Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Airfoils in Turbulent Inflow

Gilling, Lasse

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Gilling, L. (2009). *Airfoils in Turbulent Inflow*. Department of Civil Engineering, Aalborg University.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

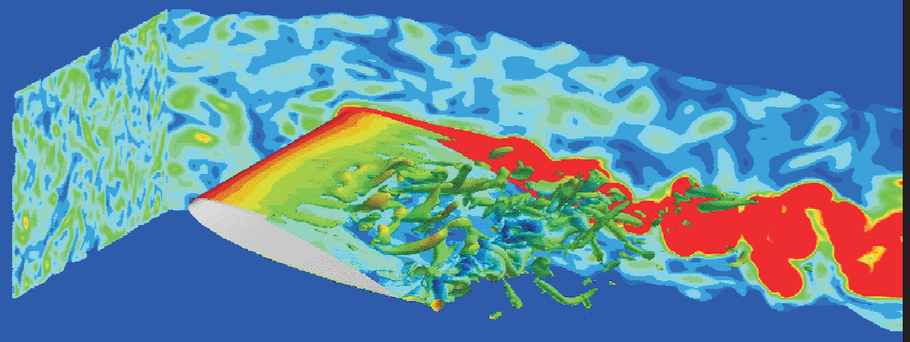
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: March 19, 2024

Airfoils in Turbulent Inflow

Airfoils in Turbulent Inflow



PhD Thesis
by
Lasse Gilling

ISSN: 1901-7294

DCE Thesis No. 23

Department of Civil Engineering, Aalborg University, October 2009

PhD Thesis by Lasse Gilling, AAU, 2009

Aalborg University
Department of Civil Engineering
Division of Structural Mechanics

DCE Thesis No. 23

Airfoils in Turbulent Inflow

by

Lasse Gilling

October 2009

© Aalborg University

Scientific Publications at the Department of Civil Engineering

Technical Reports are published for timely dissemination of research results and scientific work carried out at the Department of Civil Engineering (DCE) at Aalborg University. This medium allows publication of more detailed explanations and results than typically allowed in scientific journals.

Technical Memoranda are produced to enable the preliminary dissemination of scientific work by the personnel of the DCE where such release is deemed to be appropriate. Documents of this kind may be incomplete or temporary versions of papers—or part of continuing work. This should be kept in mind when references are given to publications of this kind.

Contract Reports are produced to report scientific work carried out under contract. Publications of this kind contain confidential matter and are reserved for the sponsors and the DCE. Therefore, Contract Reports are generally not available for public circulation.

Lecture Notes contain material produced by the lecturers at the DCE for educational purposes. This may be scientific notes, lecture books, example problems or manuals for laboratory work, or computer programs developed at the DCE.

Theses are monograms or collections of papers published to report the scientific work carried out at the DCE to obtain a degree as either PhD or Doctor of Technology. The thesis is publicly available after the defence of the degree.

Latest News is published to enable rapid communication of information about scientific work carried out at the DCE. This includes the status of research projects, developments in the laboratories, information about collaborative work and recent research results.

Published 2009 by
Aalborg University
Department of Civil Engineering
Sohngaardsholmsvej 57,
DK-9000 Aalborg, Denmark

Printed in Aalborg at Aalborg University

1st Edition 2009
ISSN 1901-7294
DCE Thesis No. 23

Preface

This dissertation is submitted in a partial fulfillment of the requirements for obtaining the degree of Doctor of Philosophy. The work was carried out from October 2006 to October 2009 at the Department of Civil Engineering at Aalborg University. The project was financed by the Danish Council for Strategic Research under the project Airfoils in Turbulent Inflow with project no. 2104-05-0074. The support is duly acknowledged.

I wish to thank my supervisor, professor Niels N. Sørensen, for his guidance and encouragement throughout the study, and for always being positive and taking time to discuss my problems. Andreas Bechmann and Pierre-Elouan Réthoré deserve my gratitude for sharing their codes for the modules they have made for EllipSys3D. Thanks to professor Søren R. K. Nielsen for his ideas and enthusiasm. I also wish to express my gratitude to Peter Fuglsang and Jesper Madsen from LM Glasfiber for sharing their experimental data. A special thanks to professor Lars Davidson for letting me stay at Chalmers University of Technology for an extended visit and for asking all the right questions. I am grateful for the support from my family, friends, and colleagues - especially from my fellow PhD-students. Finally, I will thank my girlfriend, Pernille, for cheering me up during the set-backs, and for her love and patience throughout the project.

Aalborg, October 2009

Lasse Gilling

Summary

Wind turbines operate in inflow turbulence whether it originates from the shear in the atmospheric boundary layer or from the wake of other wind turbines. Consequently, the airfoils of the wings experience turbulence in the inflow. The main topic of this thesis is to investigate the effect of resolved inflow turbulence on airfoil simulations in CFD.

The detached-eddy simulation technique is used because it can resolve the inflow turbulence without becoming too computationally expensive due to its limited requirements for mesh resolution in the boundary layer. It cannot resolve the turbulence that is formed in attached boundary layers, but the freestream turbulence can penetrate the boundary layer. The idea is that the resolved turbulence from the freestream should mix high momentum flow into the boundary layer and thereby increase the resistance against separation and increase the maximum lift. However, it turns out that the velocities in the inner part of the boundary layer only increase slightly, and there is no effect on the obtained surface pressures or lift coefficients. It appears that the resolved turbulence has a too large length scale to cause the effect as seen in experiments. This indicates that it might be necessary to run large-eddy simulations or direct numerical simulations to capture the effect of inflow turbulence.

A method for imposing resolved turbulence inside the domain in a CFD simulation is also presented. The idea is that by imposing the resolved turbulence immediately upstream of the airfoil the total number of computational cells can be reduced, which in turn reduces the total computational cost of the simulations. The method can be used in any simulation where inflow turbulence is included and is therefore not limited to the application investigated in this thesis. It is shown that synthetic turbulence can be imposed successfully since the obtained turbulence accurately resembles the synthetic field. The synthetic turbulence should be run through a very short precursor simulation and it is best imposed by adding a term to the

source terms of the discrete Navier-Stokes equations.

Finally, two methods for generating synthetic turbulence with different characteristics are given. One is a semi-analytical solution to the cross-spectral densities for turbulence in the atmospheric shear. The method is based on the same theory as the Mann method. The second method generates synthetic turbulence in arbitrary domains. The purpose is to generate a synthetic turbulence field corresponding to the field encountered by a rotating blade.

Dansk Resumé (Danish Summary)

Vindmøller er placeret i det atmosfæriske grænselag, hvor de er udsat for turbulens. Det gælder især møller i vindmølleparker, som rammes af kølvandet fra andre møller. Derfor opererer vingeprofilerne på vingerne i en turbulent strømning. I denne afhandling undersøges effekten af at opløse indløbsturbulensen i profilberegninger med CFD.

Simuleringerne er udført som detached-eddy simuleringer (DES), fordi det dermed er muligt at opløse fristrømsturbulensen uden at simuleringerne bliver for tunge at køre, da kravet til netopløsningen i grænselaget ikke er så strengt. I DES opløses turbulensen der dannes i grænselaget ikke, men turbulensen fra den fri strømning trænger ind i grænselaget. Idéen er, at denne turbulens skal blande momentum fra den fri strømning ind til bunden af grænselaget, hvilket vil øge modstandsevnen mod trykgradienten, som ellers kan få strømningen til at separere. Dermed øges det maximale løft. Det viser sig dog, at hastigheden i den indre del af grænselaget kun påvirkes svagt af den opløste turbulens. Der er en lille stigning i hastighed tæt på forkanten, men længere nedstrøms er der ingen ændring. Påvirkningerne på det beregnede løft og på trykfordelingerne er også meget små. De opløste skalaer af turbulensen er forholdsvis store sammenlignet med grænselagstykkelsen. Det vurderes, at det er nødvendigt at opløse de små skalaer af turbulensen i grænselaget, og at det er dem, som samvirker med fristrømsturbulensen i eksperimenter, hvor der observeres en effekt. Det betyder, at det sikkert er nødvendigt at køre large-eddy simuleringer eller direkte løse Navier-Stokes ligningerne uden brug af turbulensmodel for at kunne forudsige effekten af fristrømsturbulens.

Der er udviklet en metode til at påtvinge opløst turbulens i et ønsket område inde i domænet i CFD simuleringer. Metoden kan anvendes istedet for den traditionelle fremgangsmåde, hvor turbulensen påtvinges på

indløbsranden. Metoden blev udviklet til at påtvinge opløst turbulens i profilsimuleringerne, hvorved det totale antal celler kan reduceres til godt det halve, hvilket reducerer beregningsbyrden betydeligt. Den kan dog også anvendes til andre typer simuleringer, hvor der påføres opløst turbulens. F.eks. kan den anvendes til studier af kølvandet fra rotorer, der opererer i en turbulent strømning. Den opløste turbulens påtvinges bedst ved at addere et led til kildeleddet i den diskrete udgave af Navier-Stokes ligningerne. Den dannede turbulens viser meget god overensstemmelse med det påtrykte felt. Det anbefales, at køre syntetisk turbulens gennem en kort precursor simulering inden det påføres, for at lade flowløseren tilpasse turbulensen til nettet og den numeriske metode.

Endeligt er der udviklet to metoder til at generere syntetisk turbulens med samme karakteristika som atmosfærisk turbulens og turbulensen som den opleves af en roterende vinge. Den første metode giver en semi-analytisk løsning til kryds-spektraltætheden af turbulensen i et forskydningslag. Den er baseret på samme teori som Manns metode, som også anvendes i normen for lastberegninger på vindmøller. Den anden metode genererer syntetisk turbulens i domæner med vilkårlig form. Den kan anvendes til at generere et turbulensfelt, som svarer til det en roterende vinge vil møde. Rotationen ændrer fordelingen af energi på de forskellige frekvenser, og denne effekt kan modelleres med den foreslåede metode.

List of Scientific Publications

This thesis is written as a collection of papers. It includes a description of the background for the papers and a presentation of the results. The papers included in this thesis are [1–5]

Technical Report: Lasse Gilling (2009) *TuGen - Manual and User's Guide*, DCE Technical Report No. 76, Department of Civil Engineering, Aalborg University, Aalborg, Denmark, ISSN: 1901-726X.

Paper I: Lasse Gilling, Niels N. Sørensen and Lars Davidson (2009) The Effect of Resolved Turbulence in Detached-Eddy Simulations of an Airfoil, *Submitted to Wind Energy*.

Paper II: Lasse Gilling and Niels N. Sørensen (2009) Imposing Resolved Turbulence in CFD Simulations, *Submitted to Wind Energy*.

Paper III: Lasse Gilling, Bo J. Pedersen and Søren R. K. Nielsen (2009) Three Dimensional Turbulence Modeling in Shear Flow with Application to Wind Turbines, *Submitted to Wind Energy*.

Paper IV: Lasse Gilling, Søren R. K. Nielsen and Niels N. Sørensen (2009) Generation of Synthetic Turbulence in Arbitrary Domains, *ICOSSAR 2009*, September 13–17, Osaka, Japan.

The publications listed above are to be found in the second half of this thesis. Other publications are not included in the thesis because their content is covered in the newer publications above. The publications not included in the thesis are

Paper V: Lasse Gilling and Niels N. Sørensen (2008) Large Eddy Simulation of an Airfoil in Turbulent Inflow, *4th PhD Seminar on Wind Energy in Europe*, October 1–2, Magdeburg, Germany.

Paper VI: Lasse Gilling, Niels N. Sørensen and Lars Davidson (2009) Detached Eddy Simulation of an Airfoil in Turbulent Inflow, *47th AIAA Aerospace Sciences Meeting*, January 5–8, Orlando, Florida. AIAA paper no. 2009-270.

Paper VII: Lasse Gilling, Niels N. Sørensen and Pierre-Elouan Réthoré (2009) Imposing Resolved Turbulence by an Actuator in a Detached Eddy Simulation of an Airfoil, *European Wind Energy Conference and Exhibition (EWEC 2009)*, April 16–19, Marseille, France.

Contents

Preface **i**

Summary **iii**

Dansk Resumé (Danish Summary) **v**

List of Scientific Publications **vii**

1 Introduction **1**

2 Choice of Methods **5**
2.1 Computational Tools 6
2.2 Method for Generating Inflow Turbulence 6
2.3 Simulation Technique 7
2.3.1 Description of the methods 8
2.3.2 Choice of method 10
2.3.3 Subgrid model 10
2.4 DES Flavor 11
2.5 Laminar-Turbulent Transition Model 15

3 Spatial Decay of Homogenous Turbulence **19**
3.1 Cost of Simulating Spatial vs. Temporal Decay 20
3.2 Results 20

4 Detached-Eddy Simulation of the Airfoil in Resolved Inflow Turbulence **23**
4.1 Results 23
4.2 Discussion and Conclusions 26

5 Related Work **29**
5.1 Imposing Resolved Turbulence Inside Domains 29
5.1.1 The Basic Idea 29
5.1.2 Two Approaches 30
5.1.3 Results 30
5.1.4 Possible Applications 34
5.2 Three Dimensional Turbulence Modeling in Shear Flow . . . 35
5.2.1 Background 35
5.2.2 Description of the model 36
5.2.3 Applications 36
5.3 Synthetic Turbulence in Arbitrary Domains 37
5.3.1 Motivation 38
5.3.2 Description of the Method 39
5.3.3 Examples 39

6 Conclusions **43**

Bibliography **45**

Technical Report: TuGen, Synthetic Turbulence Generator - Manual and User's Guide

Paper I: The Effect of Resolved Turbulence in Detached-Eddy Simulations of an Airfoil

Paper II: Imposing Resolved Turbulence in CFD Simulations

Paper III: Three Dimensional Turbulence Modeling in Shear Flow with Application to Wind Turbines

Paper IV: Generation of Synthetic Turbulence in Arbitrary Domains

Chapter 1

Introduction

Wind turbines operate in the atmospheric boundary layer and are therefore subjected to inflow turbulence. Especially, the turbines in wind farms can experience high turbulence intensities when they are located in the wakes of other turbines. The performance of the airfoil sections of the blades are affected by the inflow turbulence. McKeough [6] and Hoffmann [7] both investigated the influence of turbulence on the NACA 0015 profile at Reynolds number around $250,000$. They found that the inflow turbulence extended the linear region of the lift curve and thereby increased the maximum lift.

While the effect is fairly easy to investigate in a wind tunnel it has not received much attention from the CFD society. One might speculate that this is due to the reason that the main applications of airfoils have been in aeronautics and astronautics, and here the inflow is largely laminar.

The inflow turbulence has two effects on the boundary layer on the airfoil. The first is to trigger the transition from laminar to turbulent boundary layer. This effect can be predicted by transition models (e.g. the $\gamma - Re_\theta$ model [8, 9]) and the effect of increased turbulence is to move the transition point upstream which gives a lower lift of the airfoil. The second effect of inflow turbulence is to increase the transport of momentum from the free flow to the inner region of the boundary layer. This will increase the resistance against adverse pressure gradients and thereby reduce the separation at high incidences. The result is that the linear range of the lift curve is extended and that the maximum lift is increased. This principle is employed by vortex generators. From the experimental data from studies of inflow turbulence it is clear that at least for the NACA 0015 the latter effect dominates the first, since the observed effect is an increase in maximum lift.

The position of the transition point can be predicted by transition models on the basis of the modeled kinetic energy in the inflow. On the other

hand, the transport of momentum to the inner region of the boundary layer is caused by the eddies from the free flow entering and mixing the flow in the boundary layer, and these eddies must be resolved to give the desired mixing. Therefore, a turbulence resolving technique must be used to predict both effects.

As will be discussed in Chapter 2, the large-eddy simulation (LES) and direct numerical simulation (DNS) techniques have a prohibitively high computational cost for them to be applied. Further, in simulations with the unsteady Reynolds-averaged Navier-Stokes equations (URANS) the resolved turbulence is suppressed by the high eddy viscosity. This leaves detached-eddy simulation (DES) in which the turbulence outside the boundary layer can be resolved and the boundary layer is treated by a URANS model.

Since the boundary layer in a DES is treated by an URANS model the mesh resolution does not have to be fine enough to resolve the small turbulence structures that forms in the boundary layer. However, the mesh is fine enough to resolve the comparably large scales from the freestream turbulence. The idea is to use DES with resolved inflow turbulence and let the unsteady forcing from the LES region make the flow in the boundary layer very unsteady, and thereby transport some of the freestream momentum into the inner region of the boundary layer. This should give at least some of the effect of freestream turbulence on the airfoil flow.

The development of DES was motivated by the flow over an wing [10]. Since then, it has frequently been applied to airfoil flows [11–16] where the inflow turbulence was modeled by the turbulence model. DES has also been applied to flows with multiple structures, where the wake from one structure impinges on other structures. Some examples are the multi-element airfoil in [15], the landing gear in [17], the tail of the fighter aircraft in [18], the rod-airfoil configuration in [19], and the two wall-mounted cubes in tandem [20]. These applications rely on the DES model to capture the effect of inflow turbulence.

A measurement campaign in the LM Wind Tunnel was carried out by LM Glasfiber in corporation with Risø-DTU [21–24]. They studied different airfoils with and without a turbulence grid placed upstream of the airfoil. The measured lift, drag and surface pressure will be used for comparison with the simulated results. At high angles of attack where the airfoil is stalled the flow is dependent on the geometry of the wind tunnel. This dependency is typically minimized by applying wall corrections to the obtained results, but in the present study it is decided to model the airfoil in the tunnel. Thereby, the influence of the tunnel walls should be similar in the experiment and in the simulations and the measured data can be compared directly to the simulated results.

In the simulations the resolved turbulence is imposed at the inlet. This gives a high computational cost compared to standard DES without resolved inflow turbulence. To minimize this computational overhead some methods

are developed to impose the resolved turbulence immediately upstream of the airfoil. Further, two different techniques are developed for generating synthetic turbulence with different characteristics.

Thesis Outline

The present thesis is a collection of papers. It consist of a technical report, three journal papers and a conference paper, from which the main results are presented and discussed over the first 49 pages of the thesis.

The main topic is the study of the effect of resolved inflow turbulence. The methods applied in this study are described in Chapter 2. To validate the method the spatial decay of homogenous, isotropic turbulence is investigated in Chapter 3. The main findings in the study are given in Chapter 4 where the main results from Paper I are given. It is concluded that the effect of resolved inflow turbulence cannot be captured by the simulations. Next, follows a description of the results obtained in the remaining papers. Common for Paper II–IV are that the methods were intended to be used with the airfoil flow, but they can be used in other situations as well. Finally, Chapter 6 presents the main conclusions from the study. The papers are appended after the list of references.

Chapter 2

Choice of Methods

For the simulation of the airfoil flow with resolved inflow turbulence a number of methods have been applied. Paper I describes and analyses the outcome of the applied methods, but it only contains a short description of the choice of methods. This chapter serves to give the full discussion of the most appropriate methods for the problems encountered.

Prior to describing the different methods used, the computational setup will be briefly presented. A sketch of the domain for the airfoil simulation is shown in Figure 2.1. The domain models an airfoil in a wind tunnel. The Reynolds number is 1.6 million. Symmetry boundary conditions are used on the top and bottom boundaries to limit the flow without having to resolve the boundary layer of a no-slip wall. For the same reason the boundaries in the spanwise directions are periodic.

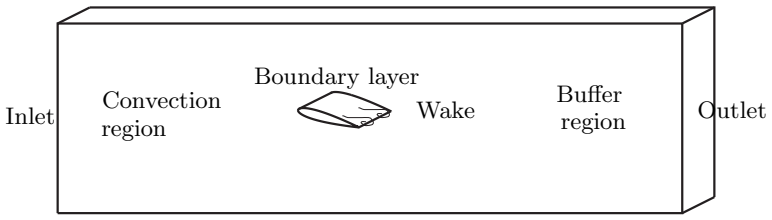


Figure 2.1: Sketch of computational domain used in airfoil flow.

The sketch shows four different regions in the flow field. The turbulence field is imposed at the inlet. The region between the inlet and the leading edge of the airfoil is marked convection region and it has two purposes. One is to determine the induced velocity field upstream of the airfoil and therefore it must not be too short. The other is to transport the resolved

turbulence from the inlet to the leading edge of the airfoil. This requires a fine resolution of the domain to prevent a too high un-physical decay of the resolved turbulence. The mesh in this region consists of cubic cells to minimize numerical dissipation. Close to the walls of the airfoil the mean velocity profile in the boundary layer is resolved by an "O-mesh". The wake behind the airfoil is resolved with the same resolution as in the convection region. Finally, in the region prior to the outlet the cells are stretched in the streamwise direction to dampen the fluctuations and prevent recirculating flow passing through the outlet.

2.1 Computational Tools

The Navier-Stokes equations are solved with the in-house CFD code EllipSys3D by Michelsen [25, 26] and Sørensen [27]. The code uses a structured multi-block mesh with co-located variables and it is parallelized with MPI.

The HypGrid-code by Sørensen [28] is used for generating the two-dimensional body-fitted "O-mesh" used in the simulations of the airfoil.

The synthetic turbulence field used for the inflow is generated with TuGen, which is my implementation of Mann's method [29] of generating synthetic turbulence. It is described below and in [1].

2.2 Method for Generating Inflow Turbulence

The inlet boundary condition should contain a constant mean velocity field superposed by a fluctuating turbulence field. The turbulence field is obtained from synthetic turbulence which is then imposed on a precursor simulation. From the precursor simulation the turbulence is sampled and this sampled field is used for the airfoil simulation.

To reduce the required length of the precursor simulation the synthetic field should be a close approximation to the governing Navier-Stokes equations. Further, it should match the geometry and boundary conditions of the computational domain. Therefore, the synthetic turbulence field should satisfy these demands

- It should contain a time history of a two-dimensional field corresponding to the inlet plane.
- Vector field should consist of three-dimensional velocity vectors.
- The three components of the velocities must be correlated to yield a divergence free field.
- The turbulence field should be homogenous and isotropic.
- The velocity field should possess the correct correlation with a desired integral length scale.
- The field should be periodic in the spanwise direction.

The method of Mann [29, 30] satisfies all these demands and is therefore chosen for generating the synthetic turbulence. The method is based on the Fourier simulation technique by Shinozuka and Jan [31] and the rapid distortion theory by Townsend [32]. The method has been implemented in Fortran as described in the technical report [1], which contains a manual and user's guide for the program. The source code can be downloaded freely.

The method of Mann generates a three-dimensional frozen field of turbulence. The discrete field consists of $N_x \times N_y \times N_z$ points arranged in a three-dimensional Cartesian grid. Due to the applied FFT algorithm the field is periodic in all three directions. The field represents a realization of a three-dimensional velocity field at a given instant $v_j(x, y, z, t = 0)$. By application of Taylor's hypothesis the frozen three-dimensional velocity field can be considered as a time series of a two-dimensional velocity field $v_j(x = 0, y, z, t)$. This time series of the two-dimensional velocity field is exactly what is needed for the inlet boundary condition.

The synthetic turbulence fields generated by the method of Mann are divergence free when isotropic turbulence fields are generated (see further details in the technical report [1]). Further, the auto- and cross-correlation agree with the theoretic correlation functions by von Karman [33].

The synthetic turbulence is run through a precursor simulation to let the flow solver adapt the turbulence to the computational method and to the mesh. In a plane orthogonal to the mean velocity the turbulent velocity field is sampled in the precursor. The sampled turbulence field has an integral length scale about $0.25c$ where c is the chord of the airfoil.

The sampled velocity field is scaled in intensity to vary the turbulence intensity without changing the integral length scale. To obtain a desired turbulence intensity all three velocity components are multiplied by a common factor a. As described in Chapter 3 the subgrid parameters (k and ω) are determined from the precursor simulation. They are scaled with the same factor a as used for the scaling of the three velocity components. Since $k_{sgs} \propto u^2$ the subgrid kinetic energy is scaled by a^2. $\omega \propto \sqrt{k}/L_{sgs}$ and thereby ω should be scaled by a since the subgrid integral length scale is constant.

2.3 Simulation Technique

The argumentation for the choice of the DES technique is given in the present section. First, the most common approaches for transient flows are described followed by the choice of technique. Finally, the employed $k - \omega$ SST subgrid model by Menter [34] is compared to the Smagorinsky model.

2.3.1 Description of the methods

The candidates for the choice of method are the following unsteady techniques. They are ordered after the level of detail contained in the solution starting with the lowest level. Further, the computational burden is related to the obtained detail. Therefore, the first listed methods are the cheapest to run while the last has the highest computational burden.

URANS: Turbulence is modeled by the turbulence model. Only the largest and slowest variations in the flow field are resolved. Small scale turbulence is suppressed by the high eddy viscosity from the turbulence model.

DES: Hybrid of LES and URANS where the switch between the two branches is given by the mesh (and by the flow for the DDES technique [15]). The original intent was to run URANS in the wall boundary layers and in regions with laminar flow. In the regions of separation and in the wake the flow is modeled by a LES model that resolves the largest scales of the turbulence.

LES: The largest energy carrying scales of the turbulence are resolved and the small scale turbulence is modeled by a subgrid turbulence model.

DNS: All fluctuations are resolved and hence no model for the turbulence is used.

In Figure 2.2 a generalized sketch of an energy spectrum is shown. Three ranges of wave numbers are marked. The wave number $k = 2\pi/L$ is defined from the dimension L of the eddies. (I) The largest energy carrying eddies. The turbulence field receives energy from the mean flow and this energy is pumped into the turbulence field at the largest scales. The largest eddies are

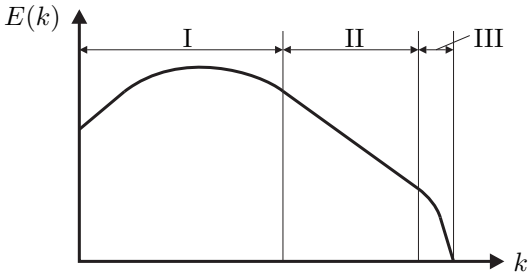


Figure 2.2: Log-log sketch of energy spectrum divided into three domains: (I) The largest energy carrying eddies where energy is added from the mean flow. (II) Inertial subrange with isotropic eddies. (III) Smallest eddies dominated by the molecular viscosity which extracts energy from the turbulence.

therefore heavily influenced by the geometry of the flow. (II) In the inertial subrange the eddies are isotropic. They dissipate energy from the largest eddies and loose energy to the smallest eddies. The spectrum is linear in this range in a double log representation and it has a slope of $-5/3$. (III) The smallest scales are given by the molecular viscosity. In this range the energy is dissipated into heat by the molecular viscosity.

The techniques URANS, LES and DNS can schematically be characterized by the range of wave numbers they resolve. In Figure 2.3 the three methods are compared. In reality the boundary between the resolved and modeled kinetic energy is not sharp as shown in the figure.

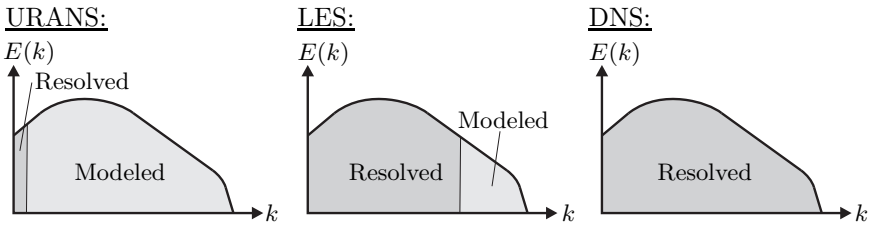


Figure 2.3: Modeled and resolved scales for different turbulence modeling strategies.

DES is based on the fact that LES is often too computational expensive at industrial Reynolds numbers and that most of the cost originates from the treatment of the boundary layer close to walls. In this region URANS is quite successful with the fine tuned turbulence models, but URANS fails to give the correct turbulence content in the separated regions. By creating a hybrid of the two techniques their advantages can be gathered in a single method. The fine tuned turbulence model of URANS is used in the boundary layer and the raw power of LES is used in the separated regions.

A single turbulence model is used in DES. In the URANS regions it works like a standard turbulence model while in the LES regions it works as a subgrid-scale (SGS) model. When using the Spalart-Allmaras turbulence model the switch between the two branches of DES is obtained exclusively through the grid spacing in the original formulation [10]. In the newer DDES [15] also information from the computed flow field is used. Since the switch in the original formulation is only dependent on the mesh it is very dependent on competent users as it easily can be misused and run on too coarse or too fine meshes. In these cases DES will yield unreliable results. For the DES formulation based on the $k-\omega$ SST subgrid model the conceptual difference between the original formulation [13] and the shielded formulation [14] is less obvious, but still severe problems are encountered if the mesh resolution is not adequate. This issue is further discussed in Section 2.4.

2.3.2 Choice of method

The choice of computational technique is simply to rule out the infeasible approaches.

The method employed for the analysis in the present work should resolve (some) turbulence. Hence, the URANS technique cannot be used as only the very largest scales are resolved and the turbulence is suppressed by the high eddy viscosity from the turbulence model.

In the original DES paper [10] a cost estimate is given for a full LES of an airliner wing without separation. Inclusion of separation would increase the cost. At $Re = 6.5 \times 10^6$ the required number of cells is in the order of 10^{11} and the number of time steps is about 10^6. Thereby, this simulation is estimated to be possible to run in a "Grand Challenge" around year 2030. The span used in the computation was 8 chords, and the Reynolds number is higher than in the simulations in Paper I. Still, correcting for these differences the required number of cells in a LES of a section of the airfoil spanning 1.5 chords at $Re = 1.6 \times 10^6$ would be about 2×10^{10}. For comparison 21 million cells were used in the DES presented in Paper I and the wall time used for each simulation was several days. Therefore, it is clearly infeasible to run full LES which have more than 10^3 times higher computational cost.

The required cost also rules out the DNS technique. This leaves DES which can resolve the largest scales of the turbulence except in the boundary layer of walls where a URANS model is used.

2.3.3 Subgrid model

The $k - \omega$ SST subgrid model of Menter [34] is used in all computations. It is chosen for its performance in predicting separation which is the most important flow feature for the analyzed airfoil flow.

In the LES regions the $k - \omega$ SST model works as Smagorinsky-like subgrid model, where the eddy viscosity approximately is proportional to the absolute value of the strain rate tensor. In the Smagorinsky subgrid model [35] the eddy viscosity is determined by

$$\nu_{smag} = (C_S \Delta)^2 |S| \tag{2.1}$$

where C_S is the Smagorinsky constant, Δ is the grid spacing and $|S|$ is the absolute value of the shear strain tensor. In [35] Smagorinsky suggests a value of the constant in the range $0.1 - 1.0$, but since then it has been found to vary between 0.065 [36] and 0.25 [37]. In [38] Davidson derives the constant to be 0.17.

To compare the applied $k - \omega$ SST model with the Smagorinsky model the value of the Smagorinsky constant that should have been used to give

the obtained eddy viscosity is determined. The equivalent Smagorinsky "constant" is

$$C_S = \frac{1}{\Delta}\sqrt{\frac{\nu_{sgs}}{|S|}} \tag{2.2}$$

where ν_{sgs} is the eddy viscosity obtained from the $k-\omega$ SST model. In Figure 2.4 the obtained values of the Smagorinsky constant are shown in the airfoil simulation. For the applied $k-\omega$ SST model to work as a Smagorinsky subgrid model the obtained eddy viscosity should be proportional to the magnitude of the strain tensor and to the square of the grid spacing. This in turn implies that the equivalent Smagorinsky "constant" should in fact be constant. As seen on the figure this is not the case, but in the free stream it is not far from true. In the free stream far from the airfoil the average value of the equivalent "constant" is 0.21 with a variation coefficient in the order of 15%. In conclusion the $k-\omega$ SST model works approximately like the Smagorinsky model in the free stream.

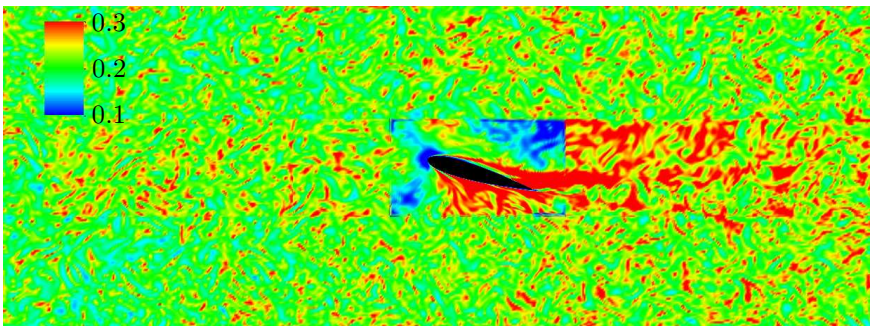


Figure 2.4: Plot of instantaneous equivalent Smagorinsky "constant".

In the free stream far from the airfoil the cells are close to cubic, and here obtained values agree quite well with the analytical value and the findings of [36, 37]. Close to the airfoil an O-mesh is used, and at the corners of the O-mesh the cells have high aspect ratios and stretching rates. Therefore, the O-mesh is visible in the plot with values deviating from the values in the cubic cells. Further, there is a difference in the definition of Δ between the Smagorinsky model where $\Delta \equiv \sqrt[3]{\Delta x \Delta y \Delta z}$ and the DES model where $\Delta \equiv \max(\Delta x, \Delta y, \Delta z)$.

2.4 DES Flavor

The original formulation of DES by Spalart *et al.* [10] used the Spalart-Almaras turbulence model [39]. It was motivated by airfoil flows at high

Reynolds numbers where LES is infeasible due to the required mesh resolution in the boundary layer as discussed in Section 2.3.2. The intention of the model is to activate the RANS branch of the model in boundary layers and activate the LES branch in the separated regions. Thereby, the detached eddies are resolved while the small eddies in the boundary layers are modeled by the RANS model.

The original formulation will be denoted DES97 from this point as proposed by [15]. In DES97 the choice of the LES or RANS branch is entirely based on the mesh and the distance from the nearest wall though the switch

$$\tilde{d} = \min(d, C_{DES}\Delta) \tag{2.3}$$

which replaces the wall distance d in the destruction term of the original Spalart-Allmaras turbulence model. Δ is here defined as the maximum side length of the cell instead of the typical definition in LES of $\sqrt[3]{\Delta x \Delta y \Delta z}$. The idea is that outside the boundary layer the destruction of eddy viscosity is increased, which in turn reduces the eddy viscosity and allows LES content to form in the solution. In the boundary layer the mesh is too coarse to resolve the LES content so in this region the modeled stresses should prevail.

A concern of the DES97 formulation is the "grey area" between the RANS and the LES regions. If the mesh is fine enough to make the model switch to the LES branch but too coarse to resolve the flow structures and thereby generate resolved stresses there will be a lack of stresses. The problem was clearest criticized by Menter and Kuntz [14] who showed that excessive grid refinement leads to premature separation. To circumvent the problem of grid induced separation (GIS) it was proposed to shield the RANS layer by the F_2 function of the $k-\omega$ SST model. The change might seem small, but it is a qualitative change of the model. Now, the choice of LES or RANS branch is not only governed by the mesh but also by the computed flow. Spalart *et al.* [15] adopted the idea and used it with the Spalart-Allmaras turbulence model and called the resulting model delayed DES (DDES). The lack of modeled stresses is called modeled stress depletion (MSD).

In Paper I the DDES formulation is used with the F_2 shielding function to prevent MSD. As will be shown in the following it is necessary for the flow case of interest and with the applied mesh resolution.

In [40, 41] we used the DES97 formulation for the airfoil flow with turbulent inflow. We believed the grid to be too coarse close to the wall to cause any problems with MSD. We found the flow to be very dependent on the intensity of the resolved inflow turbulence. An increase in the resolved turbulence intensity decreased the maximum lift, which is contradicted by experiments. There were two problems with results supporting that conclusion, and these are stretched in the following.

First, we have found that it is not the resolved stresses that change the solution but rather the modeled stresses. This is further outlined in

Section 2.5 and Chapter 4. When the intensity of the resolved turbulence field is increased the subgrid stresses also increase. It does not make sense to scale the resolved turbulence without scaling the subgrid parameters as well, so an effect of variations in subgrid parameters will *appear* as an effect of the resolved stresses in a parametric study of the intensity of resolved stresses.

Second, the simulations were carried out with the DES97 formulation and they suffered from massive MSD and thereby GIS. We believed that the grids used were too coarse to cause problems with MSD. After publication of the papers we discovered upon a closer investigation that it was an erroneous assumption. Simulations of airfoils with separations are especially prone to MSD as the boundary layer thickens prior to separation. Thereby, the boundary layer can grow to thicknesses of several percent of the airfoil chord and still the entire boundary layer should be covered by the RANS region.

The MSD problem was further worsened by the high levels of the modeled kinetic energy for the cases with high resolved kinetic energy. The turbulence intensity corresponding to the subgrid kinetic energy is about one quarter of the resolved turbulence intensity. With a resolved turbulence intensity of 2% the modeled turbulence intensity is then 0.5%, which is higher than the traditional range around $0.01\% - 0.1\%$. The SST model has a limit on the production of k intended to be active around the stagnation point. For the high eddy viscosity in this type of simulation the limiter is also activated in a large region upstream of the leading edge. This decreases the magnitude of k, which in turn lowers the modeled length scale $L_t = k^{1/2}/(\beta^*\omega)$ and thereby advances the switch to the LES branch.

The resulting RANS region in the DES97 of the airfoil flow is shown in Figure 2.5. It is very thin and close to the separation point it only covers the bottom of the boundary layer. Outside the RANS region the cells

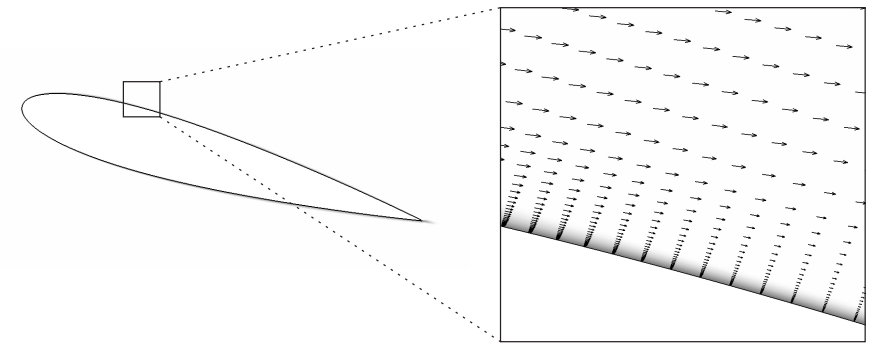


Figure 2.5: RANS region and boundary layer in the DES97 formulation. The RANS region is shown by the gray shading.

are still considerable longer in the wall parallel direction than the distance from the wall. Thereby, they are far too coarse to resolve even the largest turbulence scales in this part of the boundary layer and consequently the resolved stresses are low. Since the subgrid model has switched to the LES branch the modeled stresses are also low, and the result is a general lack of stresses.

Figure 2.6 shows the RANS region in a DDES where the RANS region is shielded by the F_2 function of the $k-\omega$ SST model. In this case the RANS region covers the entire boundary layer and MSD is avoided.

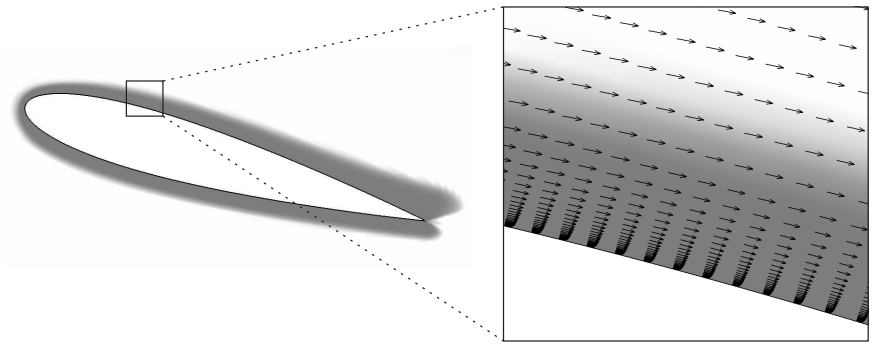


Figure 2.6: RANS region and boundary layer in the DDES formulation where the boundary layer is shielded by the F_2 function of the $k-\omega$ SST model. The RANS region is shown by the gray shading.

The differences in the computed airfoil characteristics are very large as shown in the Figures 2.7 and 2.8. Figure 2.7 shows the computed lift curves with a turbulence intensity of the resolved turbulence of 2%. The experimental data is from the LM Wind Tunnel and they are further described in Paper I. The simulations with the DES97 formulation give too low maximum lift and a very short linear region which only extends up to angles of attack about 8°. At higher incidences a trailing edge separation bubble starts to form on the suction side. The DDES with the RANS region shielded by the F_2 function gives much higher lift and a longer linear region. Although the shape of the lift curve is in better agreement with the experimental data, the maximum lift is overpredicted.

The surface pressure and the skin friction for the flow at 16° angle of attack are shown in Figure 2.8. It is clearly seen that the DES97 simulation predicts the flow to separate much closer to the leading edge and thereby to give a much lower suction peak the the DDES. The premature separation is caused by MSD.

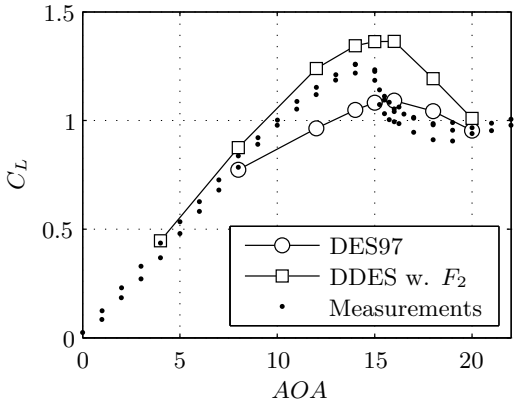


Figure 2.7: Lift coefficient with and without shielding of the RANS layer. Measurements are from the LM measurements described in [2, 21–24]. All simulations are with 2% resolved turbulence intensity. No model for the laminar-turbulent transition of the boundary layer is employed.

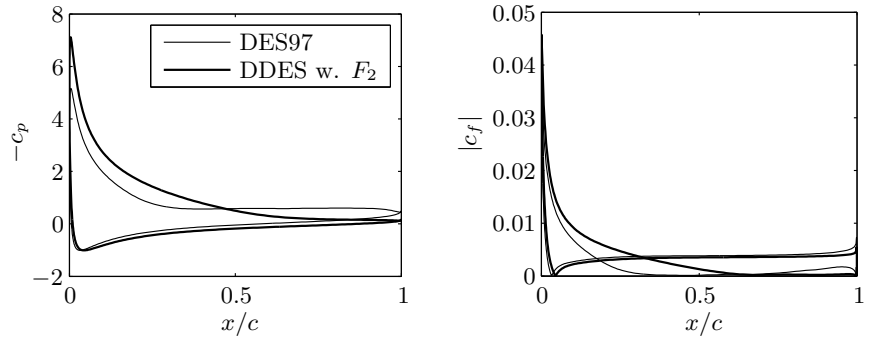


Figure 2.8: Average surface pressure and skin friction at 16° incidence and with 2% resolved turbulence intensity. No model for the laminar-turbulent transition of the boundary layer is employed.

2.5 Laminar-Turbulent Transition Model

In Paper I the local correlation-based transition model (LCTM or $\gamma - Re_\theta$ model) by Menter and co-workers [8, 9] is used to model the laminar-turbulent transition of the boundary layer. In the recent review of transition models by Cheng *et al.* [42] the LCTM was found to be one of the most promising transition models.

Although all presented results in Paper I are based on simulations with the transition model several simulations have been run without a model for

the laminar-turbulent transition. With no transition model the boundary layer is assumed to be turbulent everywhere.

Figure 2.9 shows the viscosity profiles at three chordwise positions in simulations of the airfoil flow with an angle of attack of 16°. It shows the differences between the simulations with and without a transition model. In both cases the freestream values of the eddy viscosity is $0.7\nu_{mol}$ with $Tu = 0.1\%$ and $14\nu_{mol}$ with $Tu = 2\%$. The largest difference between the transition and fully turbulent simulations is at the chordwise position $x/c = 0.1$. As shown in Chapter 4 and Paper I the transition model gives the same viscosity profile at this location independent on the freestream eddy viscosity for turbulence intensities up to 0.5%. Simulations with turbulence intensities of 2% and 4% give an extra bump on the profiles at $n^+ \approx 700$, which also can be seen in Figure 2.9. The overall picture is that there is a moderate influence in the outer part of the boundary layer from the freestream values but only for the highest freestream values of the eddy viscosity. For the fully turbulent simulations the values in the outer part of the boundary layer depend directly on the freestream eddy viscosity for all turbulence intensities. As seen in the figure there is a large build up of eddy viscosity with the high turbulence intensity.

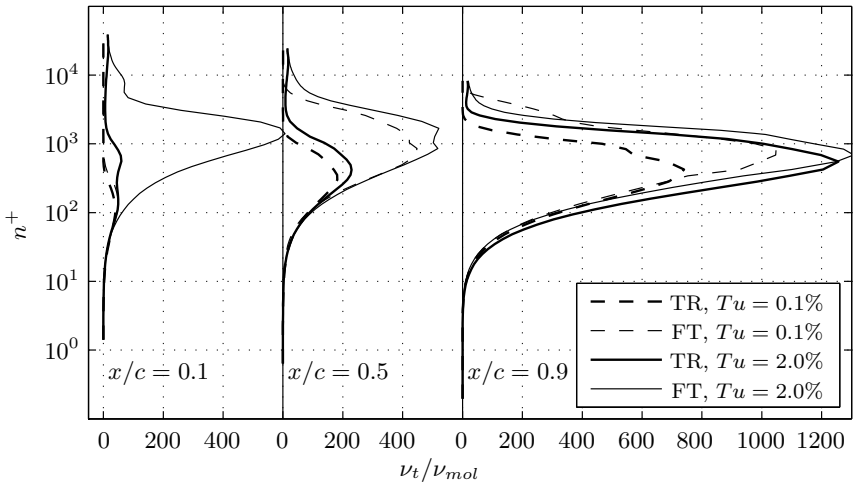


Figure 2.9: Average profiles of eddy viscosity at three chordwise positions. FT denotes simulations without a model for the laminar-turbulent transition of the boundary layer and TR denotes simulations with the local correlation-based transition model of Menter and co-workers [8, 9]. n^+ is the distance from the wall measured in viscous units.

At the position $x/c = 0.5$ the fully turbulent simulations have both started to separate and this gives high eddy viscosities. Finally, at $x/c = 0.9$

all four simulations have separated and they all show high levels of eddy viscosity in the separation bubble.

In an effort to analyze the effect of the resolved inflow turbulence we found that the velocities in the inner part of the boundary layer increased with increasing turbulence intensity in fully turbulent simulations. The effect was, however, found to depend only on the freestream values of the eddy viscosity and not the resolved turbulence itself. The cause of the higher velocity is the large values of eddy viscosity in the outer part of the boundary layer causing a higher shear. Since the eddy viscosity is not increased in the bottom of the boundary layer the result is a higher velocity close to the wall.

When changing the resolved turbulence intensity also the parameters of the turbulence model should change. Therefore, the effect of the resolved turbulence cannot be investigated independently from the effect of freestream eddy viscosity in fully turbulent simulations.

By applying the transition model, the influence of the freestream eddy viscosity on the boundary layer values is minimized. This allows an investigation of the effect of the resolved turbulence itself with only low influence of the subgrid model parameters at least for turbulence intensities up to 0.5%. Therefore, only results from simulations with the transition model are presented in Paper I and Chapter 4.

Chapter 3

Spatial Decay of Homogenous Turbulence

One of the classic approaches for testing turbulence resolving codes or methods is to simulate decay of homogenous, isotropic turbulence. Excellent experimental data was reported by Comte-Bellot and Corrsin [43, 44] who measured the spectra at three positions in a wind tunnel. Thereby, the evolution in the spectral characteristics was observed. By use of Taylor's hypothesis of a frozen turbulence field being convected with the mean flow velocity the evolution in space can be thought of as an evolution in time.

This is the reasoning behind the typical approach of modeling decaying turbulence employed in [13, 45]. The decay is modeled as a decay in time rather than in space. A computational domain is initialized to a turbulence field without a mean convective velocity. During the transient computation the turbulence decays and the data are sampled to be compared to the experiment.

In the airfoil simulation presented in Chapter 4 and Paper I the turbulence is imposed at the inlet and then convected with the mean velocity to the airfoil. For this application it is more relevant to study the spatial decay. The main difference between the two approaches is the computational cost for the two types of simulations. For the same resolution, the spatial decay is more expensive to study than the temporal decay.

Below, the reasons for the difference in cost of simulating temporal vs. spatial decay are given. Next, some results from the study of spatial decay in Paper I are presented.

3.1 Cost of Simulating Spatial vs. Temporal Decay

There are three reasons for the higher cost of simulating the temporal decay then for simulating spatial decay.

First and most important, the size of the computational domain must be larger for the study of spatial decay. When the temporal decay is studied the computational domain is typically cubic with periodic boundary conditions on all six sides. For the spatial decay the computational domain is typically much longer in the streamwise direction than in the transverse directions. For the same transverse dimensions the number of computational cells is much higher in a simulation of spatial decay.

Second, the initialization time is longer. With the temporal simulation the domain is initialized to some known turbulence field and then the simulation is run for a short time to let the turbulence field adapt to the mesh and the numerical method. With the spatial decay the turbulence is imposed at the inlet and then convected with the convection velocity to fill the domain. The statistics of the turbulence have not converged until the entire domain is filled with turbulence.

Finally, since the turbulence field is convected through the computational cells in the domain, the fine structures are smeared out more than in a simulation without a mean convective velocity. Spalart [15] argues that eddies spanning less than five cells cannot be resolved. In Paper II it is shown that eddies spanning up to about 10 cells undergo a high loss of energy as they are being convected. Thereby, the obtained resolution is coarser in a simulation of spatial decay than in temporal decay for the same mesh resolution. In turn, it is required to use a finer spatial resolution in a simulation of the spatial decay to obtain the same resolved scales as in a simulation of the temporal decay.

3.2 Results

To validate the computational method it will be demonstrated that the spatial decay of turbulence can be simulated. Two characteristics of the turbulence will shown to agree with experiments. One is that the turbulence decays with the correct rate as measured in the experiment by Comte-Bellot and Corrsin [43]. The other is that the slope of the spectra at the highest resolvable frequencies is $-5/3$ in a double log representation.

The computational domain consists of $1280 \times 64 \times 64$ cubic cells. At the inlet a turbulence field is imposed by prescribing the velocities to a constant mean velocity superposed by the turbulence field. When synthetic turbulence is imposed the corresponding fields of the subgrid parameters, i.e. k and ω, are not known. In this case constant fields of k and ω are prescribed.

The simulation technique is DES with the $k-\omega$ SST subgrid model, but since there are no walls, the DES model is run in LES mode throughout the domain. The simulated rate of decay is shown in Figure 3.1 with the thin full line. By comparison to the empirical slope it is seen that the decay is too slow for just over the first half of a second. This is due to a poor choice of subgrid parameters imposed at the inlet. The eddy viscosity is too low which causes the rate of decay to be too low. However, it is seen that after about 0.7 s the simulated rate of decay matches the empirical rate, which gives the correct slope of the curve. The cause for this is that over the first third of the domain the subgrid parameters have settled to balance the resolved turbulence. This is quite encouraging, but the stretch needed for the subgrid parameters to adjust to the resolved stresses makes it worth the effort to run a precursor simulation to achieve the correct subgrid parameters.

Next, it is verified that a turbulence field from a precursor simulation will need a shorter initialization stretch. Velocities and subgrid parameters are sampled from the simulation with the synthetic inflow turbulence and guessed inlet values of the subgrid parameters. The sample plane is orthogonal to the mean flow direction and it is located in the midpoint between the inlet and the outlet. In each time step the velocities and subgrid parameters are sampled in all cells crossing the sample plane. This gives a time series which is imposed at the inlet of a new domain.

The results are plotted in Figure 3.1 with the thick full line. The inlet velocities and subgrid parameters are sampled a distance x_0 from the inlet in the simulation with synthetic turbulence, and therefore the curve for the simulation with precursor turbulence has been moved x_0/U to the right. It is seen that with the velocities and subgrid parameters from the precursor simulation the turbulence shows the correct rate of decay from the inlet.

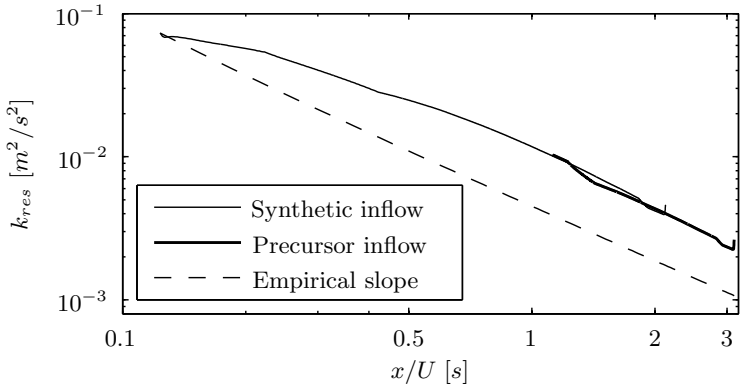


Figure 3.1: Decay of resolved turbulent kinetic energy.

Figure 3.2 shows the one-dimensional power spectra obtained from time series sampled at some position in the two simulations with synthetic and precursor inflow turbulence, respectively. For comparison the spectrum for the synthetic turbulence field is included in the figure.

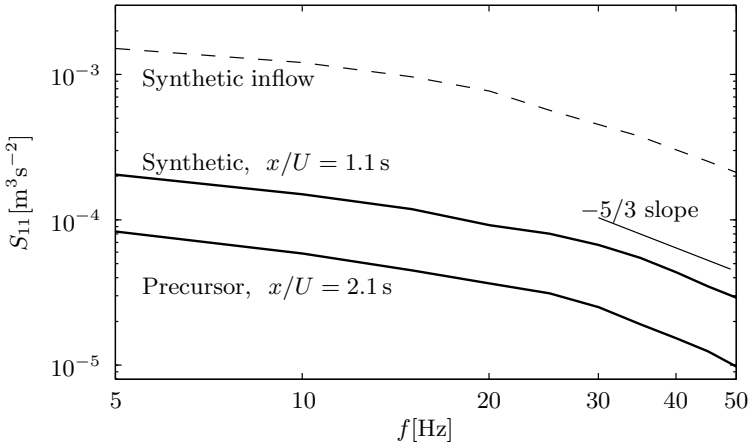


Figure 3.2: Power spectra of turbulence sampled at two positions in the streamwise direction in the simulations using synthetic and precursor turbulence. For comparison the power spectrum of the synthetic field is shown with the dashed line.

Only the resolvable range of frequencies is shown. It is observed that at the highest resolvable frequencies the slope is $-5/3$ as it should be in the inertial subrange. This leads to the conclusion that the cut-off frequency is in the inertial subrange as desired in LES. The range of resolvable frequencies is quite narrow since the turbulence is isotropic and only 64 cells are used in the transverse directions.

In conclusion the numerical method and the subgrid model give the correct behavior of the spatially decaying turbulence. By running the turbulence through a precursor simulation it is adopted to the mesh and the computational method and appropriate values of the subgrid values are determined.

Chapter 4

Detached-Eddy Simulation of the Airfoil in Resolved Inflow Turbulence

The main issue in the present thesis is the effect of resolved inflow turbulence in detached-eddy simulations of an airfoil. It has been investigated in Paper I and here the main conclusions are summarized. The details of the numerical methods and the computational setup is given in the paper along with a full discussion of the results.

The numerical setup is sketched in Figure 2.1. The airfoil is the symmetric NACA 0015 airfoil, which is chosen because of availability of experimental data. The Reynolds number in the simulations is 1.6 million which is in the lower end of practical Reynolds numbers for modern wind turbines. The mesh is structured and consists of about 21 million cells.

4.1 Results

The main results from the simulations are presented below. In the Figures 4.1 and 4.2 the obtained lift and drag coefficients are plotted and compared to two sets of experimental data. With no turbulence grid in the LM Wind Tunnel the turbulence intensity is 0.4% [23]. With the two grids the turbulence intensities have been measured to 1.2% and 1.5%. In the experiment by Mish [46, 47] the turbulence intensity is 3.9%. The turbulence length scales in the experiments and simulations are described and discussed in Paper I.

The computed lift agrees quite well with the data by Mish. It falls

somewhere in between the data from the LM tunnel without turbulence grid and the data with turbulence grids. In the experiment by Mish the drag was not measured. The computed drag compares well with the LM data with grid 2 for incidences up to 18°. At higher angles of attack the drag is overpredicted.

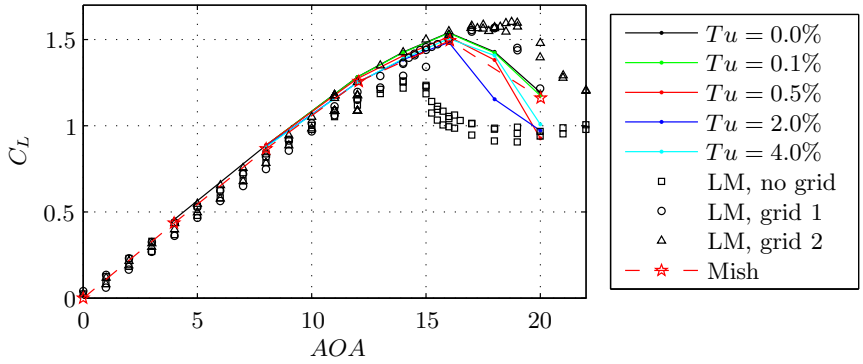


Figure 4.1: Lift coefficient from simulations and measurements. The experimental data from the LM Wind Tunnel is described in [2, 21–24]. The data by Mish is described in [46, 47].

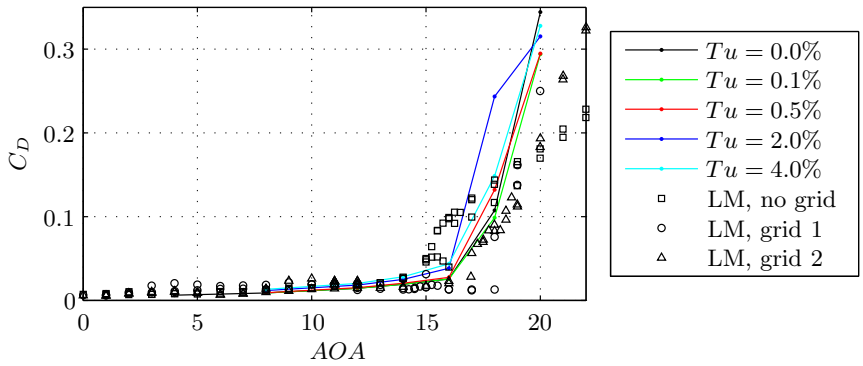


Figure 4.2: Drag coefficient from simulations and measurements. The experimental data from the LM Wind Tunnel is described in [2, 21–24].

At angles of attack up to stall the differences in computed lift and drag are small. At 18° the simulation with $Tu = 2\%$ gives spuriously low lift compared to the simulations with $Tu = 0.5\%$ and $Tu = 4\%$. At 20° the simulations with $Tu = 0.5\% - 4\%$ agree quite well again. In general the differences are low up to 20°. Compared to the difference in lift curves from

the LM experiments the simulated effect of resolved inflow turbulence is discouraging.

Next, it is shown that the resolved turbulence does penetrate the RANS region. In Paper I the velocity is sampled in four points close to the leading edge of the airfoil. The positions of the four points are shown in Figure 4.3 which also shows the power spectra of the sampled time series. The indicated cut-off frequency is based on the mesh resolution in the Cartesian grid upstream of the airfoil, and marks the highest frequencies that can be convected through the upstream convection region (c.f. Figure 2.1).

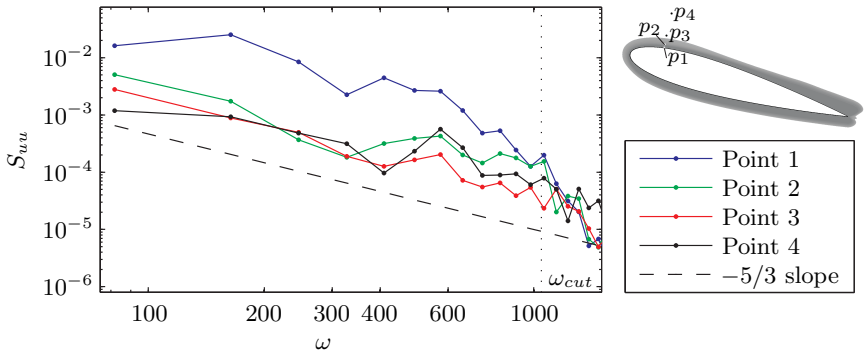


Figure 4.3: Power spectra of velocity time series sampled in and near the boundary layer. The RANS region is grey shaded.

The shown spectra are quite noisy because only relative short time series have been sampled and they have not been averaged in the spanwise direction. Nevertheless, several tendencies can be identified. Most importantly, the spectra are seen to contain energy on all resolvable scales for all four points. Closest to the wall there is a build-up of energy, especially on the large scales. The energy drops quite fast at higher frequencies than shown in the plot. Due to the limited resolution in the convection region upstream of the airfoil only eddies larger than about 5% of the chord impinge on the airfoil.

Figure 4.4 shows the averaged profiles of velocity and viscosity. The largest differences between the various simulations are in the outer part of the boundary layer. Here the eddy viscosity is increased by the freestream values as discussed in Section 2.5. In the inner part the velocity is increased with up to 6%. The increase is not systematic, however, since the two simulations with the highest turbulence intensities do not give the highest increase in velocity. In these two simulations the freestream values of the eddy viscosity cause the eddy viscosity profiles to differ strongly from the simulation without resolved inflow turbulence.

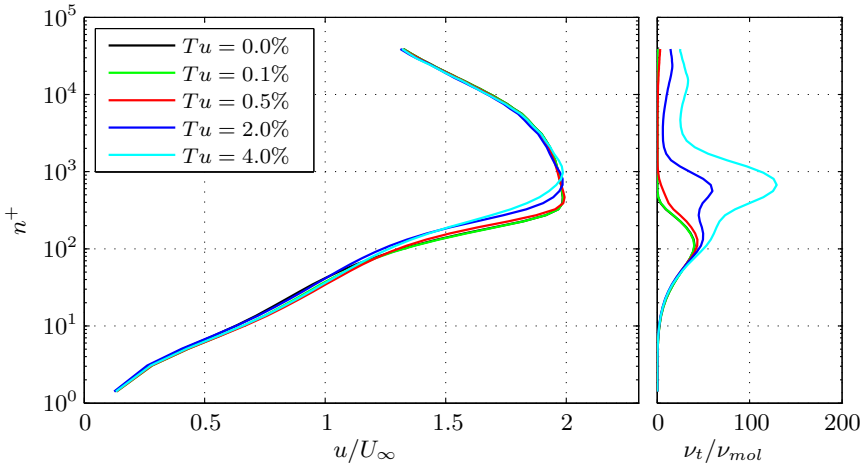


Figure 4.4: Averaged profiles of velocity and viscosity sampled on the suction side at $x/c = 0.1$ and 16° incidence.

At positions further downstream the velocity increases do not persist. Thereby, the resistance against separation is not increased as in the experiments.

4.2 Discussion and Conclusions

From the results presented above it is concluded that the effect of freestream turbulence was not captured by the presented DES. Contrary to the experimental data a small decrease in maximum lift was found when increasing the intensity of the resolved turbulence. The decrease in maximum lift can be explained by the location of the transition point, which moves forward as the subgrid kinetic energy is increased.

In nature the eddies from the freestream turbulence are convected into the boundary layer where they help to mix high velocity fluid into the lower part of the boundary layer. The increased momentum gives a higher resistance against the adverse pressure gradient and thereby against separation. In the simulations all *resolvable* scales of eddies are present in the boundary layer, but due to the limited grid resolution the diameter of the smallest eddies are comparable to the boundary layer thickness. Nevertheless, slightly higher velocities have been observed in the lower part of boundary layer close to the leading edge. The velocity increase does not persist, however, and the effect on the maximum lift is absent.

To capture the effect of inflow turbulence it might be necessary to resolve the turbulence in the boundary layer in LES or even DNS of the

airfoil. Sadly, these techniques are not applicable for wind turbine blades at industrial Reynolds numbers.

Chapter 5

Related Work

The results and conclusions from Paper I were discovered late in the PhD-project period. We had expected that the DES could capture (some of) the effect of the inflow turbulence, so we had already started to prepare for a study of the influence of the turbulence characteristics. The work was carried out on two fronts. First, we have created a method for imposing the resolved turbulence inside the domain of CFD simulations, which can be used to reduce the computational cost of the simulations. It is presented in Paper II. Second, two methods for generating the synthetic turbulence have been developed. They are presented in Paper III and IV.

Although, the methods will not be applied in this study of the effect of inflow turbulence the effort is not wasted. They can be applied to a number of other applications which will be highlighted below. The three methods are described and summarized in each of the following sections.

5.1 Imposing Resolved Turbulence Inside Domains

The computational cost of the DES presented in Chapter 4 is very high. To reduce this cost a method for imposing the resolved turbulence immediately upstream of the airfoil is developed. The method is described below along with results to verify the approach. Finally, two examples of applications are given.

5.1.1 The Basic Idea

A sketch of the computational domain used in the DES of the airfoil is shown in Figure 2.1. The computational cost for the presented DES with resolved inflow turbulence is far higher than for a standard DES without inflow turbulence. The reason is that when the turbulence field is imposed

on the inlet boundary the mesh resolution in the convection region must be fine enough to resolve the turbulence as it is convected from the inlet to the airfoil. 6.3 million cells are used in the convection region of the computational mesh in the DES of the airfoil flow, which corresponds to one third of the total number of cells.

The basic idea motivating the presented methods is to impose the resolved turbulence immediately upstream of the leading edge of the airfoil as indicated in Figure 5.1. Now, the inflow turbulence should only be resolved while it is convected the short distance from the actuator plane to the leading edge of the airfoil. A coarse mesh resolution can be used upstream of the actuator plane because now only the steady mean flow has to be resolved.

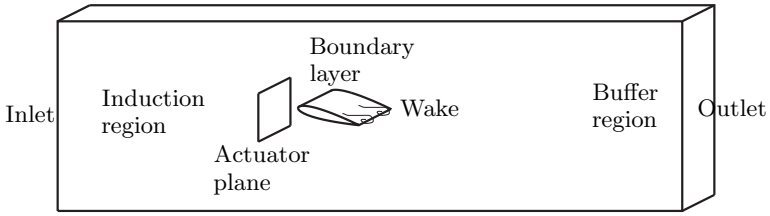


Figure 5.1: Sketch of computational domain used in airfoil flow.

Then, the problem is to impose resolved turbulence inside the computational domain in a CFD simulation.

5.1.2 Two Approaches

Two different approaches for imposing the resolved turbulence are investigated. Both are aiming to impose *a given* turbulence field and not just some realization of a turbulence field with specified statistics.

The first approach is to modify the source term of the discrete Navier-Stokes equations. By specifying a given change in the source term it is possible to prescribe a desired change of the velocity vector in each computational cell crossed by the actuator plane. The second approach is to use the actuator disc model derived and implemented by Réthoré [48] to subject the flow field in each cell to some resulting force. By choosing the appropriate force vector, a desired velocity can be obtained some distance downstream of the actuator.

Both approaches are described in detail in Paper II where the modification to the source term and the required forces are derived.

5.1.3 Results

The computational domain used to verify the approaches are shown in Figure 5.2. As seen from the figure the flow is steady and laminar upstream of

the actuator plane where the turbulence is imposed.

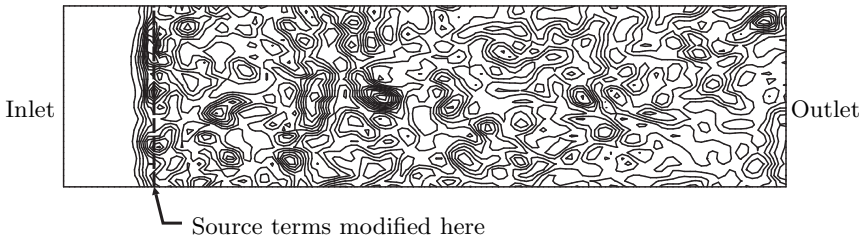


Figure 5.2: Instantaneous iso-vorticity contours. Precursor turbulence is imposed by modifying the source terms.

Figures 5.3 and 5.4 compare the resolved turbulence sampled some distance downstream of the actuator to the target field. The imposed turbulence field has been sampled from a precursor simulation. The averaged power spectra plotted in the figures show that both methods can recreate precursor turbulence with good accuracy. The indicated cut-off frequency is based on the mesh resolution. The method of the actuator disc gives too low energy on the highest resolvable frequencies especially for the streamwise component. Also, the energy on the largest scales is slightly too low. In contrast, when the turbulence is imposed by the method of modifying the source terms the sampled field compares perfectly with the target for all resolvable frequencies.

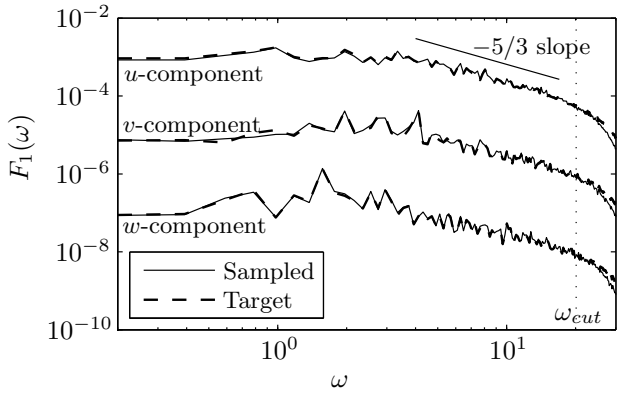


Figure 5.3: Precursor turbulence imposed by modifying the source terms. Comparison of power spectra of imposed and sampled time series. The u and w components have been moved two decades up and down, respectively.

The good results described above were obtained from imposing precursor turbulence. In this case the imposed turbulence field satisfies both the

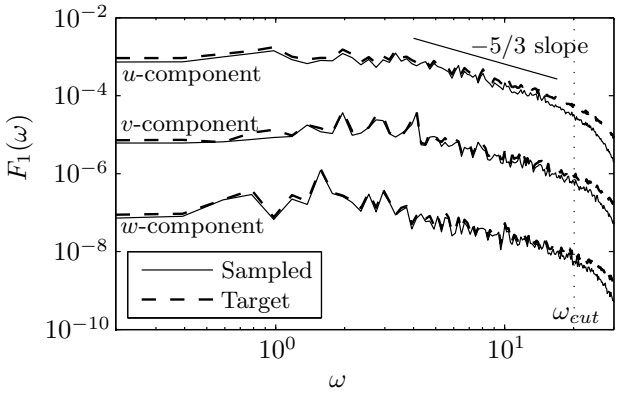


Figure 5.4: Precursor turbulence imposed by an actuator. Comparison of power spectra of imposed and sampled time series. The u and w components have been moved two decades up and down, respectively.

continuity condition and the discrete Navier-Stokes equations. Further, it contains only low energy on frequencies higher than what can be resolved by the mesh and the numerical method. These characteristics have been found to be important for the ability of both methods to recreate the imposed turbulence field.

In Figure 5.5 the power spectra are shown for an attempt to impose synthetic turbulence. The turbulence has been generated by the method of

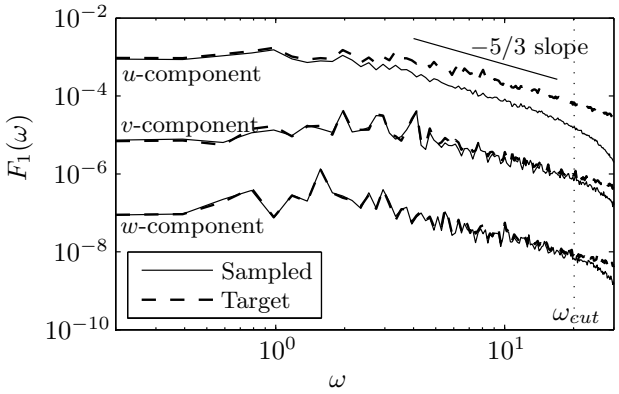


Figure 5.5: Synthetic turbulence imposed by modification of source terms. Comparison of power spectra of imposed and sampled time series. The u and w components have been moved two decades up and down, respectively.

Mann [29] as described in Section 2.2 and the technical report [1]. As seen from the figure there is a significant loss of energy for the largest decade of resolvable frequencies for the streamwise component. For the two other directions the loss is quite low. Turbulence imposed by the actuator disc shows the same tendencies, but the results are even worse.

In many applications of a precursor simulation the purpose is to determine the inlet velocity profile and let the turbulence form from the basis of small disturbances. This type of precursor simulation is very computationally expensive because a long stretch is needed or the simulated time should be long. This type of precursor was e.g. employed by Bechmann [45]. In contrast, the precursor needed to prepare the synthetic turbulence field for being imposed by the methods applied here can be very short.

In Figure 5.6 the required length of the precursor simulation is investigated. A precursor simulation is run where the synthetic turbulence field is imposed at the inlet. In the five first layers of cells after the inlet the turbulence field is sampled and these five fields are then imposed by the method of the modified source terms. The turbulence fields obtained from imposing these five fields are compared to the original synthetic field in the figure. Only the streamwise component is plotted, as this was found to be the troublesome one. It can be observed that perfect results are obtained by

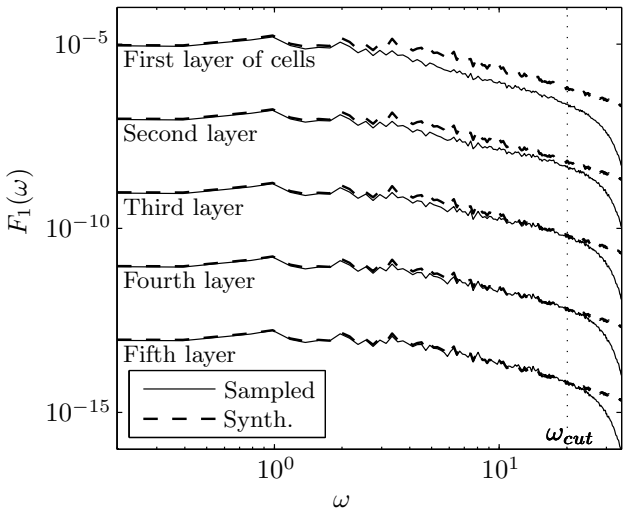


Figure 5.6: Precursor turbulence imposed by a modification of the source terms. Power spectra from simulations with turbulence from precursors of different lengths are compared to the same synthetic field. The spectra have been separated vertically by two decades for clarity.

running the synthetic turbulence through a precursor where the turbulence is sampled again after five cells in the streamwise direction. This allows a very cheap precursor to be used, as the total number of cells needed in the streamwise direction is around $20 - 50$.

Because the results with the method of the actuator disc are not as good as the method with the modified source terms, the latter is recommended. Further, it is recommended to run the synthetic turbulence field through a short precursor simulation before it is imposed.

To give an impression of the accuracy of the methods, the streamwise component of a sampled field is compared to the target time series in Figure 5.7. The sampled velocities compare very well to the target. The error in obtained standard deviation is less than 1% and the average correlation coefficient is about 0.97.

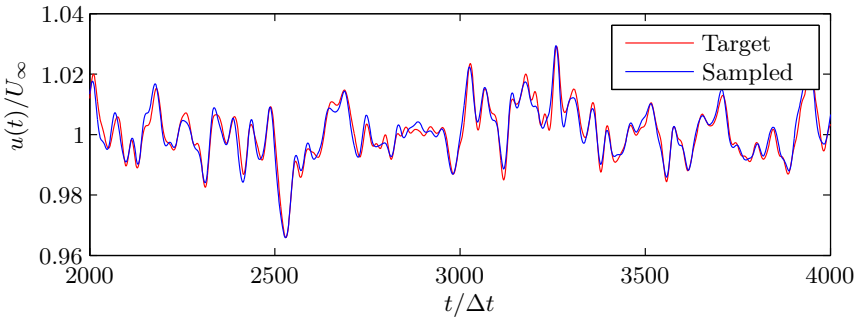


Figure 5.7: Time series of target and sampled streamwise velocity. Precursor turbulence is imposed by the modification of the source terms.

5.1.4 Possible Applications

In general the proposed method can be applied to all simulations where resolved turbulence is imposed upstream of a region or structure of interest. A few examples are given below.

The intended application of the method described above was to reduce the computational cost of the airfoil simulations by reducing the required number of cells upstream of the airfoil. Figure 5.8 shows the instantaneous magnitude of the vorticity vector in an airfoil simulation with the turbulence imposed by the actuator.

The number of cells has been reduced by 45% as a direct consequence of that it is not necessary to resolve the turbulence upstream of the actuator. This reduction has been obtained without decreasing the mesh resolution close to the airfoil.

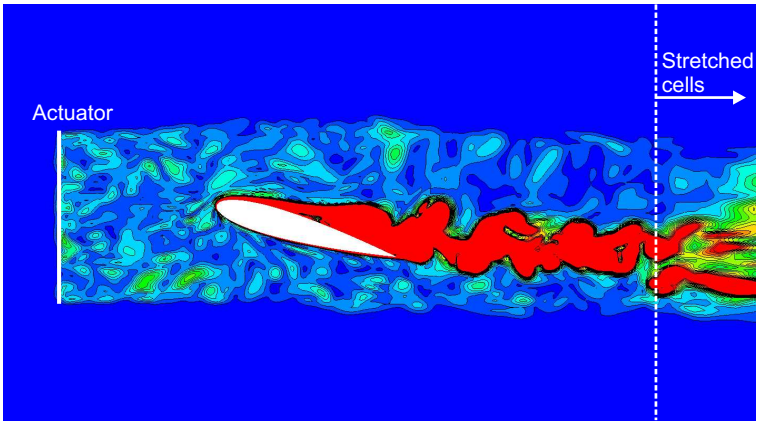


Figure 5.8: Vorticity plot from a DES of an airfoil. From [49].

The results from Paper I showed that the performed DES could not capture the effect of the resolved inflow turbulence. However, if the effect is studied with LES at lower Reynolds numbers the presented method of imposing the resolved inflow turbulence would severely reduce the computational cost.

Another application is the study of wakes of rotors in turbulent flow as performed by Troldborg [50] and Mikkelsen [51].

5.2 Three Dimensional Turbulence Modeling in Shear Flow

Paper III gives a semi-analytical model for the auto- and cross-spectra of turbulence in shear flow. The method is based on the rapid distortion theory by Townsend [32]. Below, the background for the model is given along with a description of the findings. Finally, some examples of applications are presented.

5.2.1 Background

A method for determining extreme loads on wind turbines is prescribed by the design requirements IEC61400-1 Ed. 3 and its first amendment. The method includes to run six or 15 aeroelastic simulations of 10 minutes duration for each wind velocity and then to extrapolate the obtained loads to characteristic loads with a 50 year return period. The ratio of simulated time to the desired return period is therefore very high, and the tails of the probability density functions are poorly represented.

Alternatively, the Russian roulette and splitting technique described by Pradtlwarter and Schüeller [52] can be used to predict the low probabilities of failure. Here, a better determination of the tails of the probability distributions can be obtained, and consequently, the characteristic loads can be estimated with better accuracy. With this technique, the synthetic turbulence should be determined "on-the-fly" in the aeroelastic simulations. With the Mann [29] or Sandia methods [53] the entire turbulence field is generated prior to the simulating the response. In contrast, ARMA or state space modeling by Sictani *et al.* [54] can be used to generate turbulence "on-the-fly" in load calculations.

The auto- and cross-spectral densities are needed as input to the ARMA or state space model, and in Paper III a semi-analytical model for the auto- and cross-spectra of turbulence is derived.

5.2.2 Description of the model

The work is based on the rapid distortion theory by Townsend [32], which is also the foundation for the method of Mann. The cross-spectral density function can be written as a double integral of the shear spectral tensor. In Paper III an approximation of the integrand is introduced in the form of a truncated Fourier series expansion of second order. From this approximation it is possible to evaluate the inner integral analytically, which leaves a single integral for numerical integration. Thereby, the calculation time can be reduced severely, as about 85 % of the function evaluations of parameters entering the integrand can be omitted.

5.2.3 Applications

Besides the application highlighted in Section 5.2.1 above, the presented method can be used to generate the auto- and cross-spectra for the Sandia method [53] of generating synthetic turbulence. By this approach the generated turbulence would match the properties of the turbulence generated by Mann's method [29] which by Saranyasoontorn *et al.* [55] was found to give a more realistic coherence of the turbulence.

With the Sandia model the turbulence field consists of time series of turbulence from a number of points, and these points can be placed freely. With the Mann method these points must be placed in a Cartesian grid due to the application of three-dimensional FFT algorithm. For load calculations on structures with low solidity (e.g. suspension or cable-stayed bridges) it is an advantage that the points can be placed only where the turbulence is needed and not in the free air between the structural elements.

Further, the semi-analytical model for the auto- and cross-spectra presented here can be used with the method for generating synthetic turbulence described in Section 5.3 and Paper IV. The semi-analytical model gives the

auto- and cross-spectral densities which in turn can be transformed to the auto- and cross-correlation functions by application of the Wiener-Khinchin relation. Now, a model for the correlation in a sheared turbulence field is available. In the method described below the physics of the generated turbulence is defined by the auto- and cross-correlation. Hence, by using the semi-analytical model for sheared turbulence it is possible to generate anisotropic turbulence with the method described below.

5.3 Synthetic Turbulence in Arbitrary Domains

The present section describes a method for generating synthetic turbulence in arbitrary domains. At first the intended application is described. It is well known that turbulence sampled by an anemometer mounted on the tip of a rotating wind turbine blade will show different characteristics than turbulence sampled by a stationary anemometer. The effect was studied by Connell [56]. Figure 5.9 shows the difference in power spectra obtained depending on whether the sampling point is stationary or if it rotates as on a rotor.

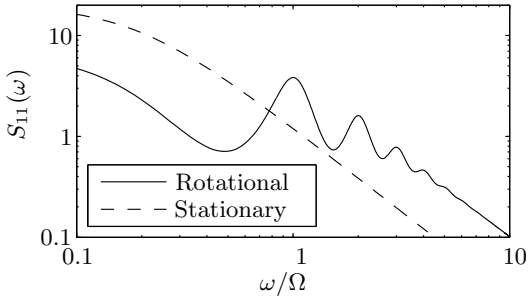


Figure 5.9: Power spectral densities for turbulence sampled by two different anemometers. One is stationary, while the other is mounted on a rotating blade of a wind turbine.

At integer multiples of the rotational frequency there is a peak in the spectrum, and at high frequencies there is a general increase in the energy. The intensity of the peaks and the amplification at high frequencies depend on the radius of motion, the turbulence length scale and on the mean wind velocity.

A section of a rotating blade of a wind turbine encounters turbulence with the spectral characteristics as shown in the figure. Therefore, it was decided to investigate the influence of this turbulence on the flow over an airfoil by CFD simulations. Below, a description is given of the effort to generate a synthetic turbulence field that could be used as inflow turbu-

lence for such a simulation. The generated method is described in detail in Paper IV.

5.3.1 Motivation

Figure 5.10 shows the turbulence encountered by a rotating section of a wind turbine blade. This turbulence field should be generated synthetically and a very high resolution is desired in the covered three-dimensional spiral. In turn, the turbulence in points not covered by the spiral is irrelevant.

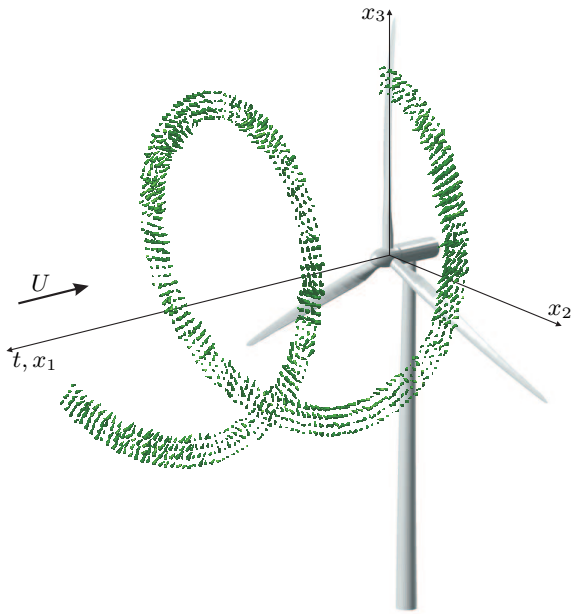


Figure 5.10: Realization of turbulence field generated by the proposed method. The shown domain contains only 4×4 discrete points that move in time.

The two most common techniques for load calculations of wind turbines are the method of Mann [29, 30] and the Sandia method by Veers [53]. They can both generate turbulence fields covering the entire rotor, and in principle it would be possible to obtain the shown spiral by interpolating in a full field determined by one of these methods. The problem, however, is that in order to get a fine resolution in the desired region the required number of points becomes prohibitively high. The limitation is in a very high demand for computer memory as shown in Example 2 in Section 5.3.3.

The Mann method is the more efficient of the two, but it is based on three-dimensional FFT. Thereby, the points in the discrete domain must be placed in a equidistant Cartesian grid. Consequently, the same resolution

must be used everywhere in the rotor plane. For the present application a very high resolution is needed in the spiral shown in Figure 5.10, while the rest of the rotor plane is not needed. Thereby, a very large percentage of the generated turbulence is dispensable and a pure waste of computational effort.

To overcome this problem the domain is resolved into a number of points that are allowed to move in time. Thereby, the domain in Figure 5.10 contains only 4×4 points that move with the rotor. This gives a very large saving in computational effort for the present application as only the desired part of the turbulence field is generated.

5.3.2 Description of the Method

In essence the proposed method is similar to the Sandia method by Veers [53] but with one important difference. In the Sandia method the spectral matrix is determined analytically based on an auto-spectral density and a coherence function. Next, the spectral matrix is factored and multiplied by complex random numbers. Finally, the result is Fourier transformed to give a realization of the turbulence field. The method proposed here determines the spectral matrix based on correlations, but the rest of the algorithm is the same as in the Sandia method.

The spectral matrix is determined from the following steps. First, the domain is resolved into a number of discrete points in each time step. The second step is to compute the correlation between two velocity vectors separated by the distance traveled by each point since the initial state. Finally, the spectral matrix is determined from the correlations by use of the Wiener-Khinchin relation. By following this approach the auto- and cross-spectral densities are determined for the target turbulence field sampled in the discrete points. The method is derived and described in detail in Paper IV.

The correlation functions employed in the paper are valid only for isotropic turbulence. However, atmospheric turbulence is not isotropic and to generate anisotropic turbulence with the presented method the auto- and cross-correlation functions for sheared turbulence are needed. From the method of Paper III and Section 5.2 the auto- and cross-spectral densities of sheared turbulence can obtained, and by use of the Wiener-Khinchin relation these can be transformed to correlation functions. Thereby, it is possible to generate synthetic turbulence which matches the statistics of turbulence as prescribed by the IEC61400-1 code for load calculations of wind turbines.

5.3.3 Examples

Two examples are given below. The first serves to validate the method while the second should demonstrate the large difference in required memory

compared to the Mann method.

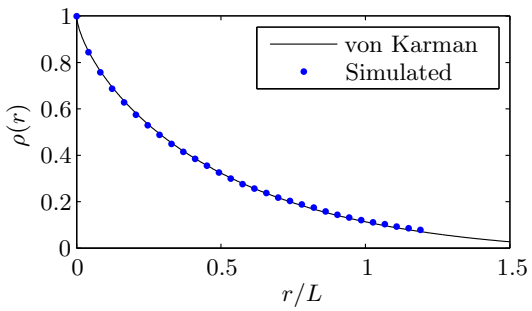


Figure 5.11: Correlation of streamwise velocity along the rotating blade of Example 1. The radial position on the blade is denoted r and L is the turbulence length scale.

Example 1: Turbulence is generated along a single rotating wind turbine blade. The length of the blade is 50 m and the turbulence length scale is 42 m. To validate the method the correlation along the blade is computed and compared to the theoretical function by von Karman [33]. The resulting curves are shown in Figure 5.11 and the agreement is seen to be very good. Further, the power spectral density is computed for the time series of streamwise velocity sampled at the tip of the blade. In Figure 5.12 it is compared to the theoretic spectrum by Connell [56]. The computed spectrum has been averaged over 100 realizations and it compares well with the target spectrum. For comparison the target power spectrum of velocities sampled in a stationary point by von Karman [33] is included in the plot.

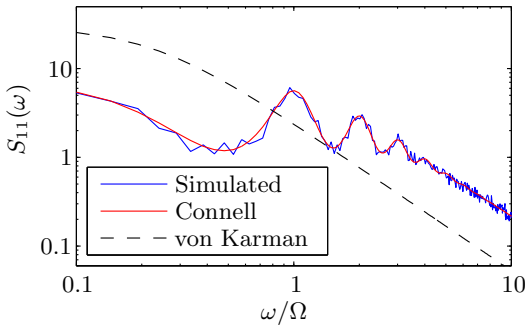


Figure 5.12: Auto-spectral density of turbulence sampled at the tip of the rotating blade.

Example 2: Turbulence is generated in a domain like the one shown

in Figure 5.10 to demonstrate the large difference in required computer memory for the present approach and the method of Mann. The domain contains 8×8 moving points and 512 time steps are simulated in a single revolution of the rotor. The domain covers an area of $1 \times 1\,\mathrm{m}^2$ in each time step and the radius of the covered spiral is 40 m. To generate this field a system matrix of 72 MB should be allocated. To generate a synthetic turbulence field with the method of Mann with the same spatial resolution it would be necessary to allocate 4.3 GB. This large number arises from the need to use the same fine spatial resolution over the entire rotor area.

Chapter 6

Conclusions

The effect of resolving inflow turbulence in detached-eddy simulations of an airfoil is investigated. The method of Mann is implemented and used for generation of synthetic turbulence. The synthetic field is run through a precursor simulation, which adapts the field to the mesh and the numerical method. In a study of the spatial decay of precursor turbulence it is shown to decay with the correct rate, and the cut-off frequency is shown to be in the inertial subrange.

The flow past an airfoil in a wind tunnel is simulated using the DES technique with the $k-\omega$ SST subgrid model. The resolved turbulence field is imposed at the inlet and convected with the mean wind velocity to the leading edge of the airfoil. The LES-branch of the DES model is active in the freestream region and here the SST subgrid model is shown to work similar to the Smagorinsky model. The RANS region close to the airfoil is shielded by the F_2 function of the SST model to prevent the switch from RANS to LES to occur in the boundary layer.

Experiments have shown two effects of an increase in the inflow turbulence intensity. One is to increase the maximum lift, which can be explained by a higher transport of high-momentum flow to the inner part of the boundary layer caused by the mixing from the turbulence. The other is to move the laminar-turbulent transition point upstream, which isolated seen will decrease the maximum lift. The latter effect is observed in the results from the DES simulations. The increase in momentum in the inner part of the boundary layer is not captured by the simulations. This is probably due to the relatively large length scale of the resolved turbulence that it is possible to resolve in the boundary layer. With the DES model, small scale turbulence cannot be resolved in the boundary layer, and these small scales might be responsible for the effect observed in experiments. This indicates

that it may be necessary to run LES or even DNS to capture the effect, and these methods are not applicable at industrial Reynolds numbers.

In the literature there are a number of applications of DES where the wake from one structure impinges on another structure. In these flows the DES model cannot be expected to predict the correct response of the boundary layer to the resolved inflow turbulence. The results will reflect the variations in inflow velocity, but not the physical interference from the resolved turbulence. Therefore, a word of caution is given for these applications of DES for bluff body flows with separation.

The computational cost is very high for the DES of the airfoil flow. To reduce the cost a method is developed to impose the resolved turbulence immediately upstream of the leading edge of the airfoil. Two methods are investigated. One is to modify the source terms of the Navier-Stokes equations and the other is to apply forces with an actuator disc. The best method is found to be the modification of the source terms, which is also the easiest method to implement. It is possible to recreate a turbulence field from a precursor simulation with good accuracy. If synthetic turbulence is imposed, it is recommended to run the turbulence through a short precursor simulation. The precursor is very cheap to preform, as the turbulence can be sampled only five cells from the inlet.

Two methods are developed for generation of synthetic turbulence with characteristics resembling the atmospheric turbulence. The first generates the cross-spectral density function of sheared turbulence. It is based on the rapid distortion theory, which also is the foundation for Manns method which is adopted by the IEC61400-1 code for load calculations of wind turbines. The cross-spectral density is defined from a double integral and by several manipulations with the integrand an analytical solution to the inner integral is given. This leaves a single dimension for numerical integration. The saving in computational cost is about a factor of 5 compared to the double integral.

Finally, a method for generation of synthetic turbulence in arbitrary domains is presented. Like in the two frequently used methods by Mann and Veers the domain is resolved into a number of points, where the velocity is determined. The big difference from these methods is that the points are allowed to move in time. Thereby, it is possible to resolve the turbulence encountered by section of a rotating blade with relative few points, which in turn saves computational cost. The method can generate synthetic turbulence with the correct spatial correlation and the correct spectral density function.

Bibliography

[1] Gilling, L. (2009) TuGen - manual and user's guide. DCE Technical Report No. 76, Aalborg University, Aalborg, Denmark.

[2] Gilling, L., Sørensen, N. N., and Davidson, L. (2009) The effect of resolved turbulence in detached-eddy simulations of an airfoil. *Submitted to Wind Energy*.

[3] Gilling, L. and Sørensen, N. N. (2009) Imposing resolved turbulence in CFD simulations. *Submitted to Wind Energy*.

[4] Gilling, L., Pedersen, B. J., and Nielsen, S. R. K. (2009) Three dimensional turbulence modeling in shear flow with application to wind turbines. *Submitted to Wind Energy*.

[5] Gilling, L., Nielsen, S. R. K., and Sørensen, N. N. (2009) Genaration of synthetic turbulence in arbitrary domains. *10th ICOSSAR 2009*, Osaka, Japan, September 13–17, CRC Press.

[6] McKeough, P. J. (1976) *Effects of Turbulence on Aerofoils at High Incidence*. Ph.D. thesis, University of London.

[7] Hoffmann, J. A. (1991) Effects of freestream turbulence of the performance characteristics of an airfoil. *AIAA Journal*, **29**, 1353–1354.

[8] Menter, F. R., Langtry, R. B., Likki, S. R., Suzen, Y. B., Huang, P. G., and Völker, S. (2006) A correlation-based transition model using local variables – part I: Model formulation. *Journal of Turbomachinery*, **128**, 413–422, DOI: 10.1115/1.2184352.

[9] Langtry, R. B., Menter, F. R., Likki, S. R., Suzen, Y. B., Huang, P. G., and Völker, S. (2006) A correlation-based transition model using local variables – part II: Test cases and industrial applications. *Journal of Turbomachinery*, **128**, 423–434, DOI: 10.1115/1.2184353.

[10] Spalart, P. R., Jou, W.-H., Strelets, M., and Allmaras, S. R. (1997) Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. Liu, C. and Liu, Z. (eds.), *1st AFOSR International Conference on DNS/LES*, Columbus OH, August, pp. 137–147, Greyden Press.

[11] Shur, M., Spalart, P. R., Strelets, M., and Travin, A. (1999) Detached-eddy simulation of an airfoil at high angle of attack. *Engineering Turbulence Modelling and Experiments*, **4**, 669–678.

[12] Strelets, M. (2001) Detached eddy simulation of massively separated flows. *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 8–11, AIAA Paper 2001-0879.

[13] Travin, A., Shur, M., Strelets, M., and Spalart, P. R. (2002) Physical and numerical upgrades in the detached-eddy simulation of complex turbulent flows. *Advandes in LES of Complex Flows*, pp. 239–254.

[14] Menter, F. R. and Kuntz, M. (2003) Adaptation of eddy-viscosity turbulence models to unsteady separated flow behind vehilces. *Proceeding Conference on the Aerodynamics of Heavy Vehicles: Trucks, Busses and Trains*.

[15] Spalart, P. R., Deck, S., Shur, M. L., Squires, K. D., Strelets, M. K., and Travin, A. (2006) A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*, **20**, 181–195.

[16] Bertagnolio, F., Sørensen, N. N., and Johansen, J. (2006) Profile catalogue for airfoil sections based on 3D computations. Tech. Rep. Risø-R-1581(EN), RISØ National Laboratory, Roskilde.

[17] Hedges, L. S., Travin, A. K., and Spalart, P. R. (2002) Detached-eddy simulations over a simplified landing gear. *Journal of Fluids Engineering*, **124**, 413–423, DOI: 10.1115/1.1471532.

[18] Cummings, R. M., Forsythe, J. R., Mortor, S. A., and Squires, K. D. (2003) Computational challenges in high angle of attack flow prediction. *Progress in Aerospace Sciences*, **39**, 369–384, DOI: 10.1016/S0376-0421(03)00041-1.

[19] Greschner, B., Thiele, F., Jacob, M. C., and Casalino, D. (2008) Prediction of sound generated by a rod-airfoil configuration using EASM DES and the generalised Lighthill/FW-H analogy. *Computers and Fluids*, **37**, 402–412, DOI: 10.1016/j.compfluid.2007.02.013.

[20] Paik, J., Sotiropoulos, F., and Porté-Agel, F. (2009) Detached eddy simulation of flow around two wall-mounted cubes in tandem. *Heat and Fluid Flow*, **30**, 286–305, DOI: 10.1016/j.ijheatfluidflow.2009.01.006.

[21] Bove, S. (2008) LSWT report of campaign NACA 0015_1. Tech. rep., LM Glasfiber, Lunderskov.

[22] Døssing, M. (2008) High frequency microphone measurements for transition detection on airfoils. Tech. Rep. Risø-R-1645(EN), RISØ National Laboratory - DTU, Roskilde.

[23] Bertagnolio, F. (2008) NACA0015 measurements in LM Wind Tunnel and turbulence generated noise. Tech. Rep. Risø-R-1657(EN), Risø DTU National Laboratory for Sustainable Energy, Roskilde.

[24] Bak, C., Andersen, P. B., Madsen, H. A., Gaunaa, M., Fuglsang, P., and Bove, S. (2008) Design and verification of airfoils resistant to surface contamination and turbulence intensity. *26th AIAA Applied Aerodynamics Conference. AIAA Paper 2008-7050*.

[25] Michelsen, J. A. (1992) Basis3D - a platform for development of multiblock PDE solvers. Tech. Rep. AFM 92-05, Technical University of Denmark, Lyngby.

[26] Michelsen, J. A. (1994) Block structured multigrid solution of 2D and 3D elliptic PDEs. Tech. Rep. AFM 94-06, Technical University of Denmark, Lyngby.

[27] Sørensen, N. N. (1995) *General Purpose Flow Solver Applied to Flow over Hills*. Ph.D. thesis, RISØ National Laboratory, Roskilde, risø-R-827(EN).

[28] Sørensen, N. N. (1998) HypGrid2D a 2-D mesh generator. Tech. Rep. Risø-R-1035(EN), Risø.

[29] Mann, J. (1998) Wind field simulation. *Probabilistic Engineering Mechanics*, **13**, 269–282.

[30] Mann, J. (1994) The spatial structure of neutral atmospheric surface-layer turbulence. *Journal of Fluid Mechanics*, **273**, 141–168.

[31] Shinozuka, M. and Jan, C. M. (1972) Digital simulation of random processes and its application. *Journal of Sound and Vibration*, **25**, 111–128.

[32] Townsend, A. A. (1976) *The Structure of Turbulent Shear Flow*. Cambridge University Press, 2nd edn.

[33] von Karman, T. (1948) Progress in statistical theory of turbulence. *Proceedings of the National Academy of Sciences of the United States of America*, **34**, 530–539.

[34] Menter, F. R. (1993) Zonal two equation $k - \omega$ turbulence models for aerodynamic flows. *24th Fluid Dynamics Conference*, Orlando, FL, July 6–9, AIAA paper no. 93-2906.

[35] Smagorinsky, J. (1963) General circulation experiments with the primitive experiments. *Monthly Weather Review*, **91**, 99–164, DOI: 10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

[36] Moin, P. and Kim, J. (1982) Numerical investigation of turbulent channel flow. *Journal of Fluid Mechanics*, **118**, 341–377.

[37] Jones, W. and Wille, M. (1995) Large eddy simulation of a jet in a crossflow. *10th Symposium on Turbulent Shear Flows*, pp. 4:1–4:6.

[38] Davidson, L. (2007) *Lecture notes for: Unsteady Simulations for Industrial Flows: LES, DES, hybrid RANS/LES and URANS*. CIT Thermoflow AB.

[39] Spalart, P. R. and Allmaras, S. R. (1994) A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale*, **1**, 5–21.

[40] Gilling, L. and Sørensen, N. N. (2008) Large eddy simulation of an airfoil in turbulent inflow. *4th PhD Seminar on Wind Energy in Europe*, Magdeburg, Germany, October 1–2, pp. 33–36.

[41] Gilling, L., Sørensen, N. N., and Davidson, L. (2009) Detached eddy simulations of an airfoil in turbulent inflow. *47th AIAA Aerospace Sciences Meeting*, Orlando, FL, January, AIAA paper no. 2009-270.

[42] Cheng, G., Nichols, R., Neroorkar, K. D., and Radhamony, P. G. (2009) Validation and assessment of turbulence transition models. *47th AIAA Aerospace Sciences Meeting*, Orlando, FL, January, AIAA paper no. 2009-1141.

[43] Comte-Bellot, G. and Corrsin, S. (1966) The use of a contraction to improve the isotropy of grid-generated turbulence. *Journal of Fluid Mechanics*, **25**, 657–682.

[44] Comte-Bellot, G. and Corrsin, S. (1971) Simple eulerian time correlation of full- and narrow-band velocity signals in grid-generated turbulence. *Journal of Fluid Mechanics*, **48**, 273–337.

[45] Bechmann, A. (2006) *Large-Eddy Simulation of Atmospheric Flow over Complex Terrain*. Ph.D. thesis, Risø National Laboratory, Roskilde.

[46] Mish, P. F. (2001) *Mean Loading and Turbulence Scale Effects on the Surface Pressure Fluctuations Occurring on a NACA 0015 Airfoil Immersed in Grid Generated Turbulence*. Master's thesis, Virginia Polytechnic Institute and State University, Virginia.

[47] Mish, P. F. and Devenport, W. (2006) An experimental investigation of unsteady surface pressure on an airfoil in turbulence - Part 1: Effects on mean loading. *Journal of Sound and Vibration*, **296**, 417–446.

[48] Réthoré, P.-E. and Sørensen, N. N. (2008) Actuator disc model using a modified Rhie-Chow/SIMPLE pressure correction algorithm. Comparison with analytical solutions. *EWEC2008*, Brussel, BE, March 31-April 3.

[49] Gilling, L., Sørensen, N. N., and Réthoré, P.-E. (2009) Imposing resolved turbulence by an actuator in a detached eddy simulation of an airfoil. *EWEC 2009*, March.

[50] Troldborg, N. (2008) *Actuator Line Modelling of Wind Turbine Wakes*. Ph.D. thesis, Technical University of Denmark, Lyngby.

[51] Mikkelsen, R. (2003) *Actuator Disc Methods Applied to Wind Turbines*. Ph.D. thesis, Technical University of Denmark, Lyngby.

[52] Pradtlwarter, H. J. and Schüeller, G. I. (1999) Assessment of low probability events of dynamical systems by controlled Monte Carlo simulation. *Probabilistic Engineering Mechnics*, **14**, 213–227.

[53] Veers, P. S. (1984) Modeling stochastic wind loads on vertical axis wind turbines. Tech. Rep. SAND83-1909, Sandia National Laboratories, Albuquerque, New Mexico.

[54] M.T.Sichani, B.J.Pedersen, and S.R.K.Nielsen (2009) Stochastic subspace modelling of turbulence. *International Conference on Civil and Environmental Engineering (ICCEE2009)*, Venice, Italy.

[55] Saranyasoontorn, K., Manuel, L., and Veers, P. S. (2004) A comparison of standard coherence models for inflow turbulence with estimates from field measurements. *Journal of Solar Energy Engineering*, **126**, 1069–1082, DOI: 10.1115/1.797978.

[56] Connell, J. R. (1982) The spectrum of wind speed fluctuations encountered by a rotating blade of a energy conversion system. *Solar Energy*, **29**, 363–375.

Technical Report

TuGen

Synthetic Turbulence Generator
Manual and User's Guide

Lasse Gilling[1]

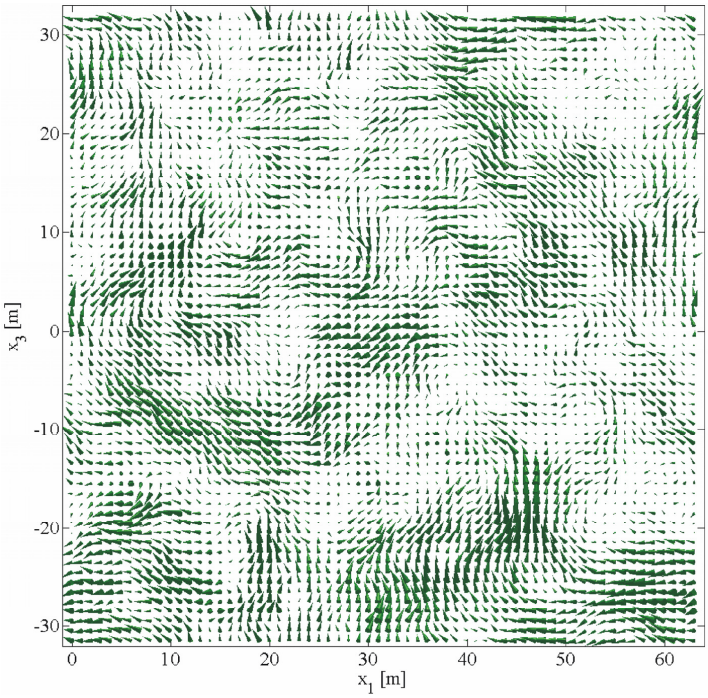
Department of Civil Engineering,
Aalborg University
DCE Technical Report No. 76
ISSN: 1901-726X

[1] Aalborg University, Department of Civil Engineering, Aalborg, Denmark

TuGen

Synthetic Turbulence Generator
Manual and User's Guide

Lasse Gilling



ISSN 1901-726X
DCE Technical Report No. 76

AALBORG UNIVERSITY
Department of Civil Engineering

ISSN 1901-726X
DCE Technical Report No. 76

Aalborg University
Department of Civil Engineering
Division of Structural Mechanics

DCE Technical Report No. 76

TuGen
Synthetic Turbulence Generator, Manual and User's Guide

by

Lasse Gilling

September 2009

© Aalborg University

Scientific Publications at the Department of Civil Engineering

Technical Reports are published for timely dissemination of research results and scientific work carried out at the Department of Civil Engineering (DCE) at Aalborg University. This medium allows publication of more detailed explanations and results than typically allowed in scientific journals.

Technical Memoranda are produced to enable the preliminary dissemination of scientific work by the personnel of the DCE where such release is deemed to be appropriate. Documents of this kind may be incomplete or temporary versions of papers—or part of continuing work. This should be kept in mind when references are given to publications of this kind.

Contract Reports are produced to report scientific work carried out under contract. Publications of this kind contain confidential matter and are reserved for the sponsors and the DCE. Therefore, Contract Reports are generally not available for public circulation.

Lecture Notes contain material produced by the lecturers at the DCE for educational purposes. This may be scientific notes, lecture books, example problems or manuals for laboratory work, or computer programs developed at the DCE.

Theses are monograms or collections of papers published to report the scientific work carried out at the DCE to obtain a degree as either PhD or Doctor of Technology. The thesis is publicly available after the defence of the degree.

Latest News is published to enable rapid communication of information about scientific work carried out at the DCE. This includes the status of research projects, developments in the laboratories, information about collaborative work and recent research results.

Published 2009 by
Aalborg University
Department of Civil Engineering
Sohngaardsholmsvej 57,
DK-9000 Aalborg, Denmark

Printed in Aalborg at Aalborg University

ISSN 1901-726X
DCE Technical Report No. 76

An implementation of Mann's method of generating synthetic turbulence is described. A brief introduction to the method is given prior to some details on the implementation. Several tests to verify the implementation follows and finally the use of the program is described.

The code is written in Fortran 90 in a very 77-like form. It is printed in Section 5 and available for download from:

http://vbn.aau.dk/fbspretrieve/18432733/Fortran_code_for_download.zip

Please note the comment given in Section 5 on page 15. If you have any comments or questions, please feel free to contact me on my e-mail:

lassegilling@hotmail.com

Disclaimer The program, source code and this document is gives "as is". There are no known errors, but no warranty is given. In no event shall the author or Aalborg University be liable for any damages caused.

Contents

1 How it Works **6**

1.1 The Method 6

1.2 Periodicity 8

1.3 Incompressibility 8

1.4 Possible Improvements 8

2 Additional Features **9**

2.1 One-Dimensional Correlation Function 9

2.2 Divergence Correction 9

3 Verifications **10**

3.1 Vector Plot 10

3.2 Isotropy 11

3.3 Spectrum 11

3.4 Correlation 12

3.5 Divergence 12

4 User's Guide **12**

4.1 Input 13

4.2 Output 14

5 The Code **15**

1 How it Works

The program is based on the paper by Mann (1998). There are however some points not addressed in the paper, which are necessary to produce real and divergence free turbulence fields. These issues will be discussed in the following.

1.1 The Method

The method is based on Fourier transform of wave number vectors. Thereby, the velocity field consist of the sum of a large number of linear waves as the one illustrated in Figure 1. If the physical domain is resolved into $N_1 \times N_2 \times N_3$ points the wind field will consist of the sum of $N_1 \times N_2 \times N_3$ many linear waves.

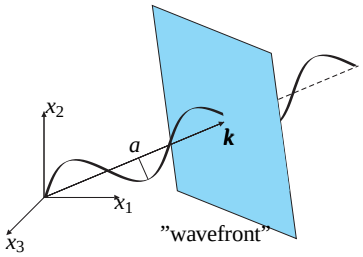


Figure 1: A single linear component of the wind field. $\boldsymbol{k}$ is the wave number vector which has the coordinates of the point in wave number space. The direction and wavelength is given by $\boldsymbol{k}$ and the amplitude a is $E(\boldsymbol{k})$ multiplied by a random number.

To introduce the concept a pseudo algorithm is presented in Algorithm 1. In Section 5 the code is printed. Below, the matrices are defined.

Algorithm 1: Pseudo algorithm

for *all wave number vectors* $\boldsymbol{k}$ *in the domain* **do**
  Determine random vector $\boldsymbol{n}(\boldsymbol{k})$. $\boldsymbol{n}$ must be Gaussian and complex
  Determine matrix $\boldsymbol{B}(\boldsymbol{k})$, c.f. (13) in Mann (1998)
  Determine covariance tensor $\boldsymbol{C}(\boldsymbol{k})$
  Determine Fourier-Stieltje vectors $d\boldsymbol{Z}(\boldsymbol{k}) = \boldsymbol{BC}\,\boldsymbol{n}$
end
Make generated field of $d\boldsymbol{Z}(\boldsymbol{k})$ "symmetric", i.e. $d\boldsymbol{Z}(-\boldsymbol{k}) = d\boldsymbol{Z}^*(\boldsymbol{k})$
Run three dimensional FFT to get velocity field $\boldsymbol{v}(\boldsymbol{x}) = \int e^{i\boldsymbol{k}\cdot\boldsymbol{x}} d\boldsymbol{Z}(\boldsymbol{k})$
Correct divergence if desired
Write velocities to file and/or do diagnostics

The Fourier-Stieltje vectors are determined from

$$d\boldsymbol{Z}(\boldsymbol{k}) = \boldsymbol{BC}\,\boldsymbol{n} \tag{1}$$

where

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & \zeta_1 \\ 0 & 1 & \zeta_2 \\ 0 & 0 & \zeta_3 \end{bmatrix} \tag{2}$$

$$\boldsymbol{C} = \frac{1}{k^2}\sqrt{\frac{E(k)}{4\pi}}\begin{bmatrix} 0 & k_3 & -k_2 \\ -k_3 & 0 & k_1 \\ k_2 & -k_1 & 0 \end{bmatrix} \tag{3}$$

$$\boldsymbol{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{4}$$

$$\zeta_1 = C_1 - \frac{k_2}{k_1}C_2 \qquad , \qquad \zeta_2 = \frac{k_2}{k_1}C_1 + C_2 \qquad , \qquad \zeta_3 = \frac{k_0^2}{k^2} \tag{5}$$

$$C_1 = \frac{\beta k_1^2(k_0^2 - 2k_{3,0}^2 + \beta k_1 k_{3,0})}{k^2(k_1^2 + k_2^2)} \tag{6}$$

$$C_2 = \frac{k_2 k_0^2}{(k_1^2 + k_2^2)^{3/2}} \arctan\left(\frac{\beta k_1\sqrt{k_1^2 + k_2^2}}{k_0^2 - k_{3,0}k_1\beta}\right) \tag{7}$$

$$k_1 = k_{1,0} \qquad , \qquad k_2 = k_{2,0} \qquad , \qquad k_3 = k_{3,0} + \beta k_1 \tag{8}$$

$$k = |\boldsymbol{k}| = \sqrt{k_1^2 + k_2^2 + k_3^2} \qquad , \qquad k_0 = |\boldsymbol{k}_0| = \sqrt{k_1^2 + k_2^2 + k_{3,0}^2} \tag{9}$$

$$\beta = (kL)^{-2/3}\Gamma \tag{10}$$

For $\boldsymbol{k} = \boldsymbol{0}$ (5)-(7) will be undefined. In this point $\boldsymbol{C}$ will be zero so $\boldsymbol{B}$ is chosen as the identity matrix to prevent the program to fail due to division with zero. For $k_1 = 0$ the limit values of ζ_1, ζ_2 and ζ_3 will be used

$$\lim_{k_1 \to 0} \zeta_1 = -\beta \qquad , \qquad \lim_{k_1 \to 0} \zeta_2 = 0 \qquad , \qquad \lim_{k_1 \to 0} \zeta_3 = 1 \tag{11}$$

Each point in the resolved spectral domain is described by a wave number vector $\boldsymbol{k}$. $E(\boldsymbol{k})$ is the energy spectrum, which gives the energy associated with the linear wave defined by the wave number vector $\boldsymbol{k}$

$$E(\boldsymbol{k}) = \alpha\varepsilon^{2/3}L^{5/3}\frac{L^4k^4}{(1 + L^2k^2)^{17/6}} \tag{12}$$

L defines the length scale and $\alpha\varepsilon^{2/3}$ scales the intensity. Γ gives the degree of anisotropy. These three parameters should be given as input to the program along with the dimensions of the domain.

To ensure the generated wind field is real the Fourier-Stieltje components must satisfy the condition

$$d\boldsymbol{Z}(-\boldsymbol{k}) = d\boldsymbol{Z}^*(\boldsymbol{k}) \tag{13}$$

Consequently, only about half of the random numbers can be chosen arbitrarily. For some special $\boldsymbol{k}$, e.g. $\boldsymbol{k} = \boldsymbol{0}$, (13) enforces the imaginary part to be zero.

The above equations are based on the assumption that the side lengths of the domain are large compared to the turbulence length scale. Mann (1998) states that the approximation is good for $dx_{(j)}N_{(j)} < 8L$ for $j = 2, 3$, where the parentheses in the index cancels the summation convention.

1.2 Periodicity

As the FFT-algorithm assumes the function to be periodic, the generated field will be periodic in all three directions. That is, there will be a correlation coefficient of (nearly) unity of the velocities on each side of the box.

In most applications this is not desired. To remedy this problem, generate a wind field larger than the desired domain and discard some of the data. The required size of the generated wind field depends on the resolution of the domain and the length scale of the turbulence. For wind turbines the largest of $2D$ or $8L$ is recommended, where D is the rotor diameter.

1.3 Incompressibility

The components of the wind field are determined from Fourier transformation of the Fourier-Stieltje vectors

$$\boldsymbol{v}(\boldsymbol{x}) = \int e^{i\boldsymbol{k}\cdot\boldsymbol{x}} d\boldsymbol{Z}(\boldsymbol{k}) \tag{14}$$

Incompressibility of the wind field requires zero divergence of the field. Using index notation with the summation convention the condition can be written

$$\frac{\partial v_j}{\partial x_j} = 0 \tag{15}$$

By combining (14) and (15) a condition can be imposed on the Fourier-Stieltje vectors

$$\frac{\partial v_j}{\partial x_j} = i \int e^{i\boldsymbol{k}\cdot\boldsymbol{x}} k_j dZ_j(\boldsymbol{k}) = 0 \quad \Rightarrow \quad \boxed{k_j dZ_j(\boldsymbol{k}) = 0} \tag{16}$$

Pre-multiplication of (1) with $\boldsymbol{k}$ yields

$$\boldsymbol{k}\, d\boldsymbol{Z}(\boldsymbol{k}) = \frac{1}{k^2}\sqrt{\frac{E(k)}{4\pi}} \begin{bmatrix} k_1k_2\zeta_1 - k_2k_3 + k_2^2\zeta_2 + k_2k_3\zeta_3 \\ k_1k_3 - k_1^2\zeta_1 - k_1k_2\zeta_2 - k_1k_3\zeta_3 \\ 0 \end{bmatrix}^T \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{17}$$

If n_1 and n_2 are independent random variables (17) will only be zero in the isotropic case where $\zeta_1 = \zeta_2 = 0$ and $\zeta_3 = 1$. If n_1 and n_2 are chosen to satisfy $n_2 = n_1k_2/k_1$ then (17) will also be zero in the anisotropic case. This is a poor choice, however, as this will give $v_3 = 0$ in all points.

If a divergence free field is required then the divergence may be corrected as described in Section 2.2.

1.4 Possible Improvements

Below are some possible improvements listed:

- In the present implementation the entire field of Fourier-Stieltje vectors are generated from independent random numbers. Then, symmetry is enforced as defined in (13). This procedure costs a little on the computation time, but it is easier to change the way symmetry is enforced.

- The divergence correction can implemented by the TDMA algorithm instead of the MINRES algorithm for solving the linear equations which might increase computational speed.
- The approximation for large dimensions of the domain compared to the turbulent length scale can be avoided, but then the spectral tensor must be obtained by numerical integration. This integration could be implemented.

2 Additional Features

Below, two features are presented. Both can be used, but they are not required for generating the velocity fields. The first determines the one-dimensional correlation function which can be used to verify or describe the field. It produces extra output but does not modify the generated turbulence field. The latter is a divergence correction method. It produces no extra output, but makes changes to the velocity field.

2.1 One-Dimensional Correlation Function

In all three directions of the domain the one-dimensional auto-correlation function is determined for all three components. The correlations are averaged over the remaining two directions.

Further, the average correlation between the three velocity components are determined.

2.2 Divergence Correction

For isotropic field the divergence of the continuous field will be zero. This is not the case for the discrete representation, however. As was shown in Section 1.3 anisotropic fields are not divergence free, even in the continuous representation. In this section a method to correct the divergence to zero will be shown.

The corrected velocity field can be determined from

$$\tilde{v}_j(\boldsymbol{x}) = v_j(\boldsymbol{x}) - \frac{\partial p(\boldsymbol{x})}{\partial x_j} \tag{18}$$

where the function $P(\boldsymbol{x})$ is defined from

$$\frac{\partial^2 p}{\partial x_j^2} = \frac{\partial v_j}{\partial x_j} \tag{19}$$

By this definition $\tilde{v}_j(\boldsymbol{x})$ will satisfy (15) identically. The definition of p is recognized as a Poisson differential equation. For a given numerical differential scheme the equation can be solved by solving a system of linear equations. Here and in the code the 2nd order central difference scheme will be used.

The divergence is determined in all points and stored in a vector $\boldsymbol{d}$. It is used as right hand side in (19). The problem is formulated as a linear system $\boldsymbol{d} = \boldsymbol{A}\boldsymbol{p}$, where $\boldsymbol{A}$ is a coefficient matrix defined from the numerical differential scheme. The linear system is solved by the MINRES algorithm, which is possible because $\boldsymbol{A}$ is symmetric.

In the MINRES algorithm matrix-vector multiplications are performed repeatedly. For typical dimensions of the domain the number of points is so large that it would require huge amounts of memory to allocate the full coefficient matrix. If for instance $N_1 = 8192$ and $N_2 = N_3 = 32$ the coefficient matrix should be of dimension $8,388,608 \times 8,388,608$. However, the matrix is very sparse and has a simple structure.

A subroutine has been written that will take any vector as input and give the product with the coefficient matrix $\boldsymbol{A}$ as output. By utilizing the simple structure of the matrix this has been implemented in a way that never forms the matrix it self (not even in some kind of sparse format). The approach has very low demands to memory and is quite fast due to the efficiency of the MINRES algorithm.

3 Verifications

To check the implementation of the program several tests have been performed. Below some of the results from these tests are presented.

3.1 Vector Plot

In Figure 2 a vector plot of a generated velocity field is shown. Each cone in the figure represents a velocity vector. The sizes of the cones are scaled to show the magnitude of the vectors. It is seen that the field is periodic. Further, it illustrates the spatial correlation as structures can be seen in the field.

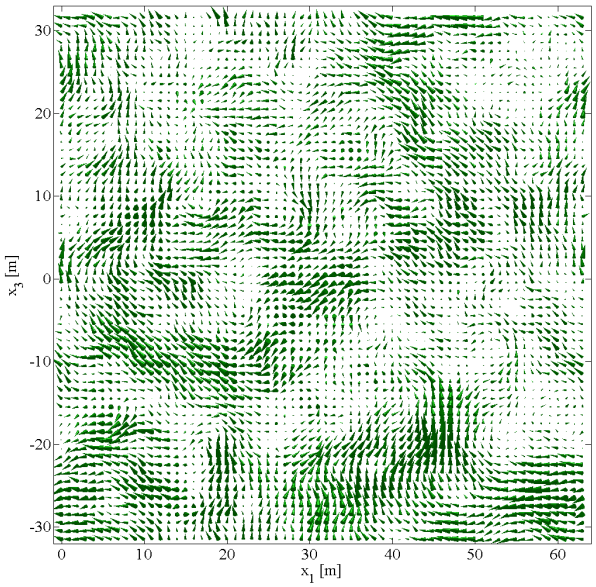


Figure 2: Vector plot of a generated velocity field. The field is isotropic with $dx_j = 1m$, $N_j = 64$ and $L = 25m$.

3.2 Isotropy

In Figure 3 the variances of the three velocity components are shown for 250 uncorrelated realizations. The generated field and the domain it self is isotropic with $\Gamma = 0$, $dx_1 = dx_2 = dx_3$ and $N_1 = N_2 = N_3$. Therefore, the average variances of the three components should be identical. It is seen that there are large variations from realization to realization but the average variances are close to identical.

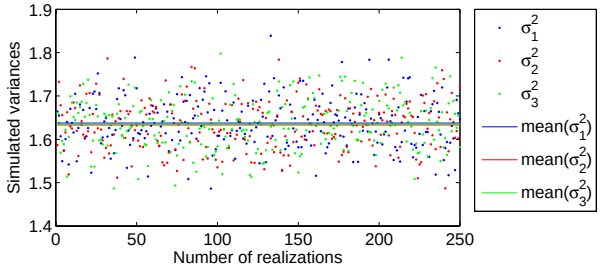


Figure 3: Variances of the three velocity components determined from the 250 realizations.

3.3 Spectrum

In Figure 4 the one-sided, one-dimensional spectrum of generated v_1 components is shown. The spectrum from the realization has been averaged over the x_2-x_3-plane. The theoretical spectrum given by

$$S_{11}(k_1) = \frac{18}{55}\alpha\varepsilon^{2/3}L^{5/3}\frac{1}{\left(1+L^2k_1^2\right)^{5/6}} \tag{20}$$

From the figure it is observed, that the agreement is good, but for large wave numbers the realization gives too low energy, due to aliasing errors, which was also reported by Mann (1998).

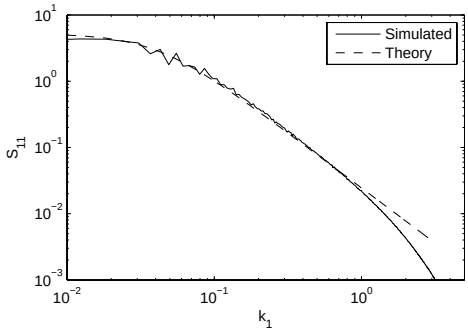


Figure 4: One-sided, one-dimensional spectrum of the u_1-components. The spectrum of the realization has been averaged over the y- and z-directions.

3.4 Correlation

One dimensional correlations are plotted in Figure 5. The computed correlations are compared to the theoretic functions by von Karman

$$f(r) = \frac{2}{\Gamma\left(\frac{1}{3}\right)}\left(\frac{r}{2L}\right)^{\frac{1}{3}} K_{\frac{1}{3}}\left(\frac{r}{L}\right) \tag{21}$$

$$g(r) = \frac{2}{\Gamma\left(\frac{1}{3}\right)}\left(\frac{r}{2L}\right)^{\frac{1}{3}} \left[K_{\frac{1}{3}}\left(\frac{r}{L}\right) - \frac{2}{2L} K_{\frac{2}{3}}\left(\frac{r}{L}\right)\right] \tag{22}$$

where r is the separation distance and $K_a(x)$ is the modified Bessel function of second kind and order a. $f(r)$ describes the correlation in the direction as the velocity component (e.g. correlation of v_2 in the x_2-direction) and $g(r)$ describes the correlation in transverse directions (e.g. correlation of v_2 in the x_3-direction). Again, the agreement is good.

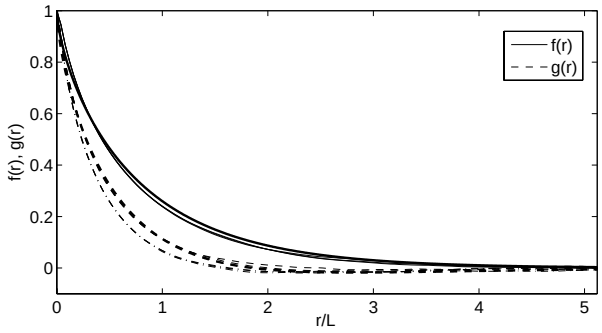


Figure 5: One dimensional correlation function calculated from realizations compared to the theoretic von Karman correlation functions of isotropic turbulence. Thick lines are (21) and (22) and thin are computed from realizations with the feature described in Section 2.1.

3.5 Divergence

The divergence of the vector field $\boldsymbol{v}$ is given by (15), and written out in components it reads

$$div(\boldsymbol{v}) = \frac{\partial v_j}{\partial x_j} = \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \approx \frac{\Delta v_1}{\Delta x_1} + \frac{\Delta v_2}{\Delta x_2} + \frac{\Delta v_3}{\Delta x_3} \tag{23}$$

In Figure 6 the three components are compared to their sum. The derivative are approximated by their derivatives from the 2nd order difference scheme.

It can be seen that the divergence evaluated by the 2nd order central difference scheme is non-zero. By applying the algorithm from Section 2.2 the divergence is corrected to zero as shown on the bottom figure.

4 User's Guide

The following briefly describes the use of the developed program. The input and output is described.

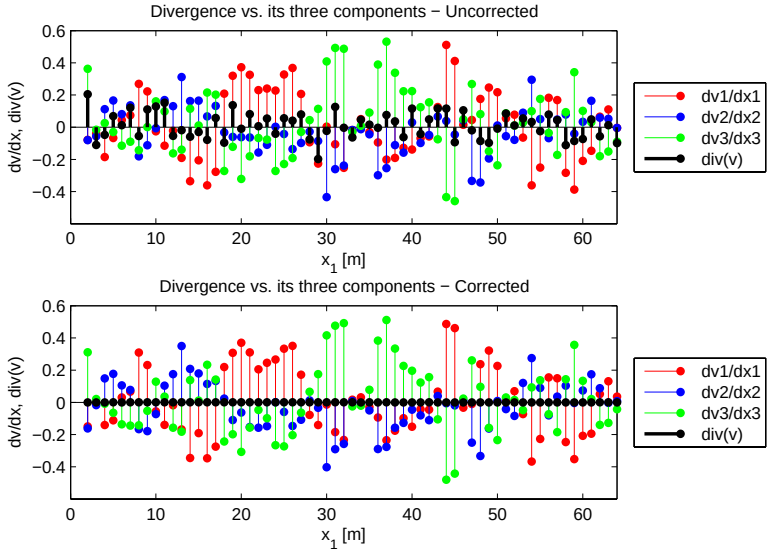


Figure 6: Comparison of the divergence and its three components. On the bottom figure the divergence has been corrected by the algorithm outlined in Section 2.2.

4.1 Input

The program reads 15 parameters from an input file. An example of such an input file is shown in Figure 7. The filename must be `input.inp`.

```
256     N1   (Variable names here are optional)
256     N2
256     N3
1.0     dx1
1.0     dx2
1.0     dx3
33.6    L
0.181   alphaeps
0.0     Gamma
-1      randseed (-1 => randseed generated from current time)
1       write velocities to file
1       print screen output
0       correct divergence
0       write variance to file
0       write correlation to file
```

Figure 7: Example of input file.

The first 10 lines define parameters and the remaining 5 lines are switches, that should be either 0 or 1. As stated in the file the names are optional and they are not read by the program. The order of the inputs must be exactly as in the example and no lines can be added or deleted. A blank space must separate the variables from the describing text.

The three first lines define the resolution of the domain in the three directions. These must be on the form 2^n where n is a positive integer. The next three lines define the grid spacing in the three directions, dx_1, dx_2, dx_3.

L is the turbulent length scale. `alphaeps` defines $\alpha\varepsilon^{2/3}$ which scales the intensity of the turbulence. Γ (`Gamma`) is a parameter defining the degree of anisotropy of the turbulence. In Mann (1998) typical values are given.

The random seed can be set in the input file. The random seed should be an integer. If it is set to -1, the program will generate a random random seed and write it to an output file. The generated random seed is based on the current time.

The last five lines define switches that must be 0 or 1. If the program should write out the velocities to files, the first switch should be set to 1.
The next line defines whether or not the program should write to the screen as the execution progresses.
The third switch should be set to 1, if the program should correct the divergence of the discrete field to zero (when evaluated by the second order central difference scheme).
The second to last switch defines whether the program should determine the variances of the three velocity components to an output file.
The last switch should be set to 1, if the program should determine the correlation of the velocity field and write it to a number of files. The correlation is plotted in Figure 5.

4.2 Output

Depending on the parameters in the input file the program will write data to the files described here.

If the random seed is not specified (`randseed` is set to -1 in the input file):

- The program generates a random random seed and writes it to `randomseed.dat`

If the velocity field should be written to files:

- Velocity components are written to `u.bin`, `v.bin` and `w.bin`. The data is written in (32-bit, real*4) binary form with the 3rd index varying fastest and the 1st varying slowest. (Example of how to read the data is given below)

If the variances should be computed and written to a file:

- The computed variances will be written to the file `variance.dat`. The order is σ_1, σ_2, σ_3

If the correlation should be computed and written to files:

- One-sided, one-dimensional correlation functions are written to files `fu.dat`, `gu2.dat`, `gu3.dat`, `fv.dat`, `gv1.dat`, `gu3.dat`, `fw.dat`, `gw1.dat` and `gw2.dat`. The file names corresponds to (21) and (22). The number in the files starting with `g` is the direction and the letter is the velocity component
- Average cross-correlation coefficients of the three components is written to the file `correlation.dat`. The order is ρ_{12}, ρ_{13}, ρ_{23}

Except for the files containing velocity components the data is written formatted to ASCII files. The data is written in lines and if old files exist when the program is run, it will append the data from the new realization.

The velocity field can be read with Algorithm 2 (written in Fortran). The part shown is not complete. It only reads the *u* components from `u.bin` and definition of most variables is omitted here.

Algorithm 2: Algorithm for reading the velocity field from files

```
real*4, allocatable :: varread(:)

allocate(varread(N1*N2*N3))
allocate(u(N1,N2,N3))
open(16,file='u.bin',form='binary')
read(16) varread
jj1=0
do j1=1,N1
   do j2=1,N2
      do j3=1,N3
         jj1=jj1+1
         u(j1,j2,j3)=varread(jj1)
      enddo
   enddo
enddo
close(16)
```

References

Mann, J. (1998). Wind field simulation. *Prob. Engng. Mech.*, 13(4):269–282.

5 The Code

The code is printed in full below, and it can be downloaded from

http://vbn.aau.dk/fbspretrieve/18432733/Fortran_code_for_download.zip

A comment should be given on the FFT subroutine and how it is called. The Fourier-Stieltje components are Fourier transformed by the subroutine `fourn`. The subroutine expects a 1D array but the 3D array stored in dZ(i,:,:,:) is given as input. The trick is that Fortran stores the components in dZ in the same order as the subroutine expects them. Therefore, the calls of `fourn` will give the correct results. On some compilers the difference in dimensions gives warnings, and unfortunately on some compilers it generates errors. Yet, I have chosen this approach to save the memory otherwise required to allocate a new 1D variable that contains all the components. The argument for this choice is that memory often sets a rather strict limitation for how large fields the code can generate.

```
!  TuGen.f90
!*****************************************************************************
!  PROGRAM: TurbulenceSimulation
!  PURPOSE: Simulate turbulence field using three dimensional
!      inverse Fast Fourier transform
!  DATE:    May 2008
!  WRITTEN by:
!           Lasse Gilling, M.Sc., Ph.D.-student
!           Department of Civil Engineering
!           Aalborg University
!           Mail: lg@civil.aau.dk and/or lassegilling@hotmail.com
!*****************************************************************************
program TuGen
!*****************************************************************************
! Variables
!*****************************************************************************
implicit none
integer N1,N2,N3                ! number of points in three directions
real(8) L,alphaeps              ! Integral length scale and alpha*eps**(2./3.)
real(8) Gamma                   ! Parameters defining anisotropy
real(8) dx1,dx2,dx3             ! grid spacing in physical space
integer j1,j2,j3                ! Counters
integer jj1,jj2,jj3             ! ~Counters: Help to make program reader friendly
real(8) dk1,dk2,dk3             ! Wave number increments
real(8) k1,k2,k3                ! Wave number components
real(8) k30                     ! Initial wave number component in 3-direction
real(8) k0square, ksquare       ! Squared length of wave number vectors
real(8) E                       ! Von Karman spectral function
real(8) beta                    ! Beta in (Mann 1998)-article
real(8) B(3,3)                  ! B-matrix from "Rapid Distorsion Theory"
real(8) B1, B2, B3              ! Coefficients in B-matrix
real(8) C1,C2                   ! Parameters used for calculating B-matrix
real(8) C(3,3)                  ! C-matrix
real(8) H(3,3)                  ! H-matrix
complex*16 W(3,1)               ! Normal distributed complex random numbers
real(8) U1(3,1), U2(3,1)        ! Uniform distributed real random numbers
integer seedsize                ! Used to set the seed of the random generator
real(8) pi                      ! Ratio of diameter to circumference in circle
complex*16 S(3,1)               ! Helping variable to define elements in dZ
real(8) starttime,endtime       ! Used to determine calculation time
integer status, ios             ! Used to chech status of the ifft and read
real(8) vars(15)                ! Array used for reading input
integer randseed                ! seed for rand. gen
integer write_vel,write_screen  ! switches: write velocity, write status to screen
integer div_cor,write_var       !           correct divergence, write variances
integer write_cor               !           write correlation function f(r)
real(8) rho, rho12,rho13,rho23  ! Correlation coef.
real(8) s1_2,s2_2,s3_2          ! Variance of the the generated vel. field
real(8) tke                     ! Turbulent kinetic energy
integer time_array(8)           ! Used for generating randomseed if unspecified
complex*8, allocatable :: dZ(:,:,:,:) ! Fourier-Stieltje coefficients
real(8), allocatable :: v1(:,:,:)     ! Velocities in 1 direction
real(8), allocatable :: v2(:,:,:)     ! Velocities in 2 direction
real(8), allocatable :: v3(:,:,:)     ! Velocities in 3 direction
real(8), allocatable :: f1(:)         ! Correlation function, f(r)
real(8), allocatable :: g2(:)         ! Correlation function, g(r) , v2-comp
real(8), allocatable :: g3(:)         ! Correlation function, g(r) , v3-comp
real(4), allocatable :: VarWrite(:)   ! Write velocities to file
logical inpbool                 ! Check if the input-file exist
!*****************************************************************************
! Read input from file
```

```
!*******************************************************************************
vars=0d0
! set default control parameters. 1 for yes, 0 for no
vars(10)=-1 ! randseed
vars(11)=1  ! writevel
vars(12)=1  ! write to screen
vars(13)=0  ! divergence cor
vars(14)=0  ! compute variances and write to file
vars(15)=0  ! compute correlations and write til file
inquire(file='input.inp',exist=inpbool)
if(.not.inpbool)then
 write(*,*) 'ERROR: Cannot find input-file! Simulation aborted'
 stop
endif
open(unit=10,file='input.inp')
do j1=1,15
  read(10,*,iostat=ios) vars(j1)
  if(ios.ne.0)then
    if(j1.lt.10)then
      write(*,*) 'ERROR: Not enough data in input-file! Simulation aborted'
      stop
    endif
    exit
  endif
enddo
close(unit=10)
! Set variables
N1=vars(1)
N2=vars(2)
N3=vars(3)
dx1=vars(4)
dx2=vars(5)
dx3=vars(6)
L=vars(7)
alphaeps=vars(8)
Gamma=vars(9)
randseed=vars(10)
write_vel=vars(11)
write_screen=vars(12)
div_cor=vars(13)
write_var=vars(14)
write_cor=vars(15)
! check size of domain
call check2(N1,j1)
if(j1.ne.1)then
  print*,'Error: N1 must be on the form 2^n'
  stop
endif
call check2(N2,j1)
if(j1.ne.1)then
  print*,'Error: N2 must be on the form 2^n'
  stop
endif
call check2(N3,j1)
if(j1.ne.1)then
  print*,'Error: N3 must be on the form 2^n'
  stop
endif
!*******************************************************************************
! Initialization
!*******************************************************************************
```

```
allocate(dZ(3,N1,N2,N3))
if(write_vel.eq.1)then
  allocate(varwrite(N1*N2*N3))
endif
if(write_screen.eq.1)then
write(*,100) '+------------------------------------------------------------+'
write(*,100) '| Simulation of incompressible turbulence field has begun  |'
write(*,100) '+------------------------------------------------------------+'
endif
! Start timer
call cpu_time(starttime)
pi=4d0*datan(1d0)
! Set random seed
if(randseed.eq.-1)then
  call date_and_time (values=time_array)
  randseed=sum(time_array)*(time_array(8)+1)
  open(unit=55,file='randseed.dat',access='append')
   write(55,*) randseed
  close(55)
endif
call random_seed(SIZE=seedsize)
call random_seed(PUT=[1:seedsize]*randseed)
!*****************************************************************************
! Generate the elements of dZ (i.e. the Fourier-Stieltje coefficients)
!*****************************************************************************
dk1=2d0*pi/dx1/dfloat(N1)
dk2=2d0*pi/dx2/dfloat(N2)
dk3=2d0*pi/dx3/dfloat(N3)
do j3=1,N3
  do j2=1,N2
    do j1=1,N1
        ! ------------------------------------------------------------
        ! Generate random complex vector W
        ! ------------------------------------------------------------
        ! Generate uniformly distributed random numbers
        call random_number(U1)
        call random_number(U2)
        ! Box-Muller transformation to make normal distributed random numbers
        W=dcmplx(dsqrt(-2d0*dlog(U2))*dcos(2d0*pi*U1),&
                 dsqrt(-2d0*dlog(U2))*dsin(2d0*pi*U1))
        ! For ji=0 or Ni/2+1 the imaginary part of W must be zero
        if ((((j1.eq.1).or.(j1.eq.N1/2+1)).and.((j2.eq.1).or.(j2.eq.N2/2+1)))&
                                    .and.((j3.eq.1).or.(j3.eq.N3/2+1))) then
          W=dcmplx(dble(W),0d0)
        end if
        ! ------------------------------------------------------------
        ! Determine initial wave number vector
        ! ------------------------------------------------------------
        ! Change variable to jj that takes negative values for j.ge.N/2
        ! - i.e. shift integral range back to [-kmax;kmax[ instead of
        !   the range in iFFT [0;2*kmax[
        if(j1.ge.N1/2)then
          jj1=-N1+j1-1
        else
          jj1=j1-1
        endif
        if(j2.ge.N2/2)then
          jj2=-N2+j2-1
        else
          jj2=j2-1
        endif
```

```
if(j3.ge.N3/2)then
  jj3=-N3+j3-1
else
  jj3=j3-1
endif
! Set components of initial wave number vector
k1=jj1*dk1
k2=jj2*dk2
k3=jj3*dk3
! ------------------------------------------------------------
! Generate B-matrix to simulate anisotropy
! ------------------------------------------------------------
ksquare=k1**2+k2**2+k3**2
if (ksquare.eq.0d0) then
 ! C-matrix will be 0 and can be multiplied with anything (except NaN)
  beta=0d0
else
  beta=Gamma*(dsqrt(ksquare)*L)**(-2d0/3d0)
endif
k30=k3+beta*k1
k0square=k1**2+k2**2+k30**2
! Determine coefficients in B-matrix
if (ksquare.eq.0d0) then
  ! C-matrix will be 0 - B-matrix can be chosen arbitrarily
  B1=0d0
  B2=0d0
  B3=1d0
elseif (k1.eq.0d0) then ! use limit values for k1->0
  B1=-beta
  B2=0d0
  B3=1d0
else
  C1=beta*(k1**2)*(k0square-2*(k30**2)+beta*k1*k30) &
    /(ksquare*(k1**2+k2**2))
  C2=k2*k0square/(k1**2+k2**2)**(3d0/2d0) &
    *datan(beta*k1*((k1**2+k2**2)**0.5d0)/(k0square-k30*k1*beta))
  B1=C1-k2*C2/k1
  B2=k2*C1/k1+C2
  B3=k0square/ksquare
endif
! Assemble matrix
B(1,:)=[1d0,0d0,B1]
B(2,:)=[0d0,1d0,B2]
B(3,:)=[0d0,0d0,B3]
! ------------------------------------------------------------
! Generate C-matrix from initial wave number vector
! ------------------------------------------------------------
C(1,:)=[ 0d0,  k30, -k2 ]
C(2,:)=[-k30,  0d0,  k1 ]
C(3,:)=[ k2 , -k1 ,  0d0]
E=alphaeps*L**(5d0/3d0)*(L**2*k0square)**2/(1d0+(L**2*k0square))**(17d0/6d0)
if (k0square.eq.0d0) then
  C=0d0
else
  C=C*dsqrt(E/4d0)/pi/k0square*dsqrt(dk1*dk2*dk3)
endif
! ------------------------------------------------------------
! Create element in dZ by multiplication of matrices and vector
! ------------------------------------------------------------
S=matmul(B,matmul(C,W))
dZ(:,j1,j2,j3)=[S(1,1),S(2,1),S(3,1)]
```

```
    end do !j1
  end do !j2
end do !j3
! Enforce symmetry
do j3=1,N3
  do j2=1,N2
    do j1=1,N1
      ! ------------------------------------------------------------------------
      ! make all the symmetry conditions reqiured for the velocities to become real
      ! ------------------------------------------------------------------------
      if    (((j2.eq.1).or.(j2.eq.N2/2+1)).and.(((j3.eq.1)&
                                .or.(j3.eq.N3/2+1))).and.(j1.gt.N1/2+1))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,j2,j3))
      elseif(((j1.eq.1).or.(j1.eq.N1/2+1)).and.((j3.eq.1)&
                                .or.(j3.eq.(N3/2+1))).and.(j2.gt.N2/2+1))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,j1,N2-j2+2,j3))
      elseif(((j2.eq.1).or.(j2.eq.N2/2+1)).and.((j1.eq.1)&
                                .or.(j1.eq.(N1/2+1))).and.(j3.gt.N3/2+1))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,j1,j2,N3-j3+2))
      elseif((j1.gt.N1/2+1).and.((j2.eq.1).or.(j2.eq.N2/2+1)))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,j2,N3-j3+2))
      elseif((j1.gt.N1/2+1).and.((j3.eq.1).or.(j3.eq.N3/2+1)))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,N2-j2+2,j3))
      elseif((j2.gt.N2/2+1).and.((j1.eq.1).or.(j1.eq.N1/2+1)))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,j1,N2-j2+2,N3-j3+2))
      elseif((j2.gt.N2/2+1).and.((j3.eq.1).or.(j3.eq.N3/2+1)))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,N2-j2+2,j3))
      elseif((j3.gt.N3/2+1).and.((j1.eq.1).or.(j1.eq.N1/2+1)))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,j1,N2-j2+2,N3-j3+2))
      elseif((j3.gt.N3/2+1).and.((j2.eq.1).or.(j2.eq.N2/2+1)))then
        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,j2,N3-j3+2))
      ! ------------------------------------------------------------------
      ! The last symmetry-condition is a "mirroring" in a plane
      ! ------------------------------------------------------------------
      ! a) the x1-x2-plane
!      elseif (j3.gt.N3/2+1) then
!        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,N2-j2+2,N3-j3+2))
       !b) plane defined by x1+x2+x3=0 (Good choise => var1=var2=var3)
      elseif((dfloat(j1)/dfloat(N1)+dfloat(j2)/dfloat(N2)&
              +dfloat(j3)/dfloat(N3)-3d0/2d0).gt.0d0)then
        dZ(:,j1,j2,j3)=conjg(dZ(:,N1-j1+2,N2-j2+2,N3-j3+2))
      endif
    enddo !j1
  enddo !j2
enddo !j3
if(write_screen.eq.1)then
write(*,100) '| Fourier coefficients have been determined               |'
endif
!*****************************************************************************
! Do the iFFT
!*****************************************************************************
! Next, the Fourier-Stieltje components are Fourier Transformed. The
! subroutine expects a 1D array and dZ(i,:,:,:) is 3D. The trick is that
! Fortran stores the components in dZ in the same order as the subroutine
! expects them. Therefore, the following calls will give the correct results.
! On some compilers the difference in dimensions gives warnings, and
! unfortunately on some compilers it generates errors.
! Yet, I have chosen this approach to save the memory otherwise required to
! allocate a new 1D variable that contains all the components of dZ. The
! argument for this choice is that memory often sets a rather strict
! limitation for how large fields the code can generate.
```

```
call fourn(dZ(1,:,:,:),[N1,N2,N3],3,-1)
call fourn(dZ(2,:,:,:),[N1,N2,N3],3,-1)
call fourn(dZ(3,:,:,:),[N1,N2,N3],3,-1)
if(write_screen.eq.1)then
write(*,100) '| Velocity field has been generated                        |'
endif
!*****************************************************************************
! Allocate velocities if they are used frequently
!*****************************************************************************
if(div_cor+write_cor+write_var.ne.0)then
  allocate(v1(N1,N2,N3))
  allocate(v2(N1,N2,N3))
  allocate(v3(N1,N2,N3))
  v1=dble(dZ(1,:,:,:))
  v2=dble(dZ(2,:,:,:))
  v3=dble(dZ(3,:,:,:))
endif
!*****************************************************************************
! Correct divergence to zero (for cds2) - Subroutine not included in this file
!*****************************************************************************
if(div_cor.eq.1)then
  call div_correction(v1,v2,v3,N1,N2,N3,dx1,dx2,dx3)
 ! If divergence is corrected the veolcities are written to files here
 ! - Otherwise v1,v2,v3 may not be allocated and the velocities are written
 !   later
 ! write v1-component
  if(write_vel.eq.1)then
   open(16,file='u.bin',form='binary')
   jj1=0
   do j1=1,N1
    do j2=1,N2
     do j3=1,N3
       jj1=jj1+1
       varwrite(jj1)=dble(v1(j1,j2,j3))
     enddo
    enddo
   enddo
   write(16) varwrite
   close(16)
 ! write v2-component
   open(17,file='v.bin',form='binary')
   jj1=0
   do j1=1,N1
    do j2=1,N2
     do j3=1,N3
       jj1=jj1+1
       varwrite(jj1)=dble(v2(j1,j2,j3))
     enddo
    enddo
   enddo
   write(17) varwrite
   close(17)
 ! write v3-component
   open(18,file='w.bin',form='binary')
   jj1=0
   do j1=1,N1
    do j2=1,N2
     do j3=1,N3
       jj1=jj1+1
       varwrite(jj1)=dble(v3(j1,j2,j3))
     enddo
```

```
    enddo
   enddo
   write(18) varwrite
   close(18)
  endif
endif
!*****************************************************************************
! Write velocity field to unformatted files
!*****************************************************************************
if((write_vel.eq.1).and.(div_cor.ne.1))then
open(16,file='u.bin',form='binary')
open(17,file='v.bin',form='binary')
open(18,file='w.bin',form='binary')
do jj2=1,3
 jj1=0
 do j1=1,N1
  do j2=1,N2
   do j3=1,N3
     jj1=jj1+1
     varwrite(jj1)=dble(dZ(jj2,j1,j2,j3))
   enddo
  enddo
 enddo
 write(15+jj2) varwrite
enddo
close(16)
close(17)
close(18)
endif
!*****************************************************************************
! Do some diagnostics
!*****************************************************************************
if(write_cor.eq.1)then
! Calculate average correlation between v1 and v3 (and v1-v2, v2-v3)
  rho12=0d0
  rho13=0d0
  rho23=0d0
  do j3=1,N3
    do j2=1,N2
      call corrcoef(v1(:,j2,j3),v2(:,j2,j3),N1,rho)
      rho12=rho12+rho
      call corrcoef(v1(:,j2,j3),v3(:,j2,j3),N1,rho)
      rho13=rho13+rho
      call corrcoef(v2(:,j2,j3),v3(:,j2,j3),N1,rho)
      rho23=rho23+rho
    end do
  end do
  rho12=rho12/dfloat(N2*N3)
  rho13=rho13/dfloat(N2*N3)
  rho23=rho23/dfloat(N2*N3)
  ! write results to file
  open(50,file='correlation.dat')
  write(50,111) [rho12,rho13,rho23]
  close(50)
  ! ** Determine correlation function f(r) and g(r) in the x-direction **
  allocate(f1(N1))
  allocate(g2(N1))
  allocate(g3(N1))
  f1=0d0
  g2=0d0
  g3=0d0
```

```
do j1=1,N1
  do j2=1,N2
    call corrcoef(v1(1,j2,:),v1(j1,j2,:),N3,rho)
    f1(j1)=f1(j1)+rho/dfloat(N2)
    call corrcoef(v2(1,j2,:),v2(j1,j2,:),N3,rho)
    g2(j1)=g2(j1)+rho/dfloat(N2)
    call corrcoef(v3(1,j2,:),v3(j1,j2,:),N3,rho)
    g3(j1)=g3(j1)+rho/dfloat(N2)
  end do
end do
open(51,file='fu.dat',access='append')
write(51,111) f1
close(51)
open(51,file='gv1.dat',access='append')
write(51,111) g2
close(51)
open(51,file='gw1.dat',access='append')
write(51,111) g3
close(51)
deallocate(f1)
deallocate(g2)
deallocate(g3)
! ** Determine correlation function f(r) and g(r) in the y-direction **
allocate(f1(N2))
allocate(g2(N2))
allocate(g3(N2))
f1=0d0
g2=0d0
g3=0d0
do j2=1,N2
  do j3=1,N3
    call corrcoef(v2(:,1,j3),v2(:,j2,j3),N1,rho)
    f1(j2)=f1(j2)+rho/dfloat(N3)
    call corrcoef(v3(:,1,j3),v3(:,j2,j3),N1,rho)
    g2(j2)=g2(j2)+rho/dfloat(N3)
    call corrcoef(v1(:,1,j3),v1(:,j2,j3),N1,rho)
    g3(j2)=g3(j2)+rho/dfloat(N3)
  end do
end do
open(51,file='fv.dat',access='append')
write(51,111) f1
close(51)
open(51,file='gw2.dat',access='append')
write(51,111) g2
close(51)
open(51,file='gu2.dat',access='append')
write(51,111) g3
close(51)
deallocate(f1)
deallocate(g2)
deallocate(g3)
! ** Determine correlation function f(r) and g(r) in the z-direction **
allocate(f1(N3))
allocate(g2(N3))
allocate(g3(N3))
f1=0d0
g2=0d0
g3=0d0
do j3=1,N3
  do j1=1,N1
    call corrcoef(v3(j1,:,1),v3(j1,:,j3),N2,rho)
```

```
      f1(j3)=f1(j3)+rho/dfloat(N1)
      call corrcoef(v1(j1,:,1),v1(j1,:,j3),N2,rho)
      g2(j3)=g2(j3)+rho/dfloat(N1)
      call corrcoef(v2(j1,:,1),v2(j1,:,j3),N2,rho)
      g3(j3)=g3(j3)+rho/dfloat(N1)
    end do
  end do
  open(51,file='fw.dat',access='append')
  write(51,111) f1
  close(51)
  open(51,file='gu3.dat',access='append')
  write(51,111) g2
  close(51)
  open(51,file='gv3.dat',access='append')
  write(51,111) g3
  close(51)
endif
111 format(4096(E14.6E2))
if(write_screen.eq.1)then
if(write_vel+write_cor+write_var.ne.0)then
write(*,100) '| Data has been written to files                          |'
endif
endif
!****************************************************************************
! Do diagnogstics and write to screen
!****************************************************************************
if(write_var+write_screen.ne.0)then
  if(div_cor.eq.1)then
  s1_2=sum((v1(:,:,:))**2d0)/dfloat(N1*N2*N3-1)
  s2_2=sum((v2(:,:,:))**2d0)/dfloat(N1*N2*N3-1)
  s3_2=sum((v3(:,:,:))**2d0)/dfloat(N1*N2*N3-1)
  else
  s1_2=sum((real(dZ(1,:,:,:)))**2d0)/dfloat(N1*N2*N3-1)
  s2_2=sum((real(dZ(2,:,:,:)))**2d0)/dfloat(N1*N2*N3-1)
  s3_2=sum((real(dZ(3,:,:,:)))**2d0)/dfloat(N1*N2*N3-1)
  endif
endif
if(write_var.eq.1)then
open(52,file='s.dat',access='append')
write(52,111) [s1_2,s2_2,s3_2]
close(52)
endif
if(write_screen.eq.1)then
tke=0.5d0*(s1_2+s2_2+s3_2)
! some old variables are 'reused'
k1=sum(real(dZ(1,N1/2,:,:)))/dfloat(N2*N3)
k2=maxval(imag(dZ))
j1=randseed
! Determine computation time
call cpu_time(endtime)
k3=endtime-starttime
! Write to screen
write(*,100) '|-------------------------------------------------------------|'
write(*,100) '| INPUT                      | CHARACTERISTICS                |'
write(*,100) '|----------------------------+--------------------------------|'
write(*,101) '| N1       = ',N1,     '     | Variances of vel. comp.:       |'
write(*,102) '| N2       = ',N2,     '     | VAR1    = ',s1_2,         ' |'
write(*,102) '| N3       = ',N3,     '     | VAR2    = ',s2_2,         ' |'
write(*,103) '| dx1      = ',dx1,        ' | VAR3    = ',s3_2,         ' |'
write(*,104) '| dx2      = ',dx2,        ' | Turbulent kinetic energy:      |'
write(*,103) '| dx3      = ',dx3,        ' | tke     = ',tke,          ' |'
```

```
write(*,104) '| L        = ',L  ,        ' |----------------------------|'
write(*,104) '| alphaeps = ',alphaeps,   ' | CONTROL                    |'
write(*,104) '| Gamma    = ',Gamma,      ' |----------------------------|'
write(*,105) '| randseed = ',j1,     '      | complex = ',k2,           ' |'
write(*,106) '|                           | div     = ',k1,           ' |'
write(*,100) '|-----------------------------------------------------------|'
write(*,107) '| Time elapsed: ',k3,          ' sec                         |'
write(*,100) '+-----------------------------------------------------------+'
endif
!*******************************************************************************
! Formats
!*******************************************************************************
100 format(A60)
101 format(A13,i11,A36)
102 format(A13,i11,A17,F17.6,A2)
103 format(A13,F15.6,A13,F17.6,A2)
104 format(A13,F15.6,A32)
105 format(A13,i11,A17,ES17.6,A2)
106 format(A41,ES17.6,A2)
107 format(A15,F17.6,A28)
! The end!
end program TuGen
!*******************************************************************************
! End program
!*******************************************************************************

!*******************************************************************************
! subroutine corrcoef: Determine correlation coefficient
!*******************************************************************************
subroutine corrcoef(x,y,n,rho)
! Calculate correlation coefficient between vectors x and y
! The algorithm is copy-pasted from Wikipedia but has been translated to fortran
!*******************************************************************************
! Variables
!*******************************************************************************
implicit none
! inputs:
integer n
real(8) x(n),y(n)
! output:
real(8) rho
! auxiliary
integer i
real(8) sum_sq_x, sum_sq_y, sum_coproduct, mean_x, mean_y
real(8) sweep, delta_x, delta_y, pop_sd_x, pop_sd_y, cov_x_y
!*******************************************************************************
! Calculations
!*******************************************************************************
sum_sq_x=0d0
sum_sq_y=0d0
sum_coproduct=0d0
mean_x=x(1)
mean_y=y(1)
do i=2,n
    sweep=(i-1.0)/i
    delta_x=x(i)-mean_x
    delta_y=y(i)-mean_y
    sum_sq_x=sum_sq_x+delta_x*delta_x*sweep
    sum_sq_y=sum_sq_y+delta_y*delta_y*sweep
    sum_coproduct=sum_coproduct+delta_x*delta_y*sweep
    mean_x=mean_x+delta_x/i
```

```
    mean_y=mean_y+delta_y/i
end do
pop_sd_x=sqrt(sum_sq_x/n)
pop_sd_y=sqrt(sum_sq_y/n)
cov_x_y=sum_coproduct/n
rho=cov_x_y/(pop_sd_x*pop_sd_y)
end subroutine
!*****************************************************************************
! end subroutine
!*****************************************************************************

!*****************************************************************************
! subroutine fourn: do n-dimensional fft
!*****************************************************************************
SUBROUTINE fourn(fourdata,nn,ndim,isign)
!  (C) Copr. 1986-92 Numerical Recipes Software *$3.
!     Translated to f90-format by
!     Lasse Gilling, Aalborg University.
!     April 16, 2008
!*****************************************************************************
! Variables
!*****************************************************************************
implicit none
integer isign,ndim,nn(ndim)
real(4) fourdata(*)
integer i1,i2,i2rev,i3,i3rev,ibit,idim,ifp1,ifp2,ip1,ip2,ip3
integer k1,k2,n,nprev,nrem,ntot
real tempi,tempr
real(8) theta,wi,wpi,wpr,wr,wtemp
!*****************************************************************************
! Calculations
!*****************************************************************************
ntot=1
do 11 idim=1,ndim
  ntot=ntot*nn(idim)
11 continue
nprev=1
do 18 idim=1,ndim
  n=nn(idim)
  nrem=ntot/(n*nprev)
  ip1=2*nprev
  ip2=ip1*n
  ip3=ip2*nrem
  i2rev=1
   do 14 i2=1,ip2,ip1
    if(i2.lt.i2rev)then
      do 13 i1=i2,i2+ip1-2,2
        do 12 i3=i1,ip3,ip2
          i3rev=i2rev+i3-i2
          tempr=fourdata(i3)
          tempi=fourdata(i3+1)
          fourdata(i3)=fourdata(i3rev)
          fourdata(i3+1)=fourdata(i3rev+1)
          fourdata(i3rev)=tempr
          fourdata(i3rev+1)=tempi
12      continue
13    continue
    endif
    ibit=ip2/2
1   if ((ibit.ge.ip1).and.(i2rev.gt.ibit)) then
      i2rev=i2rev-ibit
```

```
      ibit=ibit/2
    goto 1
    endif
    i2rev=i2rev+ibit
14 continue
  ifp1=ip1
2 if(ifp1.lt.ip2)then
    ifp2=2*ifp1
    theta=isign*6.28318530717959d0/(ifp2/ip1)
    wpr=-2.d0*sin(0.5d0*theta)**2
    wpi=sin(theta)
    wr=1.d0
    wi=0.d0
    do 17 i3=1,ifp1,ip1
      do 16 i1=i3,i3+ip1-2,2
        do 15 i2=i1,ip3,ifp2
          k1=i2
          k2=k1+ifp1
          tempr=sngl(wr)*fourdata(k2)-sngl(wi)*fourdata(k2+1)
          tempi=sngl(wr)*fourdata(k2+1)+sngl(wi)*fourdata(k2)
          fourdata(k2)=fourdata(k1)-tempr
          fourdata(k2+1)=fourdata(k1+1)-tempi
          fourdata(k1)=fourdata(k1)+tempr
          fourdata(k1+1)=fourdata(k1+1)+tempi
15      continue
16    continue
      wtemp=wr
      wr=wr*wpr-wi*wpi+wr
      wi=wi*wpr+wtemp*wpi+wi
17  continue
    ifp1=ifp2
  goto 2
  endif
  nprev=n*nprev
18 continue
return
end subroutine
!*****************************************************************************
! end subroutine
!*****************************************************************************

!*****************************************************************************
! Subroutine check2: Check if Nj is on the form 2^n
!*****************************************************************************
subroutine check2(Nj,flag)
implicit none
integer Nj,flag
real(8) N

N=float(Nj)
do while (N.gt.1d0)
  N=N/2d0
enddo
if(N.eq.1d0)then
  flag=1
else
  flag=0
endif
end subroutine
!*****************************************************************************
! end subroutine
```

```
!******************************************************************************

!******************************************************************************
! SUBROUTINE: div_correction, correct divergence to zero (in the cds2 scheme)
!******************************************************************************
subroutine div_correction(v1,v2,v3,N1,N2,N3,dx1,dx2,dx3)
implicit none
! input
integer N1,N2,N3
real(8) v1(N1,N2,N3),v2(N1,N2,N3),v3(N1,N2,N3)
real(8) dx1,dx2,dx3
! auxiliary
real(8) p(N1*N2*N3),divvec(N1*N2*N3),dP(N1,N2,N3)
integer m,j1,j2,j3
integer j1p,j1m,j2p,j2m,j3p,j3m
!******************************************************************************
! Remove the divergence from the velocity field
!******************************************************************************
m=0
do j3=1,N3
 do j2=1,N2
  do j1=1,N1
   if(j1.eq.1)  then; j1p=2;    j1m=N1
   elseif(j1<N1)then; j1p=j1+1; j1m=j1-1
   else;              j1m=N1-1; j1p=1
   end if
   if(j2.eq.1)  then; j2p=2;    j2m=N2
   elseif(j2<N2)then; j2p=j2+1; j2m=j2-1
   else;              j2m=N2-1; j2p=1
   end if
   if(j3.eq.1)  then; j3p=2;    j3m=N3
   elseif(j3<N3)then; j3p=j3+1; j3m=j3-1
   else;              j3m=N3-1; j3p=1
   end if
   m=m+1
   divvec(m)=(v1(j1p,j2,j3)-v1(j1m,j2,j3))/2d0/dx1 &
            +(v2(j1,j2p,j3)-v2(j1,j2m,j3))/2d0/dx2 &
            +(v3(j1,j2,j3p)-v3(j1,j2,j3m))/2d0/dx3
   end do
 end do
end do
write(*,100) ' Starting the iterative procedure to correct the divergence '
print*, '-----------------------------------'
call minres(p,divvec,n1,n2,n3,dx1,dx2,dx3)
print*, '-----------------------------------'
m=0
do j3=1,N3
 do j2=1,N2
  do j1=1,N1
   m=m+1
   dP(j1,j2,j3)=p(m)
  end do
 end do
end do
m=0
do j3=1,N3
 do j2=1,N2
  do j1=1,N1
   if(j1.eq.1)then;   j1p=2;    j1m=N1
   elseif(j1<N1)then; j1p=j1+1; j1m=j1-1
   else;              j1m=N1-1; j1p=1
```

```
   end if
   if(j2.eq.1)then;   j2p=2;    j2m=N2
   elseif(j2<N2)then; j2p=j2+1; j2m=j2-1
   else;              j2m=N2-1; j2p=1
   end if
   if(j3.eq.1)then;   j3p=2;    j3m=N3
   elseif(j3<N3)then; j3p=j3+1; j3m=j3-1
   else;              j3m=N3-1; j3p=1
   end if
   m=m+1
   v1(j1,j2,j3)=v1(j1,j2,j3)-(dP(j1p,j2,j3)-dP(j1m,j2,j3))/2d0/dx1
   v2(j1,j2,j3)=v2(j1,j2,j3)-(dP(j1,j2p,j3)-dP(j1,j2m,j3))/2d0/dx2
   v3(j1,j2,j3)=v3(j1,j2,j3)-(dP(j1,j2,j3p)-dP(j1,j2,j3m))/2d0/dx3
  end do
 end do
end do
write(*,100) ' Divergence corrected succesfully                          '
100 format(A60)
end subroutine
!*****************************************************************************
! End Subroutine
!*****************************************************************************

!*****************************************************************************
! SUBROUTINE: minres,  solve linear system by minres algorithm
!*****************************************************************************
subroutine minres(x,b,n1,n2,n3,dx1,dx2,dx3)
implicit none
!*****************************************************************************
! Variables
!*****************************************************************************
! input
integer n1, n2, n3
real(8) dx1, dx2, dx3
real(8) b(n1*n2*n3)
! output
real(8) x(n1*n2*n3)
! auxiliary
integer k
real(8) r(n1*n2*n3), p(n1*n2*n3), Ap(n1*n2*n3), Ar(n1*n2*n3)
integer M
real(8) alpha, beta, rArnew, rArold, maxtol
!*****************************************************************************
! The "matrix-vector multiplication"
!*****************************************************************************
M=n1*n2*n3
x=0d0
r=b
p=r
call Amatvec(p,n1,n2,n3,dx1,dx2,dx3,Ap)
rArold=dot_product(r,Ap)
maxtol=abs(rArold)*1d-12
do k=1,M
    alpha=rArold/dot_product(Ap,Ap)
    x=x+alpha*p
    r=r-alpha*Ap
    call Amatvec(r,n1,n2,n3,dx1,dx2,dx3,Ar)
    rArnew=dot_product(r,Ar)
    beta=rArnew/rArold
    rArold=rArnew
    Ap=Ar+beta*Ap
```

```
    p=r+beta*p
    print*, 'Iteration nr.: ', k, ' Residual: ', rArold
    if (abs(rArold)<maxtol) then
        exit
    end if
end do
100 format(A60)
end subroutine
!*****************************************************************************
! End Subroutine
!*****************************************************************************

!*****************************************************************************
! SUBROUTINE: Amatvec,  Matrix-vector-multiplication
!*****************************************************************************
subroutine Amatvec(x,n1,n2,n3,dx1,dx2,dx3,b)
implicit none
!*****************************************************************************
! Variables
!*****************************************************************************
! input
integer n1,n2,n3
real(8) dx1,dx2,dx3
real(8) x(n1*n2*n3)
! output
real(8) b(n1*n2*n3)
! auxiliary
integer i
real(8) f1,f2,f3
integer m,n1n2
!*****************************************************************************
! The "matrix-vector multiplication" b=A*x   (by using "black magic")
!*****************************************************************************
n1n2=n1*n2
M=n1n2*n3
f1=0.25d0/dx1**2
f2=0.25d0/dx2**2
f3=0.25d0/dx3**2
! (1)
b=-2d0*(f1+f2+f3)*x
! (2)
b([1:2*n1n2])=b([1:2*n1n2])+f3*x((M-2*n1n2+1):M)
! (12)
b([(2*n1n2+1):M])=b([(2*n1n2+1):M])+f3*x([1:(M-2*n1n2)])
! (3)
b([1:(M-2*n1n2)])=b([1:(M-2*n1n2)])+f3*x([(2*n1n2+1):M])
! (13)
b([(M-2*n1n2+1):M])=b([(M-2*n1n2+1):M])+f3*x([1:2*n1n2])
do i=0,(n3-1)
    ! (4)
    b([1:2*n1]+i*n1n2)=b([1:2*n1]+i*n1n2)+f2*x([(n1n2-2*n1+1):n1n2]+n1n2*i)
    ! (10)
    b([(2*n1+1):n1n2]+i*n1n2)=&
                      b([(2*n1+1):n1n2]+i*n1n2)+f2*x([1:(n1n2-2*n1)]+i*n1n2)
    ! (5)
    b([1:(n1n2-2*n1)]+i*n1n2)=&
                      b([1:(n1n2-2*n1)]+i*n1n2)+f2*x([(2*n1+1):n1n2]+i*n1n2)
    ! (11)
    b([(n1n2-2*n1+1):n1n2]+i*n1n2)=&
                      b([(n1n2-2*n1+1):n1n2]+i*n1n2)+f2*x([1:2*n1]+i*n1n2)
end do
```

```
do i=0,(n2*n3-1)
    ! (6)
    b([1:2]+i*n1)=b([1:2]+i*n1)+f1*x([(n1-1):n1]+i*n1)
    ! (8)
    b([3:n1]+i*n1)=b([3:n1]+i*n1)+f1*x([1:(n1-2)]+i*n1)
    ! (7)
    b([1:(n1-2)]+i*n1)=b([1:(n1-2)]+i*n1)+f1*x([3:n1]+i*n1)
    ! (9)
    b([(n1-1):n1]+i*n1)=b([(n1-1):n1]+i*n1)+f1*x([1:2]+i*n1)
end do
end subroutine
!*****************************************************************************
! End Subroutine
!*****************************************************************************
```

Paper I

The Effect of Resolved Turbulence in Detached-Eddy Simulations of an Airfoil

Lasse Gilling[1], Niels N. Sørensen[2], and Lars Davidson[3]

Submitted to *Wind Energy*.

[1] Aalborg University, Department of Civil Engineering, Aalborg, Denmark
[2] Risø-DTU National Laboratory for Sustainable Energy, Wind Energy Division, Roskilde, Denmark
[3] Chalmers University of Technology, Division of Fluid Mechanics, Gothenburg, Sweden

The Effect of Resolved Turbulence in Detached-Eddy Simulations of an Airfoil

Lasse Gilling*, Niels N. Sørensen† and Lars Davidson‡

Abstract

The effect of resolving inflow turbulence in detached-eddy simulations of airfoil flows is studied. Synthetic turbulence is generated and run through a precursor simulation before it is imposed at the inflow boundary condition. The generated turbulence fields are shown to decay according to experimental data as they are convected through the domain with the free stream velocity. The subsonic flow around an NACA 0015 airfoil is studied at Reynolds number 1.6×10^6 and at various angles of attack before and after stall. Simulations with turbulent inflow are compared to experiments and to simulations without turbulent inflow. The results show that the flow is insensitive to the resolved turbulence. This indicates that it may be necessary to run full LES or DNS to capture the effect of resolved free stream turbulence in CFD simulations.

1 Introduction

Wind turbines operate in the atmospheric boundary layer and are consequently subjected to the effects of inflow turbulence. Especially, wind turbines located in a wind farm experience high turbulence intensities.

McKeough [1] and Hoffmann [2] both studied the effect of inflow turbulence on the lift and drag of an NACA 0015 airfoil. The Reynolds number in their experiments was about 250.000 which is quite low compared to a modern wind turbine which typically operates at Reynolds numbers around 3 to 6 millions. They both found that the main effect of the turbulence was to postpone stall and extend the linear range of the lift curve. Recently, LM Glasfiber and Risø-DTU have carried out a series of experiments with the NACA 0015 airfoil in the LM wind tunnel [3] at Reynolds numbers between 1.6 and 6.0

*Aalborg University, Department of Civil Engineering, Sohngaardsholsvej 57, 9000 Aalborg, Denmark. E-mail: lg@civil.aau.dk, Phone: +45 9940 8544

†Risø-DTU National Laboratory for Sustainable Energy, Wind Energy Division, Frederiksborgvej 399, P.O. Box 49, 4000 Roskilde, Denmark

‡Chalmers University of Technology, Division of Fluid Dynamics, Department of Applied Mechanics, Hörsalsvägen 7A, 412 96 Gothenburg, Sweden

million. The experiments and obtained data are described in [4, 5, 6, 7] and basically they find the same effect as Hoffmann and McKeough.

For wind turbines operating close to stall a better estimation of the maximum lift is very useful. This paper investigates the possibility of modeling the effect of turbulence on airfoils by resolving the inflow turbulence in computational fluid dynamics (CFD) simulations.

Resolving the turbulence in CFD simulations requires the computational domain to be three-dimensional and an eddy-resolving technique should be applied, i.e. direct numerical simulation (DNS) and large-eddy simulation (LES). At high Reynolds numbers these techniques are extremely computational expensive, c.f. e.g. [8]. The reason for the high cost of LES is that even the largest energy carrying eddies in the boundary layer of the airfoil are very small and consequently the computational mesh must be very fine.

The approach of resolving the eddies in the boundary layer is not affordable for industrial applications. Alternatively, in detached-eddy simulation (DES) the turbulence in the boundary layer is treated by a Reynolds-averaged Navier-Stokes (RANS) turbulence model and the turbulence outside the boundary layer is treated by a LES model.

DES was developed for simulating flows around airplane wings [9]. Since then, the technique has successfully been applied to flows around airfoil sections [10, 11, 12] and entire wind turbine blades [13]. The full history and future challenges of DES are reviewed by Spalart in [14]. One of the major problems with the original formulation from 1997 is the grey area between the LES and RANS regions. If the grid is fine enough to switch to LES mode in the outer part of the boundary layer but not fine enough to resolve the turbulence the Reynolds stresses become too low. The result is premature separation which can be induced by grid refinement. This is clearly not desirable and in 2003 Menter and Kuntz [15] suggested to shield the RANS region by one of the shield functions in the $k-\omega$ SST turbulence model. The problem is further described in [16] where a Delayed DES (DDES) is introduced based on the Spalart-Allmaras turbulence model [17].

DES has also been applied to a number of flows with multiple bluff bodies where the wake from the first body impinges on another bluff body. Thereby, the second bluff body is exposed to inflow turbulence. A number of examples are a multi-element airfoil simulated in [16], the landing gear simulated in [18], the tail of the fighter aircraft in [19], the rod-airfoil configuration in [20] and finally, the two wall-mounted cubes in tandem in [21]. All these examples rely on the DES model to capture the effect of resolved inflow turbulence. This paper examines this ability of DES through an airfoil flow where free stream resolved turbulence impinges on the boundary layer.

Resolved inflow turbulence will be imposed in DES of an NACA 0015 airfoil. The idea is to impose the turbulence at the inlet of the simulation. By using a fine mesh everywhere in the domain the resolved eddies can be convected from the inlet to the airfoil where they will impinge on the leading edge. Inside the boundary layer the flow is treated by a RANS turbulence model, but the unsteady forcing from the LES region will make the turbulence penetrate the RANS region. Here, the resolved turbulence should mix the high momentum of the free flow into the boundary layer and thereby increase the velocity close to the wall. This will increase the resistance against the adverse pressure

gradient on the suction side, and thereby postpone stall and increase the linear range of the lift curve.

Below, the numerical methods are described. Next, the technique for generating the turbulent inflow is presented along with a study of the spatial decay of the generated turbulence. From the NACA 0015 airfoil simulations the overall effect of resolved turbulence is analyzed in terms of lift and drag along with surface pressure and skin friction distribution. Also, profiles of velocity and eddy viscosity are shown prior to a discussion and some concluding remarks.

2 Numerical Methods

The in-house computational fluid dynamics code EllipSys3D [22, 23, 24] is used for the flow simulations. The code is a multiblock finite volume discretization of the incompressible RANS equations in general curvilinear coordinates for structured meshes. The code uses a collocated variable arrangement, and Rhie/Chow interpolation is used to avoid odd/even pressure decoupling. The PISO algorithm is used to enforce pressure/velocity coupling. Solution for momentum is obtained by the fourth order central difference scheme and a second order accurate dual time stepping algorithm is used. The code is parallelized with MPI.

The present version of DES by Strelets [10] uses the two-equation $k-\omega$ SST subgrid model by Menter [25]. The RANS layer close to the airfoil is shielded by the F_2 function of the $k-\omega$ SST model as proposed by Menter and Kuntz [15] to reduce the problem of grid-induced separation. The DES flavor is therefore shielded DES also called DDES.

The laminar-turbulent transition of the boundary layer is modeled by the correlation-based $\gamma - \widetilde{\mathrm{Re}}_\theta$ model by Menter and coworkers [26, 27]. This model introduces two additional transport equations, one for intermittency and one for local transition onset momentum thickness Reynolds number. The model contains a number of empirical correlation functions, but two of these are not given in the original papers for proprietary reasons. However, in Sørensen [28] these two empirical correlation functions are determined.

3 Generation of Synthetic Turbulent Inflow

This section describes the chosen method to generate the inflow turbulence synthetically. The turbulence imposed on the inflow boundary condition should have the correct moments and spectra and appropriate phase angle between modes. The turbulence will be imposed on a domain without walls upstream of the airfoil and therefore a wind field with the correct moments and spectra can be generated synthetically from a theoretical spectrum. The applied method by Mann [29] is described briefly below. It only gives correct moments up to second order, and further, the phase information is chosen randomly. Therefore, a precursor simulation is run to let the flow solver correct these deficiencies.

3.1 The Mann Method of Generation Synthetic Turbulence

An entire family of methods originates from the paper by Shinozuka and Jan [30], including the method of Mann [29] which is used here. For a full description including all equations interested readers are referred to [29, 31, 32]. The method was designed for generating a turbulent wind field for engineering load calculations. It is based on three-dimensional FFT of a set of wave number vectors determined to give the correct covariance. This results in a three-dimensional and divergence free frozen velocity field with the correct spatial correlation. It is converted to a time series of inflow boundary conditions by application of Taylor's hypothesis. All three components of the velocity vectors are generated. This is advantageous, compared to the option of only generating the streamwise components. This will be demonstrated in the following section.

3.2 Demand for Three-Dimensional Velocity Fields

The airfoil should experience homogenous, isotropic turbulence. Some techniques for generating turbulent inflow only generate the streamwise velocity component and let the two remaining components be generated by the flow solver. This approach requires a long upstream domain or a long precursor simulation to allow the velocity field to become isotropic. If all three components are generated synthetically and imposed on the inflow boundary the problem diminishes as shown below. In Figure 1 the isotropy of the velocity field is compared for the two methods.

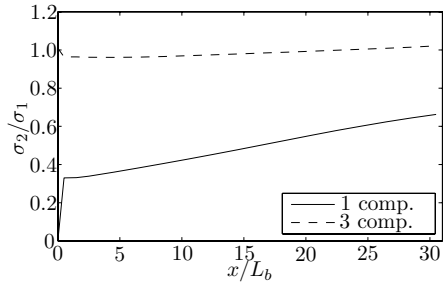


Figure 1: Development of the ratio between the standard deviation of a transverse velocity component and the streamwise velocity component. One simulation has all three velocity components in the inflow, whereas the other only has the streamwise component. x is the streamwise coordinate measured from the inflow boundary, and L_b is the width and height of the domain.

The results presented in Figure 1 are generated by two simulations with identical computational domains, but different inflow conditions. The domain has $1024 \times 32 \times 32$ cubic cells and periodic boundary conditions in the transverse directions. The two sets of inflow conditions contain the same kinetic energy. One contains three components with the energy evenly distributed, whereas the other contains all the energy in the streamwise

component.

As seen from the figure, when the inflow boundary condition only includes one component, the turbulence does not become isotropic in this domain, even though it has a considerable extent in the streamwise direction. In conclusion the inflow boundary condition should contain all three components of the turbulence to minimize the required length of the precursor domain.

3.3 Application: Spatially Decaying Homogenous Turbulence

Spatial decay of homogenous turbulence is studied to test the synthetic turbulence and the numerical method. The computational domain is illustrated in Figure 2, where the first fifth in the streamwise direction is shown. The domain consists of $1280\times64\times64$ cubic cells. The mean velocity is constant and parallel to the x_1 axis. Synthetic turbulence is superimposed on to the constant mean velocity and set as inflow boundary condition. Below, the development of resolved turbulent kinetic energy and the energy spectrum are studied.

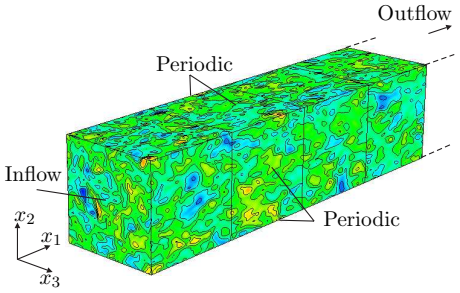


Figure 2: The first fifth of the precursor domain. The boundaries are colored by the streamwise velocity component.

When the synthetic inflow boundary condition is imposed on a computational domain, the turbulence does not decay physically in the first part of the domain. There are two reasons for this. One is that the appropriate values of the subgrid model parameters (here k and ω) are not known, and therefore, uniform constant values are imposed on the boundary. The other reason is that the synthetic turbulence does not possess the correct phase information, as the phase angles are chosen randomly. During the precursor simulation these flaws are corrected by the flow solver [33].

In Figure 3 the decay of resolved turbulent kinetic energy is shown and compared to the empirical expression [34]

$$k(t) = \frac{1}{2}U_0^2\left(\frac{1}{21}+\frac{2}{20}\right)\left(\frac{U_0 t}{M}-3.5\right)^{-1.25} \tag{1}$$

where U_0 is a reference velocity and M is the mesh spacing in the grid used to generate

the turbulence. The numerical constants have been fitted to match the experimental data.

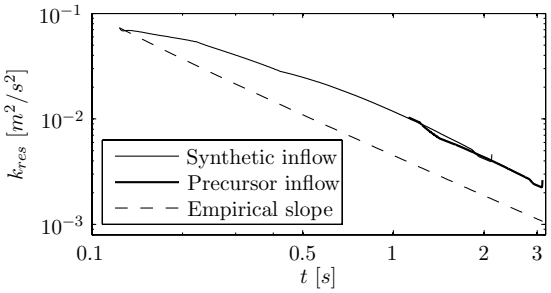


Figure 3: Decay of resolved turbulent kinetic energy.

To show the correct rate of decay the simulations must produce curves with the same slope as the empirical line. The decay is shown for two cases. One with synthetic inflow turbulence and one with precursor turbulence. The synthetic turbulence contains the same resolved kinetic energy and has the same integral length scale as the experiment. At some x_1 station the velocities and subgrid parameters are sampled in the x_2-x_3-plane and the time series of samples are stored in a file. These time series are used as time dependent inflow boundary condition for case of precursor turbulence.

When the synthetic turbulence is used for the inflow boundary condition the decay is initially too slow, which causes the curve for the synthetic turbulence to lie above the line from the empirical expression. This is due to a too low eddy viscosity on the inflow boundary. It is observed that the case with precursor turbulence behaves differently from case with synthetic turbulence. The curve starts at the point where its inflow turbulence is sampled. From there it makes a kink before it starts to decay with the correct rate as the curve is parallel to the empirical slope.

The velocities and subgrid parameters are set on the inflow boundary, so the only variable which is not identical to the values in the precursor simulation is the pressure. The incorrect pressure may cause the kink in the kinetic energy decay seen in the two last cases. Close to the outlet the outflow boundary condition causes the decay to be incorrect and another kink can be seen on both curves.

The energy spectra from time series sampled at different locations in the two simulations are shown in Figure 4. By Taylor's hypothesis the sample positions are converted to instances as in Figure 3. Spalart [35] argues that no eddies spanning less than five cells can be resolved as it is convected with the mean velocity. Thereby, only a short range of frequencies can be resolved due to the dimensions of the computational domain. The width of the domain sets the lower limit and the size of the cells set the upper one. The domain consists of only 64 cells in the transverse directions, and as seen in the figure it can only resolve a decade of frequencies. It can be seen that the largest eddies in the inertial subrange are resolved as the slopes of the spectra computed from the simulations

are $-5/3$ for high frequencies.

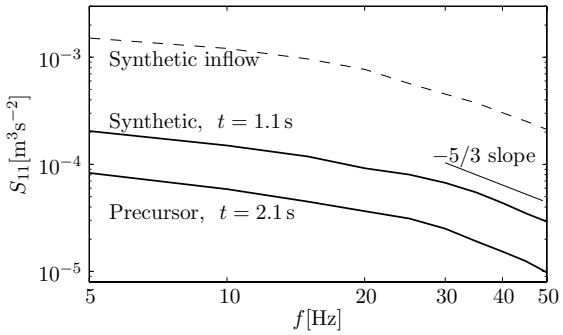


Figure 4: Power spectra of turbulence sampled at two positions in the streamwise direction in the simulations using synthetic and precursor turbulence. For comparison the power spectrum of the synthetic field is shown with the dashed line.

4 Description of Experimental Data

Two sets of experimental data are used for comparison and in this section the experimental configurations are described. The first set of data is from the LM Wind Tunnel and the second is from Virginia Tech.

The test section of the LM wind tunnel has a cross section of $2.7 \times 1.35\,\mathrm{m}$ and the airfoil chord is 0.9 m giving a span of 1.5 chords. The maximum attainable wind speed is 105 m/s but the data used here is from 26.7 m/s. This gives a chord based Reynolds number of 1.6×10^6. The lift and surface pressure used here are based on the measured surface pressure. For low angles of attack the drag is based on velocities measured by a wake-rake and for high angles of attack on measured surface pressure. Three different setups were investigated. One without a turbulence grid and two setups with different grids. The grids were placed about three chords upstream of the leading edge of the airfoil. They were of bi-planar type and consisted of rectangular metal plates with a thickness in the cross-stream direction of 3 mm. The first grid (grid 1) had a mesh spacing of 200 mm giving a solidity of 0.03. The second grid (grid 2) had a mesh spacing of 100 mm giving a solidity of 0.06. The turbulence characteristics are investigated in [6] and summed up in Table 1. For further descriptions of the wind tunnel and the experiment see [3, 4, 5, 7].

The test section of the Virginia Tech Low Stability Wind Tunnel has a square cross section with side length 1.83 m. The chord of the airfoil model is 0.61 m giving a span of three chords. The data used here is from an experiment with 30 m/s free-stream velocity and thereby a chord based Reynolds number of 1.2×10^6. The turbulence grid is bi-planar and has a mesh size of 305 mm and solidity of 0.31. It was placed 9.5 chords upstream of the leading edge in the contraction of the wind tunnel. This improves the isotropy of the

turbulence as investigated by [34]. The experiment is further described in [36, 37]. The characteristics of the turbulence are given in Table 1.

For comparison the characteristics of the resolved turbulence in the simulations are also included in Table 1. The turbulence in the simulations is further described in Section 5.2.

Wind Tunnel	Re	Grid	Tu	Length Scale
LM	1.6×10^6	No grid	0.4%	$0.008c$
LM	1.6×10^6	Grid 1: $M/c = 0.22$	1.5%	$0.017c$
LM	1.6×10^6	Grid 2: $M/c = 0.11$	1.2%	$0.017c$
Virginia Tech	1.2×10^6	$M/c = 0.50$	3.9%	$0.13c$
Simulations	1.6×10^6	–	$0.0\% - 4.0\%$	$0.25c$

Table 1: Experimental conditions. M is mesh size and c is the airfoil chord. Tu is the turbulence intensity.

5 Computational Setup

The mesh, boundary conditions and the inflow turbulence will be described below.

5.1 Computational Domain

The geometry of the domain is chosen to match the geometry of the test section of the LM Wind Tunnel. The mesh is shown in Figures 5 and 6. It is a two dimensional mesh which has been extruded in the spanwise direction to make a three dimensional mesh. It consists of an O-mesh near the airfoil and Cartesian blocks to fill the rest of the domain to match the geometry of the wind tunnel. The Cartesian grid in large parts of the domain is favorable for the energy conservation. The O-mesh consists of 384×64 cells in the wall-parallel and wall-normal directions, respectively. The spanwise direction is resolved by 128 cells of constant width. The mesh consists of a total of 21 million cells.

Symmetry boundary conditions are used on the top and bottom of the domain to limit the flow without having to resolve a wall boundary layer. For the same reason periodic boundary conditions are imposed on the spanwise boundaries. To investigate the influence of periodicity vs. symmetry on the spanwise boundaries a few simulations have been run with symmetry boundary conditions. The conclusion from this analysis is that the difference in results is negligible. Therefore, all results presented in this paper use periodic boundary conditions in the spanwise direction.

Velocities, k and ω taken from a precursor simulation are used as time dependant inflow boundary condition. On the outflow boundary the gradient is assumed to be zero normal to the outlet, except for pressure which uses a second order extrapolation normal to the boundary.

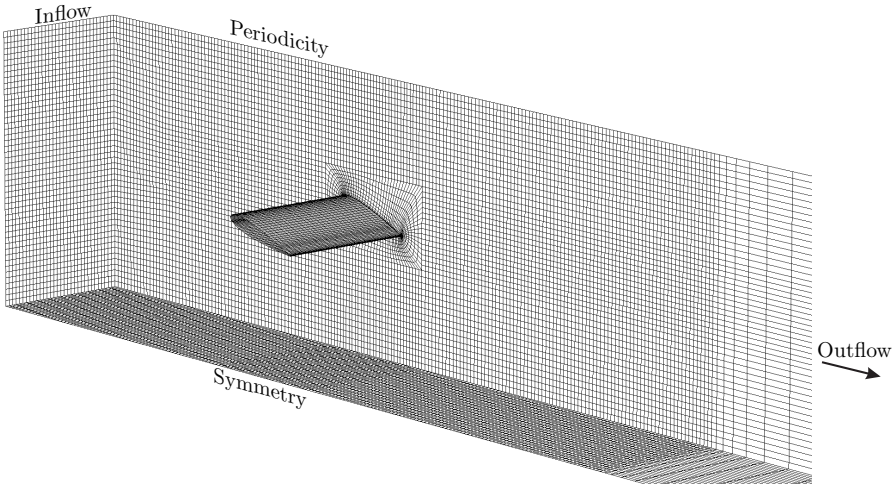


Figure 5: Three-dimensional view of grid. Only every fourth mesh point is shown. The angle of attack in the shown mesh is 15^o.

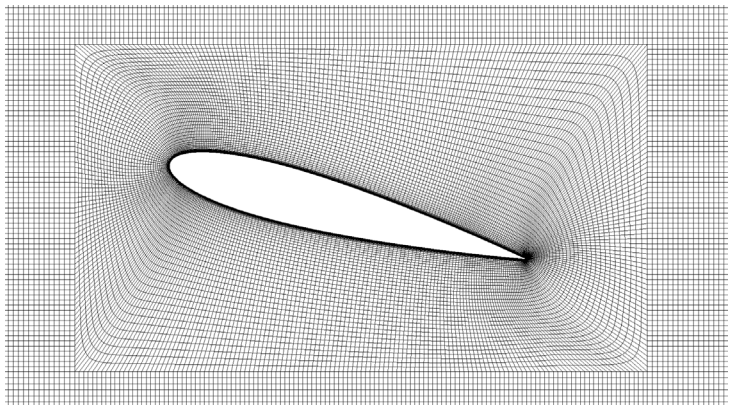


Figure 6: Close-up of the two dimensional O-mesh and some neighboring Cartesian cells. The angle of attack in the shown mesh is 15^o.

As in the experiment in the LM Wind Tunnel the spanwise extent of the domain is 1.5 chords and the height is 3 chords. The inflow boundary is located 4 chords upstream and the outflow boundary is located approximately 25 chords downstream of the airfoil. The last 64 cells in the streamwise direction are stretched to suppress resolved turbulence prior to the outflow boundary. The stretching ratio is 5%.

The size of the cells in the Cartesian mesh (excluding the stretching region) is $\Delta x \times \Delta y \times \Delta z \approx 1.4 \times 1.6 \times 1.2$ when measured in percent of the chord. x, y and z denote the streamwise, vertical and spanwise directions, respectively. In Figure 7 the sizes of the cells closest to the airfoil wall are shown. The mesh spacing has been normalized to wall units for the 16^o angle of attack (AOA) flow. Δn^+ and Δs^+ are mesh spacings in the wall-normal and wall-parallel directions, respectively. Δz is constant but Δz^+ varies due to changes in the friction velocity. The ratio of Δn^+ and Δz^+ is nearly constant, as Δn is nearly constant. The jump close to $s/s_{max} = 0.5$ is caused by the transition from laminar to turbulent boundary layer on the suction side close to the leading edge.

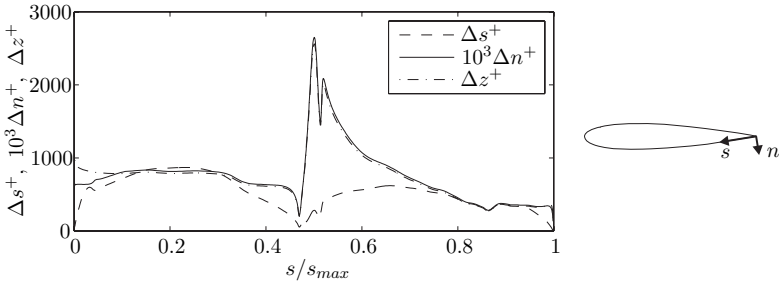


Figure 7: Left: Length, height and width of the first layer of cells measured in wall units. Right: Definition of coordinates s and n.

In Sørensen [38] the mesh dependence was investigated for a rotor simulation using the QUICK scheme in the EllipSys3D code. Decent mesh independence was shown for a mesh of similar resolution in the n and s directions and a much coarser resolution along the spanwise direction.

5.2 Inflow Boundary Condition

From Table 1 it is seen that the length scale of the turbulence in the LM Wind Tunnel is very low. In order to resolve an eddy being convected with the free stream a resolution with at least five cells per eddy diameter is required [35]. To resolve eddies with a diameter on the order of the integral length scale as in the LM experiment (0.017 chords) about 300 cells are required per chord. The grid described above has about 100 cells per chord and consequently about 27 times more cells would be needed to resolve the largest eddies as in the experiment. Further, the time step should also be decreased to one third. The computational burden of this resolution is not affordable. Instead the much more affordable length scale of 0.25 chords is chosen. With this length scale the resolution is 16 cells per length scale.

The intensity of the inflow turbulence is varied by scaling the velocities and subgrid scale model parameters, which have been sampled from a precursor simulation. Let u_0', v_0', w_0', k_0' and ω_0' denote the values sampled from a precursor simulation. The velocities are scaled by a common factor, a, to set a given intensity of the resolved inflow turbulence.

$k_0' \propto (u_0')^2$ is scaled by a^2 and $\omega_0' \propto \sqrt{k}/L$ is scaled by a. Hence, the length scales of both the resolved and subgrid turbulence are constant for all the applied inflow turbulence fields. The resolved turbulence has an integral length scale of approximately 0.25 chords. The subgrid kinetic energy k_{sgs} is about one fourth of the resolved kinetic energy k_{res}.

The imposed eddy viscosity scales linearly with the factor a. It ranges from about $0.7\,\nu_{mol}$ for a resolved turbulence intensity (Tu) of 0.1% to $28\,\nu_{mol}$ for 4.0% Tu. ν_{mol} is the molecular viscosity. In the simulations without resolved inflow turbulence constant values of k and ω are set on the inflow boundary to give an eddy viscosity of about $10^{-3}\,\nu_{mol}$ just upstream of the airfoil.

The duration of the time series sampled from the precursor simulation corresponds to 9 chord passages. To run simulations longer than this the same inflow field is simply imposed a number of times.

6 Results

In this section the simulations are compared to the experimental data. Initially, the time averaged values of lift and drag are presented. Then, the resolved turbulence in the RANS layer close to the airfoil is investigated. Next, some details of the velocity profiles close to the airfoil are compared for different turbulence intensities and finally, the effect on surface pressure and skin friction is shown.

6.1 Lift and Drag

In Figure 8 the computed lift and drag coefficients are compared to the experimental data. In the experiment by Mish [36, 37] in the Virginia Tech Low Stability Wind Tunnel the drag coefficient was not measured. In the experiment in the LM Wind Tunnel both positive and negative AOA were tested. As the airfoil is symmetric, the data from negative AOA have been mirrored to positive AOA to give an impression of the accuracy of the experimental data. A small lack of symmetry in the measurements can be seen from the differences in lift coefficient measured at positive and negative AOA. The inaccuracies arise because at this low velocity the dynamic pressure is lower than what the pressure transducers were calibrated for. The wind tunnel of LM Glasfiber has been optimized for Reynolds numbers in the range 3–6 million and in this range the results are indeed symmetric [3, 5].

The effect of the turbulence is clearly seen in the results from the LM wind tunnel. At low angles of attack there are no changes due to the turbulence, but the linear range of the lift curve is extended and stall is postponed from 15° to 18°. The result is a significant increase in maximum lift. The turbulence characteristics from the two grids in the LM tunnel are quite similar as seen in Table 1 and they also give the same lift and drag coefficients.

The lift measured by Mish agrees quite well with the lift from LM using turbulence grids, but it appears to give lower maximum lift. Unfortunately, the lift was not measured

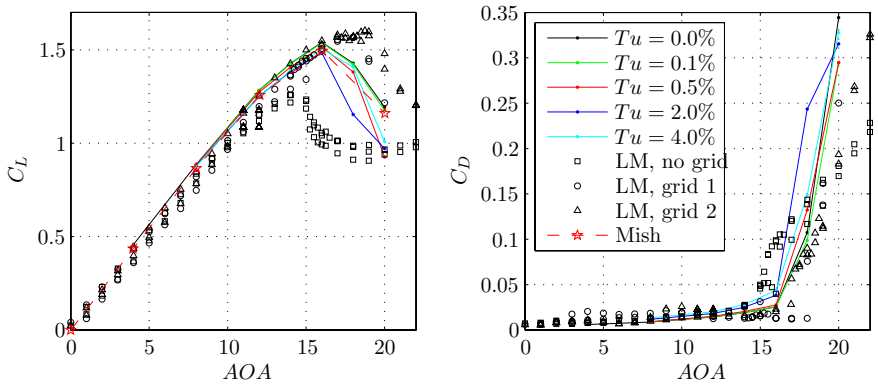


Figure 8: Lift and drag coefficients from simulations and measurements. Turbulence characteristics in the experiments are shown in Table 1.

between 16° and 20° where the data from LM give maximum lift.

The simulated data show very low dependance of the resolved turbulence at angles of attack lower than stall. For the stalled flows the simulations show some scatter, and especially, the simulation at 18° with $Tu = 2.0\%$ gives spuriously low lift and high drag. After stall the flows are very unsteady and give slowly fluctuating lift and drag coefficients as separation bubbles are formed and shed.

For the angles of attack lower than stall the simulations show good agreement with the experimental data. Also the lift measured by Mish at 20° is in very good agreement with the simulations with low turbulence. The drag coefficients are also predicted accurately up to about 18°.

Although, the simulations agree decently with the measurements the lacking influence of the resolved turbulence is discouraging. And, even the small effects that can be observed do not change the results in the same way as in the experiment. In the next subsection it will be demonstrated that the resolved turbulence does penetrate the RANS region where it should increase the momentum in the inner part of the boundary layer and thereby increase the resistance against separation.

6.2 Resolved Turbulence in the RANS Layer

In Figure 9 a) the RANS region is shown in the simulation with $AOA = 16°$ and $Tu = 4.0\%$. It covers the boundary layer. In the simulation the velocities have been sampled at the four points denoted $p_1 - p_4$. The points p_1 and p_2 are located in the RANS region while p_3 and p_4 are located in the LES region of the domain. In the Figures b)-d) the time series of the three velocity components are shown. It is seen that the forcing from the LES region causes the solution in the RANS region to be very unsteady.

The fluctuations in the four points are seen to be highly correlated especially for the v and w components. Further, from Figure 9 b) it is seen that the highest speed up is at point two.

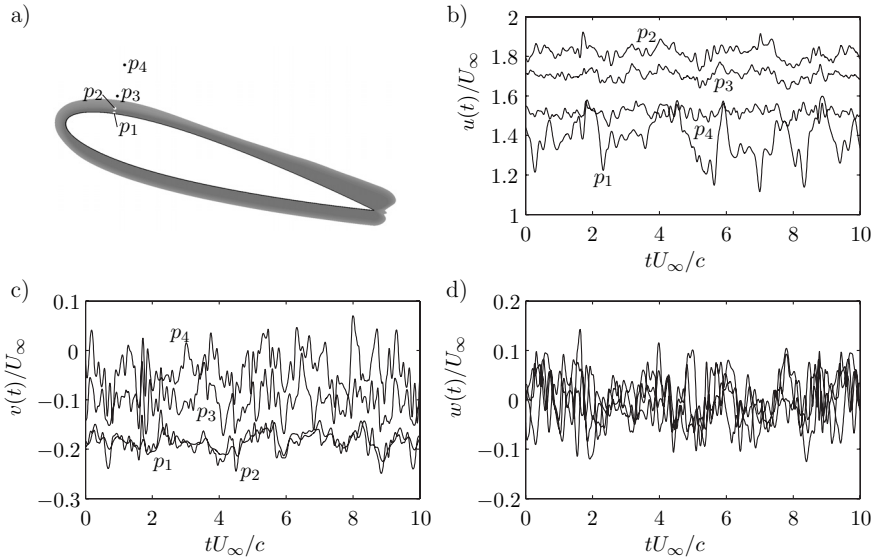


Figure 9: RANS region and time histories of velocities sampled at four points in the simulation with $AOA = 16^o$ and $Tu = 4.0\%$. a) Position of points. The RANS region is shaded by grey. b-d) Time histories of the three velocity components.

Table 2 shows the distance from the wall in wall units and the standard deviations of the sampled time series. It is seen that the standard deviations are largest for the u components close to the wall. Eddies from the free stream are convected and stretched around the leading edge of the profile and these eddies are sampled by the point p_1 located just above the profile. In turn the fluctuations of the v component are limited by the presence of the wall and this gives the low standard deviations of the v component in this point.

Point	n^+	$100\sigma_u/U_\infty$	$100\sigma_v/U_\infty$	$100\sigma_w/U_\infty$
p_1	230	10.4	1.7	5.0
p_2	1700	3.5	2.2	4.6
p_3	5700	2.7	3.1	3.9
p_4	15700	2.5	3.8	3.8

Table 2: Variances of sampled time series plotted in Figure 9

At the inlet where the isotropic turbulence is imposed the turbulence intensity is

4.0%. The point p_4 gives slightly lower standard deviations of the v and w components with about 3.8% and these values are representative for points far from the influence of the airfoil. The turbulence is expected to decay slightly from the inlet to the airfoil. In p_4 the u component is still heavily influenced by the presence of the airfoil which causes the standard deviation to be considerably lower than the values far from the airfoil.

6.3 Mean Velocity Profiles in the Boundary Layer

The resolved turbulence in the boundary layer should increase the velocity in the inner region of the boundary layer. In Figure 10 the velocity profiles are shown for three stations along the suction side of the airfoil. The first is located close to the leading edge at $x/c = 0.1$. Here a large speed up of the flow is observed and the peak velocity is twice the free stream velocity. In the inner part of the boundary layer the velocity is increased slightly and here the velocity is 6% higher with a resolved turbulence intensity of 4%. At $n^+ = 100$ the velocity increases faster without resolved turbulence and from there the velocity is highest for the flow without resolved turbulence.

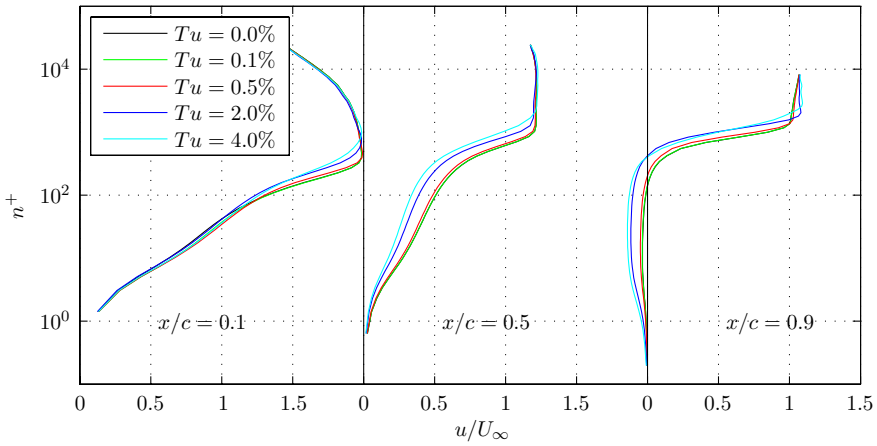


Figure 10: Velocity profiles at three positions along the suction side of the airfoil for $AOA = 16^o$.

The second velocity profile is from the middle of the airfoil at $x/c = 0.5$. Here it is seen that in the entire boundary layer the velocity is largest in the flow without resolved turbulence. The higher the free stream turbulence the lower velocity in the boundary layer. This is exactly the opposite effect as desired.

Finally, the velocity profiles are shown for the station at $x/c = 0.9$. Here the flow is separated for all turbulence intensities, and it is seen that the adverse flow is largest for high resolved turbulence intensities.

Figure 11 shows the viscosity profiles at the same stations as in Figure 10. At the first two stations the viscosity in the inner part of the boundary layer is unaffected by the free stream turbulence. This is due to the transition model which builds up viscosity in the boundary layer independent from the free stream values. At the outer edge of the boundary layer the eddy viscosity is influenced by the free stream values and here the values are dependent on the resolved turbulence. At $x/c = 0.1$ the peak value for the high turbulence flow is close to $135\nu_{mol}$ which is far higher than the free stream of $28\nu_{mol}$. From the leading edge the eddy viscosity increases along the airfoil and in the separated from at $x/c = 0.9$ high values are reached.

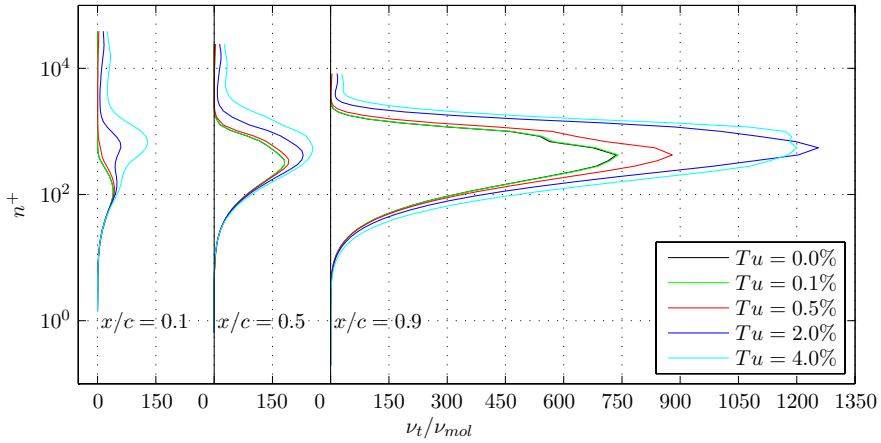


Figure 11: Viscosity profiles at three positions along the suction side of the airfoil for $AOA = 16^o$.

6.4 Surface Pressure Distribution and Skin Friction

For 16^o angle of attack the time averaged surface pressure distribution is shown in Figure 12. The simulation results are only shown for some turbulence intensities, but the differences are very small. The three lines coincide everywhere except at the suction peak and close to the trailing edge. The suction peak is slightly lower with high resolved turbulence but the differences are small which was also reflected in the small differences in lift and drag. The pressure distributions from the simulations agree perfectly with the result from experiment with turbulence grid in the LM wind tunnel.

The data by Mish also agree nicely with the LM and simulated results. A single transducer on the suction side close to the leading edge gives a too high pressure (i.e. too low $-c_p$). This is also the case for the other angles of attack which indicates a problem with that pressure transducer. The pressure distribution from the experiment without grid in the LM tunnel shows a much lower suction peak due to a large separated region.

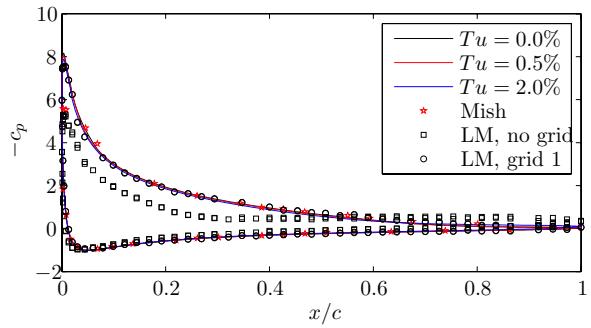


Figure 12: Surface pressure coefficient for $AOA = 16^o$.

In Figure 13 the skin friction coefficients from the simulations are plotted for 16^o angle of attack. The absolute values are shown because it is not possible to define the flow direction uniquely on a two-dimensional surface in the three-dimensional simulation. A clear effect of the turbulence is seen. The transition location moves forward as the turbulence intensity is increased. On the pressure side of the airfoil the boundary layer is laminar over the entire airfoil for low turbulence intensities, while transition takes place for the high turbulence simulations. This shows a clear effect of changing the resolved turbulence intensity, but actually, the differences are caused by the corresponding changes in the subgrid kinetic energy.

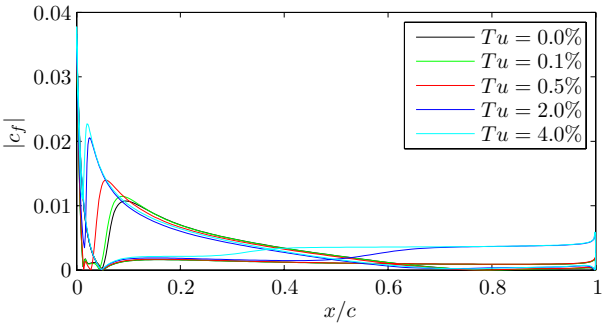


Figure 13: Skin friction coefficient for $AOA = 16^o$.

7 Discussion

The results presented above show that the resolved structures do penetrate the boundary layer. Close to the leading edge a small increase in the velocity in the inner boundary

layer is found, but it is not sustained along the airfoil. Therefore, the desired resistance against adverse pressure gradients diminishes and the maximum lift of the airfoil is not increased with increasing turbulence. Instead a small reduction in the lift is experienced.

In experiments there are two opposite effects of free stream turbulence. One is to increase the momentum inside the boundary layer which will reduce separation and thereby increase the maximum lift. This principle is applied by vortex generators. The second effect is to trigger separation from laminar to turbulent boundary layer which will decrease the lift and increase the drag. Only the latter effect was successfully modeled in the simulations and thereby the obtained lift was reduced by the turbulence.

The computational grid is not fine enough to resolve the fine eddies in the boundary layer and the large eddies from the free stream turbulence did not increase the momentum as desired. To capture the effect it may be necessary to run the simulations as LES and resolve the fine eddies in the boundary layer.

This indicates that DES should be used with precaution for flows where the wake from one bluff body impinges on a second bluff body. There are numerous examples of these flows in the literature and even though the overall results are in good agreement with measurements the underlying physics is not modeled correctly.

In an earlier publication [39] the problem with an airfoil in turbulent inflow was investigated without application of the model for the laminar-turbulent transition of the boundary layer. Here, it was concluded that the flow was highly sensitive to the intensity of the resolved turbulence. Our investigations since that paper was published has made it clear that the observed effects was merely caused by the changes in the subgrid parameters that follow from a change in the resolved turbulence.

8 Conclusions

A technique for generating synthetic turbulence is presented. The generated field must contain all three components of the velocities to minimize the need for a long precursor simulation. It is, however, suggested to run a shorter precursor simulation with the generated turbulence to allow the incorrect phase information to be corrected. Further, the subgrid parameters will then adjust to the velocity field and the computational setup. After the precursor simulation the turbulence can be imposed on the inflow boundary condition and will then decay according to empirical data.

From the study of the airfoil flow it is concluded that the DES technique cannot capture the effect of the resolved inflow turbulence. The measurements show that the effect of increasing the free-stream turbulence is to postpone stall to higher angles of attack, but this effect was not found in the simulations. Instead LES or DNS should be applied, but the computational cost of these methods is prohibitively expensive at industrial Reynolds numbers.

Acknowledgements

This work was funded by "The Danish Council for Strategic Research" under the project "Airfoils in Turbulent Inflow" with project no. 2104-05-0074. The authors gratefully acknowledge the experimental data supplied by LM Glasfiber. The simulations have been carried out on Alfheim - A linux cluster located at DTU and financed by Danish Center for Scientific Computing (DCSC) Grant nr. HDW-0107-02 from January 2007.

References

[1] McKeough PJ. *Effects of Turbulence on Aerofoils at High Incidence*. Ph.D. thesis, University of London, 1976.

[2] Hoffmann JA. Effects of freestream turbulence of the performance characteristics of an airfoil. *AIAA Journal*; 1991, **29**, 1353–1354.

[3] Fuglsang P, Bove S. Wind tunnel testing of airfoils involves more than just wall corrections. *Proceedings of European Wind Energy Conference and Exhibition 2008*; 2008, Brussels, March 31–April 3.

[4] Døssing M. *High frequency microphone measurements for transition detection on airfoils*. Tech. Rep. Risø-R-1645(EN), RISØ National Laboratory - DTU, Roskilde, 2008.

[5] Bove S. *LSWT report of campaign NACA 0015_1*. Tech. rep., LM Glasfiber, Lunderskov, 2008.

[6] Bertagnolio F. *NACA0015 measurements in LM wind tunnel and turbulence generated noise*. Tech. Rep. Risø-R-1657(EN), Risø DTU National Laboratory for Sustainable Energy, Roskilde, 2008.

[7] Bak C, Andersen PB, Madsen HA, Gaunaa M, Fuglsang P, Bove S. Design and verification of airfoils resistant to surface contamination and turbulence intensity. *26th AIAA Applied Aerodynamics Conference. AIAA Paper 2008-7050*, 2008.

[8] Spalart PR. Strategies for turbulence modelling and simulations. *Heat and Fluid Flow*; 2000, **21**, DOI: 10.1016/S0142-727X(00)00007-2, 252–263.

[9] Spalart PR, Jou W-H, Strelets M, Allmaras SR. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. Liu, C. and Liu, Z. (eds.), *1st AFOSR International Conference on DNS/LES*, Columbus OH, 1997, August, Greyden Press, pp. 137–147.

[10] Strelets M. Detached eddy simulation of massively separated flows. *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 8–11, AIAA Paper 2001-0879, 2001.

[11] Shur M, Spalart PR, Strelets M, Travin A. Detached-eddy simulation of an airfoil at high angle of attack. *Engineering Turbulence Modelling and Experiments*; 1999, **4**, 669–678.

[12] Bertagnolio F, Sørensen NN, Johansen J. *Profile catalogue for airfoil sections based on 3d computations.* Tech. Rep. Risø-R-1581(EN), RISØ National Laboratory, Roskilde, 2006.

[13] Johansen J, Sørensen NN, Michelsen JA, Shreck S. Detached eddy simulation of flow around the NREL phase VI blade. *Wind Energy*; 2002, **5**, 185–197.

[14] Spalart FR. Detached-eddy simulation. *Annual Review of Fluid Mechanics*; 2009. DOI: 10.1146/annurev.fluid.010908.165130.

[15] Menter FR and Kuntz M. Adaptation of eddy-viscosity turbulence models to unsteady separated flow behind vehilces. *Proceeding Conference on the Aerodynamics of Heavy Vehicles: Trucks, Busses and Trains*; 2003.

[16] Spalart PR, Deck S, Shur ML, Squires KD, Strelets MK, Travin A. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*; 2006, **20**, DOI: 10.1007/s00162-006-0015-0, 181–195.

[17] Spalart PR and Allmaras SR. A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale*; 1994, **1**, 5–21.

[18] Hedges LS, Travin AK, Spalart PR. Detached-eddy simulations over a simplified landing gear. *Journal of Fluids Engineering*; 2002, **124**, DOI: 10.1115/1.1471532, 413–423.

[19] Cummings RM, Forsythe JR, Mortor SA, Squires KD. Computational challenges in high angle of attack flow prediction. *Progress in Aerospace Sciences*; 2003, **39**, DOI: 10.1016/S0376-0421(03)00041-1, 369–384.

[20] Greschner B, Thiele F, Jacob MC, Casalino D. Prediction of sound generated by a rod-airfoil configuration using easm des and the generalised lighthill/fw-h analogy. *Computers and Fluids*; 2008, **37**, DOI: 10.1016/j.compfluid.2007.02.013, 402–412.

[21] Paik J, Sotiropoulos F, Porté-Agel F. Detached eddy simulation of flow around two wall-mounted cubes in tandem. *Heat and Fluid Flow*; 2009, **30**, DOI: 10.1016/j.ijheatfluidflow.2009.01.006, 286–305.

[22] Michelsen JA. *Basis3D - a platform for development of multiblock PDE solvers.* Tech. Rep. AFM 92-05, Technical University of Denmark, Lyngby, 1992.

[23] Michelsen JA. *Block structured multigrid solution of 2D and 3D elliptic PDEs.* Tech. Rep. AFM 94-06, Technical University of Denmark, Lyngby, 1994.

[24] Sørensen NN. *General Purpose Flow Solver Applied to Flow over Hills*. Ph.D. thesis, RISØ National Laboratory, Roskilde, risø-R-827(EN), 1995.

[25] Menter FR. Zonal two equation $k-\omega$ turbulence models for aerodynamic flows. *24th Fluid Dynamics Conference*, 1993, Orlando, FL, July 6–9, AIAA paper no. 93-2906.

[26] Menter FR, Langtry RB, Likki SR, Suzen YB, Huang PG, Völker S. A correlation-based transition model using local variables – Part I: Model formulation. *Journal of Turbomachinery*; 2006, **128**, DOI: 10.1115/1.2184352, 413–422.

[27] Langtry RB, Menter FR, Likki SR, Suzen YB, Huang PG, Völker S. A correlation-based transition model using local variables – Part II: Test cases and industrial applications. *Journal of Turbomachinery*; 2006, **128**, DOI: 10.1115/1.2184353, 423–434.

[28] Sørensen NN. CFD modelling of laminar-turbulent transition for airfoils and rotors using the $\gamma - \widetilde{\mathrm{Re}}_\theta$ model. *Wind Energy*; 2009, DOI: 10.1002/we.

[29] Mann J. Wind field simulation. *Prob. Engng. Mech.*; 1998, **13**, 269–282.

[30] Shinozuka M, Jan CM. Digital simulation of random processes and its application. *Journal of Sound and Vibration*; 1972, **25**, 111–128.

[31] Mann J. The spatial structure of neutral atmospheric surface-layer turbulence. *Journal of Fluid Mechanics*; 1994, **273**, 141–168.

[32] IEC. *IEC 61400-1 Wind Turburbines - Part 1: Design Requirements*. IEC, 3rd edn., IEC 61400-1:2005(E), 2005.

[33] Bechmann A. *Large-Eddy Simulation of Atmospheric Flow over Complex Terrain*. Ph.D. thesis, Risø National Laboratory, Roskilde, 2006.

[34] Comte-Bellot G, Corrsin S. The use of a contraction to improve the isotropy of grid-generated turbulence. *Journal of Fluid Mechanics*; 1966, **25**, 657–682.

[35] Spalart PR. *Young-person's guide to detached eddy simulation grids*. Tech. Rep. NASA/CR-2001-211032, Langley Research Center, NASA, 2001.

[36] Mish PF. *Mean Loading and Turbulence Scale Effects on the Surface Pressure Fluctuations Occurring on a NACA 0015 Airfoil Immersed in Grid Generated Turbulence*. Master's thesis, Virginia Polytechnic Institute and State University, Virginia, 2001.

[37] Mish PF, Devenport W. An experimental investigation of unsteady surface pressure on an airfoil in turbulence - Part 1: Effects on mean loading. *Journal of Sound and Vibration*; 2006, **296**, 417–446.

[38] Sørensen NN. CFD modeling of laminar-turbulent transition for airfoils and rotors. *26th AIAA Applied Aerodynamics Conference. AIAA Paper 2008-7323*; 2008.

[39] Gilling L, Sørensen NN, Davidson L. Detached eddy simulations of an airfoil in turbulent inflow. *47th AIAA Aerospace Sciences Meeting. AIAA Paper 2009-0270*; 2009.

Paper II

Imposing Resolved Turbulence in CFD Simulations

Lasse Gilling[1] and Niels N. Sørensen[2]

Submitted to *Wind Energy*.

[1] Aalborg University, Department of Civil Engineering, Aalborg, Denmark
[2] Risø-DTU National Laboratory for Sustainable Energy, Wind Energy Division, Roskilde, Denmark

Imposing Resolved Turbulence in CFD Simulations

Lasse Gilling* and Niels N. Sørensen†

Abstract

In large-eddy simulations the inflow velocity field should contain resolved turbulence. This paper describes and analyzes two methods for imposing resolved turbulence in the interior of the domain in CFD simulations. The intended application of the methods is to impose resolved turbulence immediately upstream of the region or structure of interest. Comparing to the alternative of imposing the turbulence at the inlet there is a large potential to reduce the computational cost of the simulation by reducing the total number of cells. The reduction comes from a lower demand for mesh resolution in the upstream part of the domain. The first method uses a modification of the source terms in the discrete Navier-Stokes equations. In the second method an actuator is used to impose the turbulence. The methods are tested and the most accurate is shown to be the approach of modifying the source terms. None of the two methods can impose synthetic turbulence with good results, but it is shown that by running the turbulence field through a short precursor simulation very good results are obtained.

Keywords: CFD, Forcing, Free stream resolved turbulence, Large-Eddy Simulation

1 Introduction

In large-eddy simulations (LES) and direct numerical simulations (DNS) the inflow boundary condition is less simple than for simulations with the Reynolds-averaged Navier-Stokes (RANS) equations. For simulations with a turbulent inflow condition the flow field impinging on the structure of interest should contain resolved eddies. Further, the structure of interest is often required to be placed far from the inlet to make the results independent from the exact position. This gives a long stretch of upstream domain in which an induced mean flow field is determined. When the resolved free stream turbulence is imposed at the inlet boundary condition the resolution in the upstream part of the domain must be fine enough for the resolved turbulence to be convected to the region of interest without an unphysical decay. The large number of cells needed makes it a

*Aalborg University, Department of Civil Engineering, Sohngaardsholsvej 57, 9000 Aalborg, Denmark. E-mail: lg@civil.aau.dk, Phone: +45 9940 8544

†Risø-DTU National Laboratory for Sustainable Energy, Wind Energy Division, Frederiksborgvej 399, P.O. Box 49, 4000 Roskilde, Denmark

very computationally expensive approach. This paper investigates some methods for imposing the resolved turbulence in the interior of the computational domain immediately upstream of the region of interest. Thereby, the resolution upstream of the region of interest can be coarse because only the mean flow field should be resolved there.

An example of the application is given in Figure 1 taken from [1]. It shows the vorticity close to an airfoil in a detached-eddy simulation (DES). Upstream of the airfoil an actuator imposes resolved turbulence on the mean flow field. The resolved turbulence impinges on the airfoil and influences the solution in the boundary layer. Upstream of the actuator plane a coarse grid is used.

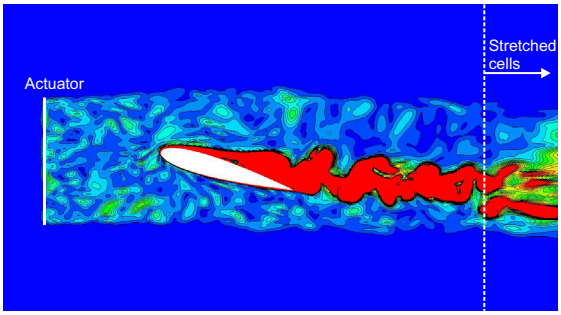


Figure 1: Vorticity plot from a DES of an airfoil. From [1].

A similar approach was applied in [2] to simulate the flow past a wind turbine rotor, which was also modeled by an actuator. Troldborg and co-workers [3, 4] imposed the free stream turbulence by modifying the source terms of the discrete Navier-Stokes equations.

The most common approach for including free stream turbulence in a CFD simulation is to impose the turbulence at the inlet, cf. e.g. [5]. In another approach described by [6] modeled kinetic energy is transferred to resolved turbulence when the modeled energy is convected into a region with finer mesh resolution. In [7] random forces are used to generate homogeneous turbulence. Another application of imposing resolved turbulence is to increase the resolved stresses in the transition region of hydrid LES-RANS. [8] uses stochastic forcing in the transition region in a LES where the detached-eddy simulation (DES) is used for wall modeling. A similar problem was faced in [9, 10] in which source terms are added to the momentum equations to promote the resolved stresses. [11] uses forcing to quickly generate a fully turbulent flow in a channel.

This paper will concentrate on imposing a *given* turbulence field and not just *some* turbulence field with specified mean, spectra, etc. Specific time series of turbulence will be imposed in interior of the computational domain. Two different methods for imposing the turbulence are investigated. The first method is to modify the source terms of the discrete momentum equations to obtain the desired velocities. In the other method an actuator model is used to apply forces to a number of cells in the domain to create the desired turbulence field. For comparison the turbulence field will also be imposed at the

inlet. These tests will be used as a baseline to access which effects are caused by diffusion, decay, etc. The requirements for the two investigated methods are

- The turbulence should be added in the interior of the domain.
- It should be superposed to the existing solution, and thereby allow the computational code to predict the mean flow field.

The scope of this paper is to clearly define the methods and to access their advantages and disadvantages. First, the numerical setup is described and the models are defined. Then, results are presented for different turbulence fields and finally, some conclusions are given.

2 Numerical Methods

The incompressible Navier-Stokes equations are solved using EllipSys3D [12, 13, 14]. It is a structured, finite volume CFD-code developed at RISØ and Technical University of Denmark. The PISO-algorithm is used to enforce the pressure-velocity coupling and the Rhie-Chow interpolation is used to avoid odd/even pressure decoupling. The momentum equations are solved with the 4th order central difference scheme and a 2nd order accurate time stepping algorithm is used. The simulations are run as DES [15], but also the incoming free stream turbulence is resolved. Thereby the model is run in LES-mode in most of the region of interest except near solid walls. The $k-\omega$ SST subgrid model by Menter [16] is used as in the DES formulation by [17].

When the turbulence is imposed by an actuator a modified version (c.f. [18]) of the Rhie-Chow pressure correction algorithm is used to avoid numerical wiggles at the actuator.

3 Methods for Imposing Resolved Turbulence

The turbulence is imposed by modifying the flow in a number of cells. These cells are located somewhere in the interior of the computational domain. The coordinates of the N cells are denoted $\boldsymbol{x}_j$ where $j = 1, 2, \ldots, N$. A turbulence field u', v', w' is imposed on the velocity field u, v, w. The velocities u, v, w describe the induced field upstream of the cells where turbulence is imposed. The turbulence field u', v', w' contains time series of velocities for each point $\boldsymbol{x}_j$. These disturbances can be turbulence from a precursor simulation or it can be synthetically generated turbulence.

3.1 Prescribing Velocities at the Inlet

In the baseline tests the turbulence is imposed at the inlet. The fluctuating velocity field given by the three velocity components u', v', w' is superposed to the mean farfield

velocities $U_\infty, V_\infty, W_\infty$

$$\left.\begin{array}{l} u_{inlet}(t) = U_\infty(t) + u'(t) \\ v_{inlet}(t) = V_\infty(t) + v'(t) \\ w_{inlet}(t) = W_\infty(t) + w'(t) \end{array}\right\} \tag{1}$$

3.2 Modifying Source Terms

By changing the source terms in the discrete Navier-Stokes equations it is possible to obtain a desired velocity. In each of the cells where the turbulence is imposed the source terms of the three momentum equations are modified.

At time step t the discrete Navier-Stokes equations are

$$\sum -A_{nb}\Phi_p^t = \sum A_{nb}\Phi_{nb}^t + S + \frac{\rho Vol}{\Delta t}\left(-\frac{3}{2}\Phi_p^t + 2\Phi_p^{t-1} - \frac{1}{2}\Phi_p^{t-2}\right) \Leftrightarrow \tag{2}$$

$$\Phi_p^t = \frac{\sum A_{nb}\Phi_{nb}^t + S + \frac{\rho Vol}{\Delta t}\left(-\frac{3}{2}\Phi_p^t + 2\Phi_p^{t-1} - \frac{1}{2}\Phi_p^{t-2}\right)}{\sum -A_{nb}} \tag{3}$$

where Φ is one of the velocity components, A is a coefficient in the discrete momentum equation, S is the source term, ρ is the density and Vol is the cell volume. The subscript p denotes the cell with neighbors indicated by nb. A turbulent correction is added

$$\Phi_p^t = \frac{\sum A_{nb}\Phi_{nb}^t + S + \frac{\rho Vol}{\Delta t}\left(-\frac{3}{2}\Phi_p^t + 2\Phi_p^{t-1} - \frac{1}{2}\Phi_p^{t-2}\right)}{\sum -A_{nb}} + \Phi_p' \tag{4}$$

Next, the equation is under relaxed and the term of Φ_p^t on the righthand side is moved to the left to make the equation more diagonal dominant. We arrive at the discrete momentum equation is the form it is solved by the flow solver

$$\left(\frac{\sum -A_{nb}}{\alpha} + \frac{3\rho Vol}{2\Delta t}\right)\Phi_p^t = \frac{\sum A_{nb}}{\alpha}(1-\alpha)\Phi_p^{t*} + \sum A_{nb}\Phi_{nb}^t + S + \frac{\rho Vol}{\Delta t}\left(2\Phi_p^{t-1} - \frac{1}{2}\Phi_p^{t-2}\right) + \sum -A_{nb}\Phi_p' \tag{5}$$

For typical applications also the velocities in the neighboring cells are modified. These contributions are added to the last term to give the correction source term that will yield the desired turbulence field

$$S' = \sum -A_{nb}\Phi_p' + \sum A_{nb}\Phi_{nb}' \tag{6}$$

Hence, by adding the correction source term in (6) to the discrete momentum equation the change in Φ_p of Φ_p' is obtained. The term is absorbed in the source term.

3.3 Actuator Disc Approach

When the turbulence is imposed by an actuator, a fluctuating force is applied in a number of cells. Below, the required forces are derived. In Figure 2 the pressure jump and the velocity change at an actuator is shown. The fluid is accelerated from U_∞ to $U_\infty + u'$ by the pressure jump. The pressure jump required to accelerate the flow is, [19]

$$\Delta p = \frac{1}{2}\rho\left((U_\infty + u')^2 - U_\infty^2\right) \tag{7}$$

where ρ is the density of the fluid. The pressure is derived from steady one-dimensional momentum theory using the Bernoulli equation.

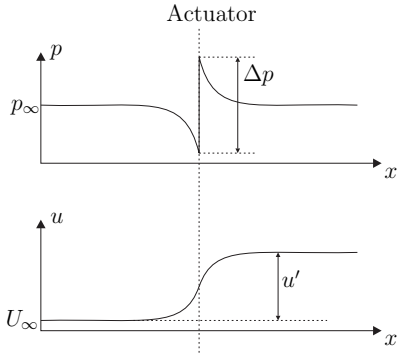


Figure 2: Pressure jump at the actuator and change of velocity at the actuator (dashed line).

The pressure jump corresponds to a force applied in each computational cell crossed by the actuator. The magnitude of these forces is

$$F_c = \frac{1}{2}\rho A_c\left(U_\infty + \frac{1}{2}u'\right)u' \tag{8}$$

where A_c is the area of the actuator crossed by the cell. This can be generalized to include velocity changes in three dimensions

$$\boldsymbol{F}_c = \frac{1}{2}\rho A_c\left(U_\perp + \frac{1}{2}u'_\perp\right)\boldsymbol{u}' \tag{9}$$

where $\boldsymbol{u}'$ and $\boldsymbol{F}_c$ are vectors containing the three components of velocity and force vectors, respectively. $U_\perp$ is the length of the mean velocity vector orthogonal to the actuator and $u'_\perp$ is the corresponding length of the added velocity vector. The forces are imposed in the cells using the method described by [18].

The above derivation is based on a steady state consideration. When applying a fluctuating velocity field this steady state consideration is not valid. However, the obtained velocity field is in good agreement with the target velocity field, as will be shown below.

Under the assumptions of stationary conditions and inviscid flow simple manipulations with the Navier-Stokes equations show that the changes in the streamwise component builds up slowly in the region around the actuator as shown in Figure 2. At the actuator the velocity change is half the final value. For the transverse components the changes behave like the Heaviside unit step function and are introduced immediately at the actuator.

4 Results and Discussion

Small simulations are run to test the different methods of imposing the turbulence. At first, the computational setup is described. Below, three types of fields are imposed. The first is simple harmonic varying fields which contain the energy at a single frequency. Then, the more difficult case of precursor turbulence is tested. Here, the imposed field contains all resolvable frequencies. The precursor simulation ensures that the field satisfies the Navier-Stokes equations and the continuity condition. Finally, synthetic turbulence is imposed. This field contains energy on both resolvable frequencies and frequencies too high for the simulation. Further, incompressibility is only approximately satisfied.

4.1 Computational Setup

For the tests of the different methods a small domain is used. The domain is sketched in Figure 3 and it contains only cubic cells. The number of cells in the three directions are $N_x \times N_y \times N_z = 128 \times 32 \times 32$. In the transverse directions periodic boundary conditions are used and the height and width of the computational domain is H. The mean flow U_∞ is steady and aligned with the x-axis.

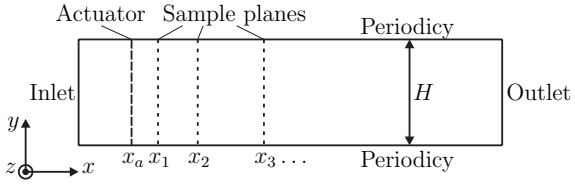


Figure 3: Computational setup.

The turbulence field is imposed in the entire cross section at $x = x_a$ and sampled in a number of planes located at $x = x_1, x_2, \ldots, x_n$. In each time step the three velocity components are sampled in all points in the $y - z$-plane of each sample plane. The values are stored to a file which then contains time series of the sampled velocities. These samples are compared to the imposed field to access the method of imposing the turbulence. The sampled field is denoted $\tilde{u}, \tilde{v}, \tilde{w}$.

A total of 11 sample planes are used to show the development of the imposed turbulence behind the actuator. The positions of the sample planes relative to the actuator normalized by the grid spacing are $(x_j - x_a)/\Delta x = -10, -5, 0, 2, 5, 10, 15, 20, 30, 40, 60$.

The effect of the boundary conditions is eliminated by using periodic boundary conditions in the transverse directions and imposing a periodic turbulence field in the entire cross section. Further, the turbulence is imposed some distance from the inlet, so the flow close to the inlet is undisturbed. The outlet boundary condition uses a simple assumption of zero gradients of the velocity in the flow direction. The assumption is not valid when the flow is turbulent, and to avoid influence of the boundary the sample planes are located far from the outlet.

4.2 Harmonics

Simple harmonic fluctuating fields are imposed to test the methods of imposing the turbulence. Two different types of fields are used. The first contains only a transverse component of the velocity vector and the field is constant over the cross section area of the computational domain. It is given by

$$v'(y,z,t) = v_0' \sin\left(2\pi\frac{t}{T}\right) \quad , \quad u'(y,z,t) = w'(y,z,t) = 0 \tag{10}$$

where t is time, T is the period of the harmonic and v_0 is the amplitude.

With the actuator the streamwise component builds up slowly while the transverse components achieve the full amplitude immediately after the actuator. Therefore, a field with non-zero streamwise component is also imposed. It is

$$\left.\begin{aligned} u'(y,z,t) &= u_0' \sin\left(2\pi\frac{t}{T}\right)\sin\left(2\pi\frac{y}{H}\right) \\ v'(y,z,t) &= -\frac{u_0' H}{T U_\infty}\cos\left(2\pi\frac{t}{T}\right)\cos\left(2\pi\frac{y}{H}\right) \\ w'(y,z,t) &= 0 \end{aligned}\right\} \tag{11}$$

where u_0 is the amplitude of the streamwise component. With application of Taylor's hypothesis the temporal variation can be written as a spatial variation under the transformation given by

$$t = -\frac{x}{U_\infty} \tag{12}$$

When (12) is used in (10) and (11) both fields are seen to be divergence free. Thereby, the fields can be superposed onto a steady mean flow without compromising the continuity condition.

In (12) the influence of the imposed streamwise velocities is ignored. The true convection velocity is $U_\infty + u'$ which varies in time and in the three spatial directions. The variation in convection velocity warps the turbulence as it is convected downstream. This effect is not included in the simple assumption used in Taylor's hypothesis. For small turbulence intensities the difference in convection speed is neglectable and the transformation (12) can be applied, see e.g. [20].

In Figures 4-6 the amplitude of the sampled oscillations in $\tilde{v}$ is shown for different CFL numbers and for different periods T. The amplitudes are normalized by the target

amplitude in (10) and the periods are normalized by the time step Δt_0 which gives a CFL number of unity. The results shown here are similar for the results obtained by imposing the field given by (11). The velocity field is sampled 20 cells downstream of the point where the disturbances are imposed.

In Figure 4 the results are shown for the case where the disturbances are imposed at the inlet boundary condition. It is seen that for the two cases of CFL numbers lower than unity the lines coincide. In these cases the resolution in space sets the lower limit for the periods of the harmonic. The normalization of the abscissa is $T/\Delta t_0$ and gives the number of cells per wave length of the harmonic, i.e. $T/\Delta t_0 = L/\Delta x$ where L is the wave length and Δx is the side length of the cells in the domain. For $T/\Delta t_0 > 15$ the simulations with low CFL numbers give an amplitude within 2% of the target. With CFL = 1 the lower limit for the resolvable periods is seen to be higher than for the low CFL numbers. For periods equal to 20 time steps the amplitude is 85% of the target value and at 40 the error is 1%. With CFL = 4 the temporal resolution limits the resolvable periods. At $T/\Delta t_0 = 20$ only 5 time steps per period is used and this wave dies out before it reaches the sample plane. For periods larger than $20\Delta t_0$ some energy is maintained and for periods larger than $100\Delta t_0$ the error is below 1%.

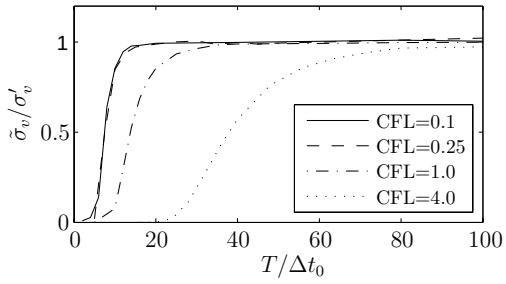


Figure 4: Amplitudes of harmonics given by (10) and imposed at the inlet. The field is sampled in a plane 20 cells from the inlet.

Figures 5 and 6 show the results obtained from imposing the disturbances by modifying the source terms and by using the actuator, respectively. The two methods give nearly similar results to the baseline test where the disturbances are imposed at the inlet. Hence, both methods give almost perfect results. For the case where turbulence is imposed by modifying the source terms it is seen that the obtained amplitudes are slightly higher than the target for high periods at CFL = 0.1 . The difference is about 2%. For the remaining CFL numbers the error is well below 1% as in the baseline test.

For the case with the actuator the obtained amplitude is slightly lower than the target amplitude. The differences are about $4 - 5\%$ at high periods.

Spalart [21] argues that an eddy being convected with a mean velocity cannot be resolved with less than five cells. Further, the CFL number should be unity in the region of interest to optimize the value from the computational cost. From Figures 4–6 it is

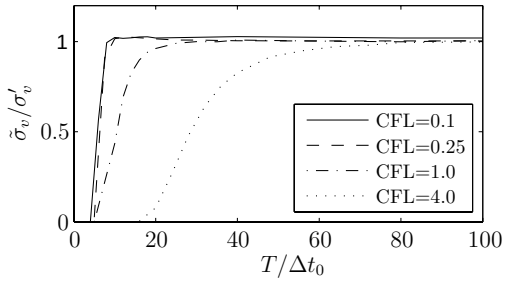


Figure 5: Amplitudes of harmonics given by (10) and imposed by modifying the source terms. The field is sampled 20 cells from the plane where it is imposed.

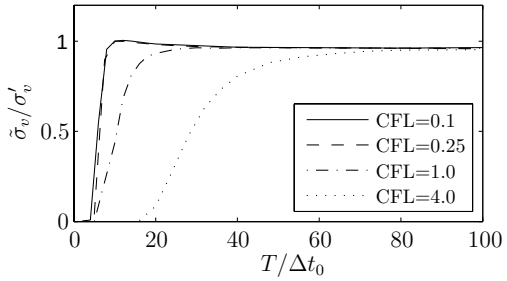


Figure 6: Amplitudes of harmonics given by (10) and imposed by the actuator. The field is sampled 20 cells from the plane where it is imposed.

seen that oscillations with wave lengths less than 5 cells cannot be resolved, and still at 10 cells per wave length the energy loss is significant.

4.3 Turbulence from Precursor Simulation

A synthetic, isotropic turbulence field has been generated by the method of Mann [22] which is based on the work by Shinozuka and Jan [23]. The generated field is imposed at the inlet of a domain like the one sketched in Figure 3. In a $y - z$-plane 20 cells downstream of the inlet the turbulence field is sampled. This sampled turbulence field is imposed in the tests presented here. The sampled field satisfies the Navier-Stokes equations and in the discrete form it is incompressible. Further, since the field is sampled from a simulation with the same mesh, it is adapted to the mesh.

A total of 4096 time steps have been run and the CFL number is 0.25. The integral length scale of the synthetic turbulence is $0.5H$. The turbulence intensity is 1.2%.

In Figure 7 instantaneous iso-vorticity contours are shown from a simulation where

the turbulence is imposed by the modification of the source terms. It is seen that the flow is laminar and steady upstream of the location where the turbulence is imposed.

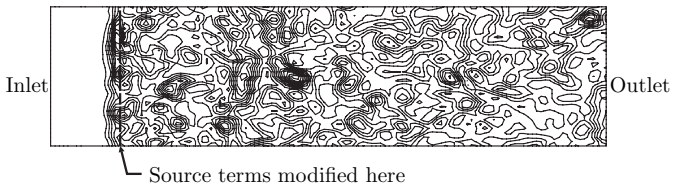


Figure 7: Instantaneous iso-vorticity contours. Precursor turbulence is imposed by modifying the source terms.

In Figures 8-10 the imposed time series are compared to the time series sampled in the sample planes. Plotted on the left part of the figures is the ratio of the standard deviations of the sampled and the imposed field. This is a measure of the intensity of the generated field. On the plots in the figures to the right is the average correlation between the sampled and the target time series. This is a measure of the agreement of the shape of the turbulent structures in the flow with the imposed field. Both the correlation coefficient and the ratios of standard deviations are plotted against the distance from the plane where the turbulence is imposed. By use of Taylors hypothesis the sampled time series are shifted in time for comparison with the imposed time series.

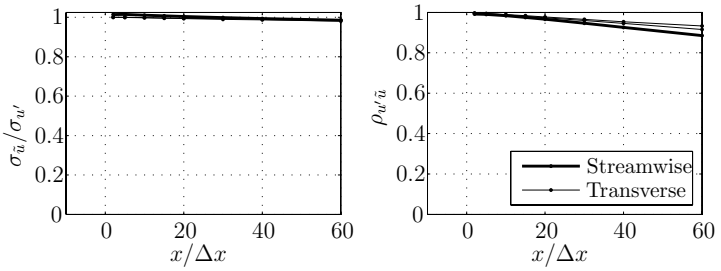


Figure 8: Precursor turbulence imposed at the inlet. Comparison of imposed and sampled time series.

In Figure 8 the results are shown from the simulation where the turbulence is imposed at the inlet. It is seen that the sampled time series agree quite well with the target time series. The standard deviations are within 2% of the targets for both the streamwise and the transverse components. For the first 5 cells closest to the inlet the correlation coefficient is above 0.99, and further downstream it decreases slowly. The reduction in standard deviation with distance from the inlet is caused by decay of the turbulence. The reduction in correlation coefficient is caused by warping of the eddies as discussed in Section 4.2. The physical decay of the turbulence is low due to a high convection

velocity U_∞ and short distances compared to the decay rate of the turbulence which was experimentally determined by [24].

The results from the simulation with modified source terms are shown in Figure 9. The turbulence field is sampled in a number of planes located both upstream and downstream of the plane where it is imposed as described in Section 4.1. The standard deviation of the streamwise component is seen to build up slowly and it does not reach its full magnitude until 15 cells downstream of x_a. The transverse and streamwise components give standard deviations within 1% of the target. 10 to 20 cells downstream of $x = x_a$ the correlation peaks at about 0.99 and in the remaining region the agreement is about as good as in the baseline test.

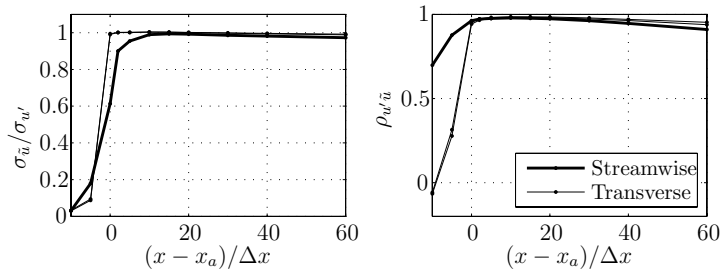


Figure 9: Precursor turbulence imposed by modifying the source terms at $x = x_a$. Comparison of imposed and sampled time series.

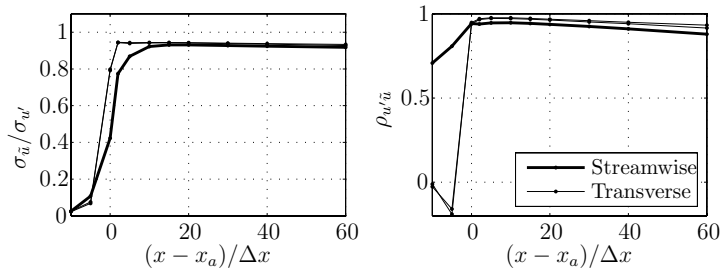


Figure 10: Precursor turbulence imposed by an actuator at $x = x_a$. Comparison of imposed and sampled time series.

Finally, Figure 10 presents results from the actuator. The results are seen to be quite similar to the ones from the modified source terms. Again, the intensity of the streamwise components builds up slowly. It is observed that at the actuator the streamwise component reaches about half its final magnitude, which is also the theoretical value as described in Section 3. In agreement with the theory the transverse components are generated much faster. The obtained intensity is about 7% too low for the streamwise

component and about 6% too low for the transverse component. The maximum correlation coefficients are about 0.95 and 0.97 for the streamwise and transverse components, respectively.

The Figures 11-13 compare the power spectra of the sampled time series to the imposed time series. The figures contain plots of the three velocity components. For clarity the u-component has been moved two decades up and the w-component has been moved two decades down. The indicated cut-off frequency ω_{cut} corresponds to an eddy spanning five cells. All spectra have been averaged over all points in the $y-z$-plane. The sampled time series have been gathered 20 cells downstream of the location where the turbulence is imposed.

Figure 11 shows results for the baseline test with turbulence imposed at the inlet. The agreement between the spectra of the sampled and target time series is seen to be very good. In the entire range of the resolvable frequencies the lines coincide. At frequencies higher than the cut off frequency both the sampled and the target spectra are seen to drop off as expected.

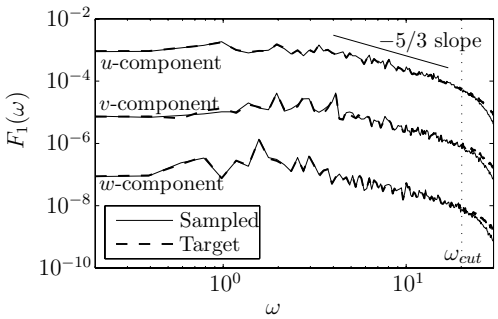


Figure 11: Precursor turbulence imposed at the inlet. Comparison of power spectra of imposed and sampled time series.

Figure 12 shows excellent agreement from the modified source terms. The agreement is seen to be as good as the baseline test with turbulence imposed at the inlet.

Finally, the results from the actuator are shown in Figure 13. It shows some deficiencies and all three components contain too little energy on all scales. The deficiency is largest at high frequencies, and in the inertial subrange the errors are small.

In [25] the spatial decay of turbulence was studied using the same numerical method as used here. The precursor turbulence was imposed at the inlet and good agreement with the emperical expression found by [24] was found.

4.4 Synthetic Turbulence

In this final series of tests a field of synthetic turbulence is imposed. The field is the same as was used for the precursor simulation, but here it is imposed directly. The CFL

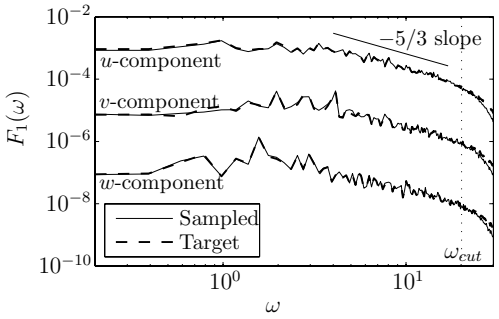


Figure 12: Precursor turbulence imposed by modifying the source terms. Comparison of power spectra of imposed and sampled time series.

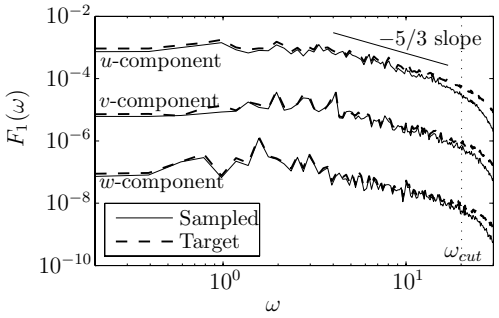


Figure 13: Precursor turbulence imposed by an actuator. Comparison of power spectra of imposed and sampled time series.

number is 0.25 and the synthetic field is generated so that it contains a new slice of turbulence every time step. Hence, the streamwise resolution in the synthetic field is four times finer than in the transverse directions.

When generating synthetic turbulence with the method of Mann [22] all three velocity components are generated at once. The resulting field is a discrete representation of a continuous field that is divergence free, but when applying e.g. a second order difference scheme to evaluate the divergence there are significant errors. The reason for this discrepancy is that the continuous field contains harmonics with very high frequencies and the discrete representation of these fast harmonics cannot be differentiated with decent accuracy. Therefore, the discrete representation of the generated field will appear to contradict the continuity condition. Further, the synthetic field contains small eddies with a wave number lower than what can be resolved in a simulation where the turbulence is convected with a mean flow velocity. Finally, the generated turbulence have correct statistical moments up to second order but the phase information is incorrect. When

the synthetic turbulence is imposed the continuity condition will be enforced by the flow solver and the high frequency scales will be filtered away by the mesh. For these reasons the agreement between the sampled and the imposed fields cannot be expected to be as good as presented for the precursor turbulence.

Figure 14 shows the results from the baseline test where the turbulence is imposed at the inlet. Close to the inlet the amplitude of the streamwise velocity is seen to be too high and the transverse component is seen to be too low. Over the first 20 cells they converge and reach about 0.95 of the target intensity. From there the turbulence decays slightly. The correlation coefficients shown to the right are significantly lower than unity. Especially, the streamwise component gives a correlation coefficient of about 0.65 while the transverse directions give about $0.8-0.85$ at a distance of 20 cells from the inlet. The large difference in streamwise and transverse directions can be attributed to the difference in resolutions. In the transverse directions a single point in the discrete turbulence field exists for every cell. In the streamwise (and thereby temporal) direction four points exist for every cell. Thereby, more information is filtered away in the streamwise direction.

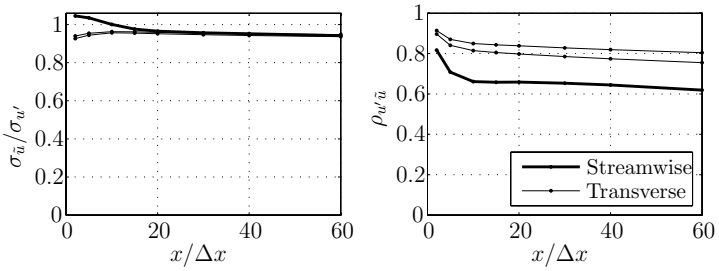


Figure 14: Synthetic turbulence imposed at the inlet. Comparison of imposed and sampled time series.

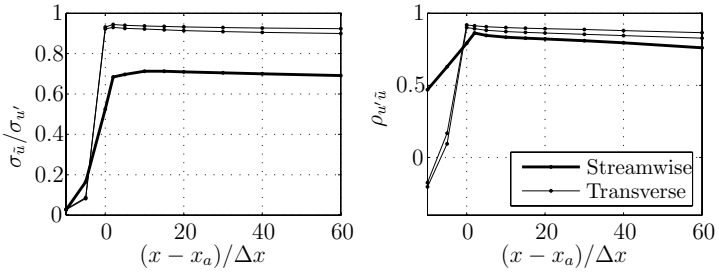


Figure 15: Synthetic turbulence imposed by modification of source terms at $x = x_a$. Comparison of imposed and sampled time series.

Figure 15 shows the results from the simulation where the turbulence is imposed by a modification of the source terms. The streamwise component gives about 30% too

low amplitude and the transverse give $5-10\%$ too low amplitude. 20 cells from the location where the turbulence is imposed the correlation coefficients are 0.8 and 0.9 in the streamwise and transverse directions, respectively. This is a higher correlation for the streamwise component than in the baseline test, but in turn the intensity is lower.

The results from the simulation with the actuator are shown in Figure 16. The shapes of the curves are quite similar to the curves from the modified source terms, but here the intensities are about 5 percent points lower. This gives an intensity of the streamwise component just over 30% too low and the transverse components are $10-15\%$ too low. The correlation coefficients are also about 5 percent points lower than with the modified source terms. The correlation coefficients are in the range $0.8-0.9$.

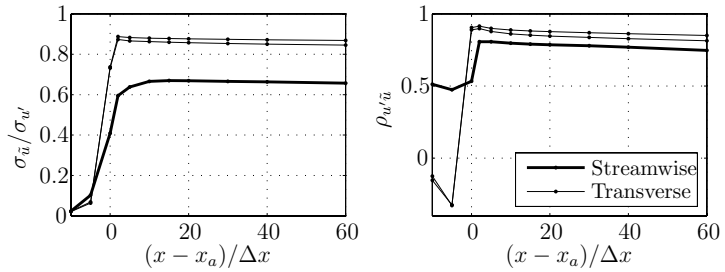


Figure 16: Synthetic turbulence imposed by an actuator at $x = x_a$. Comparison of imposed and sampled time series.

In Figures 17-19 power spectra of the target and sampled time series are shown for the case of synthetic turbulence. As in Section 4.3 the results are shown for time series sampled 20 cells downstream of the location where the turbulence is imposed.

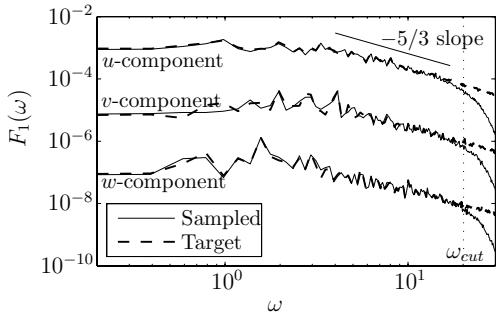


Figure 17: Synthetic turbulence imposed at the inlet. Comparison of power spectra of imposed and sampled time series.

Figure 17 shows the average power spectra of the time series from the baseline simulation where the turbulence is imposed at the inlet boundary condition. It is seen that in

most of the resolvable range the spectra of the sampled and target time series coincide. At the highest resolvable frequencies the energy of the sampled time series drops below the target. This is exactly the desired behavior. As seen in Figure 14 there is a significant loss of energy, but the energy is only lost in the frequency range too high to resolve in the simulation.

In Figure 18 the power spectra are shown from the simulation where the turbulence is imposed a modification of the source terms. For the two transverse directions the results are very similar to those described above where all the energy is preserved in the resolvable range. For the streamwise component there is a significant deficiency at highest decade of resolvable frequencies. For the lowest frequencies the agreement is good.

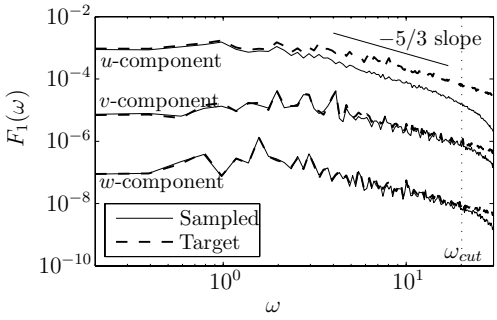


Figure 18: Synthetic turbulence imposed by modification of source terms. Comparison of power spectra of imposed and sampled time series.

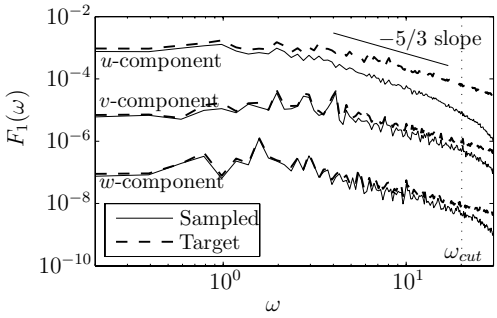


Figure 19: Synthetic turbulence imposed by an actuator. Comparison of power spectra of imposed and sampled time series.

Finally, Figure 19 shows the results for the simulation with turbulence imposed by an actuator. For the transverse components of the velocity vector the deficiency is larger than for the two other methods. Also for the streamwise component the deficiency is

larger at all frequencies. The error is almost an entire decade at the largest resolvable frequencies.

4.5 Required Length of the Precursor Simulation

Comparing the results presented in Sections 4.3 and 4.4 it is clear that both methods give good results when precursor turbulence is imposed and quite poor results when synthetic turbulence is imposed. Especially, the method of modifying the source terms gives very good results, it is consistent and it is easily implemented. Therefore, the computations in this section will be using the method with the modified source terms.

When running precursor simulations for LES of e.g. atmospheric or channel flows the precursor simulation can often be more expensive than simulation of the flow of interest, c.f. [5]. In these cases the turbulence must build up naturally and the boundary layer must be formed.

In the present application the precursor simulation filters away the unresolvable frequencies of the synthetic turbulence field, forces the discrete representation to satisfy the continuity condition and adapts the field to the mesh and to the the numerical method. This type of precursor is extremely cheap. In Section 4.3 the turbulence was sampled 20 cells from the inlet. As will be demonstrated below, the turbulence can be sampled even closer to the inlet and still be imposed with good accuracy.

From Figures 12 and 18 it is seen that the transverse components can be imposed with good accuracy both with synthetic and precursor turbulence. The streamwise component is much more challenging. The turbulence field is recreated poorly with the synthetic turbulence but with the precursor turbulence the agreement is good. It is clear that running the turbulence field through a precursor simulation where it is imposed at the inlet significantly improves the agreement with the target field.

Next, the required length of the precursor simulation is investigated. A precursor simulation is run where the turbulence is sampled in the $y-z$-plane at $x_s/\Delta x = 0.5, 1.5, 2.5, 3.5, 4.5$ corresponding to the first five layers of cell centers. These sampled fields are imposed on five different simulations by the method of modifying the source terms. The obtained turbulence in these simulations is sampled to access the required length of the precursor simulation.

Figure 20 shows the power spectra of the time series sampled from the five simulations. Only the power spectra of the streamwise component are shown. For comparison the power spectrum of the original synthetic turbulence field is included in the plot. The spectrum from the simulation with the precursor turbulence sampled in the first layer of cells is quite similar to the power spectrum from the simulation with the synthetic turbulence shown in Figure 18. The discrepancy between the sampled field and the synthetic field arises when the turbulence is imposed inside the domain by the source terms; the turbulence field sampled from the precursor simulation is nearly identical to the synthetic field, but it cannot be imposed by modification of the source terms.

To easily distinguish the simulations the results for the four consecutive simulations

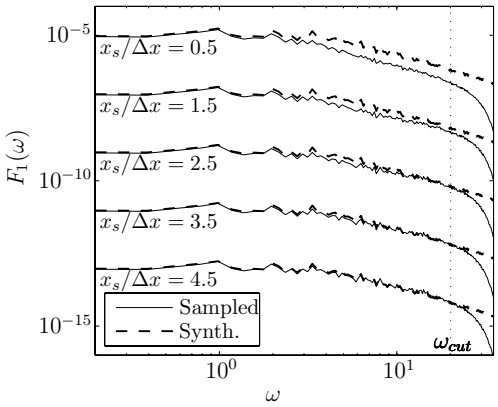


Figure 20: Precursor turbulence imposed by a modification of the source terms. Power spectra from simulations with turbulence from precursors of different lengths are compared to the same synthetic field.

have been moved downwards by two decades per simulation. The further downstream the imposed turbulence was sampled in the precursor the better agreement is obtained. When the turbulence is sampled five cells from the inlet, the agreement with the target field is as good as the baseline test in Figure 17 where synthetic turbulence was imposed at the inlet.

Using the precursor where the turbulence is sampled five cells from the inlet gives exactly the desired results. In the resolvable range of frequencies all the energy is generated and only energy at higher frequencies is lost. Thereby, the obtained turbulence is a lowpass filtered version of the synthetic turbulence field with the correct amplitudes of the oscillations.

In conclusion the synthetic turbulence should be run through a short precursor simulation before it is imposed inside the domain. The result is a field that contains the desired energy on all resolvable frequencies.

5 Example of Application

The presented methods can be applied in all cases where inflow turbulence is included in the simulations. As will be demonstrated here a considerable saving in the number of cells can be obtained.

In Figure 21 the mesh of an airfoil simulation is shown. The airfoil is located in a wind tunnel and resolved inflow turbulence is included. The mesh is three dimensional and consists of a two dimensional mesh extruded in the spanwise direction. Close to the airfoil an O-mesh is used and the rest of the domain is filled with Cartesian cells.

The inlet is located four chords upstream of the leading edge and turbulence is imposed about half a chord upstream of the airfoil. The inlet must be located a good distance upstream of the airfoil to allow the flow solver to determine the correct induced flow in the upstream region. The outlet is located far downstream of the airfoil. Figure 1 shows the instantaneous iso-vorticity close to the airfoil.

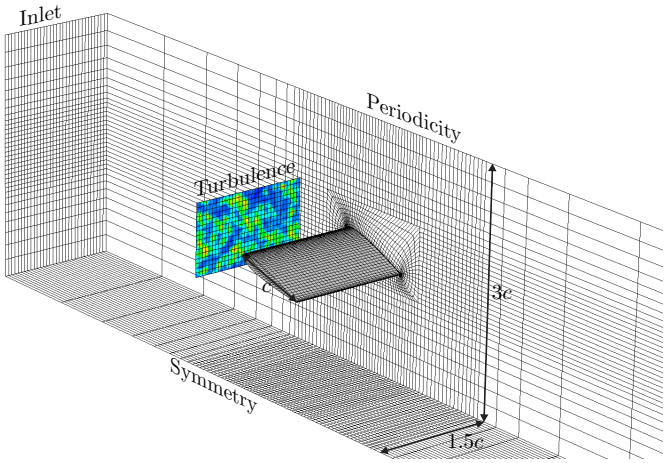


Figure 21: The computational mesh from [1]. Every fourth cell is shown.

By imposing the turbulence immediately upstream of the airfoil only a small region close to the airfoil has to be resolved with a fine mesh. Only in this region the inflow turbulence is resolved. Upstream of the plane where the turbulence is imposed only the steady mean flow has to be resolved. This is also the case for the region above and below the airfoil.

If alternatively the turbulence was imposed at the inlet a fine mesh resolution should be used everywhere upstream of the airfoil. [1] compares a mesh designed for the turbulence being imposed at the inlet to the mesh shown in Figure 21. Using identical resolutions close to the airfoil a saving of about 60% in the total number of cells was obtained by imposing the turbulence immediately upstream of the airfoil. Further, the resulting surface pressure was the same in the two simulations.

6 Conclusions

Two methods for imposing resolved turbulence in CFD simulations were described and analyzed. The first method used a modification of the source terms of the discrete momentum equations. This approach gave the best agreement with the target fields. Further, the method is easily implemented.

The second method was to impose the turbulence by an actuator. Resolved turbulence

was generated by applying forces to the cells crossed by the actuator. This approach gave decent results with the precursor turbulence, but is gave too low energy on the highest resolvable frequencies.

None of the methods gave good results when imposing synthetic turbulence. However, it was shown that by running the synthetic turbulence through a short precursor the results improved significantly. Therefore, it is recommended to run the synthetic turbulence through a short precursor where the turbulence is sampled 5 to 20 cells downstream of the inlet where the turbulence is imposed. The method using the modified source terms is recommended for imposing the resolved turbulence in the domain. This approach gave the best results and combined with a short precursor the correct energy on all resolvable frequencies, that can be obtained.

The example of application showed a considerable saving in the required number of cells without coarsening the mesh in the region of interest.

Acknowledgements

This work was funded by "The Danish Council for Strategic Research" under the project "Airfoils in Turbulent Inflow" with project no. 2104-05-0074. The simulations have been carried out on Alfheim - A linux cluster located at DTU and financed by Danish Center for Scientific Computing (DCSC) Grant nr. HDW-0107-02 from January 2007.

References

[1] Gilling L, Sørensen NN, Réthoré P-E. Imposing resolved turbulence by an actuator in a detached-eddy simulation of an airfoil. *EWEC 2009*, Marseille, March 16–19.

[2] Mikkelsen R. *Actuator Disc Methods Applied to Wind Turbines*. 2003, Ph.D. thesis, Technical University of Denmark, Lyngby.

[3] Troldborg N, Sørensen JN, Mikkelsen R. Actuator line simulation of wake of wind turbine operating in turbulent inflow. *Journal of Physics: Conference Series* 2007; **75**, DOI: 10.1088/1742-6596/75/1/012063.

[4] Troldborg N. *Actuator Line Modelling of Wind Turbine Wakes*. 2008, Ph.D. thesis, Technical University of Denmark, Lyngby.

[5] Bechmann A. *Large-Eddy Simulation of Atmospheric Flow over Complex Terrain*. 2006, Ph.D. thesis, Risø National Laboratory, Roskilde.

[6] Batten P, Goldberg U, Chakravarthy S. Interfacing statistical turbulence closures with large-eddy simulation. *AIAA Journal* 2003; **42**, 485–492.

[7] Alvelius K. Random forcing of three-dimensional homogenous turbulence. *Physics of Fluids* 1999; **11**, 1880–1889, DOI: S1070-6631(99)01007-7.

[8] Keating A, Piomelli U. A dynamic stochastic forcing method as wall-layer model for large-eddy simulation. *Journal of Turbulence* 2006; **7**, DOI: 10.1080/14685240612331392460.

[9] Davidson L, Dahlström S. Hybrid LES-RANS: Computation of the flow around a three-dimensional hill. *Engineering Turbulence Modeling and Experiments* 2005; **5**, 319–328.

[10] Davidson L, Billson M. Hybrid LES-RANS using synthesized turbulent fluctuations for forcing in the interface region. *International Journal of Heat and Fluid Flow* 2006; **27**, 1028–1042, DOI: 10.1016/j.iheatfluidflow.2006.02.025.

[11] Keating A, Piomelli U, Balaras E, Kaltenbach H-J. A priori and posteriori tests of inflow conditions for large eddy simulations. *Physics of Fluids* 2004; **16**, 4696–4712, DOI: 10.1063/1.1811672.

[12] Michelsen JA. Basis3D - a platform for development of multiblock PDE solvers. 1992, Tech. Rep. AFM 92-05, Technical University of Denmark, Lyngby.

[13] Michelsen JA. Block structured multigrid solution of 2D and 3D elliptic PDEs. 1994, Tech. Rep. AFM 94-06, Technical University of Denmark, Lyngby.

[14] Sørensen NN. *General Purpose Flow Solver Applied to Flow over Hills*. 1995, Ph.D. thesis, RISØ National Laboratory, Roskilde, risø-R-827(EN).

[15] Spalart PR, Jou W-H, Strelets M, Allmaras SR. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. Liu, C. and Liu, Z. (eds.), *1st AFOSR International Conference on DNS/LES* 1997; Columbus OH, August, pp. 137–147, Greyden Press.

[16] Menter FR. Zonal two equation $k-\omega$ turbulence models for aerodynamic flows. *24th Fluid Dynamics Conference* 1993; Orlando, FL, July 6–9, AIAA paper no. 93-2906.

[17] Strelets M. Detached eddy simulation of massively separated flows. *39th AIAA Aerospace Sciences Meeting and Exhibit* 2001; Reno, NV, January 8–11, AIAA Paper 2001-0879.

[18] Réthoré P-E, Sørensen NN. Actuator disc model using a modified Rhie-Chow/simple pressure correction algorithm. Comparison with analytical solutions. *EWEC2008*, Brussel, BE, March 31-April 3 2008.

[19] Hansen MO. *Aerodynamics of Wind Turbines*. Earthscan: 2008.

[20] Willis GE, Deardorff JW. On the use of Taylor's hypothesis for diffusion in the mixed layer. *Quarterly Journal of the Royal Meteorological Society* 1976; **102**, 817–822.

[21] Spalart PR. Young-person's guide to detached eddy simulation grids. 2001, Tech. Rep. NASA/CR-2001-211032, Langley Research Center, NASA.

[22] Mann J. Wind field simulation. *Probabilistic Engineering Mechanics* 1998; **13**, 269–282.

[23] Shinozuka M, Jan CM. Digital simulation of random processes and its application. *Journal of Sound and Vibration* 1972; **25**, 111–128.

[24] Comte-Bellot G, Corrsin S. The use of a contraction to improve the isotropy of grid-generated turbulence. *Journal of Fluid Mechanics* 1966; **25**, 657–682.

[25] Gilling L, Sørensen NN, Davidson L. Detached eddy simulations of an airfoil in turbulent inflow. *47th AIAA Aerospace Sciences Meeting* 2009; Orlando, FL, January, AIAA paper no. 2009-270.

Paper III

Three Dimensional Turbulence Modeling in Shear Flow with Application to Wind Turbines

Lasse Gilling[1] and Bo Juul Pedersen[2]
and Søren R.K. Nielsen[1]

Submitted to *Wind Energy*.

[1] Aalborg University, Department of Civil Engineering, Aalborg, Denmark
[2] LAC Engineering ApS, Hinnerup, Denmark

Three Dimensional Turbulence Modeling in Shear Flow with Application to Wind Turbines

L. Gilling*†, B.J. Pedersen‡and S.R.K. Nielsen*

Abstract

In this paper the non-isotropic turbulence in the undisturbed atmospheric boundary layer flow is modeled in terms of the cross-spectral density function. The cross-spectral density function is determined for the three velocity components for any spatial separation in the vicinity of the rotor of a wind turbine. The turbulence field is assumed to be Gaussian, time-stationary and homogeneous over the rotor area. The approach is based on Townsend's rapid distortion theory for a frozen homogeneous, non-isotropic turbulence field in combination with Taylor's convection assumption. The core of the theory is the representation of various functions defining the azimuthal anisotropy of the turbulence field in terms of truncated low order Fourier series of the azimuthal angle. Then, the cross-spectral density functions may be obtained by solving a limited number of one-dimensional quadratures. In the numerical example it is demonstrated that the characteristics of turbulence in shear wind such as non-isotropic variances and the negative correlation between the along wind and vertical turbulence components can be modeled by the theory.

Keywords: Three-dimensional turbulence, shear flow, wind turbines, cross-spectral density function.

1 Introduction

Modern wind turbine wings are flexible and hence sensible to dynamic loads such as gusts caused by the turbulence of the wind field impinging on the rotor disk. The dynamic response may lead to both fatigue failure and collapse due to the yielding bending stresses or collision between the deflected blade and the tower. It follows that turbulence must be taken into consideration at the design of wind turbines. In the IEC 61400-1, Ed. 3.0 design code of practice for larger wind turbines, [1], the applied turbulence field is prescribed in terms of a frozen, homogeneous, non-isotropic turbulence field, which is convected into the rotor according to Taylor's hypothesis. The spectral turbulence tensor of the frozen field is taken as Townsend's solution for homogeneous shear flow based on rapid distortion theory, [2].

*Department of Civil Engineering, Aalborg University, Sohngaardsholmsvej 57, 9000 Aalborg, Denmark.

†E-mail: lg@civil.aau.dk, Phone: +45 9940 8570, Fax: +45 9940 8552

‡LAC engineering ApS, Bogøvej 15, DK-8382 Hinnerup.

The specification of design response in the code is based on Monte-Carlo simulations using the Mann and Krenk approach [3] of generating synthetic turbulence. A highly limited number of realizations (six to 15) are generated of the frozen turbulence field representing the stationary turbulence in a 10 minutes referential period. Next, the maxima of the structural response are sampled based on numerical time integration of the dynamic equations of motion. A rather arbitrarily chosen distribution function is fitted to the sampled values of the local maxima. Finally, this distribution function is extrapolated to yield the design response as the 50 year return value.

It turns out that the predicted maximal response depends severely on the chosen distribution function [4, 5]. Actually, the indicated procedure merely focus on the central part of the distribution, whereas the tail is not determined. Actually, the failure probability under normal operations is in the order of magnitude 10^{-7} during the referential 10 minutes period. Therefore, the indicated so-called crude Monte-Carlo method is not suitable for the estimation of such low failure probabilities.

Low failure probabilities of wind turbines may be estimated by so-called controlled Monte Carlo simulation methods [6, 7]. These are based on a state vector formulation of the turbulence field along with a state vector description of the structural system including control systems (pitch, yaw, rotational speed, etc.), so the integrated system forms a Markov vector. This rules out spectral representations of the turbulence field such as the Mann or Sandia models. Instead, a model based on rational filtration of Gaussian white noise is needed, with the filter parameters calibrated to the cross-spectral densities in the rotor area. This may be done by means of an ARMA vector model [8].

Veers [9] has suggested the use of empirical cross-spectral density functions for the time-stationary turbulence components between two points in the rotor plane, as has long time been the practice in civil engineering [10, 11, 12]. The applied spectra presume stochastic independence among the three turbulence components, however, with the consequence that the negative correlation between the along-wind and vertical turbulence components at a certain position is not modeled. Further, the influence of the phase spectrum is ignored, rendering the coherence functions and hence the cross-spectral densities real. Saranyasoontorn *et al.* [13] found the coherence of the rapid distortion theory to match their measurements better. Finally, the rapid distortion theory is employed by the method of Mann [14] which is recommended by the IEC61400 code. Therefore, it is chosen as basis for the model derived in this paper.

The idea of the present paper is to obtain an approximate but accurate representation of the said cross-spectral density functions derived from the code provisions, which form the basis of the calibration of the filter model of the turbulence process. The calibration procedure requires calculations of many function values of the spectra, for which reason a reduction in calculation time is desired.

Another application of the derived cross-spectral densities is generation of synthetic turbulence with the method in [15]. The method is intended for generating turbulence in a domain following a section of a rotating wind turbine blade. For CFD simulations of the flow past a section of a rotating wind turbine blade, the inflow turbulence is needed. The generated turbulence field should have a high spatial resolution close to the simulated section, while the turbulence in the rest of the rotor plane is not needed. For this application the methods of Mann and Krenk [3] or Veers [9] cannot be used. The

reason is that these methods need to generate turbulence in a much larger domain, than what is needed, and thereby the demand for computer memory becomes infeasible high.

The obtained solution is based on a representation of the various functions defining the azimuthal anisotropy in terms of truncated low order Fourier series of the azimuthal angle. Then, the cross-spectral density functions of the three-dimensional turbulence may be obtained by numerical solution of a limited number of one-dimensional quadratures. In a numerical example it is demonstrated that the spectral characteristics of turbulence in shear wind, such as anisotropic variances and correlation lengths of the three turbulence components, as well as the negative correlation between the along wind and vertical turbulence components, can be modeled by the theory.

2 Background and Definitions

In the present section the background for the turbulence model is given. It is based on the work by Townsend [2] and Mann and Krenk [3].

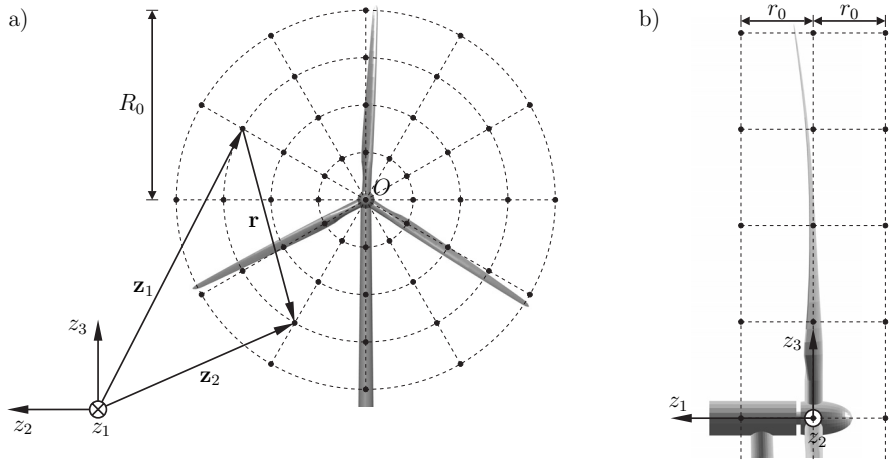


Figure 1: Rotational symmetric discretization of rotor area.

2.1 Stochastic Modeling of a Turbulence Field

The mean wind and the turbulence field are described in a Cartesian (z_1, z_2, z_3)-coordinate system with the origin placed somewhere in the rotor plane. The z_1-axis is horizontal, orthogonal to the rotor plane and oriented towards the nacelle. The z_3-axis is vertical and oriented in the upwards direction, see Figure 1. The three-dimensional turbulence field is modeled as a zero-mean, Gaussian time-stationary and spatial homogeneous stochastic process $\{v_j(\mathbf{z}, t),\ (\mathbf{z}, t) \in R^3 \times R\}$, where $v_j(\mathbf{z}, t)$ specifies the random turbulence in the coordinate direction j at the time t at the position defined by the position vector $\mathbf{z}$. Then,

the stochastic field is completely described by the cross-covariance function

$$\kappa_{jk}(\mathbf{z}_1, t_1; \mathbf{z}_2, t) = E\big[v_j(\mathbf{z}_1, t_1)v_k(\mathbf{z}_2, t_2)\big] = \kappa_{jk}(\mathbf{r}, \tau) = \kappa_{jk}(-\mathbf{r}, -\tau) \tag{1}$$

where $E[\cdot]$ denotes the expectation operator, and $\mathbf{r} = \mathbf{z}_2 - \mathbf{z}_1$ and $\tau = t_2 - t_1$ signify the spatial and temporal separations between the turbulence vectors.

For the spatial discretization shown on Figure 1 the first component r_1 of the separation vector $\mathbf{r}$ attains the value $-r_0, 0, r_0$. The cross-spectral density function is related to the cross-covariance function via the Wiener-Khintchine relation, see e.g. [16]

$$S_{jk}(\mathbf{r}, \omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{-i\omega\tau}\, \kappa_{jk}(\mathbf{r}, \tau)\, d\tau \tag{2}$$

Due to the symmetry properties of the cross-covariance function, the double-sided cross-spectral density function must fulfill the symmetry properties

$$S_{jk}(\mathbf{r}, \omega) = S^*_{kj}(-\mathbf{r}, \omega) = S^*_{jk}(\mathbf{r}, -\omega) \tag{3}$$

Taylor's hypothesis presumes that the fluctuations of the wind field at a given observation point approximately may be considered merely as the effect of the convection of a frozen turbulence field with the mean wind field V. Let $\tilde{v}_j(z_1, z_2, z_3) = v_j(z_1, z_2, z_3, 0)$ denote the turbulence field at the time $t = 0$, which is taken as the frozen field. Since the convection takes place in the positive z_1-direction this means that the turbulence at the position (z_1, z_2, z_3) at the time t has been positioned at $(z_1 - Vt, z_2, z_3)$ at the time $t = 0$. Hence, Taylor's hypothesis implies the functional relationship

$$v_j(z_1, z_2, z_3, t) = \tilde{v}_j(z_1 - Vt, z_2, z_3) \tag{4}$$

Then, use of (1) provides the following result for the cross-covariance function

$$\kappa_{jk}(\mathbf{r}, \tau) = E\big[\tilde{v}_j(\mathbf{z}_1 - Vt_1\mathbf{i}_1)\tilde{v}_k(\mathbf{z}_2 - Vt_2\mathbf{i}_1)\big] = \tilde{\kappa}_{jk}(r_1 - V\tau, r_2, r_3) \tag{5}$$

where $\mathbf{i}_1$ is the base unit vector along the z_1-axis, and $\tilde{\kappa}_{jk}(\mathbf{r})$ is the cross-covariance function of the frozen field defined as

$$\tilde{\kappa}_{jk}(\mathbf{r}) = E\big[\tilde{v}_j(\mathbf{z}_1)\tilde{v}_k(\mathbf{z}_2)\big] \tag{6}$$

Normally, the correlation structure of the frozen field is defined in terms of the spectral turbulence tensor $\Phi_{jk}(\mathbf{k})$ related to the cross-covariance function via the inverse Wiener-Khintchine relation

$$\tilde{\kappa}_{jk}(\mathbf{r}) = \int \mathrm{e}^{i\mathbf{k}\cdot\mathbf{r}}\, \Phi_{jk}(\mathbf{k})\, d\mathbf{k} \tag{7}$$

$\mathbf{k} = [k_1, k_2, k_2]$ denotes the wave number vector, and the scalar product $\mathbf{k}\cdot\mathbf{r}$ is evaluated as $k_1r_1 + k_2r_2 + k_3r_3$. In the following derivations it is convenient to introduce the quantity

$$\begin{aligned}\chi_{jk}(k_1, r_2, r_3) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{-ik_1r_1}\, \tilde{\kappa}_{jk}(r_1, r_2, r_3)\, dr_1 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{i(k_2r_2 + k_3r_3)}\, \Phi_{jk}(k_1, k_2, k_3)\, dk_2 dk_3\end{aligned} \tag{8}$$

At the derivation of the last statement it has been used that

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{i(k_1'-k_1)r_1}\, dr_1 = \delta(k_1'-k_1) \tag{9}$$

where $\delta(\cdot)$ is the Dirac delta function.

Insertion of (5) into (2) and use of (8) provides the following result for the cross-spectral density function

$$\begin{aligned} S_{jk}(\mathbf{r},\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{-i\omega\tau}\, \tilde{\kappa}_{jk}\big(r_1 - V\tau, r_2, r_3\big)\, d\tau \\ &= \frac{\mathrm{e}^{-i\frac{\omega}{V}r_1}}{V}\chi_{jk}^{*}\left(\frac{\omega}{V}, r_2, r_3\right) \end{aligned} \tag{10}$$

where $*$ denotes complex conjugation.

2.2 Sheared Spectral Turbulence Tensor

Townsend [2] obtained a solution for the double-sided spectral turbulence tensor for neutral atmospheric turbulence in shear flow. The basic assumptions are that the turbulence components are small compared to the mean wind velocity V, rendering linearization of the convective acceleration in Navier-Stokes equation possible (leaving "the rapid terms"), and that the mean wind over the rotor disk can be assumed to be varying linearly with height. Hence, the mean wind gradient $\frac{dV}{dz_3}$ evaluated at the hub is assumed to be constant. Next, the linearized Navier-Stokes equations are transferred to the wave number space, providing equations for the evolution in time of the Fourier transform of the turbulence components.

According to the Richardson cascade theory for the decay of grid generated turbulence the energy is generated at large wave numbers, and is transformed to and finally dissipated at smaller wave numbers in a way that larger vortices only interact with somewhat smaller vortices and so forth. At the same time the turbulence tends to become increasingly isotropic as sufficiently high scalar wave numbers are reached. In the equilibrium range, where the supplied turbulence energy from larger wave number balances the energy supplied to smaller wave numbers, this observation suggests that the anisotropic stationary state of vortices in the inertial sub-range of the equilibrium range, may be defined by the wave number vector $\mathbf{k} = \mathbf{k}\big(\mathbf{k_0}\big)$ obtained by starting from the wave number $\mathbf{k_0}$ in the isotropic state, and integrating the Navier-Stokes equations backwards in time during a certain time interval $\tau = \tau(k_0)$, which may be interpreted as the time interval needed for vortices with the scalar wave number $k = |\mathbf{k}|$ to return to isotropy via the cascade process. In what follows $\tau = \tau(k_0)$ will be referred to as the life time of vortices in the anisotropic state defined by the wave number $\mathbf{k} = \mathbf{k}\big(\mathbf{k_0}\big)$.

As a consequence of the linearization the anisotropic wave number vector $\mathbf{k}$ and the initial isotropic wave number $\mathbf{k}_0$ are linearly related via the mapping, [2]

$$\mathbf{k} = \mathbf{A}\big(\tau(k_0)\big)\,\mathbf{k}_0 \tag{11}$$

where

$$\mathbf{A}(\tau) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\beta(\tau) & 0 & 1 \end{bmatrix} \tag{12}$$

$$\beta(\tau) = \frac{dV}{dz_3}\tau \tag{13}$$

The final result for the anisotropic spectral turbulence tensor may be written as, [2]

$$\Phi_{jk}(\mathbf{k}) = B_{jl}(\mathbf{k}_0)\Phi_{lm,0}(\mathbf{k}_0)B_{km}(\mathbf{k}_0) \tag{14}$$

In (14) the summation convention has been applied for the dummy indices l and m, ranging from 1 to 3. $\Phi_{lm,0}(\mathbf{k}_0)$ denotes the spectral turbulence tensor for homogeneous, isotropic turbulence of an incompressible fluid given as, [17]

$$\Phi_{jk,0}(\mathbf{k}_0) = \frac{E(k_0)}{4\pi k_0^4}\big(\delta_{jk}k_0^2 - k_{j,0}k_{k,0}\big) \tag{15}$$

where δ_{jk} denotes the Kronecker delta.

$E(k_0)$ is the three-dimensional energy spectral function, for which von Karman suggested the following expression, asymptotically correct at small and at large scalar wave numbers corresponding to the inertial subrange

$$E(k_0) = \alpha\sigma_{\tilde{v}}^2 L\,\frac{(Lk_0)^4}{\big(1+(Lk_0)^2\big)^{\frac{17}{6}}} \quad , \quad \alpha = \frac{55}{9}\frac{\Gamma(\frac{5}{6})}{\Gamma(\frac{1}{2})\Gamma(\frac{1}{3})} \tag{16}$$

where $\sigma_{\tilde{v}}$ and L specify the standard deviation and the correlation length of the isotropic turbulence. $\Gamma(\cdot)$ is the gamma function.

$B_{ij}(\mathbf{k}_0)$ denotes components of the matrix $\mathbf{B}(t)$ given as

$$\mathbf{B}(\mathbf{k}_0) = \begin{bmatrix} 1 & 0 & B_1(\mathbf{k}_0) \\ 0 & 1 & B_2(\mathbf{k}_0) \\ 0 & 0 & B_3(\mathbf{k}_0) \end{bmatrix} \tag{17}$$

The non-dimensional functions $B_1(\mathbf{k}_0)$, $B_2(\mathbf{k}_0)$ and $B_3(\mathbf{k}_0)$ define the anisotropy of the turbulence field and are given as, [2]

$$B_1(\mathbf{k}_0) = k_{1,0}\frac{k_0^2}{M_0^2}\left(\frac{k_{3,0}}{k_0^2} - \frac{k_3(\tau)}{k^2(\tau)}\right) + \frac{k_{2,0}^2}{k_{1,0}}\frac{k_0^2}{M_0^3}\arctan\left(\frac{M_0\big(k_3(\tau)-k_{3,0}\big)}{M_0^2 + k_3(\tau)k_{3,0}}\right) \tag{18}$$

$$B_2(\mathbf{k}_0) = k_{2,0}\frac{k_0^2}{M_0^2}\left(\frac{k_{3,0}}{k_0^2} - \frac{k_3(\tau)}{k^2(\tau)}\right) - k_{2,0}\frac{k_0^2}{M_0^3}\arctan\left(\frac{M_0\big(k_3(\tau)-k_{3,0}\big)}{M_0^2 + k_3(\tau)k_{3,0}}\right) \tag{19}$$

$$B_3(\mathbf{k}_0) = \frac{k_0^2}{k^2(\tau)} \tag{20}$$

where

$$M_0 = \sqrt{k_{1,0}^2 + k_{2,0}^2} \tag{21}$$

In order to close the theory a model for the life time $\tau(k_0)$ is needed. Mann [18] discusses several possibilities and recommends

$$\tau(k_0) \propto k_0^{-2/3} \left[{}_2F_1\left(\frac{1}{3}, \frac{17}{6}; \frac{4}{3}, -(k_0 L)^{-2}\right)\right]^{-1/2} \tag{22}$$

which give the following expression for $\beta(k_0)$

$$\beta(k_0) = \beta_0 \left(L k_0\right)^{-2/3} \left[{}_2F_1\left(\frac{1}{3}, \frac{17}{6}; \frac{4}{3}, -(k_0 L)^{-2}\right)\right]^{-1/2} \tag{23}$$

where β_0 is a constant.

Then, the parameters entering the turbulence model consist of the standard deviation $\sigma_{\tilde{v}}$, the correlation length L of the isotropic referential turbulence, and the non-dimensional parameter β_0. $\sigma_{\tilde{v}}^2$ control the overall kinetic energy of the turbulence field, L controls the correlation lengths in the turbulence field, and β_0 controls the anisotropy responsible for the negative correlation between $\tilde{v}_1(\mathbf{z}, t)$ and $\tilde{v}_3(\mathbf{z}, t)$.

3 Semi-Analytical Solution for the Cross-Spectral Density Function

A semi-analytical solution for the cross-spectral density function will be derived in the following three subsections. First, the explicit dependence of the spectra on the wave number and the turbulence characteristics is derived in terms of a planar integral, which will be evaluated in polar coordinates. Next, approximations are introduced to allow analytical integration over the azimuthal angle. Finally, one-dimensional numerical integration is performed over the modulus to give the cross-spectral density function in semi-analytical form.

3.1 Integral Representation of the Cross-Spectral Density Function

Insertion of (14) into (8) and use of (11) provides the following solution for $\chi_{jk}(k_1, r_2, r_3)$, where the double integral is evaluated in the isotropic wave number space

$$\begin{aligned}
&\chi_{jk}\Big(k_1, r_2, r_3\Big) \; = \\
&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{i(k_2 r_2 + k_3 r_3)} B_{jl}(k_{n,0})\, \Phi_{lm,0}(k_1, k_{2,0}, k_{3,0})\, B_{km}(k_{n,0})\, dk_2 dk_3 \; = \\
&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{-ik_1 r_3 \beta(\tau)} \mathrm{e}^{i(k_{2,0} r_2 + k_{3,0} r_3)} B_{jl}(k_{n,0})\, \Phi_{lm,0}(k_1, k_{2,0}, k_{3,0})\, B_{km}(k_{n,0})\, dk_{2,0} dk_{3,0}
\end{aligned} \tag{24}$$

In (24) it has been used that $\det\big(\mathbf{A}(\tau)\big) = 1$, so $dk_{2,0}dk_{3,0} = dk_2dk_3$. For ease of notation the index '0' referring to quantities in the isotropic wave number space is skipped in what follows. Further, the evaluation of the double integral in (24) is performed in polar coordinates, for which reason the following polar representations are introduced

$$\left.\begin{array}{lll} r_2 = R\cos\theta & , & r_3 = R\sin\theta \\ k_2 = K\cos\psi & , & k_3 = K\sin\psi \end{array}\right\} \tag{25}$$

Correspondingly, B_{jl} and $\Phi_{lm,0}$ are now considered as functions of the isotropic scalar wave number $k_1 = k_{1,0} = \sqrt{k_1^2 + K^2}$, K and the azimuthal angle ψ. Insertion of (15) then provides

$$\begin{array}{l} \chi_{jk}(k_1, R, \theta) \;= \\ \displaystyle\int_0^\infty \mathrm{e}^{-ik_1 r_3 \beta(\tau)}\, \frac{E(k)}{4\pi k^4} \int_0^{2\pi} \Big(\mathrm{e}^{iKR\cos(\psi-\theta)} B_{jl}(k_n) \Big(\delta_{lm} k^2 - k_l k_m \Big) B_{km}(k_n) \Big)\, d\psi\, K dK \;= \\ \displaystyle\int_0^\infty \mathrm{e}^{-ik_1 R\sin\theta\, \beta(\tau)}\, \frac{E(k)}{4\pi k^4} \int_0^{2\pi} \mathrm{e}^{iKR\cos(\psi-\theta)}\, D_{jk}(k_1, K, \psi)\, d\psi\, K dK \end{array} \tag{26}$$

where the following tensor components have been introduced

$$D_{jk}(k_1, K, \psi) = k^2\, B_{jl}(k_1, K, \psi) B_{kl}(k_1, K, \psi) - B_{jl}(k_1, K, \psi) k_l\, B_{km}(k_1, K, \psi) k_m \tag{27}$$

where, cf. (17)

$$\left.\begin{array}{rl} D_{11}(k_1, K, \psi) = & k^2 \left(1 + B_1^2\right) - \big(k_1 + B_1\, k_3\big)^2 \\ = & k^2 \Big(1 + B_1^2 - \big(\gamma + B_1\, \delta \sin\psi\big)^2 \Big) \\ D_{12}(k_1, K, \psi) = & k^2\, B_1\, B_2 - \big(k_1 + B_1\, k_3\big)\big(k_2 + B_2\, k_3\big) \\ = & k^2 \Big(B_1 B_2 - \big(\gamma + B_1\, \delta \sin\psi\big)\big(\delta\cos\psi + B_2\, \delta \sin\psi\big) \Big) \\ D_{13}(k_1, K, \psi) = & k^2\, B_1\, B_3 - \big(k_1 + B_1\, k_3\big)\, B_3\, k_3 \\ = & k^2 \Big(B_1 B_3 - \big(\gamma + B_1\, \delta \sin\psi\big) B_3\, \delta \sin\psi \Big) \\ D_{22}(k_1, K, \psi) = & k^2 \left(1 + B_2^2\right) - \big(k_2 + B_2\, k_3\big)^2 \\ = & k^2 \Big(1 + B_2^2 - \big(\delta\cos\psi + B_2\, \delta \sin\psi\big)^2 \Big) \\ D_{23}(k_1, K, \psi) = & k^2\, B_2\, B_3 - \big(k_2 + B_2\, k_3\big)\, B_3\, k_3 \\ = & k^2 \Big(B_2 B_3 - \big(\delta\cos + B_3\, \delta \sin\psi\big) B_3\, \delta \sin\psi \Big) \\ D_{33}(k_1, K, \psi) = & \big(k^2 - k_3^2\big)\, B_3^2 = k^2 \Big(1 - \delta^2 \sin^2\psi \Big)\, B_3^2 \end{array}\right\} \tag{28}$$

In the final statements of (28) the following non-dimensional parameters have been introduced

$$\gamma = \frac{k_1}{k} \quad , \quad \delta = \sqrt{1 - \gamma^2} = \frac{K}{k} \tag{29}$$

Obviously, $|\gamma| \leq 1$. Correspondingly, the dependence of the anisotropy functions $B_j(k_1, K, \psi)$ on ψ follows by insertion of (25) into (18), (19) and (20)

$$\begin{aligned} B_1(k_1, K, \psi) &= k_1 \frac{k_0^2}{k_1^2 + k_2^2} \left(\frac{k_3}{k_0^2} - \frac{\beta k_1 + k_3}{k_0^2 + 2\beta k_1 k_3 + \beta^2 k_1^2} \right) \\ &\quad + \frac{k_2^2}{k_1} \frac{k_0^2}{(k_1^2 + k_2^2)^{3/2}} \arctan \left(\frac{\sqrt{k_1^2 + k_2^2}\beta k_1}{k_0^2 + \beta k_1 k_3} \right) \\ &= \frac{\gamma}{\gamma^2 + \delta^2 \cos^2 \psi} \left(\delta \sin \psi - \frac{\beta\gamma + \delta \sin \psi}{1 + 2\beta\gamma\delta \sin \psi + \beta^2\gamma^2} \right) \\ &\quad + \frac{\delta^2 \cos^2 \psi}{\gamma \left(\gamma^2 + \delta^2 \cos^2 \psi\right)^{3/2}} \arctan \left(\frac{\sqrt{\gamma^2 + \delta^2 \cos^2 \psi}\beta\gamma}{1 + \beta\gamma\delta \sin \psi} \right) \end{aligned} \tag{30}$$

$$\begin{aligned} B_2(k_1, K, \psi) &= k_2 \frac{k_0^2}{k_1^2 + k_2^2} \left(\frac{k_3}{k_0^2} - \frac{\beta k_1 + k_3}{k_0^2 + 2\beta k_1 k_3 + \beta^2 k_1^2} \right) \\ &\quad - k_2 \frac{k_0^2}{(k_1^2 + k_2^2)^{3/2}} \arctan \left(\frac{\sqrt{k_1^2 + k_2^2}\beta k_1}{k_0^2 + \beta k_1 k_3} \right) \\ &= \frac{\delta \cos \psi}{\gamma^2 + \delta^2 \cos^2 \psi} \left(\delta \sin \psi - \frac{\beta\gamma + \delta \sin \psi}{1 + 2\beta\gamma\delta \sin \psi + \beta^2\gamma^2} \right) \\ &\quad - \frac{\delta \cos \psi}{\left(\gamma^2 + \delta^2 \cos^2 \psi\right)^{3/2}} \arctan \left(\frac{\sqrt{\gamma^2 + \delta^2 \cos^2 \psi}\beta\gamma}{1 + \beta\gamma\delta \sin \psi} \right) \end{aligned} \tag{31}$$

$$B_3(k_1, K, \psi) = \frac{k^2}{k^2 + 2\beta\, k_1 k_3 + \beta^2 k_1^2} = \frac{1}{1 + 2\beta\gamma\delta\, \sin \psi + \beta^2\gamma^2} \tag{32}$$

The idea in the following is to approximate the tensor components $D_{jk}(k_1, K, \psi)$ in terms of truncated low-order Fourier series of the azimuthal angle ψ, from which an analytical solution to the innermost integral in (26) may be obtained.

3.2 Approximations for $D_{ij}(k_1, K, \psi)$

From this point two approaches are investigated. The first is to introduce an approximation for the anisotropy functions $B_j(k_1, K, \psi)$ in terms of truncated Fourier series in the azimuthal angle ψ, retaining harmonics up to 2nd order. The Fourier series are required to attain the exact values for $\psi = -\pi$, $-\frac{\pi}{2}$, 0, $\frac{\pi}{2}$, π. Next, these series are inserted in (28) to obtain a somewhat more involved Fourier series for $D_{ij}(k_1, K, \psi)$ using standard trigonometric identities. This will give an approximation with harmonics up to 6th order.

The second approach is to introduce similar low order truncated Fourier series as used for $B_j(k_1, K, \psi)$ on the components of $D_{ij}(k_1, K, \psi)$ directly, which will give an approximation with harmonics up to 2nd order.

In the first approach $B_j(k_1, K, \psi)$ is approximated by the following series

$$\left.\begin{aligned} B_1(k_1, K, \psi) &\simeq B_{1,0}(k_1, K) + B_{1,1}(k_1, K)\sin\psi + B_{1,2}(k_1, K)\cos 2\psi \\ B_2(k_1, K, \psi) &\simeq B_{2,1}(k_1, K)\cos\psi + B_{2,2}(k_1, K)\sin 2\psi \\ B_3(k_1, K, \psi) &\simeq B_{3,0}(k_1, K) + B_{3,1}(k_1, K)\sin\psi + B_{3,2}(k_1, K)\cos 2\psi \end{aligned}\right\} \tag{33}$$

where

$$\left.\begin{aligned} B_{1,0}(k_1, K) &= \frac{1}{4}\Big(B_1\Big(k_1, K, -\frac{\pi}{2}\Big) + 2B_1\Big(k_1, K, 0\Big) + B_1\Big(k_1, K, \frac{\pi}{2}\Big)\Big) \\ B_{1,1}(k_1, K) &= \frac{1}{2}\Big(B_1\Big(k_1, K, \frac{\pi}{2}\Big) - B_1\Big(k_1, K, -\frac{\pi}{2}\Big)\Big) \\ B_{1,2}(k_1, K) &= -\frac{1}{4}\Big(B_1\Big(k_1, K, -\frac{\pi}{2}\Big) - 2B_1\Big(k_1, K, 0\Big) + B_1\Big(k_1, K, \frac{\pi}{2}\Big)\Big) \\ B_{2,1}(k_1, K) &= B_2\Big(k_1, K, 0\Big) \\ B_{2,2}(k_1, K) &= \frac{1}{4}\Big(B_2\Big(k_1, K, \frac{\pi}{2}\Big) - B_2\Big(k_1, K, -\frac{\pi}{2}\Big)\Big) \\ B_{3,0}(k_1, K) &= \frac{1}{4}\Big(B_3\Big(k_1, K, -\frac{\pi}{2}\Big) + 2B_3\Big(k_1, K, 0\Big) + B_3\Big(k_1, K, \frac{\pi}{2}\Big)\Big) \\ B_{3,1}(k_1, K) &= \frac{1}{2}\Big(B_3\Big(k_1, K, \frac{\pi}{2}\Big) - B_3\Big(k_1, K, -\frac{\pi}{2}\Big)\Big) \\ B_{3,2}(k_1, K) &= -\frac{1}{4}\Big(B_3\Big(k_1, K, -\frac{\pi}{2}\Big) - 2B_3\Big(k_1, K, 0\Big) + B_3\Big(k_1, K, \frac{\pi}{2}\Big)\Big) \end{aligned}\right\} \tag{34}$$

The explicit expressions for the Fourier coefficients are obtained by insertion of (30), (31), (32) on the right-hand sides of (34). The resulting expressions become rather lengthy, and will not be indicated. Notice that the indicated truncated series become exact in the limits $\gamma = 0$ and $\gamma = 1$. The accuracy of the approximations have been demonstrated for $\gamma = \frac{1}{3}$ and $\gamma = \frac{2}{3}$ on Figure 2 for the realistic spectral values $Lk = 30$, $\beta_0 = 4$.

The corresponding truncated Fourier series expansions for the tensor components $D_{jk}(k_1, K, \psi)$ are obtained by inserting (33) into (28). These may be written on the form

$$D_{jk}(k_1, K, \psi) \simeq \sum_{n=0}^{6}\Big(c_{jkn}(k_1, K)\cos n\psi + s_{jkn}(k_1, K)\sin n\psi\Big) \tag{35}$$

The Fourier coefficients $c_{jkn}(k_1, K)$ and $s_{jkn}(k_1, K)$ are derived from (28) and (33) by using trigonometric identities. Several of the coefficients are zero. The results are rather lengthy and will not be given here.

In the second approach $D_{ij}(k_1, K, \psi)$ is approximated directly by truncated Fourier

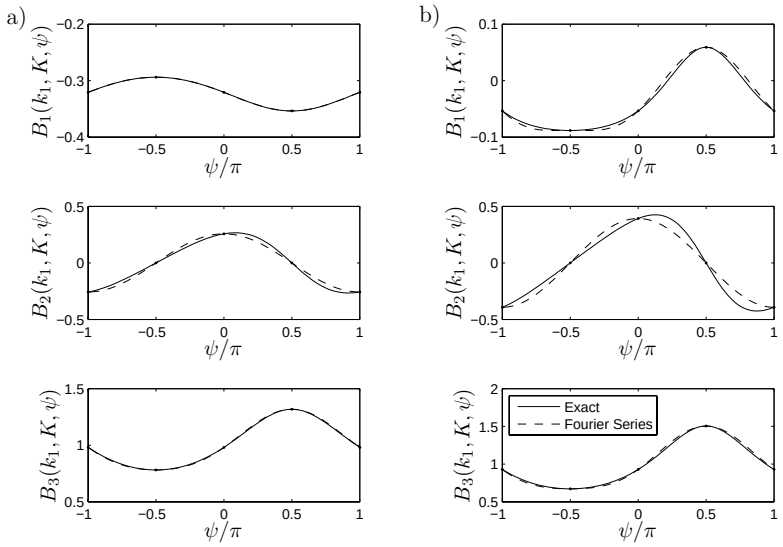


Figure 2: Exact results and approximate Fourier series for B_1, B_2 and B_3. $Lk = 30$ and $\beta_0 = 4$. a) $\gamma = \frac{1}{3}$. b) $\gamma = \frac{2}{3}$.

series with harmonics up to second order as in (33)

$$\left.\begin{aligned}
D_{11}(k_1,K,\psi) &\simeq D_{11,0}(k_1,K) + D_{11,1}(k_1,K)\sin\psi + D_{11,2}(k_1,K)\cos 2\psi \\
D_{22}(k_1,K,\psi) &\simeq D_{22,0}(k_1,K) + D_{22,1}(k_1,K)\sin\psi + D_{22,2}(k_1,K)\cos 2\psi \\
D_{33}(k_1,K,\psi) &\simeq D_{33,0}(k_1,K) + D_{33,1}(k_1,K)\sin\psi + D_{33,2}(k_1,K)\cos 2\psi \\
D_{12}(k_1,K,\psi) &\simeq D_{12,1}(k_1,K)\cos\psi + D_{12,2}(k_1,K)\sin 2\psi \\
D_{13}(k_1,K,\psi) &\simeq D_{13,0}(k_1,K) + D_{13,1}(k_1,K)\sin\psi + D_{13,2}(k_1,K)\cos 2\psi \\
D_{23}(k_1,K,\psi) &\simeq D_{23,1}(k_1,K)\cos\psi + D_{23,2}(k_1,K)\sin 2\psi
\end{aligned}\right\} \tag{36}$$

Hence, $D_{ij}(k_1,K,\psi)$ is approximated by

$$D_{jk}(k_1,K,\psi) \simeq \sum_{n=0}^{2}\Big(c_{jkn}(k_1,K)\cos n\psi + s_{jkn}(k_1,K)\sin n\psi\Big) \tag{37}$$

For $(i,j) = (1,1),(2,2),(3,3)$ and $(1,3)$ the non-zero coefficients are

$$\left.\begin{aligned}
c_{ij0}(k_1,K) &= \frac{1}{4}\left(D_{ij}\left(k_1,K,-\frac{\pi}{2}\right) + 2D_{ij}\left(k_1,K,0\right) + D_{ij}\left(k_1,K,\frac{\pi}{2}\right)\right) \\
s_{ij1}(k_1,K) &= \frac{1}{2}\left(D_{ij}\left(k_1,K,\frac{\pi}{2}\right) - D_{ij}\left(k_1,K,-\frac{\pi}{2}\right)\right) \\
c_{ij2}(k_1,K) &= -\frac{1}{4}\left(D_{ij}\left(k_1,K,-\frac{\pi}{2}\right) - 2D_{ij}\left(k_1,K,0\right) + D_{ij}\left(k_1,K,\frac{\pi}{2}\right)\right)
\end{aligned}\right\} \tag{38}$$

and for $(i,j) = (1,2)$ and $(2,3)$ the non-zero coefficients are

$$\left.\begin{aligned} c_{ij1}(k_1,K) = &\quad D_{ij}\left(k_1,K,0\right) \\ s_{ij2}(k_1,K) = &\quad \frac{1}{4}\left(D_{ij}\left(k_1,K,\frac{\pi}{2}\right) - D_{ij}\left(k_1,K,-\frac{\pi}{2}\right)\right) \end{aligned}\right\} \tag{39}$$

Both approximations of $D_{ij}(k_1,K,\psi)$ will give results close the exact solution for most values of k_1, K and ψ for realistic L and β_0.

Far the most combinations of k_1 and K will result in $D_{ij}(k_1,K,\psi)$ with a strong first or second order harmonic. However, there are also combinations for small values of kL that give a less simple variation as shown in Figure 3. In Figure 3(a) both the 2nd and 6th order approximations give results close to the exact solution, and here the 6th order is better than the 2nd order approximation. In a narrow set of k_1 and K the $D_{33}(k_1,K,\psi)$ component varies as shown in Figure 3(b). Here the 6th order gives a large overshoot, and when integrated as in (26) this will result in too high spectral density function.

Hence, in general the 6th order approximation is closer to the exact solution, but for some parameters it gives a very inaccurate result. The 2nd order solution is often slightly less accurate, but it is more stable and gives at least a fair approximation for all combinations of k_1 and K. For its simplicity and robustness the second order approximation defined by (37), (38) and (39) will be used throughout from this point.

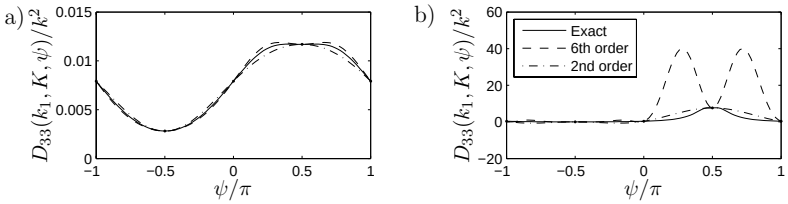


Figure 3: Comparison of 2nd and 6th order approximations to exact solution for $D_{33}(k_1,K,\psi)$. $kL = 1$ and $\beta_0 = 4$. a) $\gamma = 0.8$ and b) $\gamma = 0.2$.

3.3 Analytical Evaluation of Integrals

Based on the derived Fourier series expansion for $D_{ij}(k_1,K,\psi)$ the innermost integral in (26) may be evaluated analytically using the following identities, which have been derived by simple manipulations with the Bessel integral, [19]

$$\left.\begin{aligned} \int_0^{2\pi} \mathrm{e}^{iKR\cos(\psi-\theta)}\cos(n\psi)\,d\psi &= 2\pi i^n \cos(n\theta)J_n(KR) \\ \int_0^{2\pi} \mathrm{e}^{iKR\cos(\psi-\theta)}\sin(n\psi)\,d\psi &= 2\pi i^n \sin(n\theta)J_n(KR) \end{aligned}\right\} \tag{40}$$

$J_n(z)$ denotes the Bessel functions of the order n and the first kind. Then, insertion

of (37) into (26) provides

$$\chi_{jk}(k_1, R, \theta) \simeq \sum_{n=0}^{2} \left(C_{jkn}\left(R, k_1, \theta\right) \cos(n\theta) + S_{jkn}\left(R, k_1, \theta\right) \sin(n\theta) \right) \tag{41}$$

and thereby the following result for the cross-spectral density function is obtained

$$S_{jk}(r_1, R, \theta, \omega) = \frac{\mathrm{e}^{-i\frac{\omega}{V}r_1}}{V} \sum_{n=0}^{2} \left(C^*_{jkn}\left(R, \frac{\omega}{V}, \theta\right) \cos(n\theta) + S^*_{jkn}\left(R, \frac{\omega}{V}, \theta\right) \sin(n\theta) \right) \tag{42}$$

where C_{jkn} and S_{jkn} denote the following Hankel transforms with respect to K

$$\left.\begin{aligned} C_{jkn}(R, k_1, \theta) &= i^n \int_0^\infty \mathrm{e}^{-ik_1 R \sin\theta\, \beta(\tau)} \frac{E(k)}{2k^4}\, c_{jkn}(k_1, K)\, J_n(KR)\, K dK \\ S_{jkn}(R, k_1, \theta) &= i^n \int_0^\infty \mathrm{e}^{-ik_1 R \sin\theta\, \beta(\tau)} \frac{E(k)}{2k^4}\, s_{jkn}(k_1, K)\, J_n(KR)\, K dK \end{aligned}\right\} \tag{43}$$

The Hankel integrals entering (43) must be carried out numerically.

Especially, the double-sided auto-spectral density functions follow for $r_1 = R = 0$

$$S_{(j)(j)}(\omega) = \frac{1}{V} C_{(j)(j)0}\left(0, \frac{\omega}{V}, 0\right) \tag{44}$$

where the parentheses around the indices indicate that the summation convention is abandoned.

4 Numerical Examples

To verify the semi-analytical solution given by (42) it is compared to the exact result determined by numerical integration of (26). First, the auto-spectra will be shown. Further, the correlation coefficient between $v_1(t)$ and $v_3(t)$ is determined along with the ratios of standard deviations of the three velocity components. These values are determined for $r_1 = R = 0$. Finally, the cross-spectra are shown for $R = L/2$.

For each value of k_1 the semi-analytical model needs to integrate numerically over K, while the purely numerical integration of (26) needs to integrate over both K and ψ. The most costly operation is the many evaluations of $B_j(k_1, K, \psi)$ involved in both approaches. With the semi-analytical model only three values are needed for each set of (k_1, K), namely $\psi = \frac{\pi}{2}(-1, 0, 1)$. To obtain accurate results with the purely numerical integration a minimum of 20 values of $B_j(k_1, K, \psi)$ should be evaluated at different ψ for each combination of (k_1, K). The total simulation time for a single cross-spectral density function is in the order of $20 - 40$ seconds on a modern desktop computer. If the rotor plane is resolved into 50 points a total of 2500 auto- and cross-spectra should be computed, so the total simulation time is several hours. Therefore, a reduction in total computation time is desired, which can be obtained by use of the semi-analytical model.

4.1 Auto-Spectral Density

In Figure 4 the double-sided auto-spectra evaluated by (44) is compared to the exact solution evaluated by numerical integration of (26). The spectra are shown for the turbulence components $v_1(t)$, $v_2(t)$ and $v_3(t)$ and have been normalized with respect to the isotropic variance σ_v^2. Further, the cross-spectral density $S_{13}(\omega)$ between $v_1(t)$ and $v_3(t)$ is shown. For $r_1 = R = 0$ all the spectra are real. The spectra are symmetric and therefore results are only shown for positive frequencies. The dependency of V and L is eliminated in the normalization.

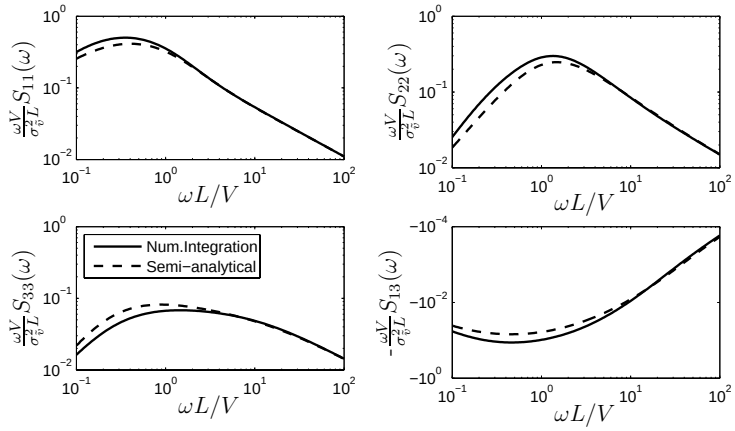


Figure 4: Exact vs. semi-analytical spectra for $\beta_0 = 6.5$ and $r_1 = R = 0$.

The agreement between the exact and semi-analytical solution is seen to be good. The differences between the semi-analytical and exact solution are smaller than the difference between the exact solution and the model spectra by Simiu and Scanlan [20] shown by Mann in [14]. Further, it is noticed that $S_{13}(\omega)$ is negative for all frequencies. For the applications outlined in the introduction, the important part of the spectral densities is the linear region, as the eigenfrequencies of the wind turbine are located in this region.

Using the semi-analytical approach the ratios of standard deviations of the three velocity components are

$$\left.\begin{aligned} \frac{\sigma_{v_2}}{\sigma_{v_1}} &= 0.71 \\ \frac{\sigma_{v_3}}{\sigma_{v_1}} &= 0.50 \end{aligned}\right\} \tag{45}$$

for $\beta_0 = 6.5$. These values match the ratios listed in the IEC 61400-1 code [1]. The correlation between the along wind and vertical components is

$$\rho_{\tilde{v}_1\tilde{v}_3} = \frac{\int_\infty^\infty S_{13}(0,0,0,\omega)d\omega}{\left(\int_\infty^\infty S_{11}(\omega)d\omega \int_\infty^\infty S_{33}(\omega)d\omega\right)^{1/2}} = -0.34 \tag{46}$$

4.2 Cross-Spectral Density

In Figure 5 the cross-spectra from (42) and numerical integration of (28) are compared for a separation distance of $R = 0.5L$ and $\theta = \frac{1}{8}\pi$. Here, all terms of (42) are active and the spectra contain both real and imaginary parts which are shown in the figure. The real parts are symmetric functions and the imaginary parts are skew-symmetric functions of frequency, and therefore the results are only shown for positive frequencies. The agreement is seen to be good and the differences are seen to be smaller than the differences reported in [14].

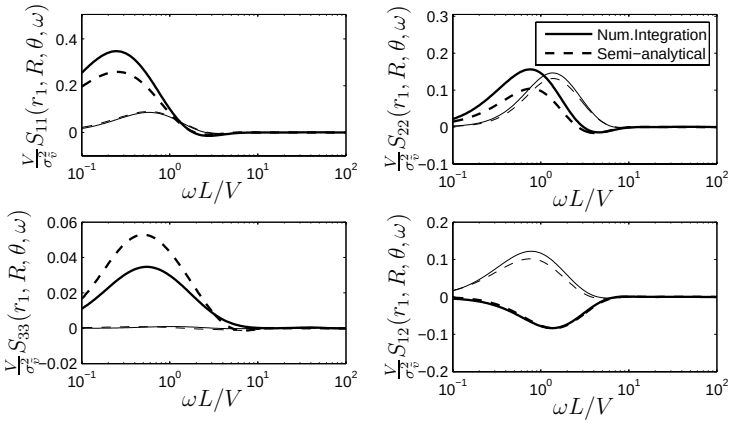


Figure 5: Exact vs. semi-analytical solution for $\beta_0 = 6.5$, $L = 30\,\mathrm{m}$, $r_1 = 0$, $R = L/2$ and $\theta = \frac{1}{8}\pi$. Thick lines mark real part and thin lines mark imaginary part.

5 Conclusions

A semi-analytical solution is derived for the cross-spectral density function. Along one dimension the integral is evaluated analytically leaving a single dimension for numerical integration. This reduces the computational cost considerably as shown in the numerical example.

In the numerical example it was verified that the ratios of standard deviations of the along-wind and transverse directions were identical to the targets set by the IEC 61400-1 code [1]. The auto- and cross-spectra showed good agreement with the exact solution. Further, the correlation between the along-wind and vertical component was -0.34. In conclusion, the presented semi-analytical model can generate the cross-spectral densities of turbulence according to the theory recommended in the IEC61400-1 code at a lower cost than by using purely numerical integration of the shear spectral tensor.

Acknowledgement

The Danish Energy Authority is acknowledged for support under the grant EFP07-II, Estimation of Extreme Responses and Failure Probability of Wind Turbines under Normal Operation by Controlled Monte Carlo Simulation. Further, the Danish Council for Strategic Research supported the work under the project Airfoils in Turbulent Inflow with project no. 2104-05-0074.

References

[1] IEC 61400-1, Ed. 3.0: *Wind Turbines - Part 1: Design Requirements*. Reference number IEC 61400-1:2005(E).

[2] Townsend AA. *The Structure of Turbulent Shear Flow*, 2nd Ed. Cambridge University Press, 1976.

[3] Mann J, Krenk S. Fourier simulation of a non-isotropic wind field model. *Proceedings of the 6th ICOSSAR*, Innsbruck, Austria, August 9-13, 1993, 1669-1674. Balkema, Rotterdam, 1993.

[4] Moriarty PJ, Holley WE, Butterfield SP. Extrapolation of Extreme and Fatigue Loads Using Probabilistic Methods. Technical report, NREL NREL/TP-500-34421, National Renewable Energy Laboratory, Colorado, 2004.

[5] Ragan P, Manuel L. Statistical extrapolation methods for estimating wind turbine extreme loads. *Journal of Solar Energy Engineering* 130, DOI: 10.1115/1.2931501. 2008.

[6] Pradlwarter HJ, Schueller GI. Assessment of Low Probability Events of Dynamical Systems by Controlled Monte Carlo Simulation. *Probabilistic Engineering Mechanics*, **14**, 213–227, 1999.

[7] Bucher C. An importance sampling technique for randomly excited systems discretized by finite elements. *Advances in Structural Dynamics*, **II**, 1135–1142, 2000.

[8] Sichani MT, Pedersen BJ, Nielsen SRK. State space modelling of along wind turbulence with application to wind turbines. Submitted to *Probabilistic Engineering Mechanics*, 2009.

[9] Veers PS. *Three-Dimensional Wind Simulation*. Technical report, SAND88-0152 UC-261, Sandia National Laboratories, Albuquerque, 1988.

[10] Choi ANL. Correlation and spectral functions of atmospheric turbulence. *Proceedings of the 3rd Conference on Wind Effects on Buildings and Structures*, Tokyo, 1971.

[11] Shiotani M, Iwayani Y. Correlation of wind velocities in relation to the gust loadings. *Proceedings of the 3rd Conference on Wind Effects on Buildings and Structures*, Tokyo, 1971.

[12] Kaimal JC, Wyngaard JC, Izumi Y, Coté OR. Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, **98**(417), 563-589, 1972.

[13] Saranyasoontorn K, Manuel L, Veers PS. A comparison of standard coherence models for inflow turbulence with estimates from field measurements. *Journal of Solar Energy Engineering*, **126**, 1069–1082, 2004.

[14] Mann J. Wind field simulation. *Probabilistic Engineering Mechanics*, **13**(4), 269-282, 1998.

[15] Gilling L, Nielsen SRK, Sørensen NN. Generation of synthetic turbulence in arbitrary domains. *ICOSSAR 2009*, 13–17 September, Osaka, Japan.

[16] Papoulis A. *Probability, Random Variables and Stochastic Processes*, 3rd Ed. McGraw-Hill, 1991.

[17] Batchelor GK. *The Theory of Homogeneous Turbulence*. Cambridge University Press, 1953.

[18] Mann J. The spatial structure of neutral atmospheric turbulence. *Journal of Fluid Mechanics*, **273**, 141-168, 1994.

[19] Watson GN. *A Treatise on the Theory of Bessel Functions*. 2nd Ed. Cambridge University Press, 1966.

[20] Simiu E, Scanlan RH. *Wind Effects on Structures*. 2nd Ed. John Wiley and Sons, 1986.

Paper IV

Generation of Synthetic Turbulence in Arbitrary Domains

Lasse Gilling[1], Søren R.K. Nielsen[1] and Niels N. Sørensen[2]

10th International Conference on Structural Safety and Reliability, *ICOSSAR2009*, 13-19 September, Osaka, Japan

[1] Aalborg University, Department of Civil Engineering, Aalborg, Denmark
[2] Risø-DTU National Laboratory for Sustainable Energy, Wind Energy Division, Roskilde, Denmark

Safety, Reliability and Risk of Structures, Infrastructures and Engineering Systems – Furuta, Frangopol & Shinozuka (eds)
© 2010 Taylor & Francis Group, London, ISBN 978-0-415-47557-0

Generation of synthetic turbulence in arbitrary domains

L. Gilling & S.R.K. Nielsen
Department of Civil Engineering, Aalborg University, Aalborg, Denmark

N.N. Sørensen
Wind Energy Division, National Laboratory for Sustainable Energy, Risø-DTU, Technical University of Denmark, Roskilde, Denmark

ABSTRACT: A new method for generating synthetic turbulence is presented. The method is intended for generating a turbulent velocity field with a fine spatial resolution but only covering a small moving part of the rotor area of a wind turbine. For this application the Mann and Sandia methods cannot be used because of the very high requirements for computer memory. In the present method the auto- and cross-correlation in all three directions is computed from analytical or empirical expressions and the auto- and cross-spectra are determined by using the Wiener-Khinchin relation. From the auto- and cross-spectra a realization of a velocity field is determined by factorization and Fourier transform as in the Sandia method.

1 INTRODUCTION

Recent work investigated the importance of resolving the inflow turbulence in computational fluid dynamics simulations of airfoil flows, c.f. Gilling et al. (2009). The applied simulation technique was a combination of detached eddy simulation (DES) and large eddy simulation (LES). The conclusions from the paper includes that the agreement with results from an experiment is improved by resolving the inflow turbulence in the simulation. The improvement is largest close to stall and consequently in the range where most wind turbines operate. The paper was concerned with a section of a blade in a wind tunnel and not a full rotor in the atmospheric boundary layer. This was chosen to reduce the computational costs.

As described by Connell (1982) a rotating blade experiences turbulence with a different auto-spectrum than the turbulence experienced by a stationary point. Therefore, it would be interesting to examine the aerodynamic performance of a section of a blade on a rotor. It is infeasible to use a computational grid that is fine enough for a LES for the entire wind turbine.

Instead the overset grid method applied by Zahle (2006) and Zahle & Sørensen (2007, 2008) can be used. Here it is possible to use a coarse background grid and a fine grid that follows the blades. The background grid resolves only the mean flow and very large scales around the entire wind turbine. The fine grid moves with the blades and resolves the much smaller scales close to the blade. The fine grid is not fine enough to run a DES with resolved inflow turbulence, as this would require a very large number of cells. By including an even finer grid that only covers a limited part of a single blade the resolution here can be fine enough to run in LES-mode, and the effect of inflow turbulence on this section can be investigated. The inflow turbulence required in such a simulation is shown in Figure 1.

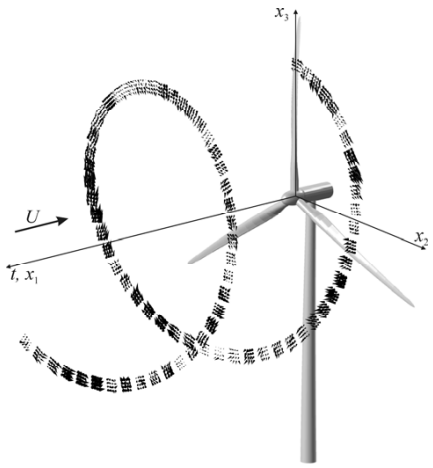


Figure 1. Example of the inflow turbulence for the LES part of the computational domain. Each of the cones represents a velocity vector. The resolution here is only 4 × 4 in the x_2-x_3-plane. U is the mean wind velocity.

This paper is concerned by generating the synthetic inflow turbulence in a domain as shown in the figure. The IEC61400-1 gives two different ways of generating the turbulent velocity field. One is the method by Mann (1994, 1998) and the other is the Sandia method by Veers (1984). Both methods are intended for generating turbulence in a domain covering the entire rotor plane.

For the application outlined above the turbulence is only needed in a small fraction of the rotor area. By only generating the velocity field in the points where it is needed considerable savings can be obtained.

In the present work a homogenous, isotropic velocity field is generated in an arbitrary domain. The domain is resolved into a number of discrete points. The points are allowed to move in time, which is the main difference from the Sandia method. Thereby, velocity fields like the one shown in Figure 1 can be generated, and the domain consists of only 16 points moving in time.

The cross- and auto-spectra are obtained from cross- and auto-correlation by the Wiener-Khinchin relations. That makes it possible to generate the velocity vectors only in the points where they are needed. The generated field will have the correct cross- and auto-spectra in all three directions.

In this paper the method for generating the synthetic turbulence is outlined. The description is followed by a comparison with the Mann and Sandia methods. Examples of generated velocity fields and their spectra and correlation are presented prior to the conclusions.

2 THE METHOD

The velocity field is generated in a number of points moving in time $\mathbf{x}_i(t)$ in three-dimensional space. There are N points and i=1,2,…,N. The orientations of the axes are defined in Figure 1. In each of the points a three dimensional velocity vector $\mathbf{v}_i(t)$ is generated.

Only homogenous and isotropic velocity fields are considered. The correlation tensor of two velocity vectors in a frozen field separated by the vector **r** is defined as

$$R_{ij}(\mathbf{r}) = \overline{v_i(\mathbf{x}) v_j(\mathbf{x}-\mathbf{r})} \tag{1}$$

for i,j=1,2,3. The overbar is used to denote ensemble average. For isotropic turbulence the components of the correlation tensor can be determined by, c.f. Connell (1982)

$$\mathbf{R}(\mathbf{r}) = \frac{R_a - R_l}{r^2}\begin{bmatrix} r_1^2 & r_1 r_2 & r_1 r_3 \\ r_2 r_1 & r_2^2 & r_2 r_3 \\ r_3 r_1 & r_3 r_2 & r_3^2 \end{bmatrix} + R_l \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where $\mathbf{r}=[r_1,r_2,r_3]$ is the separation vector, $r=|\mathbf{r}|$ is the separation distance, R_a is the spatial correlation between components in the **r** direction and R_l is the spatial correlation in the directions perpendicular to **r**. For isotropic turbulence von Karman (1948) shoved that

$$R_a(r) = \frac{2\sigma^2}{\Gamma(1/3)}\left(\frac{r}{2L}\right)^{1/3} K_{1/3}\left(\frac{r}{L}\right) \tag{3}$$

and

$$\begin{aligned} R_l(r) &= R_a(r) + \frac{r}{2}\frac{\mathrm{d}\,R_a(r)}{\mathrm{d}\,r} \\ &= \frac{2\sigma^2}{\Gamma(1/3)}\left(\frac{r}{2L}\right)^{1/3}\left(\mathrm{K}_{1/3}\left(\frac{r}{L}\right) - \frac{r}{2L}\mathrm{K}_{2/3}\left(\frac{r}{L}\right)\right) \end{aligned} \tag{4}$$

where L is a length scale, σ^2 is the variance of the turbulence, Γ is the gamma function and K_v is the modified Bessel function of second kind and the order v. The method presented here is not limited to this choice of correlation functions.

By applying Taylor's hypothesis the separation in time is transformed to separation along the mean wind direction. The separation vector **r** is therefore

$$\mathbf{r} = \left[r_1, r_2, r_3\right] = \left[\Delta x_1 + \Delta t U, \Delta x_2, \Delta x_3\right] \tag{5}$$

where the orientations of U, x_1 and t are given in Figure 1.

A correlation matrix $\mathbf{K}(t) = \mathbf{K}(\mathbf{x}(t))$ is assembled to contain the auto- and cross-correlation for all points $\mathbf{x}_i(t)$

$$\mathbf{K}(t) = \begin{bmatrix} \mathbf{R}(\mathbf{x}_{1,t}-\mathbf{x}_{1,0}) & \mathbf{R}(\mathbf{x}_{2,t}-\mathbf{x}_{1,0}) & \cdots & \mathbf{R}(\mathbf{x}_{N,t}-\mathbf{x}_{1,0}) \\ & \mathbf{R}(\mathbf{x}_{2,t}-\mathbf{x}_{2,0}) & \cdots & \mathbf{R}(\mathbf{x}_{N,t}-\mathbf{x}_{2,0}) \\ \text{sym.} & & \ddots & \vdots \\ & & & \mathbf{R}(\mathbf{x}_{N,t}-\mathbf{x}_{N,0}) \end{bmatrix} \tag{6}$$

where $\mathbf{x}_{i,t} = \mathbf{x}_i(t)$ is the coordinates to the i'th point in the domain at the time t. It has the dimensions $3N \times 3N$.

The spectral matrix $\mathbf{S}(f)$ is determined from the Wiener-Khinchin relation

$$\mathbf{S}(f) = \int_{-\infty}^{\infty} \mathbf{K}(t) e^{-j2\pi f t}\,\mathrm{d}\,t \tag{7}$$

where f is frequency and j is the imaginary unit number.

As in the Sandia method the phase information is discarded in the spectral matrix $\mathbf{S}(f)$. This will not affect the correlation of the generated turbulence. Consequently, $\mathbf{S}(f)$ is symmetric and $\mathbf{H}(f)$ is real. Now, it is possible to factorize $\mathbf{S}(f)$ by using an eigenvalue decomposition to give a matrix $\mathbf{H}(f)$ satisfying

$$\mathbf{S}(f) = \mathbf{H}^*(f)\mathbf{H}^{\mathrm{T}}(f) \tag{8}$$

The velocities are determined from the Fourier transform

$$\mathbf{v}_i(t) = \int_{-\infty}^{\infty} e^{i2\pi ft}\mathbf{H}(f)\,\mathrm{d}\mathbf{W}(f) \tag{9}$$

where d**W***(f)* is a vector of independent, complex Gaussian random numbers with zero mean and standard deviation $2^{-1/2}$.

When the correlation tensor is given by (2)-(4) the generated turbulence will be isotropic. In principle it is straightforward to simulate sheared velocity fields as in the method of Mann, as only (2)-(4) should be substituted for anisotropic expressions. However, to the knowledge of the authors, the correlation tensor of a sheared velocity field has not been given in analytical form in the literature.

3 COMPARISON TO THE MANN AND SANDIA METHODS

3.1 *The Sandia method*

The method outlined above has many similarities with the Sandia method. The difference is in how the spectral matrix is determined. In the Sandia method the spectral matrix is determined analytically from a coherence function $\gamma_{ij}(f,r)$. If the three directional components of the wind field are assumed to be uncorrelated the entries in the spectral matrix is determined from

$$\left|S_{ij}(f)\right|^2 = \gamma_{ij}^2(f,r)S_{jj}(f)S_{ii}(f) \tag{10}$$

where S_{ii} is the auto-spectrum in the i'th point. This spectral matrix is factored, multiplied by complex random numbers and Fourier transformed as in the method of this paper.

With the Sandia method the rotor plane is resolved into a number of points where the three components of the velocity vectors are determined. There is no restriction on how the points can be distributed over the rotor plane, so the points can be clustered e.g. in the projection of the spiral in Figure 1 on the x_2-x_3-plane.

Veers (1988) introduced a modification to the method intended for generating a velocity field for an aero-elastic simulation of a rotor. Instead of generating the velocity vectors in all points at all time steps it is possible to determine the velocities only at the instances where the blade passes each point. This approach can also be used to generate a velocity field as shown in Figure 1. The drawback of this modification is that the full spectral matrix still has to be stored.

3.2 *The Mann method*

In the method of Mann a three-dimensional frozen velocity field is generated based on the rapid distorsion theory of Townsend (1976). All three directional velocity components can be generated. The method is based on the method of Shinozuka & Jan (1972) and uses three-dimensional fast Fourier transform. Consequently, the points in the discrete domain must be equidistantly spaced and it is not possible to cluster points in the areas where the velocity field is needed.

The velocity field is periodic in all three directions. Therefore, the transverse dimensions of the domain should be larger than the structure of interest to avoid an unphysical high correlation from one end of the structure to the other.

The Mann method can generate anisotropic velocity fields that resemble the atmospheric turbulence better than the isotropic turbulence fields generated by the Sandia method and the method of this paper. When the Mann method is used for generating isotropic velocity fields the fields are divergence free, which is also the case for the present method.

3.3 *Memory requirements*

The computational time required for generating a velocity field by the three methods is within a few seconds to less than an hour depending of the size of the problem. Therefore, the computational time is not a severe limitation for either of the methods. In Example 1 (see Section 4.1) the computational time is given for a small example generated with the present method.

The memory requirement depends linearly on the size of the system matrix and can be a severe limitation as will be shown in Example 1. For the method of this paper the number of entries is

$$N_{el} = (3N)^2 N_t \tag{11}$$

where N_t is the number of time steps. For the Mann and Sandia methods velocities are generated in a domain covering the entire rotor area. Let the number of points in these methods be *M*. Now, the number of entries in the system matrix of the Sandia method is

$$N_{Sandia} = (3M)^2 N_t \tag{12}$$

The factor 3 can be deleted if the three velocity components are allowed to be uncorrelated. For the Mann method the number of entries is

$$N_{Mann} = 3MN_t \tag{13}$$

In applications where only a small number of points in the rotor plane are of interest *N* will be much smaller than *M*. In these applications, it is seen

that the method described in this paper has much lower memory requirements than the Sandia method. If N^2 is smaller than M the present method will also have lower memory requirements than the Mann method. An example of this is given in Example 1 below.

If the limitation on computer memory does not allow using a resolution of the synthetic turbulence as fine as the LES grid interpolation can be used. This does not resolve the small eddies in the generated turbulence which then will be a poor approximation to the atmospheric turbulence. If a very coarse resolution is used the airfoil will experience the synthetic turbulence as a variation in mean wind rather than real turbulence.

4 EXAMPLES

Below three examples of the use of the model is given. One generates turbulence in a domain like the one shown in Figure 1 and the computational cost is compared to the cost of the Mann method. The second example generates turbulence along a single rotating blade. The spectrum of velocities sampled at the tip is compared to the spectrum from Connell (1988) and the correlation of velocities along the blade is compared to (4). Finally, the versatility of the method is shown by generating the turbulence field experienced by a wind turbine with accelerating rotational speed.

4.1 *Example 1: Overlapping mesh domain*

Turbulence is generated in a domain resembling the one shown in Figure 1. The cross section of the domain is 1 m × 1 m and the radius of the spiral is 40 m. Turbulence is created for a single revolution of the rotor. The cross section is resolved into 8 × 8 points. The resolution in time is 512 points for the 3.9 s revolution time giving a time step of 7.7×10^{-3}. The length scale of the turbulence is $L = 42$ m.

In Figure 2 the generated velocity field is illustrated at three different instances. It is observed that the generated velocity vectors have a high correlation due to the short separation distance between them.

The system matrix used in this example contains $(3 \times 8^2)^2 \times 512 = 19 \times 10^6$ entries corresponding to 72 MB of ram if the entries are stored in single precision.

To generate a velocity field by the method of Mann it would be necessary to generate a field covering the entire rotor. To obtain the resolution in this example it would be required to use $M = 1024 \times 1024$ points in the rotor plane giving 537×10^6 entries in the system matrix. It would require 4.3 GB ram in single precision as these entries are complex.

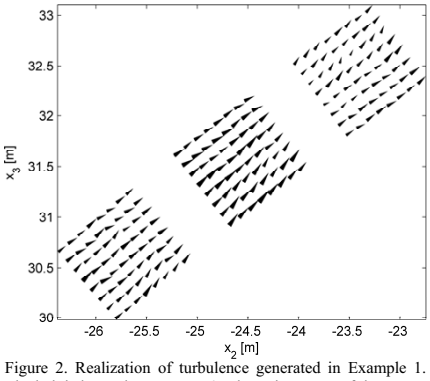


Figure 2. Realization of turbulence generated in Example 1. The hub is located at $x_2 = x_3 = 0$. Three time steps of the generated turbulence are shown. The separation time is 0.2 s. The cones are oriented in the directions of the velocity vectors and scaled by their lengths.

The computational time for the present example is about 3.5 minutes on a standard modern laptop. The algorithm has been implemented in MATLAB without special attention to computational efficiency.

4.2 *Example 2: A single blade*

In this example turbulence is created along a single blade rotating with the rotor. The first two revolutions of the generated spiral are shown in Figure 3. In total there are 26 revolutions in 1024 time steps. The mean wind speed is 15 m/s.

The length of the blade is 50 m and the turbulence length scale is 42 m.

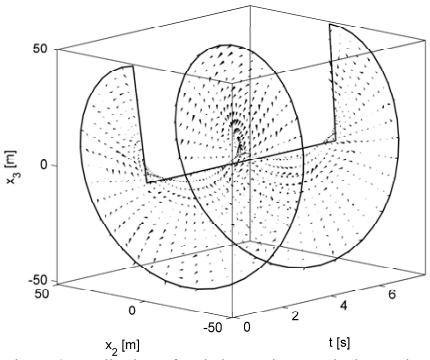


Figure 3. Realization of turbulence along a single rotating blade generated in Example 2.

In Figure 4 the correlation of the streamwise velocity component v_1 along the blade is compared to $R_l(r)$ given in (4).

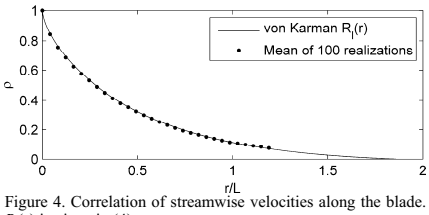


Figure 4. Correlation of streamwise velocities along the blade. $R_l(r)$ is given in (4).

In Figure 5 the power spectrum of streamwise velocity sampled along the tip of the blade is shown and compared to the spectrum by Connell (1982) and the von Karman spectrum at a stationary point.

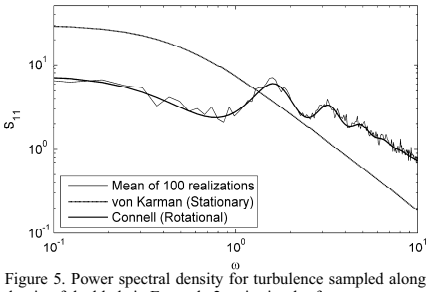


Figure 5. Power spectral density for turbulence sampled along the tip of the blade in Example 2. ω is circular frequency.

4.3 *Example 3: Full wind turbine*

The final example serves to show the versatility of the method. Here, turbulence is generated along the path a wind turbine covers while the rotational speed is accelerated.

The domain contains stationary points and points rotating with variable angular velocity. The generated turbulence has the correct correlation in space and time.

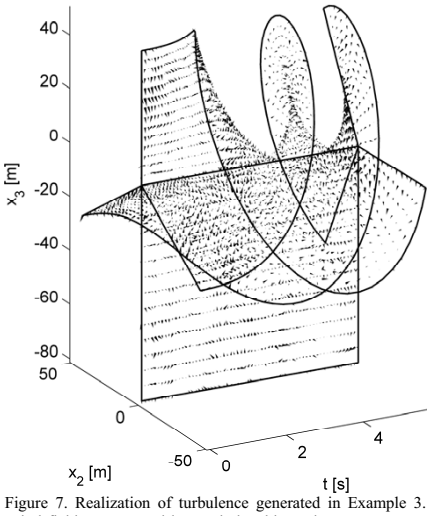


Figure 7. Realization of turbulence generated in Example 3. Wind field encountered by a wind turbine. The rotor starts at rest and accelerates.

5 CONCLUSIONS

The proposed method can generate synthetic turbulence with the correct correlation in space and the correct auto-spectrum.

The velocity field is generated only in the points where it is needed. This gives a saving in memory requirement compared to the Mann and Sandia methods. This is a clear advantage and the reduced memory requirement allows a finer resolution of the turbulence field as the memory requirement often sets the limit on the resolution. A fine resolution is needed in a LES of the flow past a blade.

The presented method reduces to the Sandia method for applications where the full turbulence field covering the entire rotor area is needed.

In the present form only isotropic turbulence can be generated. For the application outlined in the introduction the effect of anisotropy is considered to be of little importance.

6 ACKNOWLEDGMENTS

This research was supported by ”The Danish Council for Strategic Research” under the projects “Airfoils in Turbulent Inflow” and “Nonlinear Multibody Dynamics of Wind Turbines”.

7 REFERENCES

Connell, J.R. 1982. The spectrum of wind speed fluctuations encountered by a rotating blade of a wind energy conversion system. *Solar Energy* 29(5). 363-375.

Gilling, L., Sørensen, N.N. & Davidson, L. 2008. Detached Eddy Simulations of an Airfoil in Turbulent Inflow. *47th AIAA Aerospace Sciences Meeting, Orlando, 5-8 January 2008*.

von Karman, T. 1948. Progress in statistical theory of turbulence. *Proc. nat. acad. sci*. 34. 530-539.

Mann, J. 1994. The spatial structure of neutral atmospheric surface-layer turbulence. *J. fluid mech*. 273. 141-168.

Mann, J. 1998. Wind field simulation, *Prob. engng. mech*. 13(4). 269-282.

Shinozuka, M. & Jan, C. 1972. Digital simulation of random processes and its applications, *J. sound. vib*. 25. 111–128.

Taylor, A.A. 1976. *The structure of turbulent shear flow.* 2nd ed. Cambridge University Press, Cambridge.

Veers, P.S. 1984. *Modeling stochastic wind loads on vertical axis wind turbines*. SAND83-1909, Sandia National Laboratories, Albuquerque.

Veers, P.S. 1988. *Three-dimensional wind simulation*. SAND88-0152, Sandia National Laboratories, Albuquerque.

Zahle, F. 2006. *Wind turbine aerodynamics using an incompressible overset grid method*. PhD thesis, Imperial College, London.

Zahle, F. & Sørensen, N.N. 2007. On the influence of far-wake resolution on wind turbine flow simulations. *J. Physics: Conference Series* 75(2007)012042.

Zahle, F. & Sørensen, N.N. 2008. Rotor aerodynamics in atmospheric shear flow. *Scientific proceedings. 2008 European Wind Energy Conference and Exhibition, Brussels, March 31 – April 3 2008*.