Cranfield University

Julian Turnbull

A novel approach to reduce the computation time for CFD; hybrid LES–RANS modelling on parallel computers

School of Engineering

Ph.D.

Cranfield University

School of Engineering

Department of Process and Systems Engineering

Ph.D.
2003

Julian Turnbull

A novel approach to reduce the computation time for CFD; hybrid LES–RANS modelling on parallel computers

Prof. C. P. Thompson

2003

This thesis is submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy

Acknowledgements

I would like to thank my friends and colleagues, but with special thanks to Pip Jones and Dr David Wei for many informal discussions. I would like to thank the staff of Ansys CFX for their help during my research, especially Dr Ian Jones, Dr Nigel Wilkes and Dr Christiane Montavon.

This work would not have been possible without the help of the staff of the computer centres both here and at Cambridge, Chris Cooper, Dr Les Oswald, Kenneth Caines and Nick Maclaren, and the resourcefulness of the secretaries, Rachael Wiseman and Sue Randall.

Thanks once again to my supervisors, for help, advice, criticism and comment, Dr Sam Yang, Dr Simon Leefe, Dr Derek Colman and Prof. Chris Thompson.

Finally, many thanks to my wife, Caroline.

Contents

List of Figures

List of Tables

Abstract

Large Eddy Simulation is a method of obtaining high accuracy computational results for modelling fluid flow. Unfortunately it is computationally expensive limiting it to users of large parallel machines. However, it may be that the use of LES leads to an over-resolution of the problem because the bulk of the computational domain could be adequately modelled using the Reynolds averaged approach.

A study has been undertaken to assess the feasibility, both in accuracy and computational efficiency of using a parallel computer to solve both LES and RANS type turbulence models on the same domain for the problem flow over a circular cylinder at Reynolds number 3 900

To do this the domain has been created and then divided into two sub-domains, one for the LES model and one for the $k - \varepsilon$ turbulence model. The hybrid model has been developed specifically for a parallel computing environment and the user is able to allocate modelling techniques to processors in a way which enables expansion of the model to any number of processors.

Computational experimentation has shown that the combination of the Smagorinsky model can be used to capture the vortex shedding from the cylinder and the information successfully passed to the k-ε model for the dissipation of the vortices further downstream. The results have been compared to high accuracy LES results and with both $k - \varepsilon$ and Smagorinsky LES computations on the same domain. The hybrid models developed compare well with the Smagorinsky model capturing the vortex shedding with the correct periodicity.

Suggestions for future work have been made to develop this idea further, and to investigate the possibility of using the technology for the modelling of mixing and fast chemical reactions based on the more accurate prediction of the turbulence levels in the LES sub-domain.

Nomenclature

Roman

A	anisotropy measure	—
$\mathcal{C}$	cross stress term	$[m^2\, s^{-2}]$
C_D	constant	—
C_D	Kolmogorov constant	—
C_s	Smagorinsky constant	—
C_{ε_1}	constant	—
C_{ε_2}	constant	—
C_μ	constant	—
c	constant	—
D	characteristic diameter	$[m]$
D	cylinder diameter	$[m]$
$E(\kappa)$	energy at specific wavenumber	$[m^2\, s^{-2}]$
E_s	unresolved energy	$[m^2\, s^{-2}]$
F	body force	$[kg\, m\, s^{-1}]$
f	frequency	$[s^{-1}]$
f_s	smoothing function	—
h	grid spacing	$[m]$
$\mathcal{J}$	scalar in the dynamic model	$[m^2\, s^{-2}]$
k	turbulent kinetic energy	$[m^2\, s^{-2}]$
k^r	unresolved kinetic energy	$[m^2\, s^{-2}]$
$\mathcal{L}$	Leonard stress	$[m^2\, s^{-2}]$
ℓ	length	$[m]$
ℓ_{DI}	lengthscale	$[m]$
ℓ_{EI}	lengthscale	$[m]$
ℓ_f	filter width	$[m]$
ℓ_m	Prandtl mixing length	$[m]$
ℓ_o	lengthscale	$[m]$

$\mathcal{P}$	production	$[m^2 s^{-2}]$
$\mathcal{R}$	Reynolds stress terms	$[m^2\, s^{-2}]$
S_{ij}	rate of deformation tensor	$[s^{-1}]$
u_i	instantaneous velocity	$[m\, s^{-1}]$
$\bar{u_i}$	average velocity	$[m\, s^{-1}]$
u_i'	fluctuating velocity	$[m\, s^{-1}]$
u_o	characteristic velocity	$[m\, s^{-1}]$
$\hat{u}_i$	resolved velocity	$[m\, s^{-1}]$
$\tilde{u}_i$	unresolved velocity	$[m\, s^{-1}]$
x_i	coordinate direction	—

Greek

δ_{ij}	kronecker delta	—
Δ	filter size	$[m]$
Δt	time step size	$[s]$
Δx	grid size	$[m]$
η	turbulent length microscale	$[m]$
ϵ	error	—
ε	dissipation of turbulent kinetic energy	$[m^2\, s^{-3}]$
κ	wavenumber	$[m^{-1}]$
λ	Taylor microscale	$[m]$
μ	dynamic viscosity	$[kg\, m^{-1}\, s^{-1}]$
μ_t	turbulent viscosity	$[kg\, m^{-1}\, s^{-1}]$
ν	kinematic viscosity	$[m^2\, s^{-1}]$
ρ	density	$[kg\, m^{-3}]$
σ_k	constant	—
σ_ε	constant	—
τ	turbulent timescale	$[s]$
τ^r	unresolved stress tensor	$[kg\, m^{-1}\, s^{-1}]$
τ_{ij}	stress tensor	$[kg\, m^{-1}\, s^{-1}]$
τ_o	timescale	$[s]$
υ	turbulent velocity microscale	$[m\, s^{-1}]$
ω	vorticity	$[s^3\, m^{-2}]$

Dimensionless Groups

Re	Bulk Reynold's number	$\frac{\rho v \ell}{\mu}$
Re_D	Cylinder Reynolds number	$\frac{\rho v D}{\mu}$
Re_t	Turbulent Reynolds number	$\frac{k^2}{\nu\varepsilon}$
Re_μ	Microscale Reynolds number	$\frac{\eta\upsilon}{\nu}$

Chapter 1

Introduction

Stirred tanks are a very important item of hardware on a chemical plant. They are used in chemical and biological industries [87]. The importance of understanding the physical processes involved in stirred tank reactors can be easily assessed by a casual glance through the pages of '*Chemical Engineering Science*' or '*AIChE Journal*' where there are regular articles, as this is a research for a global, multi-million Euro industry.

In an attempt to give insight into the processes in the stirred tank, and in an attempt to model the processes at real scale faster than the production, commission, testing and operation of a pilot plant, much time and effort have been spent in using CFD to simulate the flow in these vessels.

CFD has now been used to model multiphase mixing[71, 90], mixing of miscible and immiscible fluids and non-newtonian fluid mixing. These may or may not be combined with chemical reactions[9, 38, 39], degradation of the fluids and calculation of power draw or associated dimensionless numbers[121].

However, all of the simulations are plagued by the underprediction of turbulence in the mixing vessel which is demonstrated when compared to experimental results[120, 78, 50, 122]. The prediction of the levels of turbulence near the impeller is generally 10-50% of that measured in experiments, though qualitatively correct[76]. The velocity field is generally well represented however. Some of these errors will be due to user error and incorrect application of the CFD models[22], and some due to the failings of the turbulence models themselves.

Large Eddy Simulation (LES) offers a possible way forward in the simulation of stirred tanks[88, 86, 2]. The work undertaken looks promising, and the explicit calculation of the large scales so responsible for the mixing in the vessel is conceptually closer to the process being modelled than the use of mean flow with the addition of a turbulent diffusion.

However, LES comes at a price, and this is a massively increased CPU overhead and associated memory cost for the explicit calculation of the flow on a very fine grid with very small timesteps[19, 20]. We describe here an alternative study investigating the possibility of using a combination of the LES and Reynolds averaging techniques to give a better prediction of the turbulence near the impeller, but without the cost of running a LES simulation over the whole tank.

Given that the regions of the stirred tank where the bulk of the turbulence energy is created and dissipated is known it would be advantageous to be able to model this region with Large Eddy Simulation. This region is the highly anisotropic, high shear region in the vicinity of the impeller. Where the turbulence is less anisotropic, and the turbulence can be more represented by a time averaged value a Reynolds averaged approach could be used. This would then reduce the CPU and memory overhead speeding up the time to delivery but would still enable the more accurate modelling of small scale processes where the smallest lengthscales and fastest timescales are. In addition, by only using the LES technique where the RANS modelling is poor, the driving force for the process is the optimum level of results from the CPU input.

A CFD model of a stirred tank reactor, is a complex model of an equally complex transient geometry and this means that the quantification of the errors arising form the hybrid simulation could well be compounded by the errors arising from the inadequate modelling of the system. In order to investigate the stability and effectiveness of a ‘hybrid’ model of both LES and RANS modelling a simpler geometry that has received both intensive modelling and experimental work has been used. That is the flow over a cylinder at a Reynolds number (based on the bulk upstream fluid properties and the cylinder diameter) of 3 900

Chapter 2

Background

2.1 Mathematics of fluid flow

The equations for fluid flow are given by the Navier-Stokes equations. Although commonly referred to as the Navier-Stokes equations, strictly this is the name for the conservation of momentum equation as is highlighted below

The first governing equation is the conservation of mass. This states that the difference between the mass flow in and the mass flow out must be due to accumulation of mass in the system by the varying of the density. For an incompressible fluid, the inflow must equal the outflow. Thus,

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0. \tag{2.1}$$

However this thesis is investigating the properties of incompressible flow and so this reduces to,

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{2.2}$$

The conservation of momentum equation is the application of Newton's second law to the fluid. This is the Navier-Stokes equation and is written (for an incompressible fluid),

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = \frac{-1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + F \tag{2.3}$$

In Equation(2.3) the parts of the equation represent the following features of the flow. The first term on the left hand side of the equation, represents the rate of change of the velocity with time. The second the rate of change of the velocity with the position. These two may be combined to give the left hand side in terms of the substantive derivative

$$\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \tag{2.4}$$

The first term on the right of Equation(2.3) is the pressure term, which provides a driving force for the movement of the fluid, the second term on the right is for the viscous stresses. The final term represents the body forces acting on the fluid, these may be buoyancy, gravitation or magnetism.

However, to solve the conservation equations directly is very challenging due to their non-linear behaviour. These non-linearities arise at high Reynolds numbers, they are complex and contain small instabilities. In many engineering applications the small features of the flow have characteristics of random fluctuations known as turbulence. In order to solve the equations approximations are used. The most common being spatial or time averaging [73, 117]. For spatial averaging the grid acts as a filter and a model is used to represent the 'unresolved' scales that are smaller than the filter. For temporal averaging an assumption that there is a clear distinction between the timescales of the mean and fluctuating parts of the flow is invoked. Either way the flow may be represented,

$$u_i = \langle u_i \rangle + u_i' \tag{2.5}$$

where the term in the brackets represents the averaged or resolved term, and the prime the unresolved or fluctuating component of the flow.

Applying this approximation to the equations of continuity and momentum gives,

$$\frac{\partial \langle u_i \rangle}{\partial x_i} = 0 \tag{2.6}$$

$$\frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial \langle u_i \rangle \langle u_j \rangle}{\partial x_j} + \frac{\partial \langle u_i' u_j' \rangle}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j \partial x_j} \tag{2.7}$$

The term $\langle u_i' u_j' \rangle$ represents the average of the unknown part of the model and it is this term which is modelled by subgrid scale models (in Large Eddy

Simulation) or turbulence models if the Reynolds averaged approach is used.

Spatial averaging, or filtering is achieved by applying a filter of size Δ to the flow. Often the filter size is related to the grid size h, the simplest approximation is $\Delta = h$.

$$\langle f(x) \rangle = \int \mathcal{G}(x, x') f(x') dx' \tag{2.8}$$

where $\mathcal{G}$ is the filter function. Most commonly a Gaussian distribution, or a 'top hat' cutoff filter in either fourier space (frequency) or time. The Gaussian distribution remains the same when transformed from wavenumber to real space, but the other two alter when moved between spaces which can introduce errors into the simulation.

The transient averaging is assessed in the following way,

$$\bar{\phi} = \frac{1}{T} \int_t^{t+T} \phi dt \tag{2.9}$$

This leaves a fluctuating component of ϕ. This is the Reynolds decomposition and is written in completion as,[117]

$$\phi = \bar{\phi} + \phi' \tag{2.10}$$

Where ϕ is the instantaneous value of the variable being modelled, $\bar{\phi}$ is the mean value of the variable, and ϕ' the fluctuating value of the variable. As you can see this is based on the approximation that the variable can be modelled in this way. ϕ' is usually defined in a way such that it will have fixed mean and random distribution. There are further problems raised if the mean flow $\bar{\phi}$ is slowly varying and this is briefly addressed by Roussinova [95]. So it follows that strictly this is a modelling approach that can only be applied to stationary turbulence.

All turbulence that is modelled by this methodology should be steady turbulence, ie the fluctuations are about a point and the velocity varies little. Turbulence models are often developed using experimental data from 'simple' geometries that are as controlled as possible. This may be one of the reasons why the modelling of flow in stirred tanks is so often compromised as the model is being forced to operate in an area where strictly speaking it is not valid.

The ergodic hypothesis is that transient averaging, spatial averaging and ensemble averaging will all give the same answer. That is, if the flow is both stationary and homogeneous with time. Stationary turbulence is usually averaged using transient averaging and on average does not vary with time. Homogeneous turbulence is usually averaged by spatial methods and is considered uniform in all directions.

In the following discussion and development the use of spatial averages has been denoted $\hat{\phi}$ for the resolved and $\tilde{\phi}$ for the unresolved part. The transient averaging has been denoted $\bar{\phi}$ and the fluctuating part ϕ'.

2.2 Overview of turbulence

2.2.1 Introduction

For a viscous fluid to move a force must ba applied to it. Initially at low velocities the flow is laminar, and this is seen in pipes for example as a quadratic velocity profile. This is because the forces have overcome the resistance to movement slightly. The packets of fluid will slide over one another with such a small difference in velocity that there is not enough friction between the packets to provide a basis for instability. This is seen particularly at the boundary layer where the laminar flow increases in velocity as the distance from the wall is increased. The velocity profile is linear and the friction generated is too small to cause instabilities at the surface. However, as the velocity at the wall is increased instabilities will arise as the friction between the wall and the fluid causes the fluid at the wall to slow down. This will then mean that the next layer of fluid out will be travelling much faster than the layer next to the wall and will therefore overtake it and expand to take its place. This expansion and the fact that it then comes into contact with the wall will then slow that packet of fluid down dramatically. This causes a random element to exist in the flowfield and this is known as turbulence.

A measure of the ratio of viscous to inertial forces is the Reynolds number. It is a dimensionless number that is used to suggest whether the flow is turbulent or not. If the viscous forces dominate then the flow is laminar, and if the inertial forces then the flow is turbulent. The turbulent Reynolds number is given by,

$$Re = \frac{\rho u' D}{\mu} \tag{2.11}$$

For a pipe the transition to turbulence happens at a Reynolds number of $2\,000 - 3\,000$ and if the flow is in this region the fluid will possess both laminar and turbulent flow patterns. If the wall of the pipe were to be made very smooth then Re could be increased before the instabilities set in [14]. This highlights another important element of the turbulence in that it has to be initiated. It is often a shear force that is introduced to the flow that then initiates the turbulence.

The turbulence is a property of the fluid in that situation, but is not a physical property of the fluid. It is affected by the physical properties and the boundary conditions, and this will determine the intensity of the turbulence $\overline{u_i' u_i'}$ As mentioned before the viscosity acts on the turbulence to try to damp it out so that the flow will return to laminar flow with no fluctuations. If there were no energy put into the system then the turbulence would die away due to the action of the viscosity of the fluid.

Turbulence is dissipative [111]. The constant fluctuations act to provide a more random flowfield and this acts to disperse the flow and enhance mixing. These fluctuations are caused by the shear or vortex actions of the fluid.

2.2.2 Visualisation of turbulent flow

Experimentally turbulence is seen as fluctuation in the flow. If the values at a point in a turbulent flowfield are plotted they will be seen to be fluctuating frequently about a mean value, in a way which makes transient averaging very easy to comprehend and forms the basis of experimental studies (eg LDA, PIV).

For isotropic turbulence, where the fluctuations are the same if the coordinate system is rotated about any arbitrary point, the fluctuations in the velocity field are random in size and direction. If the flow is averaged over a long period of time then it can be thought of as a mean flow with turbulent fluctuations about that mean. Long, in this sense in relation to the timescale of the fluctuations themselves. If the turbulent flow, on average, does not vary with time then the turbulence in said to be stationary. The three-dimensional nature of the turbulence is demonstrated by the rotational flow field that is often seen. The rotational flow can be described as eddies or vortices. A loose definition would be that, a vortex has structure in the full 3D way where as an eddy is an instability in the flow. The vortex is commonly seen in the sink as the water leaves the bowl and from here it will be noted that as the vortex becomes smaller (ie smaller diameter) the velocity of the flow increases. This is because the momentum is conserved. Vortex lines are tangential to the vorticity vector and vortex tubes are tube-like surfaces made up of vortex lines. The strength of a tube can be defined as the circulation in a closed path on its surface going once round the tube and remains unchanged with time as the tube is convected [69]. In an incompressible fluid as the tube is stretched in the flow the diameter is reduced and the velocity increased and this is a precursor to the energy cascade as will be discussed later.

2.2.3 The energy spectrum

The largest possible instability will be of the same order in size as the largest dimension in the domain that could initiate the turbulence, the pipe diameter or the impeller diameter being two examples. The largest instability will be affected by the flow, which is similar in magnitude to the characteristic dimension of the domain. It will also be acted upon by the viscosity. The combination of the viscosity and the non-linear terms will act to break up the large scale structure, but will result in the creation of smaller structures. In turn these smaller structures will be acted upon by the viscosity and will be stretched, broken or deformed into yet smaller structures. This however is a dynamic equilibrium and changes with time and position [47]. The difference in scales is easier to identify when the difference is large but there is a transfer of energy from large to small scales in all flows. This transfer is described by the energy cascade and so transfers the energy from the scales of the order of the flow where the turbulence is introduced to the flow down to the scales of the order of the Kolmogorov length (equation 2.13) where the energy is dissipated as heat.

This process is often described as the energy cascade or energy spectrum, and is only complete when the Reynolds number is high enough [111, 67, 116, 58] The name is derived from the fact that the energy of an eddy is related to its size. As the eddies are broken down by the viscosity the energy they contain reduces. However, it must be remembered that one large eddy will break down into many smaller eddies and the energy that may appear missing in an energy balance will have been returned to the fluid as heat.

Turbulence is often categorised in terms of wavenumbers, where the wavenumber is the inverse of the wavelength. Large eddies have long wavelengths, low frequencies and small wavenumbers. At the dissipation lengthscale the frequency and wavenumber are high and the wavelength small. The wavenumber is defined by [79]

$$\kappa = \frac{2\pi f}{u} \tag{2.12}$$

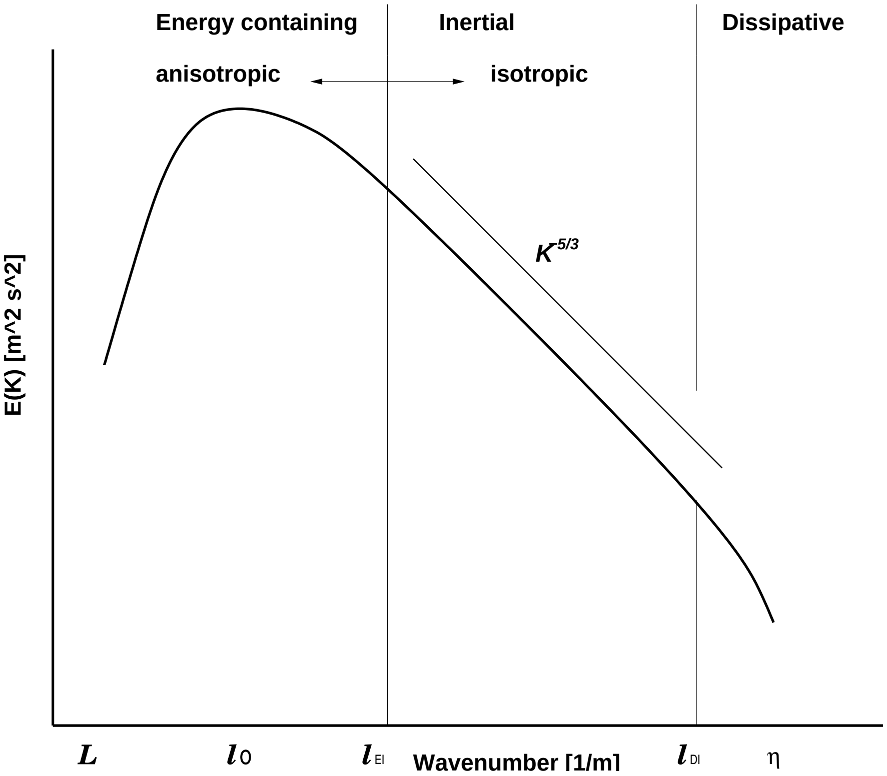

Figure 2.1: Schematic of the power spectrum highlighting major features

where f is the frequency of fluctuation $[Hz]$ and u the velocity $[m\, s^{-1}]$ of the instability. Reynolds averaged models have wavenumber zero as they are unable to capture fluctuations in the flow.

The energy cascade completes the cycle of the turbulence, from the introduction to the flow as a large scale through the gradual action of the viscosity and the reduction in size of the eddies to the dissipation of the energy as heat.

2.2.4 Use of the energy spectrum

It can also be seen that there is no definite changing point between the large and small structures in the flow, but rather a constant spread of eddy sizes. This makes the analysis of the spectrum harder. The physics of the large and the small scales are relatively well understood but in between less in known. The energy spectrum can be divided up into sections. The large scale is where the energy is provided into the turbulence from the mean flow, for if this were not the case the fluctuations would die out. The action of the viscosity on turbulence is always to damp it out and so without an input of energy the fluctuations would be removed to leave a laminar flow. The small scale or viscous subrange is where the fluctuations are indistinguishable from heat. The packets of fluid in the eddies become smaller and smaller until the size of the packet is overcome by the viscosity and the movement becomes the movement of the fluid molecules is random Brownian motion. The size at which this happens is the Kolmogorov lengthscale. It is assumed that energy only enters the spectrum at the top and only leaves the cascade at the bottom. This then means that the connecting line will transfer all of the energy from the large scales to the small scales. This is the inertial subrange and it is found that for fully developed turbulence the gradient of the line is proportional to $\kappa^{-5/3}$ for this section of the line.

2.2.5 Microscales and dissipation

The energy cascade continues until the Reynolds number is sufficiently small that the molecular viscosity acts to dissipate the flow.[84, 111]

Kolmogorov proposed that the velocity and time scales of the turbulent eddies decrease as the lengthscale of the eddy decreases. This forms the basis of the concept of local isotropy. At sufficiently high Reynolds numbers, where the local lengthscale of the turbulence is much smaller than that of the large scale turbulence, the small scale eddies are statistically isotropic. Extrapolating from this Kolmogorov [47] postulated that at sufficiently high values of Re the small scales are uniquely determined by ν and ε only, this is known as Kolmogorov's first similarity hypothesis. The size of these is given by $\ell < \ell_{EI}$ where ℓ_{EI} is the lengthscale where turbulence transforms from anisotropic to isotropic flow. The eddies that are below this size are described as being in the universal equilibrium range.

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \tag{2.13}$$

$$\tau = \left(\frac{\nu}{\varepsilon}\right)^{\frac{1}{2}} \tag{2.14}$$

$$\upsilon = (\nu\varepsilon)^{\frac{1}{4}} \tag{2.15}$$

for the length, time and velocity microscales respectively. If a Reynolds number is constructed from these scales it is of O(1), signifying the laminar and viscous nature of the flow at this point [111].

A smaller scale still is the Batchelor microscale. This signifies the point at which the effects of the turbulent structures are no longer in evidence in the fluid.

2.2.6 Introduction to the large eddies

Richardson observed in 1922 that energy is introduced at the large scales where there is a perturbation of the mean flow[89]. The largest eddy size ℓ_o is comparable to the lengthscale of the domain ℓ and has characteristic velocity u_o which is defined as a function of the velocity and the domain size,

$$u_o = u(\ell_o) \tag{2.16}$$

and is of order of the RMS of the turbulence intensity

$$u' = \left(\frac{2}{3}k\right)^{2/3} \tag{2.17}$$

and is comparable to the mean velocity. This means that the Reynolds number of the large eddies is comparable to that of the mean flow.

Pope [84] states that the bulk of the energy is contained in the large eddies and these are of the size $\ell_{EI} = \frac{1}{6}\ell_o < \ell < \ell_o$ indicating that the large scale eddies at the top of the energy spectrum contain the most energy. Kolmogorov's Second Similarity Hypothesis states that at a high enough Reynolds number, the statistics of the turbulence have a form that is independent of ν.

2.2.7 The inertial subrange

The dissipation of the energy takes place at small scales of size η where the viscosity is important. For a large scale turbulent eddy, $\ell \gg \eta$, and assuming that the Reynolds number is large then a turbulent eddy of this scale will be little affected by the viscosity. Following this rationale Kolmogorov introduced a similarity hypothesis stating that at in high Reynolds number flow, where $\eta \ll \ell \ll \ell_o$ the length is determined by the rate of turbulent energy dissipation (ε) alone and is not dependent on the dynamic viscosity ν. If a lower size of ℓ_{DI} is introduced at size $\ell_{DI} = 60\eta$ then when $\ell_{DI} < \ell < \ell_{EI}$ the eddies are dependent on the rate of dissipation and inertial effects alone. The viscosity is negligible. The energy contained by an eddy in this range is defined by

$$E(\kappa) = C_K \varepsilon^{2/3} \kappa^{-5/3} \tag{2.18}$$

Velocity and length scales can be calculated for the eddies in the inertial subrange and from these the rate of transfer of energy from eddies of size ℓ to smaller eddies. The rate of transfer of energy comes out as ε which suggests the rate of transfer is independent of the lengthscale in the inertial subrange.

The inertial subrange is the region where the energy contained in the eddies scales with the $\kappa^{-\frac{5}{3}}$. This is the region where the viscosity acts to influence the turbulent eddies and so the eddies are reduced in size. The reduction in size corresponds to a reduction in energy, though energy is conserved by either the transformation into heat or the production of many smaller eddies. Much of the turbulence modelling work assumes that the flow of energy is one way, that is, eddies only get smaller and there is no recombination of eddies to form larger structures. This approximation means that the energy lost by the larger eddies is equal to that dissipated at the Kolmogorov scale. All of the energy contained in the large scales will be dissipated as heat, at the Kolmogorov scale, by the action of viscosity.

Statistically this range is often seen as a discrete set of wavenumber sizes with the energy cascading from one size to the next. Strictly speaking the cascade is continuous, but the model allows for an analysis of the range. This demonstrates that a majority of the energy into an eddy comes form one which is slightly larger[111], and demonstrates the action of viscosity on turbulence in gradually damping it out.

The rate of dissipation at the largest scales is governed by the rate of dissipation by the largest eddies. A large eddy contains energy on the order of u_o^2 with a timescale

$$\tau_o = \frac{\ell_o}{u_o} \tag{2.19}$$

so the rate of energy transfer scales with

$$\frac{u_o^2}{\tau_o} \tag{2.20}$$

at high Reynolds number.

The transfer of the energy in this region increases dramatically as the inertial subrange is descended. [84] This can be explained using an energy balance. For there to be a constant amount of energy in the system, the system could contain one large eddy, or several smaller ones. This is seen at every wavenumber, such that the number of small instabilities is much greater than the number of large scale instabilities. These smaller eddies move faster and break down into more yet smaller eddies. In this way the transfer of energy increases. The rate of transfer of energy remains constant. This is because more energy cannot be dissipated than is present in the fluid. The dissipation rate is ε and for a steady high Reynolds number flow this is equivalent to the production of turbulence.

2.2.7.1 Backscatter

It is possible for the smaller eddies to organise themselves in an ordered way and so form a larger structure. This transfer of energy back up the energy spectrum is referred to as backscatter. It causes problems with many theories as the basic understanding is that the transfer of turbulence energy is unidirectional with the flow from large to small. Backscatter complicates the matter but the return of the energy to the large scales is thought to increase the mixing ability in the flow. Research into this area (eg [123]) is still in its infancy and the effect is ignored for the purpose of this study.

2.3 The Reynolds averaged equations

When Reynolds decomposition is substituted into the Navier-Stokes equation for momentum the following is obtained.

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu \left(\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} \right) - \frac{\partial \overline{u'_i u'_j}}{\partial x_j} \tag{2.21}$$

This then is all resolvable except for the $\rho\overline{u'_i u'_j}$ term, and it is this term that is the subject of the turbulence model.

Now the shear stress tensor τ_{ij} is defined in the following way,

$$\tau_{ij} = -\overline{\rho u'_i u'_j} \tag{2.22}$$

and a modelling assumption is made, that the turbulent stresses are approximately equal to the large scale shear.

Near a wall, the shear in the wall normal direction (y) is defined by,

$$\tau_{ij} = \mu \frac{\partial u_i}{\partial x_j} \tag{2.23}$$

where the viscosity is acting to reduce the shear. In the way that the turbulence of the fluid acts to smooth out the profile of the flow it is represented as an increase in the viscosity of the fluid. This is the eddy viscosity or Boussinesq hypothesis and forms a basis for many of the turbulence models. This is given in term of the turbulent or eddy viscosity μ_t. This then allows the turbulent stress to be modelled because it is related to the mean shear thus,

$$\tau_{ij} = -\rho\overline{u_i' u_j'} \approx \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3}\rho k \delta_{ij} \tag{2.24}$$

The eddy viscosity μ_t is proportional to the product of the turbulent viscosity scale and a length scale. The length scale used is derived in terms of the dissipation by the equation $\varepsilon \sim k^{\frac{3}{2}}\ell$ which was obtained for isotropic turbulence, which allows the description of the whole of the turbulence spectrum by the single lengthscale. This is probably the weakest assumption in the two equation models [109]

2.3.1 Law of the wall

Rapid variation of the velocity near the wall requires the use of fine cells to give adequate resolution for the modelling process. However, an approximation was used which assumes that the flow is developed and the average velocity profile is known. This came to be known as 'the law of the wall'.

In the vicinity of the wall the dimensionless distance to the wall is given by

$$y^+ = \frac{\Delta y_p}{\nu} \sqrt{\frac{\tau_w}{\rho}} \tag{2.25}$$

where Δy_p is the distance from the wall to the first cell centre, τ_w is the shear stress on the wall. It is made dimensionless by the inclusion of the density and viscosity. If the value of y^+ is less than 11.63 then the flow at that point can be considered to be entirely laminar. Formulations have also been derived for the dimensionless velocity and turbulence properties near the wall.

As such it is an approximation that allows the cell next to the wall to be positioned at a y^+ value in the order of 100 and thereby not have to explicitly model the development of the flow next to the wall. This overcomes the problem of expanding the cells away from the wall. If the cell were small enough to capture the molecular sublayer at then the whole domain would either have to be modelled with cells of this size or the cells expanded quickly, though this could introduce differencing errors if not done carefully. The most common being that of the Van Driest type relation, and many are still in use today [33]

Many have therefore sought to develop models that are not reliant on the law of the wall but can be used to resolve the sublayer. The disadvantage of this is that generally more cells are required as the minimum $y^+ \leq 10$. However, those in favour of this method argue that it should be used as the near wall region is insufficiently universal for a wall law to be applied [37]

2.3.2 The k-equation

The transport equation for k, the turbulent kinetic energy per unit mass $[m^2s^{-2}]$ is given by,

$$\rho\frac{\partial k}{\partial t} + \rho\bar{u}_j\frac{\partial k}{\partial x_j} = \tau_{ij}\frac{\partial u_i}{\partial x_j} - \rho\varepsilon + \frac{\partial}{\partial x_j}\left[\mu\frac{\partial k}{\partial x_j} - \frac{1}{2}\rho\overline{u_i'u_i'u_j'} - \overline{p'u_j'}\right] \tag{2.26}$$

where ε is the rate of dissipation of turbulent kinetic energy. These terms represent in turn the transient change in k, and the convection on the LHS of the equation. This is balanced on the RHS by the production, dissipation, and then the three diffusional terms, molecular, turbulent transport and pressure diffusion.

The LHS of equation 2.26 gives the change of k with time and with convection around the domain. The production term represents the generation of turbulent energy from the mean flow, the work of the mean strain rate against the turbulent stresses. The dissipation term is the rate at which the turbulent energy is absorbed back into the fluid as thermal energy by the effect of the viscosity. Molecular diffusion is the transport of the turbulent energy by molecular diffusion in the fluid, whereas turbulent transport is the transport of the turbulence by the turbulence itself. Finally the pressure diffusion term is the the transport of turbulence by pressure-velocity interaction.

However, to reach a transport equation that can be modelled approximations are introduced. The Reynolds stress tensor is modelled by the use of the Boussinesq hypothesis,

$$\tau_{ij} = 2\mu_t S_{ij} - \frac{2}{3}\rho k \delta_{ij} \tag{2.27}$$

Where the last term is introduced so that a normal stress reduces to $\tau_{ii} = \frac{-2}{3}\rho k$. This is because, an incompressible fluid has $S_{ii} = 0$, but $\tau_{ii} \neq 0$.

The turbulent transport and pressure diffusion terms as given in equation 2.26 cannot be represented as they depend on the fluctuating components of velocity and pressure. The transport term is modelled in analogy to the gradient - diffusion term [117],

$$\frac{1}{2}\overline{\rho u'_i u'_i u'_j} + \overline{p' u'_j} = -\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_j} \tag{2.28}$$

Where σ_k is the Prandtl-Schmidt number of order 1. The pressure diffusion term is small and so is grouped with the turbulent transport.

Dissipation is a more difficult parameter to assess. On purely dimensional grounds the turbulent kinetic energy, dissipation and turbulent length scale can be related by

$$\varepsilon \sim \frac{k^{3/2}}{\ell} \tag{2.29}$$

Where ℓ is the *integral scale* which is a characteristic length in the turbulence. For example, a function of the distance from the wall in a boundary layer problem or the thickness of a jet or wake.

The transport equation for k has now become,

$$\rho\frac{\partial k}{\partial t} + \rho\bar{u}_j\frac{\partial k}{\partial x_j} = \tau_{ij}\frac{\partial \bar{u}_i}{\partial x_j} - \rho\varepsilon + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.30}$$

2.3.3 Dissipation

The rate of dissipation of the turbulent kinetic energy back into the fluid is denoted by ε. Strictly speaking this is the dissipation per unit mass, and has units $[m^2 s^{-3}]$.

It is defined by the following equation,

$$\varepsilon \equiv \nu\overline{\frac{\partial u'_i}{\partial x_k}\frac{\partial u'_i}{\partial x_k}} \tag{2.31}$$

The transport equation for epsilon is defined [117],

$$\begin{aligned}
\rho\frac{\partial \varepsilon}{\partial t} + \rho\bar{u}_j\frac{\partial \varepsilon}{\partial x_j} = & - 2\mu\left[\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}} + \overline{\frac{\partial u_k'}{\partial x_i}\frac{\partial u_k'}{\partial x_j}}\right]\frac{\partial \bar{u}_i}{\partial x_j} - 2\mu\overline{u_k'\frac{\partial u_i'}{\partial x_j}\frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_j}} \\
& - 2\mu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_m}\frac{\partial u_k'}{\partial x_m}} - 2\mu\nu\overline{\frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_m}\frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_m}} \\
& + \frac{\partial}{\partial x_j}\left[\mu\frac{\partial \varepsilon}{\partial x_j} - \mu\overline{u_j'\frac{\partial u_i'}{\partial x_m}\frac{\partial u_k'}{\partial x_j}} - 2\nu\overline{\frac{\partial p'}{\partial x_m}\frac{\partial u_j'}{\partial x_m}}\right] \qquad (2.32)
\end{aligned}$$

The terms on the LHS are the same as for the k equation, that is the transient changes and the changes due to convection. On the RHS of the equation the terms represent the production and dissipation of dissipation, the third and fourth terms represent the molecular diffusion of dissipation, finally there is the turbulent transport of dissipation.

The transport equation for ε is complicated and uses fluctuating values to derive the answer. This is not however possible in practice and so the equation is modelled and the simplified models tuned for practical considerations using experimental results or DNS data [117].

It is important to recognize that the use of the inertial subrange and the assumption of no backscatter means that the dissipation of k at the larger lengths will match the dissipation at the Kolmogorov lengthscale.

The transport of dissipation by this equation is correct on dimensional grounds and physical reasoning alone. That is not to say that the equation is incorrect, but that it is modelled and therefore an approximation by closure of the actual ε transport equation. This is often considered to be the weakest assumption of the model.

2.4 Turbulence models

There are many turbulence models in use and often modifications are made to the constants for a models to perform better for a given problem. A brief overview of the models are given in increasing complexity.

2.4.1 Algebraic

Prandtl's Mixing length theory expresses the eddy viscosity as a function of the mixing length in the system ℓ_m. This is characteristic of the level of turbulence, and derived as an analogy to the kinetic theory of gasses [52].

If a volume of fluid at a point is moving with velocity u_i and there exists a velocity gradient in the y direction then the velocity at point $y + \delta y$ is $u_i + \delta u_i$. The change in velocity is the difference between the velocities at the two points displaced by the distance ℓ_m. If the velocity at the first point is known and the gradient known, a first approximation of the velocity at the second point is

$$u_i|_{point2} = \frac{\partial u_i}{\partial x_j} \delta y \ell_m \tag{2.33}$$

The velocity u_j must therefore change as well but this is assumed to be proportional to the change in u_i and so calculated from it. This means that the turbulent viscosity can be calculated as follows

$$\rho \overline{u_i u_j} = \rho \ell_m^2 \left(\frac{\partial u_i}{\partial x_j} \right)^2 \tag{2.34}$$

However, knowledge of the flow is required to be able to assess the mixing length. This is usually assumed to be proportional to the distance from the wall. It is possible to use ℓ_m as a function of k which ties the model to experimental data as this is where the values of k have to originate from.

2.4.2 1-equation models

2.4.2.1 Older models

In an attempt to overcome the shortcomings of the mixing-length model the next development was the implementation of the $k - \ell$ model whereby the value of the turbulence energy was modelled and thereby the size of ℓ could be calculated allowing the calculation of a turbulent viscosity.

The k-equation is derived from the momentum equation by multiplying through by k and then approximations are made. A complete description of this is given in the next section. The k equation is slightly different in that it is dependent upon ℓ thus

$$\rho\frac{\partial k}{\partial t}+\rho\bar{u}_j\frac{\partial k}{\partial x_j}=\tau_{ij}\frac{\partial\bar{u}_i}{\partial x_j}-C_\mu\rho\frac{k^{3/2}}{\ell}+\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.35}$$

$$\varepsilon=C_\mu\frac{k^{3/2}}{\ell} \tag{2.36}$$

where $C_\mu=0.09$ and $\sigma_k=1.0$.

2.4.2.2 The Spalart-Allmaras model

Rather than model the turbulence itself and from that infer the turbulent viscosity, Spalart and Allmaras [104] created a model based upon the transport equation for the turbulent viscosity directly. It has use mainly in the aerospace field of research [84] and as such has had considerable success in the fields of flows over high-lift aerofoils [32, 45]. Less commonly it has been used for flow in a stirred tank reactor [6], though this is essentially a 'publicity' paper and lacking in experimental validation.

The model uses and empirical approach to define the terms in the transport equation for ν_t, the kinematic turbulent viscosity

$$\begin{aligned}\frac{D\breve{\nu}_t}{Dt} &= c_{b_1}(1-f_{t_2})\breve{S}\breve{\nu}\\ &+ \frac{1}{\sigma}(\nabla.((\nu+\breve{\nu})\nabla\breve{\nu})+c_{b_2}(\nabla\breve{\nu})^2)\\ &- \left(c_{w_1}f_w-\frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\breve{\nu}}{d}\right)^2+f_{t_1}\Delta\bar{u}^2\end{aligned} \tag{2.37}$$

$$\tag{2.38}$$

where the production of the turbulent viscosity is given by,

$$\mathcal{P}=c_{b_1}S\nu \tag{2.39}$$

the diffusion given by

$$\mathcal{D}=\frac{1}{\sigma}(\nabla.((\nu_t\nabla\nu_t)+c_{b_2}(\nabla\nu)^2) \tag{2.40}$$

and the destruction given by

$$\mathcal{E} = c_{w_1}\frac{\nu}{d} \tag{2.41}$$

And in order to make the model scale to the wall correctly, working variables are created. For example $\breve{\nu}$ is equal to ν_t except in the viscous region where it will agree with the experimental and theoretical results. These scalings are

$$\begin{aligned}
\nu_t &= \breve{\nu} f_{v_1} \\
f_{v_1} &= \frac{\chi}{\chi^3 + c_{\nu_1}^3} \\
\chi &= \kappa y^+ \\
\breve{S} &\equiv S + \frac{\breve{\nu}}{\kappa^2 d^2} f_{\nu_2} \\
f_{\nu_2} &= 1 - \frac{\chi}{1 + \chi f_{\nu_1}}
\end{aligned} \tag{2.42}$$

and with the addition of the terms introduced to capture the correct transition from laminar to turbulent boundary layer flow, have been used to create a model that is relatively insensitive to the boundary conditions and is numerically stable giving good results [32].

2.4.3 2-equation models

2.4.3.1 Introduction

For a long time, two equation turbulence models offered the simplest complete method of modelling the turbulence with no intervention from the user. Frequently, the models use k is the turbulent kinetic energy, ε the rate of dissipation and ω the square of the vorticity fluctuations, where the modelled equations relate to each other and to the lengthscale ℓ thus;

$$k = 0.5(\overline{u_i' u_i'}) \tag{2.43}$$

$$\omega = \frac{\varepsilon}{(C_D k)^2} \tag{2.44}$$

$$\varepsilon = \nu \overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}} \tag{2.45}$$

$$\ell = C_D \frac{k^{\frac{3}{2}}}{\varepsilon} \tag{2.46}$$

This interrelation of the variables suggests that the models may be different in mathematical form alone and not in content. Launder and Spalding address these issues [56]. Firstly, it is not possible to transform the equations from one form to another easily without introducing extra terms into the equations, this is especially true for the third-order correlations that are approximated. Secondly, it is not known which term is modelled the most correctly, though this is a problem specific problem. Thirdly, there is a difference in the modelling of flow near a wall, and the fitting of these results to experimental data. Launder and Spalding conclude that this third reason is the most important and subsequently choose the k-ε model for their research.

2.4.3.2 The k-epsilon model

Originally conceived in 1972 [44], the k-ε model is now a very common model used by industry today [11]. Many of the commercial CFD packages use it as their default model [115]. The model was first used to predict the relaminarization of flow over a flat plate, and as such was working with developed flow with an inherent law of the wall approximation which allowed the model to work. Today the model is used in many more cases and often where there is separation or when the flow is not developed and still produces adequate results. Rodi [92] stated in his 1980 state of the art review that the model was robust and reliable and even with shortcomings gave robust and reliable results.

The model is in two parts. The k equation represents the turbulence energy contained in the eddies of all sizes and as such is a running balance of the generation and dissipation. The ε-equation represents the dissipation of this turbulent energy and so together thy form a dynamic partnership.

This was the first complete model of turbulence. Meaning that it described both the creation of turbulent energy, its transport and its dissipation and also estimated the length scale as well with no intervention from the user. The length scale estimated is ℓ as defined by the Prandtl mixing length model which is approximately the size of one small eddy but much larger than the Kolmogorov lengthscale

The k equation (Equation 2.35) can be approximated from the velocity equation by multiplying through by the fluctuating velocity. The units of k are $[m^2s^{-2}]$ and so a turbulent velocity can be described from the turbulence level as a multiple of $q = \sqrt{k}$ which will have units $[m^1s^{-1}]$ The k equation has relatively few modelled terms in it and so is physically realizable, meaning that it will have bounded values.

The ε equation however contains many more modelled terms and so there is more doubt about it. The modelled version of equation 2.32 is,

$$\begin{aligned} \rho\frac{\partial\varepsilon}{\partial t}+\rho\bar{u}_j\frac{\partial\varepsilon}{\partial x_j} &= C_{\varepsilon 1}\frac{\varepsilon}{k}+\tau_{ij}\frac{\partial\bar{u}_i}{\partial x_j}-C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \\ & +\frac{\partial}{\partial x_j}\left[(\mu+\frac{\mu_t}{\sigma_\varepsilon})\frac{\partial\varepsilon}{\partial x_j}\right] \end{aligned} \tag{2.47}$$

The other important equations to be derived from these modelled equations are

$$\mu_t=\rho C_\mu\frac{k^2}{\varepsilon} \tag{2.48}$$

$$\ell=C_\mu\frac{k^{\frac{3}{2}}}{\varepsilon} \tag{2.49}$$

The value of C_{ε_2} $(=1.92)$ is defined from the decay of homogeneous turbulence. C_μ $(=0.09)$ is calculated from a local equilibrium shear layer and the product of C_{ε_2} and C_μ gives $C_1=0.17$. The size of C_ε (=0.08) is calculated from a boundary layer near a wall.

The values for $\sigma_k(=1.0)$ and $\sigma_\varepsilon(=1.3)$ are chosen to fit experimental date for a round jet as is $C_{\varepsilon 1}(=1.44)$ [92]

In order to remove the need for 'the law of the wall' the 'low Reynolds number k-ε model integrates through the sublayer to predict the flow near the wall. However, there is then a need to have lower y^+ values of the order of 10. The use of the low-Reynolds number modifications allow non-constant shear to be modelled. Launder and Spalding [56] publish values for the modified values of the constants C_μ and C_2 which are functions of the turbulent Reynolds number,

$$C_\mu=0.09\times exp[\frac{-2.5}{1+Re_t/50}] \tag{2.50}$$

$$C_2=1.92\times(1.0-0.3exp[-Re_t^2]) \tag{2.51}$$

where Re_t is the turbulent Reynolds number. In addition the boundary condition for ε is set to zero at the wall and the modelled equation for ε contains an extra term that will vary k as the wall normal distance is increased. The k equation has an extra modelled term for computational rather than physical reasons to match ε near the wall

2.4.4 Reynolds (differential) stress models

The stress models allow for anisotropic turbulence by the modelling of the stress terms individually. As a result of this the Boussinesq hypothesis is not invoked and the model is theoretically better placed to represent anisotropic flow. This means that there are more equations solved but that the results should be more accurate as few real flows are isotropic. However, the Achilles heel of RANS turbulence modelling, the ε equation, remains.

Rather than only model the turbulence energy, these model the transport of the Reynolds stresses $\overline{u_i'u_j'}$. The rationale for this is that they will be able to capture the complexities of the flow for example the anisotropic aspect of the flow because the stresses are all modelled individually rather than the isotropic k in the two-equation models. The turbulent kinetic energy is then calculated directly from the stresses.

The form of the transport equation for the Reynolds stress is,

$$\begin{aligned}\frac{D\overline{u_i'u_j'}}{Dt} &= -(\overline{u_j'u_k'}\frac{\partial \bar{u}_i}{\partial x_k} + \overline{u_i'u_k'}\frac{\partial \bar{u}_j}{\partial x_k}) - 2\nu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}} + \overline{\frac{p}{\rho}(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i})} \\ &\quad -\frac{\partial}{\partial x_k}(\overline{u_i'u_j'u_k'} - \nu\frac{\partial\overline{u_i'u_j'}}{\partial x_k} + \overline{\frac{p}{\rho}(\delta_{jk}u_i' + \delta_{ik}u_j')}) \end{aligned} \tag{2.52}$$

where the terms on the RHS represent the generation, dissipation, pressure-strain and diffusion of the Reynolds stresses.

The terms that need modelling are the pressure-strain and diffusional terms. Early work undertaken to do this was by Rotta, and Daly and Harlow [15, 94], as this was then major problem facing the development of a stress model.

The pressure rate-of-strain term

$$\frac{p}{\rho}\frac{\partial^2 p}{\partial x_j \partial x_j} = \underset{\phi_1}{-\frac{\partial^2}{\partial x_i \partial x_j}\left(u_i'u_j' - \overline{u_i'u_j'}\right)} \underset{\phi_2}{- 2\frac{\partial \bar{u}_i}{\partial x_j}\frac{\partial u_i'}{\partial x_j}} \tag{2.53}$$

Can be seen to be made up of two parts. The first, ϕ_1, is the first term on the right hand side of the equation and represents the 'slow' terms, the returning to isotropy by the redistribution of the Reynolds stresses. The second term, ϕ_2, is the interaction of the pressure strain with the mean flow. It is the 'fast' term as it will react first if the flow is acted upon by a force.

Rotta[94] modelled ϕ_1 using the decay of homogeneous anisotropic turbulence to derive the model for the 'slow' pressure rate-of-strain correlation.

$$R_{ij}^{(s)} = -C_R \frac{\varepsilon}{k} \left(\overline{u_i' u_j'} - \frac{2}{3} k \delta_{ij} \right) \tag{2.54}$$

However, the 'rapid' interaction with the mean flow remained absent until Daly and Harlow [15] who found problems with their assumptions but overall the model gave good results. Launder Reece and Rodi [55] moved the model on more by proposing that only the most dominant part of the 'rapid' half of ϕ_2 need be modelled thus,

$$\phi_2 = -C_2 \left(P_{ij} - \frac{2}{3} P \delta_{ij} \right) \tag{2.55}$$

instead of the whole modelled expression,

$$\phi_2 = -C_2 \left(P_{ij} - \frac{2}{3} P \delta_{ij} \right) - C_3 \left(D_{ij} - \frac{2}{3} P \delta_{ij} \right) - C_4 k \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \tag{2.56}$$

with

$$P_{ij} = -\overline{u_i u_k} \frac{\partial \bar{u}_j}{\partial x_k} - \overline{u_j u_k} \frac{\partial \bar{u}_i}{\partial x_k} \tag{2.57}$$

$$D_{ij} = -\overline{u_i u_k} \frac{\partial \bar{u}_j}{\partial x_k} - \overline{u_j u_k} \frac{\partial \bar{u}_k}{\partial x_i} \tag{2.58}$$

$$P = \overline{u_i u_j} \frac{\partial \bar{u}_i}{\partial x_j} \tag{2.59}$$

Which from their comparison with experiments did not overly reduce the accuracy of the problem.

All of the constants can be expressed as a function of a further constant C_2'

$$C_2 = \frac{C_2' + 8}{11} \tag{2.60}$$

$$C_3 = \frac{8C_2' - 2}{11} \tag{2.61}$$

$$C_4 = \frac{30C_2' - 2}{55} \tag{2.62}$$

$$C_2' = 0.4 \tag{2.63}$$

This was strongly advocated by Hanjalić and Launder [36], and progressed greatly by Launder Reece and Rodi [55] who played particular attention to the pressure-strain correlation especially in the vicinity of a wall. Hanjalić and Launder [37], further extended the model for low Reynolds number flows so that it could be used up to a wall without the need for explicit wall functions.

Most of the Launder *et al* model remains unchanged [84] but the dissipation term is changed so that the dissipation is modified by the turbulent Reynolds number $Re_t = k^2/\nu\varepsilon$. In addition the modelled ε equation is modified so that the substantial derivative of ε scales with Re_t and that the values of ε scale with y^2 near the wall so that the value of $\frac{\varepsilon^2}{k}$ tends to a constant value rather than infinity at the wall.

In addition to this the transport terms have to be modelled as well. Daly and Harlow proposed a model for the triple correlation which was later shown not to be invariant under different combinations of the tensors, and also to be isotropic in nature. [109, 37].

$$\overline{u_i' u_j' u_k'} = C_s' \frac{k}{\varepsilon} \overline{u_i' u_l'} \frac{\partial \overline{u_j' u_k'}}{\partial x_l} \tag{2.64}$$

whereas a model that remains unchanged under the rotation of the tensors would be [55]

$$\overline{u_i' u_j' u_k'} = C_s' \frac{k}{\varepsilon} \left(\overline{u_i' u_l'} \frac{\partial \overline{u_i' u_k'}}{\partial x_l} + \overline{u_j' u_l'} \frac{\partial \overline{u_j' u_k'}}{\partial x_l} + \overline{u_k' u_l'} \frac{\partial \overline{u_i' u_i'}}{\partial x_l} \right) \tag{2.65}$$

From experiments the early authors found this to be the most dominant term for the transport of the stresses. In addition, if the model was to be used at a high enough Reynolds number then the viscous diffusion would be minimal, and the effects of the pressure diffusion could be ignored in comparison to the triple correlation and the pressure rate-of-strain term.

For boundary layer flows, the dissipation term has been modelled as a function of the turbulent Reynolds number so as to capture the highly anisotropic flows near a wall. The dissipation term still tends towards infinity at the wall but the wall flow is able to act on the dissipating eddies to redistribute the stresses. There are two forces that act on the eddies near the wall. The turbulent

Reynolds number ($Re_t = k^2/\mu\varepsilon$) and the effect of the mean flow distorting the eddy. The effect of Re_t is much greater [37] and also easier to model so this term is the only one included, giving the dissipation term as,

$$\mu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}} = \frac{2}{3}\varepsilon\left((1-f_s)\delta_{ij} + \frac{\overline{u_i'u_j'}}{\frac{2}{3}kf_s}\right) \tag{2.66}$$

where $f_s = (1 + \frac{1}{10}Re_t)$.

The dissipation transport equation for the Reynolds stress model is similar to that for the $k - \varepsilon$ model. but with two main modifications.

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_i}\left(C_\varepsilon\frac{k}{\varepsilon}\overline{u_i'u_j'}\frac{\partial\varepsilon}{\partial x_j}\right) + C_{\varepsilon_1}\frac{\mathcal{P}\varepsilon}{k} - C_{\varepsilon_2}\frac{\varepsilon^2}{k} \tag{2.67}$$

The two differences being that $\mathcal{P}$ is evaluated from the Reynolds stress directly rather that from $\mathcal{P}= 2\mu_t\bar{S}_{ij}\bar{S}_{ij}$ and that the diffusion term involves the anisotropic diffusivity.

A summary of developments for the Reynolds stress model [42] precede a new proposal for the dissipation equation. By comparison with DNS results for a variety of flows Jakirlić and Hanjalić [42] refine the approximations for the terms in the dissipation equation and show improvements for flow along a plate, over a step and inside a rotating cylinder. This is achieved without reference to the surface being modelled, and in the bulk flow away from the solid walls the value of the dissipation is as expected. The main improvements are made by using a homogeneous dissipation term along with a stress diffusion term to model the dissipation giving the improvements in predictions compared to DNS down to y^+ values of 0.1. This then is a drawback to the model, in that in order to obtain the accuracy, the cells must be of the order of, or greater in number than, a large eddy simulation.

$$\varepsilon_{ij} = \varepsilon_{ij}^h + \frac{1}{2}\mathcal{D}_{ij}^v \tag{2.68}$$

with

$$\varepsilon_{ij}^h = (1-f_s)\frac{2}{3}\delta_{ij}\varepsilon^h + f_s\frac{\overline{u_i'u_j'}}{k}\varepsilon^h \tag{2.69}$$

and

$$\mathcal{D}^v_{ij} = \mu \frac{\partial^2 \overline{u'_i u'_j}}{\partial x_k \partial x_k} \tag{2.70}$$

2.5 Large Eddy Simulation

2.5.1 Introduction

Large Eddy Simulation (LES) is based on the approximation that it is important to capture the large turbulent eddies in the flow, and that the smaller eddies that only have a small effect on the bulk flow need only be modelled as a sink for the energy. Essentially this is done using a low pass filter and the higher frequencies are then modelled by the sub-grid model. This means that the grid required need only be fine enough to capture the larger eddies so the computing requirements are much less than for DNS both spatially and temporally, allowing the simulation of higher Reynolds numbers. The modelling of the smaller eddies is done by the subgrid scale model and this is essentially dissipative. The subgrid model is a large speedup over DNS where much of the CPU time is spent modelling the dissipative scales [84]. In the same way that the small eddies have a negligible effect on the large ones, the subgrid model is expected to only have a small effect on the large eddies in the model.

It can be seen that the higher the frequency of turbulence that can be captured, the closer the LES moves toward DNS and the relative importance of the choice of the sub-grid model is reduced. However, for practical computations this is not possible and so the cut-off point is placed so that it has to model larger and larger eddies., This then means that the choice of subgrid model and its ability to represent the larger scales is more important. Ferziger, [13] suggests that if 80% of the energy is resolved then the model can be called LES, if less than that then the model is a very Large Eddy Simulation (vLES). In practise however, few authors estimate the energy resolved and so this distinction is difficult to make.

The sub-grid scale model is intended to represent the dissipation of the turbulence energy at a scale smaller than that of the filter. It could be ignored and the effects of numerical diffusion used if desired as the dissipation will still occur. Derivation of flow equations starting with the Navier-Stokes equation, the continuity equation and the equation for energy, the flows are filtered so that the simulation will not model the smaller eddies where the energy is dissipated but will only model the larger eddies which are of such a scale as to affect the mean flow. Pioneering work was undertaken by Smagorinsky [102] in the field of atmospheric modelling. Notable other early investigators include Schumann [98] and Leonard [58] who overcame many of the initial problems. Schumann's work is of particular interest as it is of a finite difference simulation

In a similar way to the Reynolds averaging approach the equations are filtered to give the resolved and the unresolved scales. In a finite volume setting the size of the filter is a multiple of the grid size with the simplest filter being $h = \Delta$. This dictates the size of eddy that can be captured and so defines the large eddy in the LES.

The filtering is applied to the Navier-Stokes equations and these are then used in the simulation. However, there is an unresolved term, the residual stress tensor, and this has to be modelled in order to close the momentum equation.

The decomposition is

$$\phi = \hat{\phi} + \tilde{\phi} \tag{2.71}$$

Where ϕ is a variable in space and time, $\hat{\phi}$ is the filtered or resolved variable and $\tilde{\phi}$ is the unresolved or subgrid-scale part of ϕ. An important difference between $\tilde{\phi}$ and ϕ' in RANS modelling is that the value of $\bar{\tilde{\phi}}$ may not be zero, whereas $\bar{\phi}' = 0$ by definition of the RANS problem.

The filtered equation then becomes [93])

$$\frac{\partial \rho \hat{u}_i}{\partial t} + \frac{\partial \rho \widehat{u_i u_j}}{\partial x_j} + \frac{\partial \rho \widehat{\tilde{u}_i \tilde{u}_j}}{\partial x_j} = -\frac{\partial \hat{p}}{\partial x_i} + \mu \left(\frac{\partial^2 \hat{u}_i}{\partial x_j \partial x_j} \right) \tag{2.72}$$

Which introduces the problem of how to resolve $\widehat{u_i u_j}$. The initial modelling by Leonard [58] and Clark *et al* [13] has been modified to give a galilean invariant solution[106, 26].

The stress to be modelled consists of the filtered product of velocity, minus the product of the filtered velocity,

$$\tau_{ij} \equiv \rho\{\widehat{u_i u_j} - \hat{u}_i \hat{u}_j\} \tag{2.73}$$

which is evaluated by

$$\tau_{ij} = \rho\{\mathcal{L}_{ij} + \mathcal{C}_{ij} + \mathcal{R}_{ij}\} \tag{2.74}$$

where,

Table 2.1: Generic types of sub-grid scale models

Constant viscosity model	$\mu_t = \rho c_0$
Strain model	$\mu_t = \rho c_1 \Delta^2 \hat{S}$
Vorticity model	$\mu_t = \rho c_2 \Delta^2 \hat{\omega}$
Unresolved energy model	$\mu_t = \rho c_3 \Delta E_s^{1/2}$

$$\begin{aligned}\mathcal{L}_{ij} &= \widehat{\hat{u}_i \hat{u}_j} - \hat{\hat{u}}_i \hat{\hat{u}}_j \\ \mathcal{C}_{ij} &= \widehat{\hat{u}_i \tilde{u}_j} + \widehat{\tilde{u}_i \hat{u}_j} - \hat{\hat{u}}_i \hat{\tilde{u}}_j - \hat{\hat{u}}_j \hat{\tilde{u}}_i \\ \mathcal{R}_{ij} &= \widehat{\tilde{u}_i \tilde{u}_j} - \hat{\tilde{u}}_j \hat{\tilde{u}}_j \end{aligned} \tag{2.75}$$

Generally the largest term to arise from the averaging is the Reynolds stress $\mathcal{R}_{ij}$, and it is this term that is representing the interaction between scales at less than the filter size and needs to be modelled. The cross term $\mathcal{C}_{ij}$ represents the interaction between the resolved and modelled terms and would need be modelled. Leonard [58] applied a Gaussian filter to approximate the values for $\mathcal{L}_{ij}$ and $\mathcal{C}_{ij}$ leaving $\mathcal{R}_{ij}$ as the only term to be modelled. Alternatively, Deardorff [17] and Schumann [98] show that the values of the cross terms become zero if the cells used in the simulation are volume averaged. The Leonard term $\mathcal{L}_{ij}$ of Germano can be calculated explicitly by the LES and so the filtered product of velocities and the momentum equation can be modelled as

$$\widehat{u_i u_j} \equiv \hat{u}_i \hat{u}_j + \frac{\tau_{ij}}{\rho} = \hat{u}_i \hat{u}_j + \tau^r + \frac{2}{3} k_r \delta_{ij} \tag{2.76}$$

where

$$k_r = \frac{1}{2} \tau \tag{2.77}$$

is the residual kinetic energy and can be seen not to be a closed equation. In order for the LES simulation to proceed a model for τ^r has to be developed, and the simplest one of these uses an eddy viscosity concept similar to that of RANS modelling. Where μ_t is the sub-grid scale viscosity. Voke and Collins [116] define four broad families for the description of μ_t and these are given in Table 2.1

For the subgrid models the value of c is a constant, Δ is a length scale that is often the filter size and so for the finite difference method that Schumann used is the cell size. $\hat{S}$ and $\hat{\omega}$ are the resolved strain rate and vorticity tensors. E_s represents the unresolved energy, that is the subgrid turbulence energy $(\frac{1}{2}\widehat{\tilde{u}_k\tilde{u}_k})$. $\hat{S}$ is obtained by contracting the resolved strain rate tensor, thus,

$$\hat{S} = (2S_{ij}S_{ij})^{1/2} \tag{2.78}$$

The constant viscosity model should only be used for homogeneous turbulence and is in effect a DNS as the turbulence is independent of the Reynolds number because the viscosity that suppresses the turbulence is moved up the energy spectrum. The strain and vorticity models are using different large scale functions on which to base the subgrid scale turbulence characteristics and the unresolved energy model needs a transport equation for the energy. The energy model is often used with a production equals dissipation assumption. This means that the subgrid energy that is dissipated can be calculated from the resolved energy.

Earlier researchers [13] used the above models and found them to be of approximately the same accuracy, and collectively much closer to experimental and LES on a much finer grid ($8^3 \rightarrow 64^3$) than those models based on the mean strain rate or the rotation of the flow. The subgrid model was also shown to have a small effect on the resolved scales [60], which is good as it shows that the subgrid-scale terms are modelling the small scale flows and that the cut-off point for the filter has been placed sufficiently down the energy spectrum for the modelled small scales to be having little or no effect on the large scales.

Much of the current LES work is based on the analysis of turbulence or the more accurate simulation using the LES method. For these more fundamental investigations, workers tend to work in wavenumber space using fast fourier transforms to study turbulence by spectral analysis. This research falls into two broad groups. Where the research has been done on two different sized grids to find the data for the simulation (*a priori* work) or where the simulations are tested against experimental data (*a posteriori* work)[83]

However, the problem with the use of models that are similar to the eddy viscosity hypothesis used by Reynolds averaged methods is that these are based in the clear difference in scales between the large and small motions. In LES the imposition of a cutoff in the inertial subrange is artificially cutting the continuum in two and so is unrealistic from the start[53]. Some would say that the use of this hypothesis is questionable and leads to a problem that is not well posed [60] leading to results that will decorrelate from reality

To obtain the accuracy in the LES model higher order schemes are used. There is debate over whether to use central or upwind differencing [99] and the order of the differencing to be used [75]. Sengupta and Nair argue that the use of a central differencing scheme will reduce the maximum wavenumber that can be captured by spectral methods and to prevent this and any aliasing effects, a higher order upwind scheme must be used.

Most LES work is still undertaken on structured grids, this is because the numerical dissipation from an upwind differencing scheme on unstructured grid may enhance robustness but can affect accuracy for LES. [74]. Work is underway to develop unstructured methodologies for LES that will enable it to be used for real engineering problems in complex geometries, but has implications for flow near a wall. One of the first problems to overcome is the development of a robust technique for filtering over a tetrahedral mesh. Pioneering work has been undertaken by Marsden *et al* [66]. Similarly, research into the effect of non-orthogonality in the mesh may be a cause of error. This also has been investigated and the skewness of the grid for a finite volume LES simulation affects the results by less that 1%. [101]. This was achieved using single precision computation, with central differencing in space and third order Runge-Kutta differencing in time.

As with DNS there is a considerable increase in the CPU time required. Indications of this are given in [91]

2.5.2 The Smagorinsky model

This is one of the most commonly used models, and is accredited to Smagorinsky [102]. Smagorinsky was simulating atmospheric flows and looking to model the large flows in the atmosphere. It uses the strain rate to calculate the eddy viscosity,

$$\mu_t = \rho C_s \Delta^2 (2 S_{ij} S_{ij})^{1/2} \tag{2.79}$$

The Smagorinsky constant C_s has a value of 0.17 and is defined so that there is a balance of energy between the large and small scales and that the cutoff filter is applied in the inertial subrange where the energy spectrum obeys the $\kappa^{-5/3}$ law. This is defined as [67],

$$C_f = \frac{\ell_o}{\ell_f} = \frac{1}{\pi}\frac{3C_K}{2}^{-1/4} \tag{2.80}$$

Where C_f is the ratio of length scale of turbulence to filter length, which defines the size of turbulent eddy that can be simulated, given that the filter will effectively remove all scales smaller than the filter length. C_K is the Kolmogorov constant and has a value 1.5. This will ensure that the dissipation in the smaller scales matches the dissipation rate from larger to smaller scales [60]. If the value of C_s is equal to C_f then the filter scale is equal to the mesh scale, given that C_s is defined by

$$C_s = \frac{\ell_o}{h} \tag{2.81}$$

The value of C_s given is then determined by the filter, the grid and the lengthscale of the turbulence. If the value of C_s as defined in Equation 2.81 is varied and with it the grid size, for a fixed filter size, then as C_s is reduced the solution becomes more accurate. However, to reduce C_s beyond 0.2 would imply that $h > \ell_f$ and is therefore not possible to simulate. Conversely, for a fixed numerical mesh, altering the filter scale by reducing C_s gives increased numerical errors [67], suggesting that an optimum value of about 0.2 should be used.

However, the values of the constant is applied over the whole domain and so is used for areas of mean shear and no shear. This is a problem as the optimal value of the constant can be seen to vary with the position in the flow. Bardina (1983) obtained values between 0.09 and 0.2 for the constant. The reason for the change may be explained by the proximity of a wall, which affects the turbulence because as there is turbulence generated there the dissipation has to be reduced. Mason and Callen [68] observe that the value of ≈ 0.2 is adequate for channel flow, but that care must be taken in choosing both the value of the the constant and the grid sizing, as they impact on the lengthscale used near the wall, when matching the model to a law-of-the-wall type function. Deardorff [17] found that the value of C_s had to be halved in the region of high shear, whereas in more homogeneous turbulence the original value could be used.

Moin and Kim used a variation of the Smagorinsky model to successfully model channel flow and achieve good correlation with experimental results by using $C_s = 0.065$. [75] The model used $16 \times 16 \times 65$ nodes in the streamwise, spanwise and wall normal direction for a Reynolds number of 13 800 and had minimum $y^+ < 2$

The Smagorinsky model, however, is overly dissipative, even for the larger eddies. This dissipation of low wavenumbers is caused by the constant C_s being fixed so that the inertial subrange is correct and this extra eddy viscosity added by the mean shear rate can make the representation of transition difficult. This excessive dissipation is caused by the correlation of the Smagorinsky term and the strain rate. [63]

Despite its shortcomings the Smagorinsky model still functions well [53, 61], one of the reasons for this could be that as long as the range over which the subgrid is spread is small, it is relatively safe to assume that the subgrid turbulence is isotropic.

2.5.3 Other LES models

2.5.3.1 No subgrid model

Lesieur and Métais [60] state higher order differencing for spectral LES introduces numerical dissipation of the energy as a function of the wavenumber. It may be regarded as *pseudo-direct simulation* [99] and is primarily for the spectral study of turbulence. Much of this work has given unsatisfactory results [60] meaning that the results have to be checked very carefully. Boris *et al* [7] also used no explicit filtering, the work is often justified on the grounds that the scheme is accurate as long as the bulk lengthscale is large compared to the gridsize, as this ensures well resolved flow. The main advantage of this method is that the turbulent motion of the flow is represented in this case. The energy removal is only where necessary. However, the details of this removal are vague and the issues surrounding the numerics and the physics of the problem become intertwined in such a way that it is not possible to obtain a 'grid independent solution', or estimate what is happening at the sub-grid scale because the physics is confounded with the numerics..

2.5.3.2 Dynamic models

Following on from the application of the same constants for the whole domain, which can be seen to be flawed for a non-trivial domain, the constant could be modified to account for the local flow field. This technique has been pioneered by Germano [27, 26, 62] and is a more promising future for LES also allowing modelling of near wall flows more accurately.

A test filter is applied to the filtered flowfield such that the size of this filter is twice the original filter size. We can then divide the turbulence scales into,

- motions larger than $\bar{\hat{\Delta}}$
- motions between $\hat{\Delta}$ and $\bar{\hat{\Delta}}$
- and, motions smaller than $\hat{\Delta}$

where $\hat{\Delta}$ is the LES filter and $\bar{\Delta}$ the test filter, and as before the motions between $\hat{\Delta}$ and $\bar{\hat{\Delta}}$ are the motions just larger than the LES filter and those just smaller than the test filter. Typically $\bar{\Delta} = 2\hat{\Delta}$

Germano *et al* noted the identity [28]

$$\mathcal{L}_{ij} = \mathrm{T}_{ij} - \bar{\tau}_{ij} \tag{2.82}$$

to relate the subtest and subgrid scale stresses to the resolved Leonard stresses.

$$\begin{aligned} \mathrm{T}_{ij} - \frac{1}{3}\delta_{ij}\mathrm{T}_{kk} &= 2C_s\bar{\Delta}^2\bar{\hat{S}}\bar{\hat{S}}_{ij} \\ \tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} &= 2C_s\hat{\Delta}^2\hat{S}\hat{S}_{ij} \end{aligned} \tag{2.83}$$

The value of C_s is optimised by the use of a least squares fit of $\mathcal{L}_{ij}$ and $\mathcal{M}_{ij}$

$$C_s = -\frac{1}{2}\frac{\langle\mathcal{L}_{ij}\mathcal{M}_{ij}\rangle}{\langle\mathcal{M}_{ij}\mathcal{M}_{ij}\rangle} \tag{2.84}$$

where (as demonstrated by Lilly to prevent the value becoming negative [62])

$$\mathcal{M}_{ij} = \bar{\Delta}^2\bar{\hat{S}}\bar{\hat{S}}_{ij} - \hat{\Delta}^2\overline{\hat{S}\hat{S}_{ij}} \tag{2.85}$$

The brackets $\langle \ldots \rangle$ denote averaging, usually over a similar plane, for example, parallel to a wall in channel flows. This is used to give stability to the solution as the value of C may be negative denoting backscatter of energy. Small numbers of negative cells are acceptable though it is usual practice to set the minimum value of C_s to be 0 [3]. This prevents backscatter and avoids numerical instabilities. Ghosal and Rogers [31] successfully model a plane wake using two dynamic models. Their dynamic model only averaged in the spanwise direction, and their dynamic localised model (DLM), averaged and then applied the $C_s \geq 0$ condition. The DLM model took over half as much again in CPU time for comparable predictions of the turbulence statistics when compared to filtered DNS results. However, the DLM may be applied to more general flows without the risk of numerical instability.

Averaging in a statistically homogeneous plane however, limits the LES model to a relatively simple geometry. One way to overcome this was proposed by Menevea *et al* [70] who proposed that weighted averaging along the fluid particle path be used to provide the value of the Smagorinsky constant.

The modelled scalars are taken from the dynamic model from Equation (2.84)

$$\begin{aligned} \mathcal{J}_{LM} &= \mathcal{L}_{ij}\mathcal{M}_{ij} \\ \mathcal{J}_{MM} &= \mathcal{M}_{ij}\mathcal{M}_{ij} \end{aligned} \tag{2.86}$$

The model is shown to work well near walls, in decaying turbulence and in transitional flow, and in addition will not become negative. The model is shown only to have a weak dependence on the timescale that the average is weighted over, and the numerical diffusion from the first-order interpolation of the scalars aids to smooth the constant.

Theoretically, the formulation for the value of C_s should allow the correct modelling of wall flows as the transitional and laminar flows should be predicted correctly, [67, 84].

It can be seen therefore, that the modification of the constant C dependent on the local flowfield allows for a more accurate simulation of the flow. This however indicates that the modelling strategy may not be entirely correct, for if it were so the monitoring of the strain rate (in the strain rate model) would represent what is happening to the flow, and a second equation to modify the result of the strain rate by the use of another property of the flow would not be required. In his review, Jiménez states that 'the dynamic model owes more to its robustness that to its correct prediction of the physics' [43]. However, for the subgrid models he tested, none correctly predicted the subgrid physics adequately but the large scale predictions compared well with experimental and DNS results.

The meeting of the UK LES/DNS consortia in March 2001 suggested the use of the dynamic model as a result of research that they have undertaken. In the case of the flame simulation and modelling of flow through an aero engine the results from the dynamic models were better than those from the static models for these fluctuating cases.

Due to the failure of the Smagorinsky model in regions of high shear, other approaches have been proposed. Germano [27] proposes a galilean invariant dynamic filter by means of the application of a second filter. The two filters are then used to define a local value for the Smagorinsky constant in time and space. This model has been validated and shown to recognize when the flow is laminar, and also performs well in the transitional and turbulent regime when compared to DNS results. This method was unstable and has been modified [26, 28, 29] to increase the stability and allow a negative eddy viscosity which represented backscatter to the correct pattern, though could itself cause numerical instabilities. The model behaves correctly near a wall giving zero subgrid scale stress and correct behaviour near a wall. However the constant is often averaged in the direction of flow making this model unsuitable for all but the simplest of domains.

2.5.3.3 Kraichnan's spectral eddy viscosity

Kraichnan proposed a spectral eddy viscosity which although developed for isotropic turbulence works well for non-isotropic and inhomogeneous large scales [60]. The prediction of backscatter is adequate. However, spectral eddy viscosity is difficult to employ when working in physical space [60], though the formulation of the eddy viscosity is close to that for the Smagorinsky model and the RNG formulation.

2.5.3.4 Scale similarity model

Bardina *et al*[5] proposed the scale similarity model based on the assumption that the most important correlations between the resolved and unresolved scales are with the smallest of the former and the largest of the latter [84, 60]. Rather than the large scale strain rate used in the Smagorinsky model, the authors used a second filtering to evaluate these large subgrid scales, which they refer to as filtered small scale motions. The modelled residual stress is,

$$\tau_{ij} = (\mathcal{L}_{ij} - \frac{1}{3}\delta_{ij}\mathcal{L}_{kk}) - 2c_s\hat{\Delta}^2\hat{S}S_{ij} \tag{2.87}$$

Giving a model for the eddy viscosity that does not explicitly use the filter width Δ, thus,

$$\nu_t = 0.41(\hat{u}_k\hat{u}_k - \hat{\hat{u}}_k\hat{\hat{u}}_k)(2\hat{S}_{ij}\hat{S}_{ij})^{\frac{1}{2}} \tag{2.88}$$

It is the inclusion of the Leonard term that differentiates it from the dynamic model.

However this model is found not to dissipate energy,[5] and has led to modifications to spatially average the strain or introduce a correction factor. Liu *et al* [63] used physical data from a turbulent round jet to assess and tune the scale similarity model bringing in dynamic elements. The relationship between the real sub-grid scale stress τ_{ij} and that modelled by the resolved scales T_{ij} is developed and the assumption that they are correlated is proved false. However, reasons for scale similarity are postulated and are given. Firstly, if an eddy is influenced by the next largest size, and that in turn is influenced by the next largest eddy, then there may be some correlation between the largest and smallest of the three sizes. Secondly, if a structure is decomposed into eddies of a fixed size, then there will be correlation as in actual fact it is one structure that is being viewed as a cascade of eddy sizes. A correlation between τ_{ij} and $\mathcal{L}_{ij} = \widehat{\hat{u}_i\hat{u}_j} - \hat{u}_i\hat{u}_j$ is demonstrated, leading to a dynamic stress-similarity model which uses a weighting function (which is a correlation of the stress and strain rates to modify the modelled sub-grid scale stress).

2.5.3.5 One-equation models

As an analog to the one-equation models used in RANS turbulence modelling, the one-equation model [16, 18, 48], models the turbulent transport equation with a dissipative model (usually the Smagorinsky model). These authors are looking towards the modelling of more of the flow with the sub-grid scale model and so have modified the dissipative equation so that the effects of the mean shear are lessened in the calculation of the eddy viscosity.

2.5.4 LES and the energy spectrum

The cutoff point between the modelled and simulated turbulence should lie in the inertial subrange. This will then ensure a smooth transition from $\hat{\phi}$ to $\tilde{\phi}$ from the approximations inherent in this range as all of the energy that is not modelled can be assumed to be dissipated. This is naturally an obstacle to modelling as the decision where to put the cutoff point decides the computer power required, the accuracy of the model and the level of approximation that is required to allow the dissipation of the sub-grid scale energy. It is important to remember though that in physical turbulence there is no distinct change from large to small scale. In the physical world there is a smooth transition. This is a point where error may be introduced and so the placement of the cutoff in the inertial subrange will attempt to mitigate these errors because the transfer of energy at that point is generally in the direction of isotropic dissipation but the effects of viscosity are minimal.

Models are being developed that will place the cutoff higher up the inertial subrange [18]. This forces the subgrid scale model to be more descriptive of the smaller eddies, but if successful will reduce the CPU time required for the simulation as the cells can be larger because there is less resolution of the flow required. The problem is that the placement of the cut-off higher up the energy spectrum means that the subgrid scale model has to be more complicated to represent more features of the flow than just the dissipation by viscosity. A balance, therefore has to be struck in the placement of the large-scale/small-scale division between the simple subgrid scale model requiring smaller cells and the more complex subgrid model with the larger cells.

2.5.5 Near wall approximations

Near a wall the flow undergoes high shear and so there is a generation of turbulent energy. In order to capture the features of the flow it is important to be able to resolve smaller scales at the wall and to better predict the shearing in the direction parallel to the wall. The shear stress acts to create small turbulent bursts but the wall acts to restrict the movement of this turbulence. There is therefore coherent movement of the turbulent structures near the wall and these cannot be represented by a model that assumes that any eddy smaller that the filter width is entirely dissipative. Development of a near wall model requires a much finer near wall grid which is computationally expensive, a model allowing generation energy at sub-grid scales, or both [53, 67]. The dynamic subgrid-scale model, is much better than the Smagorinsky model in near wall regions, due to its less dissipative character and successful reduction to laminar flow [3].

However, the use of a non-uniform grid introduces error into the simulation as the filter is changing across the domain. Ghosal and Moin [30] tackle this by introducing a term that allows the commutation error to be calculated so that the flow may be calculated with as much accuracy as required. However the use of nonuniform grids without the reduction in accuracy may be possible by utilising an explicit filter and ensuring that the grid spacing is smaller than the filter width, which though reducing the accuracy allows the filter to be specified independently of the grid. This should increase the accuracy of the dynamic model by controlling the errors, and also ensure that the filter is known exactly [114].

Another approach to wall flows is to refine the grid in the region of the wall and then to use body forces to act as though there is a wall whereas there is not. [74] This approach relies on linear interpolation between the (refined) cell centres to replicate the object being modelled and has worked successfully for stirred tank reactors and the modelling of eddies behind lorries.

In the study of liquid metal flows, as used to cool nuclear reactors, wall modelling has been used successfully in comparison with experimental data [34]. The wall sublayer extended well into the domain ensuring that it was adequately modelled and the wall functions employed for the sub-grid scales moved smoothly from laminar to turbulent simulation being based upon universal velocity and temperature profiles.

As an alternative to resolving the flow up to the wall, there are a number of near wall approximations that are available for Large Eddy Simulations. Essentially developed to reduce the CPU overhead with the required wall resolution of a Large Eddy Simulation, they are in effect 'Hybrid' models. These models fall into 2 broad categories [80]. Both accept that the resolution of the grid is not fine enough to capture the small eddies near the wall, but rely, to varying degrees, on the fact that these small eddies are isotropic and can to some extent be analysed in a statistical manner.

The first method is to use an equilibrium between the flow at the wall and the flow further from the wall. This is a fix to ensure the correct behaviour of the flow near the wall. It is only available for simple geometries and is, in effect, a law of the wall for LES models.

The second category is to attempt to model the flow near the wall. There are two popular methods to try to do this. The first is the so-called Two-Layer model (TLM) which overlays a finer grid between the first grid point and the wall in order to solve the 2-dimensional boundary layer equation in this space [4, 82]. The method is shown to work well with simple geometries but not for flow in a rotating pipe. This method does not perform well when coupled to the Smagorinsky model [82], but when used with the dynamic SGS model gives good representation of the log-layer.

The second method used to model the flow near the wall is Detatched Eddy Simulation (DES) pioneered by Spalart *et al* 1997 [105], initially for detached flow [80]. A second grid is not defined as in the Two-layer model, but rather a RANS turbulence model is used near the wall until the lengthscale from the turbulence model approaches the distance from the wall, and at that point it is fixed to the filter width of the LES. However, because there is only one grid the velocity field is smooth and does not have to be interpolated at an interface. This then requires a division of the intended work for the DES grid [103]. The model has been modified by the inclusion of a backscatter model to reduce the error in the prediction of the log-layer just inside the outer (LES) region, which may have been caused by incorrect matching of timescales from the RANS and LES models[81]. There is limited use for DES for internal flows however, as the spatial resolution is close to that of a LES. Its advantage is for external flows where the reduction in the number of required grid points is more prominent [97].

2.6 Hybrid modelling

Given the accuracy but computational overhead of Large Eddy Simulation, and the adequateness and lower computational overhead of the Reynolds averaged approach, the idea of bringing the two approaches together will offer the advantages of both. The Reynolds average technique will be used where it is appropriate, this being where the turbulence is isotropic and where the averaged approach holds. The Large eddy simulation will be used where the turbulence is anisotropic such that the application of averages greatly reduces the accuracy.

One of the large challenges facing the interlinking of the two forms of turbulence modelling is that there is a difference in turbulent viscosity between the LES and RANS procedures by an order of magnitude or more. Reasons for this are that the turbulent viscosity for the LES simulation is smaller to allow the fluctuations to be modelled, this is as a result of the derivation of the governing equations. If μ_t were as large as for the RANS modelling procedure only the mean flow would result, though on a much finer grid. To take this to its conclusion, the DNS simulation does no have a turbulent viscosity, as there are no modelling assumptions made.

Speziale proposed a workable very Large Eddy Simulation concept [107]. The drive of the work was to scale the turbulent viscosity with the grid size and so allow a sliding scale of modelling accuracy from the scaled μ_t. The results presented were convincing but there have been criticisms from those who see this as a scaled $k-\varepsilon$ model as opposed to true LES/RANS hybrid modelling. It is also required that care be taken with the timestep as that which is required for the fine cells must be used for the coarse cells also, or the accuracy of the pseudo-LES will be lost.

Dejoan and Schiestel [18] are developing a methodology similar to the ideas of Speziale in that the properties of the filter change as the filter width is altered. They develop a transport model for the resolved and sub-grid scale turbulence energy and successfully combine this with a dissipative model in such a way that the cutoff can be placed above the inertial subrange.

More recently a hybrid modelling approach has been put forward to solve problems at the wall [35]. Essentially for a pipe the assumption is made that the bulk of the flow can be resolved adequately using a 1-dimensional $k-\varepsilon$ model, but increased wall resolution is obtained by performing a LES in that region. This is the opposite of the wall modelling approximations in section 2.5.5 and used to extract data about the wall flows. However, both the wall models mentioned earlier and Hamba's work can be considered as hybrid models.

A multiple time stepping approach has been used successfully to model the flow around a sphere by Teixeira and Awruch [110]. Here the domain is simple and contains only ≈11 000 nodes, but the cpu time is reduced by over an order of magnitude in the calculation of the final solution by the modelling of some of the nodes on integer multiple timesteps of each other. This reduces the CPU overhead but is insensitive to physical phenomena. To overcome this a monitoring function has been introduced and the speedup is still eightfold. This technique lends itself to parallel computing as the processors could run at the different timesteps. This would allow many nodes to simulate the fluctuating jet, and a larger number to calculate the rest of the domain which could be much larger, a technique that could be invaluable in the accurate simulation of chemical reactions.

Schlüter *et al* [96] have recently published the results of a method of combining both LES and RANS modelling approaches with a view to a complete simulation of an aero-engine using RANS modelling for the turbines and LES for the combustion. They apply a user defined body force to the LES flow and then can alter the velocity profile to ‘blend’ the results to the inlet flow for the RANS modelling. This is done in an area where both of the models are applied at the same time. The methodology is only applied to the flow from the LES into the RANS, though it is noted that this does effect the flow in the LES region. In practice a Reynolds averaged simulation would be used to provide these boundary conditions for the integrated model.

However these approaches have not harnessed the possibilities of using a parallel computer to explicitly divide the workload between the methodologies.

2.7 Numerics and discretisation

In order to obtain a higher level of accuracy, a knowledge of how the numerical solution is reached is important. The method used in the commercial code CFX is that of finite volume approach. This may be simplified to the finite difference approach if considered in two dimensions, and an outline of the finite difference method is given below.

Additionally, for the program to be successfully applied to a parallel computer, a knowledge of how that computer will interact with the code and solve the problem is also required. An overview of the method used to parallelise the CFX4.4 CFD package has been given.

For a given function $y = f(x)$ a prediction of the value of point y_2 corresponding to point x_2 is given by the Taylor series expansion.

$$y_1 = y_0 + \Delta x \frac{\partial y}{\partial x} + \frac{1}{2}\Delta x^2 \frac{\partial^2 y}{\partial x^2} + \frac{1}{6}\Delta x^3 \frac{\partial^3 y}{\partial x^3} + \ldots + \textit{higher order terms} \tag{2.89}$$

By manipulation of the Taylor series expansion an expression for the prediction of the gradient using the two adjacent values of y may be obtained. Using the value of $y_1 = f(x_1)$ and $y_0 = f(x_0)$ and $x_1 > x_0$, a prediction for the gradient is,

$$\frac{\partial y}{\partial x} = \frac{y_1 - y_0}{\Delta x} + \mathcal{O}\left(\Delta x\right) \tag{2.90}$$

This can be used to predict the flow in the CFD domain. It is the equivalent to setting the velocity at a cell face to be that at the cell centre and then if referred to as the 'upwind differencing scheme'. Knowing the distance Δx and two of the other values the third may be calculated. It is only accurate up to the second term in the expansion and so is referred to as being 'first-order accurate'

The gradient may also be obtained by combining two different predictions

$$\begin{aligned} y(x + \Delta x) &= y_0 + \Delta x \frac{\partial y}{\partial x} + \ldots & (2.91)\\ y(x - \Delta x) &= y_0 - \Delta x \frac{\partial y}{\partial x} - \ldots & (2.92)\\ & & (2.93) \end{aligned}$$

This gives rise to the 'central differencing scheme' and is 'second order accurate' as it encompasses two different values for y and is central about the point x_1 where $y_1 = f(x_1)$

$$\frac{y_2 - y_0}{2\Delta x} = \frac{\partial y}{\partial x} + \mathcal{O}\left(\Delta x^2\right) \tag{2.94}$$

Thirdly, the 'higher upwind scheme'

$$y = \frac{3}{2}y(x - \Delta x) - \frac{1}{2}y(x - 3\Delta x) + \mathcal{O}\Delta x^2 \tag{2.95}$$

The differencing scheme is then used to generate the solution in an iterative fashion on the computational mesh that the domain has been divided into. The mesh is progressively divided up into smaller and smaller cells until the solution does not change. This is said to be a 'mesh independent solution' and occurs when the approximations introduced by the discretisation become less than the numerical errors from the differencing scheme, the models used and the computer error.

The mesh spacing defines the accuracy of the model, especially in the case of a LES. This is because the filter size is a function of the grid size, typically $\Delta = h$ where h is the grid spacing and Δ the filter size. This does mean that it is difficult discern whether the grid has an influence on the solution. It is also true that the Large Eddy Simulation is a fully transient model that is trying to represent the chaotic nature of the turbulent flow with as few approximations as possible. This makes the definition of a 'grid independent solution' very difficult.

When the Reynolds number is large enough and there is a separation of scales in the energy cascade the difference in size between the large and small eddies will be in excess of 10 000 times. The resolution of the large eddies will require the positioning of say 10 filters across the eddy size or say 20 grid points across the eddy. This would lead to a grid density of 8000 grid points per large eddy. However, usually it is more common to try to capture more of the energy in the flow than just the largest eddies and so maybe this would require $100 \times 100 \times 100$ grid points which comes to 1 000 000 cells. This is a thousand-fold increase in the number of grid points and to obtain a solution a speed-up in computer resources of the highest magnitude is required.

When a finer grid is used there is a risk that the solution will propagate across the cell in less than one timestep if the step is too large. In order for this to be prevented the 'CFL' condition or Courant number must be adhered to,

$$\frac{u\Delta t}{\Delta x} \leq 1 \tag{2.96}$$

For the DNS models that resolve the flow down to the point of dissipation of the energy the computation time is even greater. The reason for this is that the grid required to solve for the Kolmogorov lengthscale is very fine. As the Reynolds number increases the difference in size between the large and small eddies increases, which for internal flow means that the Kolmogorov scale becomes smaller requiring more cells for the simulation. This has knock-on consequences to the time to reach a solution and the size of computer required, especially as it is accepted that the Courant number should be [12].

$$\frac{u\Delta t}{\Delta x} = \frac{1}{20} \tag{2.97}$$

An estimate of the number of points required for an DNS simulation is given by [59, 93],

$$N \sim Re^{9/4} \tag{2.98}$$

The increase in size and complexity of problems leads to the development of faster computers, conversely the development of computers allows us to solve more numerically intense problems. The physical limitations of computing today mean that the larger problems and completed more quickly if broken down into smaller pieces and solved. Parallel computing provides the means to do this, and because the pieces are all solved at the same time the answer is reached faster. This speedup is defined by

$$S(n,P) = \frac{T(n,1)}{T(n,P)} \tag{2.99}$$

Where P is the number of processors allocated for the job of size n. $T(n,1)$ and $T(n,P)$ are the times to complete the job on one and P processors respectively

However, there is a penalty in spreading the work over many processors in that they have to communicate with one another to ensure that they are working at the same rate, this synchronization causes idle time on some processors and the network traffic to increase. There is also an overhead attached to the communications between the processors in the speed of the network and its capacity.

There is therefore an efficiency associated with a given number of processors, and this is defined as,

$$E(n,P) = \frac{S(n,P)}{P} = \frac{T(n,1)}{P\,T(n,P)} \tag{2.100}$$

Assuming that the a machine with P processors can go no more than P times faster than a single processor machine it can be seen that $S(n,P) \leq P$ and $E(n,P) \leq 100$. Additionally it should be noted that the problem should be scaled with the number of processors. If the problem size remains the same, then the load per processor will be reduced and more time will be spent in communication than in computation, and the efficiency will drop

The deviations from this ideal can be summed up in four main points [72]

- Amdahl's Law. The speedup achievable on a parallel computer is limited by the existence of a small fraction of inherently serial code that cannot be parallelised. This fraction of code α affects the speedup, though as the proportion of the code that can be parallelised is increased the value of α will tent towards zero.

$$S(n, P) \leq \frac{1}{\alpha + \frac{(1-\alpha)}{P}} \leq \frac{1}{\alpha} \tag{2.101}$$

- Use of less than optimal algorithms when the best algorithm will not parallelise or to reduce the communication overhead
- Parallelisation often increases the overheads associated with procedure calls, which may be called a 'relative' overhead, and also the 'pure' overhead of communication and synchronisation
- Load imbalance will force some of the processors to work for longer than others and the resulting idle time will show up as a reduction in the speedup.

2.7.1 Bulk synchronous parallel

A brief overview of the BSP model has been included as it is the method used by CFX to parallelise CFX4.4 for the ccNUMA Origin2000 machine (*mimir*) at Shrivenham Campus.

Developed by a collaborative group including Oxford and Harvard Universities, the BSP model [10, 40] for parallel computing is one of the earliest, though still receives much research as as it is an easily portable paradigm. [25, 113]. It allows fine grained parallelisation where the program is parallelised at the loop level and so not as efficient as parallelising the whole program, because the step size can be set by the user to optimise the parallelisation they have undertaken.

Originally conceived as a way of parallelising programs on a distributed memory system, it has been implemented successfully on both shared and distributed memory computers [41] who show that with some care in the design of the program the speedup is very good.

The model works in the following way. The program is divided into 'supersteps' that consist of three ordered phases.

1. Computation locally to that node
2. Communications amongst the processors, which are only done at this time.
3. Barrier synchronisation, which ends the superstep and makes the information available to the processor ready for the next superstep.

The BSP model categorizes the parallel machine using measurable characteristics that will then allow the machine to be defined and the computation and communication time predicted. The ratio of computation time to communication time is defined g, the time between successive synchronisations L the maximum number of messages passed h and the number of processors P. Ideally $g \geq 1$ or the computer is spending more time communicating that calculating. These allow the estimation of the time for one superstep as being $\max w_i + \max h_i g + l$ where w_i is the computation time of process i and l the overhead associated with the synchronisation.

Work has been done to calculate the theoretical behaviour of the BSP model on different computer including its use on a shared and distributed memory parallel machines [25, 24]. The experimentation has proved the predictions to be acceptable [51, 25] and also illustrated its portability and the need for only a recompilation of the serial code on a new computer as opposed to a re-write of the BSP calls.

2.8 Flow over a cylinder at $\mathrm{Re_D} = 3\,900$

In 1998 the Advisory Group for aerospace research and development (AGARD) proposed the adoption of flow over a cylinder as a test case for LES development. The simulation requires resolution of a moving point of detachment for the separating flow, and both physical and numerical instabilities which are seen as, transition and large scale vortical structures. Computational work has been undertaken for Reynolds numbers up to 140 000 where the Reynolds number $\mathrm{Re_D}$ is based upon the cylinder diameter, the mean upstream velocity and the physical properties of the fluid upstream from the cylinder.

Up to $\mathrm{Re_D} = 40$ the flow is steady and laminar with a pair of ‘stationary’ vortices behind the cylinder. The region $40 \leq \mathrm{Re_D} \leq 190$ is that of laminar vortex shedding. From $190 \leq \mathrm{Re_D} \leq 260$ has been labelled the ‘mode A’ instability by Williamson [119] and is where the domain and spanwise scales have a wavelength of approximately 4 diameters. At $\mathrm{Re_D} = 260$ the ‘mode

B' instability is entered where spanwise differences are seen at a spacing of approximately 1 diameter. The next change happens at between $30 < \mathrm{Re_D} < 3\,000$, though typically at $\mathrm{Re_D} = 1200$ when the shear layer becomes unstable and separates [85]. Full transition to turbulence occurs between $2 \times 10^5 \leq \mathrm{Re_D} \leq 3.5 \times 10^6$ and the boundary layer becomes turbulent before detachment at Reynolds numbers exceeding 3.5×10^6

Ong and Wallace [77] used hot-wire anemometry to provide accurate experimental results for $\mathrm{Re_D} = 3\,900$ with an aim for validating LES results. These have been compared with the LES results of Kravchenko and Moin [49] for the same Reynolds number and have shown very good agreement. Kravchenko and Moin used three grids of 0.5M 1M and 2M cells and a B-spline method to interpret the flow between the cells. In addition a central differencing scheme was used as higher-order upwind schemes were shown to posses more numerical dissipation. Accurate prediction of the Strouhal number is obtained and the power spectrum matches the experimental results of both Ong and Wallace and also Lourenco and Shih [64]. Spanwise lengthscales are predicted correctly as in the earlier work by Kalro and Tezduyar [46] and Lei *et al* [57]. For comparison a simulation is done without a subgrid scale model, in this case the mean velocities compare well but the energy spectrum shows much less dissipation than when the Smagorinsky model is used. Franke and Frank [23] used this case with a compressible LES code and obtained good results when compared to experiments.

Anderson and Reider [1] investigate closely the numerics of the model. They suggest that the grid resolution in the normal and tangential direction on the cylinder is nearly equal with the mesh size being of the order of $1/\mathrm{Re}_D$. They also demonstrate the importance for careful development of the model to reduce startup errors and used in conjunction with their fourth order differencing scheme, a CPU time of only 20% more than when the hybrid difference scheme was used was realised.

Breuer [8] studied the case for $\mathrm{Re_D} = 140\,000$ using both Smagorinsky and Dynamic subgrid scale models on a parallelised finite volume grid. The concentration of cells around the cylinder gives a slight reduction of accuracy in the wake which is especially noticeable in the $u'u'$ measurements. This is in agrement with the conclusion of Kravchenko and Moin that the grid refinement and influence of the grid on the solution has to be investigated and clarified. The results also agree very well with the LES results of Lübcke *et al* [65] However, the latter estimated that their explicit algebraic stress model with its non-linear stress-strain relationship can give comparable results at only 5% of the computational time.

Chapter 3

Method

3.1 CFX4.4

3.1.1 Overview

The work herein was completed using the commercial computational fluid dynamics (CFD) package CFX4.4, which is a block structured staggered grid segregated solver CFD suite [11]. The choice of package was influenced by the facilities within it to test whether the advantages in accuracy and computational time by the coupling of LES and RANS methodologies could be achieved. When this study was started in 1998 it was the most advanced CFD program that AEA software produced, although now has been superseded by CFX5, which is an unstructured coupled solver.

CFX4 consists of three main parts. A pre-processor where the computational grid is generated. A solver where the solution to the problem is found, and a post-processor where the solution may be viewed. (It should be noted that as with many CFD programs the solver may read in grids from many sources and there are a variety of post processors that can be used. For this study the native pre processor has been utilised and either the native post processor or 'Fieldview' by Intelligent Light has been used)

To create the grid in the pre-processor 'Build', geometric points are created, these are joined by edges, edges joined to give surfaces and these joined to give blocks. The blocks have to be topologically hexahedral, so that the cells are topologically cubes. There can be a number of blocks in a model, and this allows the use of more blocks where the geometry is more complicated. However, being a structured mesh, the cells each side of the block boundary have to match exactly and so the refinement of cells in one block may dramatically affect the number of cells in the model overall. It is possible for the number of cells each side of the boundary not to match and this is referred to as an unmatched grid.

In the pre-processor, patches may be put on the block boundaries (2D patches) or on a whole block (3D patches). 2D patches are used to mark walls, inlets and outlets, periodic and symmetry planes. In addition to this a user defined 2D patch may be set that can be referenced by the used for a specific task. 3D patches may be used to set solids, porous regions and again user defined regions that can be referenced explicitly.

Having defined the mesh or grid, and put the patches required onto it a geometry file is generated by CFX4 and this can then be used by the solver. The geometry file starts with a header which tells the number of blocks, patches cells and points. Following that there is a summary of the size of the block, given as the number of cells in the i, j and k direction. Next comes the description of the patches in the geometry, first the block faces that are glued together, then the patches that have been placed by the user. Finally there is a section containing the positions of all of the points in the geometry, which finishes with an end of file marker.

In order for the solver to be able to use the geometry that has been created it has to be provided with more information, this information is contained within the command file. The command file contains the options instructions and boundary conditions and given this and a geometry the solver will be able to provide an answer. The command file tells the solver whether the flow is laminar or turbulent, isothermal, compressible. It provides instructions as to whether there are any more patches to be added. The fluid properties are specified along with the differencing scheme, turbulence model and time stepping information for a transient run. The solver and number of iterations can be specified along with the under relaxation. The boundary conditions are then given and then the options for the printing out of the results for the post processor. There are many options for the command file and the documentation should be consulted for more information. [11]

CFX4 is a general purpose code, and so there are some areas that are not covered by the intrinsic functions. To solve for any problem that may lie outside of the realms of the code, CFX allow the addition of ‘user fortran’ subroutines, which are linked to the source code and so produce the results desired by the user. In order to allow the user to do this user will have to write the code to be added. This can be in the form of FORTRAN77, but CFX4 will accept object code from F90.

Table 3.1: Order of computation in CFX4.4

viscosity	μ
velocity in x-direction	u
velocity in y-direction	v
velocity in w-direction	w
pressure	p
turbulent kinetic energy	k
turbulent dissipation	ε
user scalars	SCAL

The solver reads in the information from the geometry, the command file and any added user fortran and then using the options specified will generate a solution to the problem and will output the results in the form of a dump file containing the results, and an output file, which contains details of how the solution was reached, the convergence history, and any error messages. The solver solves each block in turn according to their numerical value. For each block the cells within that block are numbered according to their position including the dummy cells and then the solution obtained with the dummy cells acting as boundary conditions for that block.

The post-processor enables the visualisation of the results and the printing of these results. Functions may be defined in the post-processor to express clearly the values that the user is interested in, and also the plotting of results along a line, plane or volume, that can then be compared to the same plot from another computational model.

It is also possible to restart from a previous run and this allows for the monitoring of progress and the breaking of long runs into manageable sized simulations.

3.1.2 Order of computation

The order of computation of the flow variables for a laminar simulation is given in table 3.1. When a LES model is run CFX4.4 is by default executed as a laminar model where the eddy viscosity modifies the laminar viscosity. The reason for this is that a LES model does not need the scalars that are associated with the Reynolds averaging approach and so it is best to run the model as a laminar simulation. If a RANS turbulence model is used for a turbulent simulation then the turbulent viscosity is calculated at the end of the iteration once the turbulence properties of the flow are evaluated

The CFX4 flag that controls whether the simulation is turbulent or laminar is called `LTURB` and is true for a turbulent simulation. It is not possible to have different blocks within the domain holding different values for `LTURB`, it is a global variable. It is possible to change the flag whilst the simulation is in progress though the change will not create or remove the turbulence properties though they will not be recorded after the change, which makes the use of the change inpractical.

3.1.3 CFX4.4 - Parallel

CFX4 works equally well on both serial and parallel computers. On a parallel machine the pre- and post-processors remain unchanged but the solver is altered slightly. At the start of the run there is decomposition step which collects all of the geometry information and writes it to the geometry file. This information may have been in the command file or may have to be extracted from the user fortran. Once this is done the geometry is decomposed into a number of domains corresponding to the number of processors specified, CFX tries to balance the work done by each processor to reduce the idle time of the processors by using the internal program ‘Meshimport’ to give good load balancing and vectorisation. To enable it to recombine the geometry again at the end of the simulation, there are files produced which tell CFX how the domains fit together, most notable the PRL, config and connectivity files, which are stored in the ‘para.control’ directory. Once decomposed, and if necessary the restart dump file decomposed as well, the solver proceeds on each of the processors as before. At the decomposition boundary the values are exchanged at the beginning of each timestep. This is achieved by placing dummy cells around each of the decomposed domains such that the dummy cell contains the values from the active cell that it would overlap were the domains to be fitted together again. The parallel execution of the model is controlled by the Bulk Synchronous Parallel method 2.7.1. The solution is then calculated block by block as before but for each node in parallel. At the end of the simulation the parallel solver recombines the dump files form all of the computational nodes, so that to the used it appears that the simulation has been achieved on a serial machine. A parallel output file contains details of the run and the output file from each domain is placed into a directory accessible to the user.

3.2 Development of the hybrid model

This section leads through the development of the hybrid model based on the ideas that may or may not be possible. For different problems, the potential solutions have been given to show the development of the idea.

3.2.1 Overview of the solver

To demonstrate the approach of using a Hybrid model, the simplest way is to split the domain into two sub-domains. This is not reflecting the number of processors that are to be used but rather the number of modelling approaches. One of these new sub-domains will execute the LES code for the calculation of the flow field and the other will execute a RANS model. The interface between the domains could be defined by the criteria in the program and would therefore be adjustable as the program proceeds, or it could be set by the user. At the end of the simulation the user will be presented with results from both of the solvers and this will be available for post processing as if it were from one solver. The LES solver will operate where there is generation of turbulence and the RANS solver will operate where the process is generally dissipative.

The user will construct the geometry in the first instance so that all of the cells could be used for an LES simulation. This will allow for comparison with an LES simulation of the domain. The CPU time will be compared with both RANS and LES simulations. The interface will then be placed by the user in an area deemed appropriate for it and the simulation run. This placement will put the RANS model where the flow is modelled well by the Reynolds averaged approach, the flow is mainly dissipative and that there is little generation of turbulence. The LES sub-domain will be where there is generation of turbulence and regions of strong swirl or anisotropy. Although quicker to start the simulation with a purely RANS simulation, it would not be possible to then refine the grid to allow a LES to take place on the other side of the interface, due to the restriction of no unmatched grid implementation in the parallel version of CFX4

In a fully automated LES/RANS hybrid model the position of the interface would be placed automatically and the criterion could be based upon the turbulence level, the anisotropy of the domain or the turbulence or timescale. This is the first case for the work towards the use of different sized cells each side of the interface, and then the combination of different sized time steps, or the use of a steady state RANS simulation with the transient LES.

Ideally the division of the work amongst the models, and hence the processors would be automated. This will take place after an initial analysis of the domain to indicate where placement of the LES would be most appropriate. Alternatively, the solver could also be stipulated by the user who defines the two domains. They could be given distinctive names the 'LES' and 'RANS', so that the computer will know where to use which model.

This however is an ideal situation. Initially CFX4 allocates work to processors to balance the load as far as possible so as to get the best possible speedup and most equal load balancing from the Meshimport program. The automatic allocation of cells to processors is possible, but with intervention at the source code level, it is possible for the user to allocate work, but this may not achieve the same level of speedup as for the optimised algorithm. Additionally it is not possible to refine the grid during a solver run with CFX4. A new grid is required to be generated by the user and this then requires interpolation from the results of the previous run which introduces small errors and increases the time required for the simulation.

Initially the model will be created and operated by the user. The user, who will have knowledge of the flow of interest, will then allocate the blocks to the processors for the initialisation in such a way that the same allocation will be possible for the hybrid modelling. The user will then be able to monitor the run and should it become necessary to move blocks from one sub-domain to another this could be done. What is being proposed then is a system whereby the expert user may optimise the code to more accurately represent the turbulence in a region of interest

3.2.2 Position of the Interface

Reynolds averaged turbulence models typically perform badly in areas of high swirl or anisotropy. The anisotropy is estimated in the calculation of the Boussinesq approximation, which approximates the Reynolds stresses by the mean strain rate

$$-\rho\overline{u_i'u_j'} = \tau_{ij} = \mu_t\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\rho k\delta_{ij} \tag{3.1}$$

When $i = j$ this reduces to the formulation for the normal stresses,

$$\tau_{ii} = -2\rho k \tag{3.2}$$

leaving the shear stresses

$$\tau_{ij} = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \text{for } i \; \neq \; j \tag{3.3}$$

which could be used to indicate the anisotropy.

The shear stress tensor could be made dimensionless by using the value of the turbulence in the cell, that could be used an a measure of anisotropy

$$A = \frac{\rho \overline{u_i' u_j'}}{\rho k} \text{ for } i \; \neq \; j \tag{3.4}$$

This scalar will then be calculated and stored for each cell, a marker for the level of turbulence. The results calculated in this way for the three shear stresses can then be compared and the maximum and minimum values found. If the flow is isotropic then the results will be the same if there is a uniform grid. If the grid is not uniform then there will be an effect of the grid on the results as the filter is related to the grid. For a non-uniform grid the filter size is a function of the spacings in the 3 dimensions of the grid, for example [21]

$$\Delta = (\Delta_i \Delta_j \Delta_k)^{\frac{1}{3}} \tag{3.5}$$

or

$$\Delta = (\Delta_i^2 + \Delta_j^2 + \Delta_k^2)^{\frac{1}{2}} \tag{3.6}$$

However, if the flow is anisotropic then the values will be different. Depending on the difference a tolerance can be built into the program so that the LES domain does not try to cover the whole of the domain. The user will be able to set the value of the tolerance such that if the inequality for each combination of i and j (with no summation) will be used to give the ratio of anisotropy in each cell. If the tolerance is exceeded then the code will execute the LES,

$$\frac{max\,|A|}{min\,|A|} \geq Tolerance \tag{3.7}$$

Another criterion for the position of the interface could be the turbulence level in the domain. The initial level of turbulence will be calculated by the standard k-ε model and this will highlight the areas of higher turbulence and a range of values can be drawn for the domain. The application of LES will be to the area of the highest turbulence, where the cutoff level could be defined by the user. Alternatively the application of the LES could be linked to the length scale. This could be used to define a monitoring system such that the interface has to be placed where there is less than a maximum level of turbulence or anisotropy. The use of LES will allow the exact calculation of k and ε and so the levels of these can be monitored and the boundary defined by the turbulence value.

$$k = 0.5\overline{u_i'u_i'} \tag{3.8}$$

$$\varepsilon = 2\nu\overline{S_{ij}S_{ij}} = \nu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_k'}{\partial x_i}} \tag{3.9}$$

where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \tag{3.10}$$

The use of an inequality for the level of turbulence energy across the whole domain will give a region where the turbulence is high enough to warrant the use of the large eddy simulation methodology, and would allow a control mechanism for the positioning of the LES model.

$$\frac{max\,|k|}{min\,|k|} \geq Tolerance \tag{3.11}$$

The interface must be placed so that the LES will capture the slowly fluctuating parts of the flow. As an example, it could be argued that in the centre of a jet the flow is reasonably well modelled by the RANS models and that it is on the edge of the jet that the LES should be applied. Following on from this it is of importance that the LES is used to capture any shear layers or far-field effects as it is these that are interesting from a modelling point of view. This may require the insertion of a subroutine to monitor the flow and check for periodicity. Ideally then this subroutine would be able to ensure that the LES sub-domain covered this part of the flow. This would be achieved by monitoring the averages of the velocities as well as the turbulence level.

A combination of the methods presented could be used, with weightings so that the user may stipulate the type of flow that is to be modelled. At a more simple level the division of turbulence modelling strategies will be governed by only one of the criteria presented.

Using the current version of CFX4, the domain will be divided up by the user based upon the knowledge and expectation of the user and the results of the k-ε model. This will be based on the users knowledge of the flow in anticipation of any regions of high turbulence generation. The divided domain will then be run on a parallel computer so that the initial results are obtained. After the initialisation the LES model will be switched on and the model re-run. As the $k-\varepsilon$ model is hidden in the CFX4 code it is not possible to run only the Smagorinsky model on the LES sub-domain without rewriting the $k-\varepsilon$ model in the user Fortran and running the model as a laminar simulation with the $k-\varepsilon$ and LES models modifying the viscosity. The flag for turbulent flow is either on or off for the whole domain even when divided over nodes on a parallel computer. This means that the $k-\varepsilon$ model will be running which does have certain benefits especially as it allows the values of k and ε to be overwritten and their transport equations internally will ensure that the movement of the turbulence is modelled correctly as it passes across the interface. This would be the case for whichever of the turbulence models were used and currently is a feature of CFX4. The main influence following this will be the imposition of wall functions on the LES solution thereby saving in the finer cells required at the wall by the LES sub-domain.

3.2.3 Initialisation of the LES domain

The LES code user subroutines will reach a turbulent solution more quickly if given an initial flow field from which to start the calculations. This may be from a random number generation or from an initial flowfield. In an attempt to speed up the simulation a preliminary flowfield has been used to give further indication of regions where the LES should be allocated and to allow the validation of the grids in that region. The flowfield does not have to be fully developed but has to have a majority of the flow defined, this enables the flow variables to be read into the LES solver and executed. The initial flowfield is required so that velocity gradients can be calculated and these in turn will be used to calculate the turbulent viscosity for the LES subgrid-scale model.

The k-ε model provides a mean flowfield from which the LES can proceed. This is required to gain an answer from the LES in a shorter time. The use of a turbulence model in the initialisation will provide a measure of the turbulence and so allow for an informed decision as to where to put the boundary. The model use for the turbulence modelling will have to be complete so that the prediction of the turbulence will be able to take place for all geometries without any input from the user. This means that the choice is between the two equation models and the stress models.

As this stage of the hybrid model is to provide a basis for the LES to work from there is little need for the use of a 7-equation stress model when a good approximation can be reached by a 2-equation model. When the Hybrid model is running then the LES is where the most interesting flows are taking place. The prediction of the turbulence in the RANS domain can be left to a 2-equation model as the results will be of less interest than the LES and so a quick and reasonably accurate model can be used. This will minimise the CPU time in the RANS domain and allow the use of more computer time in the modelling of the LES.

From this initial result the choice of solver could be automated so that the allocation of work and the choice of solver is left to the computer. Starting from the initialised system an automated code would be able to redistribute the blocks amongst the processors and then continue with the hybrid model. Grid refinement and coarsening could be undertaken on the fly and the simulation could run to completion, which would be easier on an unstructured grid, though this is not available in CFX4.4 so cannot be used.

It is also possible to perturb the RANS solution so as to introduce instabilities that may grow as the LES simulation progresses. However, it has been shown that the instabilities die away and then the LES turbulence statistics are built up again as the simulation progresses. It is quicker therefore to start the LES simulation directly from the results of the initialisation.

There are many perturbation methods that could be used to help initialise the flowfield, other than using a random number generator. The first considered was to use a multiplier which will not be a random number but rather a function of the length scale. This will then enable any structure captured by the k-ε model to be passed to the LES domain.

$$u = (1 + \ell) \times u \tag{3.12}$$

If this were used then the continuity will have to be monitored as it may transpire that this method to initialise the domain may violate the mass continuity by altering the velocities and hence the mass flow. This could possibly be modified so that if the value of ℓ is greater than the weighted average of all of the ℓ values then the multiplier will be

$$1 + (\ell - \bar{\ell}) \tag{3.13}$$

This will mean that there will not be an overall increase in the velocities in the LES region, as the use of a weighted average will prevent this. Though this is reliant on the assumption that there is a normal distribution of lengthscales around a mean value in the domain.

If there is a fluctuation in the flow, for example vortex shedding, then the initial flowfield could be perturbed using *user fortran* to emphasise the cross-stream velocities in synchronisation with the vortex shedding frequency

For flow in a pipe, where the turbulent velocity profile is flat this is more difficult. As the shear is at the walls of the pipe the most straightforward method would be to ensure adequate resolution of the wall boundary layer and to allow enough diameters for the LES to develop. An attempt to speed this process could be to increase the roughness of the wall, thereby increasing the friction and the shear as has been used successfully in experimental work [54].

This highlights a key point that has not been developed in this thesis, that being the flow into the LES sub-domain. As with the development of the turbulence in the LES sub-domain, the inflow into it has to be allowed to develop. The passing of isotropic k-ε turbulence levels with one lengthscale across an interface, and the transformation of that information into a turbulence structure is complicated and cannot be circumvented if the inflow is to be developed fully. However, if it is assumed that the turbulence is generated in the most part in the region of the LES then the problem is reduced to the transport of the statistics out of the large eddy simulation and into the Reynolds averaged solution.

It is an important conceptual point to note that when the hybrid model is started the velocities in the LES region change from being averaged velocities where the turbulence model is used to predict the fluctuations, to filtered velocities, which are much closer to the instantaneous velocities at that moment in time. Time is therefore required for the smooth flowfield that is the RANS solution to change into the more chaotic, instantaneous, LES prediction, which is not possible to avoid for even a hybrid model as the importance of having good results to pass from the LES to the RANS sub-domain is not one that should be overlooked.

3.2.4 Spatial averaging

3.2.4.1 LES to RANS

The cell size required for LES so that the instabilities in the flow can be captured is small because the cell size acts as a filter and any eddies smaller than this will be represented by the sub-grid model. The cells for the RANS domain can be larger as the Reynolds averaged models solve for the average and not the instantaneous values of the variables. This means that at the interface the predicted variables from the model change from representing instantaneous to averaged variables (or vice-versa).

The relationship for the passing of data to the LES sub-domain should not lead to the RANS cells damping out any large eddies near the interface by writing their value to many cells on the other side of the interface, This would mainly be achieved by the careful positioning of the interface. The proximity of the interface to the area of interest in the flow and the size difference in the cells themselves, would govern the size of eddy that could be captured by the LES cells near the interface.

Another issue to be resolved would be the manipulation of the values as they are transferred to the RANS domain. These values will be of more importance as they will be containing more accurate velocity and turbulence data. This will have to be combined so as to allow as much of the information to be preserved for the use in the k-ε model. An obvious way of accomplishing this would be the use of cells where the interfacial areas of the RANS cells on the interface are integer multiples of those on the LES side of the interface and the interpolation of an unmatched grid (section 3.2.4.3). This would then facilitate the transfer of information by removing the interpolation errors.

For a Reynolds averaged problem the value of k is given by

$$k = 0.5\overline{u_i'u_i'} \tag{3.14}$$

which is based on the fluctuating velocity. This fluctuating component is unknown in RANS calculation and so is modelled. By using the LES methodology we calculate the filtered velocity components, which, if the filter width allows for a resolution of 80% of the energy is close to the real instantaneous velocity of the flow.

If the LES model is ‘correct’ then the resolved velocity is close to the instantaneous velocity This then allows the calculation of a mean velocity based on successive approximations for the instantaneous velocity and so $\bar{u}$ and u' can be calculated, hence the turbulence values to be used as boundary conditions for the RANS sub-domain.

Moving the information from the LES domain to the RANS domain is an equally complex task, but potentially more complex as the implementation of the LES should give more accurate velocity and turbulence predictions and it is important that this accuracy is not lost in the movement across the boundary.

The proximity of the cell to the interface may also affect the results in that cell. If the smoothing in the RANS sub-domains from the higher turbulent viscosity affects the cells near the interface on the LES side of the interface, then there will be a need to enlarge the LES sub-domain so that the area of interest is completely modelled by the Smagorinsky model with as little influence from the interface boundary conditions as possible.

3.2.4.2 RANS to LES

Based on the assumption that the model is to give greater accuracy where the turbulence is generated in the LES domain, the turbulence leaving the LES domain is more important. However, for the flow from the RANS domain to the LES domain the flow is going from the time averaged to the instantaneous flow. The problems are that the velocity specified is the average and that the turbulence levels give little idea of the structure of the flow.

The impact of the average velocities used at the interface will have affect the LES region by smoothing the flow near the interface. This may require the LES region to be a little larger than is required to ensure that the turbulent fluctuations are adequately captured.

Relatively speaking, the turbulence level moving from the RANS domain to the LES is expected to be low, and so in the first instance could possibly be introduced as a source term in the calculation for k in the LES domain at the interface. Alternatively the boundary cells on the interface could be perturbed as they were in the initialisation stage and this would increase the velocities and so by implication increase the turbulence level. The problem with the source term method is that it inputs no structure into the LES but simply means that the calculation for that cell will be higher. The problem with the perturbation method is that the increase in the velocities gives no real information as to the structure of the turbulence and could cause the continuity of the problem to be lost. Another method could be to calculate the shear as used in the Boussinesq equation and pass this across the interface so that the viscosity in the LES domain becomes modified and so the turbulence levels are altered because the velocities are altered through the altered viscosity. This is more difficult to implement but could pass the information to the LES domain more accurately.

3.2.4.3 Unmatched grid interface

For a block structured CFD code such as CFX4.4, the most flexible method used to concentrate more cells in a specific region, or to allow regions to move against one another is called an unmatched grid eg Figure3.1. This is essentially a block boundary where the cells on one side do not pass the flow information directly to another cell on the other side, but that the flow is interpolated to two or more cells on the other side of the block boundary.

Following the assumption that most of the turbulence will be generated in the LES region, the main inflow into the LES sub-domain will be of a much lower turbulence level and will contain smaller, lower energy, turbulent structures than are generated therein. This would mean that although the unmatched grid would offer less accuracy than if the cells were matched, flow into the LES sub-domain is less affected by any inaccuracy. If the highest levels of turbulence are to be found in the LES sub-domain, then the relative importance of the size of numerical error matching the turbulence at the interface becomes less important. This approximation does not hold if the inflow into the LES region is isotropic, homogeneous and high in turbulence, but one of the assumptions for the thesis is that the turbulence is generated in the LES region.

If there were a difference in the cell sizes required for the domains means that there could be a difference in the cell sizes in the two sub-domains. This would introduce complications into the passing of information between the two sub-domains. With the cells being larger in the RANS sub domain that on the LES side of the interface, an unmatched grid arrangement could be utilised to allow the passing of data (fig 3.1).

If an unmatched grid were to be used at the interface then the problem with the use of an unmatched grid will be the weighting of values as they are passed form one sub-domain to another (fig 3.1). In the above figure, the cells are a factor of 2 larger in each direction. This means that the exchange of information is easier than if the multiple were not an integer. This problem has been solved for the serial version of CFX4 but the unmatched grid option is not available in the parallel code[11].

To calculate the correct flux between the two sub-domains if the grid does not match a first-order interpolation scheme will be used. To get the flux from the smaller cells of the LES to the larger cells of the RANS sub domain an addition method will be used, where if the face of the LES is covered entirely by the face of the RANS cell then all of the flux will move into the RANS cell. If the LES cell is only partially covered then the flow is assumed uniform over the face and the flux into the RANS cell is a ratio of the area. To move the information

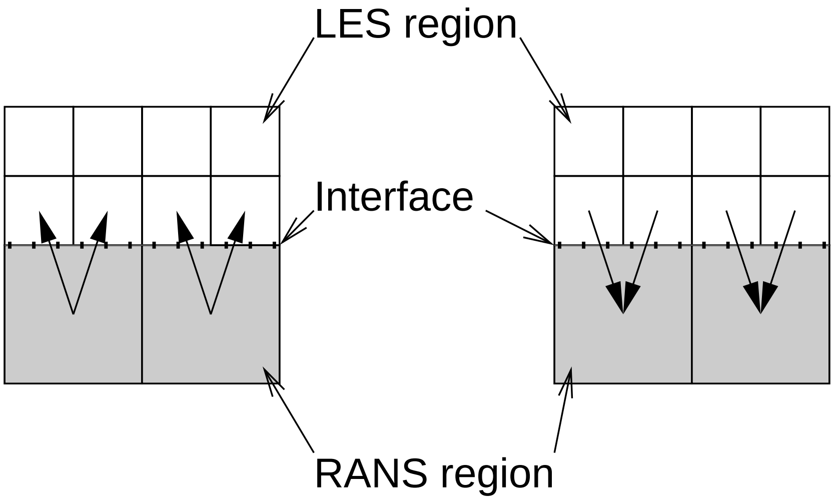

Figure 3.1: Illustrating the potential problems of using an unmatched grid

from the RANS cells to the LES cells there will be interpolation between the larger RANS cells. The gradients between the neighbouring RANS cells will be calculated and the fluxes into the LES sub domain will be graded accordingly. This will mean that there will be no shear induced at the interface and will aid the stability of the solution.

- the values can be written to all of the LES cells from the corresponding RANS cell and all of the LES cells will then write back to the RANS domain. This could set up race conditions and should be avoided.
- The RANS cell will write to all of the corresponding LES cells and the values from the LES cells will be averaged to provide a value from the RANS cell. This will increase the work done at the interface by one line of code.
- The value from the RANS cell will be written to all of the corresponding LES cells and only the value from the closest LES cell will be written to the RANS cell. This could lead to a continuity error, but the monitoring of the residual at the interface will alert the user as to whether this is so.

Initially the mesh will be matched on each side of the interface and this will then mean that each cell will be passing values to a single cell on the other side of the interface. This will then reduce the complexity of the initial case by not including any averaging in a spatial or temporal manner. The main concern is that the continuity of the problem is maintained, following on from this the velocities and turbulence properties are to be calculated in a manner so that the user, having set the interface in the pre processor will not be able to tell where the interface is in the post processor.

The use of unmatched grids in the future will be a challenge as this should enable the maximum speedup to be gained by the combination of modelling methodologies. Unmatched grids will also allow for a much greater resolution of the flow in the region of interest and this will aid accuracy to the model where the turbulence is generated.

3.2.4.4 Discretisation Mechanism

LES requires a higher-order discretisation scheme to enable the accuracy to be retained and the fluctuations in the flow to be captured. Initial work has shown that the central and higher upwind schemes provided by CFX are able to capture the instabilities and this could be used to cross the interface as well. In many of the published works this is the minimum level of accuracy used.

The RANS domain will benefit for a higher-order scheme and this will increase the accuracy. It would allow the use of larger cells were the sub-domain to be coarsened and because it is starting from an initial solution the higher order scheme should remain stable. As the CFL condition has to be met in the LES domain this will in turn add stability.

A further reduction in CPU time could be achieved by the use of a lower order scheme on the RANS domain. This would reduce the storage and make the RANS simulation faster meaning that more of the CPU effort could be concentrated on the LES domain. The greatest speedup would come from the dissasociation of the LES and RANS domains in terms of timestep. If the RANS cells are to be larger then the CFL stability limit will support a larger timestep, thereby reducing further the CPU time spent in the RANS region. Though reducing the CPU time spent in the RANS domain will have a disproportionately small effect on the overall CPU time because the majority of time would be expected to be spent in the LES sub-domain.

3.2.5 Temporal averaging

The LES simulation requires the use of small timesteps so that the fluctuations in the flow can be captured. The use of such small timesteps for the RANS

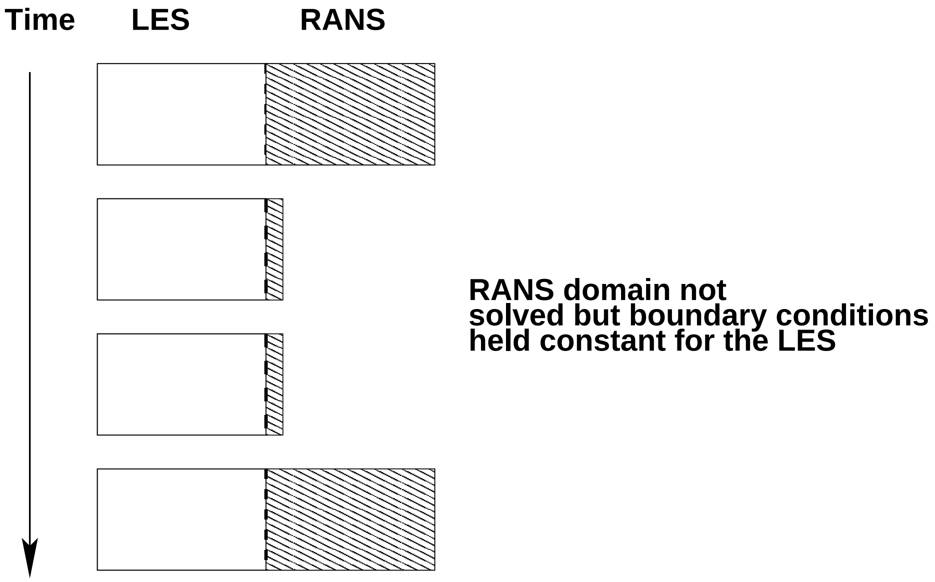

Figure 3.2: Schematic of the use of different sized timesteps

model is not required to get a solution as the CFL condition will centainly be met by the use of smaller cells. Although the CFL condition does not have to be met for an implicit code, the restraining of all calculations to this requirement will ensure an adequate level of resolution. The use of the small timesteps over the whole domain will provide a stable solution so that the algorithm for the interface can be created and optimised.

A possible future development of the idea would be the use of different sized timesteps for the different solvers (Figure 3.2). This introduces problems of how to average the results and when to exchange the details of the flow. The restriction on the exchange of information will have ramifications on the results of the LES as the boundary nodes will be unchanging between timesteps and so the restriction on the LES would seem greater. However, the small timesteps on the LES are because of the small changes that happen each timestep. As the changes are small it may be possible to exchange the data less frequently than every timestep, but not so infrequently as to jeopardise the validity or stability of the solution. For some simulations it may be possible for the RANS simulation to be steady state and not transient at all. This would be possible with the two separate domains so that the halo cells could hold the values constant for the smaller timesteps of the LES. This would not prove to be a problem for the Reynolds averaged side, as the exchange of information would be once for each of its iterations.

It should be remembered that the LES requires more computational effort than the RANS because of the smaller cell size, and so depending on the size of the two sub-domains this may be an academic exercise with little or no benefit to the user. If however the number of LES cells is small compared to the number of RANS cells then there could be a use for this optimisation. The frequency of the RANS steps would still be higher than if the domain was only solved with a RANS model but would be faster than if the exchange of variables took place every step. The trade off is the time to process all of the RANS cells against the time to exchange the data between the two solvers

3.2.6 Implementation of a moving interface

Large eddy simulation calculations are more exact and so will give more accurate predictions of flow properties than the RANS models are able to. The position of the interface will be defined by an allowable level of anisotropy and also by the user. The user will set the maximum size of the LES domain in the two domain model by the size given to the LES domain. The minimum size of the LES domain will be specified by the extent of overlap in the two domains. The actual position of the interface within the overlapping region will start in the middle of the overlap. As the solvers progress the interface will be able to move as the solution changes.

As the solution progresses the interface will be able to move within these boundaries so that the solution will always be the best given the users constraints. The interface will be positioned at the cell centres so that the use of the central differencing scheme will always take into account the cells from either side of the interface. This will then ensure that the domains do not diverge in their solutions.

An ideal option would be to implement a moving interface between the two models by positioning the interface in a block that is solved by both methodologies. This would then be solved by the processors solving the LES and also those solving the RANS model, and the cells where the answer is to be used from would be marked (fig 3.3).

When the central differencing scheme is used and the domain decomposition divides between the two solvers the values that are imported from the other solver will be smoothed out. To prevent this the value at the interface will have to be written to the cell beside the interface at each iteration so that the solutions match each other. It will also mean that the values imported from the other domain affect the results of the receiving domain (fig 3.4). This may cause problems in that the read/write operations take time and it may require

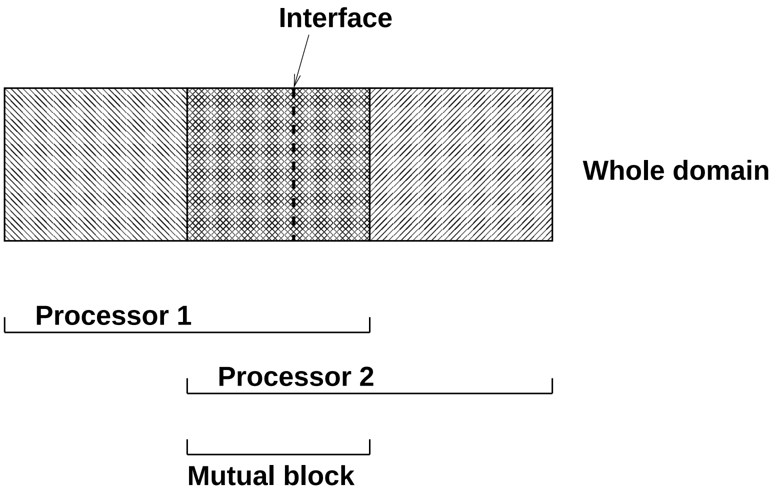

Figure 3.3: Interpretation of a moving interface

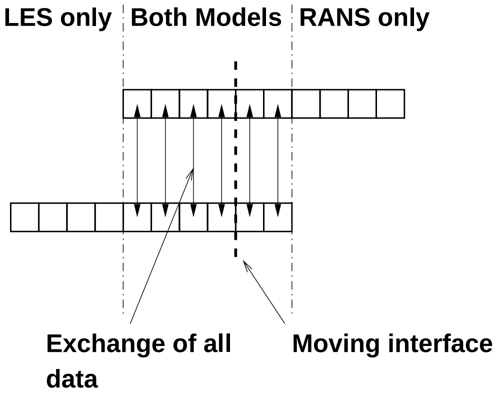

Figure 3.4: detail of the communications cost with the moving interface

a separate method of solving the problem

If a moving interface were to be realised by both processors solving with a mutual block over which the interface could move, then these are ideas as to how the movement of the interface could be realised. The solution of the 'mutual block' could then be obtained by both solvers and once the placement of the interface had been decided the unwanted results discarded. The movement of the interface would be controlled by the program internally and would use the values set by the user. These will be the values of length scale and anisotropy. If the anisotropy becomes less at the interface then the interface could be moved so as to make the LES domain smaller. It may prove unwise to have the interface moving every timestep as the computational effort would be increased every move, in addition to the stability problem that may be introduced by the changing of methodologies for the cells over which the interface is moving. Also the simulation should be given time to adjust to the change in domain after every move so as to reduce the risk of the solution being affected by the movement of the interface. Especially if the interface is going backwards and forwards each iteration. To prevent the constant movement of the interface it will be tied to the same position for a minimum of two fluctuations as defined by the Strouhal number based on the largest fluctuation. This can be calculated before the simulation by the user and the inputting of the timescale into the model will ensure the interface is not constantly moving. To allow for this a minimum time at each position will be set, this will be a multiple of the turbulence timescale so that the simulation can check as to whether there are periodic flows on the correct side of the interface before the interface is moved.

As a moving interface is not currently possible under CFX4 because the blocks are divided among the processors once and once only because the duplication of work is a waste of computational effort. The blocks are divided up along block boundaries, so a user could move the interface manually between simulations, but the increase in user time and effort may not increase the speed or accuracy of the solution. The division of a block itself is possible as the Meshimport program may achieve this. If it were to be done by hand then great care would have to be taken to ensure that the geometry remained unchanged. This still does not allow the possibility of a moving interface with out direct intervention by the user.

For the hypothetical situation with a mutual block which is solved by both processors the two domains will be initially operating on the same timestep and the overlapping regions will be of the same sized cells. The two domains will exchange values at the start of each timestep. The values will be written from the cells next to the interface to a buffer and then into the other domain next to the interface. These values will be re-written to those dummy cells that are not solved. The use of the buffer will prevent the values from overwriting one another. The other way to do this would be for only one of the processors to solve the ‘mutual block’ but the values to be automatically overwritten to the block which is also in the domain as the other processor thought not solved by it. To perform the investigation with a moving interface as described here with a ‘mutual block’ over which the interface may move would need intervention at the source code level. However, the use of a number of cells that are solved by more than one processor and then half of the results discarded is a waste of resources. The advantage over solving for a larger LES region would have to be investigated, as for the ‘mutual block’ to be of real use it would have to allow the interface to move and this would require the inclusion of extra subroutines on top of those used for the averaging at the interface. It should

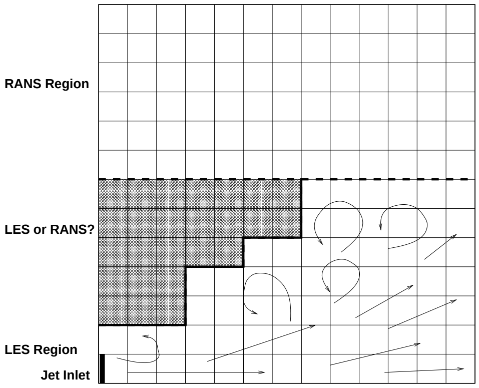

Figure 3.5: Increase in communications cost from a non-trivial interface

not be included as a mechanism for balancing the load over the processors the inclusion of unnecessary work amounts to false economy in the name of balancing the load. A mutual block would also mean that both models would have to run on the same timestep, thereby removing any advantage that may have been gained by coarsening the spatial and temporal resolution in the RANS sub-domain.

The block structured nature of CFX4 means that the allocation of work to processors is done in a static and block by block way to give the optimum load balance, and so it is not possible to move the interface as the simulation progresses.

If the interface were to be moved by the application of a criterion for the positioning of the interface on an individual cell basis. For example, by those cells that meet the criteria for the LES model being tagged in some way. Then the changing shape of the interface would be possible. A method of this sort would allow the cells to change as the large eddies passed, though would make the communication more difficult to calculate and would make the separation of the LES and RANS sub routines onto different processors much more difficult.

3.2.7 Communication between the domains

One problem that may arise is that the shape of the contour does not easily match onto a plane of cells in the block that is parallel to the direction of mean flow. The value of the plane that the interface rests along will be taken to be the maximum value required by the test and then the plane will be extrapolated along the length of the block. This will provide a starting point for the definition of the interface. In the future the domain could be modified so that the blocks are nominated in such a way as to provide an optimised shape for the domain by placing the interface nearer to the boundary specified by the criteria. This would place the LES code strictly where it is supposed to be applied and reduce the extra usage of LES where it does not have to be applied. This problem can be overcome by judicious use of blocks and techniques for this are outlined in the CFX4 documentation.

If care is taken and the cells on either side of the domain matched together well then it is possible to produce an interface of any shape that lies on cell faces. The problem will be the definition of the halo cells for this interface. If the interface is kept to a minimum number of planes in the block then the movement of the interface will be easier and the communication time for the halo cells will be reduced as there will be less halo cells because of the simple geometry of the interface. This is illustrated in figure 3.5 where the solid line interface will have a larger communications overhead than the dotted line. In addition it will be harder to move the solid line as there are corners which have to be accounted for. The use of the dotted line may reduce the time spent communicating between the domains, but this will result in the increased use of CPU resources for the LES domain and so the trade-off in easier definition of the interface may prove to not provide the minimum time for an accurate solution.

For this investigation there will be no movement of the interface and the communication between the domains will be performed using the internal CFX routines. This reliance on the internal routines means that the description of the program given in some depth in the documentation. In summary, each block has surrounding it a layer of dummy cells. These hold as boundary conditions for that block, the values of the cell at the edge of the neighbouring block, which is the reason that CFX4 desires an orderly block structure. The computer sees the blocks as being of dimension (length+2) in each of the i, j and k directions and internally glues the blocks together and writes the data from cells to dummy cells and back.

The geometry for the interface will be required to be simple to minimize the communications cost of the modelling and to make the transfer of variables as simple as possible. This will ideally be perpendicular to the bulk flow to minimize numerical diffusion

The interface requires the exchange of variables in two directions. From the RANS to the LES and from the LES to the RANS. The problems here are that the different domains will calculate different variables in different ways. The RANS domain will calculate the average values of velocity, from these it will infer the Reynolds stresses and then calculate the turbulence properties from these values. Finally the new turbulent viscosity can be estimated and this used to modify the calculations in the next iteration. LES on the other hand calculates the instantaneous velocities in the domain where the mesh acts as a filter and this means that all of the eddies larger than the filter will be captured and those smaller than the filter will be modelled with the sub grid model. The new viscosity is calculated on the basis of simulated rather than modelled quantities and for the simple Smagorinsky model is a function of the density, cell size and the large scale strain rate strain rate ($\hat{S}_{ij}$) where

$$\hat{S}_{ij} = 2\sqrt{S_{ij}S_{ij}} \tag{3.15}$$

Where all of the other quantities are then calculated from these values and the averages have been calculated and then stored as the simulation progresses.

The only other possible need for communication would be if one of the domains were taking longer to converge than the other. The monitoring of the two domains could be initialised to inform the user which part of the solution was not converging. This could be a helpful debugging tool and would also enable the setting of different tolerances for the different domains, where a flag can be used to show that the tolerance has been reached.

However, the standard k-ε model calculates the values for k and ε by the use of transport models. The transport models include a production term and this is modelled by $\tau_{ij}S_{ij}$. The values of these could be calculated in the LES domain and then transported to the RANS domain where they overwrite the calculated values. This will then introduce the modifications directly into the turbulence transport equation and this would mean that the values are used rather that acting as a boundary condition, but will only work if the values in the dummy cells are solved and this is not normal. To overcome this the dummy nodes will have to be solved for the turbulence or the imported value of k used as a boundary condition. Initially the value of k will be used as a boundary condition as this is the simplest option. The writing of k itself will allow the calculation of the gradients to be calculated and so will allow the propagation of the turbulence properties out of the LES domain.

3.2.8 Boundary conditions

Wall boundaries, periodic pairs and symmetry planes are not investigated as there should be no loss of mass. It is possible that periodic pairs on the end of an internal domain may eventually result in the fluid slowing down as the friction from the walls removes energy from the fluid but this is not important to this investigation.

The boundary of the domain is described by a flat velocity profile. The outlet is either a mass flow outlet or a pressure boundary. The main difference is that the mass flow outlet will not allow the inflow of fluid and assumes a developed velocity profile. The pressure boundary will allow inflow and does not assume fully developed flow. The position of the outlet boundary may affect the flow upstream. The mass flow outlet imposes the assumption of fully developed flow on the outlet and this may serve to dampen the turbulence if the outlet is not positioned far enough away from the source of the turbulence.

3.3 Extraction of statistics

The accurate representation of the flow statistics is very important to enable a description of the velocity and turbulence profiles. The available options are either a long running average over the whole simulation or a form of moving average which will require less memory and be more representative of the varying flow.

To obtain a long running average of a variable from a simulation of m timesteps in duration for every cell in a mesh containing g cells requires the storage of $m \times g$ values. This would allow the exact calculation of the average for the whole simulation.

$$\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{3.16}$$

If all of the values during the simulation are store and then the average calculated at the end of the simulation then the storage required for this to be calculated for a long large simulation makes it prohibitively expensive. To reduce the storage costs the summation could be carried out as the simulation progresses so that on the last timestep only the division by m need be carried out to provide the average.

Alternatively the long running average could be calculated as the simulation progresses. The average velocity can be used to indicate as to whether the solution is converged to a pseudo steady state. The running average

$$\bar{u}_i = \frac{\bar{u}_i + (u_{t-1} \times (t-1))}{t} \tag{3.17}$$

is an example of this. This average velocity can than be passed to the RANS domain and be used as the averaged velocity without any manipulation. This will be a seamless operation and so will highlight as to whether the passing of the scalars is correct.

An alternative would be to using a moving window of n values so that at timestep k the average over the last n timesteps is,

$$\bar{x}_k = \frac{1}{n} \sum_{i=k-n+1}^{k} x_i \tag{3.18}$$

and at time $k-1$ the average is

$$\bar{x}_{k-1} = \frac{1}{n} \sum_{i=k-n}^{k-1} x_i \tag{3.19}$$

Combining and rearranging gives

$$\begin{aligned}\bar{x}_k - \bar{x}_{k-1} = \frac{1}{n}\left(\sum_{i=k-n+1}^{k} x_i - \sum_{i=k-n}^{k-1} x_i\right) \\ = \frac{1}{n}(x_k - x_{k-n})\end{aligned} \tag{3.20}$$

which gives for the moving average

$$\bar{x}_k = \bar{x}_{k-1} + \frac{1}{n}(x_k - x_{k-n}) \tag{3.21}$$

This only requires the storage of n previous values and so is less demanding on the memory of the machine.

An alternative method requiring no storage of previous values is the exponentially weighted moving average (EWMA). Consider the following two different averages. The first given by equation 3.18,

$$\bar{x}_k = \frac{1}{n} \sum_{i=k-n+1}^{k} x_i \tag{3.22}$$

and the second starting calculating up to time $k+1$.

$$\bar{x}_{k+1} = \frac{1}{n+1} \sum_{i=k-n+1}^{k+1} x_i \tag{3.23}$$

Both start the averaging process at the the timestep $k - n + 1$. This means that the difference between them is the value of x_{k-n+1} so that by opening the summation, equation 3.23 can be re-written,

$$\bar{x}_{k+1} = \frac{1}{n+1} \left(x_{k+1} + \sum_{i=k-n+1}^{k} x_i \right) \tag{3.24}$$

Equation 3.22 can be rewritten as,

$$n\bar{x}_k = \sum_{i=k-n+1}^{k} x_i \tag{3.25}$$

which allows equation 3.24 for the timestep $k + 1$ to be rewritten with the substitution of $n\bar{x}_k$ for the summation to give,

$$\bar{x}_{k+1} = \frac{1}{n+1} \left(x_{k+1} + n\bar{x}_k \right) \tag{3.26}$$

or

$$\bar{x}_{k+1} = \frac{n}{n+1}\bar{x}_k \;+\; \frac{1}{n+1}x_{k+1} \tag{3.27}$$

It can be seen that this only requires the storage of the previous average and then calculates the current average from that and the current value. The difficulty comes in ascribing values to the weightings. If the EWMA is written as

$$\bar{x}_k = \alpha\bar{x}_{k-1} \;+\; (1-\alpha)x_k \tag{3.28}$$

then the value of α is bounded by $0 \leq \alpha \leq 1$ and is equal to

$$\alpha = \frac{n}{n+1} \tag{3.29}$$

where n is the number of samples in the data ‘window.’ The choice of the value of α is important as it describes how quickly previous values of the variable are ‘discounted’ or forgotten. If $\alpha = 1$ then the average remains constant throughout the simulation. If the flow is known to be oscillating about a mean then this may be desirable, but may not be useful in stationary flows. If $\alpha = 0$ then there is no averaging and the mean is always set to the current value of the variable.

Ideally α should be linked to a time constant of a process: τ_p, which will allow ‘low pass filtering’ such that the fluctuating component is removed but the definite changes with time will be recorded. For a filter time constant of τ_f and a sampling interval of Δt where $\tau_f \leq 0.1\tau_p$ and $\Delta t \leq 0.5\tau_f$, the filter time is described as

$$\alpha = \frac{\tau_f}{\tau_f + \Delta t} \tag{3.30}$$

and given the described values as a function of each other and the dominant time constant of the process the final value of α should theoretically be.

$$\alpha = \frac{\tau_f}{\tau_f + \Delta t} = \frac{\tau_f}{\tau_f + \frac{\tau_f}{2}} = \frac{2}{3} \approx 0.7 \tag{3.31}$$

For the flow over a cylinder where there have been 80 timesteps allowed for one cycle, the sampling could be done every 4^{th} timestep and the filter applied over consecutive samples. However, in practice it is just as simple to monitor every timestep and use all of these to average the flow.

From the results of previous researchers the flow behind the cylinder can be seen to posses these large scale structures and many fluctuations besides. The accurate capturing of the velocity fluctuations with relation to the fluctuating mean will be captured by the EWMA filter with the theoretical settings. However, the closer the value of α gets to one the slower the filter is to respond to changes in the flow and the more like a long average over all time the results are.

3.3.1 Example of the EWMA filter

Examples are shown of filter responses to a sinusoidal wave, where one period is described in 62 steps. The long running average is described by (where as an initial guess $\bar{x}_0 = x_0$,)

$$\bar{x}_k = \frac{k-1}{k}\bar{x}_{k-1} + \frac{1}{k}x_k \tag{3.32}$$

Table 3.2: Weightings for the moving average

$\frac{61}{62}$	for 1 period
$\frac{92}{93}$	for 1.5 periods
$\frac{123}{124}$	for 2 periods
$\frac{185}{186}$	for 3 periods
$\frac{619}{620}$	for 10 periods

Which averages over the whole simulation.

For the EWMA averages the values of α other than the theoretical value of 0.7 are,

This test is not representative of the fluctuating velocities that will be seen in the LES simulation. In order to demonstrate that for a non-smooth output the EWMA formulation will capture the turbulence whilst following a moving mean the following has been performed.

1. For $0 \leq x \leq 500$ in steps of 0.1, the value of y has been calculated for $y = \sin x$

2. In order to allow for turbulent fluctuations, two random numbers $-1 \leq r_1(x) \leq 1$ and $-1 \leq r_2(x) \leq 1$ have been multiplied together to give an error $f(x)$ for each of the values of x. The random number (ϵ) is in no way related to x and has a normal distribution about zero.

3. The 'experimental' value of y was defined by $y = sin(x) + \epsilon$.

4. The averages used above were used to again calculate the averages for the 'experimental' results.

5. For each averaging technique, the magnitude of the fluctuation about the mean was calculated for each point using the formula $f'(x) = \sqrt{(y - \bar{y})^2}$

6. The average value of the fluctuation $f'(x)$ was compared with the calculated average of the actual fluctuation $f(x)$ and the plot of the mean compared with the known mean which is the sine wave.

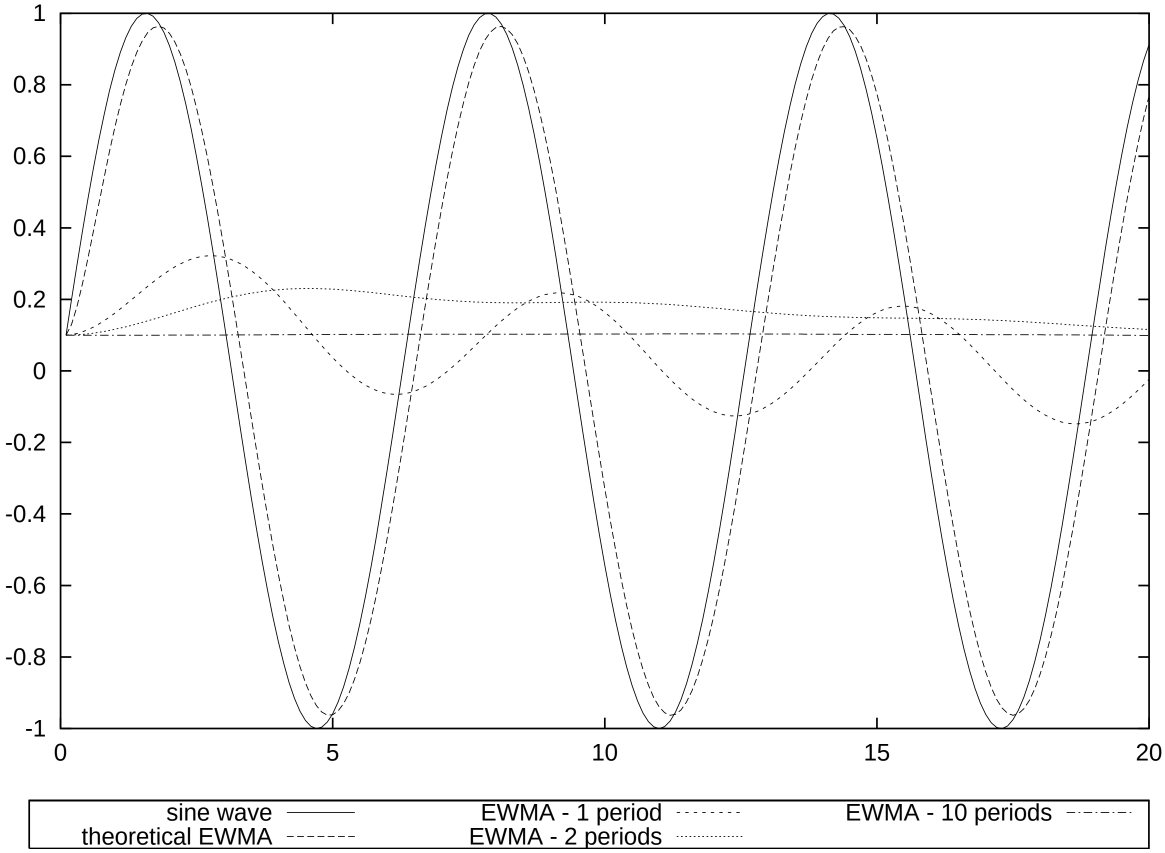

Figure 3.6: Averages calculated for a sinusoidal wave showing the effects of averaging over (implied) different periods

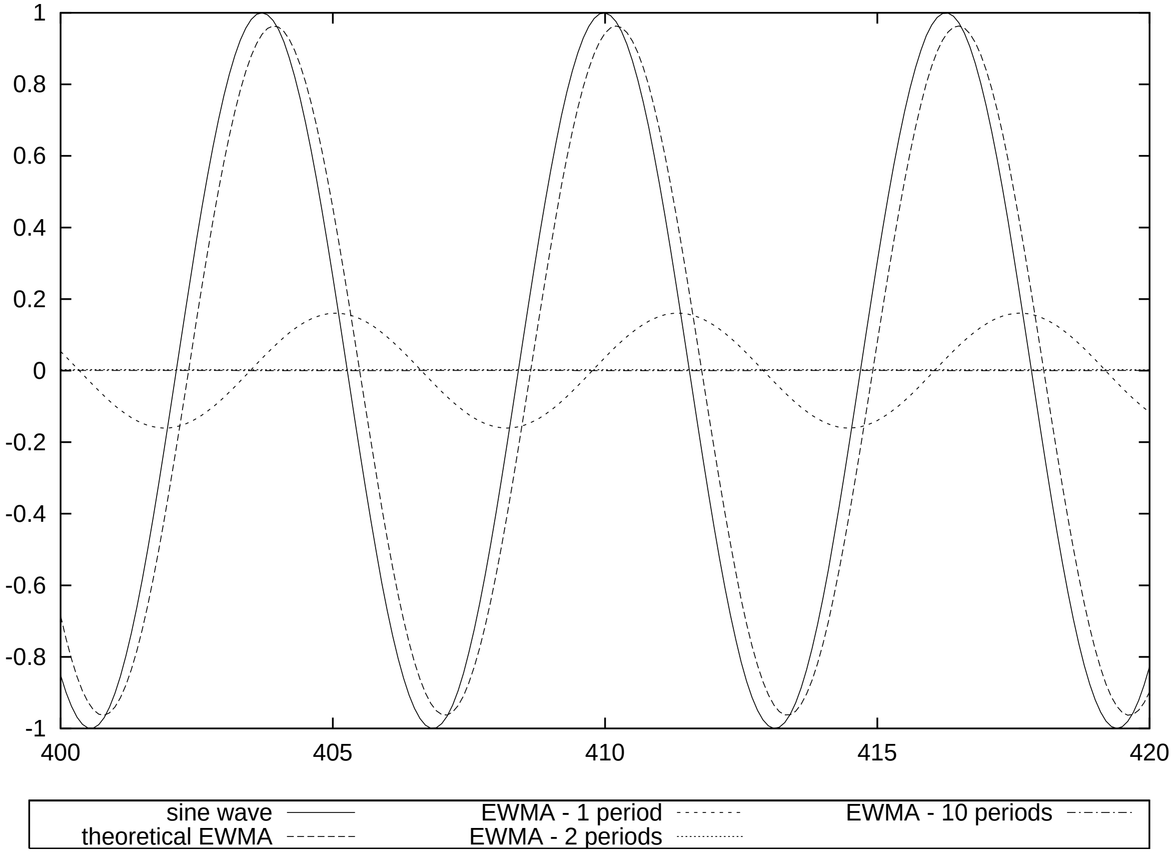

Figure 3.7: Averages calculated for a sinusoidal wave calculates many oscillations after the initial conditions

As expected the longer the implied average the longer the EWMA method takes to 'forget' the initial value and the closer to the long running average the solution becomes

The averages of the fluctuations are displayed in Table 3.3. The figures showing the effect of the averaging (Figure 3.9) on the experimental results, and the comparison of the experimental results with the true sine wave (Figure 3.8) have been combined to show the predictions of the 'turbulent' fluctuations in the flow. In Figure 3.10 it can clearly be seen that the EWMA average most accurately predicts the fluctuations by comparison with the fluctuations. The use of the values from $400 \leq x \leq 408$ is because the averages will have reached a more stable value at that point. Clearly the value of the fluctuation for the EWMA average over one period and the long running average can be seen to move in a sinusiodal way as they are predicting the amplitude of the flow and not the fluctuations about the moving mean. This shows clearly that the use of the EWMA average will allow the averaging to follow a moving mean and calculate the fluctuations from that mean rather than calculating an overall mean for a uniformly fluctuating variable. In essence the flow is decomposed into three. The overall mean, the slowly varying part and the fluctuating part, viz.,

$$\phi = \bar{\phi} + \phi_{move} + \phi' \tag{3.33}$$

A longer averaging period will cause the fluctuation in the variable to be measured as the variation from the mean flow and this will will then mean that the fluctuations vary in a sinusoidal way as illustrated by figure 3.10. This is because the variation from the mean of zero in a sine wave is dominated by the amplitude of the wave and the fluctuations from that perfect wave would mean that the average fluctuation on a perfect sine wave would have amplitude 0.707. It can clearly be seen that the longer the averaging period the closer the value tends towards that number, meaning that it is the amplitude of the oscillation that is dominating the turbulent prediction.

Clearly there is also a 'phase lag' between the EWMA average and the sine wave (Figures 3.6 & 3.7). The reason for this is the value of the weighting function α and the lag can be reduced as $\alpha \longrightarrow 0$ but this adds an increased weighting to the current value and reduces the smoothing from the 'history' associated with the averaging over a moving window. The width of the averaging window does affect the results and may influence them either to smooth out spurious data, or to remove all trends in the output. The optimisation of this process will have a large impact on the outcome of the experiments.

Table 3.3: Effect of different averaging periods

$f(x)$	0.248361
$\bar{y}$, $f'(x)$	0.668364
EWMA(theoretical), $f'(x)$	0.237657
EWMA(1 period), $f'(x)$	0.65558
EWMA(2 periods), $f'(x)$	0.670347
EWMA(10 periods), $f'(x)$	0.672467

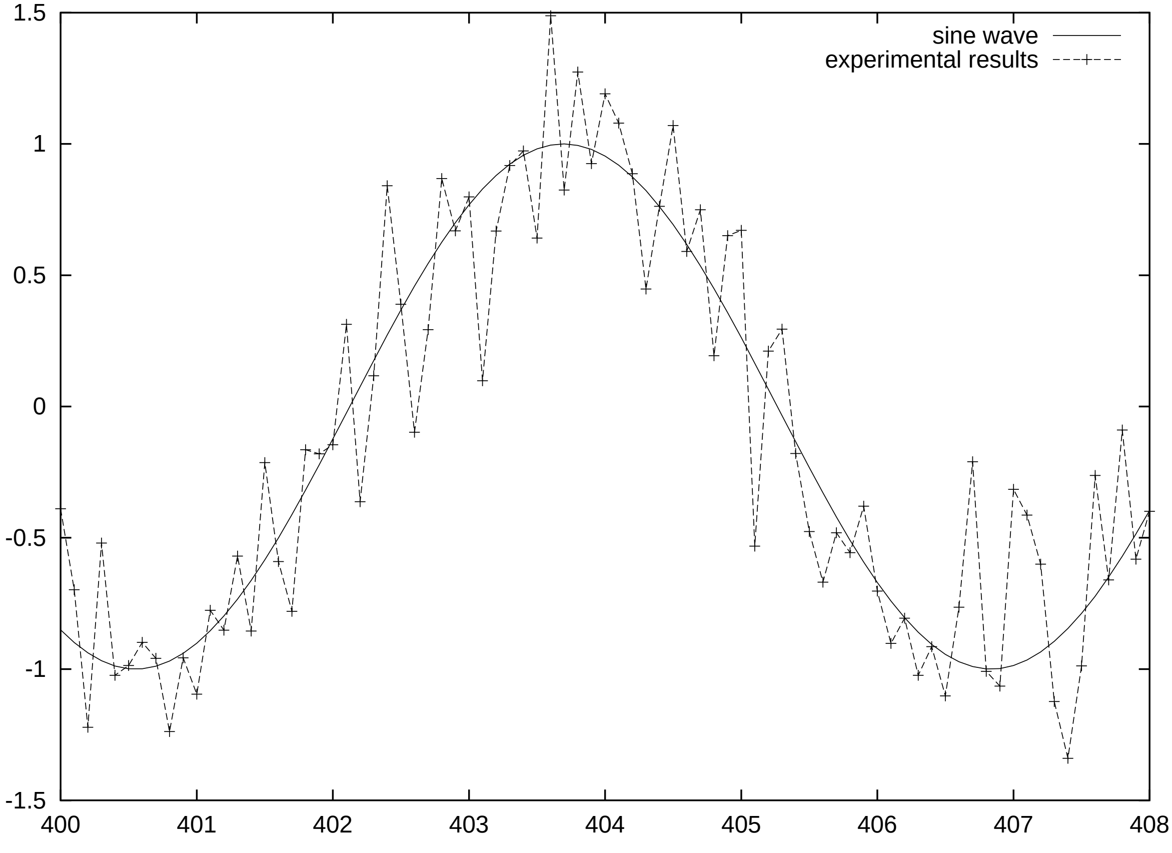

Figure 3.8: Comparison of the pseudo-experimental results and the sine wave

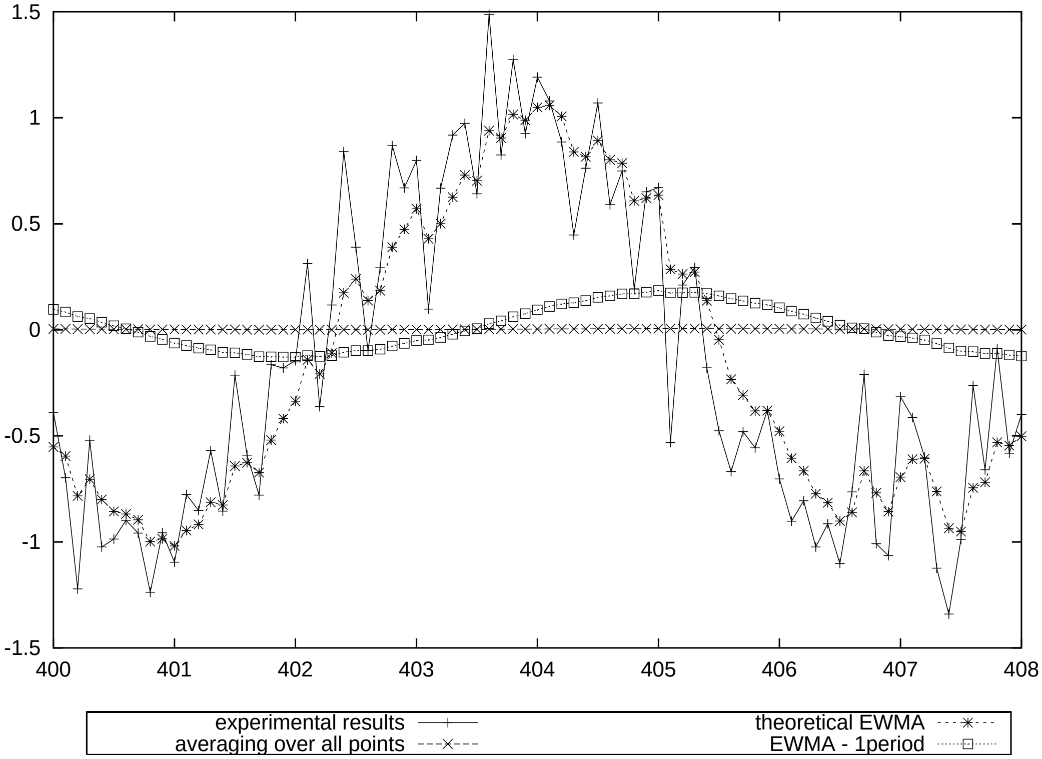

Figure 3.9: Averages of the pseudo-experimental results

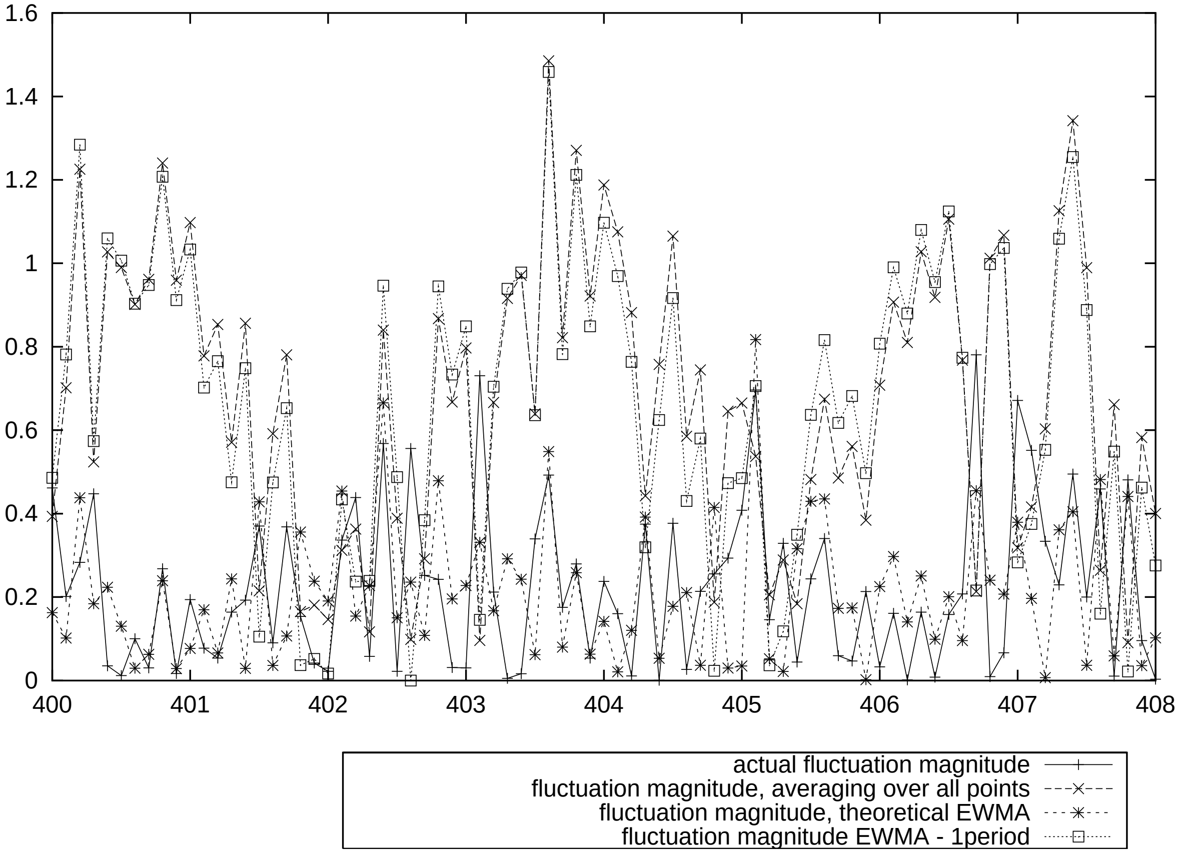

Figure 3.10: Analysis of the size of the fluctuations

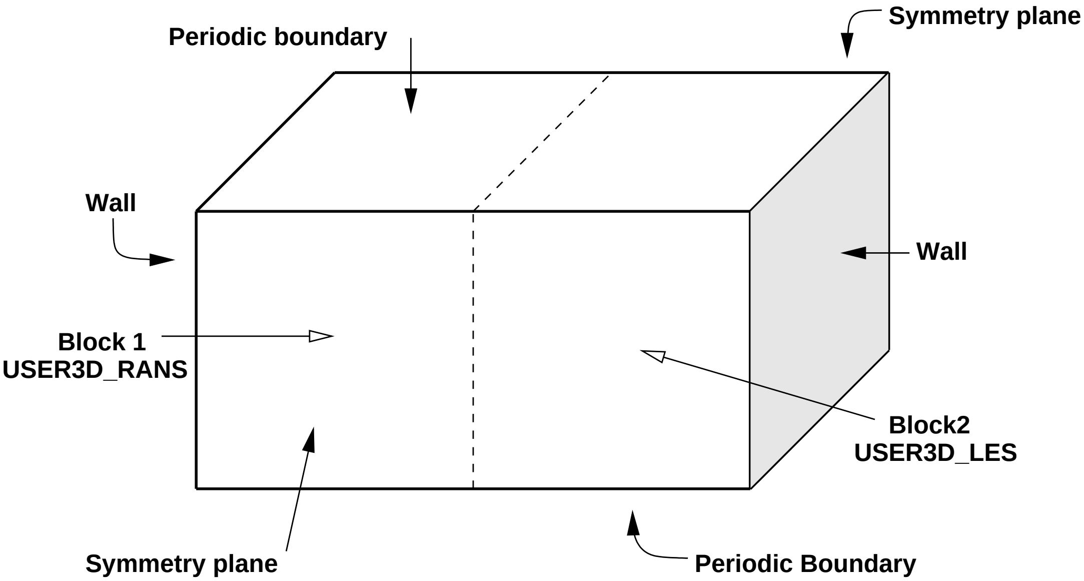

Figure 3.11: Schematic of the twoblocks domain

3.4 Twoblock test model

In order to develop the hybrid model as easily as possible, without concerns for the grid or domain decomposition a simple geometry was created consisting of two cubes, each 10 × 10 × 10 cells in size.

3.4.1 Method

The domain decomposition for the problem to work will have to be done by hand to ensure that the different domains are on different nodes in the parallel machine. The effect of this will be that the parallel code will have to be modified in line with the message passing software to enable the different nodes to execute different solvers. The computation will only be efficient if the time for the computation will have to be long compared to the time for the communication between nodes. It is also important to balance the loads on the processor so that they spend as little time as possible idling whilst waiting for another CPU to finish.

Periodic boundaries were placed at the top and bottom with vertical upward flow. on the two vertical faces symmetry patches were placed and at the ends of the domain were places walls to give a measure of turbulence generation. Periodic patches were used to allow the boundary layer to develop in the space available and to reduce the effects of inlet boundary conditions. The simulations were only executed for a short time so the effect of the drag from the walls was ignored

On the block Solid-1 was placed a 3D patch called `USER3D_RANS`, and on Solid-2 another called `USER3D_LES` so that the user fortran could be used and would be able to distinguish between the two blocks. In order to enable the code to smooth the viscosity at the interface between the two models the mutual face between Solid-1 and Solid-2 was named `USER2D_INTER`.

Control over the allocation of the blocks to the processors was achieved using a slightly modified `parsolve4` script file and a PRL file of my own creation. The `parsolve4` script allows the user to define the PRL file which allocates blocks to processors and a modified `parsolve4_mi` script means that this is then retained and may be used again. The PRL file tells the program how many processors to divide the job over, and which processor each block is allocated to. The use of these files, in conjunction with the `parsolve4` run line option of including a config file meant the allocation of blocks to processors could be defined, monitored and then modified. The config file also permits the use of user defined load balancing for the distribution of work amongst the processors, Though the grid has been specifically developed so that there are approximately the same number of cells on each processor (±10%)

At this point a short transient k-ε run was executed to provide a starting solution. From this dump file the hybrid RANS/LES model could be started.

Using the user Fortran subroutine USRVIS the Smagorinsky subgrid scale model could be used to overwrite the value of the turbulent viscosity and so allow the execution of a large eddy simulation. This was confined to Solid-2 and processor 2 by the use of IF statements, combined with `IDOMNO`, the variable set to the processor number. The code has been constructed in such a way as to allow ease of expansion for any number of processors for the LES and RANS domains. As the user fortran is compiled and run for both of the blocks, there had to be a comparable section for Solid-1 on processor 1 which would write the value of the turbulent viscosity from the k-ε model to the new value of μ_t without altering it. The user routine USRWTM was used as a turbulent wall multiplier as the modelling using LES near walls is still complex. The viscosity routine USRVIS is invoked every iteration, and so the new value of the viscosity is always up to date. USRVIS contains two critical sections, one for each modelling approach, so that the turbulent viscosity for the $k-\varepsilon$ model on the lower numbered processors remains the same, but the turbulent viscosity in the LES region is overwritten for by the Smagorinsky SGS viscosity.

The geometry has been constructed so that when the decomposition of the domain takes place the USER2D_INTER patch is on the RANS side of the interface, that is, it is marked as being on the block with the patch USER3D_RANS. This means that the smoothing is done on this side alone. Smoothing of the viscosity difference is further complicated by the fact that the minimum size for a block in CFX4.4 is 3 cells in any one direction (except for a 2D model). This means that the smoothing has to ideally be performed in 3 cells from the interface to prevent the possibility of crossing a block boundary. Initially smoothing was attempted by simple interpolation each side of the interface. Although this is simple to implement in serial it is not always easy to find the interface in the parallel run as it only resides on one of the blocks that make up the interface. This means that the smoothing can only take place on the one block and an error is generated if the application tries to find the values on the other side of the interface.

It is not possible to use a method as outlined by Speziale [107] as it is not possible to have the difference in cell size that will make a substantial difference in the turbulent viscosity in CFX4.4 because of the structured nature of the mesh.

In an attempt to overcome the differences in turbulent viscosity, smoothing was tried by the introduction of a user scalar, which would be set to the value of the LES turbulent viscosity in the LES sub-domain and then would modify the turbulent viscosity of the RANS sub-domain as it was convected into it. The diffusivity of this scalar has been set very small so that it is effect a flow followed, and so similar in conception to some of the ideas behind dynamic SGS modelling. In addition the values of k and ε that have been calculated in the LES sub-domain are convected by their transport equation (As it is not possible to run part of the parallel simulation as laminar and part as turbulent, the whole simulation is a turbulent one.)

A further user fortran subroutine USRTRN has been used to monitor the flow and calculate the values of the average velocities and deviations from those velocities, this can then be used to calculate the turbulence levels in the simulation and see how these compare to the RANS models. An exponentially weighted moving average technique has been used to calculate and modify the average for the velocity and turbulence levels, and it is these averaged values that are used to overwrite the values of k and ε

CFX International had tested a subroutine that would calculate a random field to impose over the initial solution, but have concluded that this method does not increase the speed of generating LES data as the random field is first damped out before the more chaotic LES velocity field is created. In addition,

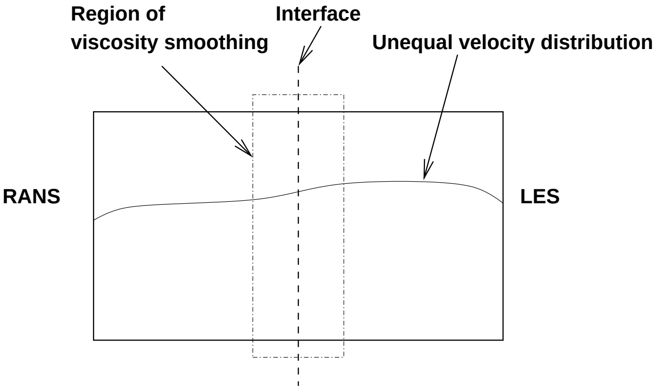

Figure 3.12: Results of the ‘twoblocks’ experiment highlighting main flow features

in a more complex domain there may be more processors working on the LES model than the RANS. By putting the smoothing on the RANS side where there is only one processor the modifications required to execute the program will be reduced.

3.4.2 Results

The use of a very simple geometry enabled the idea to be tested. The computational models could be executed side by side and the smoothing of turbulent viscosity could be achieved. One issue highlighted was the higher velocity in the LES sub-domain. This was due to the lower turbulent viscosity and for internal flows as the one modelled here should not prove to be a problem as the flow over a cylinder is contained and there can be no loss of mass from the way in which the model is created.

3.5 Implementation of hybrid technology onto the bluff body

The domain to be modelled was based on that used by Moin for the same study [49]. However, due to the computing power available the mesh was simplified. The number of cells was reduced to allow a reasonable computational time but at the same time trying not to reduce the complexity to the point of the removal of confidence in the model.

The grid was refined until a solution was obtained that was not dependent

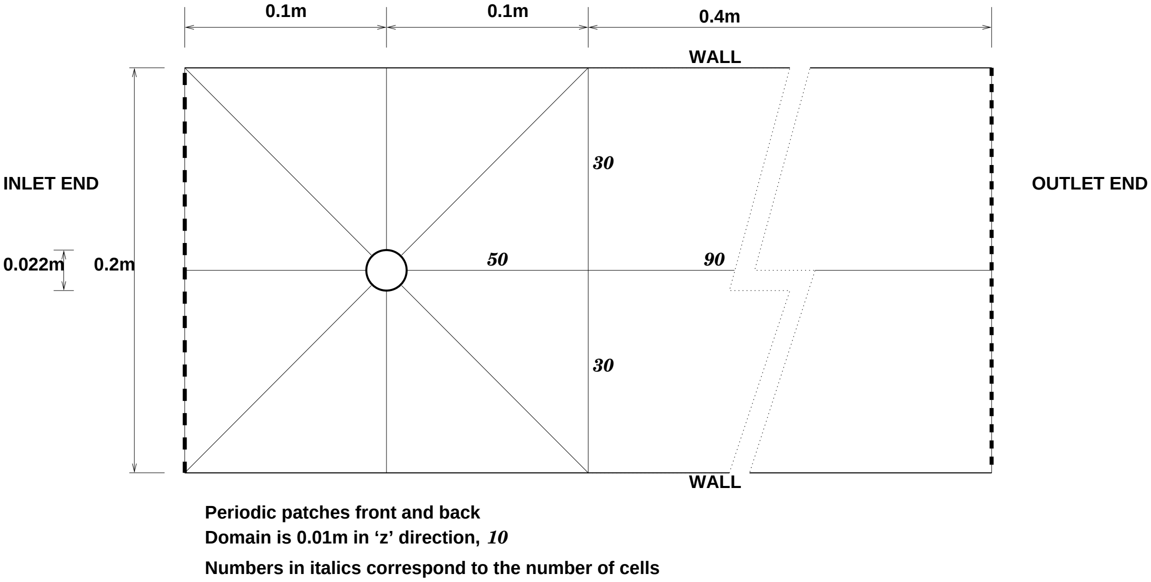
0.1m
0.1m
0.4m
WALL
INLET END
OUTLET END
30
50
90
30
0.022m
0.2m
WALL
Periodic patches front and back
Domain is 0.01m in 'z' direction, *10*
Numbers in italics correspond to the number of cells

Figure 3.13: Diagram of the computational domain for the bluff body study

on the grid, where a doubling in the number of cells failed to change the solution from the simulation so that the solution could be said to be 'grid-independent', for a reduction in residuals of over four orders of magnitude. This grid-independence was achieved for the k-ε model as the transient nature of the Smagorinsky model meant that the refining of the grid in the same way for the LES model requires the collation of averages and then the time for the recalculation at double the resolution, whilst more exact for the study, is time consuming and expensive. All of the simulations were performed on the same grid with the same boundary and initial conditions using the same timestep. The only differences were the turbulence models used.

The grid was further refined in the region of the cylinder such that the y^+ values for the Smagorinsky model were less than 7 giving a higher resolution of the wall layer. The grid was created to facilitate load balancing for the parallel solver when the domain was divided up by hand.

Four methods were used to effect the novel 'hybrid' model for the purposes of this study. In order to calculate the timestep required the Strouhal number of the vortex shedding was calculated and the timestep size specified to be 1/80 of the period of fluctuation. The Reynolds number was set to 3 900 and then the velocity at the inlet calculated with the density and viscosity set to $1000[kg\,m^{-3}]$ and $0.001[kg\,m^{-1}\,s^{-1}]$, giving an inlet velocity of $0.1772727[m\,s^{-1}]$ (assuming a flat velocity profile). As it is the flow around the cylinder that is of interest, and in order to retain similarities with previous workers, there has been no attempt made to simulate developed flow at the inlet.

Each of the models was generated in a similar way so that the difference between test models was minimised. In order to obtain an initial guess from which to start the LES models the standard k-ε model was used. This model was run for 5 periods, where a period is a time period as calculated from the Strouhal number. The 'hybrid' or LES model was then introduced (The LES model was also executed over the k-ε model in an attempt to replicate any difference in CPU time) and the model run for a further 10 periods. After this time the simulation of a final 10 periods took place and it was during this final run the the statistics were gathered. At each timestep a maximum of 30 iterations was specified.

The differencing schemes that were used were higher-order upwind (HUW) for the u- and v- velocities and central differencing in the spanwise direction. Temporal differencing was achieved through the standard backward differencing. Linear wall profiles were used as the wall function is required for the k-ε model.

At the inlet the velocity profile was flat and the viscosity modifier (if used) was given the value of the viscosity

Once working, the transition from a simple two block geometry to a larger geometry was made difficult mainly by the load balancing condition on the processors. The allocation of blocks to processors where there was a large number of cells for the LES compared to the RANS modelling technique, meant that the modelling required three processors to give a more even load distribution amongst them. This is possible in the user fortran by changing the LES IF statement from

```
IF(IDOMNO.EQ.2)
```

to

```
IF(IDOMNO.GE.2)
```

which will mean that processors 2-3 will work on the LES part of the problem. The smoothing will still be handled solely by processor 1 and the allocation of blocks described in the PRL file as before. The config file is altered so that there is a larger discrepancy allowed in the load balancing across the processors.

3.5.1 Smagorinsky model

The Smagorinsky model was executed over the k-ε model so that any effects of the RANS technique would be present in all of the cases for fairness of comparison. This will then act to remove the underlying effects of the k-ε model as they will be present in all of the simulations and the hypothesis can be tested.

3.5.2 Hybrid and Newhybrid models

The value of the turbulent viscosity will not be moved across to the RANS domain. This is because the standard k-ε model explicitly calculates the turbulent viscosity in each cell.

Where the LES model is executed the turbulent viscosity that it predicts overwrites the $k - \varepsilon$ value and this is used at that position. Once the flow is convected across the interface an interpolating algorithm is used to smooth the lower value of the turbulent viscosity on the LES side, with the higher turbulent viscosity on the RANS side. As shown in the code sample below, the averaging used was simply the algebraic mean of the two viscosities.

One of the concerns of this method is that the suppression of the predicted turbulent viscosity in the RANS region may lead to an incorrect prediction of the values of k and ε, which will affect the next prediction of the turbulent viscosity which is calculated from,

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{3.34}$$

but then the value of the turbulent viscosity affects the bulk velocity in the momentum equation. This then impacts on the turbulence levels and so in turn modifies the turbulent viscosity.

If there is flow migration into the LES region and acceleration there then the shear will also increase and so the turbulent viscosity also. This may be large enough to stabilise the problem but this will need to be investigated.

The first attempt at a 'hybrid' model involved the use of a viscosity modifier. For this a user scalar with a low diffusivity ($0.000001[kg\, m^{-1}\, s^{-1}]$) was used to minimise its diffusion so that it will follow the convected flow closely and not smooth the results by diffusion. In the LES region the value of this scalar was given the value of the turbulent viscosity as predicted by the Smagorinsky model. When the scalar is carried across the interface by the fluid, there is a discrepancy between the value of the scalar and the turbulent viscosity in the RANS sub-domain. In order to smooth out the differences the following code has been implemented. The variable names are, TE for turbulent kinetic energy, ED for dissipation rate, VIS for viscosity, VISN for the new viscosity, MU_LAM for the laminar viscosity and SCAL refers to a user scalar. Essentially the code is calculating the turbulent viscosity that would be used by the $k - \varepsilon$ model and then bounding the new viscosity by the laminar viscosity and twice $k - \varepsilon$ turbulent viscosity as a safety check. The turbulent viscosity is then compared to a scalar and the value modified to smooth the effect of the change of models. The user scalar is then set to the value of the modified viscosity to compare with the viscosity calculated in the next time step.

```
       IF(IDOMNO.LE.1)THEN
       call IPALL('USER3D_RANS','USER3D','PATCH','CENTRES',IPT
     *          ,NPT,CWORK,IWORK)
           DO 110 L=1,NPT
             INODE = IPT(L)
c SAFETY CHECK
       MU_T=MU_LAM+((0.09*TE(INODE,IPHASE)**2)/ED(INODE,IPHASE))
       VISN(INODE,IPHASE)=MIN(MAX(VIS(INODE,IPHASE),MU_LAM),
     &                                (2*MU_T))
c MODIFIER
             IF((SCAL(INODE,IPHASE,ISC14)/VISN(INODE,IPHASE)).GE.LIMIT)
     &            THEN
               SCAL(INODE,IPHASE,ISC14)=VISN(INODE,IPHASE)
             ELSE
               VISN(INODE,IPHASE)=0.5*(VISN(INODE,IPHASE)+
     &              SCAL(INODE,IPHASE,ISC14))
               SCAL(INODE,IPHASE,ISC14)=VISN(INODE,IPHASE)
             ENDIF
 110   CONTINUE
```

Where IPALL is an internal CFX routine that will allow the user to loop over all of the cells in the domain and perform operations on them. In this case the code is only executed on processor (IDOMNO) one, and only for the code in the RANS sub-domain of the grid (USER3D_RANS) which happen to coincide for this model. It was important to remember that the call only worked for the cells allocated to that processor as the division of different models to the processors was made that much easier. The code then prevents any anomalous values of the turbulent viscosity from being introduced by bounding the value of the viscosity between the laminar viscosity and twice the value calculated by the k-ε model. This is a safeguard in case of errors arising from the interpolation at the interface.

Once the checking is complete then the modified value of the viscosity is calculated. The first check is to see whether the viscosity will be modified by the scalar and this is done by specifying the ratio between then to be less than the value of `LIMIT`, which for the Hybrid model is set to 0.9. If the difference is small the value of the scalar is set to be that if the turbulent viscosity and will no longer influence the flow.

If, however, there is a larger difference between the value of the scalar and the turbulent viscosity then the turbulent viscosity is set to be half of the sum of the scalar and the turbulent viscosity and the value of the scalar set to be equal to the new (modified) viscosity.

It is anticipated that the value of the turbulent viscosity will be different across the interface and this method will allow for some smoothing at the interface. The change in the viscosities will affect the calculation of the velocities and pressure fields in the next iteration which will then affect the future time value of the turbulent viscosity. However, by that time the flow will have been convected downstream and the new turbulent viscosity will be modified by a new scalar that is convected from the LES region.

The NewHybrid model is the same as the Hybrid model but with a limit of 0.95

The results will always be dependent on the mesh because the smoothing is based on the grid. However, the mesh has been tested for grid independence with the $k-\varepsilon$ model and the placement of the interface in a position where the smoothing will not affect the results, mean that the grid effects are minimised.

3.5.3 Nomodifier

At this point the model was tried where the LES and RANS regions were simply butted together. This method was named Hybrid-Nomodifier. For this simulation the convection diffusion equation for the user scalar was still solved even though the viscosity was not modified in any way.

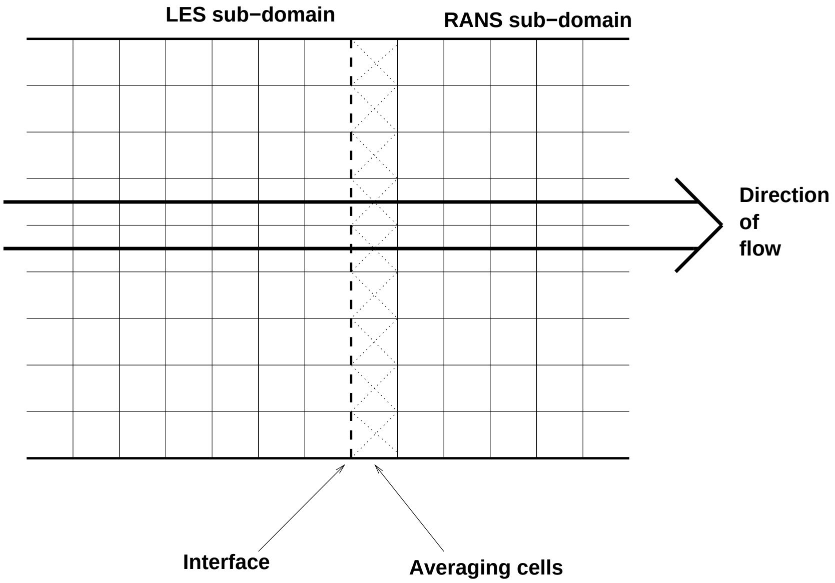

Figure 3.14: Schematic of the use of the av3 modelling technique

3.5.4 av3 model

The final method tested was to overwrite the velocities in the cells next to the interface with the time averaged values from the LES domain. This technique would then allow for the development of a different sized timestep strategy as the average value on the RANS side of the interface could be held constant for the smaller timesteps of the LES sub-domain. The averaging time chosen was that of one period and this was implemented by calculating the average in the same way as for the EWMA average weighted over 1 period and then apply the average velocity to the first cell in the RANS sub-domain by using the USRSRC subroutine so that the velocities were applied as source terms in the coefficient matrix

As the hybrid modelling technique involves the overwriting of the instantaneous velocities at the interface with the average values there may be a violation in mass conservation for the incompressible fluids studied. Two methods present themselves as potential solutions to this problem. The first is to use a pressure boundary on the domain to prevent a mass continuity error. This is highlighted in the CFX documentation as a solution where this may be a problem. The second solution is to only apply the imposition of the average velocities over most of the domain. Once the averages have been built up then there should be no problem with the imposition of values, but during the generation of the averages the small mass errors will be ‘corrected’ by the flow across the interface where there is no overwriting of the average values

The application of these averages to the solution was not started until after the completion of one period to allow the development of the averaged values. In order to save time when the overwriting is to take place, on the first timestep of the simulation a look-up table is generated so that all of the node numbers for the overwriting are stored in a 1D array for ease of access. The distance in the I direction is only 1 as the average is only required for the first cell into the RANS sub-domain. The values of JCC and KCC alter as the layer of cells at $ICC = 1$

```
C   Call IPREC for blocks 9 and 10 and get the node numbers.
      CALL IPREC('SOLID 9','BLOCK','CENTRES',IPT
     &           ,ILEN,JLEN,KLEN,CWORK,IWORK)
      ICC = 1
       DO 103 KCC = 1,KLEN
         DO 102 JCC = 1,JLEN
           INODE = IP(ICC,JCC,KCC)
C   Write the node numbers to an array
           USRSOLID9(((KCC-1)*JLEN)+JCC) = INODE
 102     CONTINUE
 103   CONTINUE
```

The array USRSOLID9 contains all of the node numbers that are to be overwritten for solid 9. This was repeated for block 10. The reason that the values are not combined into one large array is that CFX4 is solved block by block and so the creation of an array for each block may aid in faster execution.

Once the averaging has started the values are modified as below. The values for the start and end of the `do loop` are block specific and are imputed by the user. In this instance block9 is 30 × 10 in the $I = 1$ plane so the loop has to contain 300 counts, one for each node. The `SCAL(KNODE-1,1,ISC1)` refers to the user scalar that is storing the averaged velocity. The new source term, is given by the old source term, with the addition of a weighting based on the difference between the instantaneous and average velocity.

For each node previously stored in the 1D array `USRSOLID9` addressed by the counter `KOUNT`, the difference between the instantaneous velocity and the averaged value is calculated and this is then used to modify the source term for the calculation of the velocities in that iteration [11]. The pressure is left unmodified as it is to be used to smooth out any irregularities as the pressure distribution is calculated after the velocities for this iteration are calculated.

```
C  Loop over all of the cells to be modified in block9
      DO 220 KOUNT = 1,300
      KNODE = USRSOLID9(KOUNT)
C  First add the source terms to the u velocity
         If(IU.EQ.IEQN)THEN
           SU(KNODE,1) = SU(KNODE,1) +
     &        ( AM(KNODE,4,1) * (SCAL(KNODE-1,1,ISC1) - U(KNODE-1,1)))
```

The values of the averages are calculated at the interface for the first few periods with no averaging taking place. After that the averages are imposed on the flow passing across the interface. While this is happening the averages are still being calculated so as to keep the average current. This is especially important as the theoretical EWMA average is being used as the moving average has to be calculated regularly.

Due to the nature of this technique the imposition of the average at the same position as all of the other interfaces affected the vortex shedding from the cylinder. To overcome this the interface was placed at the end of the other domains with another two blocks added at the end to allow the averages to be seen in a RANS sub-domain. This however, meant that the velocity profiles would all be in the LES sub-domain and so an indication of the transfer of the variables across the interface would not be possible except by graphical means. The additional two blocks also meant that an extra 60 000 cells had to be solved. To keep the load balancing as even as possible this simulation used four processors rather than increase the work of the three in use on the previous methods. This will have the benefit of keeping the CPU loads similar to the other methods so as to be able to compare the CPU times for the different methodologies.

3.5.5 Monitoring of averages

For each case the average velocities for the simulation was calculated in the user fortran subroutine USRTRN. This routine is called at the end of each timestep when the iterations have converged and the answer for that timestep is reached.

First of all the highest Courant number is calculated so that this can be monitored. Even though CFX4.4 is an implicit code the use of the stability criteria means that the temporal resolution is such that the cells will capture the flow variations.

The fluctuating velocity is then calculated as the difference between the mean velocity and the new velocity. The fluctuating velocity is calculated and stored. Once this has happened the average velocity is calculated from the `EWMA` averaging method and this value is stored as a user scalar. User scalars are listed in the appendix. The value of `EWMA` is the weighting fraction on the averaging and `AMWE` has the value 1-`EWMA` to ensure consistency.

```
c-----------------

c  Calculate the fluctuating u v and w, as the difference

c  between the instantaneous and the mean c
        SCAL(INODE,1,ISC5) = U(INODE,IPHASE) -  SCAL(INODE,1,ISC1)
        SCAL(INODE,1,ISC6) = V(INODE,IPHASE) -  SCAL(INODE,1,ISC2)
        SCAL(INODE,1,ISC7) = W(INODE,IPHASE) -  SCAL(INODE,1,ISC3)
c c-----------------
c
c  Calculate the new average velocity as from the EWMA algorithm c
        SCAL(INODE,1,ISC1) = EWMA*SCAL(INODE,1,ISC1)+
     &                          AMWE*U(INODE,IPHASE)
        SCAL(INODE,1,ISC2) = EWMA*SCAL(INODE,1,ISC2) +
     &                          AMWE*V(INODE,IPHASE)
        SCAL(INODE,1,ISC3) = EWMA*SCAL(INODE,1,ISC3) +
     &                          AMWE*W(INODE,IPHASE)
      WRITE(NWRITE,*)'CALCULATED NEW AVERAGE VELOCITY'
           WRITE(NWRITE,*)'IDOMNO',IDOMNO,' <<'
c c-----------------
```

The turbulent kinetic energy is then calculated and the fluctuating value and average value calculated as before. If the level of kinetic energy is higher by this method than from the k-ε model, then the k-ε value is overwritten.

The periodicity of the flow has been used to allow the assumption of a long running average over all time [108]. Given that the flow is periodic and within the deterministic nature of the periodicity, it can be assumed that each period is the same as every other period once the flow has developed. This means that the statistics from one period of developed flow will be the same as for all other periods and so only the details of one flow period are required.

At four points equally spread throughout the last period 'dump' files were created as snapshots of the flow. From each set of results the profiles at 1.06, 1.54, 4, 5 & 10 diameters downstream of the centreline of the cylinder were determined, to allow comparison with Kravchenko and Moin [49]. The results were then averaged across all four dump files to give an estimate of the long running average. From the averages the turbulence predictions could be made at every snapshot and the averages of these calculated to give a prediction of the turbulent stresses.

3.6 Conclusion

In order to construct a hybrid model, the domain is split into sub-domains. Each sub-domain will solve for the flow-pattern using either the LES or the RANS approach, and each approach will be carried out on a separate processor. The many considerations that come with the construction of such a solver have been discussed and from the discussion a number of methods to provide the link between the LES and RANS methodologies have been proposed for testing.

Chapter 4

Results

4.1 Instantaneous contours

Pictures from the final timestep of the final simulation are presented for each model. The order of the models is first the $k-\varepsilon$ model, then the Smagorinsky model. Following these the results for the hybrid models are shown, these are in the order of the 'av3' model, then the 'hybrid' model. Finally the hybrid model without the viscosity modifier and the 'newhybrid' model.

For each model the contours of the three velocity components, the pressure and the turbulence properties of the flow are given. In addition the y^+ values are shown for the Smagorinsky model illustrating the distance form the wall to the first node. All of the contour plots are taken midway through the domain in the z-direction to minimise the effects of the periodic boundaries, and for equal comparison. On the left of the illustrations is the scale. This is consistent for comparable plots, except where the values were notably different from the norm.

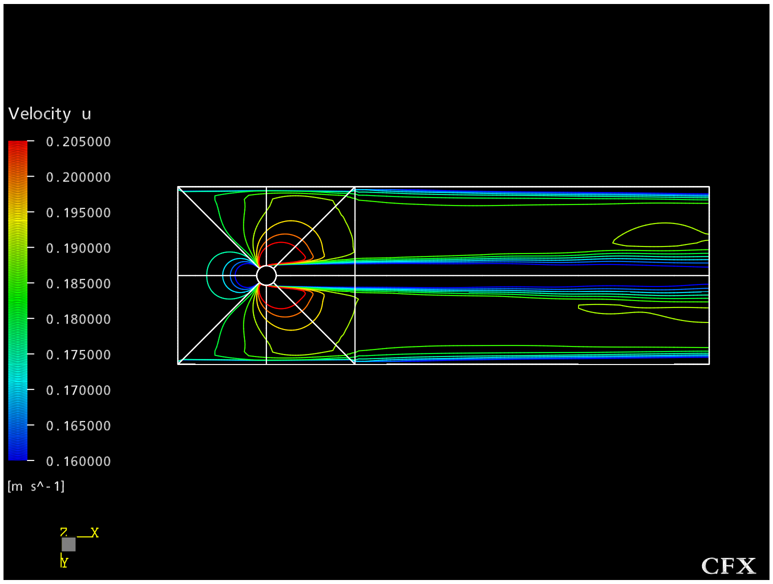

Figure 4.1: contours of u-velocity from the k-ε model

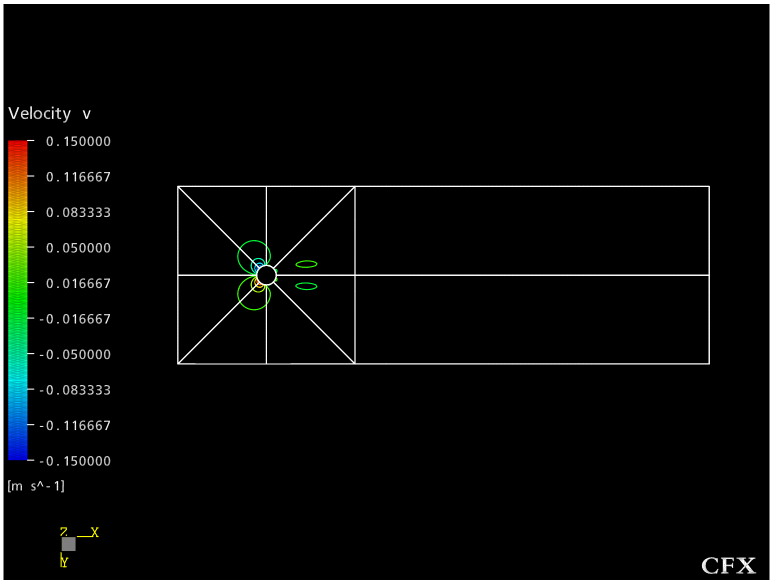

Figure 4.2: Contours of v-velocity from the k-ε model

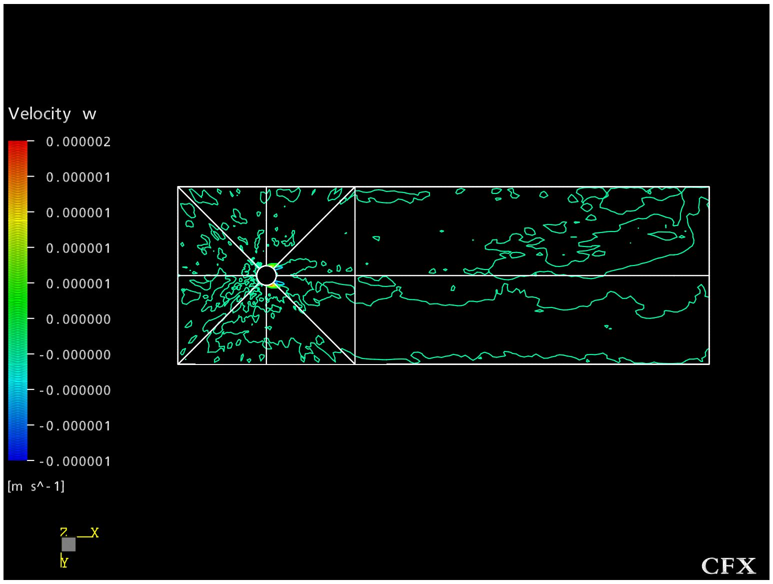

Figure 4.3: Contours of w-velocity from the k-ε model

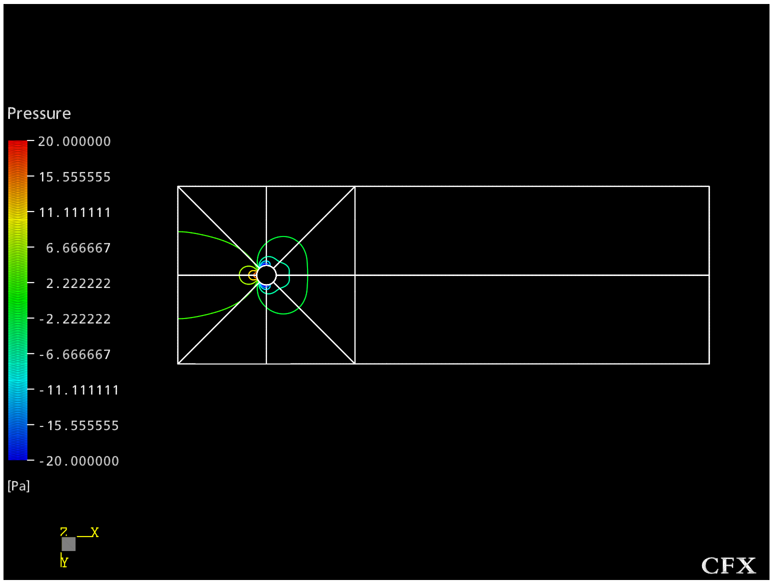

Figure 4.4: Pressure contours for the the k-ε model

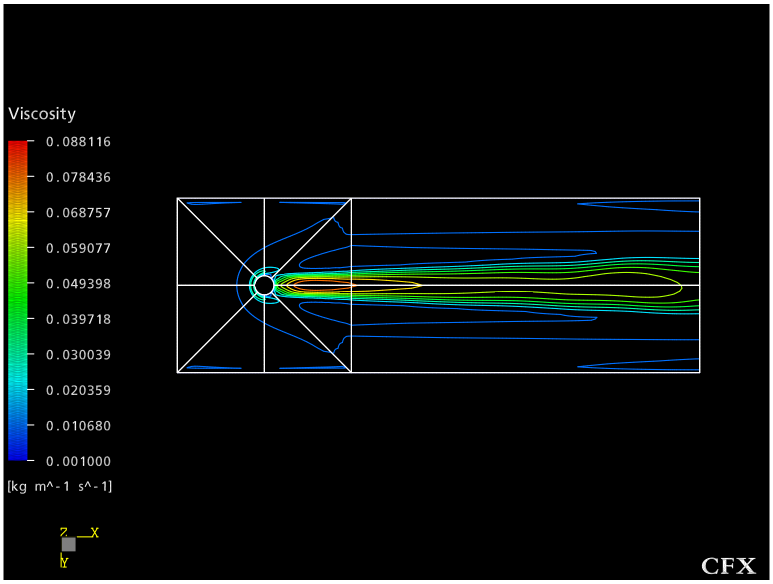

Figure 4.5: Effective viscosity predicted by the the k-ε model

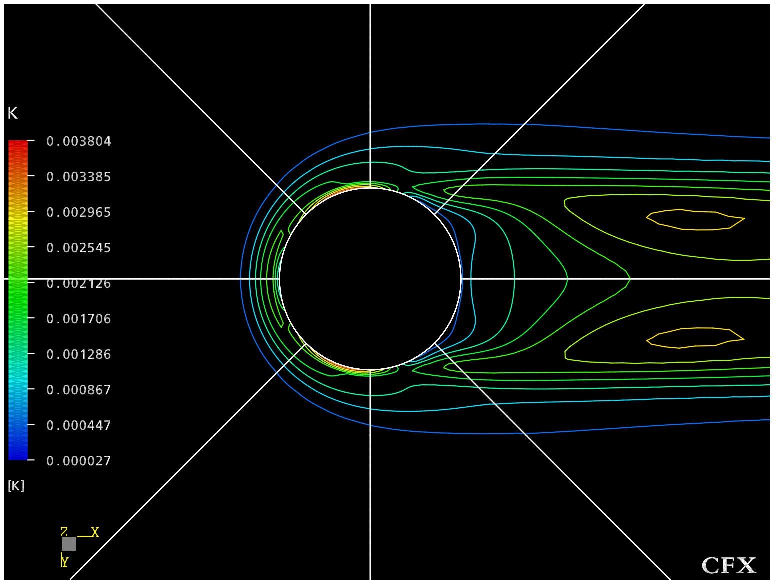

Figure 4.6: Turbulent kinetic energy from the k-ε model

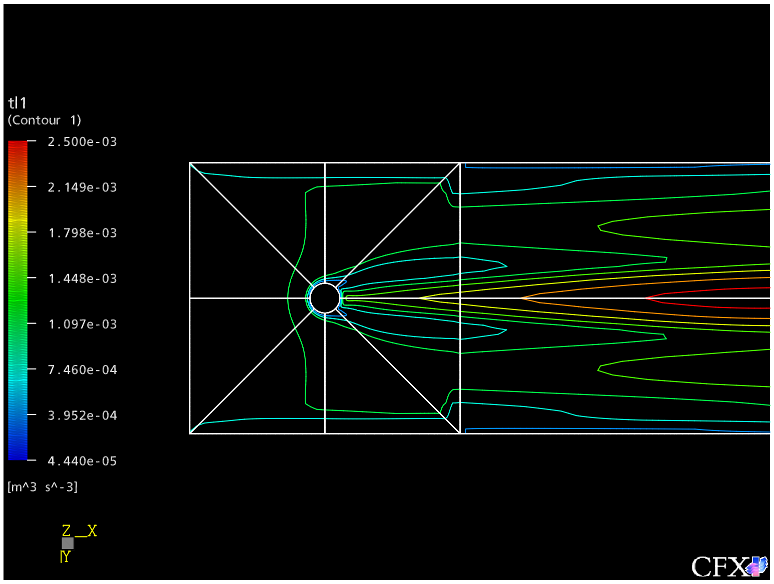

Figure 4.7: Contours of turbulent lengthscale from the k-ε model

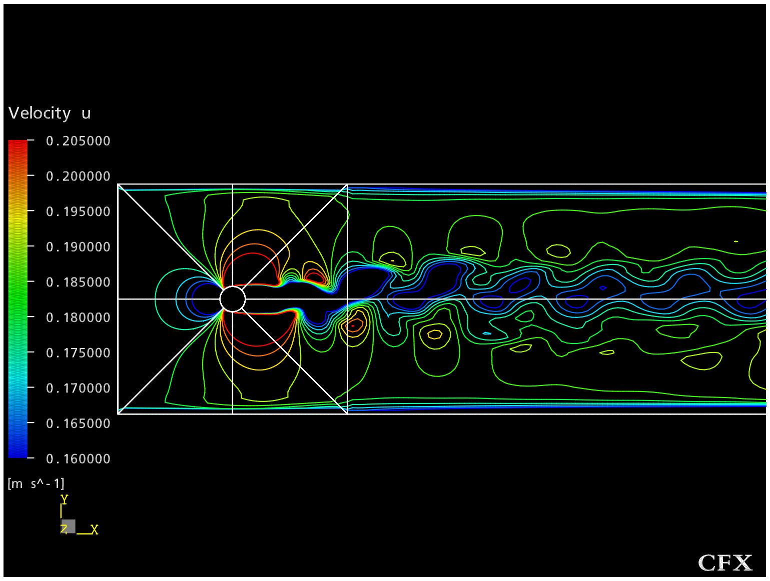

Figure 4.8: Vectors of u-velocity from the Smagorinsky model

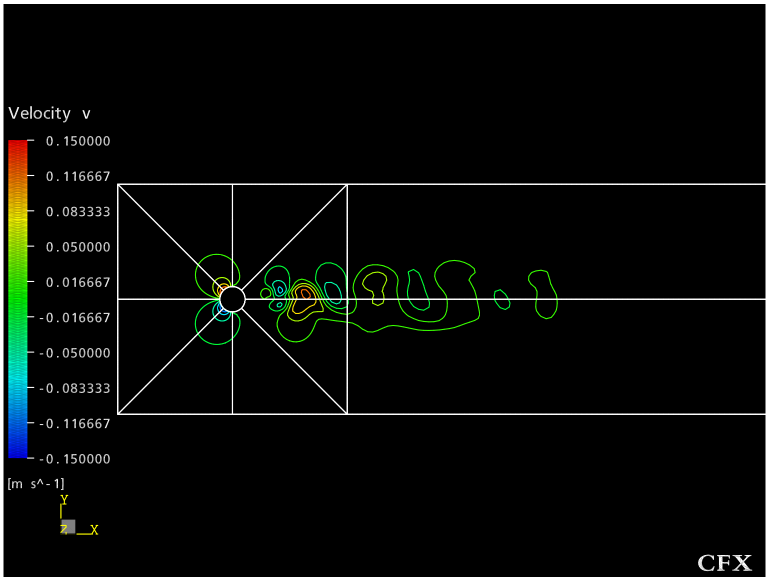

Figure 4.9: Contours of v-velocity from the Smagorinsky model

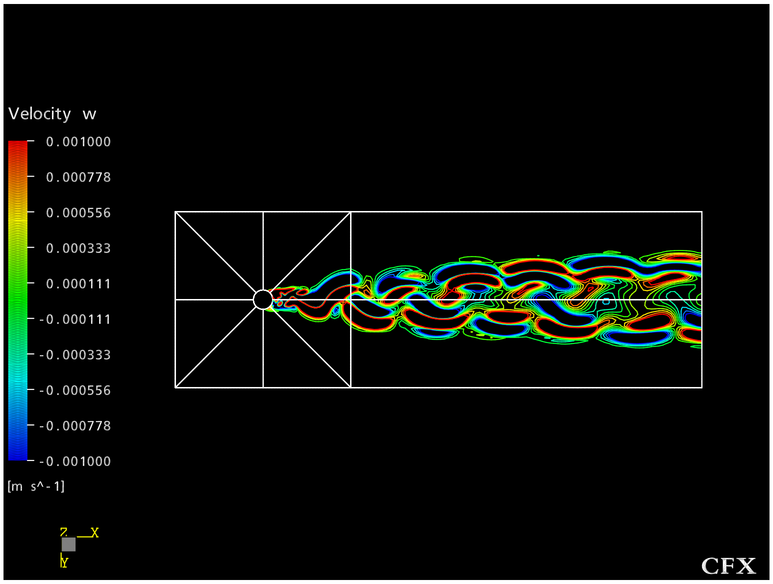

Figure 4.10: Contours of w-velocity from the Smagorinsky model

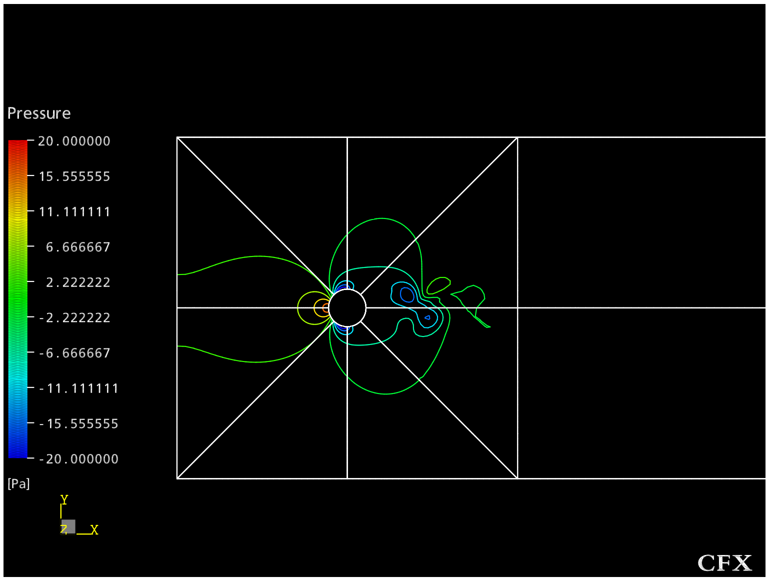

Figure 4.11: Pressure contours for the the Smagorinsky model

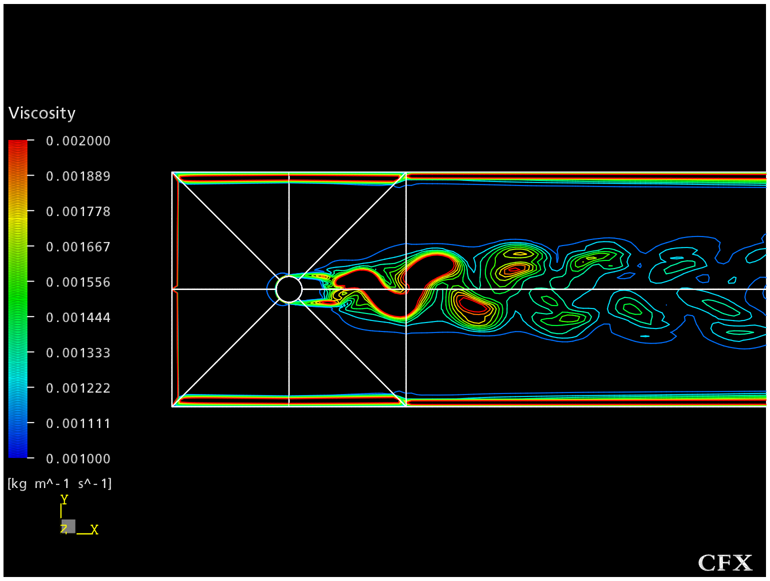

Figure 4.12: Effective viscosity predicted by the the Smagorinsky model

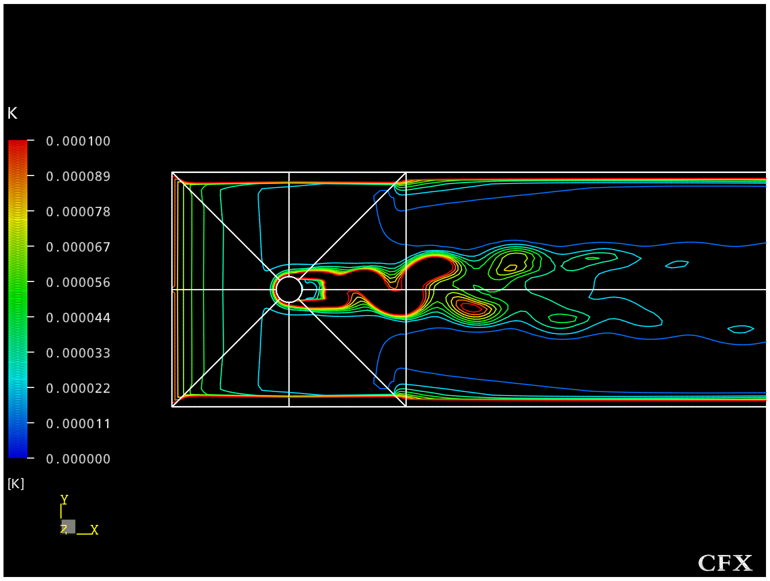

Figure 4.13: Turbulent kinetic energy from the Smagorinsky model

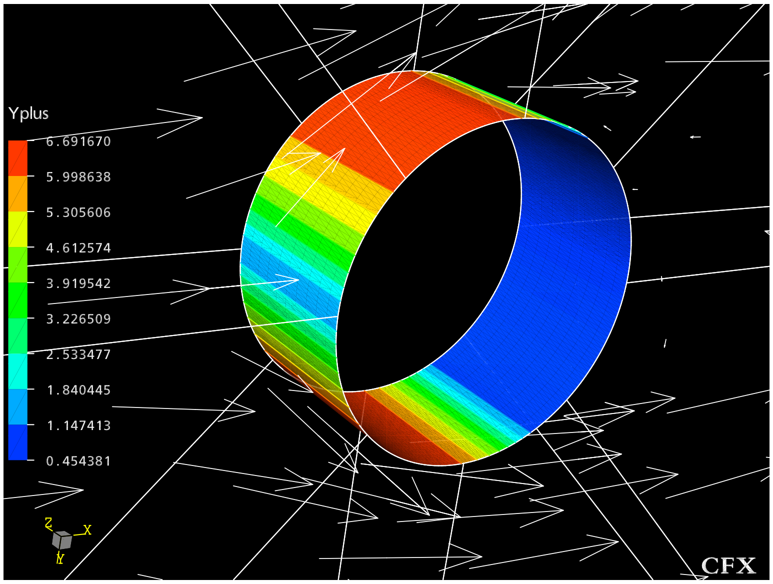

Figure 4.14: Values of y-plus on the cylinder predicted by the Smagorinsky model

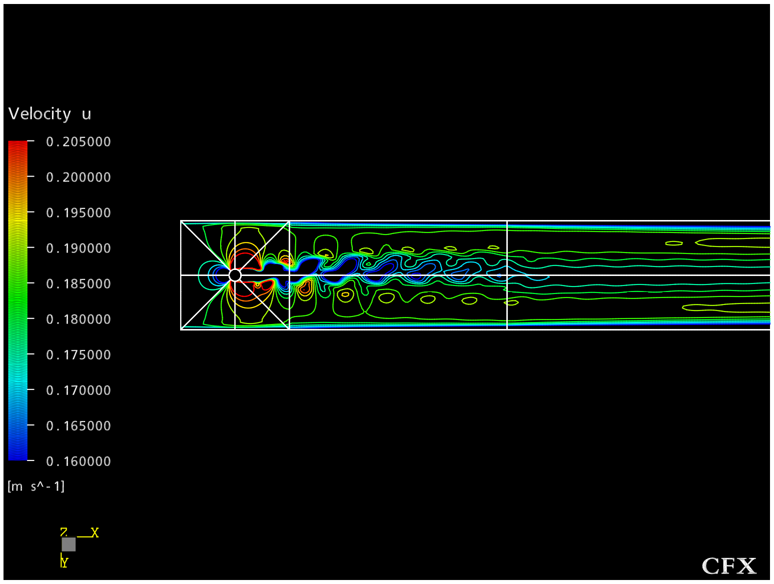

Figure 4.15: Contours of u-velocity from the av3 method

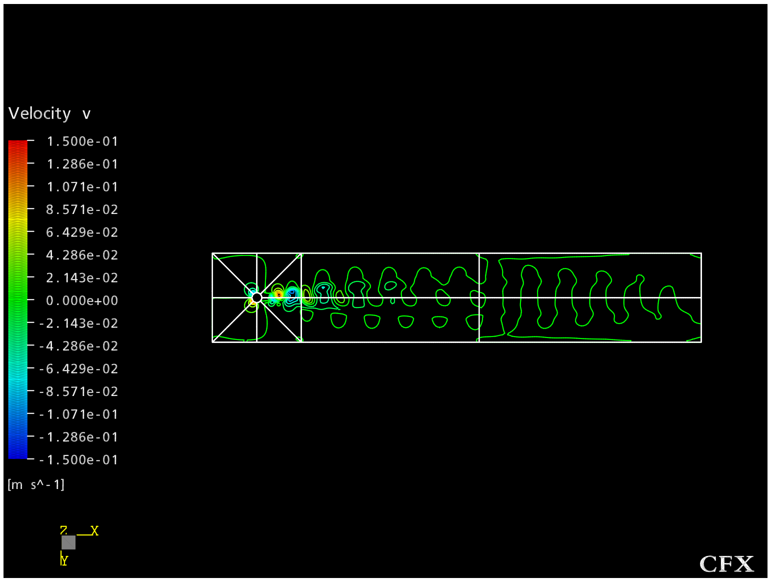

Figure 4.16: Contours of v-velocity from the av3 method

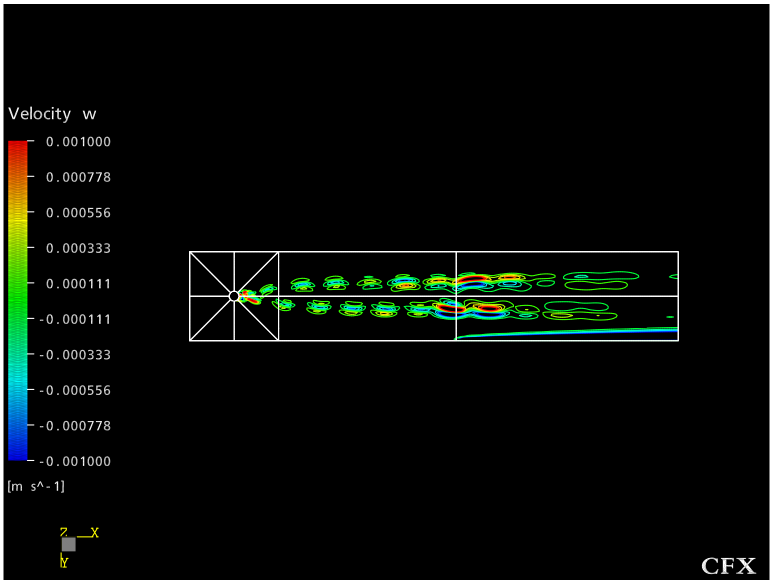

Figure 4.17: Contours of w-velocity from the av3 method

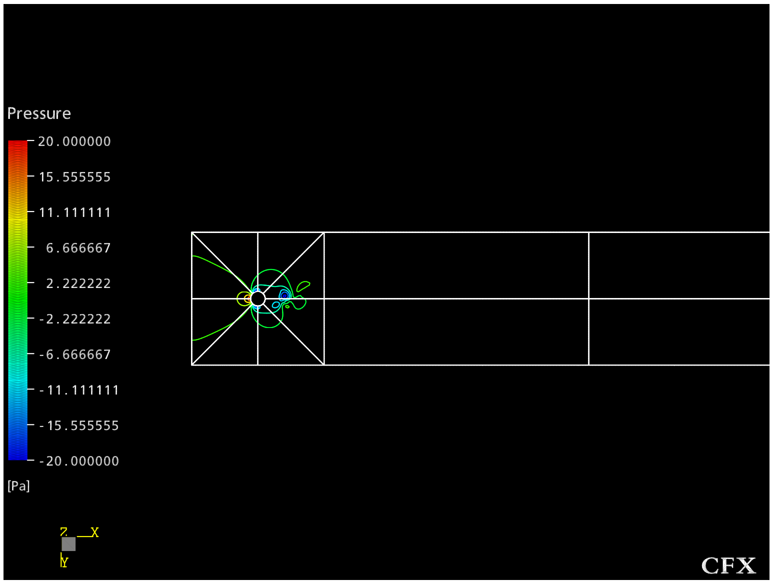

Figure 4.18: Contours of pressure from the av3 method

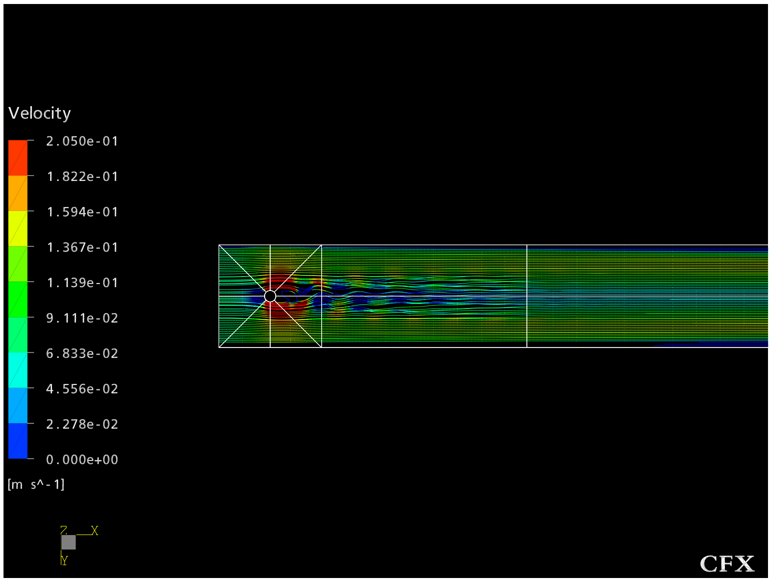

Figure 4.19: Streamlines of the av3 simulation

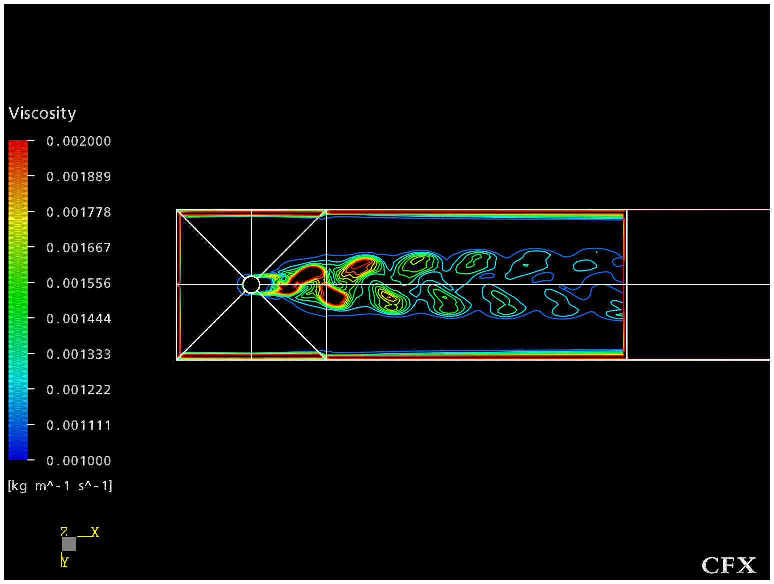

Figure 4.20: Effective viscosity using the av3 method

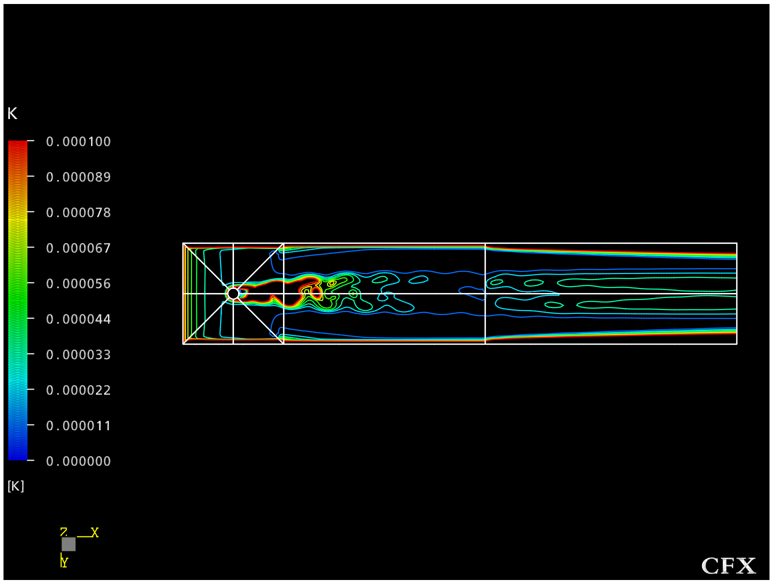

Figure 4.21: Turbulent kinetic energy from the av3 method

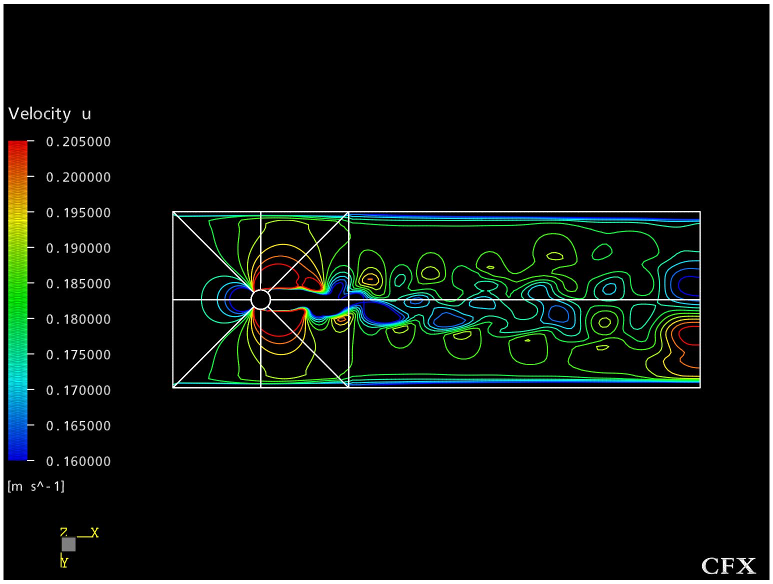

Figure 4.22: Contours of u-velocity from the hybrid method

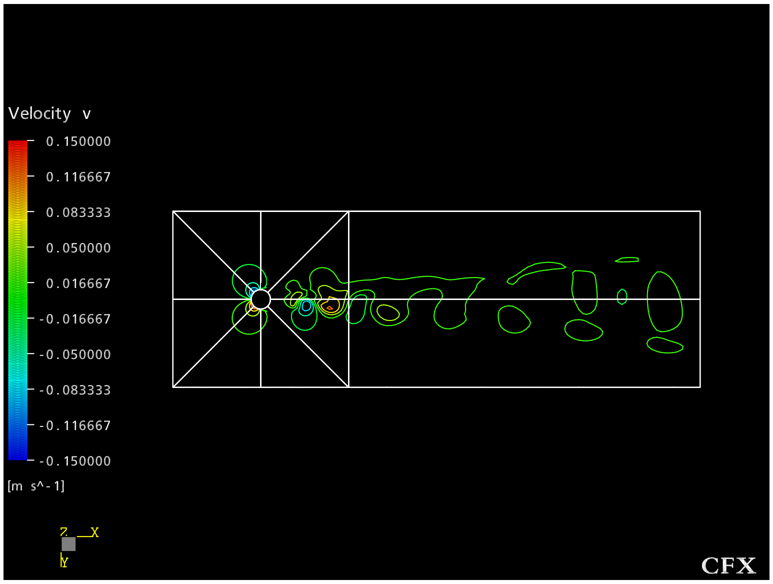

Figure 4.23: Contours of v-velocity from the hybrid method

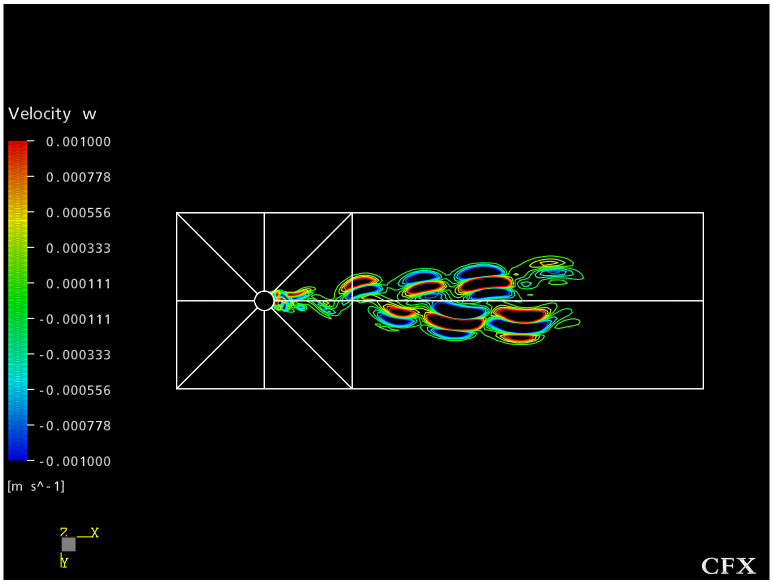

Figure 4.24: Contours of w-velocity from the hybrid method

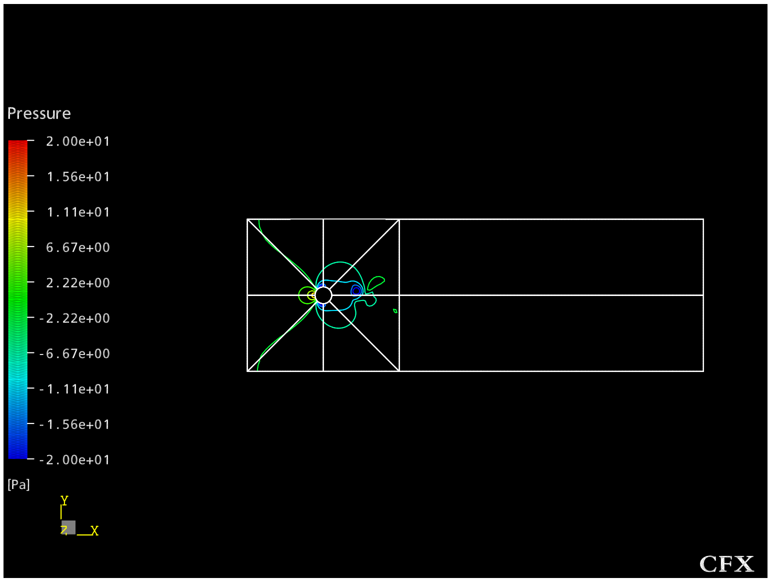

Figure 4.25: Pressure contours predicted by the hybrid model

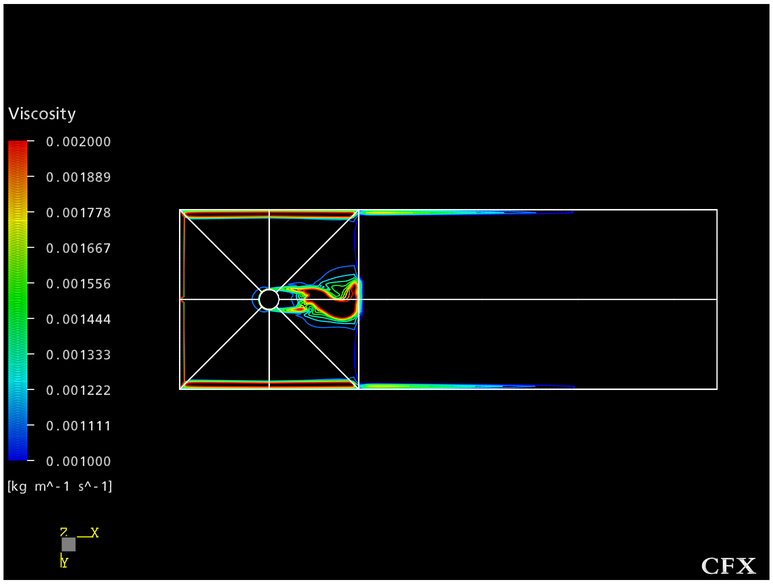

Figure 4.26: Effective viscosity using the hybrid method

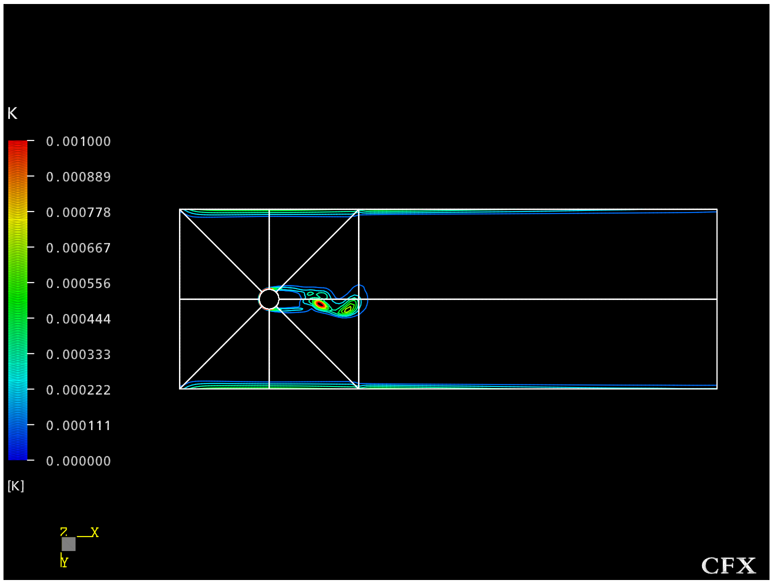

Figure 4.27: Turbulent kinetic energy from the hybrid method

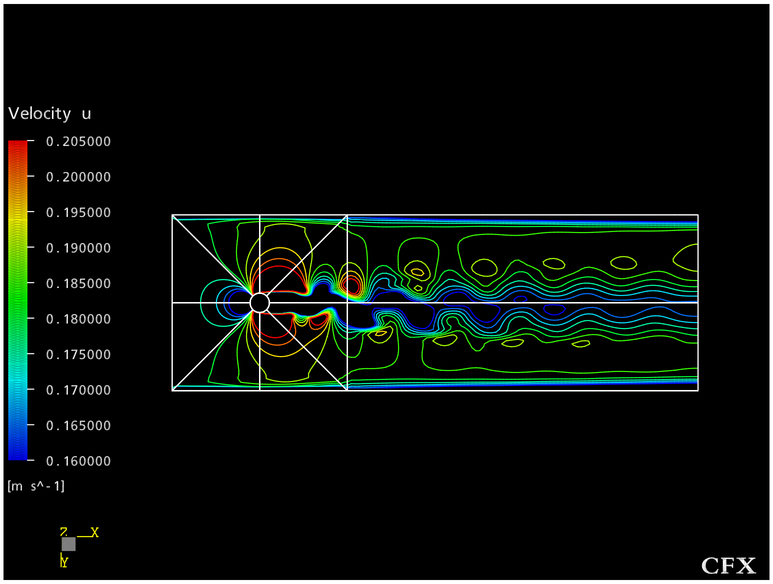

Figure 4.28: Contours of u-velocity from the hybrid method without the viscosity modifier

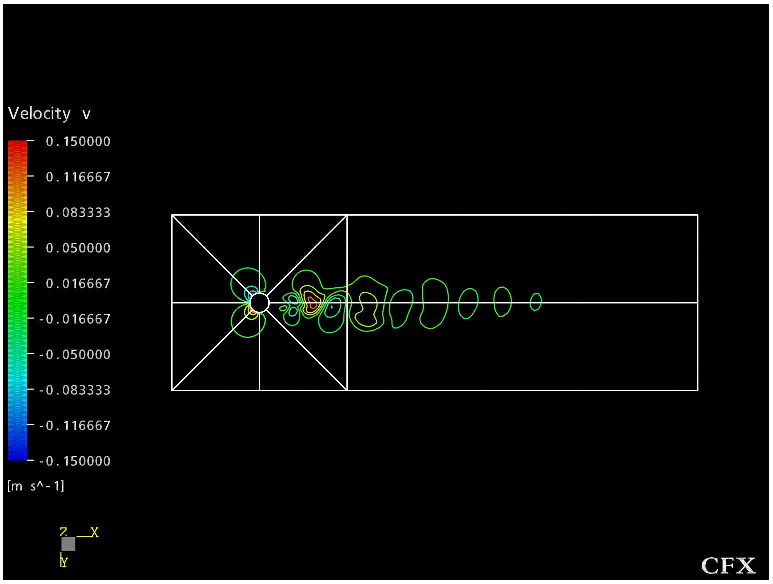

Figure 4.29: Contours of v-velocity from the hybrid method without the viscosity modifier

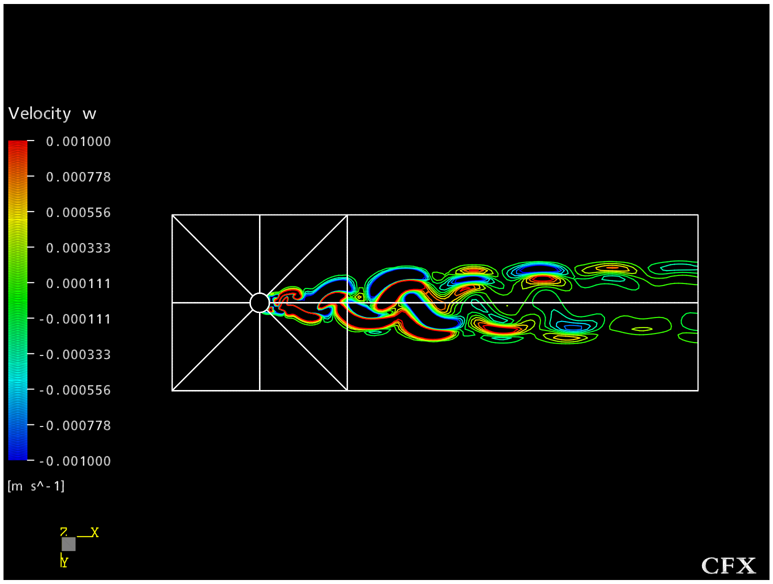

Figure 4.30: Contours of w-velocity from the hybrid method without the viscosity modifier

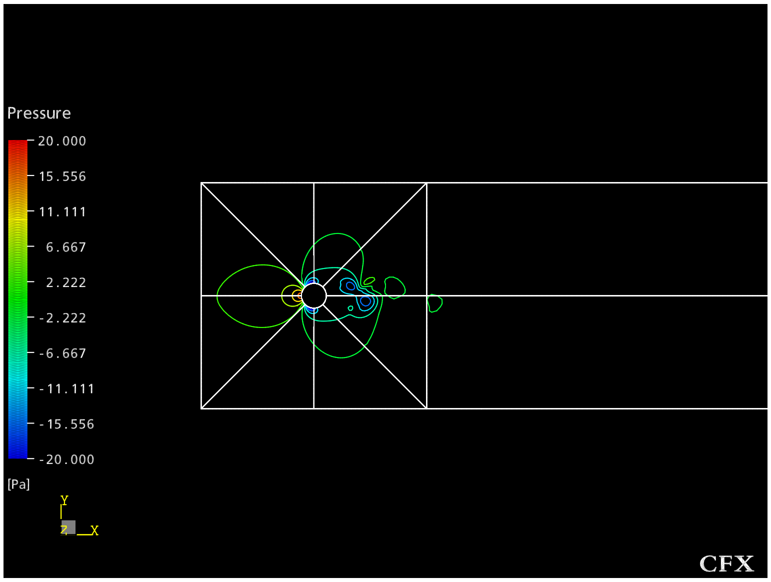

Figure 4.31: Pressure contours predicted by the hybrid model without the viscosity modifier

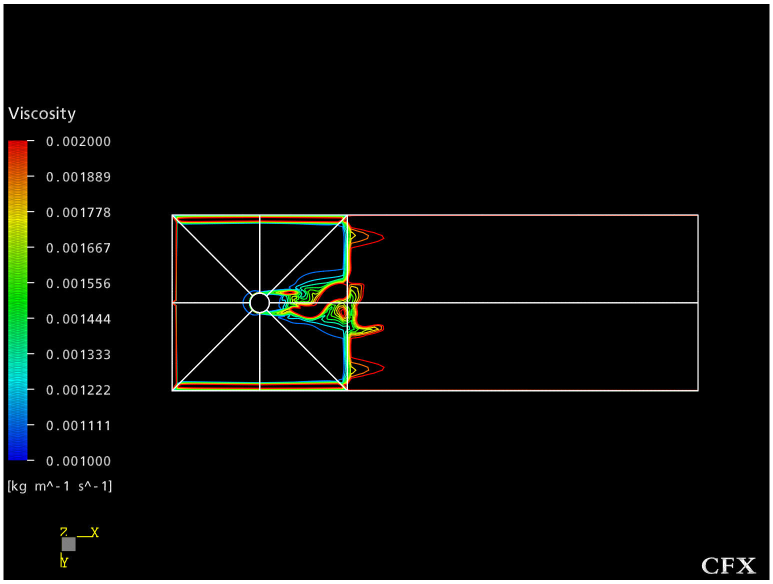

Figure 4.32: Effective viscosity using the hybrid method without the viscosity modifier

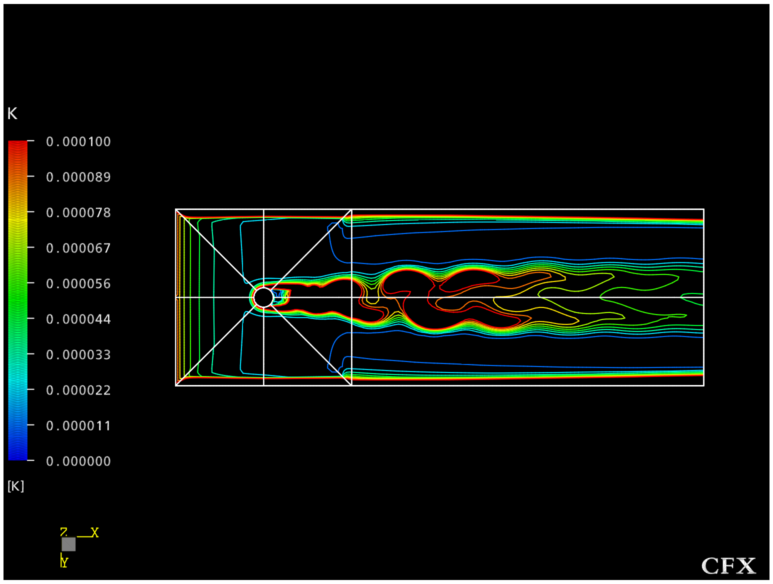

Figure 4.33: Turbulent kinetic energy from the hybrid method without the viscosity modifier

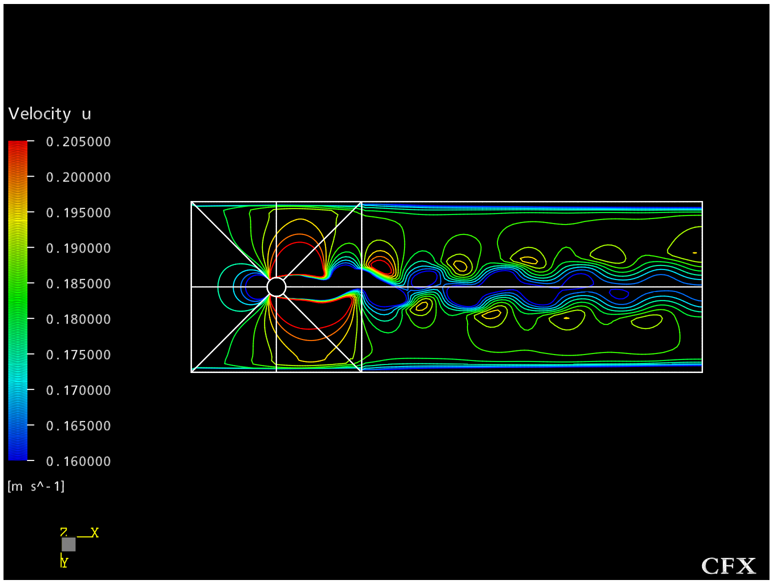

Figure 4.34: Contours of u-velocity from the Newhybrid method

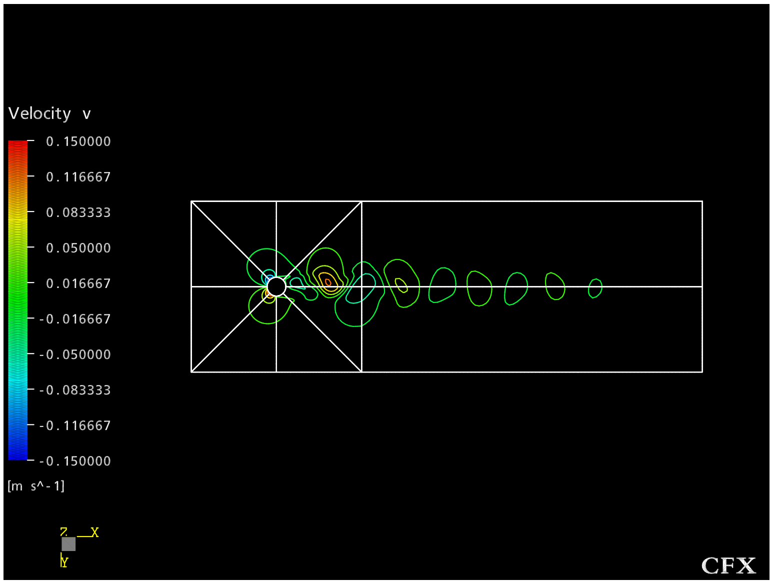

Figure 4.35: Contours of v-velocity from the Newhybrid method

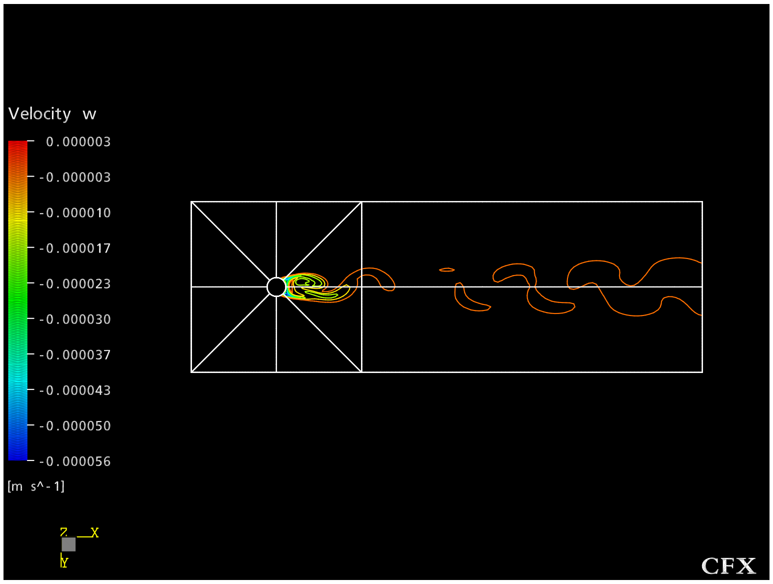

Figure 4.36: Contours of w-velocity from the Newhybrid method

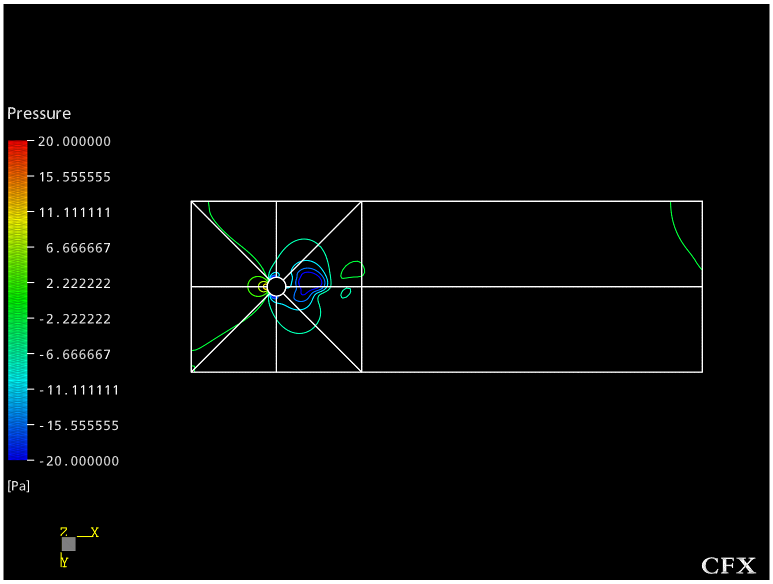

Figure 4.37: Pressure contours predicted by the Newhybrid model

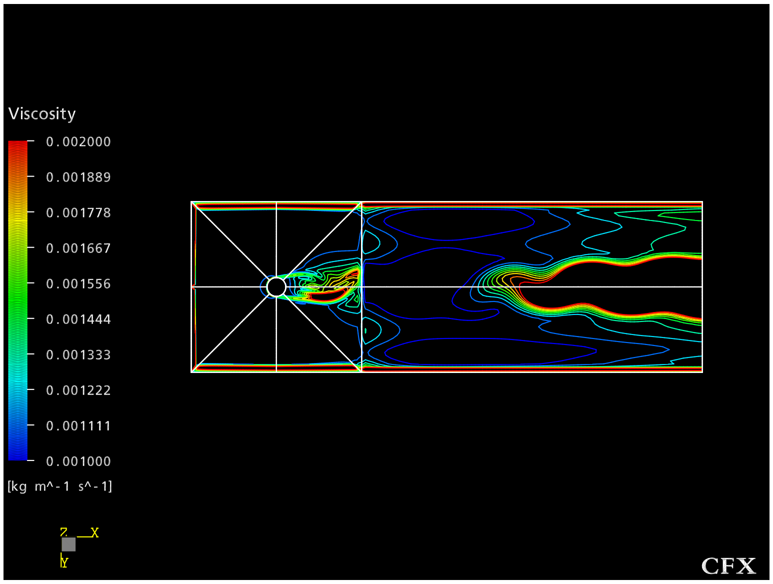

Figure 4.38: Effective viscosity using the Newhybrid method

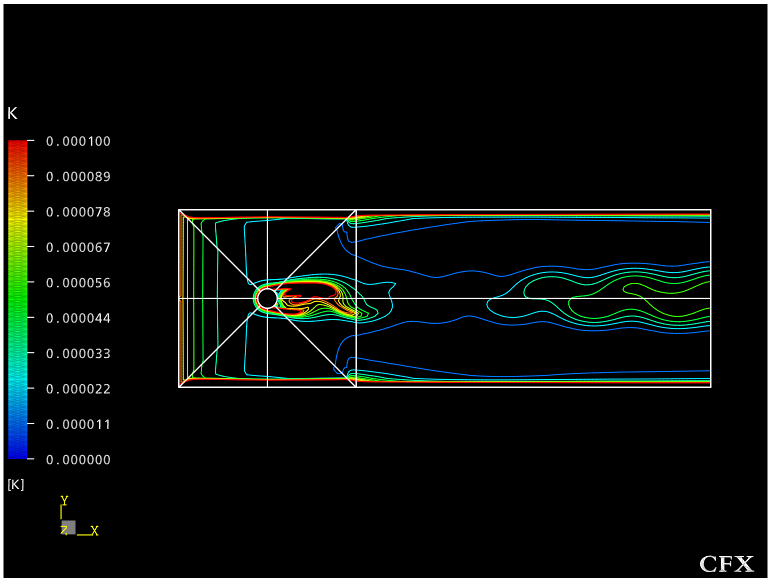

Figure 4.39: Turbulent kinetic energy from the Newhybrid method

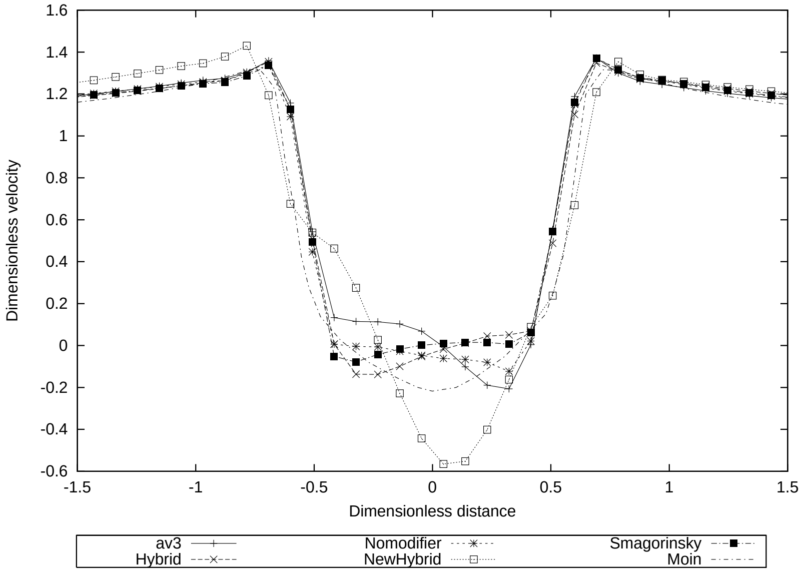

Figure 4.40: Mean u-velocity 1.06 diameters behind the cylinder calculated by averaging over 1 oscillation

4.2 Mean velocity downstream of the cylinder

In order to allow for a closer comparison with the work of Kravchenko and Moin [49] mean velocities for the three component directions have been plotted at the same distances downstream of the cylinder as in the mentioned paper. These being 1.06, 1.54 4, 5 and 10 diameters downstream. For each case the profiles of the hybrid models are compared with the Smagorinsky model that was executed on the same grid, and the Kravchenko and Moin results. These again are taken midway through the domain in the z-direction. The distances have been normalised with respect to the cylinder diameter, and the velocities with respect to the inlet velocity.

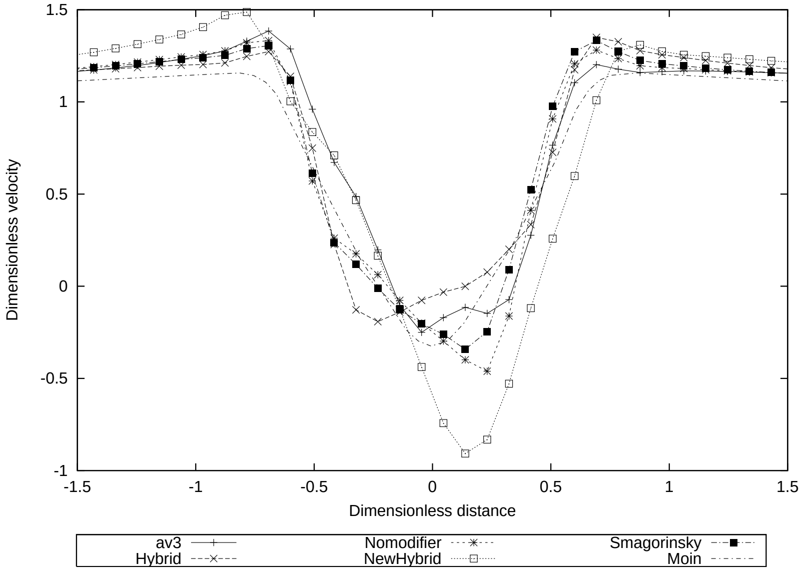

Figure 4.41: Mean u-velocity 1.54 diameters behind the cylinder calculated using the theoretical EWMA approach

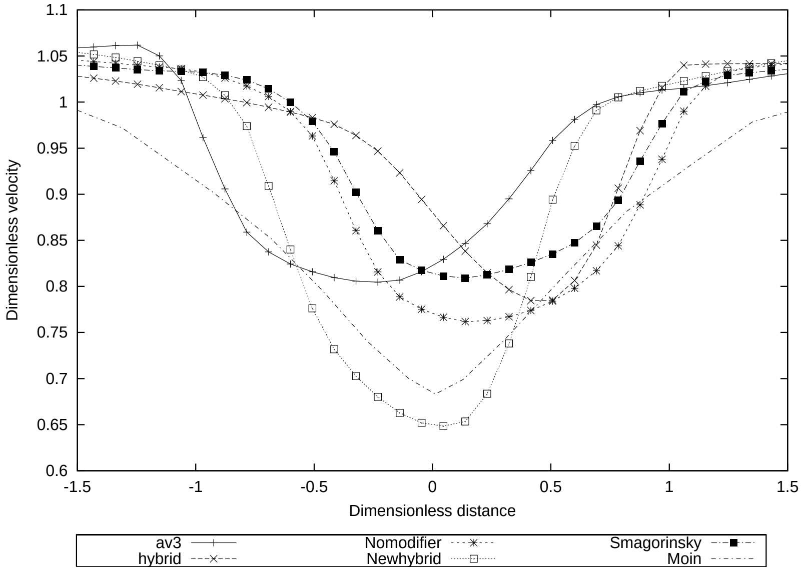

Figure 4.42: Mean u-velocity 4 diameters behind the cylinder calculated using the ‘dump’ files

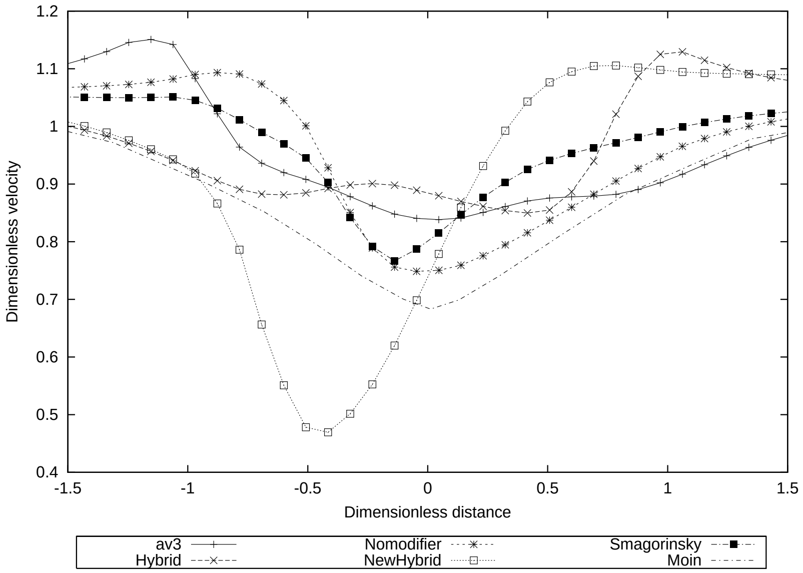

Figure 4.43: Mean u-velocity 4 diameters behind the cylinder calculated using the theoretical EWMA approach

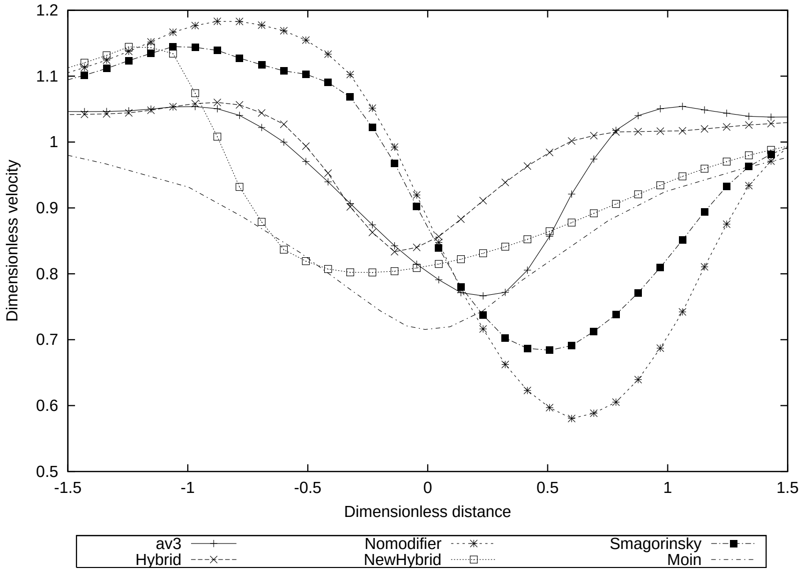

Figure 4.44: Mean u-velocity 5 diameters behind the cylinder calculated by averaging over 1 oscillation

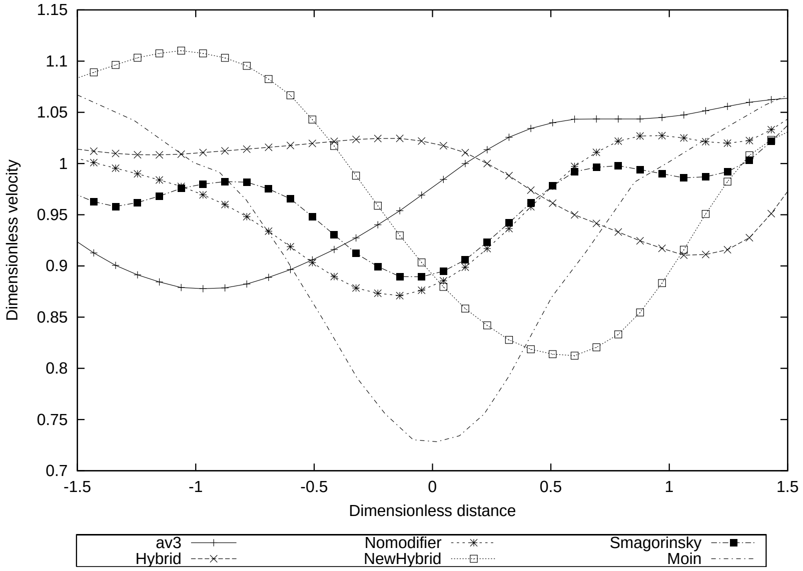

Figure 4.45: Mean u-velocity 10 diameters behind the cylinder calculated by averaging over 1 oscillation

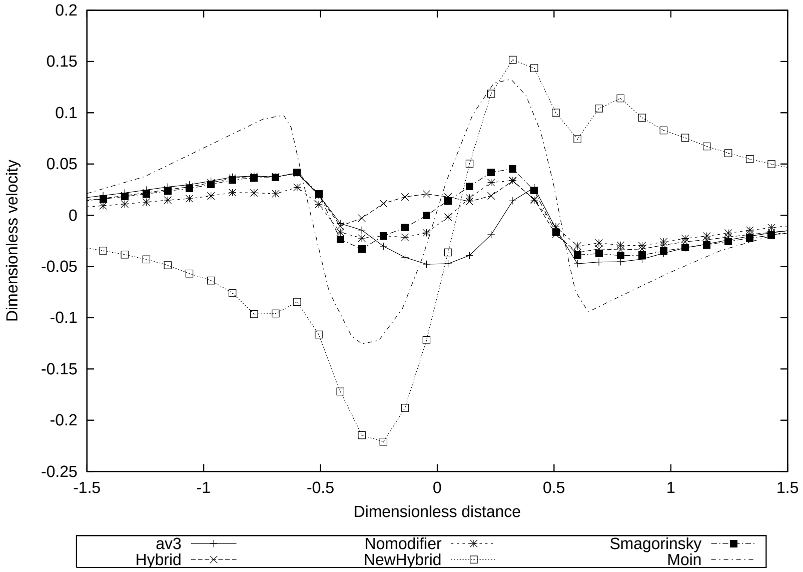

Figure 4.46: Mean v-velocity 1.06 diameters downstream of the cylinder calculated using the 'dump' files

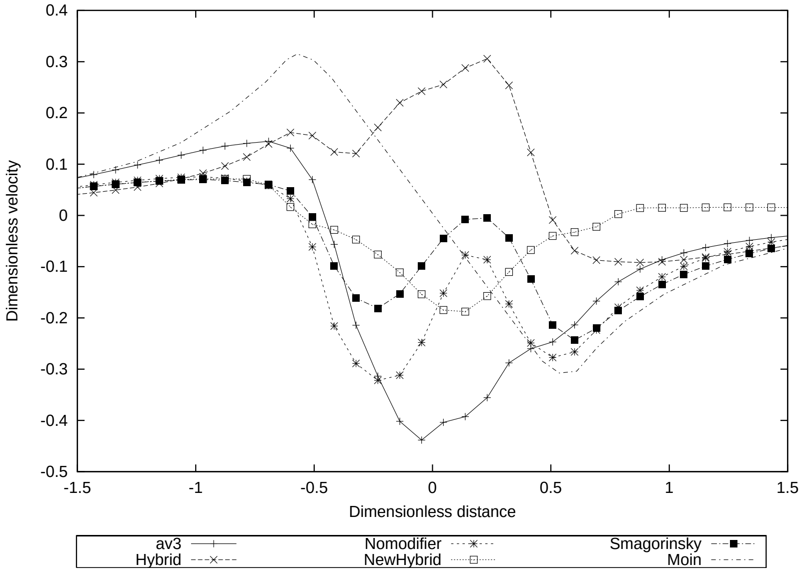

Figure 4.47: Mean v-velocity 1.54 diameters behind the cylinder calculated using the theoretical EWMA approach

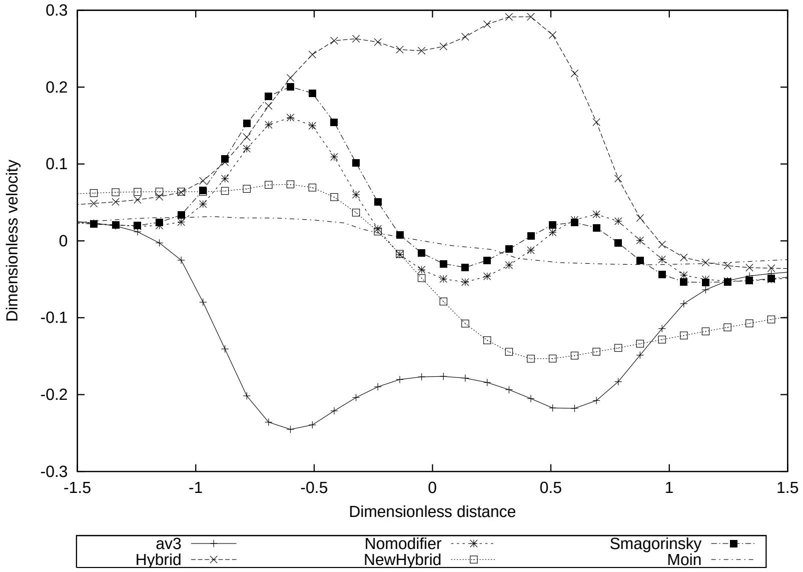

Figure 4.48: Mean v-velocity 4 diameters behind the cylinder calculated using the 'dump' files

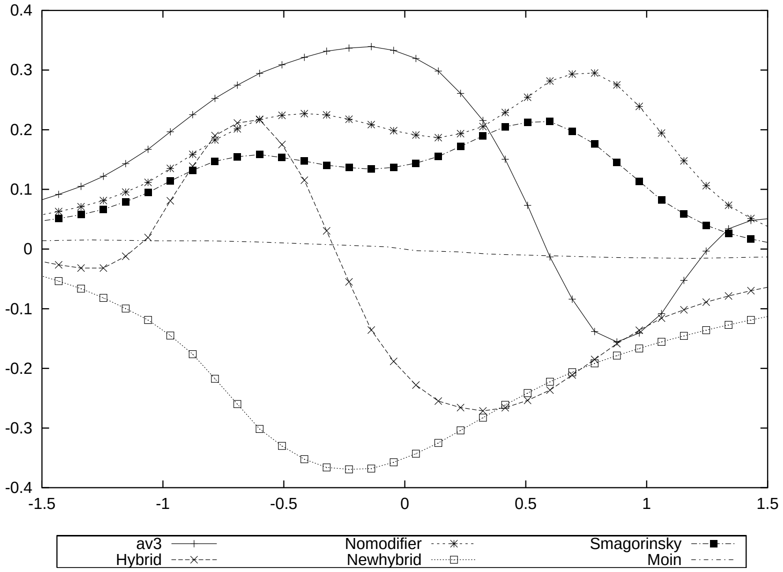

Figure 4.49: Mean v-velocity 5 diameters behind the cylinder calculated by averaging over 1 oscillation

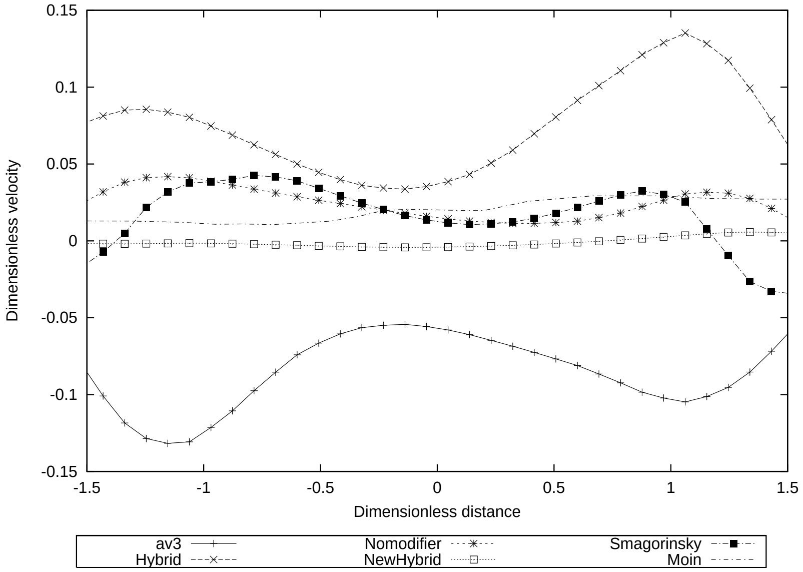

Figure 4.50: Mean v-velocity 10 diameters downstream of the cylinder calculated using the ‘dump’ files

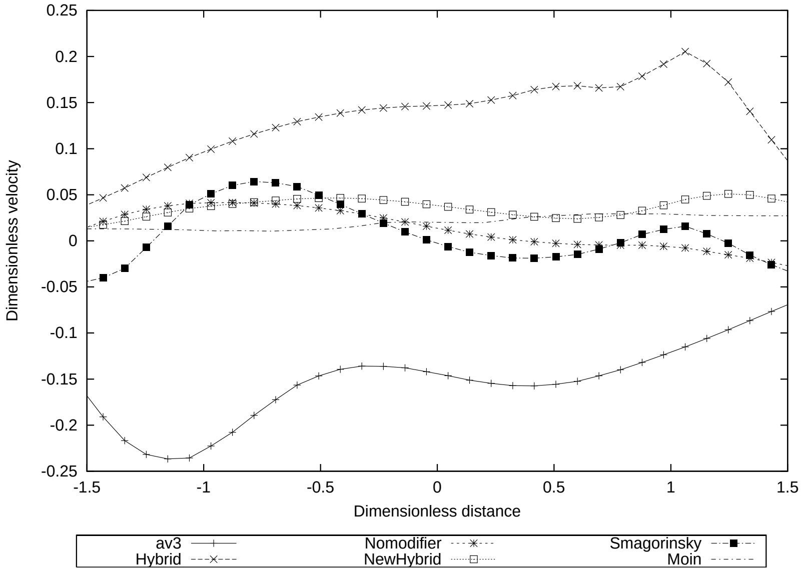

Figure 4.51: Mean v-velocity 10 diameters downstream of the cylinder calculated by averaging over 1 oscillation

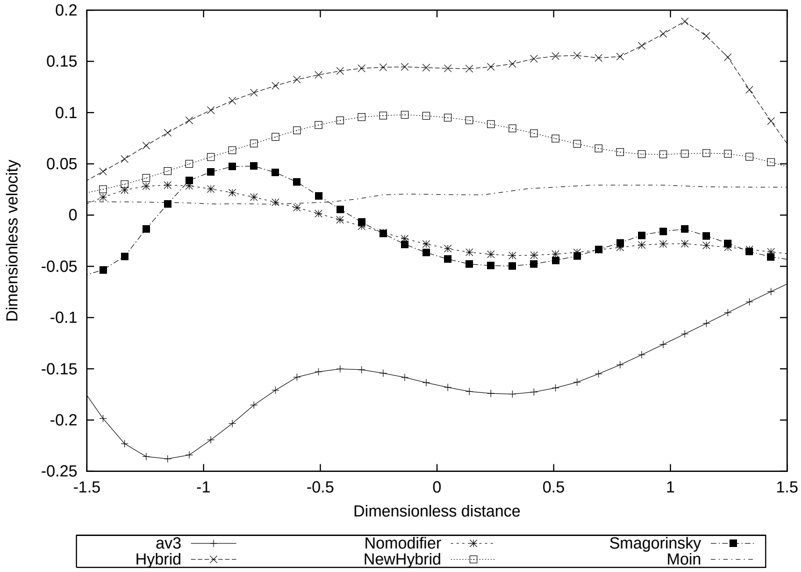

Figure 4.52: Mean v-velocity 10 diameters downstream of the cylinder calculated using the theoretical EWMA approach

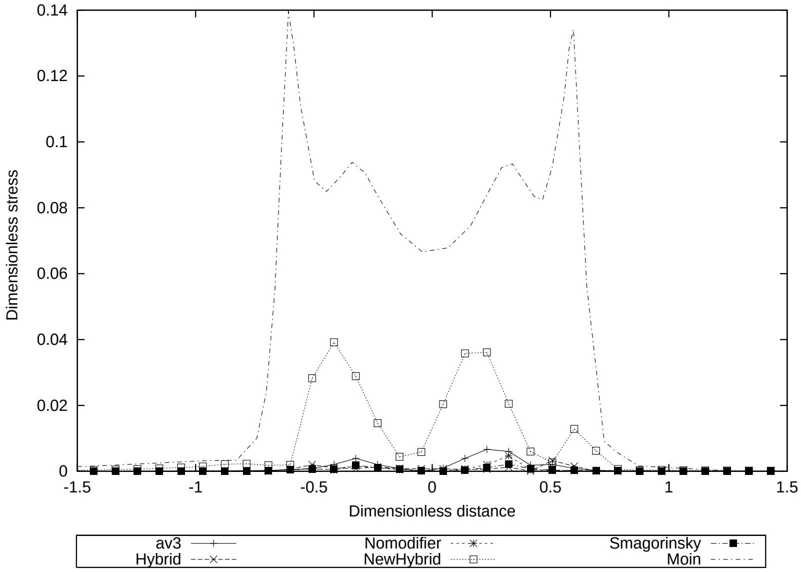

Figure 4.53: Mean u'u' stress 1.06 diameters downstream of the cylinder calculated using the 'dump' files

4.3 Turbulence predictions

Graphs are presented to show the turbulence predictions at the same distances downstream as for the velocity profiles. Again the predictions have been compared with the Smagorinsky model on this grid and the LES results of Kravchenko and Moin. The calculation of the fluctuating velocities was calculated by comparing the mean velocity for that model with the instantaneous velocity. The use of the stress means that all of the values will be positive. The $w'w'$ stress are only recorded for the 1.06 and 1.54 diameters by Moin and so have only been compared at these diameters. Turbulence properties are normalised with the square of the inlet velocity. The scales have been chosen to extract the desired features from the results. This varies depending on whether the comparison is between the calculated results and those of Moin, or whether the graph is to highlight differences between the different methodologies used herein.

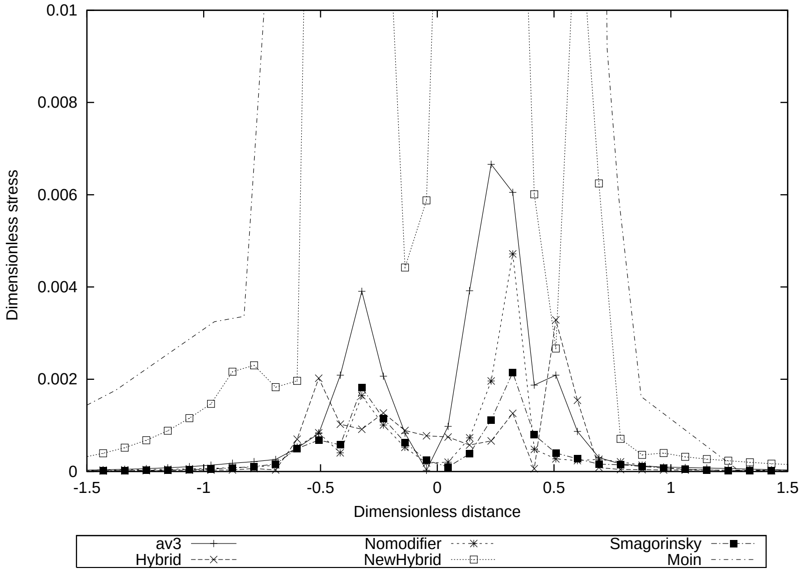

Figure 4.54: Mean u'u' stress 1.06 diameters downstream of the cylinder, calculated using the 'dump' files

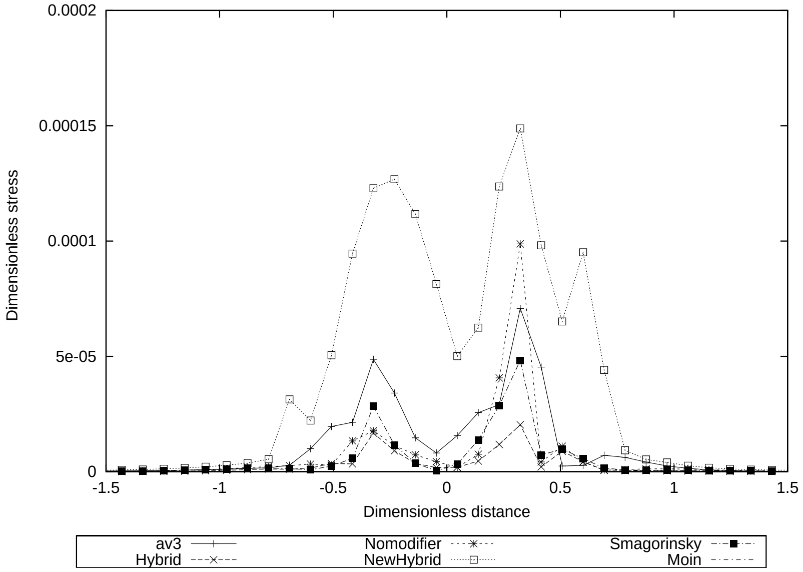

Figure 4.55: Mean u'u' stress 1.06 diameters downstream of the cylinder calculated by averaging over 1 oscillation

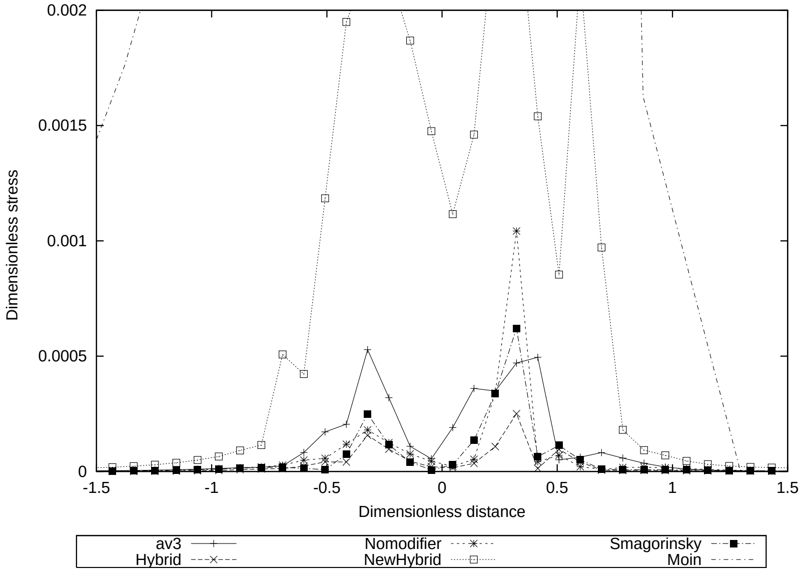

Figure 4.56: Mean u'u' stress 1.06 diameters downstream of the cylinder calculated using the theoretical EWMA approach

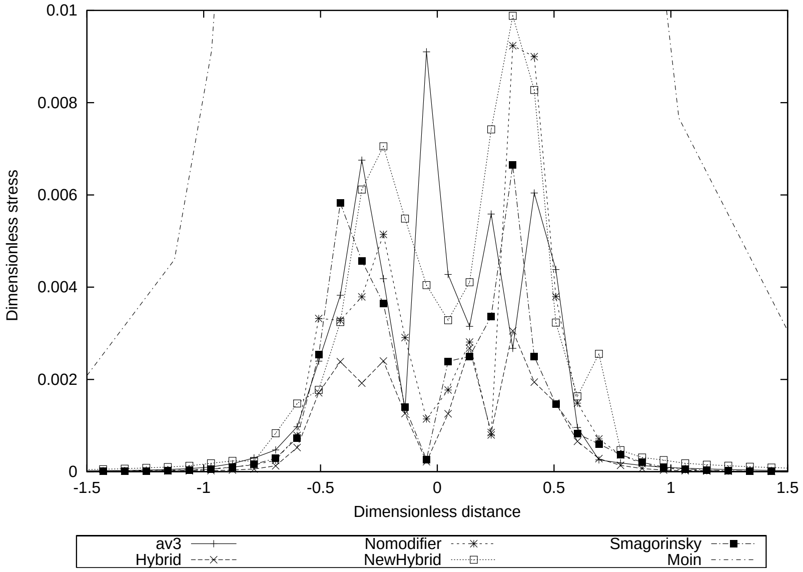

Figure 4.57: Mean u'u' stress 1.54 diameters downstream of the cylinder calculated using the theoretical EWMA approach

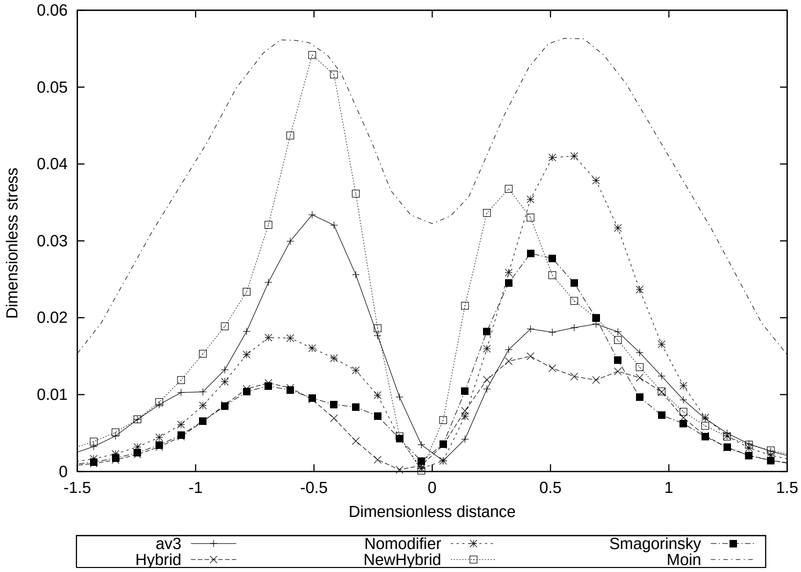

Figure 4.58: Mean u'u' stress 4 diameters downstream of the cylinder calculated using the 'dump' files

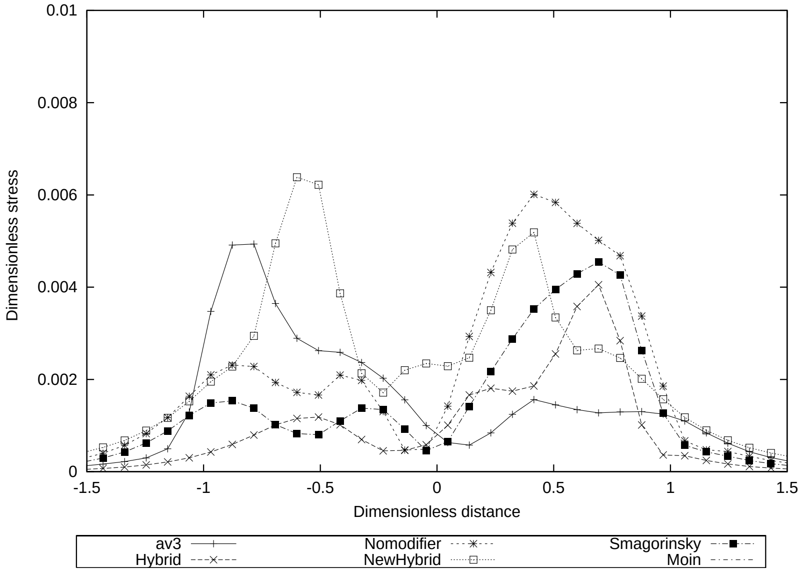

Figure 4.59: Mean u'u' stress 4 diameters downstream of the cylinder calculated using the theoretical EWMA approach

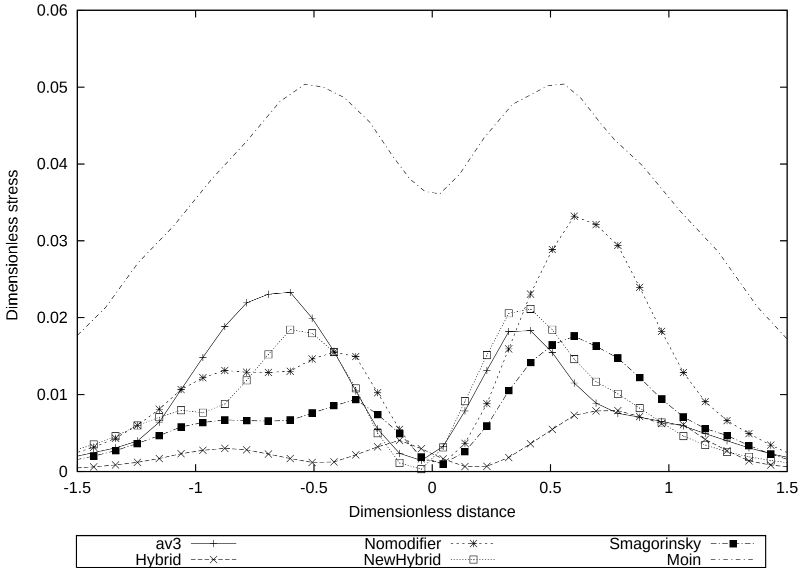

Figure 4.60: Mean u'u' stress 5 diameters downstream of the cylinder calculated using the 'dump' files

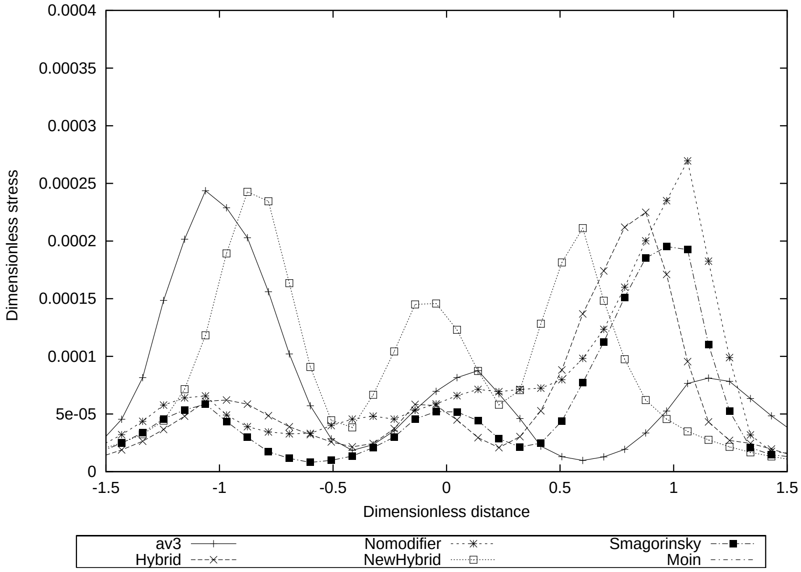

Figure 4.61: Mean u'u' stress 5 diameters downstream of the cylinder calculated by averaging over 1 oscillation

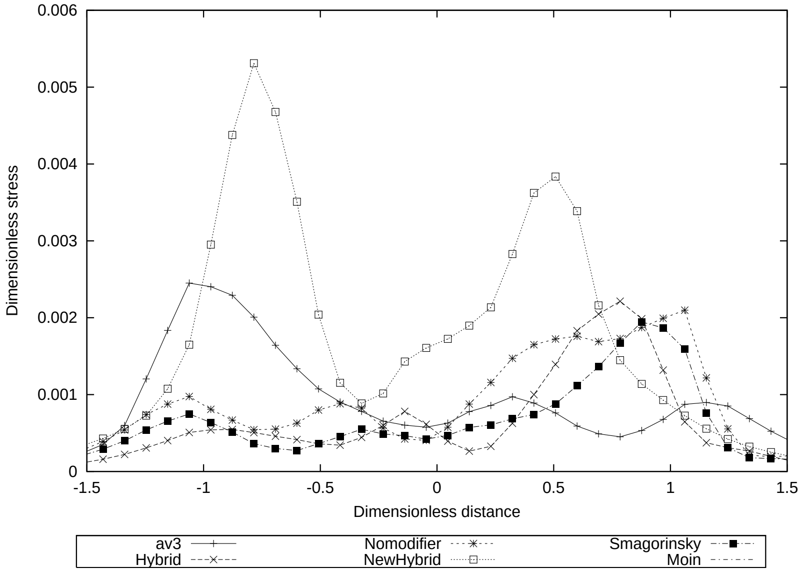

Figure 4.62: Mean u'u' stress 5 diameters downstream of the cylinder calculated using the theoretical EWMA approach

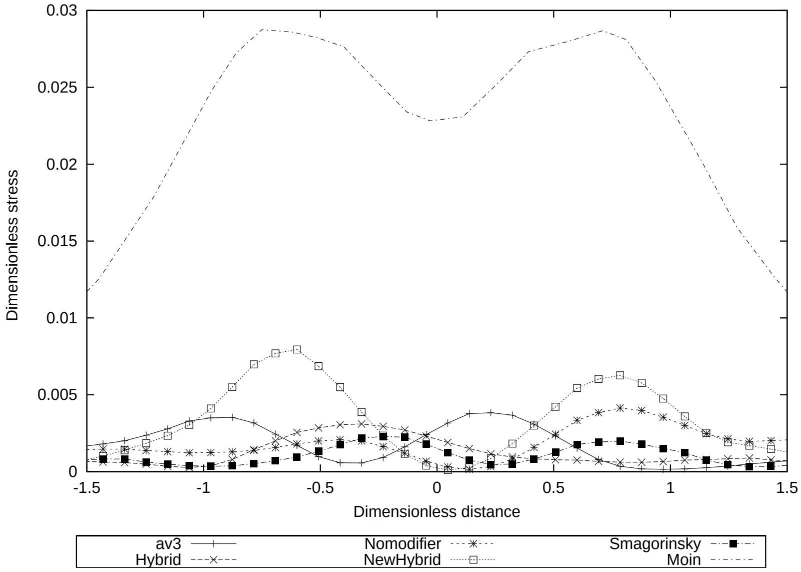

Figure 4.63: Mean u'u' stress 10 diameters downstream of the cylinder calculated using the 'dump' files

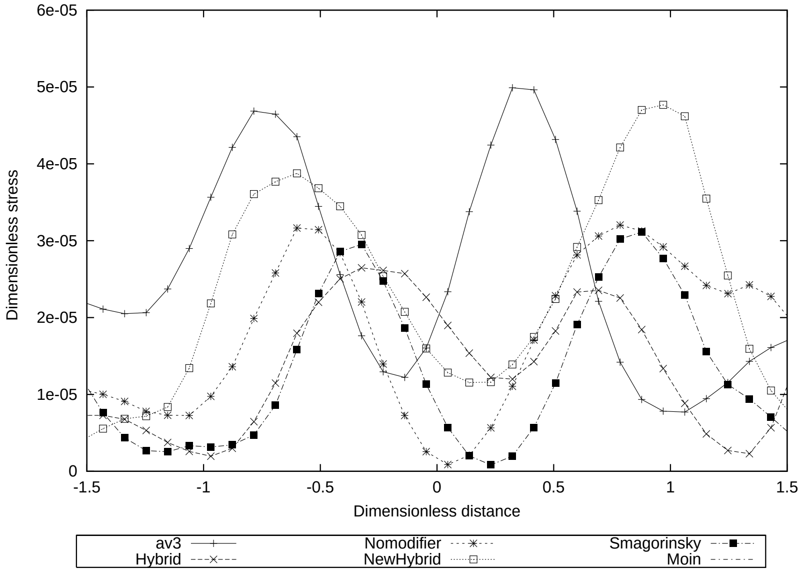

Figure 4.64: Mean u'u' stress 10 diameters downstream of the cylinder calculated by averaging over 1 oscillation

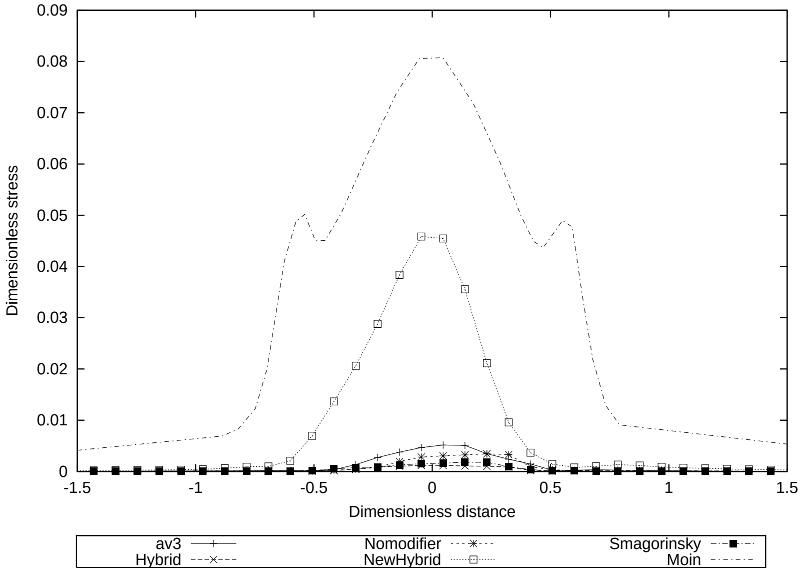

Figure 4.65: Mean v'v' stress 1.06 diameters downstream of the cylinder calculated using the 'dump' files

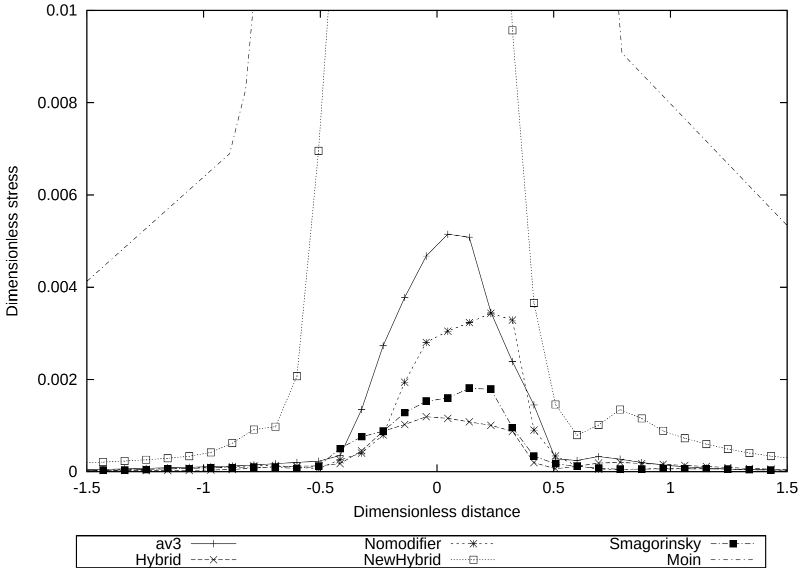

Figure 4.66: Mean v'v' stress 1.06 diameters downstream of the cylinder calculated using the 'dump' files

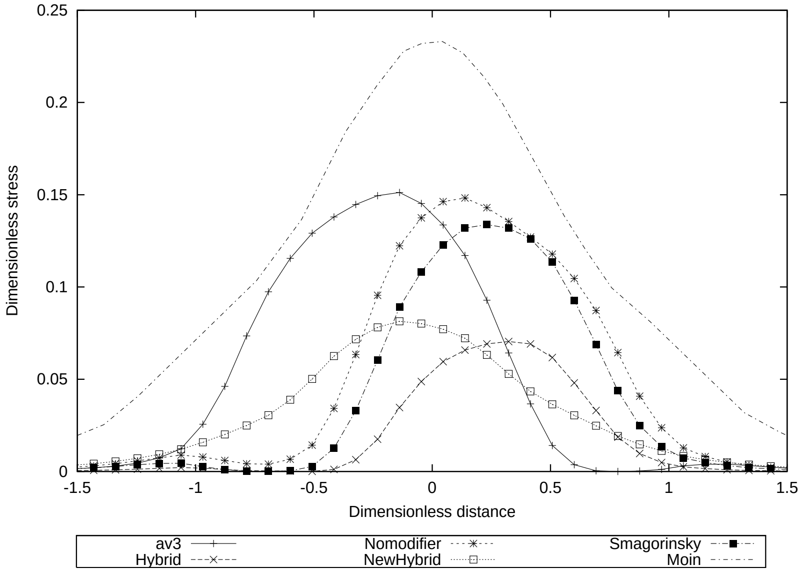

Figure 4.67: Mean v'v' stress 4 diameters downstream of the cylinder calculated using the 'dump' files

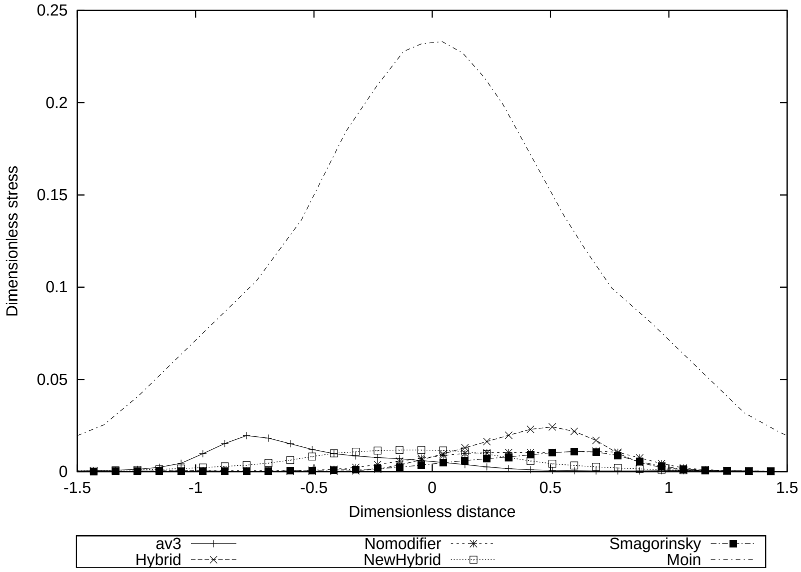

Figure 4.68: Mean v'v' stress 4 diameters downstream of the cylinder calculated using the theoretical EWMA approach

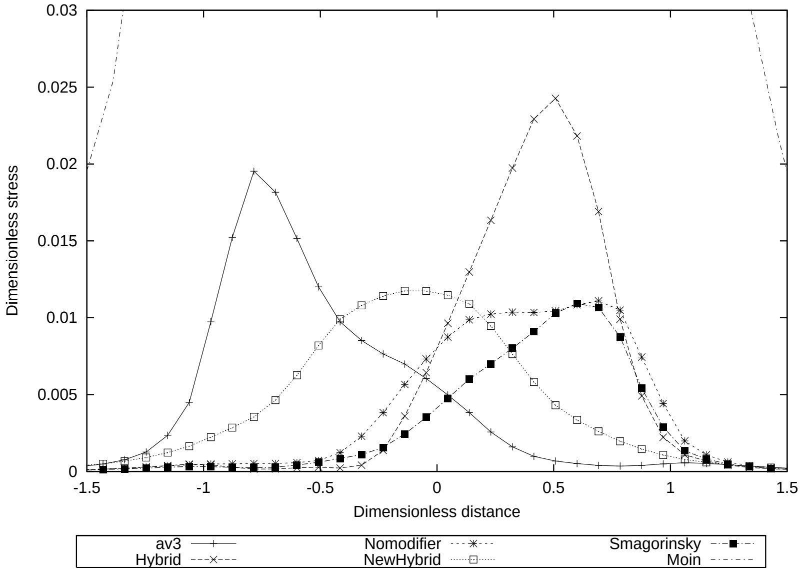

Figure 4.69: Mean v'v' stress 4 diameters downstream of the cylinder calculated using the theoretical EWMA approach

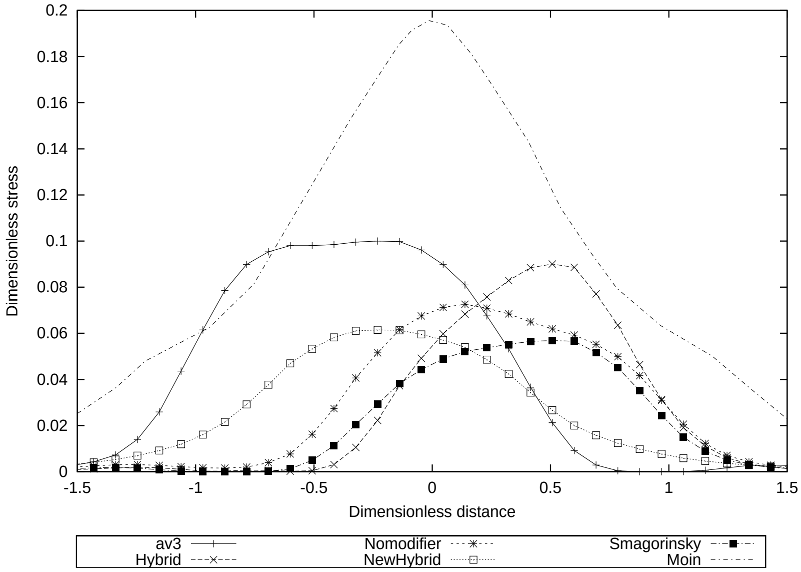

Figure 4.70: Mean v'v' stress 5 diameters downstream of the cylinder calculated using the ‘dump’ files

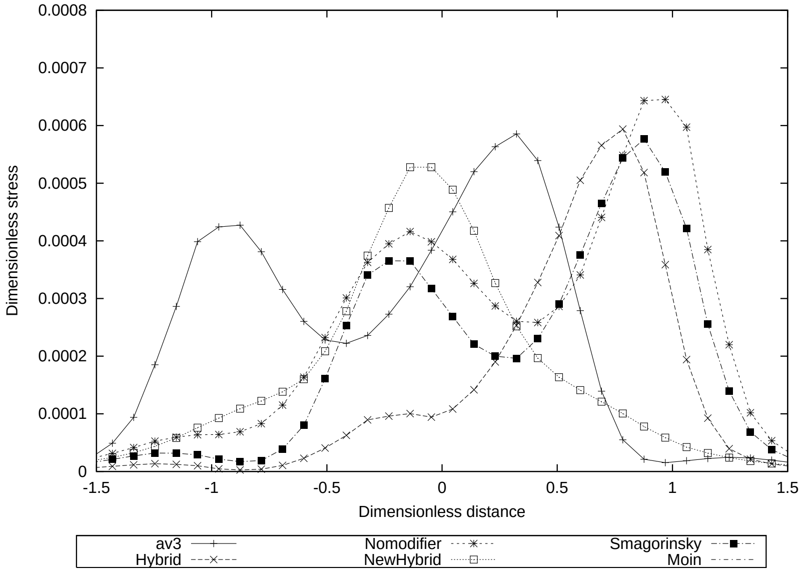

Figure 4.71: Mean v'v' stress 5 diameters downstream of the cylinder calculated by averaging over 1 oscillation

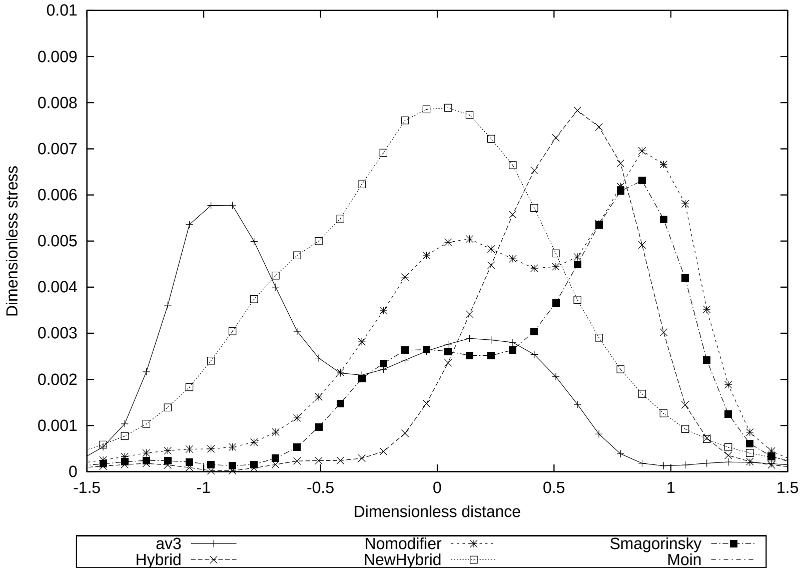

Figure 4.72: Mean v'v' stress 5 diameters downstream of the cylinder calculated using the theoretical EWMA approach

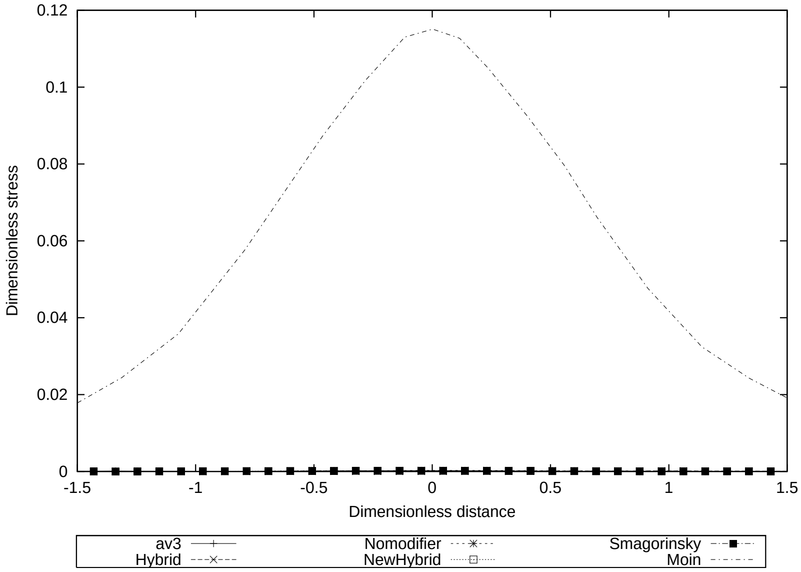

Figure 4.73: Mean v'v' stress 10 diameters downstream of the cylinder calculated by averaging over 1 oscillation

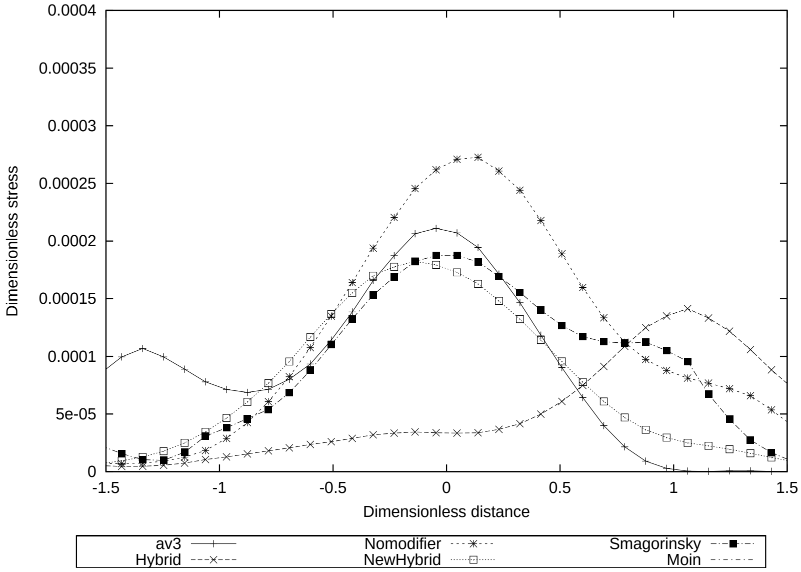

Figure 4.74: Mean v'v' stress 10 diameters downstream of the cylinder calculated by averaging over 1 oscillation

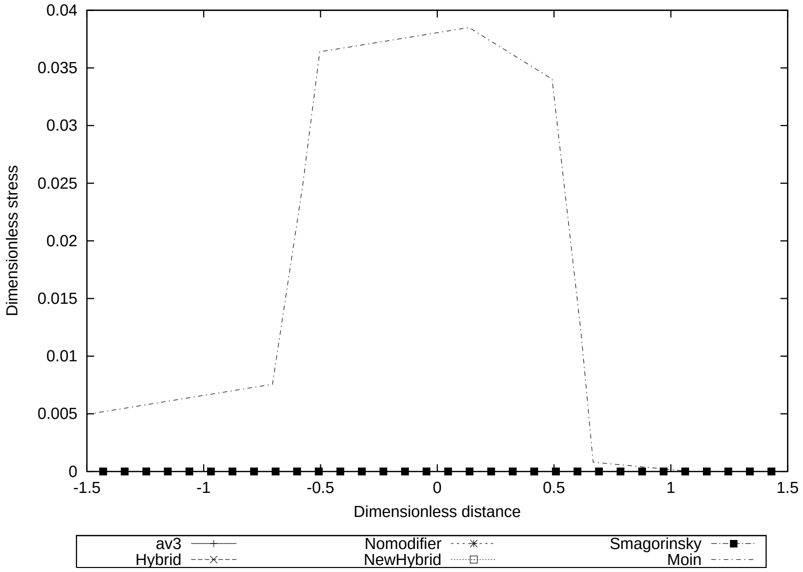

Figure 4.75: Mean w'w' stress 1.06 diameters downstream of the cylinder calculated by averaging over 1 oscillation

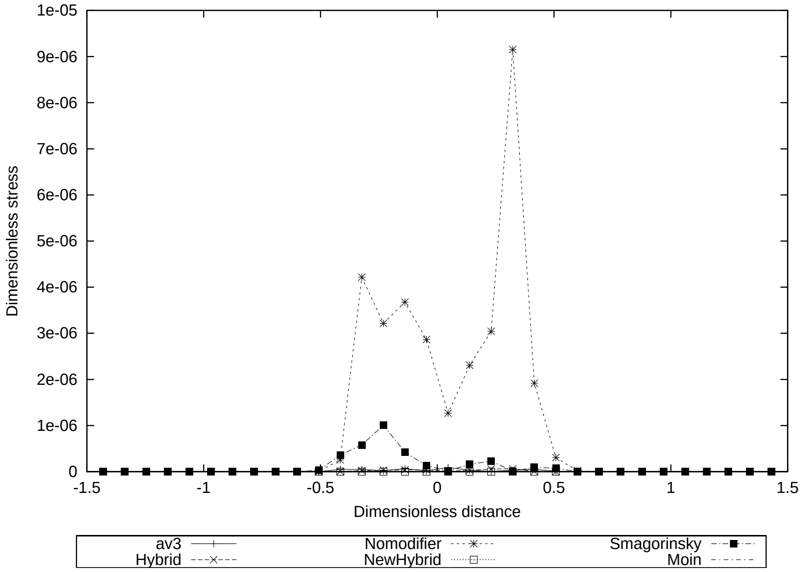

Figure 4.76: Mean w'w' stress 1.06 diameters downstream of the cylinder calculated by averaging over 1 oscillation

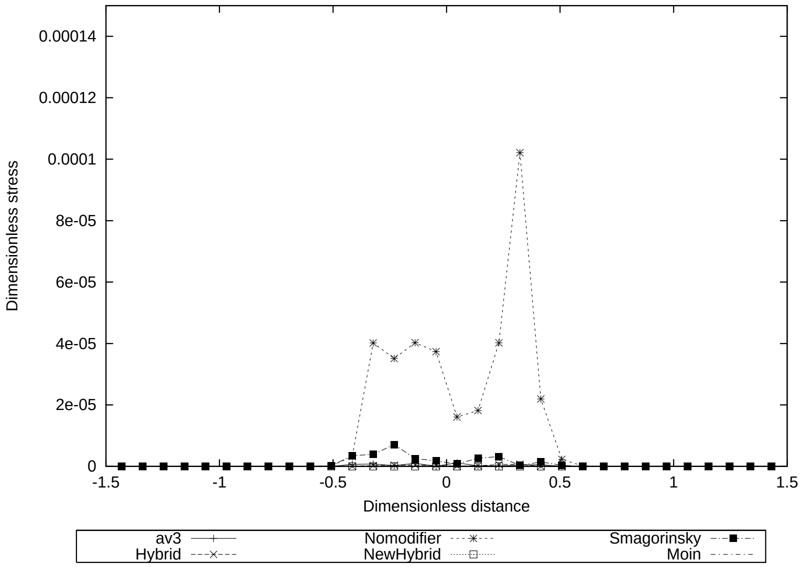

Figure 4.77: Mean w'w' stress 1.06 diameters downstream of the cylinder calculated using the theoretical EWMA approach

Table 4.1: Comparison of CPU times for the different models

simulation	time CPU1	time CPU2	time CPU3	time CPU4
k-ε	5 564	5 563	5 563	
Av3	41 810	41 690	42 440	42 430
Hybrid	41 700	41 650	41 670	
Hybrid Nomodifier	34 170	34 160	34 180	
NewHybrid	27 790	27 790	27 770	
Smagorinsky	31 800	31 670	31 800	

Table 4.2: Work distribution amongst the processors

Processor Number	Number of cells
1	60 000
2	60 000
3	54 000
4	60 000

4.4 CPU time

These results are taken from the final simulation. The time duration is ten oscillations of the wake, each cycle is 80 timesteps long and each timestep contains a maximum of 30 iterations.

The times reported are in CPU seconds so not directly comparable to wall clock time, but comparable amongst themselves. The nature of the project means that the work was allocated amongst the processors so that they may work on different parts of the domain. For this reason it is not possible to give estimated speedup of the models when used on different numbers of processors.

Chapter 5

Discussion of results

5.1 Introduction

Having laid out the results in an organised and comprehensive way, it is important to discuss their meaning so that a conclusion can be drawn as to whether there is an advantage in the Hybrid method developed.

This chapter is presented in the same order as that in which the results were presented for ease of comparison. Initially the contour plots from the final timestep have been discussed, then the velocity profiles and finally the turbulent profiles. In each section the models are presented in the same order as for the results.

A conclusion is not given at the end of the chapter, as the the next chapter collates all of the discussion into conclusions for the thesis. However, a brief overview is given before the discussion of the results.

5.2 Overview

Overall the results from the hybrid models developed herein compare well to the large eddy simulation performed using the Smagorinsky SGS model on the same grid. The velocity profiles are of the same amplitude and the turbulence predictions are of comparable amplitude and shape to the Smagorinsky model.

Although the results compare less well with the LES simulation of Moin, there is a large discrepancy in the number of cells used and the differencing scheme. I have included his results for the purposes of providing the best results available.

It can be seen from the contours of velocity and viscosity that all of the hybrid models have comparable periods to the Smagorinsky model indicating a preservation of variables across the interface.

5.3 Model development

The use of a commercial code as a basis meant that the constraints of that code influenced the hybrid model's development. In particular, the concept of a moving interface that would allow for an optimisation between the LES and RANS methodologies was not possible as the division of the cells over the processors is done so that there is no overlap. There is no need for two processors to solve the same cells twice in a commercial code so without source code intervention it would not be possible to change this and allow the moving interface.

The imposition of a USER2D patch on the interface for the purpose of smoothing the solution also proved to be difficult to control when the blocks were divided amongst the processors. The division of nodes between the processors made it very difficult to control which block the patch was allocated to and this in turn meant that it was not possible to be sure that the averaging could always be done on the RANS side of the interface. For this reason the USER2D patch could not be used to effect the smoothing at the interface.

Although the development of the model has followed the same number of timesteps each with potentially the same number of iterations the velocity and turbulence profiles are different for the different approaches. This may be due to the sensitivity of the vortex shedding to the close proximity of the interface. This however has not been investigated.

5.4 Contour plots

The first observation is that the k-ε model failed to capture the vortex shedding from the cylinder (Fig 4.1, 4.5). With a different strategy for temporal discreetisation it is possible to see this phenomenon but with the method used for this set of experiments this was not the case [112]. Reasons for this underprediction include the smoothing from the higher turbulent viscosity from the k-ε model which will act to suppress the generation of vortecies, and the poor prediction of detached flows with this model. This highlights the importance of the initial conditions and also the difficulty of capturing this vortex shedding with the k-ε model. Although the obvious candidate for blame is the turbulence model, the Reynolds-Averaged results derived from the use of the $k - \varepsilon$ compare well with the averaged results from the LES simulations thereby giving validity to its findings. However, the flow is not isotropic, and this combined with detachment are known to be areas where the $k - \varepsilon$ model performs less well. Fluctuations in the w-velocity are so small (Fig 4.3) that is impossible to distinguish between numerical error and CFD prediction. This is expected as there is no velocity gradient in the 'z' direction and therefore no expected flow in that direction.

The results of the Smagorinsky model clearly show the capturing of the vortices behind the cylinder (Figs 4.8, 4.12) and the effects of the vortex shedding can be seen to have almost completely disappeared by 10 diameters downstream of the cylinder (Fig 4.9, 4.10), which may be because of the effect of the end of the domain or because the bulk flow smooths the wake. This latter theory would concur with the knowledge that the Smagorinsky model is overly dissipative and once the energy has been given up as heat there is only the mean flow remaining. The use of the $k - \varepsilon$ model with the Smagorinsky model executed over the top of it means that the results from the simulation are exactly comparable with all of the hybrid models and that the values of k around the points of detachment are much higher as expected. This is because of the moving point of detachment and the very small length and time scales associated with the detachment of the fluid from the cylinder.

For the av3 hybrid method the prediction of the velocities is similar to that from the Smagorinsky model except that the vortices remain evident in the flow rather than stopping after 10 diameters as in the Smagorinsky model (Figs 4.9, 4.10, 4.16, 4.17). This may be because the domain is longer and so the outlet boundary condition does not influence the wake of the cylinder. The flow patterns for the w-velocity are seen to be enhanced as the fluid moves from the LES to the RANS sub-domains. There is no averaging of the w-velocity at the interface as it is very small. It is possible that the smoothing of the u- and v-velocities is causing the enhancement of the w-velocity though this effect does not last. As the fluid moves towards the outlet the w-velocity fluctuations are damped.

The higher turbulent viscosity also has a smoothing effect which makes the model ideal for steady flows (Fig 4.15). The av3 model clearly captures the vortex shedding from the cylinder and at the interface to the RANS sub-domain converts from the instantaneous to the average and passes these values to the k-ε model where the failure of the model to capture the vortex shedding is its strength as here the flow is a wake and the k-ε model can reproduce this adequately (Fig 4.20, 4.21). It would also be possible to change the length of the averaging at the interface to allow the retention or damping of the flow instabilities as required. Here is a clear example of the use of LES where required and then switching back to the RANS models for the flow governed by the mean velocity showing the hybrid methodology working well.

As can clearly be seen from the turbulent kinetic energy plot for the 'av3' model the wall layer grows rapidly in the k-ε sub-domain and this will affect the wake by preventing it spreading and eventually damping it out (Fig 4.21). The thicker wall layer changes the distribution of k and ε next to the wall which will in turn prevent the spread of the vortices from the vortex shedding. The imposed wall functions will be used more correctly with the y^+ value of approximately 30 nearer the top and bottom walls.

There is an anomaly in the wake clearly visible in the results from the av3 model. Whilst still in the LES region the wake can be clearly seen to move towards the top wall of the passage. Whilst Moin used an open domain the present work has a wall at the top and bottom it is unclear as to why the wake should move. The fact that the wake moves back to the centreline once the fluid has passed into the RANS sub domain is even more curious (Fig 4.15). One explanation is that the 'coanda' effect (Figure 5.1) is causing the effect by the faster moving flow above the cylinder causing an area of lower pressure which in turn 'sucks' the wake toward itself. However, this seems doubtful as the movement of the wake occupies the whole of one block and then moves back suddenly, whereas the movement by the pressure field would be a less sudden movement that would either continue or dynamically correct over a period of time. The inlet velocity profile is flat and there is no buoyancy used in the simulation that may affect the results in this way.

This movement of the wake to a position above the centreline occurs and is removed at each end of a block. The smoothing of the flow by averaging would not remove this effect, as the error would be passed to the RANS sub-domain.. Clearly there is an effect of passing the information into and out of this block which is causing the values to move. The sudden change at the LES/RANS interface could clearly be a sign that the interpolation at this interface is not correct, or it could be that the interpolation is correct but whatever moved the flow up at the other end of the block is now moving it down again. The movement of the wake back to the centreline is the probable cause for the increased w-velocity at the wall seen in Figure 4.17 to ensure continuity.

Despite the movement of the velocity contours those for the turbulence properties remain aligned centrally (Fig 4.21). As the generation of the turbulent kinetic energy is where there is a velocity gradient so the generation should

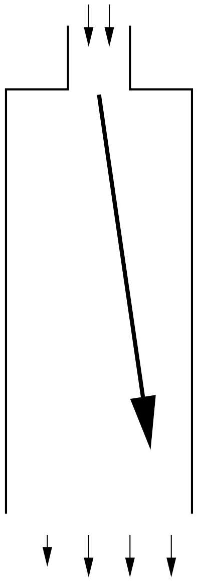

Figure 5.1: Schematic diagram of the Coanda effect

move with the velocity. The contour plots are from the same period in time so the difference could be that the gradient either side of the contour line is different or that another factor is generating the turbulence, for example that the transport away from the cylinder is affecting the result.

The Hybrid model itself captures the vortex shedding with the interface much closer to the cylinder and as with the Smagorinsky model the vortices become less coherent after approximately 10 diameters (Figs 4.22, 4.23 & 4.26). The fluctuations in the u-velocity are damped much sooner and the modification of the turbulent viscosity does not smear the flow in the RANS domain into a wake as is seen in the av3 model but rather the fluctuations can be seen to continue, spread and then be damped out. The plots of w-velocity (Fig 4.24) show the vortex pairs stopping suddenly.

The use of the hybrid model without a viscosity modifier is in essence the two models placed together with no interpolation. The pressure contours do not capture the centre of the vortices but they are marked but the u-velocity contours and the effective viscosity (Figs 4.31, 4.28 & 4.32). The turbulent kinetic energy demonstrates the fluctuations but also the peak in the values at about 1.5 diameters above and below the centreline and these peaks remain constant rather than spreading out (Fig 4.33). Clearly the peaks in the Nomodifier model prediction of the turbulent kinetic energy is in the path of the vortices at about 1 diameter above and below the centreline of the wake and this is very close to the Smagorinsky model.

The Newhybrid model captures the decaying vortex shedding from the u- and v-velocities and these are seen in the contours of pressure as well (Fig 4.34, 4.35 & 4.37). The u-velocity lacks the resolution of the vortices that is seen in some of the other modelling techniques though the v-velocity is more similar . The w-velocity suggests coherent structures that follow the vortex shedding from behind the cylinder but the magnitude of the w-velocity is much less than for the other models and of the same order as the k-ε model, though with structure. This movement is concentrated in the region immediately behind the cylinder and at the outlet of the domain suggesting that it is not well resolved at all. The effective viscosity is seen to increase as well and this then damps out the oscillations. This then could be a tunable parameter though was not intended as such. The contours of kinetic energy do not posses the twin tails that are seen from the LES simulation but is aligned centrally behind the cylinder and is higher along the centreline as would be expected by the k-ε model alone.

An obvious difference in the pictures is that of the difference in the w-velocity. The grid contains ten cells in the spanwise direction which is half a diameter thick. The patches on the front and back of the domain are periodic, meaning that what leaves the domain in one enters in the other. The Smagorinsky model captures the vortex shedding from the cylinder and for each of the vortices there is a predicted rotation about the direction of bulk flow. The rotation may be in part an effect of the boundary condition because of the small depth of the domain (0.5D). It is reported in the literature that there are spanwise movements in the wake, though for simulations that are this narrow in the spanwise direction this is less common. The periodic boundary condition simulates an infinite wake in the spanwise direction, and so this suggests that there is no rotational motion about an axis aligned with the direction of mean flow but rather countercurrent flows in the spanwise direction of infinite length. This however is not reported in the literature and is therefore an effect heightened by the boundary conditions[100].

The contours of w-velocity are different for the LES and the hybrid models. For the Smagorinsky model there are two pairs of counter rotating vortices that decrease in size after approximately ten diameters. The vortices do not spread more that two diameters from the centreline, this is after approximately seven diameters and before that there is a gradual spread from the cylinder. The av3 hybrid model does not appear to have captured the w-velocity vortex pairs at all. This is unusual as in the av3 model the LES sub-domain is used for a much longer period than the other models which do capture the vortex pairs. Another unusual feature is that the contours of w-velocity next to the wall in the RANS section for the av3 model. The most striking feature is that the contours only appear on the lower wall. There is no difference in the treatment of the upper and lower walls and the velocities have not been limited to any values. This is probably caused by an interpolation error in the averaging for the body forces and is then convected along the domain. The velocity in the spanwise direction remains low and this therefore will not affect the bulk flow along the domain which has an average velocity two orders of magnitude higher. The Hybrid method transfers the w-velocity vortex pairs into the RANS domain but the spread of the eddies is suppressed once it crosses the interface, and also in the RANS domain there are extra disturbances introduced so that the motion is not a pair of vortices but more a dance in a figure-of-eight. The Hybrid-nomodifier model predicts a better spread of the counter-rotating vortices though they are soon damped out by the turbulent viscosity from the k-ε model. Similarly to the Hybrid model, the Newhybrid model fails to resolve the vortices in the RANS domain but their spread is again restricted. This is not from the walls but rather the viscosity modifier aids the initial development of the vortices, but restricts them to the region of lower viscosity. Were the vortices to spread they would be suppressed by the higher turbulent viscosity of the flow near the walls.

5.5 Velocity profiles

The EWMA averaging technique is sensitive to where in the oscillation the simulation is when the sample is taken because of the bias of the current value to the moving average. Whilst the average turbulent fluctuation could be calculated this does mean that the answer becomes dependent on the averaging and not the physics (Section 3.3) leading to a bias towards the most recent values which is clearly seen in Figure 4.40. This however, is the cost of measuring the actual fluctuation from the moving average as opposed to the RMS value of the amplitude which would reveal nothing of the turbulent statistics. This is the reason for the distortion in the values for the mean velocities. All of the models match the results for the turbulence profiles close to the cylinder suggesting that the LES region is not adversely affected by the proximity of the interface and that the capturing of the attached vortex behind the cylinder is correct.

The robustness of the EWMA methodology and that it can calculate results as the simulation progresses means that if for some reason the solution were to stop the statistics would be calculated. The good comparison of the results with all of the models and with those of Moin add weight to the method as a way of averaging a varying mean. The main problem with the method is the time lag between the fluctuating value and the moving average. This is unavoidable though unless the weighting on the latest variable is altered such that there is little weighting on the current average, this though means that the weighting on the current value is such that it scarce can be called an average. An optimum balance has been used though this was formulated on theoretical grounds and not specifically for this case. However the altering of a general tool for specific cases only proves to show that the tool is general and is not a solution to all of the problems. For example the k-ε model itself has many documented values for its constants depending on the composition of the domain. This highlights the weakness in the model in that it has to be tuned. Whilst the k-ε model is a good general model it does need optimising for specific cases.

This returns to the rationale for this thesis. If the simulation has to be as general as possible with the fewest number of situation specific constants to be tuned then the accuracy of the model has to be increased as well. It has been demonstrated that the LES models in use today still contain a number of parameters that can be adjusted and extrapolating from this the increased in accuracy could be achieved by the use of DNS. Pragmatically though, this is not currently an option for complex engineering flows where the purpose of the CFD model is insight and the models placed on the top of the CFD are often based upon experiments and may be correct to only ±30%.[1] The underlying (in)accuracy of the fluid flow becomes less important compared with the chemical reaction being solved. Furthermore, the bulk of LES simulation being reported in the scientific journals only compare the mean velocity to experimental data. It is a waste of resource allocated to an LES simulation if the primary use of it is the production of a mean velocity data set.

One of the possible causes for the models being out of phase with one another is the method used to generate the initial flow field. This used a k-ε model solution as an initial starting point and then developed the flowfield from there without vortex shedding at the first instance. This means that the models may have initially started shedding vorteces at slightly different times as the method was followed and only the final run analysed for the results. Large eddy simulation is essentially attempting to model a chaotic flow that has a

[1] CFD models are discreet approximations of continuous fluid flow. They are based upon mathematical models which have been curve-fitted to the best experimental data. This experimental data has been gathered with a machine error and then interpreted by people. At each stage there is an error introduced, and the effect of this becomes larger the closer to the CFD model one is

certain level of deterministic behaviour to give the coherent structures and as such the expectation that all LES models will give the same results is false for the short term. This would account for why the averaging over tha last 10 oscillations will produce similar results.

The use of a theoretical Strouhal number to calculate the timestep required for this simulation may lead to an error if the periodic time is not predicted correctly by the simulation. The result of this would be that as the time period is not predicted correctly the period of averaging will be incorrect for the 'dump' averaging and the averaging over one period of the vortex shedding. For the 'dump' file averaging this will lead to a bias of the results and for the long average the results will again be skewed. If the wrong number of steps had been calculated for one time period this would not affect the average as all of the averages would have been wrongly calculated. That is of course assuming that all of the time periods are the same for all of the different models. It is also possible that the adjustment of the turbulent viscosity may have an effect on the fluid flow to smooth the change in turbulent viscosity and in doing so altered the flow characteristics. This is suggested by some of the velocity profiles (Figs 4.42, 4.44)

It is possible for the downstream conditions to affect the flow upstream and the position of the interface between four and five diameters downstream from the cylinder is quite close. In a pipe, turbulent flow is assumed to take 10 diameters to develop. A turbulent wake is assumed to require 80 diameters to obtain self-similarity [111] (where the turbulence is isotropic, governed by its immediate surroundings and changes slowly). This condition has been challenged by this model and yet it still works. One of the reasons for this may well be the robustness of the k-ε model above what it was originally intended to do by giving adequate results when used in anisotropic turbulence, an alternative is that the turbulent viscosity from the k-ε is not so high that it smooths the vortices that are convected into the RANS sub-domain. However, the influence of the interface on the LES sub-domain has not been investigated by the study but it has been inferred that the effect is small as the profiles of velocity and turbulence remain unchanged from the full LES model.

The v-velocity profiles are very different from those of Moin, though the hybrid results compare to the LES results of this study. One of the reasons for this will be the difference in averaging period. The theoretical EWMA average will be affected more by the passing of turbulent fluctuations than the other methods and so the fluctuations are much more dependent on where in the cycle the simulation is stopped.

The solutions do not match the results of Moin for a number of reasons. The first is that the domains used were of different boundary conditions where his are open and the ones herein were walls at the top and bottom of the domain. The imposition of the walls will suppress the spreading of the wake, as will the higher turbulent viscosity in the RANS sub-domain. The imposition of walls at the top and bottom of the domain will act to constrain the wake from spreading, thought the use of the RANS modelling methodology and specifically the higher turbulent viscosity, will act on the wake to smooth it prematurely[92]. The number of cells used for these simulation are an order of magnitude less that used by Moin and the differencing scheme is only second order in accuracy. This will only serve to reduce the accuracy. For this reason then the 'LES' calculation performed herein can only be assumed to be a *very* large eddy simulation though the results compare well with those of Moin for the mean velocities The turbulence statistics the results are the correct shape though the order of magnitude is wrong. The imposition of an interface at any point to impose averaging will reduce the amount of information that can be drawn from the simulation, and the positioning of the interface near the cylinder will have an as yet unquantified effect on the result. Having stated this, the results in the isotropic k-ε model sub-domain compare well when compared to the LES simulation of Moin.

At 1.06 diameters downstream of the centreline of the cylinder the u-velocity profiles are similar to those from Moin but whereas Moin's are symmetric about the centreline the bias from the most recent detachment from the cylinder is evident in the other averages such that the profile is not symmetric about the centreline (Fig 4.40). The results from the v-velocity at this distance from the cylinder is such that all of the hybrid models compare well with the Smagorinsky model. The Newhybrid model has a much larger amplitude predicted than the other models and comparable in amplitude that the results from Moin as well (Fig 4.46). However, this average was made using the 'dump' file method which has been shown to be closer to the results of Moin.

At 1.54 diameters the u-velocity profiles are more symmetric in the average from the dump file that from the moving averages indicating the history effects on the moving averages (Fig 4.41). The EWMA average over an entire period shows the symmetry of the Newhybrid model about the centreline but the other hybrid models are still affected by the wake. It is of interest to note that the av3 model would be expected to be very similar to the Smagorinsky model as the interface for the av3 model is where the domain ends for the other models (ie that the length of the Smagorinsky model is the same in both cases). However, as all of the models have been allowed to develop the vortex shedding the av3 model has become out of phase with the Smagorinsky model by π radians. This will provide a check for the results as the av3 results should remain out of phase if the hybrid methodologies are retaining the same period that is passed to them by the LES sub-domain. It is also noticeable that the Nomodifier model that uses no smoothing at the interface has very similar results to the Smagorinsky model. The v-velocity averages are all dissimilar to the results of Moin. From the dump files many of the velocities above the cylinder are positive and the centreline of the profile does nor correspond with the centreline of the wake. Many of these results can be attributed to the method used to average the results which means that the effect of the current vortex means that the profile is not smooth but has a second peak (Fig 4.47). The results from the av3 model can be seen to be closer to those of Moin. For the models using a viscosity modifier there is a minimum and maximum in the v-velocity for the Newhybrid and Hybrid models respectively.

The length of the recirculation behind the cylinder should be up to the 1.54 diameter profile. This is not easily demonstrated by the velocity profiles but rather can be seen by looking at the pictures of the v-velocity plots. All of the models compare well with this observation.

The profile at 4 diameters is indicative of the behaviour of the flow just before the interface is crossed by the fluid. The effect of the phase can be seen in that the u-velocity averages all have minima on the opposite side now to the profiles at 1.06 diameters (Fig 4.42). For the average over one period the results are further from those of Moin that from the short averaging. The Smagorinsky results are close to those of Moin and the models comparable to the LES simulation from this study. The Newhybrid model has noticeably off-centre results where the minimum is markedly smaller than the other results and the other averaging techniques (Fig 4.43). The v-velocity from the Newhybrid model has a similar shape to the results of Moin though the amplitude is larger (Fig 4.49). All of the averaging techniques predict the smaller v-velocity above the cylinder than below it though the behaviour one diameter either side of the centreline is very different. The dump file average predicts that the av3 model has a strongly negative velocity about the centreline, as does the theoretical EWMA average. The weighted average over one period however predicts that the v-velocity of the av3 model is positive by the same magnitude. In a similar way the Hybrid model has a positive average for the dump and theoretical EWMA average but has a peak below the centreline and is then negative for the average over one oscillation. At four diameters behind the cylinder the average from the dump files for the 'Hybrid' simulation show that the flow is moving vertically upwards. The domain is bounded at the top and bottom by a wall so the fluid must either move downstream or in a spanwise direction to remain incompressible. At four diameters downstream of the cylinder the mean u-velocity calculated from the dump files is not centrally aligned and the flow above the cylinder has a faster velocity. The faster velocity will mean a lower pressure and this will cause the flow to move upwards. This movement will then either cause the wake to move upwards towards the lower pressure thereby reducing the flow and allowing more fluid to move along the bottom of the domain, dynamically balancing the fluctuating velocity, or will cause the suction of the wake closer to the wall whereby the distance between the wake and the wall is reduced and so the fluid moves faster through the space available which decreases the pressure more and encourages the cycle of suction.

The velocity profiles at 5 diameters are the first inside the RANS sub-domain for all except the Smagorinsky and av3 models. The Hybrid and Newhybrid models now both have increased minimum u-velocities so that they are more comparable with the other models. The av3 model remains out of phase with the Smagorinsky model as before. All of the velocity profiles are skewed above the cylinder except for the hybrid model which is almost symmetrical (Fig 4.44). As with the averages at 4 diameters those at 5 for the v-velocity are very different between the different modelling techniques. The moving averages have values comparable to one another with the Newhybrid model retaining its 'v' shaped profile but with a negative velocity over the whole of the profile (Fig4.49), with the dump file though this profile is again similar in shape to that of Moin but with a greater amplitude. The Smagorinsky and Nomodifier models are again similar and have maxima at about ±0.6D.

The 10 diameter velocity profiles vary by up to 25% of the inlet velocity (Fig 4.52). The av3 model which is still using the Smagorinsky model at this point indicates an almost flat profile with what may be an averaging effect from the top to the bottom of the plot, which may be due to when the average was taken. The Hybrid model also has a flat profile. The newhybrid model has a very pronounced 'v' shaped profile and whilst not of the same magnitude as the work by Moin has a much more pronounced shape that the Smagorinsky model. For both the averaging over one oscillation and the theoretical EWMA average the flatter velocity profile of the av3 model is present. The profile has a more pronounced 'v' which indicates that the effect could be from the averaging used and the period within the oscillation that the solution stopped. The Hybrid model contains a similar profile which is almost a reflection of the av3 model about the centreline. The Newhybrid model retains the largest amplitude 'v' but it van be seen to be more offset (Fig 4.45) The Smagorinsky model has a 'v' shaped profile but this is not as pronounced as the results published by Moin and also indicates that the flow may have localised regions of faster flow and approximately ±0.6D before the fluid velocity again increases towards the top and bottom of the channel. It is noticeable that the mean velocity at the top and bottom of the sample region is not as large as the inlet velocity indicating that there must be a broad ‘v’ shape over the whole channel for there to be conservation of mass. The difference between the two viscosity modifier methods is seen in that the hybrid model now has a much flatter profile where the smoothing into the k-ε model is shorter and so the higher turbulent viscosity has contributed to the smoothing of the flow. The Newhybrid model has retained more of the features of the flow and hence has a larger ‘v’ shaped profile. Comparison with the contour plots indicates that the vortices have all but died out in the Smagorinsky model at this length from the cylinder and this would be indicated by a flatter profile (Fig 4.12, 4.45). However, for the Newhybrid model the turbulent viscosity is seen to increase along the centreline of the wake as the flow nears the end of the domain, this increased resistance to motion will have the effect of forcing the fluid to move faster nearer the walls to allow the same throughput of volume. This is also indicated by the v-velocity plots (Fig 4.50) and the contour plot for v-velocity and viscosity for the Newhybrid method (Fig 4.35, 4.38). For the v-velocity plots themselves, it is noticeable that the results of Moin indicate a general upwards movement of the flow especially above the cylinder. It would be expected that the general motion of the fluid above the culinder would be down and the dominant v-velocity of the fluid below the cylinder would be upwards.The boundary conditions are not the same but for a non-buoyant flow this is surprising. All of the models have peaks at approximately ± 1 diameter above and below the centreline. For all of the models except the Newhybrid model (which has a flat profile) and the av3 model the peaks are maxima. For the av3 model the peaks are minima. The importance of where in the cycle the simulation is stopped is shown here but also the continued flow of the vortices into the RANS sub-domain in the av3 model whereas in the Smagorinsky model the vortices are dying out by this point. The other large peaks are associated with the hybrid model and these positive peaks can be accounted for by the ending of the coherent structure behind the cylinder in the same way as the Smagorinsky model (Fig 4.50, 4.8 & 4.22). The breakdown of the coherent structure can account for the irregularities in the flow as the fluid motion gradually changes back to channel flow with a smooth velocity profile.

It would be expected that for the hybrid models the flatter velocity profiles associated with the 10 diameter profile will be influenced by the isotropic nature of the k-ε model which increases the spreading rate of jets and wakes. This will cause the velocity profile to become flatter sooner, and to dissipate the higher turbulence energy that is currently at the centreline of the domain.

5.6 Turbulence profiles

The comparison of a long running average velocity with the fluctuating component has been shown earlier to measure the RMS value of the amplitude of the oscillation and not the fluctuation of the velocity about the varying mean. It has also been shown that the use of a moving average will allow the comparison of the phase averaged mean and the instantaneous value to give a turbulent fluctuation at that point, and that the longer the period over which the moving average is weighted the more it becomes similar to the average over all time. For this reason it would be expected that the turbulence predictions from the EWMA model where the weighting time is longer would be closer to the 'dump' file calculated turbulence properties than the theoretical EWMA approach, especially as all of the mean velocities are comparable to one another. This is clearly not the case. One explanation for this is the use of the instantaneous value for the value of the fluctuating velocity that is used for that timestep, and as such is not a clearly representative sample. It is not an averaged $u'u'$ that is plotted but an instantaneous value and so the averaging over 1 period is particularly vulnerable to the value of the velocity compared to the mean velocity because at times during the period the value of u' will be zero. In addition to this the velocity fluctuation at a point is not that great so the longer time averages at that point will give smaller fluctuations which will always be of approximately the same value, whereas the theoretical EWMA average which moves more each timestep will always allow for larger fluctuations. In addition the larger grid will not allow for the simulation of small eddy sizes so the interpolation and other numerical errors will affect the flow more.

The $u'u'$ profiles at 1.06 and 1.54 diameters downstream of the cylinder are all similar to the Smagorinsky model and the results of Moin in shape though of a much reduced magnitude. This can be attributed to the size of the grid, the differencing scheme and the averaging. In addition the turbulence property plotted is the instantaneous for that timestep and is not an average turbulence value. If the comparisons are made with the Smagorinsky model on the same grid then the results of the hybrid models fare well. The results of Moin are used to indicate the shape of the turbulence profiles. (Fig 4.53, 4.54, 4.55 & 4.56). It is noticeable that the Hybrid, Newhybrid and av3 methods consistently give larger turbulence predictions.

For the Hybrid and Newhybrid an attempt could be made to explain the larger turbulence predictions on the grounds of their influencing the downstream boundary condition by the use of the viscosity modifier. However the change from LES to RANS still happens at the same position and the treatment of the models in the LES sub-domain is exactly the same for all of the hybrid and LES models. The difference between the Smagorinsky model and the av3 model is much more subtle. They are known to be completely out of phase from the u-velocity plots, and the turbulence statistics are instantaneous for the last timestep of the simulation. The differences then between the value calculated by the 'dump' files and the moving averages can be seen to be different whereby the moving averages have much closer predictions for the two models as at this point they are the same. Both models have over 15 diameters behind the cylinder before any change in boundary conditions (outlet for the Smagorinsky model and interface for the av3 model). However the u-velocity contour plots have indicated that the vortices in the Smagorinsky model are removed by the end of the domain but that the av3 model retains structure until the interface. Either way it is unlikely that these boundary conditions are influencing the fluid motion so close to the cylinder and so far from the boundary condition.

For the $v'v'$ stresses the predictions near the cylinder are all comparable to the Smagorinsky model (Fig 4.65, 4.66). Again the general shape of the profile is comparable to the results of Moin but the magnitude is smaller. As with the $u'u'$ stresses the Newhybrid model has the largest predicted level of turbulence and this is then followed in turn by the av3 and hybrid models. Apart from the Newhybrid model all of the other models are comparable in magnitude. As before the results calculated from the 'dump' file method are the largest and those from averaging over one whole time period the smallest. The results are all close to being symmetric about the centreline of the wake. As these profiles are both within the LES region the differences are probably because the stresses have been calculated for the final timestep and there is no averaging or measuring of the maximum value.

At 4 diameters downstream from the cylinder, which is just before the interface, the predictions of the $u'u'$ stresses from all of the models compare well with those of Moin (Fig 4.58). Again the asymmetric profile is seen as a result of the averaging and the final position of the solution impinges on these results greatly. The moving averages again show much smaller predictions than the 'dump' file results (Fig 4.59) which will be influenced by the averaging method used, though it is interesting to note that the results for the dump file all tend to zero for in the immediate wake of the cylinder rather than reducing in magnitude as is seen in the results of Moin. The moving averages all represent the shape of the profile well (Fig 4.59) with the twin peaks. As with the $u'u'$ stresses the $v'v'$ stresses are closer to the results of Moin for the 'dump' file average. The moving averages clearly indicate a single peak but the history associated with the moving average has biased the peak away from the centreline. The av3 model can be clearly seen to be directly out of phase with the Smagorinsky model as in the velocity profiles (Fig 4.47). The Hybrid and Newhybrid models have differing answers, the peak of the Hybrid model is larger though the Newhybrid model is symmetric about the centreline of the wake. These two viscosity modifier models are not entirely in phase as is indicated by the velocity profiles (Fig 4.43) and the difference in phase may account for some difference in magnitude.

At 5 diameters downstream of the cylinder the $u'u'$ stresses are represented less well by the 'dump' file averages. The effect of the RANS models is now on the models. The long averages all have 3 peaks instead of 2 (Fig 4.61, 4.60). Apart from the centrally aligned Newhybrid model the larger peak is on one side with the smaller peaks at the centreline and at approximately 1 diameter the other side of the centreline and are much smaller than the larger peak. The third peak is almost certainly not an effect of the change in sub-domain as it is clearly evident in both the Smagorinsky and av3 models which have no such change. Probably this is how the turbulence has been predicted here and the smaller values make the results seem worse. The results for the theoretical EWMA average are as the dump file, with two peaks, again with the bias to reflect where the simulation finished. For both the u'u' and v'v' stresses at 5 diameters behind the cylinder where the profile is just inside the RANS domain the stresses are seen to be closest to those predicted by Moin and having the most symmetry, suggesting that at this point the flow properties of the LES sum-domain are still influencing the flow in the RANS sub-domain, and furthermore this effect is increased by the use of the viscosity modifier. By ten diameters downstream the velocity and turbulence profiles are much the same as for the other methods of merging the models.

The $v'v'$ stresses are predicted well by the 'dump' file method and as with the $u'u'$ stresses the results are not quite as good as for the profile at 4 diameters (Fig 4.70). An obvious answer for this would be that the use of the turbulence model is affecting the size of the fluctuations seen. However, if this were the only reason then the Smagorinsky model and the av3 model would retain the comparatively more accurate results as thy have not changed to a Reynolds averaged method of calculating the flowfield. In addition to this the size of the mesh, coupled with the overly dissipative nature of the Smagorinsky model will make the dissipation of the turbulence happen more quickly. The larger lengthscale indicated by the k-ε model as the flow moves toward the outlet suggests that the turbulence is lessening and the flow instability is dominating over the turbulence. This is seen in the results of Moin as the size of the $v'v'$ stress is reduced as the fluid moves downstream of the cylinder. The coarser the grid the larger the lengthscales that the grid is able to adequately represent, and as the turbulence dies away the lengthscales become smaller. The use of only a second order differencing schemes will lead to more numerical dissipation than if a higher order scheme were used. The coupling of these two, therefore, will act to reduce the level of turbulence that can be accurately predicted. The moving average with averaging over one period gives less bias to where in the oscillation the simulation stops than does the theoretical EWMA average (Fig 4.71, 4.72). The moving average weighted over one period carries the 'M' profile seen in the coutours of tubulent kinetic energy because of this history effect. In addition it remains an order of magnitude smaller than the prediction by the theoretical EWMA average. From the EWMA average the Smagorinsky and av3 models remain out of phase and the peaks in their results are the same in magnitude and vertical distance from the centreline as each other, thereby validating each others results. As before all results are comparable to the Smagorinsky subgrid scale LES simulation performed on this grid.

At 10 diameters downstream of the cylinder the profiles of the $u'u'$ stresses have become increasingly small inline with the reduction from the profiles at 4 and 5 diameters. The 'M' shape to the profile is very prominent with the centre dip returning to zero as opposed to being a dim. The av3 model remains out of phase with the others, and remains with a much higher minimum level at all points in the simulation (Fig4.63). The Newhybrid model remains centrally aligned. The long weighted average shows this difference in minimum values more clearly (Fig 4.64) and suggests that the many of the models have moved to a more centrally aligned position. The offset of the av3 model coincides with its movement towards the top wall clearly seen in the contour plots (Fig 4.15).

All of the models, though comparable with the Smagorinsky model on this grid underpredict the $v'v'$ stresses after 10 diameters. From the moving averages the predictions of the profiles are the correct shape for the Newhybrid and Nomodifier models but not for the Smagorinsky, av3 or hybrid model which have smaller peaks to one side, though the largest level of turbulence is on the centreline(Fig 4.73). A comparison with the mean u-velocity after 10 diameters indicates that all of the models are comparable except the Newhybrid model (Fig 4.45) but this is not the case for the turbulence predictions. The mean v-velocity after 10 diameters indicates an almost flat profile for the models though this does not indicate that there are no fluctuations in the flow merely that the profile is flat. The absence of shear does indicate that the turbulence predictions will be small though. For both the long average and the theoretical EWMA average the Nomodifier model predicts higher stresses than many of the other models, and the Hybrid model gives the worst predictions. Given that the Hybrid and the Newhybrid model are very nearly the same this suggests that the viscosity modifier has a large impact on the results.

The $w'w'$ stresses are all very small compared to the results of Moin. The results are all comparable to the Smagorinsky model on this grid with the largest predictions being made by the Nomodifier and Smagorinsky models. These results are demonstrated in the contour plots for the w-velocity. The Smagorinsky model retains the strong gradients of w-velocity, the Nomodified model starts with strong gradients but they are suppressed once transition into the RANS sub-domain is achieved (Fig 4.75, 4.76). Again the results from the av3 model are surprising as this close to the cylinder it would be expected that the results would be as for the Smagorinsky model. The Hybrid model only has large gradients of w-velocity in the RANS sub-domain, whereas the smallest predictions, those from the Newhybrid model are very much lacking in contours of w-velocity at all. The lack of contours indicates a lack of turbulence. Although the $w'w'$ stresses will be affected by the other velocity components the bulk flow is away from the cylinder with the primary disturbance in the vertical direction. This would suggest that the turbulent fluctuations in the spanwise direction are much smaller and will therefore require a much finer grid (Fig 4.77,4.77,4.66,4.55).

The use of the k-ε model for the whole of the domain would mean that the inherent wall functions in the model are used. However, for a y^+ value of 7 (Fig 4.14) the first cell centre is much closer to the wall and actually in the log layer and this is where the influences of both the viscosity and the turbulence are felt by the fluid. This means that for the flow around the cylinder the wall function will not be used, though for the flow along the walls it will be utilised due to the higher y^+ values. It would possibly have been better to have used the low Reynolds number k-ε model as his integrates through the boundary layer removing the need for wall functions. Though this would not have been using what is often the default model.

Frequently the turbulence profiles have qualitatively the correct shape though the orders of magnitude are too small. For example, at 10 diameters the u'u' values are two orders of magnitude too small. This may be a combination of the coarseness of the grid and the smoothing from both the numerical diffusion and the k-ε model.

The highest courant number in the simulation is 0.4, which indicates that the fluid flow is adequately resolved in each cell and were this an explicit code that it is operating within its stability limit. In the immediate wake of the cylinder, where the LES simulation is to be used, the initial k-ε model predicts that the turbulent time scale is of the order of 0.359 to 0.43 seconds. In the region where the flow detaches from the cylinder the predicted turbulent timescale is 0.008 seconds. The timestep size is 7.051×10^{-3} seconds so the capturing of the turbulent fluctuations is possible from the transient perspective. However, the predicted turbulent timescale from the Smagorinsky when run with the k-ε model predicts a turbulent timescale of 1.42×10^{-3} seconds in the near wake of the cylinder for which the temporal discretisation is too large to capture the fluctuations on that scale. At the point where the turbulent eddy detaches from the cylinder the timescale predicted by the k-ε model is of the order of 10^{-10} and this fluctuation is far smaller than can be simulated by the current time step.

In addition the coarseness of the grid will increase the numerical diffusion in the solution and the use of only a second order differencing scheme introduces errors compared to the model of Moin. The coarser grid will not be able to capture such fine eddies and this reduces the turbulence statistics that can be captured by the model.

The size of the turbulent eddies predicted by the initial k-ε model suggest that in the region behind the cylinder the size of a turbulent eddy ranges from $4 \times 10^{-5}[m]$ where the flow detaches from the cylinder, to $1.5 \times 10^{-3}[m]$ where the centreline of the wake crosses into the RANS sub-domain. The grid size in the RANS part of the model is $4 \times 10^{-3}[m]$ in the x-direction and this is larger than the turbulent lengthscale predicted by the k-ε model for the whole of the simulation as the predicted lengthscale increases to $3 \times 10^{-3}[m]$ as the flow leaves the domain. However, the prediction of the turbulent lengthscale by the k-ε model is calculated as follows

$$\ell = C_\mu \frac{k^{\frac{3}{2}}}{\varepsilon} \tag{5.1}$$

and the lengthscale predicted in this way is smaller where there is a higher predicted level of turbulence. The k-ε model assumes local isotropy so creation and dissipation of turbulence take place at approximately the same position in space and time, this is why the lengthscale is much smaller at the point of detachment and much larger as the fluid leaves the domain. The turbulence is affecting the bulk flow and so as the fluid moves towards the outlet the turbulent fluctuations are smoothed (as is shown from the Smagorinsky model simulation) but the fluctuation in the mean flow remains hence the larger lengthscale. The flow Reynolds number is 3 900 and so many of the flow effects may be from the macro flow instabilities and not the turbulent fluctuations about these instabilities leading to an incorrect use of the k-ε model in the transitional regime, as it was originally intended for high Reynolds number flows.

It is therefore more important to have more cells in the region where the flow is rapidly changing as the flow disengages from the cylinder and the length and time scales are very small. The cell size next to the cylinder is $4 \times 10^{-4} [m]$ in the direction perpendicular to the cylinder and this is an order of magnitude higher than the predicted lengthscale at the point of detachment, making the resolution of the turbulent eddies in that region impossible. However, the wall function of the underlying k-ε model will affect the flow and the detachment points are correctly predicted as is the behaviour of the fluid in the wake of the cylinder. Apart from the point of detachment the resolution in the immediate vicinity of the cylinder is adequate enough to capture the turbulent lengthscales predicted by the k-ε model. The Smagorinsky model assumes that the filter is the same size as the cell spacing and so this means that the cell size is too large in the LES sub-domain. However, this is not an in-depth study into the the physics of turbulence but a proof of concept that the novel hybrid methodology works. As to this respect the LES model captures the flow dynamics that are possible and that are not captured by the k-ε model.

The use of the viscosity modifier can clearly be seen to affect the turbulence statistics and the mean flow. From the predictions of the u-mean velocity and the $u'u'$ and $v'v'$ stresses the effect can clearly be seen. As there is a higher viscosity immediately behind the cylinder as compared to that above and below the cylinder then the fluid continues to move faster nearer the walls. This is seen as the NewHybrid model has larger turbulent stresses in the vertical direction immediately into the RANS domain. The modification of viscosity will suppress the immediately higher turbulent viscosity of the k-ε model but the maximum viscosity in Figure 4.38 is still 8 times less than for the k-ε model itself indicating the fluctuations in the viscosity predicted by the Smagorinsky model will have little effect on that from itself and so the limited usage of a viscosity modifier.

It is unclear as to any detrimental effect that the use of the viscosity modifier may be having on the k-ε model. If the turbulent viscosity is reduced in-line with the use of a turbulent viscosity modifier then there is less resistance at that point to the movement of the fluid. If the flow were unbounded at in the model by the use of pressure boundaries then it would be likely that flow would be entrained into the wake of the cylinder faster as the lower turbulent viscosity would allow for an increased velocity and hence a lower pressure entraining flow in. This would lead to a shortening of the wake. Extrapolating from this there is reason to suspect that the use of a viscosity modifier may lead to physically unrealistic results in a case where the domain is unbounded and the turbulence generated in the centre of the domain. This behaviour could probably be mitigated by the use of a large enough domain such that the boundary conditions were far enough away from the LES sub-domain to reduce the effect of this. It is probably that the reason why this does not happen in this case is the use of walls at the top and the bottom of the domain. The enclosure provided by the walls restricts the flow in the (lower effective viscosity than would be expected) RANS sub-domain to what enters from the LES sub-domain.

It is not possible from the results to assess the difference that the interface for the av3 results may have had on the LES sub-domain. As the results are used to form an average in the first cell of the RANS sub domain it is possible that there may be an effect upstream on the LES. The results from the av3 run compare well with the those of the Smagorinsky model, though the Smagorinsky model itself has the outlet placed the same distance form the ten diameters profile as the interface is in the av3 model. The mass outlet used by the simulations enforces a fully developed flow at the outlet by assuming zero gradient in the direction normal to the boundary

The levels of turbulence predicted by the models are all comparable with each other and the Smagorinsky model. They are all smaller than the results of Moin. The results from the 'dump' file method of averaging predict the largest turbulent fluctuations. This is because the turbulence is being measured as the amplitude of the flow away from a mean value, and so tends towards the RMS value of the amplitude. Where the EWMA averaging methodology is used, the 'short' average, the measure of the turbulence statistics the fluctuation is calculated at each timestep. It is measuring the disturbance from an average that moves with the flow. The use of the EWMA methodology with an implied weighting over one period the 'long' average, should give larger predictions for the turbulence that the theoretical EWMA as the longer averaging period means that the average is closer to the long running average and so the method is closer to measuring the amplitude than the fluctuation. However, thought the mean velocities for the methods are very similar the turbulence predictions vary by up to two orders of magnitude. There are three main reasons for this. The first is that the values given from the 'long' and 'short' EWMA averages are instantaneous as they are the values for that timestep only, whereas the 'dump' averages have an implication of averaging as they are calculated and averaged at four points in that last cycle. The second is that though the mean velocities for the wake are very similar between the two EWMA averages, there is only a small difference between the mean and instantaneous value (ignoring the fact that all of the velocities are averages in the RANS sub-domain). When multiplied to give the $u'u'$ stress, the multiplication is enough to accentuate the difference between the differences, which leads into the third point. The mesh spacing used in the simulation is not fine enough to sufficiently resolve the turbulence fluctuations and so there are only very small differences between the average and instantaneous values, which in turn leads to very small predictions of turbulence. This would be expected in the RANS sub-domain as the velocities there are all averages anyway, but for the Smagorinsky and av3 models the turbulence predictions for all of the profiles downstream of the cylinder would be expected to be larger. The dump file averages compare better with those results of Moin as this method of averaging is close to the long average used by Moin. The predictions are larger therefore, as the method is predicting the RMS of the amplitude of the oscillation.

Outside of the academic community how many people will investigate the turbulence properties of the flow may be questioned. However the 'hybrid' models all provide adequate representation of the flow when quickly investigated by the use of a post processor before the closer inspection by way of assessing the turbulence properties. If this is given as a criteria then the inspection of the turbulence contour plots (Fig 4.6, 4.13,4.21,4.27,4.33 and 4.39) all show comparable turbulence properties in the wake of the cylinder.

5.7 Summary of the different hybrid methodologies

In conclusion, it can be shown that the Hybrid model with no viscosity modifier is the best solution to this problem. The turbulence predictions are close to those from the Smagorinsky model and there is the least amount to do to make the model work. The implementation of the interface is trivial as the variables are simply passed across leaving the two modelling methodologies to utilise the predicted velocities

The av3 model also predicts the turbulence well, and the use of averaging at the interface means that the RANS sub-domain is able to dissipate the turbulence that is predicted by the LES model. The averaging at the interface also means that the RANS model is passed the mean velocity as it is expecting and so the methodology of Reynolds averaging is upheld.

The Hybrid and Newhybrid models, though similar in concept, give differing results. The hybrid model with the lower limit gives good prediction of the turbulence and velocities at the expense of having to calculate the movement of a user scalar, which, with the current implementation gives a minimum viscosity as the modifier places a lower bound on the turbulent viscosity. In addition it does not enforce any mass balance for the user scalar. This is visible in that the value at the inlet is that of the laminar viscosity and at the outlet is the laminar plus turbulent viscosity which means that there is a nett gain of scalar in the model. Although this does not affect the flow of the fluid, this may be seen as untidy. Another important issue will be the spread of the scalar in the domain as its diffusion across the direction of flow will have an impact on the shearing that may be captured.

5.8 Estimated speedup from the ‘Hybrid’ model

Clearly there will be a CPU overhead from using a commercial code. Although CFX4.4 is linked at runtime the implicit nature of the code will increase the communications overhead and the extra user scalars required for the transport of the variables combined with the superposition of one modelling technique on another will mean that there is an increase in computational cost over the standard k-ε model, which as a rule of thumb consumes approximately 10% of the CPU time for a simulation [118]

It is striking that the CPU time for the k-ε model is so much less than for any of the Smagorinsky models. This is due, in the most part, for the final solution being steady. The solution is not changing each timesteps and so the convergence is very quick. This is contrary to the LES which takes more iterations each timestep to converge. The second major influence will be that lack of user scalars in the k-ε simulation. This means that the wake of the cylinder is not changing as there is no vortex shedding and so that the simulation needs only one or at most two iterations per timestep. As the CPU times listed are for the final experimental run when the data is collected this is most pronounced as the flow is not developing any more. Contrary to this the LES and the ‘hybrid’ models that have captured the vortex shedding and because of the transient nature of the flow there are more iterations required each timestep to resolve the flow to the desired accuracy.

It can also be seen that the load balancing on the processors is reasonably good. The number of cells for each processor is about 60 000, and the number of dummy nodes is comparable for all except the first two processors where it is 50% higher due to the communication between the two domains above and below the cylinder. The results of this are born out by the comparable times of the separate nodes in the parallel operation having comparable CPU times.

There is possibly a small effect on node4 of the misalignment of the i j and k

directions in the final two blocks of the av3 simulation. The longest direction has become the k and not the i direction and in doing so this may hinder effective vectorization. It may be the case that this was not seen for these results as the number of cells on processor 4 is not near its effective operating limit and so the effective use of local memory is less of an issue. (Though still important)

The CPU time is presented as reported by CFX4. However, it is unclear from the documentation as to whether this is the time spent in calculation only or whether this is also the time spent where the processor is idle but has been reserved by CFX whilst waiting for another processor to finish. This can clearly be seen in the comparison of CPU times for the LES sub-domains as these vary by as much as half as much again between the av3 and NewHybrid simulations for processor1, which is performing the same task on the same size grid in each case. For this reason it would be wrong to compare the times on processor 3 and remark on how much faster the Newhybrid model is compared to the others and how there can be seen to be a good speedup in the processing time when compared with the Smagorinsky model.

If the CPU time reported is that of the slowest processor then it would be assumed that the hybrid models would all be approximately the same as the slowest node would be the LES sub-domains around the cylinder as this is where there is the most communication cost, the largest number of wall nodes, the fastest change in the simulation requiring the most iterations and the largest use of user Fortran as it is running the LES code over the top of the k-ε model.

Based on the assumption that the time spent in the k-ε model in an 'average' simulation is no more than 10% of the total CPU time of the simulation. The positioning of the Smagorinsky model over the k-ε model and the use of further user Fortran to monitor the flow would still take only a small fraction of the total processor time. If it is assumed that the total time spent running the turbulence models in the 'hybrid' simulation is less than 20% of the total CPU time, then to try to reduce this would appear to be the speeding up of the fastest part of the program, and in addition the operation of the code as only an LES model would remove the need for a turbulence model at all and could reduce the CPU time further. However, the use of a LES model to capture 80% of the energy in the flow would require many more cells than has been used in the studies in this thesis, which would again increase the CPU time of the simulation.

If the number of cells in the LES sub-domain were quadrupled in each direction then that would increase the number of cells to 1 920 000 cells. To do this for the whole domain (excluding the av3 model) would put the total number of cells at 2 784 000 cells. Even with the amount of time spent in the turbulence modelling subroutines being less than 20% of the total time the vast number of cells included in the simulation mean that the CPU time will go up by an estimated factor of 12, with the assumption that each processor could process 16 times the number of cells with linear speedup and no additional memory requirements (which is unlikely, as the increase per processor from 60 000 cells to 960 000 cells is a considerable increase).

If, however, the 'hybrid' model were used with the current (grid independent) grid for the k-ε model used for the downstream part of the model, then the CPU time increase is only increased by a factor of 9, which is a 25% saving over the complete LES model, demonstrating the facility of placing more cells where the flow is changing faster and of more interest but being able to model the flow for the rest of the domain with adequate accuracy to predict turbulence.

Were this to be used in a mixing vessel where only one quarter of the volume is simulated using the LES model and the rest the k-ε model. Assuming as before that there is a 16 fold increase in the number of cells in the LES region then the speedup of the hybrid model over a complete LES simulation could be up to fourfold. If the volume used for the LES region were only one tenth of the vessel size then there could be a tenfold increase in the speed compared to an LES simulation.

The estimates of speedup are based upon the assumption of scalability of the code and the memory requirements.

Chapter 6

Conclusions

In this thesis I have demonstrated that it is possible to split up a domain into two parts, such that one has the flow modelled using the Smagorinsky subgrid scale model and the LES modelling technique, whilst the other part of the domain uses the Reynolds averaging approach and the k-ε model to model the turbulence.

The Hybrid LES/RANS model has been validated against high accuracy computational data for the vortex shedding behind a cylinder at $Re = 3\,900$, which is a well documented case for unsteady flow with separation which is difficult to predict because of the curvature of the cylinder. The hybrid models compare well to a large eddy simulation performed on the same grid. The velocity profiles compare well to the higher accuracy LES simulation and the turbulence predictions are qualitatively the same.

This technique been accomplished on a parallel machine whereby the different modelling approaches are used on different processors. The different models have explicitly been used on different processors and the interface designed so that it could be extended for a larger domain with variable numbers of processors in each sub-domain.

The Hybrid LES/RANS model has been compared with both the LES and the k-ε model for the same simulation. The computational time of the hybrid model is comparable to that of the LES simulation, but waiting time for the processor is measured as computational time. It is estimated that for a carefully placed interface for the hybrid model there would be a large reduction in the time taken for the hybrid model compared to the large eddy simulation of the same domain.

In addition a new application of a method for averaging the results has been used that would be equally applicable if the flow were not periodic. The development of this model and its use has been demonstrated, both that the turbulent fluctuations are captured, and that it is the turbulence and not the amplitude of the periodic flow that is measured.

Chapter 7

Further work

In order to evaluate the usefulness of the hybrid model further, it should be evaluated on different domains. Initially the domain could be enlarged. This will test the speedup of the model more thoroughly and will also allow the effect of the boundary conditions to be assessed. The performance of the hybrid models could be assessed also with different turbulence models for the RANS sub-domain in case there are numerical gains to be obtained by passing different variables across the interface. If the $k - \varepsilon$ model were implemented directly into the user fortran then the user would have a greater control of the variables passed at the interface and this may facilitate understanding of the model.

Once a greater understanding of the model is obtained then the model will have to be tested on different geometries to compare against LES and RANS models, and ideally against experimental results as well. This will allow the model to be tested in both simple geometries and also more complex cases to measure the speedup and the performance of the model with combustion or chemical reaction as well as the hybrid model. These more complex cases will test the action of the interface on user scalars other than those used for the hybrid model.

Ideally the interface should be modified such that the grid on either side of the interface no longer has to remain uniform. This could either be achieved by the use of an unmatched grid, or by transferring the model into an unstructured grid format.

However, before the model can become very useful the boundary conditions for the flow from the RANS sub-domain into the LES sub-domain will have to be investigated and demonstrated. This will then fully allow the modelling of stirred tank reactors.

Finally it may be desirable to allow for the dynamic movement of the interface depending on the predicted turbulence level. If the interface were to move depending on the turbulence level then the model would become useful for more dynamic simulations. For example, the effect of lorries turning corners, and changing from a headwind to a sidewind could be modelled, and the interface would move to optimise the LES resource usage as the wake changes shape.

Appendix A

List of user scalars

USER SCALAR1	'USRDCC U AVERAGE'
USER SCALAR2	'USRDCC V AVERAGE'
USER SCALAR3	'USRDCC W AVERAGE'
USER SCALAR4	'USRDCC K AVERAGE'
USER SCALAR5	'USRDCC U FLUCT'
USER SCALAR6	'USRDCC V FLUCT'
USER SCALAR7	'USRDCC W FLUCT'
USER SCALAR8	'USRDCC K FLUCT'
USER SCALAR9	'USRDCC SGS VISCOSITY'
USER SCALAR10	'USRDCC VISN'
USER SCALAR11	'USRDCC TAU_ MID'
USER SCALAR12	'USRDCC TAU'
USER SCALAR13	'USRDCC TMULT'
USER SCALAR14	'MODIFIER'
USER SCALAR15	'USRDCC LES EPS AVG'
USER SCALAR16	'USRDCC LES EPS INST'
USER SCALAR17	'YPLUS'

Appendix B

CFX4.4 specific files used

B.1 Sample command file for the simulation

```
>>CFX4
/*for the calcuator the names represent ...*/
/*the inlet plane velocity*/
/*the frequency of oscillation as calculater from St=l/tU*/
/*period of the oscillation*/
/*no of steps/cycle*/
/*size of the timesteps*/
/*number of oscillations to be modelled*/
/*number of steps for the whole simulation*/
/*or could use CYCLES*STEPS !*/
/*number of iterations/step*/
#CALC
  DENSITY  =   1000;
  VISCOSITY =  0.001;
  VELOCITY =   0.1772727;
  STROUHAL =   0.22;
  DIAMETER =   0.022;
  PERIOD   =   DIAMETER/(STROUHAL*VELOCITY);
  FREQ =       1/PERIOD;
  STEPS =      80;
  STEPSIZE =   PERIOD/STEPS;
  CYCLES =     10.0;
  NO_OF_STEPS= CYCLES*STEPS;
  ITER =       30;
  REYNO =      1000*VELOCITY*DIAMETER/VISCOSITY;
#ENDCALC /********************************************
Changes this command file Restarted from m16.
********************************************/
  >>SET LIMITS
```

```
    TOTAL INTEGER WORK SPACE 25000000
    TOTAL CHARACTER WORK SPACE 9000
    TOTAL REAL WORK SPACE 55000000
    MAXIMUM NUMBER OF BLOCKS 16
    MAXIMUM NUMBER OF PATCHES 100
    MAXIMUM NUMBER OF INTER BLOCK BOUNDARIES 25
    END
  >>OPTIONS
    TURBULENT FLOW
    ISOTHERMAL FLOW
    INCOMPRESSIBLE FLOW
    TRANSIENT FLOW
    USER SCALAR EQUATIONS 17
  >>USER FORTRAN
    USRTRN
    USRVIS
  >>VARIABLE NAMES
    USER SCALAR1 'USRDCC U AVERAGE'
    USER SCALAR2 'USRDCC V AVERAGE'
    USER SCALAR3 'USRDCC W AVERAGE'
    USER SCALAR4 'USRDCC K AVERAGE'
    USER SCALAR5 'USRDCC U FLUCT'
    USER SCALAR6 'USRDCC V FLUCT'
    USER SCALAR7 'USRDCC W FLUCT'
    USER SCALAR8 'USRDCC K FLUCT'
    USER SCALAR9 'USRDCC SGS VISCOSITY'
    USER SCALAR10 'USRDCC VISN'
    USER SCALAR11 'USRDCC TAU_MID'
    USER SCALAR12 'USRDCC TAU'
    USER SCALAR13 'USRDCC TMULT'
    USER SCALAR14 'MODIFIER'
    USER SCALAR15 'USRDCC LES EPS AVG'
    USER SCALAR16 'USRDCC LES EPS INST'
    USER SCALAR17 'YPLUS'
>>MODEL TOPOLOGY
  >>INPUT TOPOLOGY
    READ GEOMETRY FILE
    END

>>MODEL DATA
  >>DIFFERENCING SCHEME
    U VELOCITY 'higher upwind'
    V VELOCITY 'Higher upwind'
    W VELOCITY 'CENTRAL'

  >>SET INITIAL GUESS
    >>INPUT FROM FILE
      READ DUMP FILE
      FORMATTED
  >>WALL TREATMENTS
/*    WALL PROFILE 'LOGARITHMIC'  */
    NO SLIP

  >>PHYSICAL PROPERTIES
    >>FLUID PARAMETERS
      VISCOSITY #VISCOSITY
      DENSITY #DENSITY
    >>SCALAR PARAMETERS
      >>DIFFUSIVITIES
        MODIFIER  0.000001
    >>TURBULENCE PARAMETERS
      >>LOGLAYER CONSTANT
        MODIFIER  1.0
      >>SUBLAYER THICKNESS
        MODIFIER  1.0

    >>TRANSIENT PARAMETERS
      >>FIXED TIME STEPPING
        TIME STEPS #NO_OF_STEPS * #STEPSIZE
        INITIAL TIME 0.000E+00
        NUMBER OF VARIABLES CHANGED ON RESTART
>>SOLVER DATA
  >>PROGRAM CONTROL
    MAXIMUM NUMBER OF ITERATIONS #ITER
    MASS SOURCE TOLERANCE 1.0000E-04

>>CREATE GRID
  >>INPUT GRID
    READ GRID FILE
    END
  >>GRID OPTIONS
    COMPUTE DISTANCES TO WALLS

>>MODEL BOUNDARY CONDITIONS
  >>INLET BOUNDARIES
    PATCH NAME 'INLET'
    NORMAL VELOCITY #VELOCITY
    MODIFIER #VISCOSITY
  >>WALL BOUNDARIES
    PATCH NAME 'WALL_CYL'
>>OUTPUT OPTIONS
  >>DUMP FILE FORMAT
    FORMATTED
    SINGLE PRECISION
  >>DUMP FILE OPTIONS
    INITIAL GUESS
    ALL REAL DATA
    GEOMETRY DATA
  >>DUMP FILE OPTIONS
    TIME STEP 740
    ALL REAL DATA
  >>DUMP FILE OPTIONS
    TIME STEP 760
    ALL REAL DATA
  >>DUMP FILE OPTIONS
    TIME STEP 780
    ALL REAL DATA
  >>DUMP FILE OPTIONS
    FINAL SOLUTION
    ALL REAL DATA
    NO GEOMETRY DATA
  >>PRINT OPTIONS
    >>WHAT
      NO RESIDUAL HISTORY
  >>LINE GRAPH DATA
    XYZ 1.000000E-01 0.000000E+00 5.500000E-03
    EACH TIME STEP
    FILE NAME 'MONITOR.TXT'
    ALL VARIABLES
  >>LINE GRAPH DATA
    XYZ 5.000000E-01 0.000000E+00 5.500000E-03
    EACH TIME STEP
    FILE NAME 'FAR.TXT'
    ALL VARIABLES
>>STOP
```

B.2 Subroutine for the the turbulent viscosity

```
      SUBROUTINE USRVIS(VISN
     +                 ,U,V,W,P,VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL,
     +                  XP,YP,ZP,VOL,AREA,VPOR,ARPOR,WFACT,DIFLAM,
     +                  URFVAR,IPT,IBLK,IPVERT,IPNODN,IPFACN,IPNODF,
     +                  IPNODB,IPFACB,WORK,IWORK,CWORK)
C
C**********************************************************************
C
C   USER SUBROUTINE TO PROVIDE VARIABLE LAMINAR VISCOSITY, IF REQUIRED.
C
C   >>> IMPORTANT                                                   <<<
C   >>>                                                             <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                   <<<
C
C**********************************************************************
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINES
C      CUSR  CVIS
C
C***********************************************************************
C   CREATED
C      22/11/89  ADB
C   MODIFIED
C      08/08/91  IRH  NEW STRUCTURE
C      03/09/91  IRH  CORRECT ARGUMENT LIST AND /CHKUSR/
C      25/09/91  IRH  ADD USEFUL COMMON BLOCKS
C      29/11/91  PHA  UPDATE CALLED BY COMMENT, INCLUDE RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      03/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      03/07/92  DSC  CORRECT COMMON MLTGRD.
C      02/08/93  NSW  INCLUDE DIMENSIONING FOR URFVAR
C      21/09/93  NSW  MODIFY FOR USER TO SET NEW VALUE OF VISCOSITY
C                     TO BE UNDER-RELAXED. CHANGE IVERS TO 4
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      03/03/94  FHW  CORRECTION OF SPELLING MISTAKE
C      22/08/94  NSW  MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      02/07/97  NSW  UPDATE FOR CFX-4
C
C***********************************************************************
C
C   SUBROUTINE ARGUMENTS
C
C  *  VISN   - NEW VALUE OF VISCOSITY
```

```
C     U      - U COMPONENT OF VELOCITY
C     V      - V COMPONENT OF VELOCITY
C     W      - W COMPONENT OF VELOCITY
C     P      - PRESSURE
C     VFRAC  - VOLUME FRACTION
C     DEN    - DENSITY OF FLUID
C     VIS    - OLD VALUE OF VISCOSITY
C     TE     - TURBULENT KINETIC ENERGY
C     ED     - EPSILON
C     RS     - REYNOLD STRESSES
C     T      - TEMPERATURE
C     H      - ENTHALPY
C     RF     - REYNOLD FLUXES
C     SCAL   - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP     - X COORDINATES OF CELL CENTRES
C     YP     - Y COORDINATES OF CELL CENTRES
C     ZP     - Z COORDINATES OF CELL CENTRES
C     VOL    - VOLUME OF CELLS
C     AREA   - AREA OF CELLS
C     VPOR   - POROUS VOLUME
C     ARPOR  - POROUS AREA
C     DIFLAM - LAMINAR DIFFUSIVITY
C     URFVAR - UNDER RELAXATION FACTORS
C
C     IPT    - 1D POINTER ARRAY
C     IBLK   - BLOCK SIZE INFORMATION
C     IPVERT - POINTER FROM CELL CENTERS TO 8 NEIGHBOURING VERTICES
C     IPNODN - POINTER FROM CELL CENTERS TO 6 NEIGHBOURING CELLS
C     IPFACN - POINTER FROM CELL CENTERS TO 6 NEIGHBOURING FACES
C     IPNODF - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTERS
C     IPNODB - POINTER FROM BOUNDARY CENTERS TO CELL CENTERS
C     IPFACB - POINTER FROM BOUNDARY CENTERS TO BOUNDARY FACES
C
C     WORK   - REAL WORKSPACE ARRAY
C     IWORK  - INTEGER WORKSPACE ARRAY
C     CWORK  - CHARACTER WORKSPACE ARRAY
C
C   SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C   BE SET  BY THE USER IN THIS ROUTINE.
C
C
C   NOTE THAT THE USER SHOULD SET THE VALUE REQUIRED FOR THE
C   NEW VISCOSITY. THIS WILL BE UNDER-RELAXED WITH REGARD TO THE OLD
C   VALUE OF VISCOSITY AUTOMATICALLY BY THE PROGRAM.
C
C   NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C   ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C   USER MANUAL.
C
```

```
C***********************************************************************
C
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP
     +       ,LRECT,LCYN,LAXIS,LPOROS,LTRANS
C
      CHARACTER*(*) CWORK
C
C+++++++++++++++++ USER AREA 1 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES
C
      REAL L,DELTA,SMAG,SMAG2,VIS_SGS,CO,ONETHIRD,MU_LAM,YDIST,TAUWALL,
     &     D,YPLUS,APLUS,MU_T
      INTEGER IUDIST
      INTEGER INODE1
      REAL LIMIT, Sij

C
C+++++++++++++++++ END OF USER AREA 1 ++++++++++++++++++++++++++++++++++
C
      COMMON
     + /ALL/    NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM
     + /ALLWRK/ NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE
     + /ADDIMS/ NPHASE,NSCAL,NVAR,NPROP
     +         ,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,NRLIST,NTOPOL
     + /CHKUSR/ IVERS,IUCALL,IUSED
     + /DEVICE/ NREAD,NWRITE,NRDISK,NWDISK
     + /IDUM/   ILEN,JLEN
     + /LOGIC/  LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP
     +         ,LRECT,LCYN,LAXIS,LPOROS,LTRANS
     + /MLTGRD/ MLEVEL,NLEVEL,ILEVEL
     + /SGLDBL/ IFLGPR,ICHKPR
     + /SPARM/  SMALL,SORMAX,NITER,INDPRI,MAXIT,NODREF,NODMON
     + /TRANSI/ NSTEP,KSTEP,MF,INCORE
     + /TRANSR/ TIME,DT,DTINVF,TPARM
C
C+++++++++++++++++ USER AREA 2 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
C     THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C     NO CONFLICT WITH NON-USER COMMON BLOCKS
C
      COMMON
     + /USRWAL/ JWALLO
     & /DOMRUN/ IDOMNO,NPCHDO
C
C+++++++++++++++++ END OF USER AREA 2 ++++++++++++++++++++++++++++++++++
C
      DIMENSION DIFLAM(NVAR,NPHASE),URFVAR(NVAR,NPHASE)
C
      DIMENSION
     + VISN(NNODE,NPHASE)
     +,U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),P(NNODE,NPHASE)
     +,VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),VIS(NNODE,NPHASE)
     +,TE(NNODE,NPHASE),ED(NNODE,NPHASE),RS(NNODE,NPHASE,6)
     +,T(NNODE,NPHASE),H(NNODE,NPHASE),RF(NNODE,NPHASE,4)
     +,SCAL(NNODE,NPHASE,NSCAL)
C
      DIMENSION
     + XP(NNODE),YP(NNODE),ZP(NNODE)
     +,VOL(NCELL),AREA(NFACE,3),VPOR(NCELL),ARPOR(NFACE,3)
     +,WFACT(NFACE)
     +,IPT(*),IBLK(5,NBLOCK)
     +,IPVERT(NCELL,8),IPNODN(NCELL,6),IPFACN(NCELL,6),IPNODF(NFACE,4)
     +,IPNODB(NBDRY,4),IPFACB(NBDRY)
     +,IWORK(*),WORK(*),CWORK(*)
C
C+++++++++++++++++ USER AREA 3 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
C---- AREA FOR USERS TO DEFINE DATA STATEMENTS
C
C+++++++++++++++++ END OF USER AREA 3 ++++++++++++++++++++++++++++++++++
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K)=IPT((K-1)*ILEN*JLEN+(J-1)*ILEN+I)
C
C----VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS=4
      ICHKPR = 1
C
C+++++++++++++++++ USER AREA 4 +++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
       IUSED=1
C
C+++++++++++++++++ END OF USER AREA 4 ++++++++++++++++++++++++++++++++++
C
       IF (IUSED.EQ.0) RETURN
C
C---- FRONTEND CHECKING OF USER ROUTINE
       IF (IUCALL.EQ.0) RETURN
C
C+++++++++++++++++ USER AREA 5 +++++++++++++++++++++++++++++++++++++++++
c    julian 26/6/02
c  parallel execution of the smoothing not working.
c  left with spurious high values of the viscosity but only in the
c   keeps side.
c  there was also no smoothing of the viscosity
```

```
c
c  action taken
c  1, have tried with different geometry to see if the BC's had an
c    effect no FX
c  2, have tried to make the flow go faster so that it is turbulent
c    crashes
c  3, have tried combination of these.
c  4, have investigated k-eps run that I start from.
c   that is OK
c  5, have tried with just k-eps and smag with no smoothing
c   all OK but unsmoothed at interface
c  6 about to put smoothing back in
c   this will go into the critical sections again.
c   7 change from IPNODB to INODE1=IPNODF(IFACE,1)
c    this seems to work
c  8, make the forced values of viscosity more obvious to check
c     that they are coming through.  Esp. INODE3.
c   The smoothing is only happening n the RANS side.  I believe
c    this to be more to do with the geometry than the turbulence
c    model.  I am sure that the smoothing only happens on 1
c    side as I can only 'force' values of viscosity on that
c    side.
c  The results are encouraging, but the viscosity for INODE2 goes
c  negative. I will write out the values both before and after
c  interpolation
c  9, using TEMP.GEO, I have created 2 user2d_inter patches
c    these are numberes 1 and 2 so I can call them explicitly
c    1 is on the RANS block
c    2 is on the LES block
c   In decomposition, both of the USED2D patches are moved onto
c    block 1
c  10, Only put the USER2D patch on block2 and see what happens
c     (again by changing temp.geo)
c    works, am including INODE5 to check that it is OK by
c    forcing the viscosity
c  11, RANS processor does less work so will put smoothing there,
c  am concerned that the VIS for INODE2 is negative.
c    Will also remove smoothing algorithm from LES side.
c  12, changing the smoothing on the RANS side so that it is
c    scaled more evenly between the LES and RANS.  I will put the
c    three nodes at 0.25 0.5 and 0.75 of the distance between them.
c    I am still only averaging  over three nodes as this is the
c    minimum for a block I can do this as I know which will be the
c    higher viscosity because I am putting these on the RANS side.
c  13 writing out the values of k eps and coordinates for INODE2 in
c     the smoothing algorithm.
c    values for variables not realistic, thereforededuce that INODE2
c    is not the dummy cell containing the values from the next domain.
c    kmax=1000 xmin=0.001, xmax=0.011 zmin=-0.0005 zmax=0.0099
```

```
c  14 print out values k, eps, xp,yp,zp for the interface patch node to
c     see if I can use that for smoothing.
c     values for k,eps reasonable, but cannot calculate mu_t from it.
c     values of xp,yp,zp all over the place - But, when I use IBDRY
c     to point to INODE1, INODE1 is next to the interface.
c     Therefore I am disregarding the values of xp,yp,zp and using
c     IBDRY
c  15 Use IBDRY to modify the viscosity with.  Limit the damage by
c    calculating mu_t by k and eps and setting the max possible
c    mu_t to be twice this.'
c  16  23/7/02 Will use a user scalar to smooth the viscosity.  In
c    the LES region the  value of the scalar will be set to the
c    same as the viscosity. In the RANS domain the value of the
c    viscosity will be set to the mean of the scalar and the
c    viscosity of that cell.  The scalar will then be set to the
c    same value.  The scalar will be allowed to diffuse.  When
c    the value of the scalar is >90% of the viscosity in the RANS
c    domain the value will be set to that of the RANS domain and
c    there will be no further change.
c  17  24/7/02 Removing the min/max protection from the RANS domain
c      as I am concerned that the viscosity her eis an order of
c      magnitude lower than the laminar viscosity
c  18  05/08/02 Inserting error traps as I am concerned that the
c      viscosity from the LES is lower than the laminar viscosity
c  19  21/08/02
c      Running on 8 processors in th BOX geometry
c      THis means that I have 4 LES and 4 RANS nodes
c      To get around this I have to change the node number for the
c      critical regions so that this works
c  20  10/09/02
c      Remove references to modifier scalar.  I am trying witht the
c      modification of k and eps instead
c  21  12/09/02
c      Move the calculation of the instantaneous values for
c      epsilon in the LES section to USRVIS. K left in USRTRN as this
c      does not require the calculation of gradients and is not
c      dependent on Sij, or mu_t

c&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c  END COMMENTS SECTION
c&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

C---- SETTING CONSTANTS
      APLUS    = 25.
      CO       = 0.1
      ONETHIRD = 1./3.
      IPHASE   = 1
      MU_LAM   = DIFLAM(1,IPHASE)
      LIMIT    = 0.9
```

```
c echo printing
      IF(NITER.EQ.1)THEN
        WRITE(NWRITE,*)’APLUS’,APLUS
        WRITE(NWRITE,*)’CO’,CO
        WRITE(NWRITE,*)’ONETHIRD’,ONETHIRD
        WRITE(NWRITE,*)’IPHASE’,IPHASE
        WRITE(NWRITE,*)’MU_LAM’,MU_LAM
      ENDIF

C---- RESERVE REAL WORKSPACE FOR VELOCITY GRADIENTS
      CALL SETWRK(’USRVIS’,’WORK  ’,’UGRAD ’,3*NCELL,JUGRAD)
      CALL SETWRK(’USRVIS’,’WORK  ’,’VGRAD ’,3*NCELL,JVGRAD)
      CALL SETWRK(’USRVIS’,’WORK  ’,’WGRAD ’,3*NCELL,JWGRAD)
C
C---- GET ADDRESS FOR DISWAL
      CALL GETADD(’USRVIS’,’GEOM  ’,’DISWAL’,ILEVEL,JDISWA)
C
C---- GETTING SCALAR NUMBERS
      CALL GETSCA(’USRDCC SGS VISCOSITY’,ISC9,CWORK)
      CALL GETSCA(’USRDCC VISN’,ISC10,CWORK)
      CALL GETSCA(’USRDCC TAU_MID’,ISC11,CWORK)
      CALL GETSCA(’USRDCC TAU’,ISC12,CWORK)
c      CALL GETSCA(’MODIFIER’,ISC14,CWORK)
      CALL GETSCA(’USRDCC LES EPS INST’,ISC16,CWORK)

C---- COMPUTE VELOCITY GRADIENTS
      CALL GRADV(’USRVIS’,IPHASE,U(1,IPHASE),V(1,IPHASE),
     +           W(1,IPHASE),WORK(JUGRAD),WORK(JVGRAD),WORK(JWGRAD),
     +           XP,YP,ZP,VOL,AREA,
     +           IBLK,IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB,
     +           WORK,IWORK,CWORK)

c\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
c  this is the start of the critical part of the code

       IF(IDOMNO.LE.1)THEN

c      WRITE (NWRITE,*) ’--ENTERING USRVIS FOR KSTEP,NITER’,KSTEP,NITER
c      WRITE(NWRITE,*) ’NPCHDO’,NPCHDO

       call IPALL(’USER3D_RANS’,’USER3D’,’PATCH’,’CENTRES’,IPT
     *           ,NPT,CWORK,IWORK)
           DO 110 L=1,NPT
             INODE = IPT(L)
c             MU_T=MU_LAM+((0.09*TE(INODE,IPHASE)**2)/ED(INODE,IPHASE))
c             VISN(INODE,IPHASE)=MIN(VIS(INODE,IPHASE),(2*MU_T))
             VISN(INODE,IPHASE)=MAX(VIS(INODE,IPHASE),MU_LAM)
c               VISN(INODE,IPHASE)=MIN(MAX(VIS(INODE,IPHASE),MU_LAM),
c     &                                 (2*MU_T))
```

```
c             IF((SCAL(INODE,IPHASE,ISC14)/VISN(INODE,IPHASE)).GE.LIMIT)
c     &            THEN
c               SCAL(INODE,IPHASE,ISC14)=VISN(INODE,IPHASE)
c             ELSE
c               VISN(INODE,IPHASE)=0.5*(VISN(INODE,IPHASE)+
c     &              SCAL(INODE,IPHASE,ISC14))
c               SCAL(INODE,IPHASE,ISC14)=VISN(INODE,IPHASE)
c             ENDIF
 110   CONTINUE

C        WRITE(NWRITE,*)' NO MODIFIER USED '


cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c endif for IDOMNO
       ENDIF

c this is the end of the code for the rans part
c/////////////////////////////////////////////////

c\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
c this is the start of the code for the LES part
C--------------------------------
C---- SMAGORINSKY SGS VISCOSITY
C--------------------------------
      IF(IDOMNO.GE.2)THEN

c      WRITE (NWRITE,*) '--ENTERING USRVIS FOR KSTEP,NITER',KSTEP,NITER
C      write(NWRITE,*) 'this is domain IDOMNO',IDOMNO
C      WRITE(NWRITE,*) 'NPCHDO',NPCHDO
      CALL IPALL('USER3D_LES','USER3D','PATCH','CENTRES',
     &            IPT,NPT,CWORK,IWORK)

C---- LOOP OVER ALL INTERIOR CELLS
      DO 100 I=1,NPT
         INODE=IPT(I)
         DUDX = WORK(JUGRAD+INODE-1)
         DUDY = WORK(JUGRAD+NCELL+INODE-1)
         DUDZ = WORK(JUGRAD+2*NCELL+INODE-1)
C
         DVDX = WORK(JVGRAD+INODE-1)
         DVDY = WORK(JVGRAD+NCELL+INODE-1)
         DVDZ = WORK(JVGRAD+2*NCELL+INODE-1)
C
         DWDX = WORK(JWGRAD+INODE-1)
         DWDY = WORK(JWGRAD+NCELL+INODE-1)
         DWDZ = WORK(JWGRAD+2*NCELL+INODE-1)
C
```

```
C----- CALCULATION OF THE SMAGORINSKY TERM
         Sij   = 2.0*(DUDX**2 + DVDY**2 + DWDZ**2)
     +           + (DUDY + DVDX)**2 + (DUDZ + DWDX)**2
     +           + (DVDZ + DWDY)**2
         SMAG2 = 2*Sij
         SMAG  = SQRT(SMAG2)
         IF(SMAG.LT.0.0)THEN
          WRITE(NWRITE,*)'(SMAG.LT.0.0)'
        WRITE(NWRITE,*)'XP ',XP(INODE),',   YP ',YP(INODE),'
     & ,   ZP ',ZP(INODE)
       ENDIF
C
C----- CALCULATION OF WALL DAMPING
c         IWALL      = IWORK(JWALLO+INODE-1)
c         IBDRY      = IWALL - NCELL
c         INODE1     = IPNODB(IBDRY,1)
C----- USE EITHER THE TIME AVERAGE OR INSTANTANEOUS SHEAR STRESS
C      FOR WALL DAMPING CALCULATION
C         TAUWALL    = SCAL(INODE1,IPHASE,ISC11)
C         TAUWALL    = SCAL(INODE1,IPHASE,ISC12)
C         YDIST      = WORK(JDISWA+INODE-1)
C         YPLUS      = YDIST/(MU_LAM/MAX(SMALL,DEN(INODE,IPHASE)))
C     +              *SQRT(TAUWALL/MAX(SMALL,DEN(INODE,IPHASE)))
C         D          = 1-EXP(-YPLUS/APLUS)
C
C----- CALCULATION OF FILTERLENGTH
         DELTA    = VOL(INODE)**ONETHIRD
C         L          = D * CO * DELTA

        L = CO * DELTA
C      !RUN WITHOUT WALL DAMPING
c       write(*,*)'calculation of filterlength'
       IF((DELTA.GT.0.1).OR.(DELTA.LT.0.0))THEN
       write(*,*)'DELTA = ',DELTA
       ENDIF
       IF(L.LT.0.0)THEN
        WRITE(NWRITE,*)'(L.LT.0.0)'
        WRITE(NWRITE,*)'XP ',XP(INODE),',   YP ',YP(INODE),'
     & ,   ZP ',ZP(INODE)
       ENDIF
C
C----- CALCULATION OF SGS VISCOSITY
c
         VIS_SGS  = L**2 * SMAG
         IF(VIS_SGS.LT.0.0)THEN
           WRITE(NWRITE,*)'(VIS_SGS.LT.0.0)'
        WRITE(NWRITE,*)'XP ',XP(INODE),',   YP ',YP(INODE),'
     & ,   ZP ',ZP(INODE)
         ENDIF
```

```
c
C----- CALCULATION OF NEW VISCOSITY
c
         VISN(INODE,IPHASE)       = MU_LAM + VIS_SGS*DEN(INODE,IPHASE)
         SCAL(INODE,IPHASE,ISC9)  = VIS_SGS*DEN(INODE,IPHASE)
         SCAL(INODE,IPHASE,ISC10) = VISN(INODE,IPHASE)
c         SCAL(INODE,IPHASE,ISC14) = VISN(INODE,IPHASE)
C-- Calculation of Instantaneous dissipation
         SCAL(INODE,IPHASE,ISC16) = VIS_SGS*Sij*Sij

c
      IF(VISN(INODE,IPHASE).GT. 0.1)Then
      write(NWRITE,*)'VISN',VISN(INODE,IPHASE),
     & '   XP ',XP(INODE),',   YP ',YP(INODE),',   ZP ',ZP(INODE)
     & ,',   TE',TE(INODE,IPHASE),',   ED',ED(INODE,IPHASE)
      ENDIF
      IF(VISN(INODE,IPHASE).LT.MU_LAM)THEN
        WRITE(NWRITE,*)'(VISN(INODE,IPHASE).LT.MU_LAM)'
        WRITE(NWRITE,*)'XP ',XP(INODE),',   YP ',YP(INODE),'
     & ,   ZP ',ZP(INODE)
      ENDIF

      IF(SCAL(INODE,IPHASE,ISC9).LT.0.0)THEN
        WRITE(NWRITE,*)' NEGATIVE SGS VISCOSITY'
        WRITE(NWRITE,*)'XP ',XP(INODE),',   YP ',YP(INODE),'
     & ,   ZP ',ZP(INODE)
      ENDIF

100   CONTINUE


ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c  endif for IDOMNO
      ENDIF

c //////////////////////////////////////////

C---- DELETE REAL WORKSPACE FOR VELOCITY GRADIENTS
      CALL DELWRK('USRVIS','WORK  ','UGRAD ')
C    DELWRK DELETES EVERYTHING ABOVE LABEL 'UGRAD'
c  --------------------------------------------------------------------

C
C+++++++++++++++++ END OF USER AREA 5 ++++++++++++++++++++++++++++++++++
C
      RETURN
      END
```

B.3 Subroutine for averaging variables

```
      SUBROUTINE USRTRN(U,V,W,P,VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL,
     +                  XP,YP,ZP,VOL,AREA,VPOR,ARPOR,WFACT,CONV,IPT,
     +                  IBLK,IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB,
     +                  WORK,IWORK,CWORK)
C
c
c
C**********************************************************************
C
C   USER SUBROUTINE TO ALLOW USERS TO MODIFY OR MONITOR THE SOLUTION AT
C   THE END OF EACH TIME STEP
C   THIS SUBROUTINE IS CALLED BEFORE THE START OF THE RUN AS WELL AS AT
C   THE END OF EACH TIME STEP
C
C   >>> IMPORTANT                                                   <<<
C   >>>                                                             <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                   <<<
C
C**********************************************************************
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINES
C      CUSR  TRNMOD
C
C***********************************************************************
C   CREATED
C      27/04/90  ADB
C   MODIFIED
C      05/08/91  IRH  NEW STRUCTURE
C      01/10/91  DSC  REDUCE COMMENT LINE GOING OVER COLUMN 72.
C      29/11/91  PHA  UPDATE CALLED BY COMMENT, ADD RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      05/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      03/07/92  DSC  CORRECT COMMON MLTGRD.
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      22/08/94  NSW  MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      02/07/97  NSW  UPDATE FOR CFX-4
C      02/07/99  NSW  INCLUDE NEW EXAMPLE FOR CALCULATING FLUX OF A
C                     SCALAR AT A PRESSURE BOUNDARY
C
C***********************************************************************
C
C   SUBROUTINE ARGUMENTS
C
```

```
C     U      - U COMPONENT OF VELOCITY
C     V      - V COMPONENT OF VELOCITY
C     W      - W COMPONENT OF VELOCITY
C     P      - PRESSURE
C     VFRAC  - VOLUME FRACTION
C     DEN    - DENSITY OF FLUID
C     VIS    - VISCOSITY OF FLUID
C     TE     - TURBULENT KINETIC ENERGY
C     ED     - EPSILON
C     RS     - REYNOLD STRESSES
C     T      - TEMPERATURE
C     H      - ENTHALPY
C     RF     - REYNOLD FLUXES
C     SCAL   - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP     - X COORDINATES OF CELL CENTRES
C     YP     - Y COORDINATES OF CELL CENTRES
C     ZP     - Z COORDINATES OF CELL CENTRES
C     VOL    - VOLUME OF CELLS
C     AREA   - AREA OF CELLS
C     VPOR   - POROUS VOLUME
C     ARPOR  - POROUS AREA
C     WFACT  - WEIGHT FACTORS
C     CONV   - CONVECTION COEFFICIENTS
C
C     IPT    - 1D POINTER ARRAY
C     IBLK   - BLOCK SIZE INFORMATION
C     IPVERT - POINTER FROM CELL CENTERS TO 8 NEIGHBOURING VERTICES
C     IPNODN - POINTER FROM CELL CENTERS TO 6 NEIGHBOURING CELLS
C     IPFACN - POINTER FROM CELL CENTERS TO 6 NEIGHBOURING FACES
C     IPNODF - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTERS
C     IPNODB - POINTER FROM BOUNDARY CENTERS TO CELL CENTERS
C     IPFACB - POINTER FROM BOUNDARY CENTERS TO BOUNDARY FACESS
C
C     WORK   - REAL WORKSPACE ARRAY
C     IWORK  - INTEGER WORKSPACE ARRAY
C     CWORK  - CHARACTER WORKSPACE ARRAY
C
C   SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C   BE SET  BY THE USER IN THIS ROUTINE.
C
C   NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C   ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C   USER MANUAL.
C
C**********************************************************************
C
C
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP
     +       ,LRECT,LCYN,LAXIS,LPOROS,LTRANS
```

```
C
      CHARACTER*(*) CWORK
C
C++++++++++++++++ USER AREA 1 ++++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES

      CHARACTER*1 CDUM
      REAL        TKE_INSTANT,EPS
C
C++++++++++++++++ END OF USER AREA 1 +++++++++++++++++++++++++++++++++++
C
      COMMON
     + /ALL/    NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM
     + /ALLWRK/ NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE
     + /ADDIMS/ NPHASE,NSCAL,NVAR,NPROP
     +         ,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,NRLIST,NTOPOL
     + /CHKUSR/ IVERS,IUCALL,IUSED
     + /CONC/   NCONC
     + /DEVICE/ NREAD,NWRITE,NRDISK,NWDISK
     + /IDUM/   ILEN,JLEN
     + /LOGIC/  LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP
     +         ,LRECT,LCYN,LAXIS,LPOROS,LTRANS
     + /MLTGRD/ MLEVEL,NLEVEL,ILEVEL
     + /SGLDBL/ IFLGPR,ICHKPR
     + /SPARM/  SMALL,SORMAX,NITER,INDPRI,MAXIT,NODREF,NODMON
     + /TIMUSR/ DTUSR
     + /TRANSI/ NSTEP,KSTEP,MF,INCORE
     + /TRANSR/ TIME,DT,DTINVF,TPARM
C
C++++++++++++++++ USER AREA 2 ++++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
C     THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C     NO CONFLICT WITH NON-USER COMMON BLOCKS

      COMMON
     + /USRPIP/ PIPER,PIPEL
     + /USRWAL/ JWALLO
     + /USRMID/ MIDSTART
     + /USRSPC/ JYPLUS,JTAUX,JTAUY,JTAUZ,JTAU,JUWALL,JVWALL
     +         ,JWWALL,JPRESS
     +         ,JNODPL,JNODPT,JIUW,NMAX,NIUW
      COMMON
     & /UCAVERAGE/ EWMA,AMWE
     & /DOMRUN/ IDOMNO,NPCHDO
C
C++++++++++++++++ END OF USER AREA 2 +++++++++++++++++++++++++++++++++++
C
      DIMENSION
     + U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),P(NNODE,NPHASE)
     +,VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),VIS(NNODE,NPHASE)
     +,TE(NNODE,NPHASE),ED(NNODE,NPHASE),RS(NNODE,NPHASE,6)
     +,T(NNODE,NPHASE),H(NNODE,NPHASE),RF(NNODE,NPHASE,4)
     +,SCAL(NNODE,NPHASE,NSCAL)
      DIMENSION
     + XP(NNODE),YP(NNODE),ZP(NNODE)
     +,VOL(NCELL),AREA(NFACE,3),VPOR(NCELL),ARPOR(NFACE,3)
     +,WFACT(NFACE),CONV(NFACE,NPHASE)
     +,IPT(*),IBLK(5,NBLOCK)
     +,IPVERT(NCELL,8),IPNODN(NCELL,6),IPFACN(NCELL,6),IPNODF(NFACE,4)
     +,IPNODB(NBDRY,4),IPFACB(NBDRY)
     +,IWORK(*),WORK(*),CWORK(*)
      DIMENSION SGNWL(6)
C
C++++++++++++++++ USER AREA 3 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
      DIMENSION XYZ(3),XPL(200),YPL(200)

C---- AREA FOR USERS TO DEFINE DATA STATEMENTS

      DATA IFIRST/0/
      SAVE IFIRST,NODPL1,NODPL2,NODPL3,NODPL4,NODPL5,NODPL6
     + ,NODPL7,NODPL8
     + ,XPL,YPL
     + ,NODPT1,NODPT2,NODPT3,NODPT4,NODPT5

C
C++++++++++++++++ END OF USER AREA 3 ++++++++++++++++++++++++++++++++++
C
      DATA SGNWL / 1.0, 1.0, 1.0, -1.0, -1.0, -1.0 /
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K)=IPT((K-1)*ILEN*JLEN+(J-1)*ILEN+I)
C
C----VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS=3
      ICHKPR = 1
C
C++++++++++++++++ USER AREA 4 +++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
       IUSED=1
C
C++++++++++++++++ END OF USER AREA 4 ++++++++++++++++++++++++++++++++++
C
       IF (IUSED.EQ.0) RETURN
C
```

```
C---- FRONTEND CHECKING OF USER ROUTINE
       IF (IUCALL.EQ.0) RETURN
C
C++++++++++++++++ USER AREA 5 +++++++++++++++++++++++++++++++++++++++++
c  1  Input the values of k calculated in the LES model, to overwrite
c     the values of the TE given by the k-epsilon model.  If this
c     fails that I will put the routine into usrvis.  This will
c     require an IF statement to ensure that the overwriting only
c     happens in the LES region.
c_NOTE_ corrected the formulation of K instant and K fluctuating.
c_NOTE_ in order to reference the domains the common block /DOMRUN/
c       must be used.
c  2  In order to calculate a value for epsilon to communicate
c     with the k-eps model it is necessary to calculate the gradients
c     of the velocity
c  3  gradients calculating OK.  Use these values in the loop to
c     calculate the value of epsilon
c  4  Instead of calculating the value of epsilon explicitly
c     I will calculate it from the viscosity and turbulence levels
c     and then overwrite the value in the LES domain with this lower
c     epsilon.  This will then be convected/diffused by the k-eps
c     equation and may set to modify the values of mu_t in the RANS
c     domain
c  5  Stopped this as it was leading to crashes.  Latest idea is to
c     use a user scalar which is set to the vscosity of the cell in the
c     LES domain but to modify the viscosity of the cell in the RANS
c     domain.  The value of the scalar will then be modified itself,
c     and in this way the effect of this scalar will only be felt
c     near the interface  This will be implimented in USRVIS 23/7/02
c  6  26/7/02 Inputting calculation for the courant number into the
c     code.  Putting into USRTRN as it only needs to be calculated
c     once per time step
c  7  29/07/02  Insert error trappings so that if the courant number is
c     about to divide by zero, the answer is set so that it does not
c     affect the results
c
c  8  04/09/02
c     Calculate the instantaneous values for k and epsilon and store
c     these at cell centres. Then average them with the EWMAmethod
c     Then overwrite the k-epsilon model values.
c  9  10/09/02
c     Removing references to modifier to see what happens if I
c     modify k and epsilon
c  10 Move the instantaneous calculation ofi epsilon to
c     USRVIS.  Only the averaging will be done here.  K left in USRTRN as this
c     does not require the calculation of gradients and is not
c     dependent on Sij, or mu_t
c     19/09/02
c     tidy up the formula and insert the code to detect the inter-node
```

```
c     boundaries in the RANS section.
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
c      WRITE(NWRITE,*) 'ENTERING USRTRN FOR KSTEP',KSTEP
C
C
C----------------------------------------------------------------------
C
C---- USER PARAMETERS FOR OUTPUT
C     --------------------------
C
C     NDTPRT   : NUMBER OF TIME STEPS BETWEEN CONSECUTIVE WRITES
C                OF PROFILES
C     MIDSTART : NUMBER OF TIME STEPS BEFORE STARTING CALCULATING
C                LES STATISTICS
C     TPARM    : TIME PARAMETER IN CRANCK NICOLSON
C                SET TO 0.6 TO DIFFUSE CHEQUERBOARDING
C     NPROF    : NUMBER OF PROFILES FOR OUTPUT
C     NPOINT   : NUMBER OF POINTS FOR OUTPUT
C     EWMA     : WEIGHTING FACTOR IN MOVING AVERAGE
C
      NDTPRT   = 200
      MIDSTART = 1
C      TPARM    = 0.6
      NPROF    = 8
      NPOINT   = 5
      IPHASE   = 1
      EWMA     = 0.7
      AMWE     = (1-EWMA)
      IPHASE   = 1
C
c -------------------
c  call getvar to find the variable number for the scalar array
c  so that I may calculate the gradient of the array
      CALL GETVAR('USRTRN','SCAL  ',I1)

c  get the scalar numbers for the scalars
c  remember to include the new scalars in the command file
c  first the scalars for the averages that were set up in usrtrn_1

      CALL GETSCA('USRDCC U AVERAGE',ISC1,CWORK)
      CALL GETSCA('USRDCC V AVERAGE',ISC2,CWORK)
      CALL GETSCA('USRDCC W AVERAGE',ISC3,CWORK)
      CALL GETSCA('USRDCC K AVERAGE',ISC4,CWORK)
      CALL GETSCA('USRDCC LES EPS AVG',ISC15,CWORK)
```

```
c
c  now the numbers for the fluctuating componennts
c  that are used in this routine
c
      CALL GETSCA('USRDCC U FLUCT',ISC5,CWORK)
      CALL GETSCA('USRDCC V FLUCT',ISC6,CWORK)
      CALL GETSCA('USRDCC W FLUCT',ISC7,CWORK)
      CALL GETSCA('USRDCC K FLUCT',ISC8,CWORK)
      CALL GETSCA('USRDCC LES EPS INST',ISC16,CWORK)

c  space must be allocated to put the values of the gradients in
      CALL SETWRK('USRTRN','WORK  ','UGRAD ',3*NCELL,JUGRAD)
      CALL SETWRK('USRTRN','WORK  ','VGRAD ',3*NCELL,JVGRAD)
      CALL SETWRK('USRTRN','WORK  ','WGRAD ',3*NCELL,JWGRAD)

c  find scalar numbers for the fluctuating component that
c  is to have the gradient found
      IEQN_UF = I1+ISC5
      IEQN_VF = I1+ISC6
      IEQN_WF = I1+ISC6

C---- COMPUTE VELOCITY GRADIENTS
      CALL GRADV('USRTRN',IPHASE,U(1,IPHASE),V(1,IPHASE),
     +           W(1,IPHASE),WORK(JUGRAD),WORK(JVGRAD),WORK(JWGRAD),
     +           XP,YP,ZP,VOL,AREA,
     +           IBLK,IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB,
     +           WORK,IWORK,CWORK)

C--------------------
c  open a file so that the information about
c  each cell can be written there
      OPEN(UNIT=50,STATUS='UNKNOWN',FILE='VIS.TXT',FORM='FORMATTED')

C----------------------------------------------------------------------
c  calculation of the courant number which will give an
c  indication of the stability of the problem

      IF (KSTEP.EQ.1) THEN
c           to IPALL to calculate for all of the blocks
          CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
         COURMAX=-999
         COURMIN=999
c         WRITE(NWRITE,*)'SMALL',SMALL
c         WRITE(NWRITE,*)'DT   ',DT
         DO 120 I =1,NPT
           INODE = IPT(I)
                 INODEX1 = IPNODN(NCELL,1)
                 INODEZ1 = IPNODN(NCELL,2)
                 INODEY1 = IPNODN(NCELL,3)
```

```
                  DX = ABS(XP(INODEX1)-XP(INODE))
                  DY = ABS(YP(INODEY1)-YP(INODE))
                  DZ = ABS(ZP(INODEZ1)-ZP(INODE))
         IF(DX.LE.SMALL)THEN
          WRITE(NWRITE,*)'DX',DX
         ENDIF
          IF(DY.LE.SMALL)THEN
         WRITE(NWRITE,*)'DY',DY
         ENDIF
          IF(DZ.LE.SMALL)THEN
         WRITE(NWRITE,*)'DZ',DZ
         ENDIF
                  COURX = DX/(ABS(U(INODE,1))+SMALL)/DT
                  COURY = DY/(ABS(V(INODE,1))+SMALL)/DT
                  COURZ = DZ/(ABS(W(INODE,1))+SMALL)/DT
         IF(COURX.LT.SMALL)THEN
           WRITE(NWRITE,*)'COURX',COURX
           COURX = 1.0
         ENDIF
         IF(COURY.LT.SMALL)THEN
           WRITE(NWRITE,*)'COURY',COURY
           COURY = 1.0
         ENDIF
         IF(COURZ.LT.SMALL)THEN
           COURZ = 1.0
           WRITE(NWRITE,*)'COURZ',COURZ
         ENDIF
                  COURMIN = MIN(COURMIN,COURX)
                  COURMIN = MIN(COURMIN,COURY)
                  COURMIN = MIN(COURMIN,COURZ)
         IF(COURMIN.LT.SMALL)THEN
           WRITE(NWRITE,*)'COURMIN',COURMIN
         ENDIF
                  COURMAX = MAX(COURMAX,COURX)
                  COURMAX = MAX(COURMAX,COURY)
                  COURMAX = MAX(COURMAX,COURZ)
         IF(COURMAX.LT.SMALL)THEN
           WRITE(NWRITE,*)'COURMAX',COURMAX
         ENDIF

 120     CONTINUE
         WRITE(NWRITE,*) 'MAX COURANT NUMBER= ', 1/COURMIN
         WRITE(NWRITE,*) 'MIN COURANT NUMBER= ', 1/COURMAX
      ENDIF

c---------------------------------------------------------------------
c      write(*,*)'LES STATISTICS'
c           WRITE(NWRITE,*)'IDOMNO',IDOMNO,'   < '
```

```
c  Julian 10/01/02
c  call Ipall to get the variables for all of the cell centres
c  This will allow the variables to be used
c
      CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK
     &             ,IWORK)

      IF (KSTEP .EQ. MIDSTART) THEN
         DO 246 K = 1, NPT
            INODE = IPT(K)
            SCAL(INODE,IPHASE,ISC1)  = U(INODE,IPHASE)
            SCAL(INODE,IPHASE,ISC2)  = V(INODE,IPHASE)
            SCAL(INODE,IPHASE,ISC3)  = W(INODE,IPHASE)
            SCAL(INODE,IPHASE,ISC4)  = TE(INODE,IPHASE)
            SCAL(INODE,IPHASE,ISC5)  = U(INODE,IPHASE)/20
            SCAL(INODE,IPHASE,ISC6)  = V(INODE,IPHASE)/20
            SCAL(INODE,IPHASE,ISC7)  = W(INODE,IPHASE)/20
            SCAL(INODE,IPHASE,ISC8)  = TE(INODE,IPHASE)/20
            SCAL(INODE,IPHASE,ISC15) = ED(INODE,NPHASE)
            SCAL(INODE,IPHASE,ISC16) = ED(INODE,NPHASE)
 246     CONTINUE
         WRITE(NWRITE,*)'USER SCALARS INITIALISED'
       ENDIF
c
c-------------------
c  now if the value is higher and I wish to start averaging
c  then I use my averaging algorithms here

        IF (KSTEP .GE. MIDSTART) THEN

            DO 15 I=1,NPT
               INODE = IPT(I)
c-----------------
c  Calculate the fluctuating u v and w vwlocities, as the difference
c  between the instantaneous and the mean
c
        SCAL(INODE,1,ISC5) = U(INODE,IPHASE) -  SCAL(INODE,1,ISC1)
        SCAL(INODE,1,ISC6) = V(INODE,IPHASE) -  SCAL(INODE,1,ISC2)
        SCAL(INODE,1,ISC7) = W(INODE,IPHASE) -  SCAL(INODE,1,ISC3)
c
c-----------------
c  Calculate the new average velocity as from the EWMA algorithm
c
        SCAL(INODE,1,ISC1) = EWMA*SCAL(INODE,1,ISC1)+
     &                          AMWE*U(INODE,IPHASE)
        SCAL(INODE,1,ISC2) = EWMA*SCAL(INODE,1,ISC2) +
     &                          AMWE*V(INODE,IPHASE)
        SCAL(INODE,1,ISC3) = EWMA*SCAL(INODE,1,ISC3) +
     &                          AMWE*W(INODE,IPHASE)
```

```
c      WRITE(NWRITE,*)'CALCULATED NEW AVERAGE VELOCITY'
c           WRITE(NWRITE,*)'IDOMNO',IDOMNO,' <<'
c
c-----------------
c  Calculate k from the instantaneous velocities
c   This is valid for the LES region
c
         TKE_INSTANT = 0.5*(SCAL(INODE,1,ISC5)**2 +
     &                      SCAL(INODE,1,ISC6)**2 +
     &                      SCAL(INODE,1,ISC7)**2 )
c
c----------------
c  Calculate k fluctuating and k average as before
c
         SCAL(INODE,1,ISC4) = EWMA*SCAL(INODE,1,ISC4) +
     &                       AMWE*TKE_INSTANT
         SCAL(INODE,1,ISC8) = TKE_INSTANT - SCAL(INODE,1,ISC4)

c
c----------------
c  Calculate the average value for epsilon
        SCAL(INODE,IPHASE,ISC15) = EWMA* SCAL(INODE,IPHASE,ISC15)
     &                          + AMWE* SCAL(INODE,IPHASE,ISC16)

c-----------------
c  Overwrite the value of TE and ED in the LES region only
c
      IF(IDOMNO.GE.2)THEN
         TE(INODE,IPHASE)=MAX(TE(INODE,IPHASE),SCAL(INODE,1,ISC4))
         ED(INODE,IPHASE)=MAX(ED(INODE,IPHASE),SCAL(INODE,IPHASE,ISC15))

c endif for IDOMNO
      ENDIF
c
c
 15   CONTINUE


C--overwrite the average values of velocity for the fluctuating
C  value of velocity at the interface

c      CALL IPREC('SOLID 1','BLOCK','CENTRES',IPT
c     &           ,ILEN,JLEN,KLEN,CWORK,IWORK)
c      IF(IDOMNO.EQ.3)THEN
c      ICC = 50
c       DO 103 KCC = 1,KLEN
c         DO 102 JCC = 1,JLEN
c           INODE = IP(ICC,JCC,KCC)
c           U(INODE,IPHASE) =  0.5
```

```
c           V(INODE,IPHASE) = 1
c           W(INODE,IPHASE) = 0
c         ICC=40
c           INODE = IP(ICC,JCC,KCC)
c           U(INODE,IPHASE) =  0.5
c           V(INODE,IPHASE) = 1
c           W(INODE,IPHASE) = 0

c 102     CONTINUE
c 103   CONTINUE



C endif for: IF(IDOMNO.LE.1)THEN
c      ENDIF

c----------------
C  endif for: IF(KSTEP.GT.MIDSTART)
       ENDIF

************************************************************************
c  call IPALL again and this time use the results to write out values
c  to the file UNIT=50

      CALL IPALL(’*’,’*’,’BLOCK’,’CENTRES’,IPT,NPT,CWORK,IWORK)
         WRITE(50,499)’VIS’,’XP’,’YP’,’ZP’
c         WRITE(NWRITE,*)’in IPALL’
       DO 206 I=1,NPT
          INODE=IPT(I)
          WRITE(50,500)VIS(INODE,1)
     &          ,XP(INODE),YP(INODE),ZP(INODE)
 206   CONTINUE

 499   FORMAT(A6,A16,A16,A16,A16)
 500   FORMAT(F12.8,4X,F12.8,6X,F10.6,6X,F10.6,6X,F10.6)

c      IF(IDOMNO.EQ.1)THEN
c  close the file used for writing to
      CLOSE(UNIT=50)
c      ENDIF

C-----------------------------------------------------------------------
C
c----------
c   delete the space created fro the gradients
       CALL DELWRK(’USRTRN’,’WORK  ’,’UGRAD ’)
C
c      WRITE(NWRITE,*) ’----------EXITING USRTRN----------’
```

```
C++++++++++++++++ END OF USER AREA 5 ++++++++++++++++++++++++++++++++++
C
      RETURN
      END
```

B.4 Subroutine to add body forces

```
      SUBROUTINE USRSRC(IEQN,ICALL,CNAME,CALIAS,AM,SP,SU,CONV
     +                 ,U,V,W,P,VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL
     +                 ,XP,YP,ZP,VOL,AREA,VPOR,ARPOR,WFACT,IPT
     +                 ,IBLK,IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB
     +                 ,WORK,IWORK,CWORK)
C
C**********************************************************************
C
C   UTILITY SUBROUTINE FOR USER-SUPPLIED SOURCES
C
C   >>> IMPORTANT                                                   <<<
C   >>>                                                             <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                   <<<
C
C**********************************************************************
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINES
C      CUSR  SCDF  SCDS  SCED  SCENRG  SCHF  SCMOM  SCPCE  SCSCAL
C      SCTE  SCVF
C
C***********************************************************************
C   CREATED
C      08/03/90  ADB
C   MODIFIED
C      04/03/91  ADB  ALTERED ARGUMENT LIST.
C      28/08/91  IRH  NEW STRUCTURE
C      28/09/91  IRH  CHANGE EXAMPLE + ADD COMMON BLOCKS
C      10/02/92  PHA  UPDATE CALLED BY COMMENT, ADD RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      03/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      07/12/93  NSW  INCLUDE CONV IN ARGUMENT LIST AND CHANGE IVERS
C                     TO 4
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      03/03/94  FHW  CORRECTION OF SPELLING MISTAKE
C      08/03/94  NSW  CORRECT SPELLING
C      09/08/94  NSW  CORRECT SPELLING.
C                     MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA.
C                     INCLUDE COMMENT ON MASS SOURCES.
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      02/07/97  NSW  UPDATE FOR CFX-4
C
C***********************************************************************
C
C   SUBROUTINE ARGUMENTS
```

```
C
C     IEQN   - EQUATION NUMBER
C     ICALL  - SUBROUTINE CALL
C     CNAME  - EQUATION NAME
C     CALIAS - ALIAS OF EQUATION NAME
C     AM     - OFF DIAGONAL MATRIX COEFFICIENTS
C     SU     - SU IN LINEARISATION OF SOURCE TERM
C     SP     - SP IN LINEARISATION OF SOURCE TERM
C     CONV   - CONVECTION COEFFICIENTS
C     U      - U COMPONENT OF VELOCITY
C     V      - V COMPONENT OF VELOCITY
C     W      - W COMPONENT OF VELOCITY
C     P      - PRESSURE
C     VFRAC  - VOLUME FRACTION
C     DEN    - DENSITY OF FLUID
C     VIS    - VISCOSITY OF FLUID
C     TE     - TURBULENT KINETIC ENERGY
C     ED     - EPSILON
C     RS     - REYNOLD STRESSES
C     T      - TEMPERATURE
C     H      - ENTHALPY
C     RF     - REYNOLD FLUXES
C     SCAL   - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP     - X COORDINATES OF CELL CENTRES
C     YP     - Y COORDINATES OF CELL CENTRES
C     ZP     - Z COORDINATES OF CELL CENTRES
C     VOL    - VOLUME OF CELLS
C     AREA   - AREA OF CELLS
C     VPOR   - POROUS VOLUME
C     ARPOR  - POROUS AREA
C     WFACT  - WEIGHT FACTORS
C
C     IPT    - 1D POINTER ARRAY
C     IBLK   - BLOCK SIZE INFORMATION
C     IPVERT - POINTER FROM CELL CENTERS TO 8 NEIGHBOURING VERTICES
C     IPNODN - POINTER FROM CELL CENTERS TO 6 NEIGHBOURING CELLS
C     IPFACN - POINTER FROM CELL CENTERS TO 6 NEIGHBOURING FACES
C     IPNODF - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTERS
C     IPNODB - POINTER FROM BOUNDARY CENTERS TO CELL CENTERS
C     IPFACB - POINTER FROM BOUNDARY CENTERS TO BOUNDARY FACESS
C
C     WORK   - REAL WORKSPACE ARRAY
C     IWORK  - INTEGER WORKSPACE ARRAY
C     CWORK  - CHARACTER WORKSPACE ARRAY
C
C   SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C   BE SET  BY THE USER IN THIS ROUTINE.
C
C   NOTE THAT WHEN USING MASS SOURCES, THE FLOWS THROUGH MASS FLOW
```

```
C   BOUNDARIES ARE UNCHANGED. THE USER SHOULD THEREFORE INCLUDE AT
C   LEAST ONE PRESSURE BOUNDARY FOR SUCH A CALCULATION.
C
C   NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C   ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C   USER MANUAL.
C
C***********************************************************************
C
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP
     +       ,LRECT,LCYN,LAXIS,LPOROS,LTRANS
C
      CHARACTER*(*) CWORK
      CHARACTER     CNAME*6, CALIAS*24
C
C+++++++++++++++++ USER AREA 1 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES
C
      REAL USRSOLID11(400),USRSOLID12(400)

C+++++++++++++++++ END OF USER AREA 1 ++++++++++++++++++++++++++++++++++
C
      COMMON
     + /ALL/    NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM
     + /ALLWRK/ NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE
     + /ADDIMS/ NPHASE,NSCAL,NVAR,NPROP
     +         ,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,NRLIST,NTOPOL
     + /CHKUSR/ IVERS,IUCALL,IUSED
     + /DEVICE/ NREAD,NWRITE,NRDISK,NWDISK
     + /IDUM/   ILEN,JLEN
     + /LOGIC/  LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP
     +         ,LRECT,LCYN,LAXIS,LPOROS,LTRANS
     + /MLTGRD/ MLEVEL,NLEVEL,ILEVEL
     + /SGLDBL/ IFLGPR,ICHKPR
     + /SPARM/  SMALL,SORMAX,NITER,INDPRI,MAXIT,NODREF,NODMON
     + /TRANSI/ NSTEP,KSTEP,MF,INCORE
     + /TRANSR/ TIME,DT,DTINVF,TPARM
C
C+++++++++++++++++ USER AREA 2 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
C     THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C     NO CONFLICT WITH NON-USER COMMON BLOCKS
C
      COMMON
     &  /UCUSRSOLID/ USRSOLID11,USRSOLID12
      COMMON
     + /USRPIP/ PIPER,PIPEL
     + /USRWAL/ JWALLO
     + /USRMID/ MIDSTART
     + /USRSPC/ JYPLUS,JTAUX,JTAUY,JTAUZ,JTAU,JUWALL,JVWALL
     +         ,JWWALL,JPRESS
     +         ,JNODPL,JNODPT,JIUW,NMAX,NIUW
      COMMON
     & /UCAVERAGE/ EWMA,AMWE
     & /DOMRUN/ IDOMNO,NPCHDO


C+++++++++++++++++ END OF USER AREA 2 ++++++++++++++++++++++++++++++++++
C
      DIMENSION AM(NCELL,6,NPHASE),SP(NCELL,NPHASE),SU(NCELL,NPHASE)
     +,CONV(NFACE,NPHASE)
C
      DIMENSION
     + U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),P(NNODE,NPHASE)
     +,VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),VIS(NNODE,NPHASE)
     +,TE(NNODE,NPHASE),ED(NNODE,NPHASE),RS(NNODE,NPHASE,6)
     +,T(NNODE,NPHASE),H(NNODE,NPHASE),RF(NNODE,NPHASE,4)
     +,SCAL(NNODE,NPHASE,NSCAL)
C
      DIMENSION
     + XP(NNODE),YP(NNODE),ZP(NNODE)
     +,VOL(NCELL),AREA(NFACE,3),VPOR(NCELL),ARPOR(NFACE,3)
     +,WFACT(NFACE)
     +,IPT(*),IBLK(5,NBLOCK)
     +,IPVERT(NCELL,8),IPNODN(NCELL,6),IPFACN(NCELL,6),IPNODF(NFACE,4)
     +,IPNODB(NBDRY,4),IPFACB(NBDRY)
     +,IWORK(*),WORK(*),CWORK(*)
C
C+++++++++++++++++ USER AREA 3 ++++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
C---- AREA FOR USERS TO DEFINE DATA STATEMENTS
C
C+++++++++++++++++ END OF USER AREA 3 ++++++++++++++++++++++++++++++++++
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K)=IPT((K-1)*ILEN*JLEN+(J-1)*ILEN+I)
C
C----VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS=4
      ICHKPR = 1
C
C+++++++++++++++++ USER AREA 4 ++++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
       IUSED=1
C
```

```
C+++++++++++++++++ END OF USER AREA 4 ++++++++++++++++++++++++++++++++++
C
       IF (IUSED.EQ.0) RETURN
C
C---- FRONTEND CHECKING OF USER ROUTINE
       IF (IUCALL.EQ.0) RETURN
C
C---- ADD TO SOURCE TERMS
      IF (ICALL.EQ.1) THEN
C
C+++++++++++++++++ USER AREA 5 +++++++++++++++++++++++++++++++++++++++++

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Julian 20/09/02
C
C  Adding a source term to overwrite the value of the instantaneous
C   velocity with the value of the average velocity.
C  This is only to happen in the last cell of the LES domain next to
C   the RANS Domain
C
C  20/09/02
C   First attempt
C   What I am trying to do:
C   At the boundary between two domains on a parallel run, I would
C   like to overwrite the value of the velocity  with the value
C    of the average velocity.
C   I have been advised that this will not work just by overwriting
C    the node values in USRTRN as the internal CFXroutines would
C    overwrite my overwriting.  This I have tested and found to
C    be true.  The overwriting of cells in the middle of the domain
C    is possible but that on the edge is not.
C   Now comes the role of USRSRC.  What I am going to do is to
C    add on to the source term the difference between the average
C    and the instantaneous value and this will then put the average
C    into the RANS domain.
C   I have been keeping a moving average of the velocities so these
C    are ready to be used.
C   The geometry has changed so that the U velocity is aligned
C    along the K direction in the last two blocks where I want to
C    do the averaging. I am changing the code to reflect the
C    change in alignment of the blocks.  eg the upstream cell
C    is no longer INODE-1 but rather INODE-(ILEN*JLEN)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

c      WRITE(NWRITE,*)'-----ENTERING USRSRC-----'

C-----
C  Get the scalars and the variables that I will need.
```

```
C   The values of the averages are calculated on a
C   moving average with a timescale of one large eddy turnover time.
C
      CALL GETVAR('USRSRC','U     ',IU)
      CALL GETVAR('USRSRC','V     ',IV)
      CALL GETVAR('USRSRC','W     ',IW)

      CALL GETSCA('USRDCC U AVERAGE',ISC1,CWORK)
      CALL GETSCA('USRDCC V AVERAGE',ISC2,CWORK)
      CALL GETSCA('USRDCC W AVERAGE',ISC3,CWORK)

C------
C  Things to initialise on the first timestep only.
C    I need to find the node numbers of the cells where the values are
C    to be overwritten.  From the geometry file and my PRL file I
C    know that it is the high I face of the High I cells in
C    solids 1 & 8.
C    I am using IPREC to gather the node numbers and then write
C    them into a 1D array.  I will then be able to move through
C    this array later and modify the velocities without having
C    to loop through all of the cells
C QUESTION - Do I need to allow space for the dummy nodes as well?
C    I think not but have left space for them just in case.
      IF(KSTEP.EQ.1)THEN

C  Find out which cells are required to be overwritten.
C   Call IPREC for blocks 1 and 8 and get the node numbers.
      CALL IPREC('SOLID 11','BLOCK','CENTRES',IPT
     &           ,ILEN,JLEN,KLEN,CWORK,IWORK)

        KCC = 1
         DO 102 JCC = 1,JLEN
           DO 101 ICC = 1,ILEN
           INODE = IP(ICC,JCC,KCC)
C   Write the node numbers to an array
           USRSOLID11(((KCC-1)*JLEN)+JCC) = INODE
 101     CONTINUE
 102   CONTINUE

      CALL IPREC('SOLID 12','BLOCK','CENTRES',IPT
     &           ,ILEN,JLEN,KLEN,CWORK,IWORK)
       KC = 1
         DO 106 JC = 1,JLEN
           DO 105 IC = 1,ILEN
           INODE = IP(IC,JC,KC)
C   Write the node numbers to an array
           USRSOLID12( (((KC-1)*JLEN)+JC) ) = INODE
 105     CONTINUE
 106   CONTINUE
```

```
C   Put the array in a common block
C  End the work that has to be done on the first timestep
      ENDIF

C-----
C  For successive timesteps,
      IF(KSTEP.GT.(MIDSTART+79))THEN
C   Call the array
c     - it is in a common block so available.

C      ?  Should I do k as well?
C   Add the source term Su = Su + AM() * (AVG - INST)
C
C   Now I loop through the 1D arrays that were created with the node
C    mumbers of the High I cells in blocks 1 and 8.  When the equation
C    number is equal to the equation number then the source term is
C    added.


      DO 220 KOUNT = 1,300
      KNODE = USRSOLID11(KOUNT)
C       First add the source terms to the U velocity
        If(IU.EQ.IEQN)THEN
          SU(KNODE,1) = SU(KNODE,1) +
c     &       ( AM(KNODE,4,1) * (2 - U(KNODE-1,1) ) )
     &       ( AM(KNODE,4,1) * (SCAL(KNODE-300,1,ISC1)-U(KNODE-300,1)))
c             WRITE(NWRITE,*)'MODIFIED U SOLID11'
        ELSEIF(IV.EQ.IEQN)THEN
          SU(KNODE,1) = SU(KNODE,1) +
c     &       ( AM(KNODE,4,1) * ( 1 - V(KNODE-1,1) ) )
     &       ( AM(KNODE,4,1) * (SCAL(KNODE-300,1,ISC2)-V(KNODE-300,1)))
c             WRITE(NWRITE,*)'MODIFIED V SOLID 11'
        ELSEIF(IW.EQ.IEQN)THEN
          SU(KNODE,1) = SU(KNODE,1) +
c     &       ( AM(KNODE,4,1) * (1 - W(KNODE-1,1) ) )
     &       ( AM(KNODE,4,1) * (SCAL(KNODE-300,1,ISC3)-W(KNODE-3001,1)))
c             WRITE(NWRITE,*)'MODIFIED W SOLID 11'
        ENDIF

 220   CONTINUE




      DO 230 JON = 1,300
      JNODE = USRSOLID12(JON)
C       First add the source terms to the U velocity
        If(IU.EQ.IEQN)THEN
```

```
          SU(JNODE,1) = SU(JNODE,1) +
C     &       ( AM(JNODE,4,1) * (2 - U(JNODE-1,1) ) )
     &       ( AM(JNODE,4,1) * (SCAL(JNODE-300,1,ISC1)-U(JNODE-300,1)))
C             WRITE(NWRITE,*)'MODIFIED U SOLID12'
        ELSEIF(IV.EQ.IEQN)THEN
          SU(JNODE,1) = SU(JNODE,1) +
C     &       ( AM(JNODE,4,1) * ( 1 - V(JNODE-1,1) ) )
     &       ( AM(JNODE,4,1) * (SCAL(JNODE-300,1,ISC2)-V(JNODE-300,1)))
C             WRITE(NWRITE,*)'MODIFIED V SOLID12'
        ELSEIF(IW.EQ.IEQN)THEN
          SU(JNODE,1) = SU(JNODE,1) +
C     &       ( AM(JNODE,4,1) * (1 - W(JNODE-1,1) ) )
     &       ( AM(JNODE,4,1) * (SCAL(JNODE-300,1,ISC3)-W(JNODE-300,1)))
C             WRITE(NWRITE,*)'MODIFIED W SOLID12'
        ENDIF

 230   CONTINUE



C Endif for: IF(KSTEP.GE.MIDSTART)THEN
       ENDIF

c      WRITE(NWRITE,*)'-----LEAVING USRSRC-----'
C+++++++++++++++++ END OF USER AREA 5 +++++++++++++++++++++++++++++++++++
      ENDIF
C
C---- OVERWRITE SOURCE TERMS
      IF (ICALL.EQ.2) THEN
C
C+++++++++++++++++ USER AREA 6 ++++++++++++++++++++++++++++++++++++++++++
C+++++++++++++++++ END OF USER AREA 6 +++++++++++++++++++++++++++++++++++
C
      ENDIF
      RETURN
      END
```

Bibliography

[1] C. R. Anderson and B. Reider, M. A high order explicit method for the computation of flow about a circular cylinder. *Journal of Computational physics*, 125:207–224, 1996.

[2] A. Bakker, L.M. Oshinowo, and E. M. Marshall. Use of large eddy simulation to study stirred vessel hydrodynamics. 10^{th} *European conference on mixing*, pages 247–254, 2000.

[3] E. Balaras, C Benocci, and U Piomelli. Finite-difference computations of high reynolds number flows using the dynamic subgrid-scale model. *Theoretical and computational fluid dynamics*, 7:207–216, 1995.

[4] E. Balaras, C. Benocci, and U. Piomelli. Two-layer approximate boundary conditions for large-eddy simulations. *AIAA journal*, 34(6):1111–1119, 1996.

[5] J. Bardina, J. H. Ferziger, and W. C. Reynolds. Improved subgrid-scale models for large-eddy simulation. *AIAA paper*, 80-1357, 1980.

[6] K.J. Bittorf and K Johnson. Computer aided mixing modeling using the galerkin least-squares finite element technique. 2001.

[7] J.P. Boris, F.F. Grinstein, E.S. Oran, and R.L Kolbe. New insights into large eddy simulation. *Fluid dynamics research*, 10:199–228, 1992.

[8] M. Breuer. A challenging test case for large eddy simulation: high reynolds number circular cylinder flow. *International journal of heat and fluid flow*, 21:648–654, 2000.

[9] A. Brucato, M. Ciofalo, F. Grisafi, and R. Tocco. On the simulation of stirred tank reactors via computational fluid dynamics. *Chemical engineering science*, 55:291–302, 2000.

[10] Bulk synchronous parallel at oxford university, www.bsp.org. *BSP web address.*

[11] CFX International, Harwell International Business Park, Oxon, UK. *CFX4.4 Documentation*, 2001.

[12] D. R. Chapman. Computational aerodynamics development and outlook. *AIAA Journal*, 17(12):1293–1313, 1979.

[13] R. A. Clark, J. H. Ferziger, and W. C. Reynolds. Evaluation of subgrid scale models using accurately simulated flow. *Journal of fluid mechanics*, 91:1–16, 1979.

[14] J. M. Coulson and J. F. Richardson. *Chemical engineering, volume 1.* Pergamon, fourth edition, 1993.

[15] B. J. Daly and F. H. Harlow. Transport equations in turbulence. *Physics of fluids*, 13(11):2634, 1970.

[16] L. Davidson. Large eddy simulation: A dynamic one-equation subgrid model for three dimensional recirculating flow. *11th International symposium on turbulent shear flow, Grenoble*, 3:26.1–26.6, 1997.

[17] J.W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large reynolds numbers. *Journal of Fluid Mechanics*, 41:453–480, 1970.

[18] A. Dejoan and R. Schiestel. LES of unsteady turbulence via a one-equation subgrid scale transport model. *International journal of heat and fluid flow*, 23:398–412, 2002.

[19] J. Derksen and H.E.A. Van den Akker. Large eddy simulations of the flow driven by a rushton turbine. *AIChE Journal*, 45(2):209–221, 1999.

[20] J.G.M. Eggels. Direct and large eddy simulation of turbulent fluid flow using the lattice-boltzman scheme. *International journal of heat and fluid flow*, 1995.

[21] J.H. Ferziger. Large eddy simulation:its role in turbulence research. In D.L. Dwoyer, H.Y. Hussaini, and Voigt R.G., editors, *Theoretical approaches to turbulence*, volume 58 of *Applied Mathematical Sciences*, pages 51–72. Springer-Verlag.

[22] S.M. Fokema, M.D.and Kresta and P.E. Wood. Importance of using the correct impeller boundary conditions for cfd simulations of stirred tanks. *Canadian journal of chemical engineering*, 72:177–183, 1994.

[23] J. Franke and W. Frank. Large eddy simulation of flow past a circular cylinder at re = 3900. *Journal of wind engineering and industrial aerodynamics*, 90:1191–1206, 2002.

[24] A. V. Gerbessiotis. Practical considerations of parallel simulations and architecture independent parallel algorithm design. *Journal of parallel and distributed computing*, 53:1–25, 1998.

[25] A. V. Gerbessiotis and Valiant L.G. Direct bulk-synchronous parallel algorithms. *Journal of parallel and distributed computing*, 22:251–267, 1994.

[26] M. Germano. A proposal for a redefinition of the turbulent stresses in the filtered navier-stokes equations. *Physics of Fluids*, 29(7):2323–2324, 1986.

[27] M. Germano. Turbulence: the filtering approach. *Journal of fluid mechanics*, 238:325–336, 1992.

[28] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Centre for Turbulence Research, Proceedings from the summer program 1990*, pages 5–17, 1990.

[29] S. Ghosal, T.S. Lund, P. Moin, and K. Akselvoll. A dynamic localizational model for large-eddy simulation of turbulent flows. *Journal of fluid mechanics*, 286:229–255, 1995.

[30] S. Ghosal and P Moin. The basic equations for the large eddy simulation of turbulent flows in complex geometry. *Journal of Computational physics*, 118:24–37, 1995.

[31] S. Ghosal and M. M. Rogers. A numerical study of self-similarity in a turbulent plane wake using large-eddy simulation. *Physics of Fluids*, 9(6):1729–1739, 1997.

[32] P Godlin, D.W. Zingg, and T.E. Nelson. High-lift aerodynamic computations with one and two-equation turbulence models. *AIAA Journal*, 35(2):237–243, 1997.

[33] E. Goncalves and R. Houdeville. Reassessment of the wall functions approach for RANS computations. *Aerospace science and technology*, 5:1–14, 2001.

[34] G. Grotzbach. Numerical simulation of turbulent temperature fluctuations in liquid metals. *International Journal of heat and mass transfer*, 24:457–490, 1981.

[35] F. Hamba. An attempt to combine large eddy simulation with the k-epsilon model in a channel-flow calculation. *Theoretical and computational fluid dynamics*, 14:323–336, 2001.

[36] K. Hanjalic and B. E. Launder. A reynolds stress model of turbulence ands its application to thin shear flows. *Journal of fluid mechanics*, 52:609, 1972.

[37] K Hanjalić and B.E. Launder. Contribution towards a reynolds-stress closure for low-reynolds-number turbulence. *Journal of fluid mechanics*, 74:593–610, 1976.

[38] J. Hannon. Mixing and chemical reaction in tubular reactors and stirred tanks. *PhD Thesis, Cranfield University*, 1992.

[39] J. Hannon, S. Hearn, L. Marshall, and W. Zhou. Assessment of cfd approaches to redicting fast chemical reactions. *1998, Annual AICHE meeting, Miami Beach Nov 15-20*, Chemical and Biological reactors session, 1998.

[40] J. M. D. Hill, B. McColl, D. C. Stefanescu, M. W. Goudreau, K. Lang, S. B. Rao, T. Suel, T. Tsantilas, and R. H. Bisseling. BSPlib: The BSP programming library. *Parallel computing*, 24:1947–1980, 1998.

[41] J. M. D. Hill and D. B. Skillcorn. lessons learned from implementing BSP. *Future generation computer systems*, 13:327–335, 1997.

[42] S. Jakirlić and K. Hanjalić. A new approach to modelling near wall turbulence energy and stress dissipation. *Journal of fluid mechanics*, 459:139–166, 2002.

[43] J. Jiménez. LES: Where are we and what can we expect. *AIAA Paper*, 98-2891:1–11, 1998.

[44] W. P. Jones and B. E. Launder. The predictions of laminarisation with a two equation model of turbulence. *Int J Heat Mass Transfer*, 15:301, 1972.

[45] G. Kalitzin. Application of turbulence to high-lift airfoils. *Centre for Turbulence Research, Annual Research Briefs*, pages 165–177, 1997.

[46] V Kalro and T. Tezduyar. Parallel 3d computation of unsteady flows around circular cylinders. *Parallel computing*, 23:1235–1248, 1997.

[47] A.N. Kolmogorov. Dissipation of energy in the locally isotropic turbulence. *Proc. R. Soc. Lond.*, 434, 1991.

[48] S. Krajnović and L. Davidson. A mixed one-equation subgrid scale model for large eddy simulation. *International journal of heat and fluid flow*, 23:413–425, 2002.

[49] A. G. Kravchenko and P. Moin. Numerical studies of flow over a circular cylinder at re = 3900. *Physics of fluids*, 12(2):403, 2000.

[50] S. M. Kresta and P. E. Wood. The flow field produced by a pitched blade turbine: characterization of the turbulence and estimation of the dissipation rate. *Chemical engineering science*, 48(10):1761–1774, 1993.

[51] D. Krizanc and A. Saarimaki. Bulk synchronous parallel: practical experience with a model for parallel computing. *Parallel computing*, 25:159–181, 1999.

[52] M. T. Landahl and E. Mollo-Christensen. *Turbulence and random processes in fluid mechanics.* CUP, 2nd edition edition, 1992.

[53] J. A. Langford and R. D. Moser. Optimal LES formulation for isotropic turbulence. *Journal of fluid mechanics*, 398:321–346, 1999.

[54] J. Laufer. The structure of turbulence in a fully developed pipe flow. *National advisory communications on aeronautics*, 1954.

[55] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a reynolds stress turbulence closure. *Journal of fluid mechanics*, 68(3):537–566, 1975.

[56] B. E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Computer methods in applied mathematics and engineering*, 3:269–289, 1974.

[57] C. Lei, L. Cheng, and K. Kavanagh. Spanwise length effects on three-dimensional modelling of flow over a circular cylinder. *Computer methods in applied mechanical engineering*, 190:2909–2923, 2001.

[58] A. Leonard. Energy cascade in large-eddy simulations of turbulent fluid flows. *Advances in Geophysics*, 18:237–248, 1974.

[59] M.A. Leschziner. *Statistical turbulence modelling for the computation of physically complex flows.* Springer-Verlag, first edition, 2000.

[60] M. Lesieur and O. Métais. New trends in large-eddy simulation of turbulence. *Annual review of fluid mechanics*, 28:45–82, 1996.

[61] D.C. Leslie and G.L. Quarini. The application of turbulence theory to the formation of subgrid modelling closures. *Journal of fluid mechanics*, 91:65–91, 1979.

[62] D. K. Lilly. A proposed modification of the germano subgrid-scale closure method. *Physics of fluids, A*, 4:633–635, 1992.

[63] S. Liu, C. Meneveau, and J. Katz. On the properties of similarity subgrid scale models as deduced from measurements in a turbulent jet. *Journal of Fuild Mechanics*, 275:83–119, 1994.

[64] L.M. Lourenco and C. Shih. Characteristice of the plane turbulence near wake of a circular cylinder. a particle image velocimetry study. 2000.

[65] H. Lubcke, St. Schmidt, T. Rung, and F. Thiele. Comparison of LES and RANS in bluff body flows. *Journal of wind engineering and industrial aerodynamics*, 89:1471, 2001.

[66] A. L. Marsden, O. V. Vasilyev, and P. Moin. Construction of commutative filters for LES on unstructured meshes. *Journal of computational physics*, 175:584–603, 2002.

[67] P. J. Mason. Large-eddy simulation: A critical review of the technique. *Quarterly Journal of the Royal Meteological Society*, 120:1–26, 1994.

[68] P.J. Mason and N.S. Callen. On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulations of turbulent channel flow. *Journal of Fluid Mechanics*, 162:439–462, 1986.

[69] J. Mathieu and J. Scott. *An Introduction to turbulent flow.* Cambridge university press, first edition, 2000.

[70] C. Meneveau, T. S. Lund, and W. H. Cabot. A lagrangian dynamic subgrid-scale model of turbulence. *Journal of Fluid Mechanics*, 319:353–385, 1996.

[71] G. Micale, G Montante, F. Grisafi, A. Brucato, and J. Godfrey. CFD simulation of particle distribution in a stirred vessel. *Trans IChemE*, 78:435–444, 2000.

[72] E. Minty, R. Davey, A. Simpson, and D. Henty. *Decomposing the potentially parallel.* Edinburgh parallel computing centre, second edition.

[73] B. Mohammadi and O. Pironneau. *Analysis of the K-Epsilon turbulence model.* research in applied mathematics, first edition, 1994.

[74] P. Moin. Advances in large eddy simulation methodology for complex flows. *International journal of heat and fluid flow*, 23:710–720, 2002.

[75] P. Moin and J. Kim. Numerical investigation of turbulent channel flow. *Journal of fluid mechanics*, 118:341–377, 1982.

[76] K. Ng, N.J. Fentiman, K.C. Lee, and M. Yianneskis. Assessment of sliding mesh cfd predictions and lda measurements of the flow in a tank stirred by a rushton impeller. *Transactions of the IChemE*, 76(A):737–747, 1998.

[77] L. Ong and J. Wallacw. The velocity field of the turbulent very near wake of a circular cylinder. *Experiments in fluids*, 20:441–453, 1996.

[78] M. Pettersson and Rasmuson A.C. Application of three-dimensional phase-doppler anemometry to mechanically agitated crystallizers. *Trans IChemE*, 75(A):132–141, 1997.

[79] F. Peyrin and A. Kondjoyan. Effect of turbulent integral length scale on heat transfer around a circular cylinder placed cross to an air flow. *Experimental thermal and fluid science*, 26:455–460, 2002.

[80] U. Piomelli and E. Balaras. Wall layer models for large eddy simulation. *Annual review of fluid mechanics*, 34:349–374, 2002.

[81] U. Piomelli, E. Balaras, H. Pasinato, K.D. Squires, and P.R. Spalart. The inner-outer layer interface in large eddy simulations with wall layer models. *International Journal of heat and fluid flow*, 24:538–550, 2003.

[82] U. Piomelli, E. Balaras, K.D. Squires, and P.R. Spalart. Zonal approaches to wall-layer models for large eddy simulation. *AIAA*, (2002-3083):1–10, 2002.

[83] U. Piomelli, P. Moin, and J.H. Ferziger. Model consistency in large eddy simulations of turbulent channel flow. *Physics of fluids (A)*, 31:1884, 1988.

[84] S.B. Pope. *Turbulent flows.* CUP, first edition, 2000.

[85] A. Prasad and C. H. K. Williamson. The instability of the separated shear layer from a bluff body. *Physics of Fluids*, 8:1347, 1996.

[86] R. Prasad, A. Bakker, and L. Marshall. Numerical modelling of mixing processes-what can LES offer? *1998, Annual AICHE meeting, Miami Beach Nov 15-20*, Session 238, 1998.

[87] B.K. Revill. *Mixing in the Process Industries.* Butterworth-Heinmann, second edition, 1997.

[88] J. Revstedt, L Fuchs, and C. Tragardh. Large eddy simulations of the turbulent flow in a stirred reactor. *Chemical Engineering Science*, 53(24):4041–4053, 1998.

[89] Richardson. *Weather prediction by numerical process.* Cambridge University Press, first edition, 1922.

[90] C. D. Rielly and A.J. Marquis. A particles eye view of crystallizer fluid mechanics. *Chemical Engineering Science*, 56:2475–2493, 2001.

[91] D. Rocchi and A. Zasso. Vortex shedding from a circular cylinder in a smooth and wired configuration: comparison between 3d LES and experimental results. *Journal of wind engineering and industrial aerodynamics*, 90:475–489, 2002.

[92] W. Rodi. *Turbulence models and their application in hydraulics: A state of the art review.* first edition, 1980.

[93] R.S. Rogallo and P. Moin. Numerical simulation of turbulent flows. *Annual review of fluid mechanics*, pages 99–137, 1984.

[94] J. Rotta. Statistiche theorie nichthomegen turbulenz. *Z Phys.*, 129:547–572, 1951.

[95] V. T. Roussinova and S. Kresta. Study of macro-instabilities in stirred tanks using a velocity decomposition technique. *Transactions of the IChemE*, 78:1040–1052, 2000.

[96] J. U. Schluter, H. Pitsch, and P. Moin. Consistent boundary conditions for integrated LES/RANS simulations: LES outflow conditions. *AIAA*, 2002(3121), 2002.

[97] S. Schmidt and F. Thiele. Comparison of numerical methods applied to the flow over wall mounted cubes. *International journal of heat and fluid flow*, 23:330–339, 2002.

[98] U. Schumann. Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of computational physics*, 18:376–404, 1975.

[99] T. K. Sengupta and M. T. Nair. Upwind schemes and large eddy simulation. *International journal of numerical methods in fluids*, 31:879–889, 1999.

[100] S. Sherwin, R. Darekar, J. Owen, and P. Bearman. Passive bluff body drag reduction and vortex supression due to spanwise waviness. In N.D. Sandham, editor, *Understanding and predicting turbulence and turbulent flows by Direct and Large Eddy Simulation*. UK Turbulence Consortium, April 2001. EPSRC Grant GR/M08424(1998-2001).

[101] A. Silva Lopes and J. M. L. M. Palma. Numerical simulation of isotropic turbulence using a collocated approach and a nonorthogonal grid system. *Journal of computational physics*, 175:713–738, 2002.

[102] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather review*, 91(3):99–164, 1963.

[103] P.R. Spalart. Young-persons guide to detached-eddy simulation grids. *NASA*, (CR-2001-211032):1–18, 2001.

[104] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale*, 1:5–25, 1994.

[105] P.R. Spalart, M. Strelets, and S. Allmaras. Comments on the feasibility of les for wings and on a hybrid rans/les approach. In C. Lui and Z. Lui, editors, *Advances in DNS/LES*. Greyden Press, 1997.

[106] C.G. Speziale. Galilean invariance of subgrid-scale stress models in the large-eddy simulation of turbulence. *Journal of fluid mechanics*, 156:55–62, 1985.

[107] C.G. Speziale. Turbulence modelling for time-dependent RANS and VLES: a review. *Proc.13th AIAA Computational Fluid Dynamics conference*, 2:1071–1090, 1998.

[108] I. Stewart. *Does God play dice?* Penguin, second edition, 1997.

[109] D. Taulbee. *Advances in Turbulence.* Springer-Verlag, first edition, 1989.

[110] P.R.F. Teixeira and A.M. Awruch. Three-dimensional simlation of high compressible flows using a multi time step integration technique with subcycles. *Applied mathematical modelling*, 25:613–627, 2001.

[111] Tennekes and Lumley. *A first course in Turbulence.* MIT Press, fourth edition, 1972.

[112] Chris Thompson. Private communication, 16^{th} september 2002.

[113] A Tiskin. The bulk-synchronous parallel random access machine. *Theoretical computer science*, 196:109–130, 1998.

[114] O. V. Vasilyev, T. S. Lund, and P. Moin. A general clas of commutative filters for LES in complex geometries. *Journal of computational physics*, 146:82–104, 1998.

[115] Versteeg and Malalaskera. *An introduction to CFD, The finite volume method.* Longman, 1995.

[116] P.R. Voke and M.W. Collins. Large eddy simulation: Retrospect and prospect. *PhysicoChemical Hydrodynamics*, 4(2):119–161, 1983.

[117] D.C. Wilcox. *Turbulence modelling for CFD.* DCW Industries, second edition, 1994.

[118] Nigel Wilkes. Private communication, 20^{th} november 2002.

[119] G. H. K. Williamson. Vortex dynamics in the cylinder wake. *Annual review of fluid mechanics*, 28:477, 1996.

[120] H. Wu and G.K. Patterson. Laser-doppler measurements of turbulent-flow parameters in a stirred mixer. *Chemical engineering science*, 44(10):2207–2221, 1989.

[121] Y. Xu and G. McGrath. Cfd predictions of stirred tank flows. *Transactions of the IChemE*, 74(A):471–475, 1996.

[122] G. Zhou and S. M. Kresta. Impact of the tank geometry on the maximum turbulence energy dissipation rate for impellers. *AIChE Journal*, 42(9):2476–2490, 1996.

[123] W.B. Zimmerman. Fluctuations in passive tracer concentration due to mixing by coherent structures in anisotropic homogeneous helical turbulence. In H. Benkreira, editor, *Mixing V*, IChemE Symposium Series No. 140, pages 213–224. IChemE, 1996.