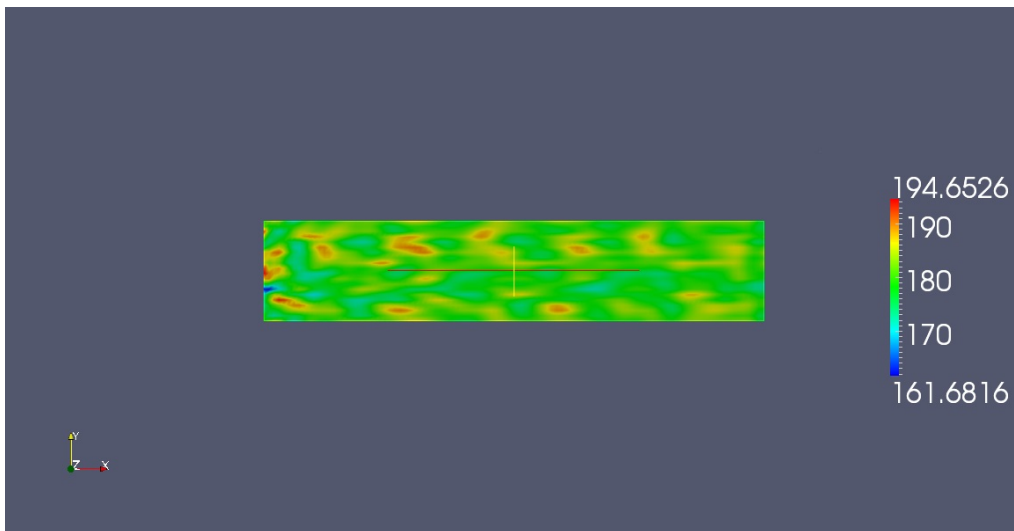
(a) Two-point correlation using the method by Batten. (b) Two-point correlation using the method by Davidson.



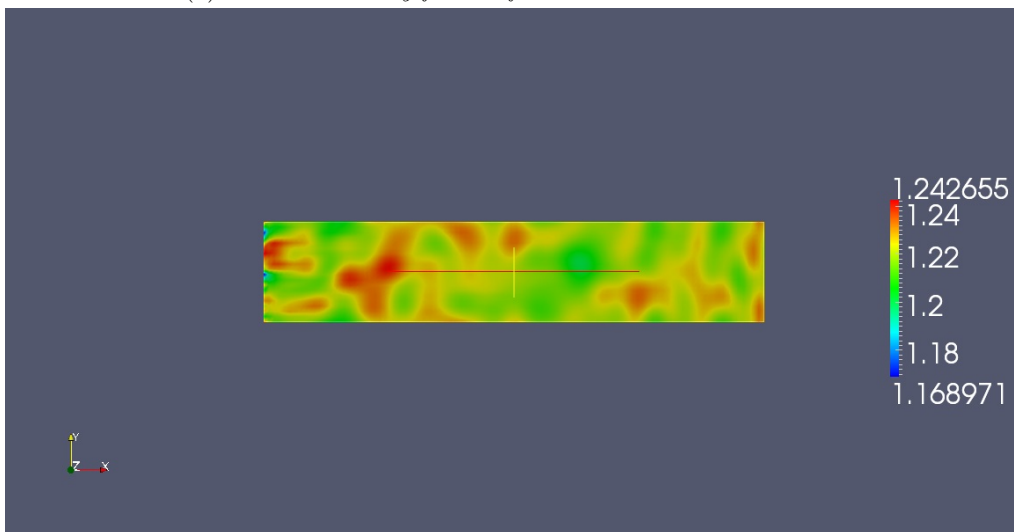
(c) Two-point correlation using SEM.

(d) Two-point correlation using DFSEM.

Figure 4.2: Two-point correlation in the x_3 -direction.

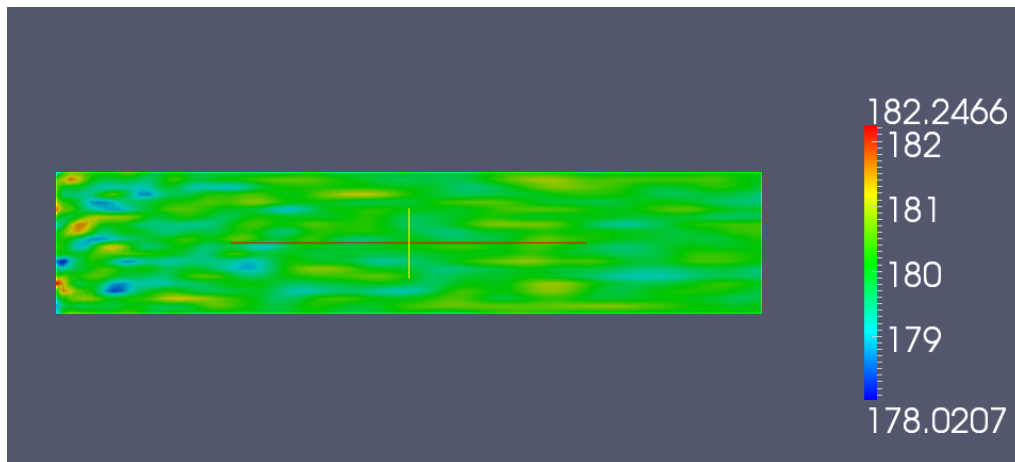


(a) *Relative velocity for the flow in streamwise direction.*

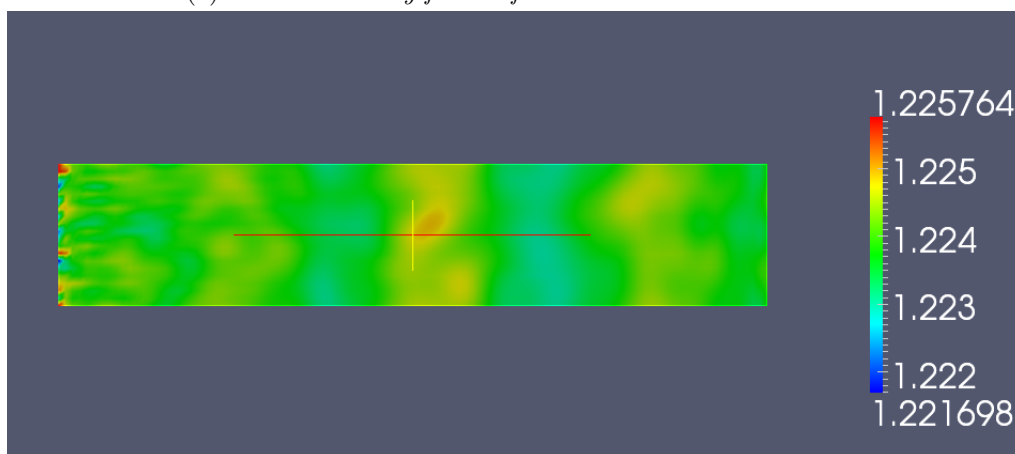


(b) *Visualisation of the density in the channel flow.*

Figure 4.3: *The relative velocity and density for the simulated channel flow using SEM at the inlet.*

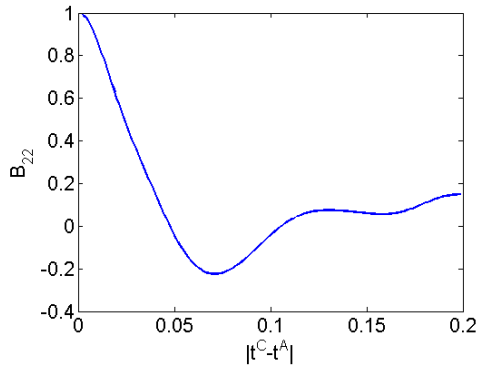


(a) *Relative velocity for the flow in streamwise direction.*

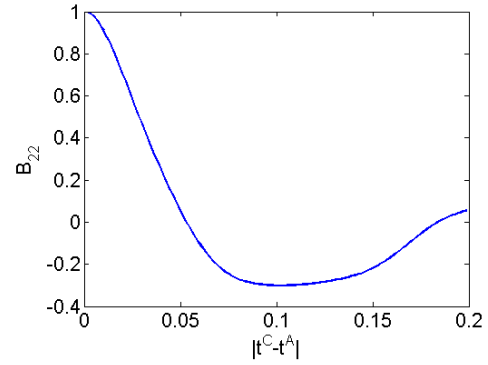


(b) *Visualisation of the density in the channel flow.*

Figure 4.4: *The relative velocity and density for the simulated channel flow using DFSEM at the inlet.*

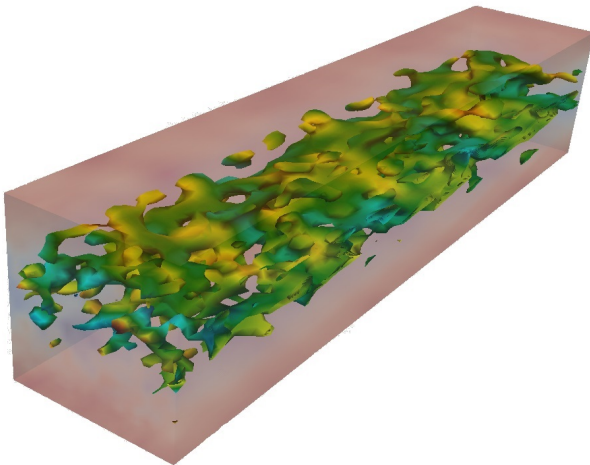


(a) Autocorrelation for the velocity using the SEM at the inlet.

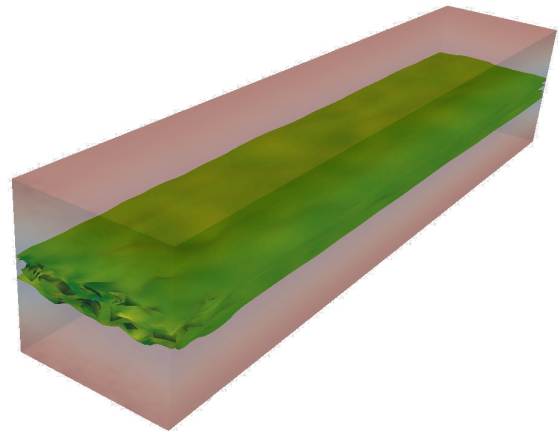


(b) Autocorrelation for the velocity using the DFSEM at the inlet.

Figure 4.5: The autocorrelation for the channel flow when using SEM and DFSEM at the inlet respectively.



(a) Turbulence using SEM for inlet conditions.



(b) Turbulence using white noise for inlet conditions.

Figure 4.6: Visualised turbulence when using different methods as inlet condition.

5 Conclusions

This thesis has described the implementation and analysis of four different methods for generating synthetic inlet conditions for LES. The four methods implemented are a spectral method by Batten, synthetic fluctuations by Davidson, SEM by Jarrin and DFSEM by Poletto. The last three have all been implemented into HYDRA CFD Code.

This work was motivated by the increasing interest for using LES in industrial applications, and the need for cost effective methods for generating inflow data. The four methods implemented, all require statistical quantities as input, available from a RANS simulation. The Reynolds stress tensor were obtained using an EARSM by Wallin and Johansson. The generated fluctuations were first investigated in terms of autocorrelation and two-point correlation, in order to examine the accuracy of the physical appearance of the turbulence generated. The fluctuations were generated along one line in spanwise direction of the flow. It was shown that all of the methods provided fluctuations with both autocorrelation and two-point correlation with the physical appearance expected, meaning that the correlations were converging towards zero when the distance between the measuring points were close to the value of the time scale and length scale respectively. The methods by Davidson, Jarrin, and Poletto were then implemented into HYDRA CFD Code. The implementation of the SEM and the DFSEM were further investigated by using the two methods to generate inlet conditions for an LES simulation of a channel flow. The results show that there is turbulence all through the channel, which indicates that the implementation of the SEM and the DFSEM into HYDRA was successful.

As future work, the implementation of the method by Davidson should also be investigated further in form of channel flow simulation. The methods should all be tested and compared to DNS data to get a more quantitative indication of how well the methods perform. It could also be beneficial to test the methods on more complex geometries to get a more thorough understanding of how well the implemented methods perform. Also, while this report focused on using the methods as inlet conditions for LES, it should be possible to use the methods for boundary conditions in the interface between the RANS region and the LES region in a hybrid LES-RANS simulations.

References

- [1] H. J. Kaltenbach et al. “Study of flow in a planar asymmetric diffuser using large-eddy simulation”. In: *Journal of Fluid Mechanics* 390 (July 1999), pp. 151–185. ISSN: 1469-7645. DOI: [10.1017/S0022112099005054](https://doi.org/10.1017/S0022112099005054). URL: http://journals.cambridge.org.proxy.lib.chalmers.se/article_S0022112099005054.
- [2] J. L. Aider and A. Danet. “Large-eddy simulation study of upstream boundary conditions influence upon a backward-facing step flow”. In: *Comptes Rendus Mécanique* 334.7 (2006), pp. 447–453. ISSN: 1631-0721. DOI: [http://dx.doi.org/10.1016/j.crme.2006.05.004](https://dx.doi.org/10.1016/j.crme.2006.05.004). URL: <http://www.sciencedirect.com/science/article/pii/S1631072106000866>.
- [3] P. Batten, U. Goldberg, and S. Chakravarthy. “Interfacing Statistical Turbulence Closures with Large-Eddy Simulation”. In: *American Institute of Aeronautics and Astronautics* 42.3 (2004), pp. 485–492. URL: <http://arc.aiaa.org/doi/abs/10.2514/1.3496>.
- [4] L. Davidson. “Using isotropic synthetic fluctuations as inlet boundary conditions for unsteady simulations”. In: *Advances and Applications in Fluid Mechanics* 1.1 (2007), pp. 1–35.
- [5] N. Jarrin et al. “A synthetic-eddy-method for generating inflow conditions for large-eddy simulations”. In: *International Journal of Heat and Fluid Flow* 27.4 (2006), pp. 585–593. ISSN: 0142-727X. DOI: [http://dx.doi.org/10.1016/j.ijheatfluidflow.2006.02.006](https://dx.doi.org/10.1016/j.ijheatfluidflow.2006.02.006). URL: <http://www.sciencedirect.com/science/article/pii/S0142727X06000282>.
- [6] R. Poletto, T. Craft, and A. Revell. “A New Divergence Free Synthetic Eddy Method for the Reproduction of Inlet Flow Conditions for LES”. English. In: *Flow, Turbulence and Combustion* 91.3 (2013), pp. 519–539. ISSN: 1386-6184. DOI: [10.1007/s10494-013-9488-2](https://doi.org/10.1007/s10494-013-9488-2). URL: <http://dx.doi.org/10.1007/s10494-013-9488-2>.
- [7] L. Davidson. “Large Eddy Simulations: how to evaluate resolution”. In: *International Journal of Heat and Fluid Flow* 30 (5 2009), pp. 1016–1025. ISSN: 0142-727X.
- [8] L. Davidson and S. H. Peng. “Embedded Large-Eddy Simulation Using the Partially Averaged Navier–Stokes Model”. In: *AIAA Journal* 51 (5 2013), pp. 1066–1079. ISSN: 0001-1452.
- [9] S.R. Hanna et al. “Comparisons of model simulations with observations of mean flow and turbulence within simple obstacle arrays”. In: *Atmospheric Environment* 36.32 (2002), pp. 5067–5079. ISSN: 1352-2310. DOI: [http://dx.doi.org/10.1016/S1352-2310\(02\)00566-6](https://dx.doi.org/10.1016/S1352-2310(02)00566-6). URL: <http://www.sciencedirect.com/science/article/pii/S1352231002005666>.
- [10] M. Klein, A. Sadiki, and J. Janicka. “A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations”. In: *Journal of Computational Physics* 186.2 (2003), pp. 652–665. ISSN: 0021-9991. DOI: [http://dx.doi.org/10.1016/S0021-9991\(03\)00090-1](https://dx.doi.org/10.1016/S0021-9991(03)00090-1). URL: <http://www.sciencedirect.com/science/article/pii/S0021999103000901>.
- [11] L. Davidson and M. Billson. “Hybrid LES-RANS using synthesized turbulent fluctuations for forcing in the interface region”. In: *International Journal of Heat and Fluid Flow* 27.6 (2006), pp. 1028–1042. ISSN: 0142-727X. DOI: [http://dx.doi.org/10.1016/j.ijheatfluidflow.2006.02.025](https://dx.doi.org/10.1016/j.ijheatfluidflow.2006.02.025). URL: <http://www.sciencedirect.com/science/article/pii/S0142727X06000488>.
- [12] L. Davidsson. “Fluid mechanics, turbulent flow and turbulence modeling”. In: *Chalmers University of Technology, Goteborg, Sweden (Nov 2011)* (2011).
- [13] R. H. Kraichnan. “Diffusion by a Random Velocity Field”. English. In: *Physics of Fluids* 13.1 (1970), p. 22. URL: www.summon.com.
- [14] T. S. Lund, X. Wu, and K. D. Squires. “Generation of Turbulent Inflow Data for Spatially-Developing Boundary Layer Simulations”. In: *Journal of Computational Physics* 140.2 (1998), pp. 233–258. ISSN: 0021-9991. DOI: [http://dx.doi.org/10.1006/jcph.1998.5882](https://dx.doi.org/10.1006/jcph.1998.5882). URL: <http://www.sciencedirect.com/science/article/pii/S002199919895882X>.

- [15] N. Jarrin. “Synthetic Inflow Boundary Conditions for the Numerical Simulation of Turbulence”. In: *PhD Thesis, Faculty of engineering and physical sciences, University of Manchester* (2008).
- [16] S. Wallin and A. V. Johansson. “An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows”. In: *Journal of Fluid Mechanics* 403 (Jan. 2000), pp. 89–132. ISSN: 1469-7645. DOI: 10.1017/S0022112099007004. URL: http://journals.cambridge.org.proxy.lib.chalmers.se/article_S0022112099007004.
- [17] L. Davidson. “Two-equation Hybrid RANS-LES Models: A Novel Way to Treat k and ω at the Inlet”. In: *Turbulence, Heat and Mass Transfer 8*, Begell House Inc. (2015).
- [18] G. Östling and E. Wiberger. “Potential of Rolls-Royce HYDRA Code for Ship Hull Performance Prediction”. In: *Report. X, Department of Shipping and Marine Technology, Chalmers University of Technology, Gothenburg, Sweden* 248 (2011).
- [19] A. Jameson. “Formulation of Kinetic Energy Preserving Conservative Schemes for Gas Dynamics and Direct Numerical Simulation of One-Dimensional Viscous Compressible Flow in a Shock Tube Using Entropy and Kinetic Energy Preserving Schemes”. English. In: *Journal of Scientific Computing* 34.2 (2008), pp. 188–208. ISSN: 0885-7474. DOI: 10.1007/s10915-007-9172-6. URL: <http://dx.doi.org/10.1007/s10915-007-9172-6>.

A Correlations

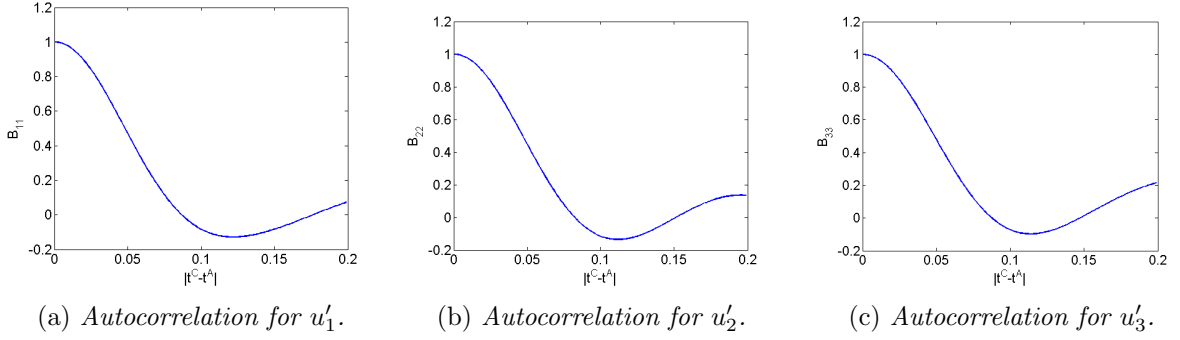


Figure A.1: Autocorrelation for the fluctuations generated with the method by Batten.

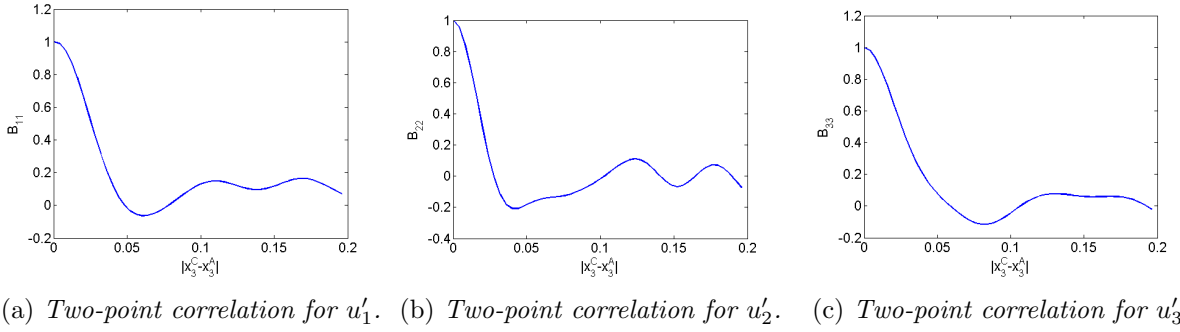


Figure A.2: Two-point correlation in the x_3 -direction for the fluctuations generated with the method by Batten.

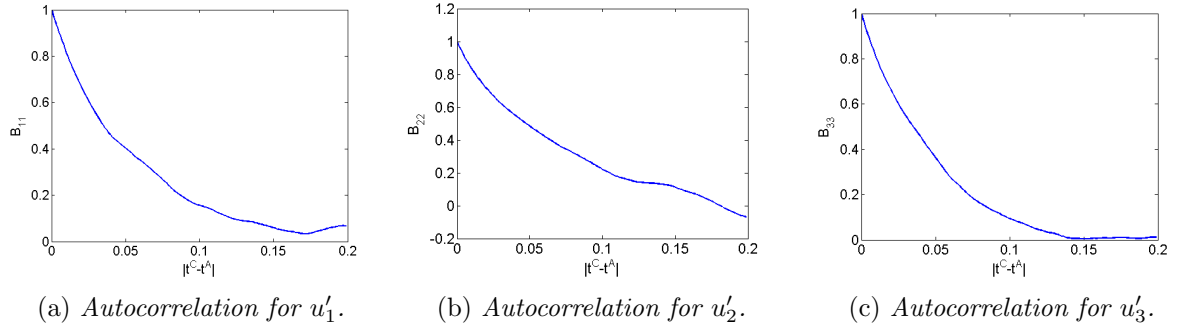
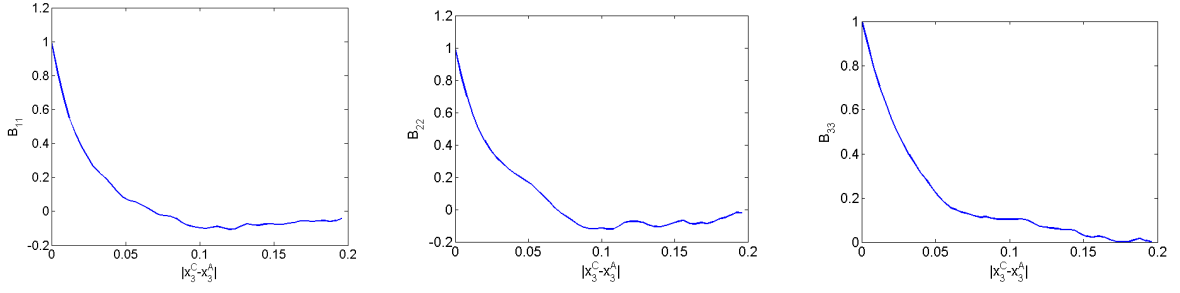
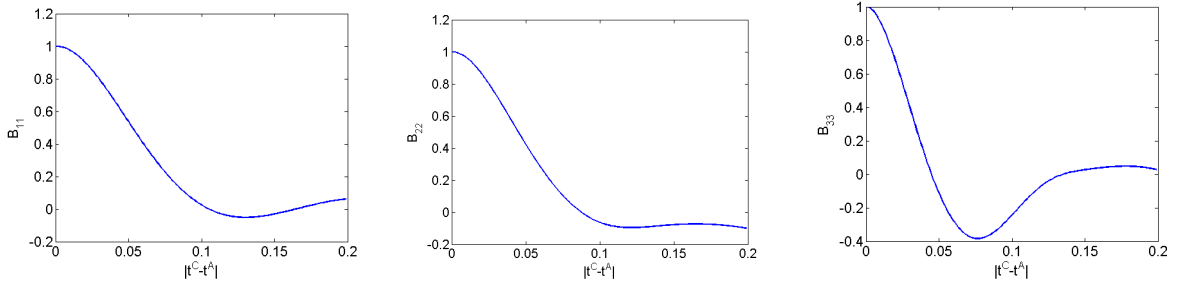


Figure A.3: Autocorrelation for the fluctuations generated with the method by Davidson.



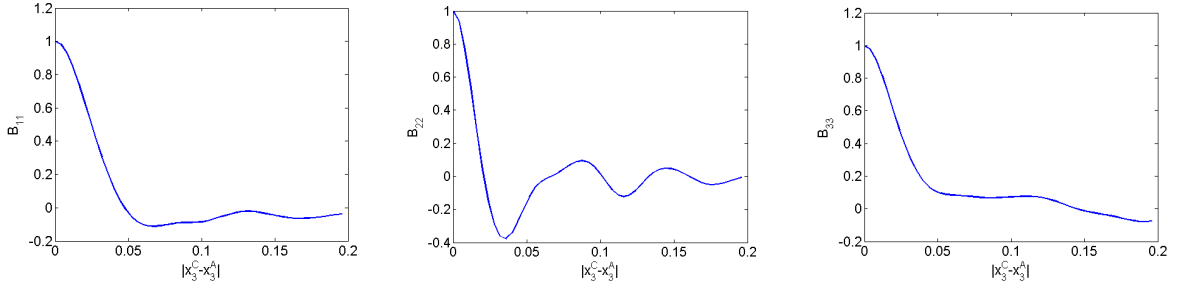
(a) Two-point correlation for u'_1 . (b) Two-point correlation for u'_2 . (c) Two-point correlation for u'_3 .

Figure A.4: Two-point correlation in the x_3 -direction for the fluctuations generated with the method by Davidson.



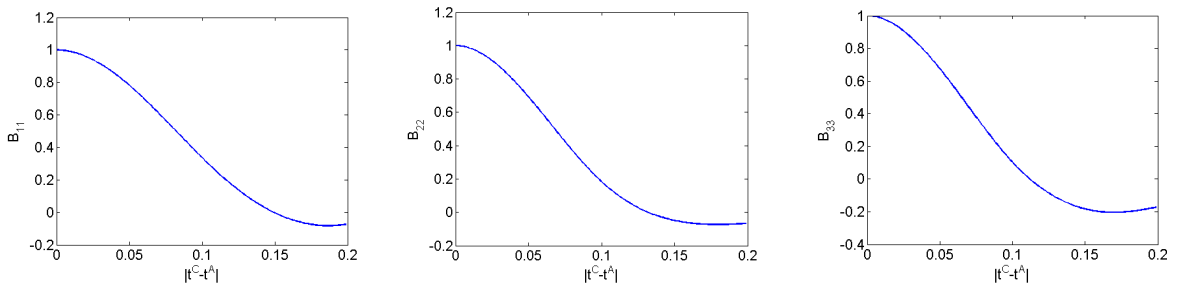
(a) Autocorrelation for u'_1 . (b) Autocorrelation for u'_2 . (c) Autocorrelation for u'_3 .

Figure A.5: Autocorrelation for the fluctuations generated with the DFSEM.



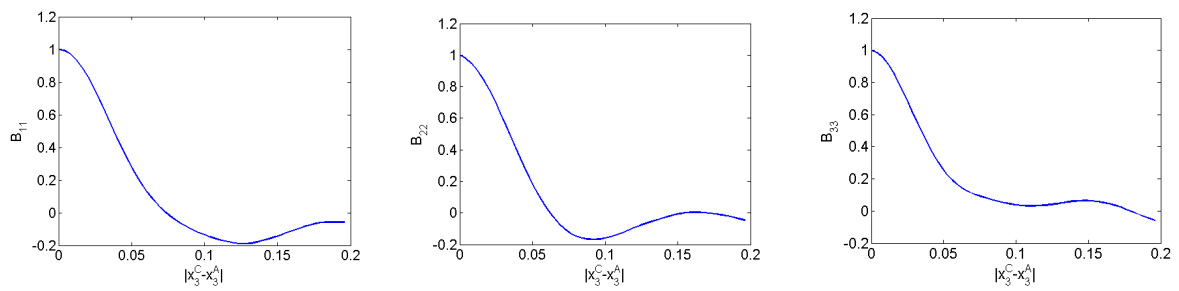
(a) Two-point correlation for u'_1 . (b) Two-point correlation for u'_2 . (c) Two-point correlation for u'_3 .

Figure A.6: Two-point correlation in the x_3 -direction for the fluctuations generated with the DFSEM.



(a) Autocorrelation for u'_1 . (b) Autocorrelation for u'_2 . (c) Autocorrelation for u'_3 .

Figure A.7: Autocorrelation for the fluctuations generated with the SEM.



(a) Two-point correlation for u'_1 . (b) Two-point correlation for u'_2 . (c) Two-point correlation for u'_3 .

Figure A.8: Two-point correlation in the x_3 -direction for the fluctuations generated with the SEM.