Direct Numerical Simulation of the LDA Sampling Process

Bachelor Eind Project
Contents

1 Summary 3

2 Introduction 4

3 The LDA sampling process 6
  3.1 Introduction to Laser Doppler Anemometry ............... 6
  3.2 Velocity bias of LDA ...................................... 9
  3.3 Correction factors ........................................ 10
    3.3.1 Average flux ........................................ 11
    3.3.2 3D inverse velocity .................................. 11
    3.3.3 2D inverse velocity .................................. 12
    3.3.4 1D inverse velocity .................................. 12
    3.3.5 Transit-time .......................................... 12
    3.3.6 Interarrival-time .................................... 12
  3.4 Introduction to Direct Numerical Simulation ........... 13

4 Numerical experiment 14
  4.1 Finding the correct weighting factor using 'LDA of DNS' .. 14
  4.2 Relevant statistical quantities ............................. 15
    4.2.1 Mean and variance .................................. 15
    4.2.2 Skewness .............................................. 15
    4.2.3 Kurtosis ............................................... 17
  4.3 Programming in Fortran ................................. 18
    4.3.1 Program setup ....................................... 18
    4.3.2 Creating Histograms and Scatterplots ............. 19

5 Results 20
  5.1 Result tables and histograms ............................. 20
    5.1.1 No weighting factor .................................. 21
    5.1.2 Average flux correction ............................... 22
    5.1.3 3D inverse velocity correction ....................... 23
    5.1.4 2D inverse velocity correction ....................... 25
    5.1.5 1D inverse velocity correction ....................... 27
    5.1.6 Transit-time correction .............................. 29
5.1.7 Interarrival-time correction .......................... 32
5.2 Comparison of Results ................................. 35

6 Conclusion and recommendations ...................... 36
A Appendix ................................................... 39
Chapter 1

Summary

There are many applications in which the speed of a turbulent flow of gas or liquid has to be determined, and a popular non-intrusive way of doing this is by Laser Doppler Anemometry. This technique uses the fact that light scattered by moving particles undergoes a Doppler shift which is proportional to the particle velocity.

An intrinsic problem with this technique is the 'velocity bias', which is caused by the random distribution of the reflection particles and the changing mass flux in a turbulent flow. This velocity bias causes measured statistical quantities such as the mean velocity and the variance, to be biased towards the higher velocities. There are many ways to correct for this velocity bias, but opinions vary over which is the right one.

In this experiment we will be simulating a turbulent flow by means of a Direct Numerical Simulation and we will also simulate an LDA measurement done on this DNS. By applying several correction factors we will try to determine which is the best correction method for the velocity bias.
Chapter 2

Introduction

There are many applications in which it is useful to measure single-point statistics in liquids and gases. Traditionally this is done using hot-wire anemometry or pressure probes. The use of these techniques is widespread and they have proven to be effective in many situations. However, these conventional techniques have a major drawback: a physical probe is always needed in the flow, thus disturbing the flow during the measurement.

Another drawback is the inability of these methods to resolve (instantaneous) flow reversals. In many situations accurate single-point velocity data is needed in highly turbulent flows. Laser Doppler Anemometry (LDA) is suitable for this task because of its non-intrusive nature, its high spatial and temporal resolution and its ability to measure flow reversals in highly turbulent flows.

LDA samples the velocity of small flow tracers that move with the flow. Since these tracers are randomly distributed in the flow the sampling times are also random. The randomness of the sampling times precludes the use of standard data processing techniques. For example, the Fast Fourier Transform can no longer be used to compute the power spectral density of the velocity fluctuations. Another problem of working with LDA is the existence of a correlation between the sampling process and the velocity. As a consequence of this correlation, statistical quantities, such as the arithmetic mean velocity, are systematically too high, which is known as the velocity bias. Many ways to correct for the velocity bias have been proposed in the literature. Still it remains difficult to prove that one method is better than another because the true, unbiased statistics of the turbulent flow are unknown in an experiment. This study follows another approach. The sampling process in LDA is replicated by tracking seed particles in a flow that follows from an accurate Direct Numerical Simulation. The advantage of this procedure is that the flow statistics are known in all spatial and temporal detail, thus allowing an accurate evaluation of the various velocity bias correction methods.
Figure 2.1: The schematic representation of a 2D LDA system.
Chapter 3

The LDA sampling process

3.1 Introduction to Laser Doppler Anemometry

The main advantage of LDA over standard measuring techniques is that there is no need for a physical probe in the flow, so the flow is not perturbed in any way [1]. Measuring flow velocities using LDA depends upon the Doppler-shift created when a moving particle traverses/crosses a laserbeam. The amount of Doppler-shift is directly related to the speed of the particle and thus the flow velocity if it is assumed that the particles are perfect flow tracers.

Since the Doppler frequency (Doppler-shift) is much smaller than the frequency of the light (typically \( f_D / f_0 \approx 10^{-13} \)), the direct detection of the Doppler frequency requires a very high resolution of the detector. Such detectors do not exist. A solution for this problem is using two beams that intersect in the measurement volume, as can be seen in figures 3.1 and 3.2. If these incident beams are properly aligned, the intensity of the light in the overlap region is given by

\[
I = E_{10}^2 + E_{20}^2 + 2E_{10}E_{20}\cos(2k_0y\sin(\theta/2) + \phi_1 - \phi_2)
\] (3.1)

where \( E_{10} \) and \( E_{20} \) are the intensities of the two incident beams, \( k_0 \) is the wave number (\( = 2\pi/\lambda_0 \)), \( \theta \) is the angle between the unit vectors \( \hat{e}_{1z} \) and \( \hat{e}_{2z} \), and \( \phi_1 - \phi_2 \) is the phase difference between the incident light waves.

As you can see from equation 3.1, we now get a pattern of lines of constant intensity in the x-direction. So the intensity varies periodically in the y-direction.

The distance between two lines of constant intensity is given by

\[
d_f = \frac{\lambda_0}{2\sin(\theta/2)}
\] (3.2)

where \( d_f \) is the 'fringe distance'. This is the main calibration factor of the LDA system.
Figure 3.1: The optical arrangement for the dual-beam heterodyne LDA.

Figure 3.2: The interference of two plane light waves
Figure 3.3: The effect of frequency shift on the relationship between the particle velocity and the frequency of the system output signal.

If a particle passes through the measuring volume with velocity $\vec{v} = |\vec{v}| \sin \alpha$, it scatters light with an intensity that is proportional to the local value of $I$. So the scattered light then oscillates with frequency

$$f_D = \frac{\vec{v}}{d_f} = \frac{2 \sin(\theta/2)}{\lambda_0} |\vec{v}| \sin \alpha \quad (3.3)$$

This frequency is equal to the Doppler frequency\[1\].

In the above system it is impossible to distinguish between positive and negative $\vec{v}$, so the system does not know whether a particle with speed $|\vec{v}|$ is going in the positive or negative y-direction.

This problem is solved by changing the frequency of one of the incident beams with a known, constant value $f_\sigma$. This is known as the frequency pre-shift. The intensity of the light in the overlap region now becomes

$$I = E_{1o}^2 + E_{2o}^2 + 2E_{1o}E_{2o}\cos(2\pi f_s t + 2k_0 y \sin(\theta/2) + \phi_1 - \phi_2) \quad (3.4)$$

So the fringes in the interference pattern now move with velocity $v_s = d_f f_s$, and the detector output oscillates with a frequency of

$$f_D = f_s + \frac{2 \sin(\theta/2)}{\lambda_0} |\vec{v}| \sin \alpha \quad (3.5)$$

This effect of the pre-shift frequency shift is illustrated in figure 3.3. This enables us to distinguish between positive and negative velocity. It is common to set the pre-shift frequency $f_s$ about two times larger than the Doppler frequency that is associated with the smallest anticipated speed in the flow\[2\].
3.2 Velocity bias of LDA

Since the particles are distributed randomly in space, the sampling times are random as well. A model that is often used to describe the interarrival times is by Poisson statistics. With Poisson statistics the probability density function of the interarrival times, $\Delta t$, is given by

$$p(\Delta t) = \nu e^{-\nu \Delta t}$$

Here $\nu$ is the 'rate parameter' of the Poisson process, which can be interpreted as the mean number of samples per unit time.

Normal arithmetic averages can be used to calculate the mean, variance etc, if there is no correlation between the instantaneous data rate and the instantaneous fluid velocity. But in this case there is such a correlation.

If the flow velocity varies with time (as is the case with turbulent flows), then there will be a varying mass flux going through the measuring volume. When the flux is relatively high, there will be flowing more particles through the measuring volume then in times of low flux. This increases the chance of measuring a particle with high velocity and decreases the chance of measuring a particle with low velocity. To illustrate it more clearly: Think of the extreme situation where the velocity of the flow is zero. It would be impossible to make LDA measurements in this situation because there is no way for a particle to get into the measuring volume.

The sampling process thus has a dependance on the flow velocity. This intrinsic error in the measurement is known as the 'velocity bias' [1].

The velocity bias can be illustrated by looking at the measuring volume depicted in figure 3.4. The projection of the measuring volume on the plane normal to the instantaneous velocity vector, $\vec{v}$, is denoted by $A_p$. When we define the number of particles per unit volume as $M$, the expected number of
particles that pass through the measuring volume is given by

$$\lambda = |v|A_pM$$  \hspace{1cm} (3.6)

As you can see from equation 3.6, $\lambda$ is time-varying and proportional to
the volume flux through the measuring volume, so the probability of measuring
a particle with a high velocity is higher than the probability of measuring a
particle with low velocity.

This will result in the velocity histogram to be biased towards the high
velocities and all the calculated statistical quantities to be biased as well.

This can be simply shown for the mean velocity. Consider a one dimen-
sional flow (that means that $A_p$ and $M$ are constant!) with mean velocity $\bar{u}$
and variance $\sigma^2$ and where it is assumed that the instantaneous data rate is
proportional to the instantaneous volume flux. The arithmetic average $u_r$ of
the velocity samples is then a measure for

$$u_r = \frac{u(t)\lambda}{\lambda} = \frac{u(t)|u(t)|A_pM}{|u(t)|A_pM} = \frac{u(t)|u(t)|}{|u(t)|}$$  \hspace{1cm} (3.7)

but for low turbulence situations, i.e. $\frac{\sigma^2}{\bar{u}^2} \ll \frac{\sigma^2}{\bar{u}^2}$, this reduces to

$$u_r = \frac{\bar{u}^2}{u(t)} = \bar{u} + \frac{\sigma^2}{\bar{u}}$$

This shows that the arithmetic average $u_r$ is higher than the true mean velocity $\bar{u}$. The difference increases with increasing turbulence intensity.

This has the consequence that measurement averages tend to give higher
velocities than they should. It also shows that the effect of the velocity bias
increases when the turbulence intensity goes up.

### 3.3 Correction factors

In section 3.2 it was stated that the velocity bias inhibits the use of arithmic
averages to calculate the velocity statistics. To take the effects of the velocity
bias into account we have to change the data-processing methods. There are two
types of correction methods, namely sampling techniques and weighting factors
[1]. Sampling techniques try to sample the velocity at approximately equidistant
times or try to reconstruct the original velocity signal, whereas weighting factors
compensate for the effect of the velocity bias during the processing of the velocity
samples. Only the weighting factors will be discussed here.

The weighting factors are applied on each individual velocity sample. The
weighting factor has a different value for each individual sample, so an algorithm
is needed to do this calculation. In equation 3.8 an example is shown. This is the
equation for calculating the weighted mean velocity where $u_i$ is the $i$-th velocity
sample, $N$ is the total number of velocity samples and $w$ is the weighting factor
[1].
3.3.1 Average flux

As the velocity bias is caused by the flux dependance of the probability of making a sample, the most ideal way of correcting the LDA data would be to use \( \frac{1}{\phi x} \) as the weighting factor. This weighting factor is pictured in equation 3.9. Here \( \phi x \) is the flux in the x-direction through the measuring volume at the time of the \( i \)-th measurement.

\[
\omega_i = \frac{1}{\sqrt{\phi_{x}^2 + \phi_{y}^2 + \phi_{z}^2}}
\]  

(3.9)

With equation 3.9 as the weighting factor, the abundance in high velocity measurements is counterweighted by the high flux (and thus low \( \frac{1}{\phi x} \)) at the time of measurement.

The only problem with this correction method is that in real measurement situations, the flux at the time of measurement is unknown.

Still it is useful to consider this weighting factor, as the flux is known in this study, where we get our data from a Direct Numerical Simulation.

3.3.2 3D inverse velocity

When the flux through the measuring volume is high, one could expect the velocity of the air to be high as well. So the assumption they are correlated is right (taken the assumption that the density is constant [1]). For an ellipsoidal measuring volume with its long axis parallel to the z-direction, this weighting factor is shown in equation 3.10. Here \( u_i, v_i \) and \( w_i \) are the \( x, y \) and \( z \) component of the velocity at the time of the \( i \)-th measurement, \( d \) is the diameter and \( l \) is the length of the measuring volume [1].

\[
\omega_i = \frac{1}{\sqrt{u_i^2 + v_i^2 + (\frac{d}{l})^2 w_i^2}}
\]  

(3.10)

But there is a problem with this correction factor. All three velocity components have to be measured simultaneously. Most LDA systems currently available measure only 1 or 2 velocity components. Although this correction factor would be the most precise of all three inverse velocity correction factors, it is not suited for the LDA techniques available today.

A way of getting around this problem is by taking a look at the ratio of \( \frac{d}{l} \). This is discussed in the next section.
3.3.3 2D inverse velocity

Because in many practical cases in equation 3.10 the ratio $\frac{\delta}{\lambda} \approx 0.1$, the contribution of the $w$ component in equation 3.10 is very limited. For that reason it is interesting to look at the possibility to omit that component. The resulting weighting factor is shown in equation 3.11.

$$\omega_i = \frac{1}{\sqrt{u_i^2 + v_i^2}}$$  \hspace{1cm} (3.11)

3.3.4 1D inverse velocity

Even though by omitting more and more velocity components the weighting factor will become less accurate, it is still interesting to consider the 1D model of the inverse velocity correction method. There might be measuring systems where only one velocity component is known, so it is useful to see whether this correction method has any positive effect on the measurement data. This weighting factor is shown in equation 3.12.

$$\omega_i = \frac{1}{\sqrt{u_i^2}} = \frac{1}{|u_i|}$$  \hspace{1cm} (3.12)

The question immediately rises whether this correction factor is still useful, because by omitting 2 of the 3 velocity components a lot of information about the flow won’t be included in the correction factor.

3.3.5 Transit-time

Another way of describing how high the flux in the measuring volume is, is by looking at the time a particle spends in the measuring volume. In a real life measurement, the LDA signal processors will register how long each individual Doppler-burst lasts. This can be assumed to be equal to the time that the measured particle has stayed in the measuring volume. This time is commonly known as the ‘transit-time’. On average, since a particle can also just skim the surface of the measuring volume, it can be expected that the lower the transit time is for a specific measurement, the higher the flux was at time of measurement. This weighting factor is shown in equation 3.13. Where $t_{r_i}$ stands for the transit-time of the $i$-th measured particle.

$$\omega_i = t_{r_i}$$  \hspace{1cm} (3.13)

3.3.6 Interarrival-time

When the flux in the measuring volume is high, there is a high probability that more particles will flow through it per unit time than when the flux is low.
So the time between successive measurements is correlated to the flux. This weighting factor is shown in equation 3.14. Here, \( t_i \) stands for the time that the \( i \)-th particle was measured.

\[ \omega_i = t_{i+1} - t_i \]  \hspace{1cm} (3.14)

### 3.4 Introduction to Direct Numerical Simulation

A Direct Numerical Simulation, or DNS, is a computational representation of a fluid flow, computed using the Navier-Stokes equations. Since performing a DNS requires full numerical discretisation of all components of the momentum and continuity equations in time and three-dimensional space, it uses a lot of computer resources[3]. It is not something that can be done on your average personal computer.

The advantage of using the data from a DNS instead of real experimental data, is that the DNS data is much more accurate, and (in this experiment) that it comes with a lot of extra data, such as flux through the measuring volume, speed of the particles and exact transit-time.

The use of DNS data in this experiment will give the required accuracy to determine the best correction method for the velocity bias.
Chapter 4

Numerical experiment

4.1 Finding the correct weighting factor using 'LDA of DNS'

In this experiment we will be working with two types of datasets. A short description of these datasets will be given here.

In the first place we have the output-data of the DNS. This DNS was done for a turbulent airflow in a cubic shaped volume with a mean velocity of $0.1 \text{ m/s}$ in the u-direction.

This DNS data tells us, in timesteps of $1 \cdot 10^{-3} \text{ s}$, the value of the $x$, $y$ and $z$ component of the velocity of the (turbulent) airflow in the simulated volume. A period of 2000 seconds was simulated, so this amounts to $2000 \cdot 10^3$ timesteps in the simulation.

With the DNS data an LDA measurement was simulated. In other words, we simulated measuring the velocity of the (simulated) DNS airflow with LDA. The LDA measuring volume is cigar-like shaped and lies within the cubic DNS volume.

The LDA output-data tells us the value of the $x$, $y$ and $z$ component of the 'measured' speed, the point in time the particle was measured, the transit-time, the influx and the outflux (of the measuring volume) at the time of measurement. These quantities are respectively represented by $u$, $v$, $w$, $t$, $tr$, $\phi_{in}$, $\phi_{out}$.

If we compare the data from the DNS with the results from the LDA we will probably see that due to the velocity bias, the LDA data suggests higher velocities than the DNS. With the aid of the various weighting factors we are going to try to get the LDA results as close to the original DNS data as possible. If we achieve this we will have a practical solution to the intrinsic measurement error of the velocity bias.
4.2 Relevant statistical quantities

When comparing large amounts of data, it is easy to do this by means of certain commonly known statistical quantities. The quantities that will be used in this experiment will be discussed here.

4.2.1 Mean and variance

The mean is probably the most widely known statistical quantity of interest. For an arithmetic mean one just has to add all the data together, and then divide by the number of summations you have made:

\[ \bar{u} = \frac{1}{N} \sum_{i=1}^{N} u_i \] (4.1)

It becomes a bit more interesting when dealing with a weighting factor. The algorithm for obtaining this 'weighted mean' is shown in equation 4.2.

\[ \bar{u} = \frac{\sum_{i=1}^{N} u_i \omega_i}{\sum_{i=1}^{N} \omega_i} \] (4.2)

Equation 4.2 shows that for a weighting factor of 1, \( \omega_i = 1 \), the arithmetic mean of equation 4.1 is obtained. The mean is also known as the 'average'.

The variance is a measure for how much the data differs from the mean. The variance is also known as the square of the 'standard deviation', which is a measure for how wide the probability density function is. The mean and standard deviation are illustrated in figure 4.1. For convenience we will restrict ourselves to the term 'variance'.

Calculating the normal variance of a dataset is not that hard, but once again things start getting interesting as soon as we want to implement the weighting factor. The algorithm for calculating the 'weighted variance' is shown in 4.3.

\[ \sigma^2 = \frac{\sum_{i=1}^{N} (u_i - \bar{u})^2 \omega_i}{\sum_{i=1}^{N} \omega_i} \] (4.3)

4.2.2 Skewness

The skewness, \( \gamma_1 \), is a measure of the asymmetry of the probability density function. When the probability distribution is not symmetrical around the mean, it is said to to have positive or negative skew. In picture 4.2 an example can be seen of two probability distributions, one with negative and one with positive skew.
Figure 4.1: The probability density function of a Gaussian distribution with the mean and standard deviation illustrated.

Figure 4.2: An example of two probability distributions with negative and positive skew.
The skewness of a data-set is defined as equation 4.4

$$\gamma = \frac{\mu_3}{\sigma^3}$$  \hspace{1cm} (4.4)

where $\mu_3$ is the third moment about the mean, as defined in equation 4.5 and $\sigma$ is the square root of the variance (the standard deviation).

$$\mu_3 = \frac{\sum_{i=1}^{N} (u_i - \bar{u})^3 \omega_i}{\sum_{i=1}^{N} \omega_i}$$  \hspace{1cm} (4.5)

Putting equations 4.3, 4.4 and 4.5 together, we get equation 4.6, which describes the way to calculate the skewness $\gamma_1$.

$$\gamma_1 = \frac{\sum_{i=1}^{N} (u_i - \bar{u})^3 \omega_i}{\left( \sum_{i=1}^{N} (u_i - \bar{u})^2 \omega_i \right)^{3/2}}$$  \hspace{1cm} (4.6)

### 4.2.3 Kurtosis

The kurtosis, $\gamma_2$, of a dataset measures how 'peaked' the probability distribution of that dataset is. Imagine we have two probability distributions both with the same average and variance, but one of them is peaked at the mean and has longer 'tails', and the other is wider at the mean with shorter 'tails'. With these distributions the former will have a higher kurtosis than the latter. This is illustrated in figure 4.3.

A high kurtosis means that the variance is caused by large, infrequent deviations from the mean, and a low kurtosis means that the variance is caused by smaller, more frequent deviations from the mean.

Because it is often easy to compare the kurtosis of a data-set to that of the normal distribution, a factor '-3' is often added in the equations for the kurtosis. In this way, the kurtosis of a Gaussian distribution will be zero. In this experiment the factor -3 is also present in the calculations.

The definition for the kurtosis is given by equation 4.7

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3$$  \hspace{1cm} (4.7)

where $\mu_4$ is the fourth moment about the mean, as defined in equation 4.8.

$$\mu_4 = \frac{\sum_{i=1}^{N} (u_i - \bar{u})^4 \omega_i}{\sum_{i=1}^{N} \omega_i}$$  \hspace{1cm} (4.8)
By combining equation 4.3, 4.7 and 4.8, equation 4.9 is obtained for calculating the kurtosis.

\[
\gamma_2 = \frac{\sum_{i=1}^{N} (\omega_i - \bar{\omega})^4 \omega_i}{\left( \sum_{i=1}^{N} (\omega_i - \bar{\omega}) \omega_i \right)^2} - 3 \tag{4.9}
\]

4.3 Programming in Fortran

To be able to work with the big datasets used in this experiment, a programming language had to be chosen. The mathematical requirements for this experiment were not that large, but the program would have to be able to work with large data-matrices. Because Fortran and Matlab are two languages that meet this requirement and because they are taught at the TU Delft, it was easiest to pick one of these two. Our choice for Fortran simply relies on the fact that we found it easier to work in this language.

For the programming itself PSPad editor was used in combination with the G95 Fortran compiler.

The final programming code can be found in the appendix.

4.3.1 Program setup

The programming code is set up in a way that in one run of the program, calculations are done for one DNS dataset and its corresponding LDA dataset.
At first the code reads the datasets and saves the columns of the datasets as separate columns. These columns represent the separate measurement quantities, like speed in the u, v and w direction and measured fluxes.

Next a weighting factor is chosen from the ones discussed in section 3.3.

The statistical quantities (average, variance, skewness and kurtosis) are calculated by means of subroutines. A subroutine lets you program a portion of code only once, and then execute that piece of code several times during the program. This is especially convenient for the (weighted) average, variance, skewness and kurtosis, because they have to be calculated several times during one run of the program. These subroutines are called on three times during the program, namely for calculating the statistical quantities in the u, v and w direction.

The fact that there are separate subroutines being used for the calculation of the statistical quantities of the (weighted) LDA data and the DNS data is a product of the fact that the programming code was not written in one go. We started out by writing a basic code for the unweighted data and continued building the programming code from there on. This resulted in a code that might not be the most clarifying and efficient, but it is correct.

The output of the program is printed on the screen or to a file, whichever is easiest at that time.

4.3.2 Creating Histograms and Scatterplots

To successfully create histograms of the datasets a piece of code had to written that would divide the speed-domain in a certain number of slots, and then count per slot how many speed measurements fall into that slot. In this way a histogram is obtained which gives quite a good representation of the probability density function of the speed distribution. A subroutine was used in the programming code, because histograms had to be made of the u, v and w direction and with the use of a subroutine this only had to programmed once.

For the code to be valid for all datasets in all directions we took quite a wide speed-domain, from $-0.5 \text{ m/s}$ to $+0.5 \text{ m/s}$ (this way we were sure no measurement would fall outside this speed-domain). We divided this domain in 500 slots, so the slot size was set at $2 \times 10^{-3} \text{ m/s}$. The output of the programming code was written to a file, which we used to create the histograms in the graphing software Origin.

The scatterplots are made to compare weighting factors. For this we wrote a (very short) programming code that lets the program write the values of two weighting factors to a file for all measurements in the dataset. In this experiment we compared every weighting factor to the 'average flux' weighting factor.

The scatterplots are then created from this data with the graphing software Origin. These scatterplots give us an idea of how a weighting factor is correlated with the 'average flux' weighting factor.
Chapter 5

Results

5.1 Result tables and histograms

In the following section the results of the numerical experiment are shown. They are grouped per weighting factor, and only the results for the u-direction are shown (the results for the other directions can be found in the appendix).

For every measurement the average ($\bar{u}$), variance ($\sigma^2$), skewness ($\gamma_1$) and kurtosis ($\gamma_2$) of the DNS and the (weighted) LDA measurement are given in two tables. A histogram of the data is also shown for every measurement to clarify the difference in speed distributions between the DNS and the LDA data.
5.1.1 No weighting factor

<table>
<thead>
<tr>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{u} )</td>
<td>0.0911</td>
<td></td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>0.00378</td>
<td></td>
</tr>
<tr>
<td>( \gamma_1 )</td>
<td>-0.000302</td>
<td></td>
</tr>
<tr>
<td>( \gamma_2 )</td>
<td>-0.344</td>
<td></td>
</tr>
<tr>
<td>( \bar{u} )</td>
<td>0.115</td>
<td>+25%</td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>0.00365</td>
<td>-3%</td>
</tr>
<tr>
<td>( \gamma_1 )</td>
<td>-0.0156</td>
<td></td>
</tr>
<tr>
<td>( \gamma_2 )</td>
<td>-0.0440</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.1: Probability density function of the DNS data and the unweighted LDA data.

The histogram shows the comparison of the unprocessed DNS and LDA data. By looking at this figure it is immediately clear that the LDA data is biased towards the higher velocities. This velocity bias causes the calculated LDA mean to be 25% higher than the DNS mean.

Here it is also visible that the DNS simulation was performed for a flow with a mean speed in the \( u \) direction of 0.1 m/s.
5.1.2 Average flux correction

<table>
<thead>
<tr>
<th></th>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{u}$</td>
<td>0.0911</td>
<td>0.0918</td>
<td>+0.8%</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.00378</td>
<td>0.00379</td>
<td>+0.5%</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>-0.000302</td>
<td>-0.000678</td>
<td></td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>-0.344</td>
<td>-0.343</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2: Probability density function of the DNS data and the corrected LDA data with the 'average flux' weighting factor.

From the tables and figure 5.2 it is clear that the 'average flux' correction method counteracts the effects of the velocity bias. In the histogram the DNS and LDA data follow roughly the same path and the statistical quantities have errors of less than one percent.

As discussed in section 3.3.1, the information necessary to apply this weighting factor in real measurements will not be available. It is however useful to compare the results of other weighting factors to these results. In this way we can establish which weighting factor comes as close as possible to the results obtained by applying the 'average flux' weighting factor. A linear relation between the two would indicate that the weighting factor is a good substitution for the average flux correction method. This is done by constructing scatterplots, which are shown in the following sections.
5.1.3 3D inverse velocity correction

<table>
<thead>
<tr>
<th></th>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{u}$</td>
<td>0.0911</td>
<td>0.0915</td>
<td>$+$0.5%</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.00378</td>
<td>0.00380</td>
<td>$+$0.5%</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>-0.000302</td>
<td>-0.000274</td>
<td></td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>-0.344</td>
<td>-0.352</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.3: Probability density function of the DNS data and the corrected LDA data with the '3D inverse velocity' weighting factor.

By looking at the results in the tables and figure 5.3 it is immediately visible that this is an effective way of correcting the velocity bias. The DNS and LDA data have roughly the same distribution function and the weighted mean of the LDA data differs only 0.5% from the mean of the DNS data.

Because we established that the 'average flux' weighting factor is the optimal correction method, it is interesting to compare the '3D inverse velocity' weighting factor to the 'average flux' weighting factor. This is done in figure 5.4.
Figure 5.4: comparison of the '3D inverse velocity' weighting factor to the 'average flux' weighting factor.

This scatterplot shows us that there is a high correlation between the two correction methods. The fact that the datapoints seem to have a roughly linear relationship tells us that the '3D inverse velocity' weighting factor is an effective substitute for the 'average flux' weighting factor.

As mentioned before, there is little or no practical usefulness for this weighting factor, because there exist only a few real 3D LDA systems. As most LDA systems measure only one or two velocity components, this weighting factor can not be implemented in these systems.
5.1.4 2D inverse velocity correction

<table>
<thead>
<tr>
<th>DNS</th>
<th>LDA</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{u}$</td>
<td>0.0911</td>
<td></td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.00378</td>
<td></td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>-0.00302</td>
<td>-5%</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>-0.344</td>
<td>+4%</td>
</tr>
</tbody>
</table>

Figure 5.5: Probability density function of the DNS data and the corrected LDA data with the '2D inverse velocity' weighting factor.

Compared to figure 5.3 the LDA data in the histogram in figure 5.5 is clearly less accurate. The tables confirm this with a weighted mean and variance that differ from the DNS data with respectively -5% and +4%.

In figure 5.5 it is also visible that a peak has formed around $u = 0$ m/s. This can be explained by the fact that the velocity component in the w-direction is omitted in the weighting factor, as depicted in equation 3.11. There is a portion of the velocity samples that have most of their velocity in the w-direction, so the u and v velocity components are practically zero for these samples. When calculating the weighted value of these samples, equation 3.11 tends to go to infinity, increasing the significance of these samples in the statistical quantities and the histogram. This explains the abundance of 'zero-measurements' in figure 5.5.
Figure 5.6: comparison of the '2D inverse velocity' weighting factor to the 'average flux' weighting factor.

For comparison with the average flux correction method, the scatterplot in figure 5.6 was created. When comparing this to figure 5.4 it is visible that figure 5.6 has a wider spread of data close to the origin. This again can be explained by the fact that for some samples the u and v velocity components are practically zero, because the measured particle was travelling in the w-direction. This causes some samples to have a near zero value on the vertical axis of figure 5.6 while having a non-zero value on the horizontal axis.
5.1.5 1D inverse velocity correction

<table>
<thead>
<tr>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{u}$</td>
<td>0.0911</td>
<td></td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.00378</td>
<td></td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>-0.000302</td>
<td></td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>-0.344</td>
<td></td>
</tr>
<tr>
<td>$\bar{u}$</td>
<td>0.0290</td>
<td>-68%</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.00281</td>
<td>-25%</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.0904</td>
<td></td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>2.13</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.7: Probability density function of the DNS data and the corrected LDA data with the '1D inverse velocity' weighting factor.

In figure 5.7 the earlier explained peak at $u = 0 \text{ m/s}$ is even more evident. The omission of the all but the $u$-direction velocity component in this weighting factor clearly causes the LDA data to be corrected in the wrong way. This effect is also shown in the calculation of the weighted mean and variance as is shown in the tables.
Figure 5.8: comparison of the '1D inverse velocity' weighting factor to the 'average flux' weighting factor.

In the scatterplot in figure 5.8 this effect is visible in the large amount of samples that have near zero value on the vertical axis over a wide domain on the horizontal axis.

Figures 5.7 and 5.8 together with the calculated mean and variance indicate that this weighting factor is practically useless for correcting the LDA data to approach the DNS data.
5.1.6 Transit-time correction

<table>
<thead>
<tr>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u )</td>
<td>0.0911</td>
<td></td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>0.00378</td>
<td></td>
</tr>
<tr>
<td>( \gamma_1 )</td>
<td>-0.000302</td>
<td></td>
</tr>
<tr>
<td>( \gamma_2 )</td>
<td>-0.344</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u )</td>
<td>0.0943</td>
<td>+3%</td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>0.00385</td>
<td>+2%</td>
</tr>
<tr>
<td>( \gamma_1 )</td>
<td>-0.00246</td>
<td></td>
</tr>
<tr>
<td>( \gamma_2 )</td>
<td>-0.346</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.9: Probability density function of the DNS data and the corrected LDA data with the 'transit-time' weighting factor.

The calculated mean and variance and figure 5.9 indicate that the ‘transit time’ weighting factor is quite suitable for correcting the biased LDA data. As can be seen in figure 5.9, the probability density function of the LDA data has roughly the same shape as the probability density function of the DNS data which indicates a successful correction of the LDA data.
Figure 5.10: comparison of the 'transit-time' weighting factor to the 'average flux' weighting factor.

In figure 5.10 the scatterplot can be seen where the 'transit-time' weighting factor is compared to the 'average flux' weighting factor. A wide distribution of samples can be seen here. This effect is caused by the fact that not all particles travel through the measuring volume in the same way. Some particles might travel through the longest possible route and some particles might just skim the surface of the measuring volume.
This problem can be solved by averaging the samples, as is done in figure 5.11. In this figure the linear relationship between the inverse of the transit time and the average flux is clearly visible. This linear behaviour indicates that the 'transit-time' weighting factor is a suitable replacement for the 'average flux' weighting factor.
5.1.7 Interarrival-time correction

<table>
<thead>
<tr>
<th>DNS</th>
<th>LDA</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu )</td>
<td>0.0911</td>
<td>0.0946</td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>0.00378</td>
<td>0.00377+0.03%</td>
</tr>
<tr>
<td>( \gamma_1 )</td>
<td>-0.000302</td>
<td>-0.00498</td>
</tr>
<tr>
<td>( \gamma_2 )</td>
<td>-0.344</td>
<td>-0.239</td>
</tr>
</tbody>
</table>

Figure 5.12: Probability density function of the DNS data and the corrected LDA data with the 'interarrival-time' weighting factor.

From figure 5.12 and the tables at the top of this page it is clear that the 'interarrival-time' weighting factor is suitable for correcting the LDA data. The calculated LDA mean and variance differ respectively 3% and 0.03% from the DNS data. In figure 5.12 it can also be seen that the probability density function of the corrected LDA data has roughly the same shape as the probability density function of the DNS data.
In figure 5.13 the 'interarrival-time' weighting factor is compared to the 'average flux' weighting factor. A linear relationship is not clearly visible, so the data in figure 5.13 was averaged, just like was done in figure 5.11. Unfortunately no physically relevant results were obtained.

Another less accurate way of determining whether a linear relationship exists between the two weighing factors was asking the computer software 'Origin' to fit a linear relationship through the measurement data. This linear fit can be seen in figure 5.13 as the red line. This linear fit is described by the equation:

\[
\omega_{\text{interarrival}} = A + B \times \omega_{\text{aver Flux}}
\]

where \( \omega_{\text{interarrival}} \) is the 'interarrival-time' weighting factor, \( \omega_{\text{aver Flux}} \) is the 'average flux' weighting factor and \( A \) and \( B \) are constants that are determined by the linear fit. In this case 'Origin' computed the values for \( A \) and \( B \) to be:

\[
A = 8.007 \text{ with an uncertainty of 9.627} \\
B = 2.547 \times 10^8 \text{ with an uncertainty of } 0.262 \times 10^8
\]

This result does not prove that there is a clear linear relationship between the two weighting factors, because we simply ordered the computer program to find the best linear relationship the data offered. It is however striking that the value for \( A \) is so close to zero, especially if you take a look at the large domain of the vertical axis of figure 5.13.
The linear relationship that the two weighting factors are expected to have if they are substitutes for another, should go through the origin of the graph. So even though we can’t prove these two weighting factors have a linear relationship, the small value of $A$ suggests there might be such a relationship.
5.2 Comparison of Results

It is clear from the results that the ‘average flux’ weighting factor comes closest to reconstructing the original DNS data. But since the data for this weighting factor is not known in practical situations, we are searching for another weighting factor that comes as close as possible.

Of the inverse velocity weighting factors, it is clear that the 3D model works best. In the histogram of the 3D model the LDA and DNS data seem to have the same distribution, and the scatterplot of the 3D inverse velocity against the ‘average flux’ shows a linear correlation.

With the 1D and 2D models it is clearly visible that around 0 m/s peaks form. This is a result of the fact that, in the 1D model, when the speed is practically zero, the weighting factor tends to infinity. And since a weighted histogram is constructed by adding the weighting factors of the measurements in each slot (see programming code), these peaks occur. In the 2D model the same effect is present, only less profound.

We can also conclude that the transit-time and interarrival-time weighting produce very reasonable results. The fact that the scatterplot of the transit-time weighting has to be averaged first to get a linear relation comes from the fact that not every particle goes through the measurement volume via the longest path. This can result in short transit times (and thus high weighting factors) at relatively low flux. By averaging we can clearly see the linear relation to the average flux weighting.
Chapter 6

Conclusion and recommendations

In the evaluation of the results obtained with the different weighting factors we have to make a separation between the weighting factors that are possible to implement and those that are not. The 'average flux' and '3D inverse velocity' are practically not applicable because in real-life measurements the information required for these correction methods is not available. This is pitiful, because the results presented in this report show that the best results are obtained by these weighting factors. The results also showed that these two weighting factors are practically interchangeable.

On the other hand, for the '2D inverse velocity', '1D inverse velocity', 'transit-time' and 'interarrival-time' weighting factors all the necessary information is available in conventional LDA measurements. So for practical reasons it has to be determined which of these weighting factors is best suitable for correcting the LDA 'velocity bias'.

The results for the '2D inverse velocity' and '1D inverse velocity' weighting factors show that these correction methods are practically useless. This is mainly caused by the earlier discussed formation of peaks around $u = 0 \text{ m/s}$. The goal of a weighting factor correcting the 'velocity bias' is to let samples measured at low flux have a greater significance in the statistical quantities than the samples measured at high flux. This intended effect is completely overshadowed by the great amount of samples of which the weighting factor is too high because of the omission of one or two of the velocity components in these last two weighting factors.

This leaves the 'transit-time' and 'interarrival-time' weighting factors, which both managed to correct the LDA data in such a way that the calculated mean differed only 3% from the DNS mean. In earlier simulations, where smaller
datasets were used, the difference between the calculated LDA and DNS mean was slightly bigger, which indicates that a further reduction in this difference can be expected when doing a simulation with even bigger datasets then we used.

Because for the 'transit-time' weighting factor a linear relationship with the 'average flux' weighting factor was evident after averaging the scatterplot and a much more elaborate (and less accurate) way had to found to find a linear relationship between the 'interarrival-time' and 'average flux' weighting factor, our preference goes to the 'transit-time' weighting factor for correcting the 'velocity bias'.

It is not possible to argue with the obtained results of the 'interarrival-time' weighting factor, but for the 'transit-time' weighting factor the best physical evidence is found that it is a substitute for the 'average flux' weighting factor and thus the most effective correction method for the velocity bias.
Bibliography

[1] Investigation of a turbulent wake in an adverse pressure gradient using laser Doppler anemometry; M.J. Tummers, Delft University of Technology, 1999


Appendix A

Appendix

In this appendix is included:

- The results for the simulations in all three velocity directions
- The fortran programming code
Total dataset, no correction factor (cf=1)

**U-direction DNS output**
- average: 0.091127984
- variance: 0.0037761428
- skewness: -0.0003023258
- kurtosis: -0.343814

**U-direction LDA output**
- average: 0.11459187 (+25%)
- variance: 0.0036534609 (-3%)
- skewness: -0.01564477
- kurtosis: -0.043956973

**V-direction DNS output**
- average: 0.0007596612
- variance: 0.0037851701
- skewness: 0.0004996974
- kurtosis: -0.3603837

**V-direction LDA output**
- average: 0.0011602237 (+52%)
- variance: 0.0046643857 (+23%)
- skewness: 0.0001932147
- kurtosis: -0.54032636

**W-direction DNS output**
- average: 0.0001415673
- variance: 0.0037610622
- skewness: -0.00054187543
- kurtosis: -0.36040986

**W-direction LDA output**
- average: 0.0005211514 (+268%)
- variance: 0.004002336 (+6%)
- skewness: -0.0010570962
- kurtosis: -0.41337928
Total dataset, correction factor = 1/average flux

**U-direction DNS output**
- average: 0.091127984
- variance: 0.0037761428
- skewness: -0.0003023258
- kurtosis: -0.343814

**U-direction LDA output**
- average: 0.09183972 (+0.8%)
- variance: 0.0037933006 (+0.5%)
- skewness: -0.00067808176
- kurtosis: -0.34288856

**V-direction DNS output**
- average: 0.0007596612
- variance: 0.0037851701
- skewness: 0.0004996974
- kurtosis: -0.3603837

**V-direction LDA output**
- average: 0.0006261045 (-18%)
- variance: 0.00385709 ( +2%)
- skewness: 0.0008300821
- kurtosis: -0.34852257

**W-direction DNS output**
- average: 0.0001415673
- variance: 0.0037610622
- skewness: -0.00054187543
- kurtosis: -0.36040986

**W-direction LDA output**
- average: 0.00015629093 (+10%)
- variance: 0.0038107268 (+1%)
- skewness: -0.00080359774
- kurtosis: -0.38384387
Total dataset, correction factor $= \frac{1}{\sqrt{u^2 + v^2 + \left(\frac{d}{l}\right)^2 w^2}}$

**U-direction DNS output**
- average: 0.091127984
- variance: 0.0037761428
- skewness: -0.0003023258
- kurtosis: -0.343814

**U-direction LDA output**
- average: 0.09154868 (+0.5%) 
- variance: 0.0037956943 (+0.5%)
- skewness: -0.00027388232
- kurtosis: -0.35191783

**V-direction DNS output**
- average: 0.0007596612
- variance: 0.0037851701
- skewness: 0.0004996974
- kurtosis: -0.3603837

**V-direction LDA output**
- average: 0.0006210757 (-18%)
- variance: 0.003840082 (+1%)
- skewness: 0.00084702956
- kurtosis: -0.33894742

**W-direction DNS output**
- average: 0.0001415673
- variance: 0.0037610622
- skewness: -0.00054187543
- kurtosis: -0.36040986

**W-direction LDA output**
- average: 0.00015721063 (+11%)
- variance: 0.003800668 (+1%)
- skewness: -0.0008010473
- kurtosis: -0.3788185
Total dataset, correction factor = $\frac{1}{\sqrt{u^2 + v^2 + \left(\frac{d}{l}\right)^2w^2}}$
Total dataset, correction factor $= 1/sqrt(u^2 + v^2)$

**U-direction DNS output**
- average: 0.091127984
- variance: 0.0037761428
- skewness: -0.0003023258
- kurtosis: -0.343814

**U-direction LDA output**
- average: 0.08587501 (-5%)
- variance: 0.003939026 (+4%)
- skewness: 0.0056322943
- kurtosis: -0.49763638

**V-direction DNS output**
- average: 0.0007596612
- variance: 0.0037851701
- skewness: 0.0004996974
- kurtosis: -0.3603837

**V-direction LDA output**
- average: 0.0005214582 (-31%)
- variance: 0.0035977077 (-4%)
- skewness: 0.0010604493
- kurtosis: -0.19405258

**W-direction DNS output**
- average: 0.0001415673
- variance: 0.0037610622
- skewness: -0.00054187543
- kurtosis: -0.36040986

**W-direction LDA output**
- average: -0.0003731263 (+363%)
- variance: 0.0042809024 (+13%)
- skewness: -0.0024319843
- kurtosis: -0.51277626
Total dataset, correction factor $= \frac{1}{\sqrt{u^2 + v^2}}$
Total dataset, correction factor $= 1/sqrt(u^2)$

**U-direction DNS output**
- average: 0.091127984
- variance: 0.0037761428
- skewness: -0.0003023258
- kurtosis: -0.343814

**U-direction LDA output**
- average: 0.029017434 (-68%)
- variance: 0.002810699 (-25%)
- skewness: 0.09038429
- kurtosis: 2.1268036

**V-direction DNS output**
- average: 0.0007596612
- variance: 0.0037851701
- skewness: 0.0004996974
- kurtosis: -0.3603837

**V-direction LDA output**
- average: 0.020930575 (+2655%)
- variance: 0.005344112 (+41%)
- skewness: -0.02801259
- kurtosis: -1.0532792

**W-direction DNS output**
- average: 0.0001415673
- variance: 0.0037610622
- skewness: -0.0005418754
- kurtosis: -0.36040986

**W-direction LDA output**
- average: -0.022127558 (+15730%)
- variance: 0.0039295577 (+4%)
- skewness: 0.036380716
- kurtosis: -0.39942148
Total dataset, correction factor = $1/\sqrt{u^2}$
Total dataset, correction factor = transit time

**U-direction DNS output**
- average: 0.091127984
- variance: 0.0037761428
- skewness: -0.0003023258
- kurtosis: -0.343814

**U-direction LDA output**
- average: 0.09427955 (+3%)
- variance: 0.0038536715 (+2%)
- skewness: -0.0024593796
- kurtosis: -0.34615666

**V-direction DNS output**
- average: 0.0007596612
- variance: 0.0037851701
- skewness: 0.0004996974
- kurtosis: -0.3603837

**V-direction LDA output**
- average: 0.0008447603 (-11%)
- variance: 0.0039502354 (+4%)
- skewness: 0.0004396962
- kurtosis: -0.36135072

**W-direction DNS output**
- average: 0.0001415673
- variance: 0.0037610622
- skewness: -0.00054187543
- kurtosis: -0.36040986

**W-direction LDA output**
- average: 0.0000733126 (-48%)
- variance: 0.0038132016 (+1%)
- skewness: -0.0005418191
- kurtosis: -0.38487568
Total dataset, correction factor = transit time

1 / transit time vs. Average Flux

1 / transit time vs. Average Flux
Total dataset, correction factor  =  $t_{i+1} - t_i$
Total dataset, correction factor = $t_{i+1} - t_i$

![Graph showing the relationship between $1/(t_{i+1} - t_i)$ and average flux.](image)

Detailopname:
program BEP

implicit none

integer :: i,j,k !voor alle do-loops
integer, parameter :: m=29252 !de lengte van de LDA output
integer, parameter :: n=2000000 !de lengte van de DNS output
real, dimension(m) :: t,transit,u,v,w,u_in,v_in,w_in,steps, nr, influx, outflux
real, dimension(n) :: t_dns, u_dns, v_dns, w_dns, x_dns, y_dns, z_dns, influx_dns, outflux_dns
real :: speed_aver, speed_aver_dns
real :: skew, skew_dns
real :: kurt, kurt_dns
real :: variantie, variantie_dns
real :: gew_skew, gew_kurt
real, dimension(m) :: wgf

integer :: hist_grootte
real, dimension(hist_grootte) :: hist_output
real, dimension(hist_grootte) :: hist_output_dns
real, dimension(n) :: wgf_dns

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!! VUL HIER DE WEEGFACTOR IN (VECTOR GROOTTE 'M')!!!!!!!!!!!!!!!!!!!!!!!!!!!!

real :: first_mom, second_mom, third_mom, fourth_mom
real :: gew_skew, gew_kurt
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

integer :: hist_grootte, allocatable :: hist_output, hist_output_dns
real, dimension (::, ::), allocatable :: wgf_dns
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

open (unit=60, file='scatterplot.dat')
open (unit=50, file='all_LDA_data.dat')
open (unit=40, file='all_DNS_data.dat')

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!! het inlezen van de data

! het inlezen van de data

do i=1,m
  read (50,*) t(i), transit(i), u(i), v(i), w(i), u_in(i), & v_in(i), w_in(i), steps(i), nr(i), influx(i), outflux(i)
enddo

! het inlezen van de data

do j=1,n
  read (40,*) t_dns(j), u_dns(j), v_dns(j), w_dns(j), x_dns(j), y_dns(j), & z_dns(j), influx_dns(j), outflux_dns(j)
enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!! VUL HIER DE WEEGFACTOR IN (VECTOR GROOTTE 'M')!!!!!!!!!!!!!!!!!!!!!!!!!

do i=1,m
  wgf(i)=1/((influx(i)+outflux(i))/2)
endo

! wgf=1

! wgf=transit

! do i=1,m
!  wgf(i)= 1/(sqrt(u(i)**2 + v(i)**2 + ((0.2/0.4)**2)*w(i)**2))
! enddo

! do i=1,m
!  wgf(i)= 1/(sqrt(u(i)**2 + v(i)**2))
! enddo

! do i=1,m
! enddo

! do i=1,m

! \ wgf(i) = 1/\sqrt{(u(i))^2)\n!enddo

! do i=1,m-1
! \ wgf(i) = t(i+1) - t(i)
! \ if (wgf(i) < 0) then
! \ \ wgf(i) = 0
! \ \ u(i) = 0
! \ \ v(i) = 0
! \ \ w(i) = 0
! \ \ influx(i) = 0
! \ \ outflux(i) = 0
! \ end if
! enddo
! \ wgf(m) = wgf(m-1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! aver_flux=0
! aver_wgf=0
! do i=1,m
! \ aver_flux = aver_flux + ((influx(i) + outflux(i)) / 2)
! \ aver_wgf = aver_wgf + wgf(i)
! enddo
! aver_flux = aver_flux / m
! aver_wgf = aver_wgf / m

 do i=1,m
 \ write (60,*) ((influx(i) + outflux(i)) / 2), (1/wgf(i))
 enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! OPSTELLEN DATABESTANDEN HISTOGRAMMEN

! hoeveel vakjes krijgt het histogram???

hist_grootte = 500

allocate (hist_output(hist_grootte,3))
allocate (hist_output_dns(hist_grootte,3))

wgf_dns = 1

! U

open (unit=30, file='histogramU.dat')

call weighted(m, u, wgf, -0.5, 0.5, hist_grootte, hist_output)
call weighed(n, u_dns, wgf_dns, -0.5, 0.5, hist_grootte, hist_output_dns)

do i=1,hist_grootte
 \ write (30,*) hist_output_dns(i,1), hist_output_dns(i,2), &
 \ hist_output(i,1), hist_output(i,2), hist_output(i,3)
enddo

! V
open (unit=20, file='histogramV.dat')
call weighed(m,v,wgf,-0.5,0.5,hist_grootte,hist_output)
call weighed(n,v_dns,wgf_dns,-0.5,0.5,hist_grootte,hist_output_dns)
do i=1,hist_grootte
   write (20,*) hist_output_dns(i,1), hist_output_dns(i,2),&
   hist_output(i,1), hist_output(i,2),hist_output(i,3)
enddo
!

call weighed(m,w,wgf,-0.5,0.5,hist_grootte,hist_output)
call weighed(n,w_dns,wgf_dns,-0.5,0.5,hist_grootte,hist_output_dns)
do i=1,hist_grootte
   write (10,*) hist_output_dns(i,1), hist_output_dns(i,2),&
   hist_output(i,1), hist_output(i,2),hist_output(i,3)
enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!berekening U LDA output incl weegfactor

!berekening gewogen gemiddelde

call first(m,u,wgf,first_mom)

!berekening gewogen variantie

call second(m,u,wgf,first_mom,second_mom)

!berekening gewogen skewness

call third(m,u,wgf,first_mom,third_mom)
gew_skew=(third_mom/(second_mom**3/2))

!bereking gewogen kurtosis

call fourth(m,u,wgf,first_mom,fourth_mom)
gew_kurt=(fourth_mom/(second_mom**2)**3)

!!!!!!!!!!!!!!!!!!!!GEDEELTE U DNS OUTPUT

!berekening gemiddelde

call average(n,u_dns,speed_aver_dns)

!berekening variantie

call variance(n,u_dns,speed_aver_dns,variantie_dns)

!berekening skewness

call skewness(n,u_dns,speed_aver_dns,skew_dns)

!berekening kurtosis
call kurtosis(n,u_dns,speed_aver_dns,kurt_dns)

write (*,*) '******************************************************************************
write (*,*) 'INCLUSIEF WEEGFACTOR:
write (*,*) 'de gemiddelde LDA gemeten snelheid is ',first_mom,'m/s in de U richting.'
write (*,*) 'met een variantie van ',second_mom,'m/s'
write (*,*) 'met een skewness van ',gew_skew
write (*,*) 'met een kurtosis van ',gew_kurt
write (*,*) ''
write (*,*) 'de gemiddelde DNS berekende snelheid is ',speed_aver_dns,'m/s in de U richting.'
write (*,*) 'met een variantie van ',variantie_dns,'m/s'
write (*,*) 'met een skewness van ',skew_dns
write (*,*) 'met een kurtosis van ',kurt_dns
write (*,*) '******************************************************************************
open (unit=90, file='copie-paste.txt')

write (90,* ) speed_aver_dns
write (90,* ) variantie_dns
write (90,* ) skew_dns
write (90,* ) kurt_dns
write (90,* )
write (90,* ) first_mom, ((first_mom - speed_aver_dns)/speed_aver_dns)
write (90,* ) second_mom, ((second_mom - variantie_dns)/variantie_dns)
write (90,* ) gew_skew
write (90,* ) gew_kurt
write (90,* )

!!!!!!!!berekening V LDA output incl weegfactor
!berekening gewogen gemiddelde

call first(m,v,wgf,first_mom)

!berekening gewogen variantie

call second(m,v,wgf,first_mom,second_mom)

!berekening gewogen skewness

call third(m,v,wgf,first_mom,third_mom)

gew_skew=(third_mom/(second_mom**2))/3)

!berekening gewogen kurtosis

call fourth(m,v,wgf,first_mom,fourth_mom)

gew_kurt=(fourth_mom/(second_mom**2))-3

!GEDEELTE V DNS OUTPUT
!berekening gemiddelde

call average(n,v_dns,speed_aver_dns)

!berekening variantie

call variance(n,v_dns,speed_aver_dns,variantie_dns)

!berekening skewness
call skewness(n,v_dns,speed_aver_dns,skew_dns)

!berekening kurtosis

call kurtosis(n,v_dns,speed_aver_dns,kurt_dns)

write (*) , 'INCLUSIEF WEEGFACTOR'
write (*) , 'de gemiddelde LDA gemeten snelheid is ',first_mom,'m/s in de V richting.'
write (*) , 'met een variantie van ',second_mom,'m/s'
write (*) , 'met een skewness van ',gew_skew
write (*) , 'met een kurtosis van ',gew_kurt
write (*) , 'de gemiddelde DNS berekende snelheid is ',speed_aver_dns,'m/s in de V richting.'
write (*) , 'met een variantie van ',variantie_dns,'m/s'
write (*) , 'met een skewness van ',skew_dns
write (*) , 'met een kurtosis van ',kurt_dns
write (*) , '*******************************************************************'
write (90,*), speed_aver_dns
write (90,*), variantie_dns
write (90,*), skew_dns
write (90,*), kurt_dns
write (90,*), first_mom, ((first_mom - speed_aver_dns)/speed_aver_dns)
write (90,*), second_mom, ((second_mom - variantie_dns)/variantie_dns)
write (90,*), gew_skew
write (90,*), gew_kurt
write (90,*)

!!!!!!!!!!!!berekening W LDA output incl weegfactor

!berekening gewogen gemiddelde

call first(m,w,wgf,first_mom)

!berekening gewogen variantie

call second(m,w,wgf,first_mom,second_mom)

!berekening gewogen skewness

call third(m,w,wgf,first_mom,third_mom)

gew_skew=(third_mom/(second_mom**(3/2)))

!bereking gewogen kurtosis

call fourth(m,w,wgf,first_mom,fourth_mom)

gew_kurt=(fourth_mom/(second_mom**2))-3

!GEDEELTE W DNS OUTPUT

!berekening gemiddelde

call average(n,w_dns,speed_aver_dns)
!berekening variantie

call variance(n,w_dns,speed_aver_dns,variantie_dns)

!berekening skewness

call skewness(n,w_dns,speed_aver_dns,skew_dns)

!berekening kurtosis

call kurtosis(n,w_dns,speed_aver_dns,kurt_dns)

write (*,*) 'INCLUSIEF WEEGFACOR'
write (*,*) 'de gemiddelde LDA gemeten snelheid is ',first_mom,'m/s in de W richting.'
write (*,*) 'met een variantie van ',second_mom,'m/s'
write (*,*) 'met een skewness van ',gew_skew
write (*,*) 'met een kurtosis van ',gew_kurt
write (*,*) ''
write (*,*) 'de gemiddelde DNS berekende snelheid is ',speed_aver_dns,'m/s in de W richting.'
write (*,*) 'met een variantie van ',variantie_dns,'m/s'
write (*,*) 'met een skewness van ',skew_dns
write (*,*) 'met een kurtosis van ',kurt_dns
write (*,*) '*******************************************************************'

write (90,*) speed_aver_dns
write (90,*) variantie_dns
write (90,*) skew_dns
write (90,*) kurt_dns
write (90,*)
write (90,*) first_mom, ((first_mom - speed_aver_dns)/speed_aver_dns)
write (90,*) second_mom, ((second_mom - variantie_dns)/variantie_dns)
write (90,*) gew_skew
write (90,*) gew_kurt
write (90,*)

end program BEP

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine average(aantal,snelheid,gemiddelde) !hier is alleen 'gemiddelde' een output
implicit none

integer :: k,aantal
real, dimension(aantal) :: snelheid
real :: sum_snelheid,gemiddelde

sum_snelheid=0
   do k=1,aantal
      sum_snelheid=sum_snelheid + snelheid(k)
   enddo
   gemiddelde=sum_snelheid/aantal

end subroutine average

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine variance(aantal,snelheid,gemiddelde,variantie) !hier is alleen 'standdev' een output
implicit none
integer :: k,aantal
real, dimension(aantal) :: snelheid,variantie_matrix
real :: gemiddelde,variantie,sum_variantie

    sum_variantie=0
    do k=1,aantal
       variantie_matrix(k)=(snelheid(k)-gemiddelde)**2
       sum_variantie=sum_variantie+variantie_matrix(k)
    enddo
    variantie=sum_variantie/aantal
end subroutine variance

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine skewness(aantal,snelheid,gemiddelde,skew)
implicit none
integer :: k, aantal
real, dimension(aantal) :: snelheid, third_cumulant, variantie
real :: third_cumulant_sum,sum_variantie,gemiddelde,skew

    third_cumulant_sum=0      
    sum_variantie=0
    do k=1,aantal
       third_cumulant(k)=(snelheid(k)-gemiddelde)**3
       third_cumulant_sum=third_cumulant_sum + third_cumulant(k)
       variantie(k)=(snelheid(k)-gemiddelde)**2
       sum_variantie=sum_variantie+variantie(k)
    enddo
    skew=(third_cumulant_sum/aantal)/((sum_variantie/aantal)**(3/2))
end subroutine skewness

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine kurtosis(aantal,snelheid,gemiddelde,kurt)
implicit none
integer :: k, aantal
real, dimension(aantal) :: snelheid, fourth_cumulant, variantie
real :: fourth_cumulant_sum,sum_variantie,gemiddelde,kurt

    fourth_cumulant_sum=0
    sum_variantie=0
    do k=1,aantal
       fourth_cumulant(k)=(snelheid(k)-gemiddelde)**4
       fourth_cumulant_sum=fourth_cumulant_sum + fourth_cumulant(k)
       variantie(k)=(snelheid(k)-gemiddelde)**2
       sum_variantie=sum_variantie+variantie(k)
    enddo
    kurt=((aantal*fourth_cumulant_sum)/((sum_variantie)**2)-3)
end subroutine kurtosis

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine first(aantal,snelheid,weegfactor,gewogen_gemiddelde)
implicit none
integer :: k,aantal
real, dimension(aantal) :: snelheid,weegfactor
real :: sum_snelheid,gewogen_gemiddelde,sum_weegfactor
sum_snelheid=0
sum_weegfactor=0
  do k=1,aantal
      sum_snelheid=sum_snelheid + (snelheid(k)*weegfactor(k))
      sum_weegfactor=sum_weegfactor + weegfactor(k)
  enddo
gewogen_gemiddelde=sum_snelheid/sum_weegfactor
end subroutine first

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subroutine second(aantal,snelheid,weegfactor,gewogen_gemiddelde,second_moment)
implicit none
integer :: k,aantal
real, dimension(aantal) :: snelheid,weegfactor,gewogen_variantie
real :: gewogen_gemiddelde,second_moment,sum_variantie,sum_weegfactor

sum_variantie=0
sum_weegfactor=0
  do k=1,aantal
      gewogen_variantie(k)=((snelheid(k)-gewogen_gemiddelde)**2)*weegfactor(k)
      sum_variantie=sum_variantie + gewogen_variantie(k)
      sum_weegfactor=sum_weegfactor + weegfactor(k)
  enddo
second_moment=(sum_variantie/sum_weegfactor)
end subroutine second

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subroutine third(aantal,snelheid,weegfactor,gewogen_gemiddelde,third_moment)
implicit none
integer :: k,aantal
real, dimension(aantal) :: snelheid,weegfactor,third_cum
real :: gewogen_gemiddelde,third_moment,sum_third_cum,sum_weegfactor

sum_third_cum=0
sum_weegfactor=0
  do k=1,aantal
      third_cum(k)=((snelheid(k)-gewogen_gemiddelde)**3)*weegfactor(k)
      sum_third_cum=sum_third_cum + third_cum(k)
      sum_weegfactor=sum_weegfactor + weegfactor(k)
  enddo
third_moment=(sum_third_cum/sum_weegfactor)
end subroutine third

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subroutine fourth(aantal,snelheid,weegfactor,gewogen_gemiddelde,fourth_moment)
implicit none
integer :: k,aantal
real, dimension(aantal) :: snelheid,weegfactor,fourth_cum
real :: gewogen_gemiddelde,fourth_moment,sum_fourth_cum,sum_weegfactor

sum_fourth_cum=0
sum_weegfactor=0
  do k=1,aantal
      fourth_cum(k)=((snelheid(k)-gewogen_gemiddelde)**4)*weegfactor(k)
  enddo
sum_fourth_cum=sum_fourth_cum + fourth_cum(k)
sum_weegfactor=sum_weegfactor + weegfactor(k)
enddo
fourth_moment=(sum_fourth_cum/sum_weegfactor)

end subroutine fourth

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine weighed(aantal, snelheid, weegfactor, min, max, nr_of_slots, output)
implicit none
integer :: i,k,aantal,nr_of_slots
real :: min,max,his2,his4,delta,time
real, dimension(aantal) :: snelheid, weegfactor
real, dimension(nr_of_slots) :: his1,his3
real, dimension(nr_of_slots,3) :: output

do k=1,nr_of_slots
   his1(k)=0
   his2=0
   his3(k)=0
   his4=0
endo

delta=(max-min)/real(nr_of_slots)

do i=1,aantal
   k=int((snelheid(i)-min)/delta)+1
   his1(k)=his1(k)+1d0 !voor het deel excl wgf
   his2=his2+1
   his3(k)=his3(k)+weegfactor(i) !voor het deel incl wgf
   his4=his4+weegfactor(i)
endo

time = min

do k=1,nr_of_slots
   output(k,1)= time !output zonder weegfactor
   output(k,2)= (his1(k)/his2) !output met weegfactor
   output(k,3)= (his3(k)/his4)
   time=time+delta
endo

end subroutine weighed