POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale – Dipartimento di Energia

Corso di Laurea in
Ingegneria Energetica – Orientamento Produzione e Conversione

Generalized Wall Functions for RANS Computation of Turbulent Flows

Relatore: Prof. Fabio Inzoli
Correlatore: Prof. Hisashi Ninokata

Tesi di Laurea di:

Matteo MONTICELLI Matr. 750276

Anno Accademico 2010-2011

Contents

Index of Figures

List of tables

Abstract

In the field of Computational Fluid Dynamics (CFD) a challenge is represented by the wall treatment for wall bounded turbulent flows. Indeed the so-called wall blocking effect causes a different behavior of the flow in the near-wall region. There are two methods for dealing with this challenge: Low Reynolds models and Wall Functions. The first method requires a fine near wall mesh in order to solve the entire wall affected region, but this requirement makes the computational time and costs increase. The Wall Functions method, used along with a k-ε turbulence model, allow the user to adopt a coarser near-wall mesh, but the behavior of the flow in the proximity of the wall (in terms of wall shear stress, production of turbulent kinetic energy and turbulent dissipation rate) is estimated through pre-integrated expressions. In Fluent 6.3 there are two Wall Functions available, i.e. Standard Wall Functions (SWF) and Non-Equilibrium Wall Functions (NEWF). However, the pre-integrated expressions used by these two Wall Functions are not universal. An alternative is given by Generalized Wall Functions (GWF), proposed by Popovac and Hanjalic, whose pre-integrated expressions, according to the authors, are based on more general assumptions. To be more specific, the law of the wall adopted by GWF is sensitized to the local non-equilibrium effects of the flow. First of all, a review of challenges related to wall bounded turbulent flows, of *k*-ε turbulence models available in Fluent 6.3 and of Wall Functions approach (including the way SWF and NEWF are implemented into Fluent 6.3) has been presented. The first work made consists of the writing of a user defined function (udf) for the implementation of GWF into Fluent 6.3 (this work includes a deep analysis of the way the macro DEFINE_WALL_FUNCTIONS works). After that, an evaluation of the quality of the results obtained with GWF has been carried out. Their performance has been tested for an axial symmetric conical diffuser geometry. In order to evaluate the goodness of the Wall Functions approach itself, sensitivity analyses (using always SWF) of near wall mesh, core mesh and turbulence model have been made. These analyses have allowed to obtain the best combination of these three parameters in terms of results obtained with a CFD simulation. This combination has been used along with GWF. A sensitivity analysis regarding the velocity scale used inside GWF law of the wall has been carried out. The slopes of the GWF log law of the wall along the diffuser are found out to have a trend similar to experimental slopes. Wall shear stress shows an improvement too, but the accordance to the experimental data is still not perfect. GWF are promising because they have a more physical background and they are easy to implement inside a commercial CFD code. However, their performance must be tested for other benchmarks different from the conical diffuser, like a backward facing step.

Keywords: Wall Bounded Turbulent Flows, Wall Functions, Generalized Wall Functions, DEFINE_WALL_FUNCTIONS

Sommario

Nel campo della fluidodinamica computazionale (CFD), una sfida è rappresentata dal trattamento di parete per fluidi in moto turbolento. La parete, con il cosiddetto effetto di bloccaggio, agisce sul flusso, provocandone una variazione del comportamento nella regione ad essa adiacente. Questo effetto è dovuto alla condizione fisica per cui la velocità sulla parete di un fluido viscoso è rigorosamente nulla (per la condizione di non-scorrimento e per la condizione per cui il fuido non può fisicamente attraversare la parete). Per flussi turbolenti, in cui l'alto numero di Reynolds è sinonimo del fatto che le forze viscose sono trascurabili rispetto alle forze d'inerzia, esiste una regione vicino alla parete dove il numero di Reynolds locale, calcolato utilizzando la distanza da parete come lunghezza caratteristica, rientra nel range laminare e di transizione. Nelle adiacenze della parete quindi la viscosità non è più trascurabile. La regione di parete è suddivisa utilizzando la distanza adimensionale da parete (detta y^+) come parametro. Partendo dalla parete, si incontrano le seguenti zone:

- Sottostrato viscoso: in questa regione la viscosità non è trascurabile; la legge adimensionale di parete è lineare (infatti questa regione è detta anche sottostrato lineare)
- Buffer layer: è la regione di transizione in cui gli effetti viscosi e quelli d'inerzia sono ugualmente importanti
- Regione pienamente turbolenta: è la regione dove gli effetti turbolenti sono prevalenti rispetto a quelli viscosi; una possibile legge di parete in questa regione è rappresentata dalla legge logaritmica; tuttavia la sua esistenza, il suo range di validità e i suoi parametri non sono universali
- Outer region: è la regione successiva alla regione pienamente turbolenta che prosegue fino alla fine dello strato limite; la legge di parete in questa regione è rappresentata dalla cosiddetta velocity defect law

L'effetto di bloccaggio della parete agisce anche sulla turbolenza. Per la condizione di non scorrimento tutte le fluttuazioni turbolente devono andare a zero alla parete; inoltre le fluttuazioni in direzione perpendicolare alla parete sono smorzate e sono redistribuite nelle altre due direzioni; ciò acuisce l'anisotropia della turbolenza nelle vicinanze della parete.

Nel campo della CFD il trattamento della regione di parete comporta delle problematiche legate all'esistenza di elevati gradienti in uno spessore ristretto e all'assegnazione delle condizioni al contorno sulla parete (in particolare per un modello di turbolenza k-ε devono essere assegnati il valore dello sforzo di taglio a parete e le condizioni al contorno legate alle due equazioni di trasporto per k

ed ε). Esistono due famiglie di metodi per affrontare il trattamento della regione di parete: modelli Low Reynolds e Wall Functions.

I modelli Low Reynolds comportano una modifica della equazioni di trasporto delle grandezze turbolente (aggiunta di damping functions, ecc.) e richiedono una mesh fitta nella regione di parete; tuttavia il vincolo sulla mesh causa un aumento dei tempi e dei costi computazionali che può essere non accettabile. Un'alternativa è data dalle cosiddette Wall Functions che, abbinate ad un modello di turbolenza k-ε, consentono all'utente di utilizzare una mesh coarse nella regione di parete, facendo risparmiare tempi e costi computazionali. Tuttavia il metodo delle Wall Functions non è rigoroso in quanto il comportamento del flusso in questa regione è stimato attraverso espressioni pre-integrate, la cui validità non è universale.

L'idea di base delle Wall Functions è quella di porre il primo nodo computazionale nella regione completamente turbolenta, di modo da non dover considerare la regione in cui la viscosità non è trascurabile. Lo sforzo a parete viene assegnato partendo da una legge di parete (ovvero una funzione che lega la velocità adimensionale parallela alla parete con la distanza adimensionale da parete) che viene assunta a priori. N.B. l'adimensionalizzazione della distanza dalla parete e della velocità parallela alla parete è effettuata utilizzando sia u_τ che u_k come scale di velocità, ottenendo le cosiddette grandezze star (*). L'equazione di trasporto per k viene risolta ponendo la diffusione in direzione perpendicolare alla parete nulla e assumendo la produzione e la dissipazione sulla base di espressioni preintegrate. L'equazione per ε non viene risolta nelle celle adiacenti a parete ma viene assegnato il valore della grandezza nel centro cella. Il metodo appena descritto prevede una variazione se il primo centro cella cade nel sottostrato viscoso. Viene assegnato il limite superiore del sottostrato viscoso (in termini di y^*); se il valore di y^* del centro cella è inferiore al limite dato, la legge di parete assunta è lineare. Lo sforzo a parete è ricavato dalla legge di parete lineare mentre il procedimento per le equazioni di trasporto di k ed ε rimane invariato.

In Fluent 6.3 sono disponibili due Wall Functions. Queste sono chiamate Standard Wall Functions (SWF) e Non-Equilibrium Wall Functions (NEWF). Le Standard Wall Functions assumono come legge di parete la seguente legge logaritmica $u^* = \frac{1}{\kappa}\log(E\,y^*)$ in cui κ ed E sono costanti. Per quanto riguarda la produzione di k ed ε, esse sono calcolate utilizzando rispettivamente le seguenti espressioni: $\tau_w \frac{\partial u}{\partial y}$ ed $\frac{u_k}{\kappa y_P}$. Queste due ultime espressioni fanno sì che lo strato limite sia in equilibrio, ovvero la produzione di k e il sua tasso di dissipazione sono uguali. Le espressioni e le assunzioni appena mostrate tuttavia non sono affatto universali. Per esempio le SWF danno risultati non accettabili quando vengono utilizzate in presenza di forti gradienti di pressione, forze di volume o forti effetti tridimensionali. Le NEWF, invece, utilizzano una legge di parete che comprende gli effetti dovuti a gradienti di pressione ed utilizzano un approccio two layer per il calcolo delle grandezze turbolente. Tuttavia anche le assunzioni delle NEWF non sono universali.

Obiettivo della tesi è la valutazione dei risultati ottenuti utilizzando le funzioni di parete chiamate Generalized Wall Functions (GWF), proposte da Popovac & Hanjalic). Queste Wall Functions hanno un background più generale e fisico rispetto alle SWF e NEWF. Il loro punto di partenza è dato dall'equazione della quantità di moto in direzione parallela alla parete (con le ipotesi semplificative di strato limite). Utilizzando un profilo di viscosità turbolenta assunto a priori è possibile ottenere una legge di parete sensibile agli effetti di non equilibrio del flusso. La legge di parete ottenuta è la seguente: $u^* = \frac{1}{\kappa\psi}\log(Ey^*)$. Essa è simile alla legge di parete utilizzata dalle SWF, quindi è relativamente semplice da implementare in un codice CFD commerciale. La novità è la presenza della funzione di non equilibrio Ψ, definita come $\Psi = 1 - \frac{C_U y_p}{\rho U_P \kappa u_k}$. Gli effetti di non equilibrio sono inclusi nel termine C_U, che è definito dalla seguente espressione: $\rho U \frac{\partial U}{\partial x} + \rho V \frac{\partial U}{\partial y} + \frac{\partial p}{\partial x}$. Riassumendo brevemente, le GWF hanno allo stesso tempo due interessanti caratteristiche: assunzioni più generali di base e relativa facilità di implementazione in un codice CFD.

Le GWF sono state implementate nel codice CFD commerciale Fluent 6.3 attraverso la scrittura di una user defined function in linguaggio C++. È stato effettuato uno studio preliminare delle grandezze da calcolare, degli step con cui tali grandezze devono essere calcolate, delle celle interessate dal calcolo, e delle macro offerte da Fluent necessarie per la scrittura della udf. Particolare rilievo nella tesi è stato dato allo studio del funzionamento della Macro DEFINE_WALL_FUNCTIONS (macro che permette la scrittura di una user defined wall functions) e al calcolo della funzione Ψ.

Per quanto riguarda la Macro DEFINE_WALL_FUNCTIONS, uno studio approfondito del suo funzionamento è stato effettuato ed esso è presentato all'interno della tesi. In particolare i seguenti aspetti sono stati studiati:

- Variabili passate dal solutore alla macro
- Quantità che l'utente deve fornire alla macro
- Output della Macro

- Come il codice CFD utilizza l'output della Macro per calcolare le condizioni al contorno a parete (sforzo a parete, produzione di energia cinetica turbolenta e tasso di dissipazione di energia cinetica turbolenta)

Per quanto riguarda il calcolo della funzione Ψ, vi è una problematica legata al termine C_U. I singoli termini che compongono C_U sono definiti per un sistema di riferimento relativo alla parete (nello specifico x è la direzione parallela alla parete mentre y è la direzione perpendicolare ad essa). Tuttavia i dati accessibili da Fluent 6.3 sono definiti per un sistema di riferimento fisso xy. Per cui, nel caso in cui la parete non sia orientata come una direzione del sistema di riferimento fisso xy, è necessario adottare un algoritmo per il calcolo delle grandezze necessarie nel sistema di riferimento legato all'orientamento della parete, partendo dalle grandezze disponibili dal post processing di Fluent. Tale algoritmo è presentato dettagliatamente nel capitolo 6 della tesi.

Le performance delle GWF sono state testate utilizzando come geometria un diffusore conico assial simmetrico di cui sono disponibili dei dati sperimentali (Trupp et al. [1]). Prima di tutto è stata effettuata un'analisi dei dati sperimentali relativi al comportamento del fluido nella vicinanza della parete (utilizzando anche le conclusioni ottenute dagli autori dell'esperimento). I dati sperimentali mostrano che la regione di esistenza (in termini di y^+) della log law è nettamente ridotta rispetto al caso di flusso in un tubo circolare (in cui la log law è valida per valori di y^+ compresi tra circa 30 e 300); inoltre la pendenza della log law non è costante ma varia con la posizione all'interno del diffusore. Quest'ultima caratteristica in particolare porta ad affermare che la legge di parete utilizzata dalle SWF (in cui la pendenza è costante) non è rigorosamente valida per questo caso. Le GWF, in cui la pendenza è funzione degli effetti di non equilibrio nella regione adiacente alla parete, diventano quindi una promettente soluzione.

Una volta definite la geometria del dominio di calcolo e le condizioni al contorno è stato possibile procedere alle varie simulazioni. La logica seguita è di seguito descritta.
Una prima simulazione è stata effettuata utilizzando le SWF, il modello di turbolenza k-ε standard, una mesh di parete per cui la y^+ dei centri cella è compresa tra 30 e 300 (valori consigliati dalla guida di Fluent per l'uso delle SWF) e una core mesh medium (lunghezza caratteristica dell'elemento quadrilaterale pari a un centesimo del diametro di ingresso del diffusore). Lo scopo di questa simulazione è quello di valutare i risultati che si ottengono utilizzando le SWF e i consigli dati dalla guida di Fluent (è ciò che un utente non esperto farebbe). I valori di y^+ dei centri cella delle celle adiacenti a parete escono dal range sperimentale di validità della log law lungo tutto il diffusore. Conseguentemente lo sforzo di taglio a parete ottenuto dalla simulazione è

scorretto (per essere precisi vi è una sovrastima di esso). Le grandezze di cui lo sforzo a parete è funzione sono : la velocità tangenziale alla parete, l'energia cinetica turbolenta e la legge di parete assunta. La funzione utilizzata per il calcolo di esso è la seguente: $\tau_w = \frac{\rho u_w u_k}{f(y^*)}$. I valori sperimentali dell'energia cinetica turbolenta non sono disponibili; sono invece disponibili i valori della velocità tangenziale alla parete nei centri cella delle celle adiacenti alla parete. Una verifica di quest'ultima grandezza mostra che la simulazione CFD fornisce valori scorretti. Sono state esaminate anche altre grandezze i cui valori numerici sperimentali sono disponibili; tra queste vi sono i profili adimensionali di velocità, i valori di pressione statica alla parete, i valori del gradiente cinematico di pressione sull'asse del diffusore e il cosiddetto pressure gradient parameter.

Lo step successivo consiste in un'analisi di sensitività per la mesh di parete, utilizzando sempre le SWF, il modello k-ε standard e la core mesh già utilizzata. Infatti lo scopo della tesi è quello di valutare la bontà delle Wall Functions adottate. È necessario quindi trovare il peso dell'errore nei risultati delle sole WF, riducendo il peso sull'errore causato da altri parametri. Tramite una riduzione dello spessore della mesh di parete, la y^+ entra nel range sperimentale di validità della log law. I risultati mostrano un miglioramento dello sforzo di taglio a parete nella prima parte del diffusore ma vi è un lieve peggioramento nella parte finale. La velocità parallela alla parete è questa volta correttamente stimata, eccetto che per una sottile regione situata subito dopo l'ingresso nel diffusore. In questa regione tale grandezza è sottostimata e ciò causa un peggioramento dello sforzo di taglio a parete. Si è trovato che la mesh di parete non ha significativa influenza su: velocità sull'asse del diffusore, pressione statica sulla parete e gradiente cinematico di pressione. Il pressure gradient parameter migliora, così come i profili adimensionali di velocità.

Il secondo parametro preso in considerazione per un'analisi di sensitività è la core mesh. L'analisi di sensitività per questo parametro è stata effettuata utilizzando le SWF e il modello di turbolenza k-ε standard; la mesh di parete utilizzata è quella che garantisce che tutti i valori di y^+ ricadano nel range di validità della log law (mesh di parete ottenuta dall'analisi di sensitività precedente). Lo scopo principale di quest'analisi è quello di valutare quanto è l'effetto della core mesh sullo sforzo di taglio a parete. Tramite la procedura per il calcolo del GCI è stata calcolata l'ampiezza della fascia di incertezza in cui ricade il valore dello sforzo di taglio mediato sull'intera parete che si otterrebbe con una mesh avente un numero infinito di elementi. L'ampiezza ha un valore percentuale molto ridotto: è allora possibile utilizzare la mesh già utilizzata nelle analisi precedenti (da ora chiamata medium) senza introdurre un errore eccessivo. La core mesh non mostra una significativa influenza sulle altre grandezze, eccetto che per il pressure gradient parameter, che migliora leggermente. Comunque, una veloce analisi del trade off tra il numero di elementi della mesh fine e il lieve miglioramento ottenuto ci porta a non considerare vantaggioso l'utilizzo di tale mesh per le successive analisi.

Utilizzando le SWF, la mesh di parete ottenuta dalla prima analisi di sensitività e la core mesh medium è possibile fare un'analisi di sensitività variando il modello di turbolenza. Sono state effettuate simulazioni utilizzando il modello Realizable e il modello RNG. Lo sforzo di taglio a parete migliora sensibilmente pur utilizzando ancora le SWF come trattamento di parete. Stessa cosa accade per il pressure gradient parameter. Il modello di turbolenza fa variare la velocità parallela alla parete e l'energia cinetica turbolenta.nel primo centro cella La velocità parallela a parete risulta essere sottostimata con i modelli Realizable e RNG, mentre il modello standard la stimava abbastanza correttamente. Possiamo concludere quindi che il miglioramento dei valori dello sforzo di taglio a parete è dovuto a effetti di compensazione tra gli errori di u_k, di u_W e della pendenza della log law. Ciò ci porta a mantenere il modello standard per le successive simulazioni.

Minimizzati gli errori introdotti dai tre parametri sopra descritti, è possibile valutare i risultati ottenuti utilizzando le GWF e le NEWF. N.B. in questo step le GWF sono implementate utilizzando una scala di velocità (che non è identica a quella proposta dagli autori) che favorisce la convergenza. In pratica la funzione Ψ scritta è la seguente: $\Psi = 1 - \frac{y^*}{u^* \kappa} C_U \frac{\mu}{\rho^2 u_k^3}$. Con questa GWF vi è un miglioramento dello sforzo di taglio a parete e del pressure gradient parameter; comunque l'accordo con i dati sperimentali non è ancora perfetto. Inoltre il trend della pendenza della log law è simile al trend sperimentale. L'utilizzo delle NEWF peggiora risultati. Si è trovato inoltre che le grandezze dimensionali praticamente non cambiano con il variare delle Wall Functions adottate.

È interessante valutare l'effetto della scala di velocità presente nella funzione Ψ. Sono state effettuate simulazioni utilizzando GWF che adottano tre diverse scale di velocità, rispettivamente u_k^3, $u_k^2 u_\tau$ e $u_k u_\tau^2$ (la prima è quella utilizzata nella simulazione precedente, mentre l'ultima è quella rigorosamente proposta dagli autori delle GWF). Per quanto riguarda lo sforzo di taglio a parete, l'uso dell'ultima scala lo sottostima nella parte iniziale ma l'accordo con i dati sperimentali migliora verso l'uscita del diffusore. L'utilizzo della scala intermedia cambia poco i valori ottenuti (comunque si apprezza un lieve miglioramento). Le grandezze dimensionali non variano significativamente con la scala di velocità adottata.

Concludendo brevemente, si può affermare che l'utilizzo delle GWF per il trattamento a parete di flussi turbolenti è promettente per il futuro. Tuttavia, le

loro performance devono essere testate per altri benchmark difersi dal diffusore conico, come per esempio il backward facing step.

Parole chiave: Flussi turbolenti delimitati da pareti, funzioni di parete, Generalized Wall Functions, DEFINE_WALL_FUNCTIONS

Introduction

This master thesis has been realized thanks to the partnership between CFD laboratory of Politecnico di Milano, headed by professor Fabio Inzoli, and Ninokata laboratory of Tokyo Institute of Technology, headed by professor Hisashi Ninokata. According to the agreement made by the two partners, the author started his work at Ninokata laboratory in Tokyo (Japan) and the work he made there lasted for three months (from the 1st of July 2011 to the 1st of October 2011). After that, the author continued his work in CFD laboratory in Milan (Italy). The period spent abroad was useful because the author could access the knowledge owned by Ninokata laboratory related to the topic of this master thesis, and because he found a stimulating work environment with skillful and smart individuals. Moreover, the link between the two institutions has been strengthened. This experience allowed also the author to get in contact with a different culture and a different way to approach academic problems, things that allowed him to develop his character.

Nowadays Computational Fluid Dynamics (CFD) has become a powerful and widespread tool both in the academic and in the industrial field. It has had an exponential rise since the 1980's. For example, Versteeg stated in 1995: "The use of CFD to predict internal and external flows has risen dramatically in the past decade. In the 1980's, the solution of fluid flow problems by means of CFD was the domain of the academic, postdoctoral or postgraduate researcher or the similarity trained specialist with many years of grounding in the area. The widespread availability of engineering workstations together with efficient solutions algorithms and sophisticated pre- and post-processing facilities enable the use of commercial CFD codes by graduate engineers for research, development and design tasks in industry" [2]. This situation was the trend in 1995. Now it is still rising; one of the reason of this rise is that "CFD has enabled us to understand the world in new ways" [3]. For example: "CFD allows numerical simulation of fluid flows, results for which are available for study even after the analysis is over. This is a big advantage over, say, wind tunnel testing where analysts have a shorter duration to perform flow measurements" [3]. Or: "CFD allows observation of flow properties at locations which may not be accessible to (or harmful for) measuring instruments. For example, inside a combustion chamber, or between turbine blades" [3]. However, "CFD is not yet at the level where it can be blindly used by designers or analysts without a working knowledge of numeric involved" [3]. Concerning academic field, CFD is useful to enlarge the knowledge of fluid dynamics. Indeed it allows to simulate the behavior of several flows and also it allows to obtain results which sometimes cannot be obtained from experimental data [4].

About industrial field, CFD is used for the fluid dynamic design and optimization of several components. It is used along with experimental analysis, or in some cases in place of it, essentially because of the lower times and costs required. This last aspect is the predominant one for the diffusion of CFD. Indeed a simulation can be launched in a little time, compared to the time necessary to set an experimental facility, and the time to complete a simulation is small too. Moreover the costs of a CFD simulation have lower order of magnitude compared to the costs of an equivalent experiment (for example we can think to the cost for building and setting a wind tunnel compared to the costs for the launch of an equivalent CFD simulation).

Regarding the flows of interest for industrial field, most of them are turbulent wall bounded flows. Turbulence modeling is still a challenge of Computational Fluid Dynamics and it can be considered a bottleneck [5]. For example, when dealing with combustion, if we can model perfectly the chemical reaction but we cannot model properly turbulence the results provided by a CFD simulation are wrong. A family of methods to model turbulence is called RANS (Reynolds Averaged Navier-Stokes equations). This method solves the Navier-Stokes equation averaged with respect to the time; therefore it solves the mean flow. Turbulence effect appears after the time averaging inside momentum equation as a new tensor, similar to stress tensor. This tensor is called Reynolds stress tensor. A subfamily of models moves the attention from the modeling to all the terms of the Reynolds stress tensor to the modeling of only one term (which Reynolds tensor terms are related to), called turbulent viscosity. k-ε turbulence models belongs to this subfamily [6]. A particular aspect, but not negligible, of turbulence modeling is the treatment of the flow behavior in the near wall region. When dealing with wall bounded turbulent flows the presence of the wall causes a different flow behavior in the wall region that affects the flow in the whole domain. The presence of the wall affects the flow because of the kinematic condition (the flow cannot cross the wall, i.e. the velocity on the wall normal to it must be zero) and because of the non-slip condition (in a viscous flow the velocity on the wall of the flow is equal to the velocity of the wall itself). Wall effects on turbulence can be synthetized into two different aspects. First, "the presence of the wall reduces velocity fluctuation normal to the wall; the turbulence tends, at the wall, toward a two component behavior" [4]. Second, "in the wall region, the turbulent kinetic energy goes to zero because of the non-slip condition. Hence, the turbulence activity decreases close to the wall" [4].

There are two main different approaches for the treatment of this region when using a k-ε turbulence model: Low Reynolds models [7] and Wall Functions. In the first family of methods (Low Reynolds models) a finer grid in the wall region is required in order to solve the entire wall-affected region. The grid constraints to be respected are two: first computational node must have a dimensionless wall distance (y^+) value of 1 or less; and there must be more than 10 computational nodes within y^+ 10 (that represents an approximate upper limit of viscous affected region) [5]; this constraints are justified by the presence of strong gradients close to the wall. Moreover, the k-ε turbulence models have been written for fully turbulent flows (i.e. for the region which is not affected by the presence of the wall). With a Low Reynolds model treatment it is therefore necessary to modify the turbulence models. The modifications proposed by one possible Low Reynolds model are [4]:

- ε transport equation is written for a so-called isotropic dissipation ε^* (which is supposed to tends toward zero at the wall);
- extra source terms D and E must be added respectively to the turbulent kinetic energy transport equation and to the turbulent dissipation rate transport equation (in order to ensure the right balances at the wall);
- damping functions must be introduced for turbulent viscosity and for ε dissipation term inside ε transport equation.

On one hand these methods lead to better results because they allow to solve the entire near wall region. On the other hand there are disadvantages. The main one is that grid constraints lead to a grid with many cells, and therefore computational time and costs can significantly rise up. Wall Functions methods follow a different logic [5]. The first computational node is placed in the so called fully turbulent region of boundary layer (e.g. for a pipe flow y^+ lays between 30 and 300) in order to bridge the viscosity affected region (therefore wall region is not entirely solved). The near wall flow behavior must be estimated through pre-integrated expression. An advantage in the use of a Wall Functions method instead of a Low Reynolds model is the possibility to work with a coarser mesh, and this leads to less computational nodes and so computational time and costs are lower. On the other hand, since the assumptions made for the derivation of the pre-integrated expressions are not universal, the use of one kind of Wall Functions for a flow where such assumptions are not valid leads to poor results. Even though Wall Functions method is quite popular between CFD users, not many functions are available. In Fluent 6.3 only two functions are available: Standard Wall Functions (SWF) and Non-Equilibrium Wall Functions (NEWF). SWF are the first (chronologically) Wall Functions proposed (they have been proposed by Launder & Spalding in 1973 [8]). The assumptions Standard Wall Functions are based on are valid only if the boundary layer is in equilibrium (production of turbulent kinetic energy is equal to turbulent dissipation rate) and if the law of the wall in the fully turbulent region is logarithmic. These assumptions are strictly true only in few situations, like pipe flow or channel flow. The only one alternative in Fluent 6.3 to Standard Wall Functions is called Non-Equilibrium Wall Functions (NEWF) [9] These functions have been developed to be used when dealing with pressure gradient affected flows, but the validity of the assumptions used for their derivation is still not universal. The purpose of this master thesis work is the implementation of a new set of Wall Functions, called Generalized Wall Functions (GWF) [10], into Fluent version 6.3. These functions are supposed to have a validity field wider than SWF and NEWF. Although many developed Wall Functions have been proposed through the years (e.g. Analytical Wall Functions (AWF), proposed by Craft et al., 2002 [11]), GWF were chosen basically for two reasons. Firstly, "the single assumption that the non-dimensional eddy viscosity varies linearly with the distance from the wall was found to hold reasonably well even in strongly non equilibrium flows" [10], and "while the [..] assumption for turbulent viscosity may not hold universally, especially in and around singularities such as stagnation, separation and reattachment, it is reasonably general" [10]. Secondly, "The expressions are compatible with the Standard Wall Functions expressions, thus easy implementable in existing CFD codes" [10]. GWF are going to be tested with a k-ε turbulence model and an axisymmetric 2D, steady and incompressible flow. The geometry chosen is a conical diffuser, which presents at the same time three desirable characteristics:

- a simple geometry
- a non-equilibrium boundary layer and an adverse pressure gradient
- it has a practical purpose, i.e. the recovery of pressure at the expense of kinetic energy, and it is used in a lot of situations in industrial field (e.g. it is used in conjunction with gas turbine combustors [12])

1 Wall Bounded Turbulent Flows

1.1 Reynolds Number and Local Reynolds Number

The characteristics of a turbulent flow in the proximity of a wall are different from the characteristics of a turbulent flow in the free stream. The reason of the difference in the behavior can be easily explained. For example, we can look at Reynolds number, which is defined as:

$$Re = \frac{\rho\, v\, L}{\mu} = \frac{\text{inertia forces}}{\text{viscous forces}} \tag{1.1}$$

In a turbulent flow the global Reynolds number is computed using a length scale L taken in the flow direction (or pipe diameter for pipe flows), and the number results to be very high. This means that inertia forces are much greater than viscous forces; thus viscosity can be neglected. But if viscosity is neglected in the whole domain, then the flow presents the so-called D’Alembert paradox [13]. The paradox states that for an inviscid and incompressible flow the drag force is zero on a body moving with constant velocity relative to the fluid. But zero drag is in direct contradiction to the observation of the reality, which shows that the drag on a moving body is different from zero, and it is higher with the velocity of it (and consequently with the Reynolds number). We must conclude that, even if Reynolds number is high enough to neglect viscosity, there must be a region close to the wall where viscosity cannot be neglected. A practical consequence of it is that there are two length scales for the evaluation of the Reynolds number. The first is the length scale L above mentioned, while the second is the wall distance. Therefore close to the wall we can compute a local Reynolds number (Re_y), using the second length scale:

$$Re_y = \frac{\rho\, v\, y}{\mu} = \frac{\text{inertia forces}}{\text{viscous forces}} \tag{1.2}$$

There will be a range of values of wall distance y for which Re_y is small enough to make viscous properties predominant; the flow in that wall distance range behaves close to laminar. Moving further away from the wall there will be a region where viscous and inertial properties of the flow are both important. Moving further there will be a region where the turbulent properties of the flow play a major role. The flow in the near wall region will be affected by viscous forces, and not by free stream parameters. Therefore, if the viscosity in the near wall region becomes significant, while it is negligible in free stream region; it is expectable that the behavior of the turbulent flow changes when approaching to the wall. Same conclusion can be made if we think to non-slip condition, which is directly related to a fluid whose viscosity is not negligible. The condition states that the velocity of the flow on the wall is equal to the velocity of the wall itself.

$$(\boldsymbol{V} \cdot \boldsymbol{w})_{\text{wall}} = V_{\text{wall}} \tag{1.3}$$

Therefore if the wall is stationary, also the flow close to the wall is stationary. There will be a region where the Reynolds number, this time computed with local velocity and using the scale length L discussed before, falls into the transitional or laminar range discussed above. Another difference in flow behavior between free stream and near wall region is due to the so-called kinematic condition. This condition is a direct consequence of the non-permeability of the wall. Indeed, the flow cannot cross the wall, and mathematically this means that the velocity on the wall normal to the wall must be zero.

$$(\boldsymbol{V} \cdot \boldsymbol{n})_{\text{wall}} = 0 \tag{1.4}$$

1.2 Law of the Wall

What we want to do now is to find a relationship between the wall tangential velocity and the wall distance inside wall affected region. This relationship is called law of the wall. First thing to do is to discuss the parameters that can appear inside the law of the wall. Since we stated that the flow in wall region is not affected by free stream parameters, mean flow velocity will be function of wall region parameters and of fluid properties. These quantities are: wall distance, density, dynamic viscosity and wall shear stress. The general expression of the law of the wall is then:

$$U = f(y, \rho, \mu, \tau_w) \tag{1.5}$$

By using Buckingham π theorem [14], which comes from dimensional analysis, we can reduce a function of n dimensional quantities (which are expressible in terms of m independent fundamental physical quantities) in a function of *n-m* dimensionless quantities. What we obtain after applying Buckingham theorem to our function is:

$$\frac{U}{u_\tau} = f\left(\frac{\rho u_\tau y}{\mu}\right) \tag{1.6}$$

$$u^+ = \frac{U}{u_\tau} \tag{1.7}$$

$$y^+ = \frac{\rho u_\tau y}{\mu} \tag{1.8}$$

This function (1.6), whose explicit expression is still unknown, is called law of the wall. It relates dimensionless wall tangential velocity, defined by equation (1.7), with dimensionless wall distance, defined by equation (1.8). It is important to point out the definition of the velocity scale present in the law of the wall. This is called friction velocity and its definition contains wall shear stress:

$$u_\tau = \sqrt{\left(\frac{\tau_w}{\rho}\right)} \tag{1.9}$$

Furthermore we can notice that the expression of dimensionless wall distance is very similar to the expression of local Reynolds number, which is defined by equation (1.2). The only difference is that, instead of using a velocity scale representing the free stream velocity, it is used an inner velocity scale (a boundary layer quantity). Using the dimensionless wall distance, it is possible to divide wall affected region into sub-regions. The first one (closer to the wall) is called sub-viscous layer (or also linear sub-layer) and has values of y^+ between 0 and about 10. This is the region where viscous properties are predominant and so flow behaves close to laminar. Actually the upper limit of sub-viscous layer is not univocally defined, and different values can be found in the literature (e.g. Versteeg and Malalasekera set the upper limit to 5 [2], while Fluent User Guide set the upper limit to 11.225 [15]). Since this layer results to be very thin, in order to find an explicit expression for the law of the wall it is possible to use the assumption that shear stress inside the layer is constant and equal to wall shear stress.

$$\tau(y) = \mu\,\frac{\partial U}{\partial y} \approx \tau_w \tag{1.10}$$

Integrating this expression with respect to y and using the non-slip condition as boundary condition

$$U_{y=0} = 0 \tag{1.11}$$

We obtain a relationship between dimensionless quantities

$$\int_0^y \tau_w dy = \int_{y=0}^{y} \mu dU \tag{1.12}$$

$$U = \frac{\rho y}{\mu}\left(\frac{\tau_w}{\rho}\right) \tag{1.13}$$

$$U = \frac{\rho y u_\tau}{\mu} u_\tau \tag{1.14}$$

$$\frac{U}{u_\tau} = y^+ \tag{1.15}$$

$$u^+ = y^+ \tag{1.16}$$

In viscous sub-layer there is a linear relationship between wall tangential velocity and distance from the wall. That's the reason why sub-viscous layer is also called linear sub-layer. Moving further outside, there is the region where inertial properties become to be predominant. This sub-region is therefore called fully turbulent layer. Under some assumptions one possible relationship between dimensionless velocity and dimensionless wall distance can be written:

$$u^+ = \frac{1}{A}\log(y^+) + B \tag{1.17}$$

This expression is called logarithmic law of the wall (log-law); A and B are two constants that must be determined experimentally. If the flow actually follows the logarithmic law, fully turbulent region is called logarithmic region or log-law region. The limits of existence of logarithmic region are not general at all. For example, for situations like pipe flow or channel flow, the limits are usually set for a range of y^+ between 30 and 300 (some authors set the upper limit to 500; however the upper limit is not universal too but it changes with Reynolds number and with the geometry studied [5]). However, it is very important to point out that this law of the wall is not valid universally. It can happen that in some situations the logarithmic region does not even exist, or the width (in terms of y^+) of this region is smaller than the cases of pipe flow or channel flow. Furthermore, even if the logarithmic region exists, the constants A and B can change with the case studied (they are not universal too).

There is not a sharp distinction between sub-viscous layer and fully turbulent region. As said before, between these two regions there is a part where viscous and inertial effects are of the same order of magnitude, and therefore they are both important. This region is called buffer (or blending) region, for which there is not a well-defined relationship between $u^+$and y^+. However, an expression can be obtained from an interpolation of experimental data. These three regions put together compose the so-called inner layer.
Figure 1.1 shows a typical dimensionless velocity profile close to the wall. Usually the *x-axis* on the plot (where there are reported the values of y^+) is logarithmic. This chose is made in order to increase the visibility of the near-wall region, since this region starts from 0 and goes until a value of y^+ on the order of magnitude of 10^3.

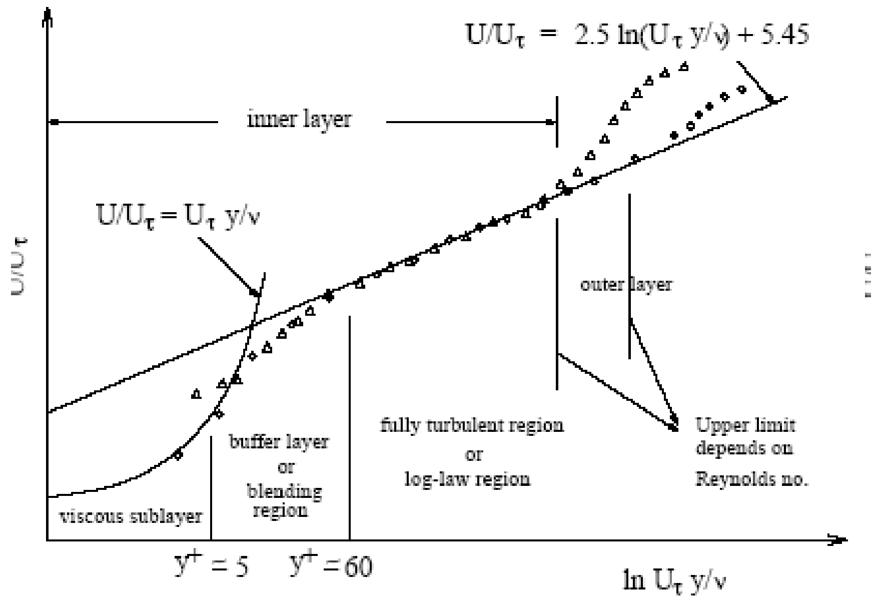

Figure 1.1. Dimensionless wall tangential velocity profile for a turbulent flow close to a solid wall

As we can see, in this situation experimental data (the black dots) fits very well the log law for y^+ values greater than 60. Close to the wall we can see that the flow respects a linear relationship between velocity and wall distance (note: the logarithmic scale of *x-axis* makes it look as an exponential relationship). The passage between sub-viscous layer and logarithmic region is gradual and the transition region called buffer layer or blending region can be noticed.

Experimental measurements show that the log law is valid in the region [16]

$$0.02 < \frac{y}{\delta} < 0.2 \tag{1.18}$$

In the expression (1.18), δ represents the thickness of boundary layer, evaluated as the distance where the velocity is equal to 99% of the free stream or axis velocity. For greater values of y, we fall into the outer layer, where the so-called velocity defect law is the one that represents better the relationship between mean velocity and wall distance. The defect law has the following general expression:

$$\frac{(U_{max} - U)}{u_{\tau}} = g(\frac{y}{\delta}) \tag{1.19}$$

In this equation U is the free stream velocity or the axis velocity (for axisymmetric flows). As we can see, neither the viscosity nor the density is a variable in the relationship. The meaning of velocity defect law is clear if we view the wall shear stress as the cause of a velocity deficit (compared to free stream or axis velocity) which decreases the closer we get to the edge of the boundary layer (or to pipe centerline).
Summarizing, the turbulent boundary layer adjacent to a solid surface can be divided into the two main following regions:

- Inner region: it occupies the 10%-20% of the total thickness of boundary layer and it is in turn subdivided into sub-viscous layer, buffer region and fully turbulent region
- Outer region: inertia dominated core flow; free from direct viscous effects

1.3 Effects of the Presence of the Wall on the Turbulence

The wall affects also the turbulent behavior of the flow. Indeed just the presence of the wall modifies the turbulent fluctuations of velocity in the three directions. Figure 1.2 shows dimensionless route mean square values of turbulent velocity fluctuation in the three directions (velocity scale is free stream velocity) and it shows also the dimensionless Reynolds stresses $< u'v' >$ for a flat plate boundary layer zero pressure gradient.

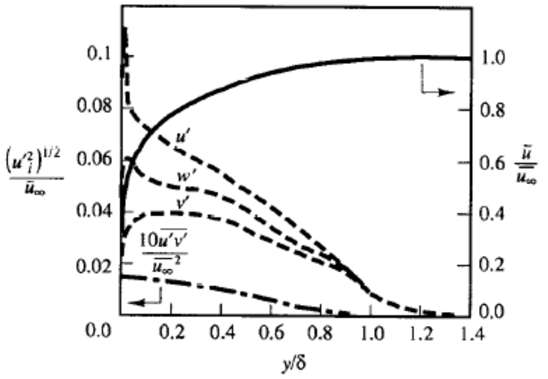

Figure 1.2. Mean velocity distribution and turbulent properties for a zero pressure gradient flat plate boundary layer

As we can see, maximum values of turbulent quantities are found in the region adjacent to the wall, where the large mean velocity gradients ensure a high turbulence production. Walls are therefore a main source of turbulence [17]. However, because of the non-slip condition, turbulent activity must be null at the wall; therefore all turbulent stresses must decrease sharply to zero. Another characteristic we can deduce by looking at the Figure 1.2 is that turbulence is completely anisotropic in the wall region, while leaving the wall all the velocity fluctuations in the three directions tends to have the same route mean square values (turbulence tends to be isotropic far from the wall). This can be explained because the velocity fluctuations in the direction perpendicular to the wall (called v' in Figure 1.2) are damped (because of the physical presence of the wall, which for the kinematic condition cannot be crossed) and are redistributed in the other two directions. Furthermore, neither the values of u' nor the values of w' are equal in the wall region, but the first one is greater. A conclusion we can get from an analysis of Figure 1.2 is that the wall presence modifies in a complex way the turbulent flow behavior. All this features regarding near wall turbulent flows behavior must be taken into account when dealing with near wall turbulence modeling, which is the general topic of this master thesis. Taking into account all the things which we discussed about in this chapter, it results clear that the modeling of this region for turbulent flows is not an easy challenge.

2 k-ε RANS Models for Turbulent Flows

Since Wall Functions have been developed and are available only for k-ε turbulence models in Fluent 6.3, it is convenient to show briefly the background of this family of turbulence models. The purpose of this chapter is neither to show in every particular how every turbulence model have been obtained nor to discuss all the single terms present in every transport equation. The purpose is to make the reader aware of the main characteristics of every turbulence model, starting from the logical derivation of the RANS equation, passing through the main approach of the k-ε models family, arriving until the main differences between the three available models of this family (Standard, RNG, Realizable). This is useful because these three models are the ones used in this master thesis work, and the knowledge of the approach of every turbulence models can make more understandable the difference in the results obtained.

2.1 General Characteristics of Turbulent Flows

First of all we must give a definition of turbulent flow. According to Hinze [18], "Turbulence fluid motion is an irregular condition of flow in which the various quantities show a random variation with time and space coordinates, so that statistically distinct average values can be discerned". Turbulent flow is not predictable, is chaotic in time and space and is an irregular condition of motion. Turbulent flows show an unsteady and non-periodical motion in which the three velocity components float and mass, energy and momentum mixing is obtained. Moreover similar fluctuations are encountered in pressure, temperature and concentration.

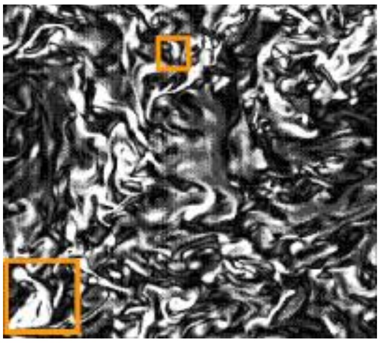

Figure 2.1. Different scale structures inside a turbulent flow

The Figure 2.1 represents an instantaneous picture of a turbulent flow. We can notice that there are many coherent vorticity structures inside; we can notice also that they differ for the size; and that the different size structures are present at the same time. A proof that confirms our first impressions can be found out by looking at a typical turbulent energy spectrum (shown in Figure 2.2). Turbulent energy is the energy associated to the coherent structures and it is not equally distributed between the different sizes. It is instead a function of the length scale of the coherent structure. Must be noted that the integral of the energy spectrum (the area subtended by the curve) represents the total energy of the turbulent structures, and it is called turbulent kinetic energy.

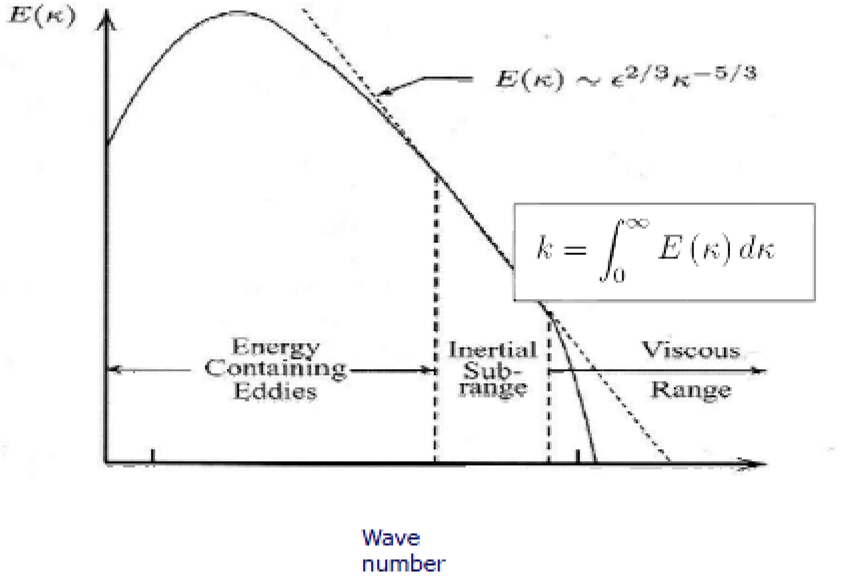

Figure 2.2. Turbulent energy spectrum

Briefly, the theory behind turbulent flows states that, if the Reynolds number is high enough, the energy that is introduced in the mean flow is transferred firstly to the larger scales of turbulent structures and in turn is transferred to the smallest ones, where it is dissipated by the viscosity.

2.2 RANS (Reynolds Averaged Navier-Stokes) Turbulence Models

Figure 2.3 shows a typical time trend for a velocity component (in this case is wall tangential velocity) measured in one point of a turbulent flow.

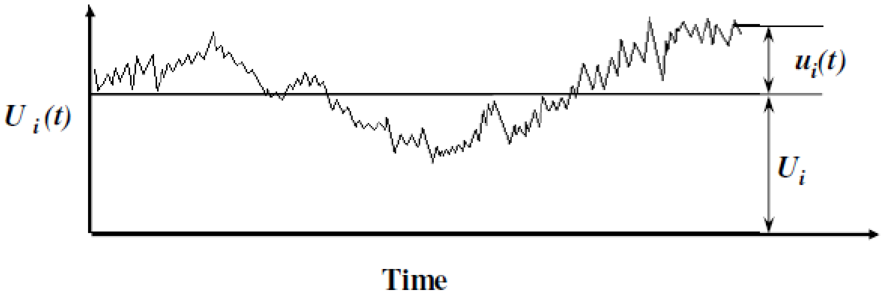

Figure 2.3. Typical velocity fluctuation measured in one point of a turbulent flow

Even though the velocity shows random fluctuations, it's possible to decompose it using the so-called Reynolds decomposition. Two different methods can be applied. The first one is suitable for flows whose averaged quantities are steady (like the one shown in Figure 2.3). In this case the velocity can be decomposed into a mean value called U (which is not time dependent) and a fluctuating component called $u'(t)$, which is always time dependent. Therefore the decomposition becomes:

$$u(t) = U + u'(t) \tag{2.1}$$

The same decomposition can be done for all the velocity components and also for the pressure. The formal expression for the obtainment of the time averaged quantity U is here shown:

$$U = \lim_{T\to\infty} \frac{1}{T}\int_0^T u(t)dt \tag{2.2}$$

This decomposition has been proposed by Reynolds in 1895 and it is useful for the writing of the so-called Reynolds Averaged Navier Stokes (RANS) equations.

On the other hand, when dealing with flows whose averaged quantities are unsteady, a second method is followed. This path leads to the writing of the so-called URANS equations (Unsteady Reynolds Averaged Navier-Stokes equations). In this case the decomposition leads to:

$$u(t) = U(t) + u'(t) \tag{2.3}$$

In this situation the mean value $U(t)$ is time dependent too. In order to obtain this mean value, a different expression for the time averaging must be adopted. The definition is:

$$U(t) = \frac{1}{T}\int_0^T u(t)dt \tag{2.4}$$

The difference with the expression (2.2) is that the averaging is made using a period of time which isn't infinite but which is finite and greater than the characteristic time of turbulent fluctuations τ.

$$T \gg \tau \tag{2.5}$$

Since this master thesis deals only with turbulent flows whose averaged quantities are steady, only the RANS approach is adopted. The process of derivation of RANS equations for the continuity and for the momentum is here briefly shown. The starting point is the writing of the instantaneous Navier-Stokes equations (continuity and momentum conservation):

$$\nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{2.6}$$

$$\nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2.7}$$

In the momentum equation $\boldsymbol{\tau}$ is called stress tensor, and it is defined as follows

$$\boldsymbol{\tau} = \mu \left[(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T) - \frac{2}{3}(\nabla \cdot \boldsymbol{v})\boldsymbol{I}\right] \tag{2.8}$$

It is possible to use Reynolds decomposition (equation (2.3)) to substitute the instantaneous quantities, and then we can make a time averaging using an infinite period of time of every term of the equations. The formal expression of the time average of a generic quantity has been already shown in equation (2.2); the only difference is that we want to obtain the mean value $< \phi >$ of a generic quantity ϕ.

$$< \phi > = \lim_{T\to\infty} \frac{1}{T}\int_0^T \phi(t)dt \tag{2.9}$$

The averaged equations for continuity and momentum become:

$$\nabla \cdot (\rho \boldsymbol{V}) = 0 \tag{2.10}$$

$$\nabla \cdot (\rho \boldsymbol{VV}) = -\nabla P + \nabla \cdot \boldsymbol{T} - \nabla \cdot \boldsymbol{R} \qquad \textbf{(2.11)}$$

In these two equations the terms written in capital letter are the averaged values (or, like ***T***, are the terms computed using only averaged values). Therefore the use of RANS equation allows to obtain the solution only in terms of averaged quantities. A new tensor is appeared after the time averaging, and it is called Reynolds stress tensor (***R***). It contains the time averaged cross products of the fluctuating components of the velocity. For a 3D Cartesian coordinate system, it becomes:

$$\boldsymbol{R} = -\rho \begin{bmatrix} < u'u' > & < u'v' > & < u'w' > \\ < v'u' > & < v'v' > & < v'w' > \\ < w'u' > & < w'v' > & < w'w' > \end{bmatrix} \qquad \textbf{(2.12)}$$

It contains six different terms. The half of the sum of the elements belonging to the main diagonal provides a new quantity, called turbulent kinetic energy (whose physical meaning has been described in paragraph 2.1). Its definition is then:

$$k = \frac{1}{2}(< u'^2 > + < v'^2 > + < w'^2 >) \qquad \textbf{(2.13)}$$

RANS turbulence modeling deals with the modeling of this tensor. The question behind RANS models is: how is it possible to model Reynolds stress tensor in order to take into account its effects on the mean flow?

2.3 Turbulent Viscosity Based Turbulence RANS Models

One possibility for the modeling of Reynolds stress tensor is the use of the so-called Boussinesq hypothesis, which relates Reynolds stresses tensor to a fake viscosity called turbulent (or eddy) viscosity, which is not a physical property of the fluid but it's a characteristic of the flow.

$$-\rho\langle u_i' u_j' \rangle = 2\ \mu_t \boldsymbol{S}_{ij} - \frac{2}{3}\rho k \delta_{ij} \qquad \textbf{(2.14)}$$

Where $\boldsymbol{S}_{ij}$ is the ij element of mean strain rate tensor.

$$\boldsymbol{S}_{ij} = \frac{1}{2}(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}) \qquad \textbf{(2.15)}$$

And δ_{ij} is the so-called Kronecker delta

$$\delta_{ij} = \begin{cases} 1 \; if \; i = j \\ 0 \; if \; i \neq j \end{cases} \tag{2.16}$$

Using this hypothesis we are switching the problem from the modeling of all the terms of Reynolds stress tensor to the modeling of only one quantity, called turbulent viscosity. The k-ε family of turbulence modeling models this quantity using both the dimensional analysis and the concept of mixing length. Using this approach, the general expression for the turbulent viscosity becomes:

$$\mu_t = \frac{\rho C_\mu k^2}{\varepsilon} \tag{2.17}$$

A new quantity called ε has appeared inside the expression. It is called turbulent dissipation rate, and it represents the rate of which viscosity dissipates turbulent kinetic energy. Its dimensions are [L^2T^{-3}]. The quantities k and ε are determined by solving two different transport equations (one for each quantity). There are three different k-ε models available in Fluent 6.3:

- Standard
- RNG (Re-Normalization Group)
- Realizable

In the next three subparagraphs the transport equations and the main characteristics of the k-ε turbulence models available are briefly shown.

2.3.1 k-ε Standard Model

The transport equations of turbulent kinetic energy and turbulent dissipation rate for the k-ε Standard model and for an incompressible flow are [19], [20]:

$$\frac{\partial}{\partial x_i}(\rho k U_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon \tag{2.18}$$

$$\frac{\partial}{\partial x_i}(\rho \varepsilon U_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{C_{1\varepsilon}\varepsilon}{k}(P_k) - \frac{C_{2\varepsilon}\varepsilon^2}{k}\rho \tag{2.19}$$

P_k represents the production of turbulence kinetic energy due to the mean velocity gradients.

$$P_k = -\rho\langle u_i' u_j'\rangle \frac{\partial U_i}{\partial x_i} \tag{2.20}$$

The one here presented is the exact expression; it is modeled by using a way consistent with Boussinesq hypothesis:

$$P_k = \mu_t S^2 \tag{2.21}$$

where

$$S = \sqrt{2\boldsymbol{S}_{ij}\boldsymbol{S}_{ij}} \tag{2.22}$$

The constants of the model are:

Table 2.1. Constants of Standard k-ε turbulence model

$C_{1\varepsilon} = 1.44$	$C_{2\varepsilon} = 1.92$	$C_\mu = 0.09$	$\sigma_k = 1.0$	$\sigma_\varepsilon = 1.3$

As Fluent User Guide reports, "these default values have been determined from experiments with air and water for fundamental turbulent shear flows including homogeneous shear flows and decaying isotropic grid turbulence. They have been found to work fairly well for a wide range of wall-bounded and free shear flows".

2.3.2 k-ε RNG Model

The RNG model was derived using a rigorous statistical technique (called renormalization group theory) [21]. It is similar in form to the Standard k-ε model, but includes the following refinements (according to Fluent 6.3 User Guide [22])

- The RNG model has an additional term in its ε equation that significantly improves the accuracy for rapidly strained flows.
- The effect of swirl on turbulence is included in the RNG model, enhancing accuracy for swirling flows.
- The RNG theory provides an analytical formula for turbulent Prandtl numbers, while the Standard model uses constant values.
- While the Standard model is a high-Reynolds-number model, the RNG theory provides an analytically-derived differential formula for effective viscosity that accounts for low-Reynolds-number effects. Effective use of this feature does, however, depend on an appropriate treatment of the near-wall region.

The transport equations of turbulent kinetic energy and turbulent dissipation rate are:

$$\frac{\partial}{\partial x_i}(\rho k U_i) = \frac{\partial}{\partial x_j}\left[\left(\mu_{eff}\alpha_k\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon \quad \textbf{(2.23)}$$

$$\frac{\partial}{\partial x_i}(\rho \varepsilon U_i) = \frac{\partial}{\partial x_j}\left[\left(\mu_{eff}\alpha_\varepsilon\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{C_{1\varepsilon}\varepsilon}{k}(P_k) - \frac{C_{2\varepsilon}\varepsilon^2}{k}\rho - R_\varepsilon \quad \textbf{(2.24)}$$

If we compare these transport equation with the ones of the Standard model we can notice three main differences.

- The terms α_k and α_ε are the inverse effective Prandtl number for k and ε (called σ_k and σ_ε in the Standard model). These values are not constant but instead are computed using a formula derived analytically by the RNG theory.
- Second, the term R_ε is the additional term in ε equation above mentioned.
- Last, in the equations the term μ_{eff} appears. This is slightly differently computed from the Standard computation of μ_t for k-ε model. However it is only important to point out that the difference resides in the fact that the expression for RNG model takes into account possible low Reynolds effects. In the high-Reynolds-number limit the computation of μ_{eff} becomes equal to the computation of μ_t (equation (2.17)).

The constants of the model are derived using the RNG theory. It is good to point out that in the Standard model the constants are obtained from experimental data. The numerical values of the constants are:

Table 2.2. Constants of RNG k-ε turbulence model

$C_{1\varepsilon} = 1.42$	$C_{2\varepsilon} = 1.68$	$C_\mu = 0.0845$

2.3.3 k-ε Realizable Model

The differences between Realizable model and Standard are [23], [24]:

- It contains a new formulation for the turbulent viscosity
- A new transport equation for the turbulent dissipation rate has been derived from an exact equation for the transport of the mean square vorticity fluctuation

Realizable model is also likely to provide superior performance for flows involving rotation, boundary layers under strong adverse pressure gradients, separation, and recirculation. The term Realizable means that the model satisfies certain mathematical constraints on the Reynolds stresses, consistent with the physics of turbulent flows. Neither the Standard model nor the RNG model is realizable. The two constraints for the realizability are:

- Mean value of u'^2 must be always positive
- Schwarz inequality $< u'_a u'_b >^2 \leq \ < u'^2_a u'^2_b >$ must be always ensured

The most straightforward way to ensure the realizability is to make C_μ variable by sensitizing it to mean flow (mean deformation) and to the turbulence parameters (k and ε).

The transport equations for turbulent kinetic energy and turbulent dissipation rate are:

$$\frac{\partial}{\partial x_i}(\rho k U_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon \qquad \textbf{(2.25)}$$

$$\frac{\partial}{\partial x_i}(\rho \varepsilon U_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \rho C_1 S\varepsilon \ - \ \frac{C_2\varepsilon^2}{k + \sqrt{\varepsilon \upsilon}}\rho \qquad \textbf{(2.26)}$$

C_1 is a function of strain rate, turbulent kinetic energy and turbulent dissipation rate; C_2 is a constant. Note that the k equation is the same as that in the Standard model and in the RNG, except for the model constants. However, the form of the turbulent dissipation rate equation is quite different from those in the Standard and RNG models. One of the noteworthy features is that the production term in the equation (the second term on the right-hand side) does not involve the production of k. It is believed that the present form better represents the spectral energy transfer. Another desirable feature is that the destruction term (the last term on the right-hand side) does not have any singularity; i.e., its denominator never vanishes, even when k vanishes.

The constants of the model have been established to ensure that the model performs well for certain canonical flows. The numerical values are:

Table 2.3: Constants of Realizable k-ε turbulence model

$C_{1\varepsilon} = 1.44$	$C_{2\varepsilon} = \ 1.9$	$\sigma_k = 1.0$	$\sigma_\varepsilon = 1.2$

3 Wall Functions Approach

3.1 Wall Boundary Conditions

The challenge of wall treatment in wall bounded turbulent flows deals with the problem to give proper boundary conditions at the wall and at the same time deals with the near-wall region, where there are strong gradients in a thin layer and where, as already discussed in chapter 1, the behavior of the flow is different from the region which is not affected by the presence of the wall. Since this work is focused on Wall Functions for a k-ε model, and since CFD deals with the resolution of partial derivative equations, in order to obtain results which are good we must provide proper boundary conditions at the wall for all the equations involved, i.e.:

- Momentum equation – wall tangential and wall normal component
- Turbulent kinetic energy equation
- Turbulent dissipation rate equation

Wall Functions are one possible approach for wall treatment of wall bounded turbulent flows. A reason why Wall Functions approach is very popular is that it gives CFD user the possibility to use a coarse near-wall mesh. A detailed solution of the entire turbulent boundary layer (that can be obtained by using a Low Reynolds model) can be prohibitively large in terms of computational time and costs. At the same time Wall Functions approach allow to obtain good and physical results, provided that the assumptions used for their derivation are valid for the case studied.

3.2 Momentum Equation – Wall Boundary Conditions

The logical approach of Wall Functions wall treatment for k-ε RANS eddy viscosity turbulence models is the following. In chapter 1, the near-wall region has been described. Briefly, it can be divided into three zones: sub-viscous layer, buffer region and fully turbulent region. Figure 3.1 shows a simple wall tangential velocity profile, where the three zones can be easily discerned.

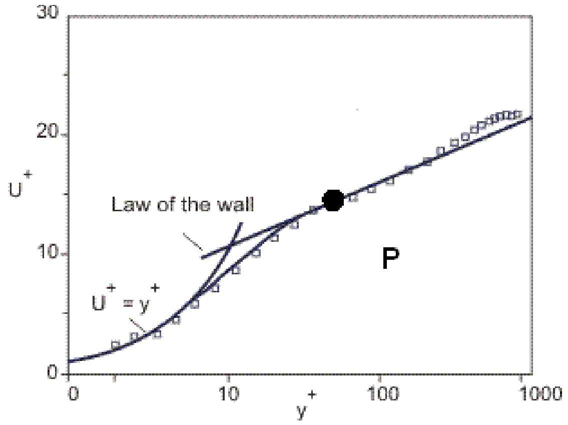

Figure 3.1. Dimensionless velocity profile close to a solid wall; placement of first computational node P when adopting Wall Functions approach

The first computational node (the center of the near wall cell) is placed in the fully turbulent region, in order to bridge the viscosity affected region (i.e. sub-viscous layer and buffer layer). Point P in the Figure 3.1 represents a possible position of near wall center cell. Conventionally the fully turbulent region starts with a value of y^+equal to 30 and goes until y^+ equal to 300. It's good to remind that these are the values suggested by Fluent User Guide for the Standard Wall Functions [15], but in other situations this limits can be totally different. The momentum equation for the wall tangential component for a near wall volume control in a 2D Cartesian grid, when discretized, is the following:

$$a_P U_P = a_E U_E + a_W U_W + a_N U_N + a_S U_S + S_U \qquad (3.1)$$

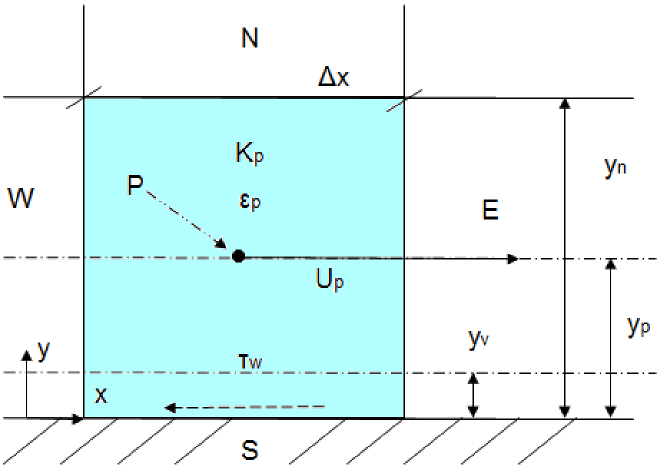

Figure 3.2. Near wall mesh for a 2D Cartesian grid

Where U_P represents the wall tangential velocity at the center of the cell P (the cell with the light blue background in the Figure 3.2) and U_E, U_W, U_N, U_S, represent the wall tangential velocities at P neighborhood center cells.
The coefficient a_S represents the link between center cell wall tangential velocity and the boundary (i.e. the wall). It arises from the shear stress at the south face of the cell (the wall shear stress). But since the point P falls into logarithmic region we cannot estimate wall shear stress using a linear approximation for the velocity profile:

$$\tau_w \neq \mu \frac{U_P - U_S}{y_P} \tag{3.2}$$

The expression above is wrong because, since P is put in the fully turbulent region, the velocity profile isn't linear at all (it is strictly linear only in the sub-viscous region). The use of this expression will lead to a wrong estimation of wall shear stress and consequently to wrong results in the entire domain. A way to overcome this challenge is to assume a wall tangential velocity profile in the near wall region and to obtain wall shear stress from the assumed profile. After doing that the discretized momentum equation will be modified in the following way.

- The link with the boundary is suppressed

$$a_S = 0 \tag{3.3}$$

- A volume source is put inside the equation which is computed as:

$$S_U = \tau_W \Delta x \tag{3.4}$$

Where Δx is the thickness of the cell in x-direction.

At this point we can easily notice the weak spot of Wall Functions approach. We are assuming "a priori" a velocity profile for the wall tangential velocity in the near wall region in order to estimate wall shear stress. But we don't know if the assumed velocity profile is right. If it is not, we are evaluating a non-correct wall shear stress, and the results will be affected by this wrong boundary condition.

Regarding wall perpendicular velocity transport equation, the boundary condition comes from the kinematic condition. Therefore it's enough to put wall perpendicular velocity at the wall equal to zero.

3.3 Wall Boundary Conditions for Turbulent Kinetic Energy and Turbulent Dissipation Rate Transport Equations

After providing the value of wall shear stress (boundary condition at the wall for momentum equation), we must be able to provide boundary condition at the wall also for the other two equations that the code is going to solve, which are turbulent kinetic energy and turbulent dissipation rate transport equation. The same wall tangential velocity profile assumed for the evaluation of wall tangential momentum equation boundary condition is used, and it is used also the wall shear stress previously determined. All the Wall Functions presents in the literature evaluate the boundary conditions in the following way.

- For the turbulent kinetic energy transport equation, the boundary condition is not provided for k, but two different boundary conditions are provided, respectively for production of turbulent kinetic energy and for turbulent dissipation rate. These two values are the source values inside k transport equation, which is solved putting the diffusion of k in the direction perpendicular to the wall equal to zero.
- For turbulent dissipation rate transport equation, on the contrary, it's provided the value of ε for the near wall center cell. Therefore the transport equation is not solved for near wall cells.

The Wall Functions distinguish themselves because of the different way to compute these values, to be more precise the differences resides in the velocity profile assumed and in the way used to compute production of turbulent kinetic energy and turbulent dissipation rate.

3.4 What Happens if First Node Falls into Sub-viscous Layer?

The logical approach of Wall Functions plans also an alternative solution if the first computational node falls into the sub-viscous layer. This is not a secondary problem because it can easily happen that we cannot guarantee that all the near wall center cells are in the fully turbulent region. For example, if we are dealing with a separating and reattaching flow there will be a region close to separation and one close to reattachment where velocity is low, and consequently also y^+ will be low (because wall shear stress and consequently friction velocity will be low). Another typical example is when we have a complex geometry and it's hard to guarantee that with the mesh created all the near wall cells have a proper wall distance. If the computational node is in the sub-viscous layer, the approach is modified as follows:

- First of all, a threshold value for the dimensionless thickness of sub-viscous layer (y_v^+) must be assigned. The code will check if the value of y^+ of first computational node is higher or lower than y_v^+. If it is lower, the following procedure will be implemented.
- Wall shear stress is obtained assuming a linear relationship between velocity and wall distance.

$$\tau_w = \mu \frac{\partial U}{\partial y} \approx \mu \frac{U_P}{y_P} \tag{3.5}$$

- Regarding turbulent quantities, no information is found in literature. Therefore an analysis about this point must be carried out further in this master thesis.

3.5 Merits and Imperfections of Wall Functions Approach

The main advantages of this approach are clearly visible. Since we do not entirely solve the wall affected region, we do not need a fine wall mesh, and therefore a coarser mesh can be used. Computational time and costs can be saved while keeping good results, provided that the Wall Functions used estimate properly near wall flow behavior. Low computational costs are the main reason why Wall Functions are still widespread in CFD. Moreover it is quite simple to introduce empirical correlations when estimating near wall flow. For example, this can be useful if we want to take into account some effects such as wall roughness.

The fact that the placement of the first computational node is in the fully turbulent region leads to another advantage of the use of Wall Functions. The region where the viscosity isn't negligible is bypassed. Then the turbulence models, which are written for fully turbulent flows, don't need any modifications in order to make them suitable to the whole domain. Indeed all the computational nodes, for which the transport equations of turbulent kinetic energy and of turbulent dissipation rate are solved, are fully turbulent. This allows to avoid the complication of adding modifications to the models; complication which is present for Low Reynolds models (as discussed in the Introduction of this master thesis). However, this problem re-appears when near wall center cell falls into sub-viscous layer. It's true that there is a modification of the Wall Functions procedure in order to take into account the linear velocity profile inside this sub-region, but on the other hand the fully turbulent equations for k and ε are used also for this cell, which falls into a region where the flow is not fully turbulent.

On the other hand, it’s obvious that the use of pre-integrated expressions for the determination of boundary conditions is not rigorous. Since we need to assume the flow behavior, the goodness of the results is strongly dependent on the assumptions made. Then, a correct use of Wall Functions cannot prescind from a preliminary analysis of the flow we want to simulate. To be more precise we must know the dimensionless velocity profile close to the wall and the turbulence behavior in the same region. This procedure, however, cannot be always realized. For example we may not have enough information about the wall affected region (we must remember that for high Reynolds number flows this region is very thin). Moreover, purpose of CFD is to simulate the fluid dynamic of a flow in order to obtain information about it. If we need to know “a priori” some results of the simulation, then it’s like we are using CFD as an “a posteriori” checking tool. That’s why it is important to develop Wall Functions whose assumptions are generally valid for several situations.

Different Wall Functions have been proposed through CFD history. However in Fluent version 6.3 there are only two Wall Functions available: Standard Wall Functions (SWF) and Non Equilibrium Wall Functions (NEWF). The logic behind this two functions and the working of CFD code when these functions are used will be further down investigated. In addition there will be investigated the working of Generalized Wall Functions, whose implementation into Fluent 6.3 is the goal of this master thesis.

3.6 Standard Wall Functions

3.6.1 Simplest Implementation

Standard Wall Functions have been proposed by Launder and Spalding in 1974. These are the first Wall Functions proposed in the literature. The logic and the assumptions used to determine SWF are the following:

- The dimensionless velocity distribution in the wall region follows a semi-logarithmic profile
- Uniform shear stress prevails in the boundary layer
- Production and dissipation of turbulent kinetic energy in the boundary layer are in balance (the boundary layer is a so-called “equilibrium boundary layer”)

$$P_k = \rho\varepsilon \tag{3.6}$$

- Turbulent dissipation rate is assumed to be

$$\varepsilon \propto \frac{k^{1.5}}{y} \tag{3.7}$$

Starting from these assumptions it is possible to write wall boundary conditions. The first assumptions can be written as:

$$u^{+} = \frac{\log(E\,y^{+})}{\kappa} \tag{3.8}$$

Where u^{+}, y^{+} and u_{τ} have been defined in equations (1.7), (1.8) and (1.9). E and κ are two constants which must be determined experimentally.
Wall shear stress can be easily obtained from the expression of the semi logarithmic velocity profile. Writing explicitly the non-dimensional velocity profile we obtain

$$\frac{U_P}{\sqrt{\frac{\tau_w}{\rho}}} = \frac{\ln(E\,\frac{\rho y\sqrt{\frac{\tau_w}{\rho}}}{\mu})}{\kappa} \tag{3.9}$$

And wall shear stress can be obtained by inverting the expression above:

$$\sqrt{\frac{\tau_w}{\rho}} = \frac{\kappa U_P}{\ln(E\,\frac{\rho y\sqrt{\frac{\tau_w}{\rho}}}{\mu})} \tag{3.10}$$

$$\tau_w = \rho\left(\frac{\kappa U_P}{\ln\left(E\,\frac{\rho y\sqrt{\frac{\tau_w}{\rho}}}{\mu}\right)}\right)^2 \tag{3.11}$$

To be rigorous, since wall shear stress is both on LHS and RHS of the equation, every time the code operates one iteration, an iterative process for wall shear stress evaluation is required. However, in order to quicken the calculation, CFD codes take the value of τ_w at RHS as constant and known from the previous iteration. Therefore it is possible to compute directly the updated value of wall shear stress.

The assumptions about boundary layer lead to

$$\frac{\tau_W}{\rho} = C_\mu^{0.5} k = \text{constant} \tag{3.12}$$

From this expression it is easy to give a value for turbulent kinetic energy at near wall center cell. From an inversion of the expression (3.12) we obtain:

$$k_P = \frac{u_\tau^2}{C_\mu^{0.5}} \tag{3.13}$$

As a consequence turbulent kinetic energy equation is not solved for the near wall cells.

Also turbulent dissipation rate equation is not solved. The value of turbulent dissipation rate at the center cell is prescript starting from equation (3.7):

$$\varepsilon_P = \frac{k^{1.5}}{c_l y} \tag{3.14}$$

c_l is a length scale to be determined. The expression (3.14) derives from experimental analyses.

3.6.2 Standard Implementation

The one just presented is the simplest implementation of Standard Wall Functions, but it’s not the implementation proposed by Launder and Spalding and used in commercial CFD codes. Theoretically, using the implementation shown in previous paragraph all the wall boundary conditions we need can be computed by the code. However this is implementation is not used because it can lead to non-physical results. One typical situation is the following. When wall shear stress tends to zero (it is a very common situation in fluid dynamics, e.g. separating and reattaching flows or impinging jets) the implementation shown before leads to:

$$\tau_w \to 0 \tag{3.15}$$

$$k \propto \tau_w \tag{3.16}$$

$$\varepsilon \propto \tau_w^{1.5} \qquad \textbf{(3.17)}$$

Since we are using a k-ε model, the turbulent viscosity obtained is

$$\mu_t = \rho\, C_\mu \frac{k^2}{\varepsilon} \propto \frac{k^2}{k^{1.5}} \rightarrow 0 \qquad \textbf{(3.18)}$$

But experimental data shows that near reattachment or in impinging jets turbulent viscosity (and consequently turbulent kinetic energy) is not null. In order to overcome this problem, an alternative implementation has been proposed (which is the original implementation proposed by Launder and Spalding). One difference between the two implementations resides in the use of another velocity scale together with u_τ. The new velocity scale is defined as

$$u_k = k^{0.5} C_\mu^{0.25} \qquad \textbf{(3.19)}$$

That is equal to u_τ if the assumptions described before are valid (see equation (3.12)). It is now possible to rewrite the semi-logarithmic velocity profile expression (3.8) using both the velocity scales. Multiplying LHS by u_k and RHS by u_τ, and substituting u_τ with u_k in the definition of y^+ (these operations are licit because the two velocity scales are equal) we obtain:

$$\frac{U}{u_\tau} u_k = u_\tau \frac{\log(E\ \frac{\rho y u_k}{\mu})}{\kappa} \qquad \textbf{(3.20)}$$

$$\frac{U\, u_k}{u_\tau^2} = \frac{\log(E\ \frac{\rho y u_k}{\mu})}{\kappa} \qquad \textbf{(3.21)}$$

Now we call

$$\frac{U\, u_k}{u_\tau^2} = u^* \qquad \textbf{(3.22)}$$

And

$$\frac{\rho y u_k}{\mu} = y^* \qquad \textbf{(3.23)}$$

The semi logarithmic dimensionless profile can be rewritten with the new dimensionless quantities (* *adimensionalization*):

$$u^* = \frac{\log(E\ y^*)}{\kappa} \qquad \textbf{(3.24)}$$

Wall shear stress is obtained from the inversion of the new non-dimensional velocity profile

$$\tau_w = \frac{\rho\ U_P u_k \kappa}{\log(\ E y^*)} \qquad \textbf{(3.25)}$$

Regarding turbulent kinetic energy, the approach is different from the simplest implementation shown before. Instead of giving a value of turbulent kinetic energy for the cell center, transport equation is solved using the following assumptions:

- Diffusion of turbulent kinetic energy in direction perpendicular to the wall is set to zero. This condition can be written as

$$\frac{\partial k}{\partial \boldsymbol{n}} = 0 \qquad \textbf{(3.26)}$$

- Production of turbulent kinetic energy in the near wall cell is assumed known; rigorously the value to put inside the equation should be averaged on the whole near wall cell. The averaging is necessary because turbulent kinetic energy production varies significantly inside near wall cell. Directly putting the value at center cell as a source term inside turbulent kinetic energy equation can cause a not acceptable approximation; however some authors suggest to put the center cell value of P_k as source term in the equation
- Turbulent dissipation rate in turbulent kinetic energy equation is evaluated with the same logic used for production of turbulent kinetic energy

These values can be estimated starting from simplified turbulent quantities profiles into wall region. The profiles are shown in the Figure 3.3. These profiles follow a two-layer approach (there is a sharp difference between the sub-viscous layer, where the viscosity is not negligible, and the fully turbulent region, where the viscosity is negligible).

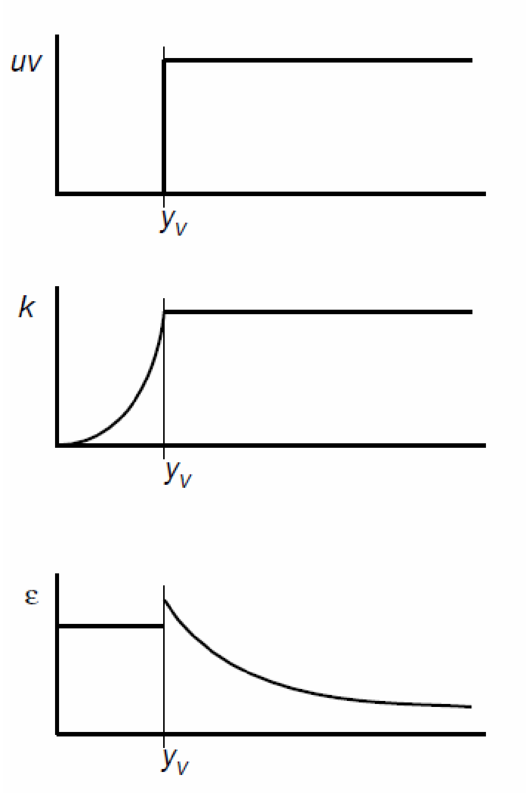

Figure 3.3. Turbulent quantities profiles in wall region

In particular:

- Reynolds stresses (the non-diagonal terms of Reynolds stress tensor) are supposed to be equal to zero in the sub-viscous layer, and equal to minus wall shear stress divided by the density outside sub-viscous layer

$$< u'v' > = \begin{cases} 0 & for\, y < y_v \\ -\frac{\tau_w}{\rho} & for\ y \geq y_v \end{cases} \tag{3.27}$$

- Turbulent kinetic energy goes with the 2nd power of wall distance in the sub-viscous layer and it is supposed to be constant and equal to the value at sub viscous layer edge in the fully turbulent region

$$\begin{cases} k\ \alpha\ y^2 & for\, y < y_v \\ k = k_{y_v} & for\ y \geq y_v \end{cases} \tag{3.28}$$

- Turbulent dissipation rate is constant in sub viscous layer and has an inverse proportional dependence from the wall distance in the fully turbulent region

$$\varepsilon = \begin{cases} \dfrac{2\,\upsilon k}{y_v^2} & for\, y < y_v \\ \dfrac{k^{1.5}}{c_l y} & for\, y \geq y_v \end{cases} \tag{3.29}$$

The knowledge of turbulent quantities profiles close to the wall is not enough in order to evaluate source terms of k equation. We need also assumptions about how P_k and ε are related to these quantities.
Regarding P_k, it's correct expression is:

$$P_k = -\rho < u'v' > \frac{\partial U}{\partial y} \quad \left[\frac{kg}{ms^3}\right] \tag{3.30}$$

This is the correct expression (like shown also in equation (2.20)). By using the profile of Reynolds stresses (equation (3.27)) it is possible to evaluate P_k at center cell, if this falls into fully turbulent region:

$$P_k = \tau_w \frac{\partial U}{\partial y} \quad \left[\frac{kg}{ms^3}\right] \tag{3.31}$$

The value averaged on the whole near wall cell is defined by the following equation:

$$\overline{P_k} = \frac{1}{y_n}\int_0^{y_n} -\rho < u'v' > \frac{\partial U}{\partial y} dy \tag{3.32}$$

Using the two layer approach it is possible to split the integral:

$$\overline{P_k} = \frac{1}{y_n}\int_{y_v}^{y_n} \tau_w \frac{\partial U}{\partial y} dy \tag{3.33}$$

It is now possible to pass from the derivative of the dimensional wall tangential velocity with respect to the wall distance to the derivative of the dimensionless wall tangential velocity with respect to the dimensionless wall distance by using the definitions (3.22) and (3.23).

$$\overline{P_k} = \frac{1}{y_n}\{\frac{\tau_w^2}{\mu}\int_{y_v}^{y_n} \frac{\partial u^*}{\partial y^*} dy\} \tag{3.34}$$

We can substitute now the first derivative of the dimensionless law of the wall adopted by SWF, which is:

$$\frac{\partial u^*}{\partial y^*} = \frac{1}{\kappa y^*} \tag{3.35}$$

$$\overline{P_k} = \frac{\tau_w^2}{\mu y_n}\int_{y_\nu}^{y_n} \frac{1}{\kappa y^*} dy \tag{3.36}$$

After substituting y^* with its definition (equation (3.23)), the integral in the equation (3.36) can now be solved

$$\overline{P_k} = \frac{\tau_w^2}{\mu y_n}\int_{y_v}^{y_n} \frac{1}{\frac{\kappa \rho y u_k}{\mu}} dy \tag{3.37}$$

$$\overline{P_k} = \frac{\tau_w^2}{\kappa \rho u_k y_n} \log(\frac{y_n}{y_v}) \tag{3.38}$$

Regarding ε (source term inside k transport equation) it is computed using the profile shown before (equation (3.29)) and averaging it on the whole near wall cell:

$$\bar{\varepsilon} = \frac{1}{y_n}\int_{0}^{y_v} \frac{2\,\upsilon k}{y_v^2} dy + \int_{y_\nu}^{y_n} \frac{k^{1.5}}{c_l y} dy \tag{3.39}$$

$$\bar{\varepsilon} = \frac{k^{1.5}}{y_n}(\frac{2}{y_v^*} + \frac{1}{c_l}\log\left(\frac{y_n}{y_v}\right)) \tag{3.40}$$

Regarding turbulent dissipation rate transport equation, the process is the same as simplest implementation. To be more precise, transport equation is not solved for near wall cells, but a value at center cell is prescript. The value is prescript by using the equation (3.14), where c_l is usually set to the value of 2.55.

3.6.3 Fluent 6.3 Implementation

This section shows the implementation of Standard Wall Function into Fluent 6.3 according to Fluent 6.3 User's Guide [15]. The equations reported are respectively: the relationship between non-dimensional velocity profile and non-dimensional wall distance, the definition of these two quantities, the expressions for the computation of turbulent kinetic energy production and turbulent dissipation rate at near wall center cell, and the values of the constant used.

$$u^* = \frac{1}{\kappa}\ln(E\ y^*) \tag{3.41}$$

$$u^* = \frac{U_P C_\mu^{0.25} k_P^{0.5}}{\frac{\tau_w}{\rho}} \tag{3.42}$$

$$y^* = \frac{\rho C_\mu^{0.25} k_P^{0.25} y_P}{\mu} \tag{3.43}$$

$$P_k \approx \tau_w \frac{\partial U}{\partial y} = \tau_w \frac{\tau_w}{\kappa \rho C_\mu^{0.25} k_P^{0.5} y_P} \tag{3.44}$$

$$\varepsilon_P = \frac{C_\mu^{0.75} k_P^{1.5}}{\kappa y_P} \tag{3.45}$$

Table 3.1. Fluent 6.3 SWF constants

$\kappa = 0.4187$	$E = 9.793$	$y_v^+ = 11.225$

After a comparison between the Standard implementation of SWF shown before and the information obtained from Fluent User guide, the following observations can be made:

- The dimensionless velocity profile adopted by Fluent follows the same semi-logarithmic law used in the Standard implementation; moreover the dimensionless velocity and wall distance are computed using the * *adimensionalization* (i.e. with both velocity scales u_k and u_τ).
- Turbulent kinetic energy is solved in the wall domain including wall-adjacent cells by setting the diffusion of k in direction perpendicular to the wall to zero and by using the expressions (3.44) and (3.45) to compute P_k and ε (that are the source values to be put inside the equation).

- P_k is evaluated as the center cell value and not as the averaged value. It is computed using the approach shown before but without the averaging. Briefly:

$$\overline{P_k} = \frac{\tau_w^2}{\mu}\frac{\partial u^*}{\partial y^*} \tag{3.46}$$

$$\overline{P_k} = \frac{\tau_w^2}{\mu}(\frac{1}{\kappa y^*}) \tag{3.47}$$

$$\overline{P_k} = \frac{\tau_w^2}{\mu}(\frac{1}{\frac{\kappa\rho y u_k}{\mu}}) \tag{3.48}$$

Equation (3.48) is equal to equation (3.44).

- Turbulent dissipation rate (the source term for k equation) is computed using center cell value without averaging on the whole near whole cell
- Length scale c_l inside turbulent dissipation rate expression is taken equal to:

$$c_l = \frac{\kappa}{C_\mu^{0.75}} = \frac{0.4187}{0.09^{0.75}} \approx 2.5481 \tag{3.49}$$

An interesting observation can be made for the term y_v^*. This term represents the dimensionless thickness of sub viscous layer (its working is explained in chapter 3.4). It is interesting to note that the dimensionless thickness of sub-viscous layer is given with the * *adimensionalization*.

Note: the equilibrium boundary layer condition is verified. Indeed:

$$P_k = \rho\varepsilon_P \tag{3.50}$$

$$\frac{\tau_w^2}{\kappa\rho u_k y_P} = \rho\frac{u_k^3}{\kappa y_P} \tag{3.51}$$

$$\frac{\tau_w^2}{\rho u_k} = \rho u_k^3 \tag{3.52}$$

$$\frac{\tau_w^2}{\rho^2 u_k} = u_k^3 \tag{3.53}$$

$$u_\tau^4 = u_k^4 \tag{3.54}$$

But equation (3.12) showed that if there is an equilibrium boundary layer the two velocity scales are equal. Therefore production of turbulent kinetic energy is equal to turbulent dissipation rate.

3.6.4 Developments of SWF

Standard Wall Functions success is due mainly to their simplicity and robustness. However these functions are not universal. Its use is rigorous only if the case studied presents an equilibrium boundary layer and a semi-logarithmic velocity distribution. But these conditions occur in a very few situations (e.g. pipe flow and channel flow). When dealing with strong pressure gradients, body forces or strong 3D effects, the assumptions used for SWF are not valid and results obtained can be wrong. Moreover, the law of the wall assumed contains only constants; it is not sensitized to the conditions of the flow at all.

Several developed Wall Functions have been proposed through the years. The developments are generally driven by the attempt to generalize the validity field of the assumptions behind each function, and also by the attempt to introduce a law of the wall that is partially sensitized to the conditions of the flow. For example we can mention Non Equilibrium Wall Functions (NEWF), which take into account pressure-gradient effects, or Analytical Wall Functions (AWF), which have been proposed by Craft et al. However it is difficult to obtain a more general validity field and at the same time to have functions comparable to Standard Wall Functions (in terms of easiness of the implementation into a CFD code). Object of this master thesis work is Generalized Wall Functions (proposed by Popovac & Hanjalic in 2007). The reason why GWF were chosen as topic of the work is because these functions show at the same time the two characteristic already mentioned, i.e. general validity field of the functions themselves and relative easiness of the implementation of them into CFD codes.

3.7 Non Equilibrium Wall Functions

In addition to Standard Wall Functions (described in paragraph 3.6), which are the default near-wall treatment in Fluent 6.3, there is an alternative available in Fluent 6.3. This is called Non Equilibrium Wall Functions (NEWF) [25], and they have been developed in 1995 by S. E. Kim and D. Choudhury [9]. These are mainly an improvement of Standard Wall Functions, where the key elements are the following:

- Launder and Spalding's log law for mean velocity is sensitized in order to account for pressure gradient effect
- A two layer based concept is adopted in order to compute the budget of turbulent kinetic energy in the wall neighboring cells

The log-law for mean wall tangential velocity sensitized to pressure gradient is, according to Fluent 6.3 User Guide:

$$\frac{\hat{U} C_\mu^{0.25} k^{0.5}}{\frac{\tau_w}{\rho}} = \frac{1}{\kappa} \log(E \frac{\rho c_\mu^{0.25} k^{0.5} y}{\mu}) \quad \textbf{(3.55)}$$

That is equal to Standard Wall Function log law, except for the definition of $\hat{U}$, which is:

$$\hat{U} = U - \frac{1}{2}\frac{\partial P}{\partial x}\left[\frac{y_v}{\rho\kappa\sqrt{k}}\log\left(\frac{y}{y_v}\right) + \frac{y - y_v}{\rho\kappa\sqrt{k}} + \frac{y_v^2}{\mu}\right] \quad \textbf{(3.56)}$$

In this expression there is the definition of y_v, which is the physical viscous sub-layer thickness. It is computed as usual as:

$$y_v = \frac{\mu y_v^*}{\rho C_\mu^{0.25} k_P^{0.5}} \quad \textbf{(3.57)}$$

And, as in Standard Wall Functions, y_v^* is set to be equal to 11.225.

Regarding the computation of budget of turbulent kinetic energy, the Non Equilibrium Wall Functions employ the two layer concept. To be more precise, the wall neighboring cells are assumed to consist of a viscous sub-layer and a fully turbulent layer. This is the approach shown in paragraph 3.6.2, which was found not to be used in Fluent 6.3 implementation. The following profile assumptions for turbulence quantities are made:

$$\tau(y) = \begin{cases} 0, & y < y_v \\ \tau_w, & y > y_v \end{cases} \quad \textbf{(3.58)}$$

$$k = \begin{cases} \left(\frac{y}{y_v}\right)^2 k_P, & y < y_v \\ k_P, & y > y_v \end{cases} \quad \textbf{(3.59)}$$

$$\varepsilon = \begin{cases} \dfrac{2\upsilon k}{y^2}, & y < y_v \\ \dfrac{k^{\frac{3}{2}}}{C_l^* y}, & y > y_v \end{cases} \tag{3.60}$$

Where the length scale in turbulent dissipation rate profile is defined as

$$C_l^* = \kappa C_\mu^{-\frac{3}{4}} \tag{3.61}$$

Using these profiles, the cell-averaged production of k and the cell-averaged dissipation rate can be computed.

This way to compute turbulent kinetic energy budget is effectively sensitized to the proportions of the viscous sub-layer and the fully turbulent layer, which varies widely from cell to cell in highly non-equilibrium flows. It effectively relaxes the local equilibrium assumption (production equal to dissipation) that is adopted by the Standard Wall Function in computing the budget of the turbulence kinetic energy at wall-neighboring cells. Thus, the Non-Equilibrium Wall Functions, in effect, partly account for non-equilibrium effects neglected in the Standard Wall Function.

Compared to Generalized Wall Functions, since Non Equilibrium Wall Functions are already implemented into Fluent, they are stable, reliable, and without bugs. At least they should be more stable and reliable and they should have fewer bugs than GWF, which are going to be implemented with a user defined functions manually written.

3.8 Generalized Wall Functions

These Wall Functions have been proposed in 2007 by Popovac & Hanjalic [10]. The interest in the use of these Wall Functions is based on the fact that, as said before, the assumptions made for their derivation have a wider validity than SWF ones and at the same time the implementation into a commercial CFD code is quite simple. As the authors say in their paper: “The expressions are compatible with the Standard Wall Functions expressions, thus easy implementable in existing CFD codes”.

The first assumption regards turbulent viscosity profile for the near wall cell. The profile proposed is the following:

$$\mu_t = \begin{cases} 0 & for\; y < y_v \\ \kappa\rho u_\tau y & for\; y \geq y_v \end{cases} \tag{3.62}$$

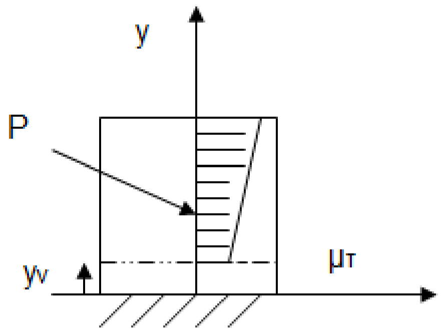

Figure 3.4. Assumed turbulent viscosity profile for the near wall cell

In their paper the authors state also that: “The single assumption that the non-dimensional eddy viscosity varies linearly with the distance from the wall was found to hold reasonably well even in strongly non equilibrium flows”. And: “While the above assumption for turbulent viscosity may not hold universally, especially in and around singularities such as stagnation, separation and reattachment, it is reasonably general”. One may notice that the profile proposed shows a discontinuity that seems nonphysical. However this is not a problem because this profile is useful just for the derivation of the law of the wall. We consider now the simplified two-dimensional momentum equation for the wall tangential direction and for a steady case:

$$\rho U\frac{\partial U}{\partial x} + \rho V\frac{\partial U}{\partial y} + \frac{\partial P}{\partial x} = C_U \tag{3.63}$$

$$C_U = \frac{\partial}{\partial y}[(\mu + \mu_t)\frac{\partial U}{\partial y}] \tag{3.64}$$

What we want to do is to integrate the expression above in the near wall cell in order to obtain an expression for wall shear stress as a function of center cell quantities. We can manage the integration because:

- The turbulent viscosity profile is given (equation (3.62)) and can be substituted in the expression.
- C_U is assumed to be constant and known from the previous iteration

The result of the integration, as reported in Popovac & Hanjalic paper, is:

$$\tau_w = \frac{\rho U_P - \frac{C_U y_p}{\kappa u_\tau}}{\frac{\rho y_v}{\mu} + \frac{\log\left(\frac{y_P}{y_v}\right)}{\kappa u_\tau}} \tag{3.65}$$

This expression represents a function between wall shear stress and quantities at near wall center cell, to be more precise wall tangential velocity u_P and wall distance y_p. In order to obtain an expression similar to SWF one, this expression must be rewritten. It's good to say that the two constants that appeared after the double integration of the equation (3.64) are deduced from the imposed conditions of continuity and smoothness of the velocity profile at the edge of sub-viscous layer y_v. This method is useful also to overcome the problem tied to the nonphysical meaning of the discontinuity in the turbulent viscosity profile assumed. Having a continuous and smooth profile of velocity at the interface between sub-viscous layer and turbulent region allows smoothing this discontinuity. In order to obtain an expression compatible with Standard Wall Functions we need to properly rewrite the expression of wall shear stress obtained.

The denominator is rearranged in this way:

$$\frac{\rho y_v}{\mu} + \frac{\ln\left(\frac{y_P}{y_v}\right)}{\kappa u_\tau} = \frac{1}{\kappa u_\tau}\left[\frac{\rho y_v \kappa u_\tau}{\mu} + \log\left(\frac{y_P}{y_v}\right)\right] \tag{3.66}$$

$$\frac{1}{\kappa u_\tau}\left[\frac{\rho y_v \kappa u_\tau}{\mu} + \log\left(\frac{y_P}{y_v}\right)\right] = \frac{1}{\kappa u_\tau}\log((e^{\kappa y_v^+}\frac{y^+}{y_v^+}) \tag{3.67}$$

$$\frac{1}{\kappa u_\tau}\log((e^{\kappa y_v^+}\frac{y^+}{y_v^+}) = \frac{1}{\kappa u_\tau}\log(E y^+) \tag{3.68}$$

Summarizing, the denominator is rewritten as:

$$\frac{\rho y_v}{\mu} + \frac{\ln\left(\frac{y_P}{y_v}\right)}{\kappa u_\tau} = \frac{1}{\kappa u_\tau}\ln(E y^+) \tag{3.69}$$

Using this rearrangement, it is possible to discover the physical meaning of constant E, which is the same constant present in Standard Wall Functions. The numerator becomes

$$\rho U_P - \frac{C_U y_p}{\kappa u_\tau} = \rho U_P \left(1 - \frac{C_U y_p}{\rho U_P \kappa u_\tau}\right) = \rho U_P \Psi \tag{3.70}$$

$$\rho U_P \left(1 - \frac{C_U y_p}{\rho U_P \kappa u_\tau}\right) = \rho U_P \Psi \tag{3.71}$$

Where Ψ is defined as

$$\Psi = 1 - \frac{C_U y_p}{\rho U_P \kappa u_\tau} \tag{3.72}$$

Ψ represents a non-equilibrium function. Its value gives a measure of how much the boundary layer is a non-equilibrium one. More correctly Ψ contains C_U, which is the term that represents the non-equilibrium terms, i.e. unsteady term, advective terms and pressure gradient term. If we are dealing with an equilibrium boundary layer C_U is null and therefore $\Psi = 1$. Using Ψ instead of C_U as a measure of non-equilibrium is more significant because the first one is dimensionless, while the second one (that is dimensional) can assume any values and we cannot assign them a physical meaning. Even though Ψ is already non-dimensional, it can be rewritten using only non-dimensional quantities.

$$\Psi = 1 - \frac{C_U y_p}{\rho U_P \kappa u_\tau} \frac{u_\tau v u_\tau}{u_\tau v u_\tau} \tag{3.73}$$

$$1 - \frac{C_U y_p}{\rho U_P \kappa u_\tau} \frac{u_\tau v u_\tau}{u_\tau v u_\tau} = 1 - \frac{y^+}{u^+ \kappa} C_U \frac{v}{\rho u_\tau^3} \tag{3.74}$$

Where we can call for simplicity

$$C_U^+ = \frac{v}{\rho u_\tau^3} C_U \tag{3.75}$$

It is now easy to reduce this function to a conventional Wall Functions expression:

$$\tau_w = \frac{\rho U \Psi}{\frac{1}{\kappa u_\tau} \log(E y^+)} \tag{3.76}$$

$$\frac{U u_\tau}{\frac{\tau_w}{\rho}} = \frac{1}{\kappa \Psi} \log(E y^+) \tag{3.77}$$

$$u^{+} = \frac{1}{\kappa \Psi} \log(Ey^{+}) \tag{3.78}$$

A comparison with SWF shows that the expressions are equal, except for the non-equilibrium wall function Ψ. If Ψ is equal to 1 the expression reduces to the standard log-law. This ensures that if GWF are used for dealing with an equilibrium boundary layer these become equal to SWF, which are known to work well in this situation. Regarding the constants κ and E, although the authors of the paper suggest their own values it is convenient to take the constant values in GWF equal to the ones used for SWF by Fluent 6.3. We make this choice because the goal of the work is to make a comparison between SWF and GWF in order to evaluate the goodness of the functions approaches themselves, not the goodness of the constants.

Actually the authors propose to modify the function above in order to avoid singularities when wall shear stress tends to zero. The modification is based essentially on the use of the other velocity scale u_k inside the expression, so that

$$y^{+} \rightarrow y^{*} \tag{3.79}$$

$$u^{+} \rightarrow u^{*} \tag{3.80}$$

$$\Psi = 1 - \frac{C_U y_p}{\rho U_P \kappa u_k} \tag{3.81}$$

Expression (3.81) can be written with only dimensionless quantities:

$$\Psi = 1 - \frac{y^{*}}{u^{*}\kappa} C_U \frac{\mu}{\rho^2 u_\tau^2 u_k} \tag{3.82}$$

The law of the wall becomes:

$$u^{*} = \frac{1}{\kappa \Psi} \log(E\, y^{*}) \tag{3.83}$$

Wall shear stress can be easily obtained from the inversion of the expression (3.83):

$$\tau_w = \frac{\rho \kappa u_k U}{\log(E\, y^{*})} \Psi \tag{3.84}$$

Until now, only the law of the wall useful for the determination of wall shear stress has been discussed. Regarding the other boundary conditions, they are evaluated in the following way.

- Production of turbulent kinetic energy is evaluated with the same approach used for SWF (see paragraph 3.6):

$$P_k = -\rho < u'v' > \frac{\partial U}{\partial y} \tag{3.85}$$

$$-\rho < u'v' > \frac{\partial U}{\partial y} = \frac{\tau_w}{\mu}\frac{\partial u^*}{\partial y^*} \tag{3.86}$$

The following is the expression proposed by the authors of GWF

$$\frac{\tau_w}{\mu}\frac{\partial u^*}{\partial y^*} \approx \rho\frac{u_k^3}{\Psi \kappa y_P} \tag{3.87}$$

- Turbulent dissipation rate is evaluated with the same expression of SWF (see equation (3.45)):

$$\varepsilon_P = \frac{u_k^3}{\kappa y_P} \tag{3.88}$$

4 Logical Path Followed for the Implementation of GWF

In order to write the user defined function (udf) for the implementation of GWF into Fluent 6.3, a preliminary study about the way to write it must be done. In particular it is necessary to define:

- the quantities required by GWF;
- the order to be followed during the iteration process for the computation of these quantities;
- which macros provided by Fluent must be used in order to carry out the purpose

This work is essential because it allows the writing of a draft udf, and, more important, it allows to understand which challenges we have to face with when we'll write the final udf.

4.1 Quantities Required by GWF

First of all, we have to reason on the quantities needed for the writing of GWF. Basically Wall Functions are based on a law of the wall which relates the dimensionless wall tangential velocity with the dimensionless wall distance. The law of the wall used by GWF is, as already presented in equation (3.83):

$$u^* = \frac{1}{\kappa \Psi} \log(E y^*) \tag{4.1}$$

Rigorously, all the quantities present inside the expression above must be computed. κ and E are constants and they are taken equal to the constant used by SWF, i.e.:

Table 4.1. Numerical values of the constants of the law of the wall for Generalized Wall Functions

κ	E
0.4187	9.793

The other quantities which must be computed are:

- Non equilibrium function Ψ

- Dimensionless wall distance y^*

Regarding Non-equilibrium function Ψ, it contains (see equation (3.82)):

- κ
- Dynamic viscosity μ
- Density ρ
- C_U
- y^*
- u^*
- Velocity scale u_τ
- Velocity scale u_k

The evaluation of the terms κ, μ and ρ is not a challenge because they are constant. First challenge is related to the assumptions behind GWF which states that the term C_U is constant inside the near wall cell and during the iteration process it is taken equal to the value computed at the previous iteration. Therefore we need to compute the values of C_U before the beginning of every iteration. The values must be computed using the center cell values. Second challenge regards which cells we must compute C_U for. These cells are all the near wall cells, which can be defined as all the ones which have one face who belongs to the wall. When writing the user defined function we have to find a command that allows to access all and only these cells. Another challenge we have to face with is the following: the term C_U comes from the writing of the wall tangential momentum equation. But the wall orientation can change with the cell we are considering and it's not mandatory that it is oriented as the x direction (or the y direction) of a Cartesian grid (or axial/radial direction for an axial symmetric coordinate system). Unfortunately, the data we can access from Fluent 6.3 are defined for a fixed xy Cartesian coordinate system (i.e. axis orientation does not change inside the domain), or for a cylindrical coordinate system (if we are dealing with an axial symmetric simulation); post-processing data are not defined for a coordinate system whose axis orientation changes with the orientation of the wall. Therefore a unique method for the computation of the quantities required in the wall based coordinate system using the quantities available from Fluent must be set. The solution of this problem will be shown and discussed later in this thesis, to be specific in chapter 6.

Regarding y^*, there is a slight difference between the explicit values of the law of the wall (the ones which are argument of the logarithm) and the ones inside Ψ. Theoretically for both of them we are required to provide the values computed at the previous iteration so that the code can use these values inside the expression of Ψ and consequently inside the law of the wall. For the explicit terms Fluent offers the user a Macro called DEFINE_WALL_FUNCTIONS that allows the introduction of a user law of the wall, in terms of $u^* = f(y^*)$. According to Fluent user defined manual, the values of y^* are taken directly from the solver; therefore there is no need to compute them inside the udf. On the other hand the values of y^* inside Ψ must be computed using the same logic of C_U mentioned before.

The computation of Ψ requires both the velocity scales u_τ and u_k. Regarding friction velocity, its computation can be carried out by accessing the wall shear stress at the previous iteration. Same thing must be done for u_k, where in this case we must access turbulent kinetic energy at previous iteration. Must be noted that friction velocity contains also density and u_k contains the term C_μ; however these two terms are not a challenge for us because these are constant.

4.2 Logical Scheme of CFD Code when It Adopts GWF for Wall Treatment

Using the preliminary observation about the quantities needed by GWF made in paragraph 4.1, we can define a draft of the logical scheme of how the CFD code works when using a user defined law of the wall.

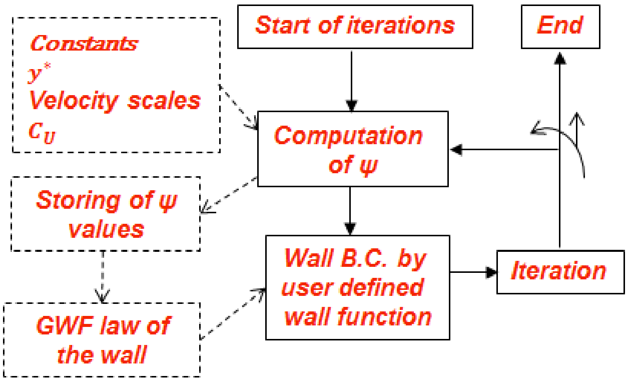

Figure 4.1. Logical scheme of CFD code when it adopts GWF

The scheme shown can be synthetized as follows:

- The looping process starts when we decide to start the iteration process, i.e. by clicking on *Solve* → *Iterate* button of Fluent 6.3.
- At this point, before the code operates the first iteration (or the next iteration, if the code has already performed the first iteration), we want to compute the values of Ψ for every near wall cell; the values used for

the computation are the ones used for the initialization of the solution, if the code is operating the first iteration, or are the ones stored from the previous iteration.

- The computed values of Ψ must be stored in order to use them in the second step of the process, i.e. inside the Macro DEFINE_WALL_FUNCTIONS.
- The Macro DEFINE_WALL_FUNCTIONS uses as input the law of the wall of GWF and it provides to the code all the wall boundary conditions
- At this point the code has all the information it requires and it can perform one iteration.
- After the iteration, we can exit the looping process by stopping manually the iteration or the looping process can continue. The process will restart from the second point of this list.

4.3 Macros and Commands of Fluent Useful for the Writing of the UDF

Regarding the second step of the logical scheme shown in paragraph 4.2, we need a Macro that allows to operate on the data right before the iteration. This Macro is named DEFINE_ADJUST, and its position in the calling sequence is shown in the Figure 4.2

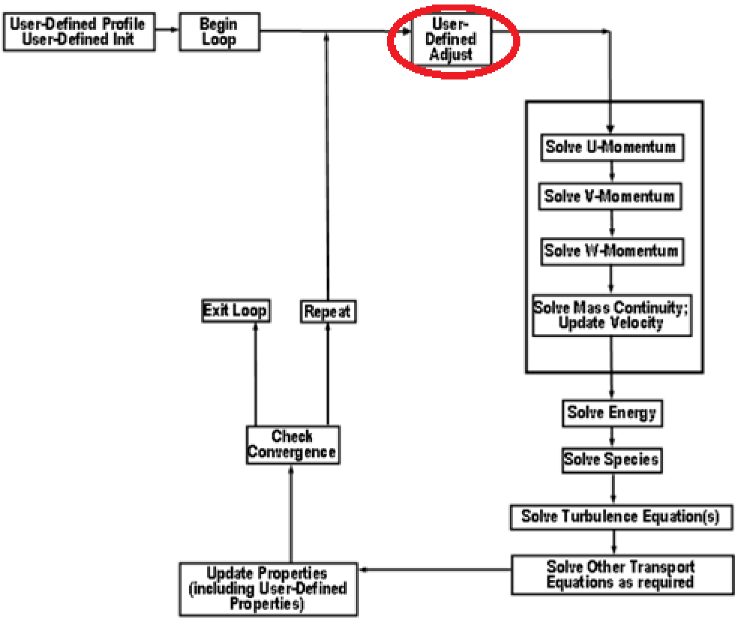

Figure 4.2. Calling sequence for DEFINE_ADJUST in a pressure based segregated solver

As we can see, this Macro is suitable for the goal we need because it is called from the solver before the beginning of the iteration. We can put all the process of evaluation of Ψ inside this Macro. The whole process is here shown.

First of all we must be able to access all and only the near wall cells; and we must access both the center cell and the face centroid (see Figure 4.3).

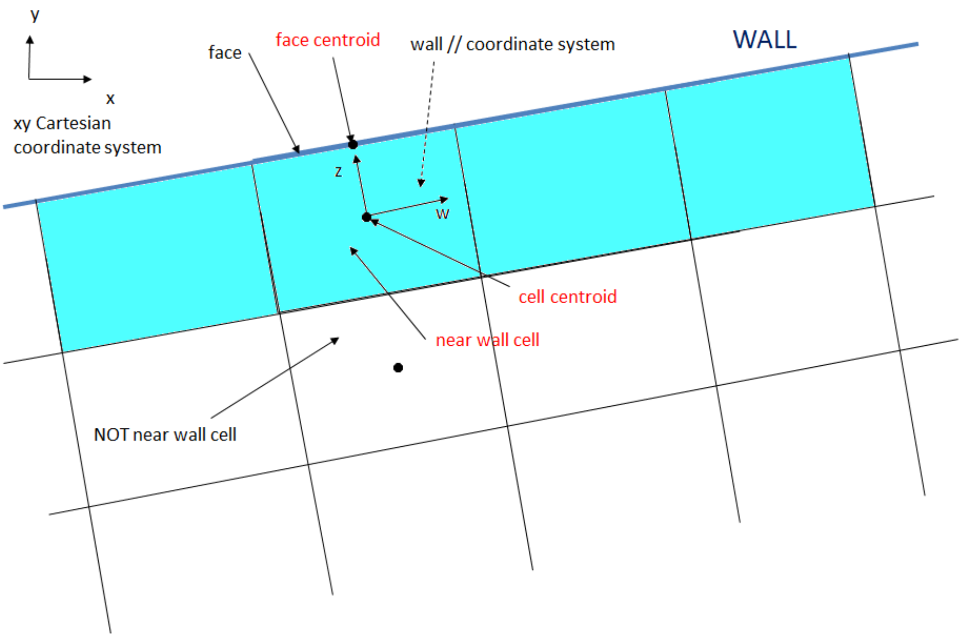

Figure 4.3. 2D Cartesian grid: near wall particular. Cells with light blue background are the one that are affected by the computation of Ψ

Before discussing the solution adopted it is good to describe the way a cell or a face can be identified by Fluent 6.3. Basically, every cell or face (and also every domain) is identified by using two variables: the first one represents the index number of the element, while the second one represents a thread pointing to this element. Since we want to compute Ψ values for every near wall cell, we must be able to create an index and a thread pointing to all and only these cells. The process used is here described. First of all, we must create a thread pointing to the wall. We can use this command:

- *tf = Lookup_Thread(d, zone_id)*

Where the arguments passed to this command are:

- d, which represents the pointer to the domain structure. We found out that the solver passes the pointer to the domain d to the Macro DEFINE_ADJUST
- zone_id, which represents the identifying number of the zone we want the thread to point. This number can be found in the *Define* → *Boundary conditions* panel in Fluent. By clicking on the zone we are interested (which is the wall) we can obtain the number needed.

The output of this command is the thread pointing to the wall, which is stored into the variable tf (which must be defined as thread type). Now we are able to access the data of centroids of wall faces, but we need also to access data of centroids of near wall cells. This purpose can be achieved by using the following two commands:

- *c = F_C0(f,tf);*
- *tc = THREAD_T0(tf);*

In order to explain the working of these two commands, it is necessary to take a look at the way Fluent 6.3 considers two adjacent cells (which can be defined as two cells which share a face).

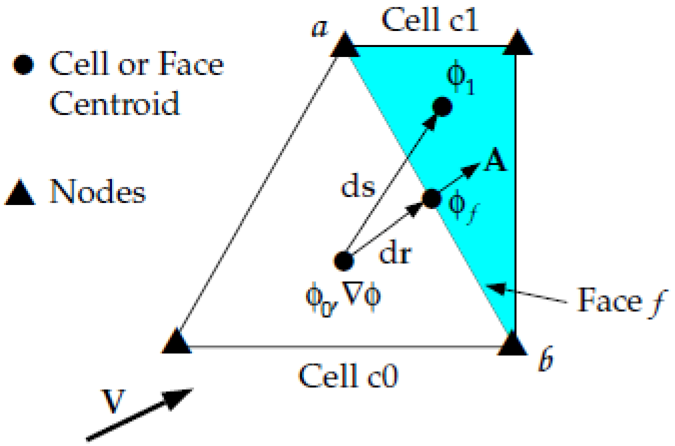

Figure 4.4. Two adjacent cells in Fluent 6.3

As the figure above shows, a generic face *f* belongs at the same time to two different cells, respectively called *c0* and *c1*. If a face is on the boundary of a domain, there is only one cell which owns it, which is *c0*. Therefore only *c0* exists. *F_C0(f,tf)* can be used to create the identification index of cells that are adjacent to a given face thread *tf*, if the face belongs to the wall. This command expands to a function that returns the index of a face's neighboring cells.

THREAD_T0 works with the same logic of *F_C0* but it expands to a function that returns the thread pointing to a face's neighboring cell.
In order to access all the wall face centroids and all the centroids of near wall cells we must use a looping macro, i.e. a macro that allows to loop between all the elements of a thread (in this case between all the faces the wall is divided). The looping macro chosen is the one that allows to loop over all faces in a face thread, whose syntax is the following:

```
begin_f_loop(f, tf)
        {
        }
end_f_loop(f, tf)
```

f represents the face index and *tf* the face thread. For every face the code performs all the commands written into the brackets, and then the loop continues for all the faces in a thread.

The operations to be put into the brackets are the ones needed for the computation of Ψ (see equation (3.82)). First of all we must define the constants. These are:

- κ
- C_μ

The definition of those constants inside the Macro DEFINE_ADJUST is not mandatory. We can simply define them at the beginning of the udf so that they are valid for all the function. The syntax which has been used is:

#define "name" "value"

And the values used are:

Table 4.2. Values of the constants defined into the udf for the GWF law of the wall

κ	C_μ
0.4187	0.09

Regarding dynamic viscosity and density of the fluid, we have two alternatives possible. Since we want to work with a fluid for which these properties are constant, we can define them in the same way used for κ and C_μ. The alternative is to access their values which are stored inside near wall center cells. This way has been chosen because it is more general and can be adopted also if these properties are not constant. The commands to be used for accessing these data at center cell are:

- ρ → *C_R(c,tc)*
- μ → *C_MU_L(c,tc)*

The cell we are accessing data is defined by the two parameter c and tc, which have been previously defined.

The challenge related to the computation of the term C_U has been already presented in paragraph 4.1. Since the solution adopted and the approach used for the obtainment of this term is tricky and long, the author has considered more effective to dedicate to it a whole chapter, to be more precise chapter 6. For this moment it's enough to say that the computation method adopted allows to obtain velocity components, velocity gradient and pressure gradient in a wall based coordinate system where the orientation of the axis changes with the orientation of the wall.

Next quantity to be computed is dimensionless wall distance y^*. Its expression is well known (see for example equation (3.23)). For its computation we need, besides density and dynamic viscosity, also velocity scale u_k and wall distance. Regarding the velocity scale u_k, its expression is given by equation (3.19). We need to access turbulent kinetic energy at the centroids of near wall cell. The command used is:

- k → *C_K(c,tc)*

Since this command allows to obtain turbulent kinetic energy at centroid of near wall cells, u_k can be easily computed. Regarding center cell wall distance, it can be obtained with the following command:

- Center cell wall distance y → *C_WALL_DIST(c,tc)*

However this command requires the udf to be compiled into Fluent, but since we just want to interpret the udf, a constant wall distance has been used. The error introduced by this assumption must be checked, but we can say "a priori" that the error will be small (and thus negligible) if the thickness of boundary layer mesh is constant all along the wall. In this case the wall distance used is equal to half of the thickness of boundary layer mesh.

For the evaluation of u^*, three quantities are needed: wall tangential velocity (which is available from the computation method adopted for the evaluation of

C_U), u_k and u_τ (see equation (3.22)). The challenge then is to access wall shear stress. There is no command that allows the user to access directly the wall shear stress, but the problem can be overcome in the following way. There is a command which allows to access wall shear, which is dimensionally a force and it is defined as:

$$\boldsymbol{wallshear} = \boldsymbol{\tau_w}\, A_{\mathrm{face}}\ [N] \qquad \textbf{(4.2)}$$

The command mentioned is:

- *NV_V(wallshear,=,F_STORAGE_R_N3V(f,tf,SV_WALL_SHEAR));*

This command stores the wall shear inside the variable *wallshear*, which is defined as vector type with the same dimension of the geometry considered. The way this vector is defined is:

- *real wallshear[ND_ND];*

Therefore simply dividing the components of the vector *wallshear* by the face area we obtain the wall shear stress.

$$\boldsymbol{\tau_w} = \frac{\boldsymbol{wallshear}}{A_{\mathrm{face}}}\ [\mathrm{Pa}] \qquad \textbf{(4.3)}$$

Face area can be obtained by using the following commands:

- *F_AREA(A,f,tf)*
- *A_Mag=NV_MAG(A)*

The result of these two commands is a vector *A[ND_ND]* whose absolute value *A_Mag* is equal to the area of the face. The vector representing wall shear stress (*P[ND_ND]*) can then be obtained dividing the components of vector *wallshear* by *A_MAG*

- *P[0]=wallshear[0]/AMag;*
- *P[1]=wallshear[1]/AMag;*

And the absolute value of wall shear stress, which is the value to put inside the expression of friction velocity, is evaluated as:

- *PMag=NV_MAG(P);*

So far, we have computed all the terms necessary for the writing of the function Ψ, which values can be computed using equation (3.82).
After the Ψ values have been computed, the same values must be stored in order to use them inside the macro DEFINE_WALL_FUNCTION. Fluent offers the user the possibility to store values inside two different types of so-called user defined memories (UDMI). These are respectively called F_UDMI and C_UDMI, and their syntax is

- *F_UDMI(f,tf,"number of udmi")="variable to be stored"*
- *C_UDMI(c,tc,"number of udmi")="variable to be stored"*

The first one can be used to store variables on a face (defined by the two parameters *f* and *tf*), while the second can be used store variables on a center cell (defined by the two parameters *c* and *tc*). In our situation it is suitable to allocate the values of variable Ψ inside the center cell; then C_UDMI has been used.

So far, we have just discussed about the computation of the values of the non-equilibrium function Ψ. We haven't spoken yet about the second step of the process, i.e. the macro DEFINE_WALL_FUNCTIONS, which is used for the implementation of a user defined wall function. Briefly, the Macro allows the user to introduce a user defined law of the wall, and the code, starting from the values obtained from the Macro, computes all the wall boundary conditions. It is necessary to make a deep study of the working of the macro. Unfortunately there is very little documentation available online related to this macro. Therefore a deep study about the macro is important in order to make a smarter use of CFD and not to use it just as a black box tool. The results of the study are discussed in the chapter 5.

5 Macro DEFINE_WALL_FUNCTIONS

As said in chapter 4, Fluent 6.3 provides a macro for the writing of user defined wall functions, called DEFINE_WALL_FUNCTIONS. Due to the lack of documentation available about it, it's convenient to check how the macro works, in terms of:

- variables passed from the solver to the macro;
- quantities that the user is required to provide;
- macro output;
- the way adopted by CFD code for the use of macro output.

5.1 Case Studied for the Check

The geometry used for the study of the macro is the conical diffuser from the experiment made by Karunakaran et al [12]. It's important to state that at this point the quality of the results isn't important. The purpose of this chapter is only to find out how macro DEFINE_WALL_FUNCTION works. Therefore, the mesh adopted is a coarse mesh and the boundary conditions have been simplified. 2D axisymmetric and steady simulations have been carried out. All the simulations have been performed with a second order discretization scheme for all the advective terms. Fluent 6.3 has been opened with a single precision solver. Figure 5.1 represents the domain, the mesh and the boundary conditions which has been adopted. Table 5.1, Table 5.2, Table 5.3 and Table 5.4 show geometrical data, flow conditions, inlet boundary conditions and the parameters of the mesh adopted.

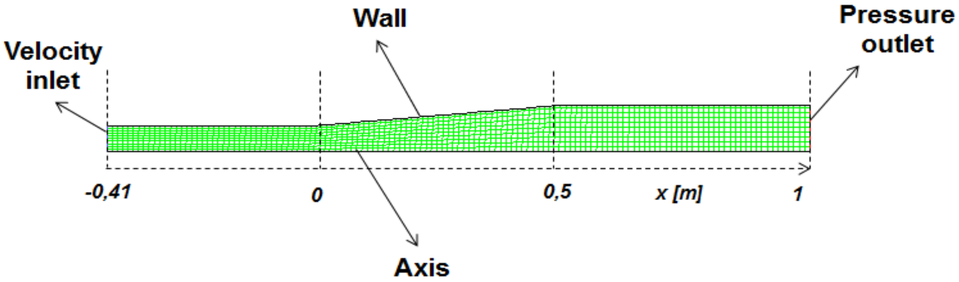

Figure 5.1. Domain, mesh and boundary conditions used for the study of macro DEFINE_WALL_FUNCTIONS

Table 5.1. Geometrical parameters of the CFD domain adopted for the study of macro DEFINE_WALL_FUNCTIONS

Coordinate system origin	At diffuser inlet	
Inlet section		
Diameter	0,107	m
Length	4,10E-01	m
Diffuser		
Length	5,00E-01	m
Inlet diameter	0,107	m
Outlet diameter	0,195	m
Outlet pipe		
Diameter	0,195	m
Length	500	m

Table 5.2. Flow conditions of the case adopted for the study of macro DEFINE_WALL_FUNCTIONS

Re	256000	
µ	1,46E-05	m^2/s
ρ	1,225	kg/m^3

Table 5.3. Inlet boundary conditions of the case adopted for the study of macro DEFINE_WALL_FUNCTIONS

Type	Velocity inlet	
Velocity profile	Flat profile	
Velocity	35	m/s
Turbulence intensity	1	%
Hydraulic diameter	0,107	m

Table 5.4. Parameters of the mesh of CFD domain adopted for the study of macro DEFINE_WALL_FUNCTIONS

Boundary layer. mesh thickness	0,005	m
Quadrilateral core elements characteristic length	0,0107	m
Number of elements	1320	

Regarding the mesh, it must pointed out that, even if core mesh has been made very coarse, near wall mesh has been realized with a constant thickness such that y^+ at outlet is bigger than 50. Therefore at inlet y^+ is higher (because since velocity is higher then also wall shear stress and consequently friction velocity are higher), and so there should be no risk that one or more cells fall into sub-viscous layer. That one is the only constraint used for mesh creation. The step by step process for the evaluation of the boundary layer mesh thickness adopted is reported in the Table 5.5.

Table 5.5. Step by step process for the creation of near wall mesh adopted for the study of macro DEFINE_WALL_FUNCTIONS

Velocity outlet	10,54	m/s
Reynolds outlet	140472	
Cf/2 outlet	2,01E-03	
Friction velocity inlet	0,47	m/s
y+ outlet	50,00	
y centroid outlet	1,55E-03	m
y layer outlet	3,09E-03	m
y layer choosen	5E-3	m
Cf/2 inlet	0,001734	
Friction velocity inlet	1,457372	m/s
y+ inlet	249,06	

The step by step procedure adopted is here described:

- Bulk velocity at outlet is computed using continuity equation::

$$U_{\text{outlet}} = \frac{\rho U_{\text{inlet}} A_{\text{inlet}}}{\rho A_{\text{outlet}}} \tag{5.1}$$

- Computation of Reynolds number at outlet using bulk velocity at outlet
- Friction coefficient at outlet estimated with the following expression (note: this is strictly valid for a pipe) [5]

$$\frac{C_f}{2} = 0.039\ Re_D^{-0.25} \tag{5.2}$$

$$\frac{C_f}{2} = \frac{\tau_w}{\rho U^2} \tag{5.3}$$

- Wall shear stress (and consequently friction velocity, which is defined by equation (1.9)) is computed from the inversion of expression (5.3)

- y^+ of 1st center cell at outlet is set to 50, in order to make it fall into fully turbulent region.
- Center cell wall distance at outlet is evaluated from an inversion of the definition of y^+ (equation (1.8)):

$$y = \frac{y^+ \mu}{\rho u_\tau} \tag{5.4}$$

- The thickness of boundary layer mesh at outlet is obtained by doubling the value just obtained. However, since the correlation used for the friction coefficient is strictly valid only for a pipe flow, the value obtained has been rounded (to be specific it has been rounded from 3.09E-3 m to 5E-3 m); it is like we have applied a factor of safety.
- The same thickness of boundary layer mesh has been adopted for the entire wall.
- y^+ at inlet is evaluated using the friction velocity that comes from the correlation for friction factor already used (see equation (5.2)). However, for the purpose of this chapter there are no constraints related to this quantity.

5.2 Macro Description

First of all, here is reported the macro definition, according to Fluent user defined functions manual [26].

- *DEFINE_WALL_FUNCTIONS (name, f, t, c0, t0, wf_ret, yPlus, Emod)*

Table 5.6. Variables and output type of the macro DEFINE_WALL_FUNCTIONS

Argument Type	*Description*
symbol name	UDF name.
face_t f	face index.
Thread *t	pointer to cell thread
cell_t c0	cell index.
Thread *t0	pointer to face thread.
int wf_ret	wall function index
real yPlus	y+ value
real Emod	wall function E constant
Function returns	real

An example of the Macro (which is taken from Fluent 6.3 user defined manual) is reported below. It is important to state that this is the only example found online. Moreover, since there is not a complete guide about this Macro, most part of the information which have been found out and have reported in this chapter comes from a study of this example and of the results obtained from the implementation of it into Fluent.

```
/****************************************************************
*** User-defined wall functions: separated into turbulent and
laminar regimes
/****************************************************************
**/#include "udf.h"

DEFINE_WALL_FUNCTIONS(user_log_law, f, t, c0, t0, wf_ret, yPlus,
Emod)
{
  real wf_value;

  switch (wf_ret)
    {
    case UPLUS_LAM:
      wf_value = yPlus;
      break;
    case UPLUS_TRB:
      wf_value = log(Emod*yPlus)/KAPPA;
      break;
    case DUPLUS_LAM:
      wf_value = 1.0;
      break;
    case DUPLUS_TRB:
      wf_value = 1./(KAPPA*yPlus);
      break;
    case D2UPLUS_TRB:
      wf_value = -1./(KAPPA*yPlus*yPlus);
      break;
    default:
        printf("Wall function return value unavailable\n");
    }
  return wf_value;
}
```

Below it is reported the description of the macro made by Fluent 6.3 Udf manual:

> “There are eight arguments to DEFINE_WALL_FUNCTIONS: name, f, t, c0, t0, wf_ret, yPlus, and Emod. You supply name, the name of the UDF. f, t, c0, t0, wf_ret, yPlus, and Emod are variables that are passed by the **FLUENT** solver to your UDF. Your UDF will need to compute

> the real value of the wall functions U+, dU+, and dY+ for laminar and turbulent regions and return them to the solver.”

The example shown above represents the writing of Standard Wall Functions through a user defined wall functions. We can state that because, looking at the function written, we can see that the term *UPLUS_TRB* represents the law of the wall used in Standard Wall Functions (equations (3.8) and (3.24)).

$$wf_{\text{value}} = \log(Emod * yPlus)\ /KAPPA \qquad \textbf{(5.5)}$$

5.3 Variables Passed from the Solver to the Macro

It’s good to check what the variables passed to the software actually are. This purpose has been achieved thanks to the *printf* command used into the udf example shown in paragraph 5.2, which has been interpreted into Fluent 6.3. The logic followed is here described. The code has been run just for one iteration and the quantities we want to check have been printed on the screen. Then we have compared the output of *printf* with data available from Fluent post-processing.

```
#include "udf.h"

#define VKC 0.4187

DEFINE_WALL_FUNCTIONS(swf, f, t, c0, t0, wf_ret, yPlus, Emod)
{
  real wf_value;
          real M[ND_ND];
          real Q[ND_ND];
          int a, b;
          F_CENTROID(M,f,t);
          C_CENTROID(Q,c0,t0);
          a=THREAD_ID(t0);
          b=THREAD_ID(t);
  switch (wf_ret)
    {
    case UPLUS_LAM:
      wf_value = yPlus;
      break;
    case UPLUS_TRB:
      wf_value = log(Emod*yPlus)/VKC;
    break;
  case DUPLUS_LAM:
    wf_value = 1.0;
    break;
  case DUPLUS_TRB:
    wf_value = 1./(VKC*yPlus);
    break;
  case D2UPLUS_TRB:
    wf_value = -1./(VKC*yPlus*yPlus);
    break;
  default:
          printf("Wall function return value unavailable\n");
  }
          printf("%f x f centroid ", M[0]);
          printf("%f x c centroid ", Q[0]);
          printf("%i t0 ", a);
          printf("%i t ", b);
          printf("%f yPlus ", yPlus);
          printf("%f Emod ", Emod);
          printf("%f k\n", KAPPA);
return wf_value;
```

Note: *F_CENTROID(M,f,t)* stores into the vector M the position of the face centroids; *C_CENTROID(Q,c0,t0)* stores the position of cell centers.

The results obtained are: the following

- *Emod* = 9.793

  This is the standard value of E constant inside SWF law of the wall, and it is already implemented into Fluent. It’s possible to change this value in *Define* → *Model* → *Viscous* Fluent 6.3 panel

- *yPlus* → y^*

  This conclusion has been obtained with a check with y^* values exctracted from post-processing.

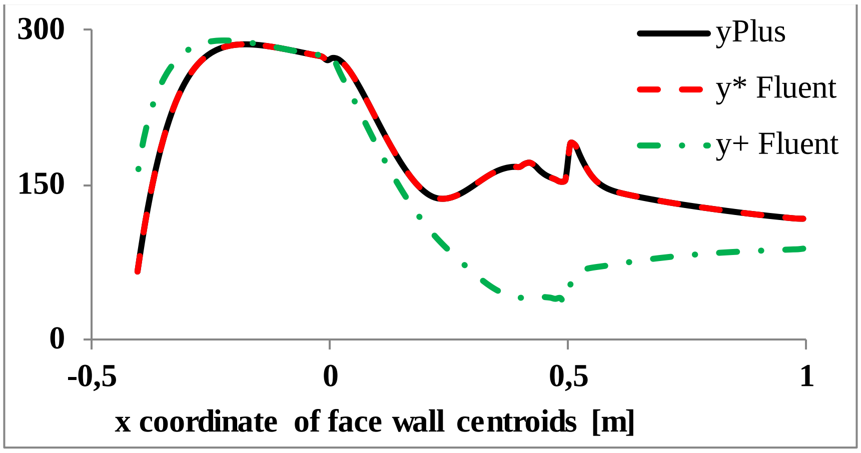

Figure 5.2. Values for all the wall faces of yPlus (the variable passed from the solver to the Macro), y^+ and y^* (which are extracted from Fluent post-processing)

Therefore the user defined wall function adopts the * *adimensionalization.*

- thread t → it has been found out that this is the pointer to face thread, and not to cell thread (which is what Fluent udf manual reports).

This check has been made thanks to the command *THREAD_ID(t)*. This command is the opposite of the command *Lookup_Thread* (described in paragraph 4.3). It gives as output the number of the zone the thread t is pointing to. In this situation the number printed on the screen is 3. Wall ID number (on *Define* → *Boundary condition* Fluent 6.3 panel) is 3. In Table 5.7 are reported the ID number of all the boundaries of the domain.

Table 5.7. ID number of all the boundaries of the geometry adopted

Name	ID
Axis	6
Defaul interior	8
Fluid	2
inlet	5
Outlet	4
Wall	3

- thread t0: it is the pointer to cell thread, and not to face thread (as Fluent udf manual reports); to be more precise it is the pointer to the cells adjacent to the wall since the number printed on the screen is 2 (see Table 5.7)

- *KAPPA* = 0.4187

This is a constant value which is already present inside Fluent 6.3, and it represents the Von Karman constant. Regarding this value there is no difference in using the command *KAPPA* or in introducing a constant with the same value.

- *c0* and *f* are cell index and face index. This can be deduced by the statement of Fluent udf manual. This result cannot be verified.

5.4 Quantities the User Must Provide

From an analysis of the example udf presented before, we can affirm that the user must provide to the macro the following quantities

- A dimensionless law of the wall in the form $u^* = f(y^*)$. Two different version of the law of the wall must be provided. The first one is the law of the wall of the sub-viscous layer. This law of the wall is always linear, as discussed in paragraph 1.2. The law adopted is therefore:

$$u^* = y^* \tag{5.6}$$

The second version to provide is the law of the wall for the fully turbulent region. This is the law of the wall which is proposed by the Wall Functions we want to implement

- The first derivative of the two laws of the wall which have been described at the previous point. The law of the wall is derived with respect to the dimensionless wall distance y^* $(\frac{\partial u^*}{\partial y^*})$. For the laminar case, the first derivative is simply

$$\frac{\partial u^*}{\partial y^*} = 1 \tag{5.7}$$

- The second derivative, where the derivation is still performed with respect to the dimensionless wall distance y^*, $(\frac{\partial^2 u^*}{\partial y^{*2}})$ of the only law of the wall of the fully turbulent region.

5.5 Output of the Macro

Next step concerns the study of the output of the macro DEFINE_WALL_FUNCTIONS, which is represented by the variable *wf_value*. The use of the command *printf* before the command *return* allows to print on the screen the values of some interesting variables, interesting for the understanding of the output of the macro. Here the commands used and their position inside the udf are presented:

```
printf("%f x fcentroid ", M[0]);
printf("%i wf_ret ", wf_ret);
printf("%f wf_value ", wf_value);
printf("%f yPlus \n", yPlus);
return wf_value;
```

We printed also *wf_ret* because it is the integer parameter used by the switch command. Therefore from the values assumed by this parameter we can know which expression is used by the code to compute the correspondent *wf_value*. We printed also the x coordinate of the face centroid because this variable allows to check how many different *wf_values* are computed for every cell. The logic of the switch command is here explained (note: the turbulent law of the wall implemented is the SWF law of the wall; the udf shown as an example before has been interpreted into Fluent 6.3):

Table 5.8. Summary of the switch command for the Macro DEFINE_WALL_FUNCTIONS

wf_ret	*wf_value*
0	y^*
1	$\frac{1}{\kappa}\log(E\,y^*)$
2	1
3	$\frac{1}{\kappa y^*}$
4	$-\frac{1}{\kappa y^{*2}}$

From an analysis of the *printf* output we notice that there are 4 different sets of data, i.e. there's a first loop over the wall faces and for every face some different values of *wf_value* are computed, then there is a second loop, and the same process continues until the fourth loop. The values computed for every loop are shown in Table 5.9.

Table 5.9. Output of the macro DEFINE_WALL_FUNCTION: *wf_value*

Loop	Output			
1	$\frac{1}{\kappa}\log(E\,y^*)$	$\frac{1}{\kappa y^*}$	-	-
2	$\frac{1}{\kappa}\log(E\,y^*)$	-	-	-
3	$\frac{1}{\kappa y^*}$	$\frac{1}{\kappa}\log(E\,y^*)$	$\frac{1}{\kappa y^*}$	
4	$\frac{1}{\kappa}\log(E\,y^*)$	$\frac{1}{\kappa y^*}$	$\frac{1}{\kappa y^*}$	$-\frac{1}{\kappa y^{*2}}$

In this case all the quantities printed on the screen are computed using the fully turbulent law of the wall and its first and second derivatives. This happens because all the values of both y^* and y^+ for every near wall center cells are higher than dimensionless sub-viscous layer, which has been kept equal to the default value of Fluent 6.3 (see Table 3.1). Figure 5.3 shows the values of y^* and y^+ for every near wall center cells and the dimensionless thickness of sub-viscous layer. An analysis about the way the dimensionless thickness of sub-viscous layer is defined is presented further down.

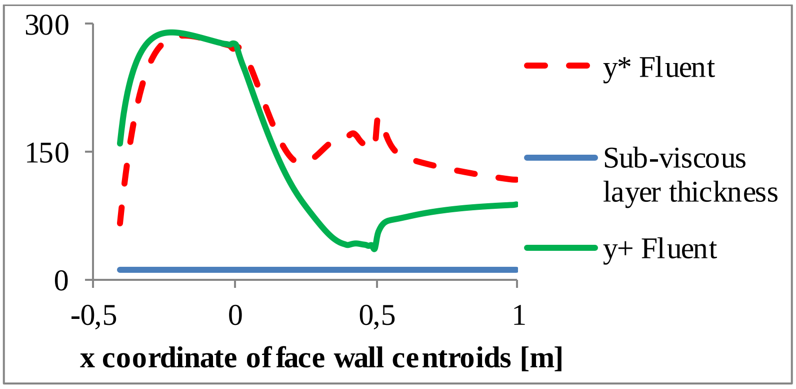

Figure 5.3. y* and y+ values of near wall center cells and dimensionless thickness of sub-viscous layer

Thus far, we have not discussed yet about the threshold value that represents the dimensionless thickness of the sub-viscous layer. This value can be changed into *Define* → *Model* → *Viscous* panel of Fluent 6.3 (note: it can be changed only when a user defined wall function is adopted). By default this value is defined in the following way:

$$y_V^+ = 11.225 \tag{5.8}$$

We can ask two questions about this topic:

- Is the dimensionless value of sub-viscous layer thickness given as y_V^+ or as y_V^*?
- What happens if we modify this value in order to make some cells fall into sub-viscous layer?

In order to answer these questions the following work has been performed. First work made was the change of the dimensionless value of sub-viscous layer thickness so that it will be greater than the minimum value of y^+ and at the same time lower than the minimum value of y^*. In the simulation run before, the minimum values obtained of y^+ and y^* are respectively:

$$y_{\text{minimum}}^+ = 35.93 \tag{5.9}$$

$$y_{\text{minimum}}^* = 65.77 \tag{5.10}$$

The new thickness of sub-viscous layer has been chosen in order to have it greater than y_{minimum}^+ but at the same time smaller than y_{minimum}^*. The value chosen is

$$\text{Dimensionless thickness of subviscous layer} = 40 \tag{5.11}$$

The output of the *printf* commands already adopted doesn't change, i.e. all the values of *wf_value* are still computed using the fully turbulent law of the wall. The second thickness of sub-viscous layer is then chosen in order to have it greater also than y_{minimum}^*. The value chosen is:

$$\text{Dimensionless thickness of subviscous layer} = 70 \tag{5.12}$$

Again, the output of the *printf* commands already adopted has been checked. With this dimensionless thickness of sub-viscous layer the output changes. It has been noticed that the macro before all performs a check of the dimensionless wall distance y^* of the near wall center cells in order to verify if the value of it is greater or smaller than the dimensionless thickness of sub-viscous layer y_v^*. If it is smaller, the output of the macro changes compared to the output shown in Table 5.9. Table 5.10 show the generalized output of the macro.

Table 5.10. Generalized output of macro DEFINE_WALL_FUNCTIONS

Loop	*Output*			
1	$\begin{cases}\frac{1}{\kappa}\log(E\,y^*)\,if\ y^* > y_v^* \\ y^*\ if\ y^* < y_v^*\end{cases}$	$\begin{cases}\frac{1}{\kappa y^*}\,if\ y^* > y_v^* \\ 1\ if\ y^* < y_v^*\end{cases}$		
2	$\begin{cases}\frac{1}{\kappa y^*}\,if\ y^* > y_v^* \\ 1\ if\ y^* < y_v^*\end{cases}$			
3	$\frac{1}{\kappa y^*}$	$\frac{1}{\kappa}\log(E\,y^*)$	$\frac{1}{\kappa y^*}$	
4	$\frac{1}{\kappa}\log(E\,y^*)$	$\frac{1}{\kappa y^*}$	$\frac{1}{\kappa y^*}$	$-\frac{1}{\kappa y^{*2}}$

The conclusion that can be derived is that the dimensionless thickness of sub-viscous layer is given with the dimensionless wall distance that uses u_k as velocity scale; in other words it is assigned the value of y_v^*.

5.6 How does Fluent Use Macro Output?

Since the purpose of this work is the implementation of a wall function different to SWF, it is better to check how wall boundary conditions (wall shear stress, production of turbulent kinetic energy and turbulent dissipation rate) are computed when using a user defined wall function. Remembering the expression for Standard Wall Functions shown in paragraph 3.6 it is possible to make a guess at how Fluent uses the output of the macro DEFINE_WALL_FUNCTIONS for the computation of wall boundary conditions. The guess is:

$$\tau_w = \frac{\rho U u_k}{f(y^*)} \tag{5.13}$$

$$P_k = \frac{\tau_w}{\mu} f'(y^*) \tag{5.14}$$

$$\varepsilon = \frac{u_k^4}{\upsilon} f'(y^*) \tag{5.15}$$

However, it is necessary to check if these expressions are correct. This check has been carried out by implementing SWF through the udf example shown above, and also implementing some different wall functions through udf.

The logic followed for carrying out this work is here presented.

- Computation of the values of wall shear stress, P_k and ε at the near wall center cells with the expression shown above
- Comparison of the quantities obtained with the same quantities extracted from Fluent 6.3 post-processing.
- Values of relative errors have been computed

$$\% \text{ error} = \frac{|\text{Fluent value} - \text{Computed value}|}{\text{Fluent value}} 100 \qquad \textbf{(5.16)}$$

Figure 5.4 shows the values of the relative errors, defined by expression (5.16), which have been obtained with the SWF implemented with the udf shown as an example in paragraph 5.2.

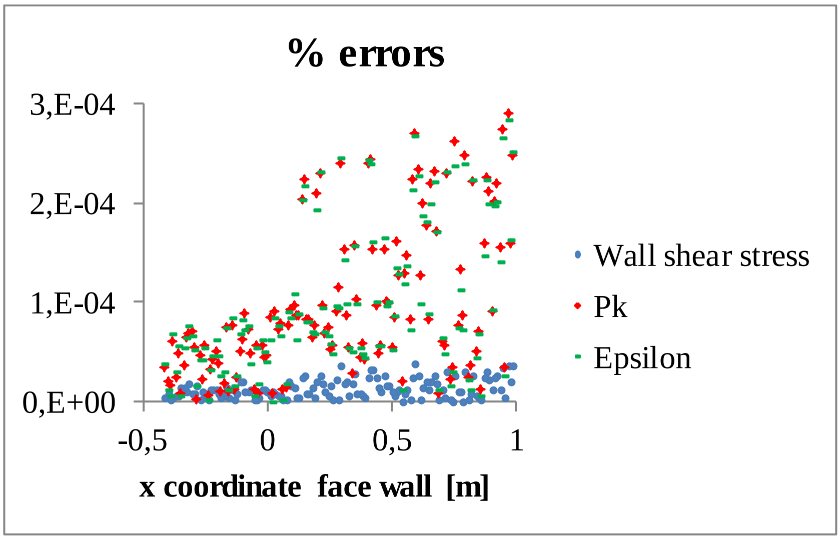

Figure 5.4. Relative percentage error between quantities obtained from Fluent and quantities computed with guess expressions obtained with the SWF implemented through udf

Since the relative percentage errors are very small (less than $3 * 10^{-4}$; then they are almost negligible) it can be stated that with the SWF implemented through udf the expression guessed are right. Introducing the SWF law of the wall inside the expressions (5.13), (5.14) and (5.15) we obtain:

$$\tau_w = \frac{\rho U u_k}{\frac{\log(Ey^*)}{\kappa}} \tag{5.17}$$

$$P_k = \frac{\frac{\tau_w^2}{\mu}}{\kappa y^*} \tag{5.18}$$

$$\varepsilon = \frac{\frac{u_k^4}{v}}{\kappa y^*} \tag{5.19}$$

While the expression for the wall shear stress is already equal to the one reported in Fluent user guide and proposed by SWF authors (equation (3.25)), the other two must be rewritten in order to check if they are equal to the way proposed for SWF (expressions (3.44) and (3.45)). The results for the production of turbulent kinetic energy is

$$P_k = \frac{\tau_w^2}{\mathrm{\mu}} \frac{\mu}{\kappa \rho u_k y_P} \tag{5.20}$$

$$P_k = \frac{\tau_w^2}{\kappa \rho u_k y_P} \tag{5.21}$$

The result for turbulent dissipation rate is:

$$\varepsilon = \frac{\frac{\rho u_k^4}{\mu}}{\frac{\kappa \rho y_P u_k}{\mu}} \tag{5.22}$$

$$\varepsilon = \frac{u_k^3}{\kappa y_P} \tag{5.23}$$

The expressions (5.21) and (5.23) are equal respectively to the equations (3.44) and (3.45). Therefore SWF implemented through udf is equal to SWF already implemented into Fluent 6.3.

In order to generalize this work it is necessary to make the just presented check about the guessed expressions (5.13), (5.14) and (5.15) also with different Wall Functions. The Wall Functions used are:

$$\begin{cases} u^* = \frac{1}{\kappa}\log(Ey^*) + C \\ C = 1e-1 \end{cases} \tag{5.24}$$

$$\begin{cases} u^* = \frac{1}{\kappa'}\log(Ey^*) \\ \kappa' = 1.1\kappa \end{cases} \tag{5.25}$$

$$\begin{cases} u^* = \frac{1}{\kappa}\log(E'y^*) \\ E' = 9 \end{cases} \tag{5.26}$$

$$\begin{cases} u^* = \frac{1}{\kappa}\log(Ey^*) + Cy^* \\ C = 1e-1 \end{cases} \tag{5.27}$$

$$\begin{cases} u^* = \frac{1}{\kappa}\log(Ey^*) + Cy^{*2} + Dy^* \\ C = 5e-6 \\ D = 1e-4 \end{cases} \tag{5.28}$$

The chart of the relative percentage errors (expression (5.16)) obtained for all these functions is similar to the one obtained for the SWF implemented through udf (Figure 5.4). Therefore we can conclude that the expression guessed before are generally right.

5.6.1 Computation of Wall Boundary Conditions when Near Wall Center Cell Falls into Sub-viscous Layer

So far, the only thing which has not been discussed yet is how wall boundary conditions are computed for near wall cells whose center falls into sub-viscous layer. We must follow the procedure shown before, i.e. make a guess at the expressions Fluent uses for the computation of wall boundary condition, and verify them “a posteriori” by using post-processing data. The law of the wall adopted for this work is the SWF implemented through udf. In order to force near wall center cells to fall into sub-viscous layer, we increase the value of the dimensionless thickness of sub-viscous layer (y_v^*) to 100, while all the other conditions are taken equal to the ones used in previous paragraph. With this dimensionless thickness of sub-viscous layer all the near wall center cells fall into sub-viscous layer. If near wall center cell falls into sub-viscous layer, than the dimensionless velocity profile is linear, as already shown (see paragraph 1.2 and equation (5.6)). The expression for the computation of wall shear stress can be easily determined, i.e. substituting (5.6) into (5.13) we obtain:

$$\tau_w = \frac{\rho U u u_k}{y^*} \tag{5.29}$$

Expression (5.29) can be rewritten using the definition of y^*. It becomes then

$$\tau_w = \frac{\rho U u_k}{\frac{\rho y u_k}{\mu}} \tag{5.30}$$

$$\tau_w = \mu \frac{U}{y} \tag{5.31}$$

The challenge instead concerns the way wall boundary conditions for the turbulent quantities are computed. The rigorous expressions for production of turbulent kinetic energy and turbulent dissipation rate for a point inside sub-viscous layer are:

$$P_k = -\rho < u'v' > \frac{\partial U}{\partial y} \tag{5.32}$$

But the production of turbulent kinetic energy becomes null if the profile for Reynolds stresses shown in Figure 3.3 is adopted.

$$\varepsilon = \frac{2\ vk}{y_v^2} \tag{5.33}$$

A check "a posteriori" of these expressions allow us to state that the expressions (5.32) (5.33) are wrong. An alternative can be represented by the substitution of the first derivative of the laminar law of the wall in the expressions (5.14) and (5.15). The substitution leads to:

$$P_k = \frac{\tau_w}{\mu} \tag{5.34}$$

$$\varepsilon = \frac{u_k^4}{v} \tag{5.35}$$

However, after a check, the expressions (5.34) and (5.35) are found out to be wrong. The last option that can be taken into account is that turbulent wall boundary conditions are computed using the turbulent expressions (5.18) and (5.19); i.e. the expressions (5.14) and (5.15) adopt the first derivative of the law of the wall for the fully turbulent region. Therefore, even though the center cells fall into sub-viscous layer, turbulent properties are computed as if the same center cells fall into fully turbulent region. Figure 5.5 show the relative percentage errors (defined by expression (5.16)) obtained with the use of expressions (5.18) and (5.19) for the computation of wall boundary conditions.

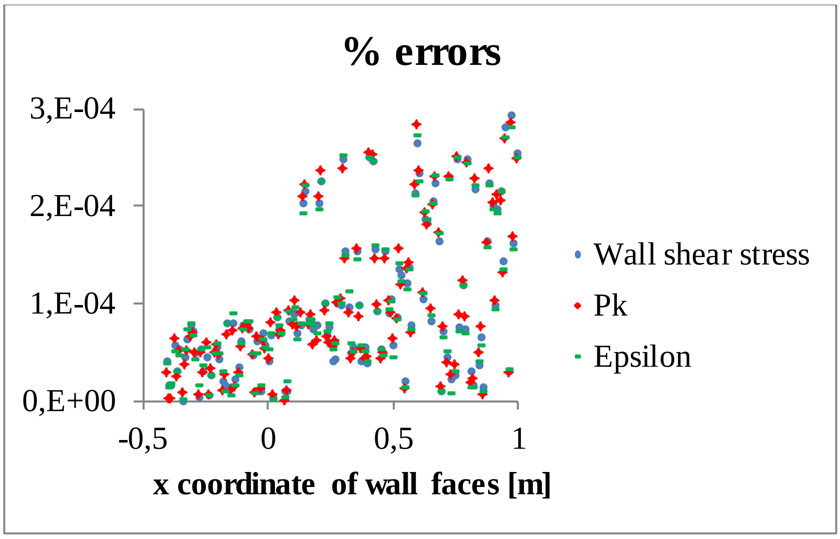

Figure 5.5. Relative percentage error between quantities obtained from Fluent and quantities computed with guessed expressions (5.18) and (5.19); near wall center cells fall into sub-viscous layer

6 Computation of Ψ

As discussed in the chapter 4, one big challenge in the writing of GWF concerns the way to write the term C_U. This term is defined by equation (3.63) and it comes from the writing of the wall tangential momentum component equation. Therefore the coordinate system which the term C_U is defined for is not fixed, but it is dependent on the orientation of the wall. If the wall is parallel to x-axis, we don't face a big challenge in the writing of C_U. However, in the general case, the wall cannot be parallel to x-axis. It is therefore necessary to define a unique computation method if we want to write C_U for every case study. The step by step method adopted is here presented.

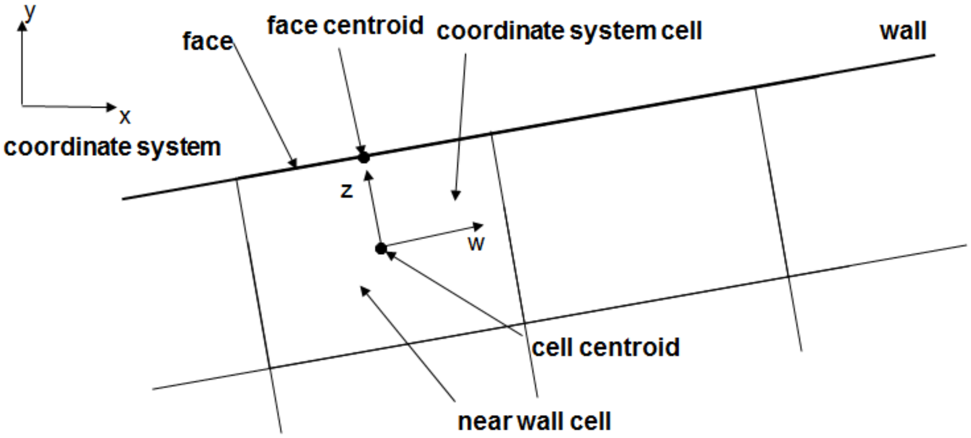

Figure 6.1. Wall based coordinate system (*wz*) and fixed Cartesian coordinate system (*xy*)

Figure 6.1 shows the situation we are dealing with. The fixed coordinate system is called xy while wz is the wall based coordinate system. To be more precise w represents the direction parallel to the wall and z represents the direction perpendicular to the wall. Using this nomenclature, we can rewrite C_U using the proper quantities. Note: equation (3.63) has been written for a situation where the wall was parallel to x-direction.

$$C_{Uw} = \rho U_w \frac{\partial U_w}{\partial \boldsymbol{w}} + \rho U_z \frac{\partial U_w}{\partial \boldsymbol{z}} + \frac{\partial P}{\partial \boldsymbol{w}} \tag{6.1}$$

In this expression U_w and U_z are respectively the component of velocity vector in the direction parallel and perpendicular to the wall. The derivatives which appear into the equation represent the two directional derivatives of U_w in the direction parallel and perpendicular to the wall. The last term represents the derivative of the pressure in the direction parallel to the wall. The unsteady term can be neglected since in this master thesis work we are dealing with steady flows. Summarizing, we need the following quantities:

$$U_w \,;\; U_z \tag{6.2}$$

$$\frac{\partial U_w}{\partial \boldsymbol{w}} \,;\, \frac{\partial U_w}{\partial \boldsymbol{z}} \tag{6.3}$$

$$\frac{\partial P}{\partial \boldsymbol{w}} \tag{6.4}$$

But into the udf we are able to access only the following quantities:

$$U \,;\, V \tag{6.5}$$

$$\nabla U \;;\, \nabla V \tag{6.6}$$

$$\nabla P \tag{6.7}$$

Note: these quantities are defined for the fixed xy coordinate system. The commands to be used in order to access them are:

- for the velocity components

  C_U(c,tc);
  C_V(c,tc);

- for the components of velocity vector gradient

  DUDX = C_DUDX(c,tc);
  DUDY = C_DUDY(c,tc);
  DVDX = C_DVDX(c,tc);
  DVDY = C_DVDY(c,tc);

- for pressure gradient

  DPDX = C_P_G(c,tc)[0];
  DPDY = C_P_G(c,tc)[1];

The parameters c and tc are respectively the index of and the thread pointing to the near wall cells. The method for the obtainment of these parameters has been already presented in paragraph 4.3.

A method that allows to obtain the desired quantities from the accessible data must be set. The method used for an axial symmetric conical diffuser geometry is here described step by step.

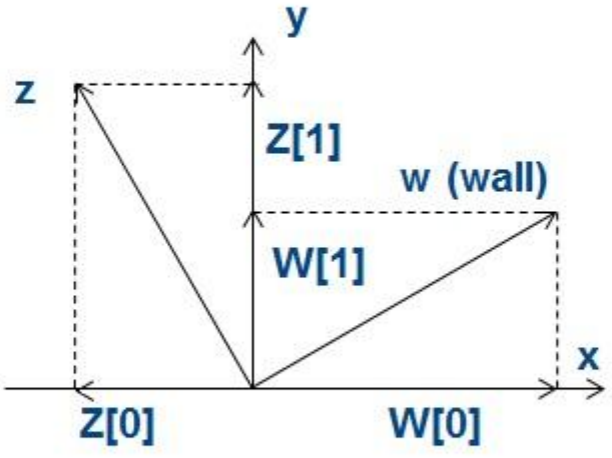

Figure 6.2. Unit vectors defining the two coordinate systems (*wz* and *xy*)

First step is the evaluation of the unit vectors which define the wall based coordinate system *wz*. This purpose is carried out thanks to the command (already discussed in paragraph 4.3) *F_AREA(A,f,tf)*. Besides allowing to obtain the value of the area of the face, this command returns a vector (which is stored in the variable A) whose direction is perpendicular to the face and, if the face belongs to the boundary, it points out of the domain. Its absolute value is equal to the value of face area. The unit vector can be obtained just by the normalization of the vector A (i.e. by the division of the single components of the vector A by its absolute value). The unit vector has been called *z*, and it represents the direction perpendicular to the wall of a wall based coordinate system. The unit vector representing *w* direction derives from a 90° clockwise rotation of *z* vector (see Figure 6.2).

$$\boldsymbol{w}[0] = \boldsymbol{z}[1] \tag{6.8}$$

$$\boldsymbol{w}[1] = -\boldsymbol{z}[0] \tag{6.9}$$

At this point we have the xy components of the two unit vectors representing the wall based coordinate system wz. Next step is the evaluation of the angle between the two coordinate systems. The angle is called α and it is the angle between x and w directions. It is positive if counterclockwise (see Figure 6.3).

$$\alpha = \arcsin(W[1]) \tag{6.10}$$

Now that we have completely defined the wall based coordinate system, we can move on the decomposition of the velocity vector in the wall based coordinate system. First of all it is necessary to determine the angle between velocity vector and x-axis, called β (see Figure 6.3). Starting from the values of U and V, we know that:

$$U = |\boldsymbol{V}| \cos(\beta) \tag{6.11}$$

$$V = |\boldsymbol{V}| \sin(\beta) \tag{6.12}$$

Dividing the second by the first we obtain

$$\frac{V}{U} = tan(\beta) \tag{6.13}$$

Therefore β can be easily computed as:

$$\beta = \arctan(\frac{V}{U}) \tag{6.14}$$

Once we have the value of β we are able to compute the value of the angle between the wall and velocity vector, called γ (see Figure 6.3).

$$\gamma = \beta - \alpha \tag{6.15}$$

One should note that the angle γ is negative if the flow is moving away from the wall; it is positive in the other situation. If γ is equal to zero, this means that the flow is moving parallel to the wall. Using the angle γ it's easy to obtain the velocity component in w and z direction, provided that we have computed the absolute value of velocity.

$$|\boldsymbol{V}| = \sqrt{U^2 + V^2} \tag{6.16}$$

$$U_w = |\boldsymbol{V}| \cos(\gamma) \tag{6.17}$$

$$U_z = |\boldsymbol{V}|\, sin(\gamma) \tag{6.18}$$

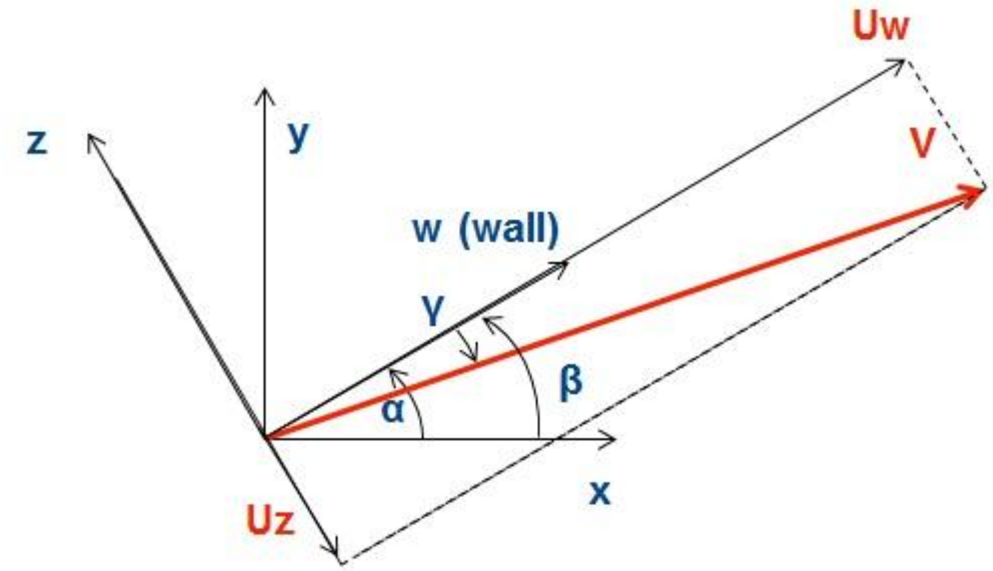

Figure 6.3. Velocity decomposition in w and z direction and characteristic angles

The derivative of the pressure in *w* direction can be easily obtained if we remember the definition of directional derivative. Since we can access the two components of pressure gradient (which are the partial derivatives of pressure in direction *x* and *y*), and can evaluate the directional derivative of p in *w* direction. The directional derivative of a quantity Q evaluated along a direction defined by the vector $\boldsymbol{x}$ is defined as:

$$\frac{\partial Q}{\partial \boldsymbol{x}} = \nabla Q \cdot \boldsymbol{x} \tag{6.19}$$

Where $\boldsymbol{x}$ represents a unit vector which points in the direction which we want to compute the directional derivative for. ∇Q is the gradient of the quantity Q evaluated with a coordinate system xy, $\cdot$ is the operator of the scalar product between two vectors. The vector $\boldsymbol{x}$ must be decomposed into the direction x and y of the fixed coordinate system xy. The expression (6.19) applied for the pressure leads to:

$$\frac{\partial P}{\partial \boldsymbol{w}} = \nabla P \cdot \boldsymbol{w} = \frac{\partial P}{\partial x}\boldsymbol{w}[0] + \frac{\partial P}{\partial y}\boldsymbol{w}[1] \tag{6.20}$$

Regarding the terms $\frac{\partial U_w}{\partial w}$ and $\frac{\partial U_w}{\partial z}$, the logic behind their computation is a bit more complex. These two terms are the components in direction w and z of the gradient of the wall tangential velocity component. The logical path followed for their computation is:

- Evaluation of ∇U_w in the direction x and y. This purpose can be carried out by finding a function that relates U_w to U and V, so that

$$U_w = f(U, V) \qquad \textbf{(6.21)}$$

The function must have the form of

$$U_w = aU + bV \qquad \textbf{(6.22)}$$

In the expression (6.22) a and b are two constant; basically we must be able to express U_w as a lienar combination of U and V. Indeed, if the function has the form of equation (6.22), the evaluation of the derivative of U_w in x direction becomes:

$$\frac{\partial U_w}{\partial x} = \frac{\partial (aU + bV)}{\partial x} = \frac{\partial (aU)}{\partial x} + \frac{\partial (bV)}{\partial x} = a\frac{\partial U}{\partial x} + b\frac{\partial V}{\partial x} \qquad \textbf{(6.23)}$$

$$\frac{\partial (aU + bV)}{\partial x} = \frac{\partial (aU)}{\partial x} + \frac{\partial (bV)}{\partial x} \qquad \textbf{(6.24)}$$

$$\frac{\partial (aU)}{\partial x} + \frac{\partial (bV)}{\partial x} = a\frac{\partial U}{\partial x} + b\frac{\partial V}{\partial x} \qquad \textbf{(6.25)}$$

And therefore $\frac{\partial U_w}{\partial x}$ can be evaluated by using only available and known quantities. The challenge is the determination of the coefficients a and b. Note: the same procedure must be adopted for the evaluation of $\frac{\partial U_w}{\partial y}$. Once $\frac{\partial U_w}{\partial x}$ and $\frac{\partial U_w}{\partial y}$ are computed, using the definition of directional derivative we can evaluate $\frac{\partial U_w}{\partial \boldsymbol{w}}$ and $\frac{\partial U_w}{\partial \boldsymbol{z}}$.

Here is described the procedure followed for the evaluation of the coefficient a and b of the linear combination defined by the expression (6.22).

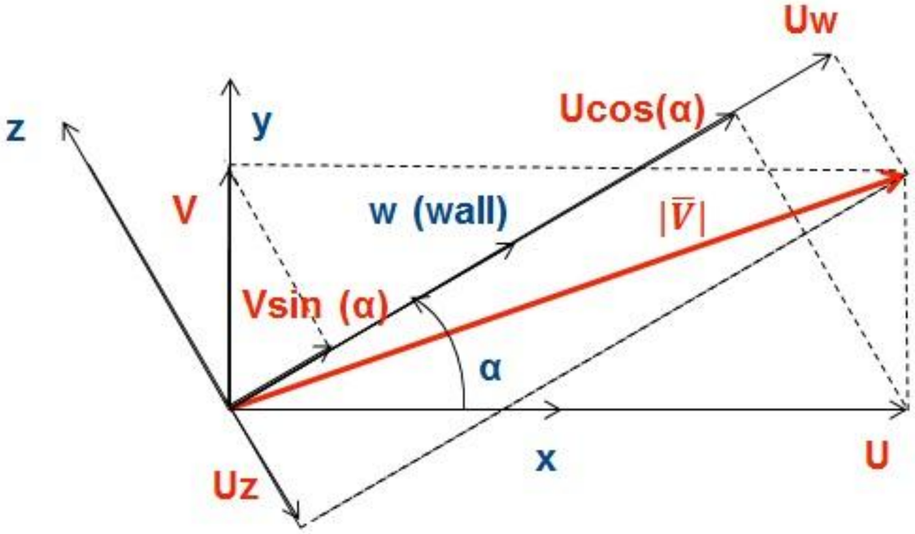

Figure 6.4. Linear combination of U and V in order to obtain U_w and U_z

The starting point is the two following equalities that comes from the decomposition of velocity vector into two different coordinate systems:

$$\boldsymbol{V} = U\boldsymbol{i} + V\,\boldsymbol{j} \tag{6.26}$$

$$\boldsymbol{V} = U_w\boldsymbol{w} + U_z\boldsymbol{z} \tag{6.27}$$

Where $\boldsymbol{i}$ and $\boldsymbol{j}$ are the unit vectors defining the fixed xy coordinate system and $\boldsymbol{w}$ and $\boldsymbol{z}$ are the unit vectors defining the wz wall based coordinate system. What we want is to write $\boldsymbol{i}$ and $\boldsymbol{j}$ as function of $\boldsymbol{w}$ and $\boldsymbol{z}$. In particular we want to write them as a linear combination of the two unit vectors defining the wall based coordinate system.

$$\boldsymbol{i} = f(\boldsymbol{w},\boldsymbol{z}) = c_1\boldsymbol{w} + c_2\boldsymbol{z} \tag{6.28}$$

$$\boldsymbol{j} = f(\boldsymbol{w},\boldsymbol{z}) = c_3\boldsymbol{w} + c_4\boldsymbol{z} \tag{6.29}$$

Since we are dealing with a linear combination problem, we can evaluate the values of c_1 and c_2 of expression (6.28) by writing two equations for the two component in the direction x and y:

$$\begin{cases} 1 = c_1\cos(\alpha) + c_2\ \cos(\alpha + \frac{\pi}{2}) \\ 0 = c_1\sin(\alpha) + c_2\sin(\alpha + \frac{\pi}{2}) \end{cases} \tag{6.30}$$

The solution of this system is:

$$\begin{cases} c_2 = -\sin(\alpha) \\ c_1 = \cos(\alpha) \end{cases} \tag{6.31}$$

Therefore we have an expression for the unit vector $\boldsymbol{i}$ as a function of $\boldsymbol{w}$ and $\boldsymbol{z}$:

$$\boldsymbol{i} = \boldsymbol{w}\cos(\alpha) - \boldsymbol{z}\sin(\alpha) \tag{6.32}$$

The same procedure has been followed for $\boldsymbol{j}$

$$\begin{cases} 0 = c_3\cos(\alpha) + c_4\ \cos(\alpha + \frac{\pi}{2}) \\ 1 = c_3\sin(\alpha) + c_4\sin(\ \alpha + \frac{\pi}{2}) \end{cases} \tag{6.33}$$

The solution of the system is:

$$\begin{cases} c_3 = \sin(\alpha) \\ c_4 = \cos(\alpha) \end{cases} \tag{6.34}$$

So that

$$\boldsymbol{j} = \boldsymbol{w}\sin(\alpha) + \boldsymbol{z}\cos(\alpha) \tag{6.35}$$

We can now substitute the expressions obtained for $\vec{\imath}$ and $\vec{\jmath}$ inside the expression (6.26).

$$\boldsymbol{V} = U(\boldsymbol{w}\cos(\alpha) - \boldsymbol{z}\sin(\alpha)) + V(\boldsymbol{w}\sin(\alpha) + \boldsymbol{z}\cos(\alpha)) \tag{6.36}$$

$$\boldsymbol{V} = \boldsymbol{w}\ (U\cos(\alpha) + V\ \sin(\alpha))\ + \boldsymbol{z}\ (-U\sin(\alpha) + V\ \cos(\alpha)) \tag{6.37}$$

But the expression just shown must be equal to the expression of velocity shown in the expression (6.27). Equalizing the two expressions we obtain

$$\begin{cases} U_w = U\cos(\alpha) + V\sin(\alpha) \\ U_z = -U\sin(\alpha) + V\cos(\alpha) \end{cases} \tag{6.38}$$

This is the linear combination we were looking for. Using the expressions (6.38), (6.23) and (6.19) we can evaluate in sequence:

$$\frac{\partial U_w}{\partial x}; \frac{\partial U_w}{\partial y} \tag{6.39}$$

$$\frac{\partial U_w}{\partial \boldsymbol{w}}; \frac{\partial U_w}{\partial \boldsymbol{z}} \tag{6.40}$$

The first two terms arise from these expressions:

$$\frac{\partial U_w}{\partial x} = \cos(\alpha)\frac{\partial U}{\partial x} + \sin(\alpha)\frac{\partial V}{\partial x} \tag{6.41}$$

$$\frac{\partial U_w}{\partial y} = \cos(\alpha)\frac{\partial U}{\partial y} + \sin(\alpha)\frac{\partial V}{\partial y} \tag{6.42}$$

And represents the gradient of U_w evaluated in the fixed coordinate system xy. From the definition of directional derivative (expression (6.19)), we can evaluate the components of the gradient of U_w in the directions of the wall based coordinate system wz.

$$\frac{\partial U_w}{\partial \boldsymbol{w}} = \frac{\partial U_w}{\partial x}\, \boldsymbol{w}[0] + \frac{\partial U_w}{\partial y}\boldsymbol{w}[1] \tag{6.43}$$

$$\frac{\partial U_w}{\partial \boldsymbol{z}} = \frac{\partial U_w}{\partial x}\, \boldsymbol{z}[0] + \frac{\partial U_w}{\partial y}\boldsymbol{z}[1] \tag{6.44}$$

The last two terms are the last ones which are required for the writing of the term C_U.

7 Performance of GWF

The purpose of this chapter is the evaluation of the goodness of the results obtained with the use of GWF for the wall treatment. It is desirable to obtain improved results when using GWF than when using SWF or NEWF. This because the assumptions used for the derivation of GWF should have a more general validity field than the ones behind SWF and NEWF. Moreover, the law of the wall which is adopted by GWF is partially sensitized to all the non-equilibrium effects of the near wall flow thanks to the function Ψ. On the other hand the law of the wall of SWF is always constant and it has no dependence at all on the conditions off the near wall flow. NEWF can be considered an intermediate solution because their law of the wall is sensitized to possible pressure gradients in the near wall region, but the advective effects are not considered at all. However, we do not expect that the use of GWF allows to obtain results from a CFD simulations which have a perfect accordance with experimental data. Indeed, CFD is related to several different aspects that must be taken into account at the same time if we want to obtain an acceptable goodness of the results. Besides wall treatment, we can mention between these aspects the turbulence model adopted, the grid used for the subdivision of the domain in finite volumes and the order of the discretization schemes adopted for the advective terms. At least we expect that the results strictly related to the wall treatment, e.g. wall shear stress, improve. Note: in this master thesis work the energy equation is not considered.

The geometry chosen for the study of the performance of GWF is a conical diffuser geometry which presents an incompressible, axial symmetric and steady flow. One characteristic of that the geometry taken into account that is desirable is that the assumptions behind SWF are not strictly valid; at the same time this geometry shows a quite simple flow behavior (e.g. there is neither separating flow nor reattaching flow, which can cause convergence problems with GWF). Regarding the first characteristic, the pressure gradient is not null, the derivatives of the wall tangential velocity along the two directions of the wall based coordinate system (see Figure 6.1) are not null (the one evaluated in w direction is not null because the cross section is increasing, and therefore the velocity should decrease; the one evaluated in z direction is not null because of the non-slip condition), and moreover it is expected that the component of the velocity in the direction perpendicular to the wall is not null for all the diffuser. These characteristics are not considered in the derivation of SWF, while NEWF take into account only the pressure gradient. Regarding the second characteristic (a quite simple flow), if the angle of the diffuser is not too high, there is no separating flow. It's expected that the results obtained with SWF and NEWF will not be right, and that the use of GWF will provide better results. One should note that the comparison of GWF with NEWF is opportune in order to evaluate the concrete utility of the use of GWF. Indeed, if the results obtained with GWF are worse (or at least similar) than the ones obtained with NEWF, the interest in the use of GWF would be low. In this situation there is more convenience in using NEWF (already implemented into CFD code and consequently more stable and reliable) than in using GWF. Besides the characteristics already exposed, another reason that drove me to choose this geometry is the simplicity of the geometry itself. CFD simulations of an axial symmetric conical diffuser can be carried out with 2D axial symmetric simulations. This leads to an easier writing of the user defined function because of the 2D geometry and because the wall is just practically a straight line. This characteristic has a great importance and it allows a relatively easy writing of the udf for the implementation of GWF. Let's make a simple example in order to clarify this concept. In a conical diffuser for every near wall cell the distance from the wall of the center cell is univocally defined. On the other hand, if we take a backward facing step, not all the center cells in the region where the flow recirculates have a unique definition of wall distance. There are the cells close to the lowest corner where two different wall distances can be computed, e.g. the distance from the vertical wall and the distance from the horizontal wall. The first challenge is therefore the way wall distance is computed. Moreover the flow in these cells has not a prevalent velocity direction, e.g. close to the vertical wall it can have a not negligible horizontal component and vice versa. But the term C_U of GWF is derived starting from the wall tangential component of momentum equation. Therefore another challenge arises: which component of momentum equation should we use? Last thing, the wall based coordinate system is defined in this way: $\boldsymbol{w}$ is the direction parallel to main flow direction, and $\boldsymbol{z}$ is the direction perpendicular to the wall. The direction we choose for $\boldsymbol{z}$ affects neither the sign of the term $\rho U_z \frac{\partial U_w}{\partial \boldsymbol{z}}$ nor its absolute value. On the other hand, the direction of w affects the sign of the other two terms, $\rho U_w \frac{\partial U_w}{\partial w}$ and $\frac{\partial P}{\partial w}$. The first term does not change with w direction, while the second does. Close to the vertical edge of backward facing step there will be some cells where the flow goes upward and other where the flow goes downward. The conclusion of it is that there is not a unique way to compute C_U. Figure 7.1 presents all the challenges related to the backward facing step just discussed in this paragraph.

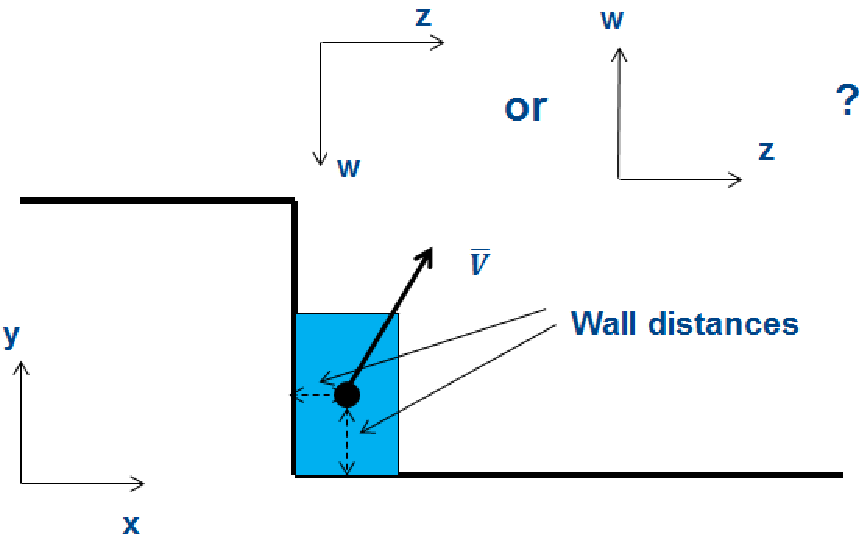

Figure 7.1. Lower corner cell of a backward facing step: challenges that arise from the writing of GWF for this cell

These challenges do not arise in a conical diffuser geometry, and therefore the implementation of GWF is simpler and the results obtained are not affected by these problems. Moreover, the method for the computation of C_U which has been set in chapter 6 is suitable for a conical diffuser geometry. The experimental apparatus and data set used for the comparison of CFD results are taken by the following paper by Trupp et al.: “Trupp A.C., Azad R.S., Kassab S.Z., Near-wall velocity distribution within a straight conical diffuser, Experiments in Fluids 4, 319-331 (1986)” [1].

7.1 Description of Experimental Apparatus and of Experimental Data Obtained

The scheme of the experimental facility used by the authors is shown in the Figure 7.2; the Table 7.1 summarizes the geometrical data of the facility and the experimental flow conditions:

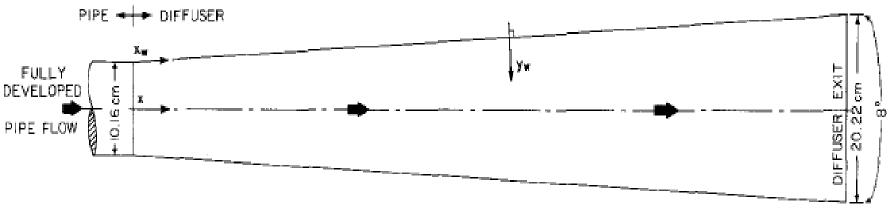

Figure 7.2. Trupp et al. experimental apparatus

Table 7.1. Data of Trupp et al. experimental apparatus

Feeding pipe length	7,512	m
Feeding pipe diameter	0,1016	m
L/D feeding pipe	73,93	
Feed of diffuser	Fully developed turbulent pipe flow	
Bulk velocity at inlet	18,06	m/s
Reynolds number at inlet	115000	
Diffuser outlet diameter	0,2022	m
Diffuser length	0,72	m
Diffuser angle α	4	°
A out / A in	4	

Regarding the experimental data, the authors provide a data set for seventeen different stations inside the diffuser (plus one extra station placed inside the feeding pipe). To be more precise the data provided are:

- dimensionless static pressure at the wall;
- friction velocity;
- centerline velocity (called U_c);
- dimensionless pressure gradient parameter Δ (which is related to axial kinematic pressure gradient α);
- dimensionless mean axial velocity profiles (provided in (y^+, u^+) charts) for every station of measurement. Note: the points of measurement for these profiles have been taken starting from the wall and moving in the wall perpendicular direction (called y_w in the Figure 7.2)

One should note that two different coordinate systems have been used by the experiment authors. The first one is defined by the axial coordinate x and the radial coordinate y; the second is defined by the wall coordinate x_w (which corresponds to the wall itself) and by the direction perpendicular to the wall y_w. The two coordinate systems are visible in Figure 7.2.

Table 7.2. Quantities measured for every station and position of the stations inside the diffuser

Station	x_w *[cm]*	U_c	*α [m/s²]*	u_τ *[m/s]*
1	1	21,1	-208	0,82
2	3	20,82	585	0,795
3	6	20,2	598	0,642
4	9	19,45	502	0,565
5	12	18,8	405	0,483
6	15	18,2	324	0,425
7	18	17,7	260	0,38
8	21	17,3	211	0,345
9	24	16,9	176	0,316
10	30	16,3	129	0,272
11	36	15,8	99,4	0,242
12	42	15,3	79,7	0,22
13	48	14,85	66,3	0,201
14	54	14,4	55,2	0,188
15	60	14	47	0,176
16	66	13,45	40,4	0,17
17	71	13,05	34,6	0,164

First work to do is to check if dimensionless experimental velocity profiles follows or not (and if not the differences must be pointed out) the logarithmic law of the wall of SWF. This work has been already carried out by the authors of the paper. The authors report in the paper that “departures from the law of the wall begin as soon as the flow enters the diffuser. This is seen first (Station 1) in the buffer region, and later (beginning with station 2) in the outer portion of the fully turbulent region. There is a rapid erosion of the width of the log law such that by Station 6 there is little evidence of a log law at any substantial thickness. However, beginning at about Station 11, the width of the log region appears to begin to increase and the departures in the outer region becomes progressively less. Hence in the final stages of the diffuser, the log law appears to re-emerge”. Moreover: “In some respects, the above described pattern suggests two boundary layers; the inner one developing on the diffuser wall, and the outer one representing a decaying remnant of the feed pipe boundary layer. In any event, since the outer log regions have slopes which are both abnormally high and variable, and hence do not scale on the local wall shear stress, the possibility of fitting these regions by log law distributions is not considered further”. The authors have found out that the velocity profiles can be fitted with the following laws, which are (starting from the wall): logarithmic law, inner power law, linear law and outer power law (this is present only until Station 9). Here are presented the experimental dimensionless profiles for feeding pipe, Station 1, 5, 10 and 15 compared to the SWF log law.

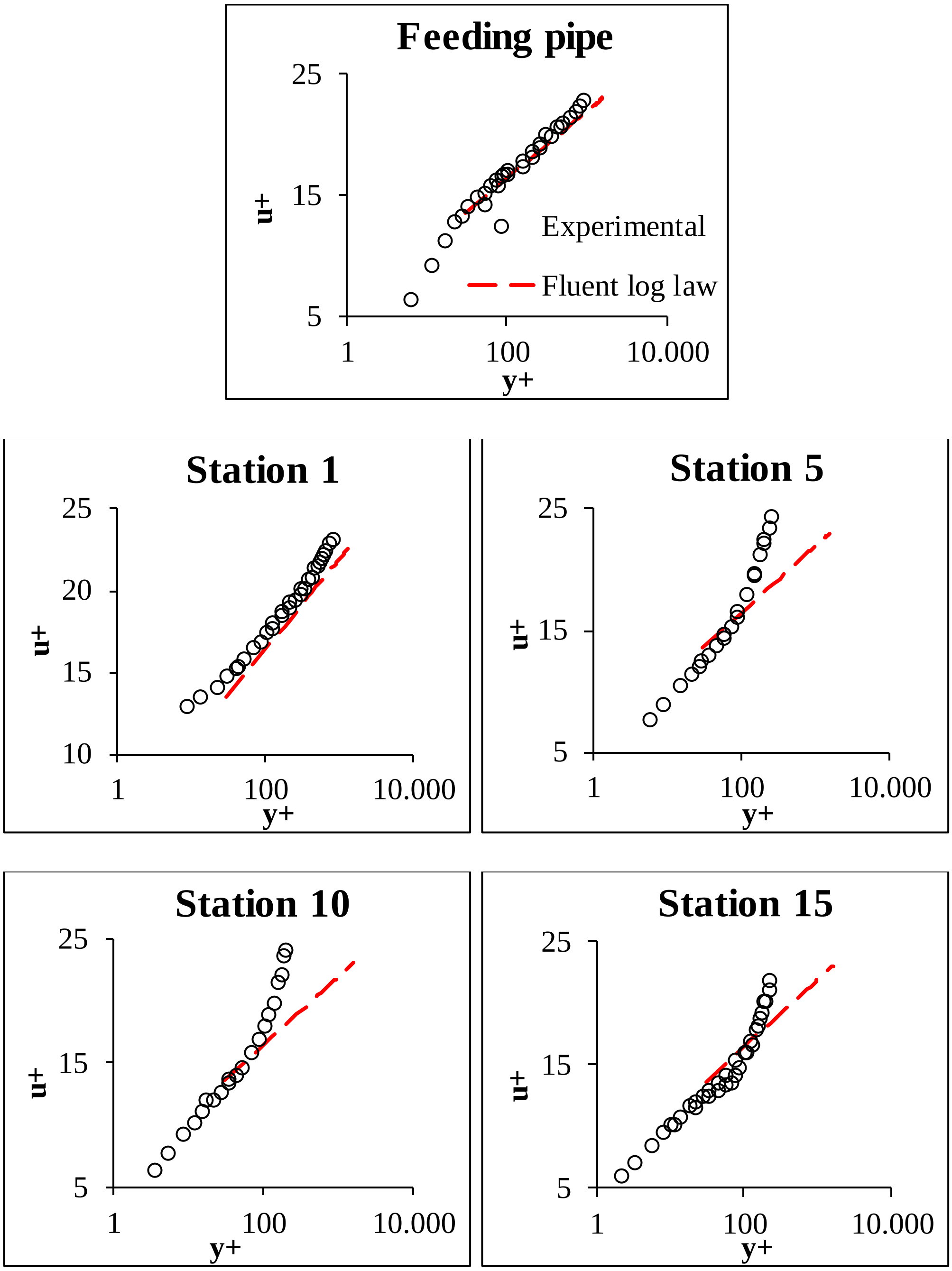

Figure 7.3. Experimental dimensionless velocity profile (y+, u+) for feeding pipe and stations 1, 5, 10, 15; also SWF log law is shown

The width of the range of the logarithmic law of the wall has been evaluated by the authors, and the results in terms of y^+ are presented in the Table 7.3. Note: if the symbol ~ is used, it means that the log-law can extend to values external to the one written, because that one is an extreme point of measurement; therefore the value of y^+ is not the boundary of the log region. The data of Table 7.3 have been plotted in a chart, which is presented in Figure 7.4

Table 7.3. y+ range of logarithmic region according to experiment authors

Station	*Lower limit*	*Upper limit*
1	50	310
2	30	140
3	8~	70
4	7~	65
5	6~	65
6	5~	60
7	5~	60
8	4~	60
9	4~	60
10	3~	55
11	3~	35
12	3~	35
13	3~	20
14	2~	20
15	2~	20
16	2~	20
17	2~	20

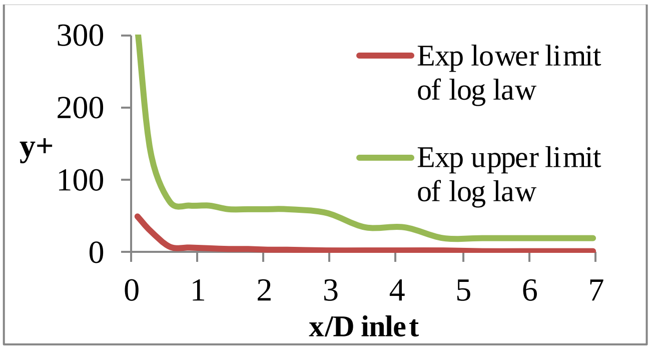

Figure 7.4. Experimental lower and upper limit of the range of the log law

The authors of the paper computed also the values of the slope of the logarithmic law of the wall, whose range of validity has just been discussed. The slope reported in the paper refers to the following logarithmic law of the wall:

$$u^{+} = A\log(y^{+}) + B \tag{7.1}$$

The values of A^{-1} have been computed and their values for every station of measurement are reported in Table 7.4.

Table 7.4. Inverse of the slope of the experimental log law of the wall for all the 17 stations of measurement

Station	A^{-1}
1	0,41
2	0,41
3	0,33
4	0,3
5	0,34
6	0,33
7	0,31
8	0,32
9	0,3
10	0,33
11	0,33
12	0,33
13	0,33
14	0,33
15	0,36
16	0,37
17	0,38

The inverse of the slope of the SWF log law is always constant and it is equal to the Von Karman constant. From an analysis of Table 7.4 it can be noticed that the values of the inverse of the slope of the experimental log law is not constant but it is dependent on the position inside the diffuser. This is at the same time a black mark for SWF and a plus point for GWF. Indeed, the use of SWF involves the use of a log law whose slope is always constant, and therefore the assumed law of the wall does not agree with the experimental one. On the other hand GWF have a slope of the log law which is not constant but it is sensitive to the non-equilibrium effects of the flow. Therefore the values of the slopes can change all along the diffuser. In order to make clearer the fact that experimental laws of the wall does not comply with SWF log law, Figure 7.5 shows in the same chart the dimensionless velocity profiles for Stations 9 and 15 compared to a typical boundary layer equilibrium law of the wall (see for example Figure 1.1). This law of the wall is composed by a linear law for the sub-viscous layer and by the SWF log law for the fully turbulent region. The two laws intersect each other for a value of y^+ around 10.

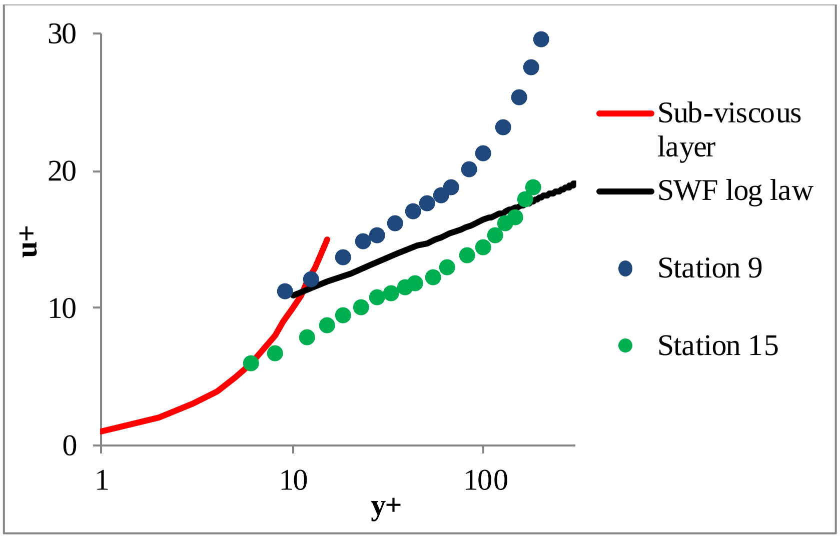

Figure 7.5. Comparison of experimental dimensionless velocity profiles for stations 9 and 15 with the equilibrium-boundary layer dimensionless velocity profile (linear law of the wall followed by SWF log law of the wall)

Figure 7.5 shows clearly that the dimensionless velocity profiles do not follow the SWF log law, and it shows also that where the experimental profiles can be fitted with a log law the slopes of them are different from the slope of SWF log law. Moreover the width of the experimental log laws is significantly lower than SWF log law width (note: the maximum value reported on the x-axis is equal to 300). It must also be noticed that the experimental dimensionless velocity profiles change significantly between the stations considered. It’s clear that we are dealing with a flow which differs from the equilibrium-boundary layer ones (e.g. pipe flow, flat plate flow) for which the use of SWF is strictly valid.

After these considerations, we expect that the use of SWF doesn’t provide results which are in accordance with the experimental data set just discussed. But poor results can be caused by other factors different from the wall treatment. We want to reduce to the minimum the negative influence of these other factors on the results, so that when we change the wall function we can appreciate more clearly the effects on the results of the only wall function. The factors which have been taken into account are:

- boundary layer mesh

- core mesh
- turbulence model

The logical approach adopted is here explained. First simulation has been carried out using SWF. The boundary layer mesh adopted ensures that the dimensionless wall distances of all near wall center cells fall into the range suggested by Fluent User Guide for the use of SWF. Indeed, if an unaware CFD user wants to carry out a simulation of this conical diffuser using SWF as wall treatment and he follows the suggestions of Fluent 6.3 User Guide, he will create a mesh where all the values of y^+ of near wall center cells are between 30 and 300. The core mesh adopted for this first simulation is a *medium* mesh (the characteristic length of the quadrilateral element is equal to one hundredth of the diameter of feeding pipe). The turbulence model adopted for the first simulation Standard k-epsilon. After this first simulation, two different mesh sensitivity analyses have been performed. The sensitivity analyses have been carried out by keeping on using SWF and Standard k-epsilon turbulence model. First sensitivity analysis regards boundary layer mesh. Different boundary layer meshes have been tested, and the one which provides the best results has been adopted for the further sensitivity analysis, which regards core mesh. A coarse mesh and a fine mesh have been created, respectively doubling and halving the characteristic length of quadrilateral elements. A check of the different results obtained has been made, and a grid convergence index (GCI) procedure has been performed. The core mesh that ensures the best results (the trade-off with number of cells and improvement of the results must also be taken into account) has been set at the reference core mesh. A turbulence model sensitivity analysis has been subsequently carried out, and the model which gives best results is taken for the next works, which regards the implementation of GWF. The mesh composed by boundary layer and core mesh obtained from the first two sensitivity analyses is adopted for the further simulations.

7.2 CFD Domain

Starting from the experimental facility data, the choice of a proper CFD domain has been made. First of all, since we are dealing with an axial symmetric conical diffuser and since the experimental data provided for every station by the authors are not dependent on the angular position of the cross-section, 2D axial symmetric simulations can be carried out. Therefore the geometry created is 2D and axial symmetric. The outlet of the domain has been placed at the same position of the outlet of the experimental facility (i.e. the outlet of the diffuser). Regarding the placement of inlet boundary, the authors say: “the diffuser feed is fully developed turbulent flow” [27]. This sentence is confirmed by the fact that the ratio between length and diameter of feeding pipe is about 74, values greater than developing length for a turbulent pipe flow. An empirical correlation for the developing length for turbulent flows [28]:

$$\left(\frac{L}{D}\right)_{\text{developing length}} = 4.4\, Re_D^{\frac{1}{6}} \tag{7.2}$$

gives an approximate value of 31 for this situation. Therefore the inlet of the geometry has been placed at one pipe diameter before the diffuser inlet. Even though the diffuser affects all the upstream flow (we are dealing with an incompressible - and therefore subsonic - flow), it is reasonable that the effects on the diffuser at the inlet position chosen are small, and thus negligible. With this choice, we avoid to include in the calculation domain the whole feeding pipe; we can therefore reduce the total number of cells, thus the computational time is reduced and also the convergence of the calculation is favored. Table 7.5 summarizes the dimensions of the CFD domain created. Just for completeness, the origin of coordinate system (x, y) has been reported in Table 7.5. The origin has been placed on the axis of the diffuser and on the inlet of the diffuser. This choice has been made in order to keep the same coordinate system used in the experiment and described in the paper.

Table 7.5. CFD domain data

Simulation	2D axisymmetric	
Inlet pipe length	0,1016	m
Pipe radius	0,0508	m
Diffuser outlet radius	0,1011	m
x coordinate system origin	Inlet of diffuser	
y coordinate system origin	Axis	

Regarding boundary conditions, the first ones that have been set are the axis and the wall (for which a non-slip condition has been defined). The inlet has been set as a velocity inlet type. The reason of this choice is that the diffuser feed is a fully developed turbulent pipe flow. Therefore we can easily obtain profiles of axial velocity, turbulent kinetic energy and turbulent dissipation rate from a 2D axial symmetric periodic pipe CFD simulation. Outlet has been set as pressure outlet. This choice can be explained by the fact that – as the authors report in the paper - the conical diffuser “discharged to the atmosphere”. It is therefore reasonable to assume that the pressure at the outlet of the diffuser – which is equal to atmospheric pressure - is constant.

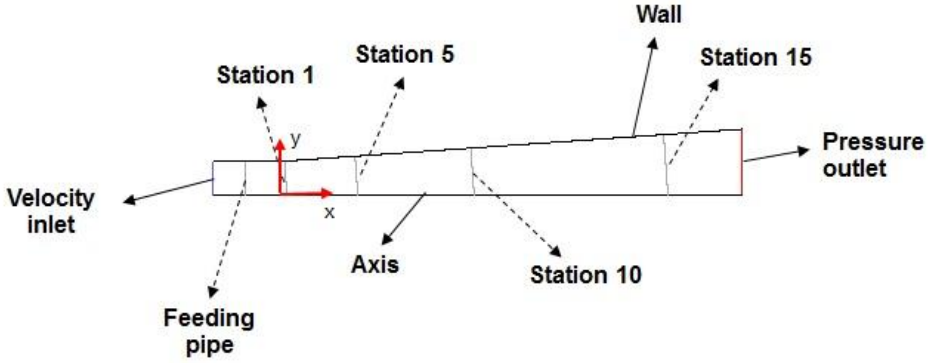

Figure 7.6. CFD domain and boundaries type; the positions of Station 1, 5, 10 and 15 is shown, like the position chosen for the comparison of experimental feeding pipe data

Table 7.6. Boundaries type: summary

Inlet	Velocity inlet
Outlet	Pressure outlet
Axis	Axis
Wall	Wall – non slip condition

Regarding the flow condition at the inlet of the diffuser, the authors of the experiment provide all the data required for the evaluation of all the parameters needed. Bulk velocity and Reynolds number at inlet are provided (Table 7.7). The value of cinematic viscosity can be obtained from an inversion of the definition of Reynolds number

$$Re = \frac{U_b D}{\upsilon} \tag{7.3}$$

$$\upsilon = \frac{U_b D}{Re} \tag{7.4}$$

Fluent 6.3 requires the user to set both the values of density and dynamic viscosity. It’s good to remember that in fluid dynamics only dimensionless quantities are important. Then, we can choose random values of density and dynamic viscosity, with the only constraint that their ratio must be equal to the value of cinematic viscosity that ensures Reynolds number to be equal to the experimental one. So, the value of density has been set equal to the default value used by Fluent 6.3 for the air (1.225 $\mathrm{kg/m^3}$). Dynamic viscosity is obtained from the inversion of the definition of cinematic viscosity.

$$\upsilon = \frac{\mu}{\rho} \tag{7.5}$$

The value of dynamic viscosity obtained, along with the value of density chosen, can be put inside *Fluent 6.3 Define → Material* panel. Flow conditions and properties are reported in Table 7.7.

Table 7.7. Flow conditions at the inlet of diffuser: summary

Inlet	Fully developed pipe	
Bulk velocity at inlet	18,06	m/s
Reynolds number at inlet	115000	
Cinematic viscosity	1,5956E-05	m^2/s
Density	1,225	kg/m^3
Dynamic viscosity	1,9546E-05	Pa s
Mass flow rate	0,1794	kg/s

The computation of mass flow rate (which value is reported in Table 7.7) comes from the expression (7.6).

$$\dot{m} = \frac{\rho U_{b,\text{inlet}} \pi D_{\text{inlet}}{}^{2}}{4} \tag{7.6}$$

Its value is important because it’s the constraint which must be used for the periodic pipe simulation, which allows to obtain the profiles that are adopted for the velocity inlet boundary.

7.3 Obtainment of Inlet Boundary Conditions

Regarding the evaluation of inlet boundary conditions, we are going to carry out a periodic 2D axial symmetric simulation of the feeding pipe. The profiles of axial velocity, turbulent kinetic energy and turbulent dissipation rate for one cross section of the pipe will be extracted and used as inlet conditions for the diffuser case. The domain adopted is shown in Figure 7.7. We must point out that: the boundary type of the two vertical edges is set periodic; the bottom edge is set as an axis type; the upper edge is set as a wall type with non-slip condition. The length of the domain is set equal to one pipe diameter.

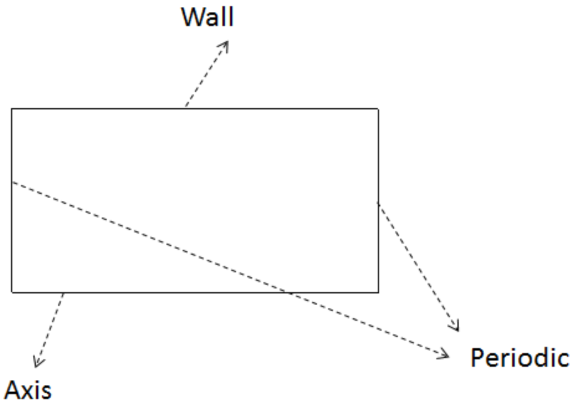

Figure 7.7. CFD geometry adopted for the 2D axial symmetric periodic pipe simulation

We want to realize a periodic simulation of this pipe with an imposed mass flow rate (whose value computed using equation (7.6) is reported in Table 7.7). SWF for wall treatment has been used. A check of the goodness of the results obtained has been made in terms of friction velocity (which is known from experimental data) and in terms of dimensionless velocity profile (which is known from experimental data too). This periodic simulation is carried out using a double precision solver and using second order discretization schemes for all the advective terms of the equations the code is going to solve (pressure, momentum, turbulent kinetic energy and turbulent dissipation rate). The boundary layer mesh has been created in order to have the value of y^+ for near wall center cells equal to 50 (so that the first computational node falls into logarithmic region; consequently SWF are suitable for the wall treatment). Table 7.8 shows the parameters used in the step by step process for the determination of boundary layer mesh thickness.

Table 7.8. Parameters used for the step-by-step process for the determination of boundary layer thickness; parameters of core mesh

Experimental friction velocity	0,83	m/s
1st cell y+	50	
1st cell y	9,61E-04	m
1st layer y	1,92E-03	m
Core mesh characteristic length	0,001016	m
Number of cells	4900	
Number of elements in radial direction	49	

The step by step procedure for the evaluation of the thickness of boundary layer mesh is here briefly described.

- Starting points are: experimental friction velocity obtained inside feeding pipe; constraint for y^+ of near wall center cells (which is set equal to 50)
- From these values we can compute the value of dimensional wall distance of the near wall center cell by inverting the definition of y^+ (equation (1.8)).
- Boundary layer mesh thickness is equal to the double of the wall distance of near wall center cell computed at the previous point.

Characteristic length of quadrilateral element of core mesh has been taken equal to one hundredth of feeding pipe diameter. The mesh created is shown in Figure 7.8.

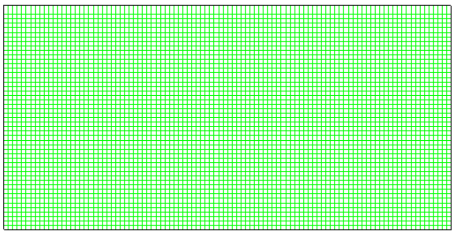

Figure 7.8. Mesh adopted for 2D axial symmetric periodic pipe simulation

The results obtained from the simulation along with the check of their goodness are here described. The value of friction velocity from the simulation has been compared with the experimental value. Note: the numerical value of wall shear stress present inside the definition of friction velocity (equation (1.9)) has been obtained from an area-weighted average of the wall shear stresses on the entire wall.

Table 7.9. Results from 2D axial symmetric periodic pipe simulation: averaged wall shear stress and friction velocity

	SWF	Experimental	Difference %
Wall shear stress [Pa]	0,8567	-	-
Friction velocity [m/s]	**0,836**	**0,83**	**0,75**
y^+	50,4	-	-

We can see that the relative percentage difference (equation (7.7)) between the experimental value of friction velocity and the value obtained from the CFD simulation is very small (lower than 1%).

$$\% \text{ difference} = \frac{\left| u_{\tau\,\mathrm{CFD}} - u_{\tau\,\exp} \right|}{u_{\tau\,\exp}} * 100 \tag{7.7}$$

We can conclude that the value of friction velocity obtained with 2D axial symmetric periodic pipe simulation can be considered correct. The second check of the goodness of the results is made using dimensionless velocity profile. As we can see from Figure 7.9, there is an almost perfect accordance between experimental and CFD data. In the chart also the law of the wall used by SWF is reported (expression (7.8)).

$$u^{+} = \frac{\ln(9.793\, y^{+})}{0.4187} \tag{7.8}$$

As expected, there is good accordance between experimental data and data obtained from the CFD simulations (called *SWF Simulation* in the legend). Moreover there is a good accordance between the two data sets just mentioned and the SWF log law for values of y^{+} smaller than (about) 300. For greater values the difference starts to grow up. This trend is expectable since the pipe flow is a typical situation where the SWF log law is valid in a range of y^{+} that goes from 30 to 300. It is also important to notice that the point representing the first computational node (the point of the red line with the lower value of y^{+}) lies perfectly on the SWF log law.

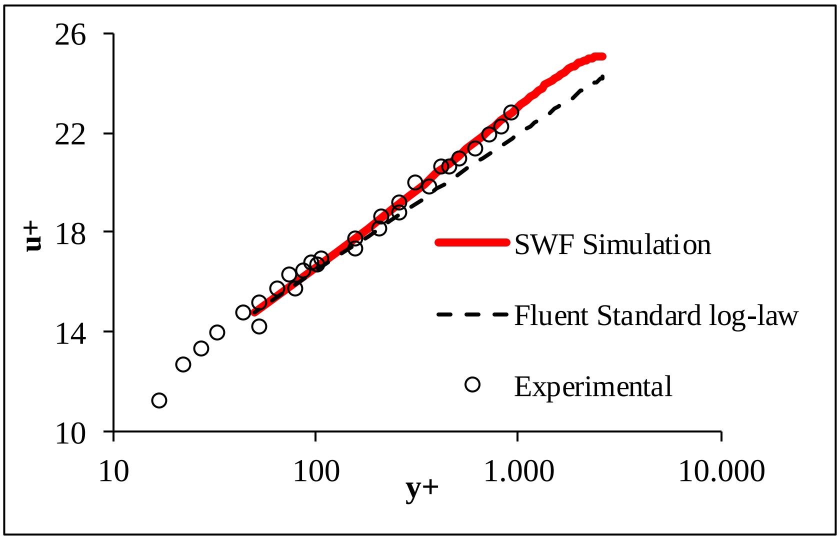

Figure 7.9. Feeding pipe: experimental dimensionless velocity profile compared to dimensionless velocity profile obtained with CFD simulation which uses SWF and compared to Fluent SWF log-law

These two checks just presented show that results obtained from periodic pipe simulation are very close to experimental results of feeding pipe. The CFD simulation can be considered acceptable; because of that, a sensitivity analysis concerning core mesh sensitivity is not mandatory. We can now extract the cross section profiles of the quantities needed for the inlet boundary condition of the diffuser. The quantities are: axial velocity, turbulent kinetic energy and turbulent dissipation rate. Figure 7.10 shows all the profiles extracted.

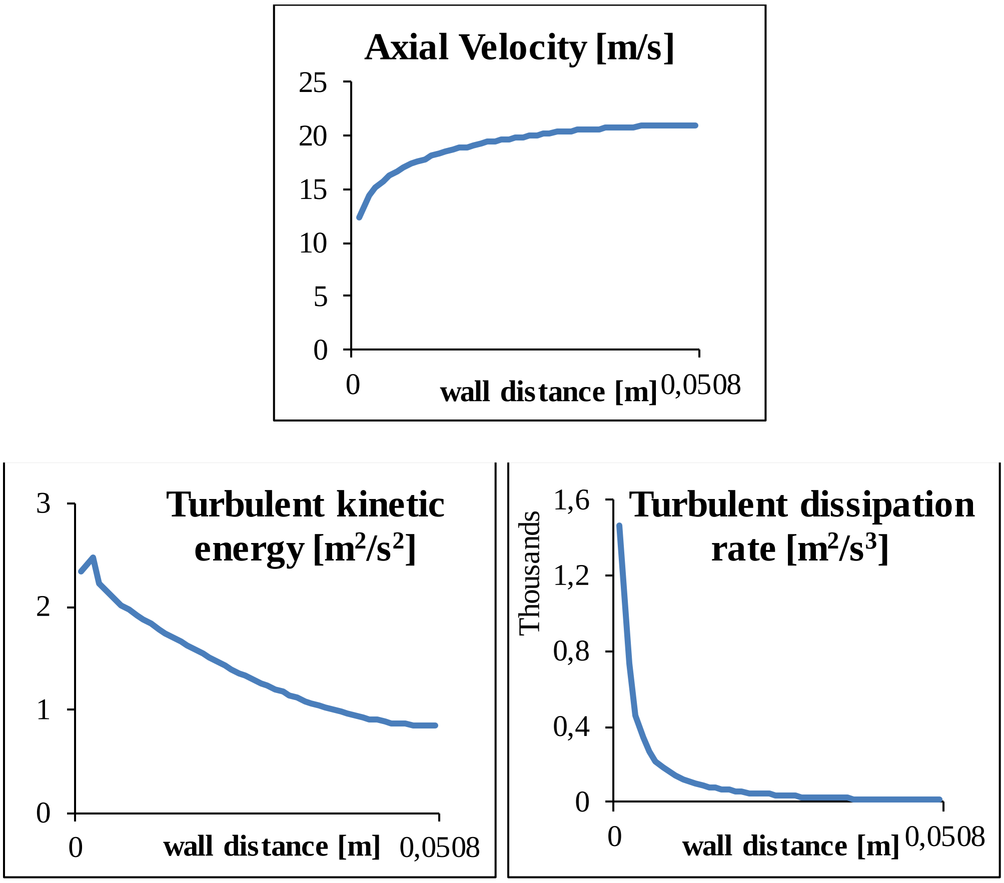

Figure 7.10. Profiles obtained from 2D axial symmetric periodic pipe: axial velocity, turbulent kinetic energy and turbulent dissipation rate

These profiles show all the characteristics expected for a turbulent flow. Mean axial velocity profile has a flat central region with strong gradients at the wall (it is not reported in the chart, but the velocity on the wall is null). Regarding turbulent quantities profiles, they show the characteristics of turbulent flows too. To be more precise: turbulent kinetic energy shows a peak close to the wall (in the region where there is the highest production of it) and then goes towards zero when approaching it (because of the kinematic and of the non-slip condition, which cause the turbulence to be null on the wall). On the other hand, approaching the axis the wall turbulent kinetic energy decreases and tends to centerline value. Turbulent dissipation rate instead doesn’t go to zero at the wall but it goes to a finite value; this behavior is typical of turbulent flows.

7.4 Creation of the First Mesh for the Conical Diffuser

Now that the inlet boundary conditions have been obtained, we have to define the mesh which will be used for the diffuser. First of all, it's necessary to discuss the mesh parameters used, which are presented in Table 7.10

Table 7.10. Parameters of the first mesh for the conical diffuser

Friction velocity inlet	0,83	m/s
Friction velocity outlet	0,164	m/s
1st cell y+ outlet	50	
1st cell y outlet	4,86E-03	m
1st layer y outlet	9,73E-03	m
1st cell y+ inlet	2,5305E+02	
Core mesh charact length	0,001016	m
Number of cells	73801	

Since this work deals with wall treatment, boundary layer mesh is very important and must be created carefully. As already stated, the first simulation is carried out using a boundary layer mesh that guarantees that the values of y^+ for all the near wall center cells are between 30 and 300 (which is the range of validity of SWF according to Fluent 6.3 User Manual).The logical path followed for the creation of the boundary layer mesh is the following:

- constraint used → all the values of y^+ for the near wall center cells must be between 30 and 300;
- the friction velocity at outlet (properly it is the value of friction velocity measured at Station 20, which is placed 1 cm before the outlet) is known from experimental data
- imposing a value of y^+ of 50 at the outlet (in order to be in the logarithmic region according to Fluent user guide) we can evaluate the thickness of boundary layer mesh from the inversion of the definition of y^+
- the same thickness of boundary layer mesh is used for the entire wall. It is necessary to check a priori if the value of y^+ at inlet is lower than 300 (friction velocity inside the pipe is given by experiment authors, therefore the value of y^+ can be easily computed)

Regarding core mesh, a size function of Gambit where the characteristic length of quadrilateral elements is constant and equal to one hundredth of the inlet diameter has been used. The mesh obtained has 73801 elements, and it is shown in the Figure 7.11.

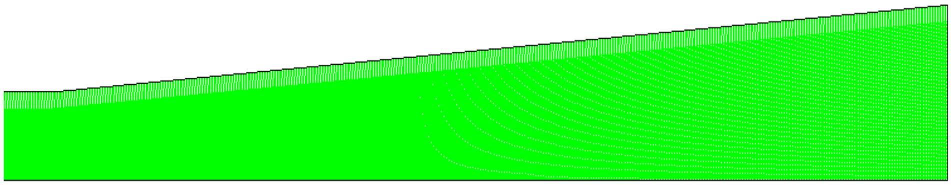

Figure 7.11. First mesh created for the conical diffuser

7.5 Results Obtained with the First Mesh Created and with SWF

With the mesh created and with the inlet boundary profiles obtained in paragraph 7.3, a simulation using Standard k-epsilon turbulence model and using SWF for wall treatment has been launched. Double precision solver and second order discretization scheme for all the advective terms have been adopted. The simulation has been called *Diffuser_mesh1*, and this is the name used in all the legends of the charts.

Table 7.11. Summary of the data of the simulation *Diffuser_mesh1*

Name of the case	Thickness of boundary layer mesh [m]	Characteristic length of quadrilateral element of core mesh [m]	Number of cells	Turbulence model	Wall Functions
Diffuser_mesh1	9,73E-03	0,001016	73801	k-epsilon standard	SWF

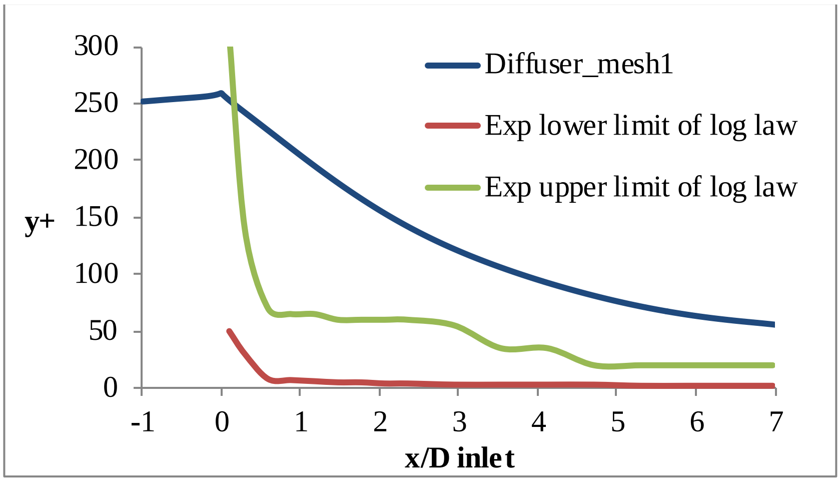

Figure 7.12. y+ values of the near wall cells obtained with the simulation performed with the first mesh created, standard k-ε turbulence model and SWF compared to the experimental range of validity of log law

As recommended by Fluent user guide, all the values of y^+obtained from the CFD simulation are between 30 and 300. However, as Figure 7.12 shows, the values of y^+ for near wall center cells obtained from CFD simulation are higher than the upper limit of the experimental range of validity of log law for the entire diffuser. We can expect that the results will show a bad accordance with experimental data because the law of the wall assumed for the near wall center cells is wrong. By the way, it is necessary to post process the results (qualitatively and quantitatively) in order to verify our observation.

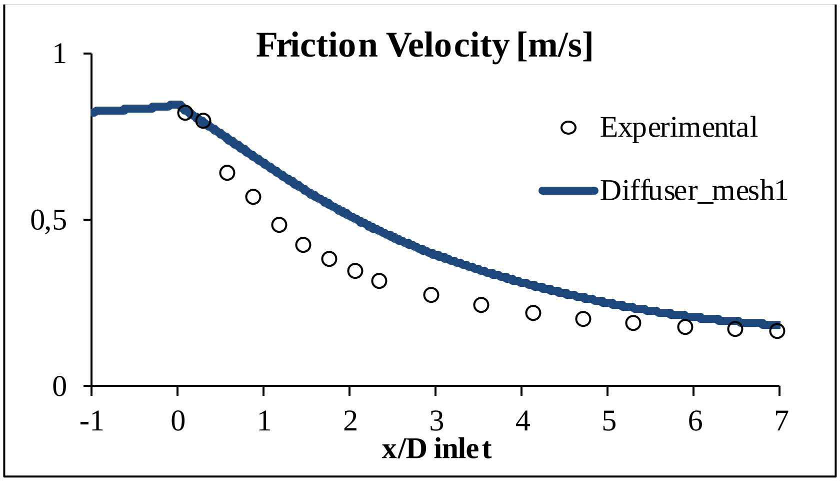

Figure 7.13. Friction velocity obtained with the simulation performed with the first mesh created, standard k-ε turbulence model and SWF compared to the experimental data

First physical quantity which has been compared with experimental data is friction velocity. This quantity is related to wall shear stress (equation (1.9)), which is a quantity directly affected by Wall Functions. Therefore friction velocity can be set as a target quantity which is useful for the evaluation of the goodness of the Wall Functions adopted. It is interesting to make some considerations about the expression (5.13), which is the expression adopted by the solver for the computation of wall shear stress. Neglecting the density, which is constant, wall shear stress is proportional to:

$$\tau_w \approx \frac{U_w u_k}{f(y^*)} \tag{7.9}$$

Figure 7.13 shows that the results obtained from the simulation *Diffuser_mesh1* overestimates wall shear stress. From expression (7.9), the overestimation, or (talking more generally) the error, can be due to these three factors (note: their effects can appear at the same time):

- U_w (wall tangential velocity at near wall center cells) is bad estimated;
- u_k, and consequently turbulent kinetic energy at near wall center cells, is bad estimated;
- the law of the wall assumed by the Wall Function adopted is wrong

Note: compensation effects cannot be excluded; for example the wall shear stress computed can be right because wall tangential velocity at near wall center cells is overestimated and turbulent kinetic energy is underestimated. Therefore we need to compare these three parameters with experimental correspondent quantities. Regarding wall tangential velocity at near wall center cell we have the experimental values for the comparison (indeed it is easy to pass from dimensionless velocity profiles to dimensional ones since experimental friction velocity for all the stations is provided). Regarding the law of the wall, we have the experimental range of logarithmic region, and we have the inverse of the slopes of the log laws. Note: the experimental log laws have been obtained by the fitting of the data in a chart (y^+, u^+), while the log laws adopted by the Wall Functions are in the form (y^*, u^*). Unluckily no experimental values for the turbulence are provided. However, since wall shear stress experimental values are provided (they can be easily obtained from the experimental friction velocity data), the lack of experimental information about the turbulence is not a big challenge. Indeed, the four parameters just discussed (law of the wall, wall tangential velocity, turbulent kinetic energy and wall shear stress) are related each other by the expression (5.13). The fact that the law of the wall assumed by SWF does not comply with the experimental one can be stated by looking at Figure 7.12. What we want to do now is to check the agreement between values of dimensional wall tangential velocity at near wall center cells from *Diffuser_mesh1* and the experimental values. Dimensional wall tangential velocity profiles are compared with experimental profiles for stations 1, 5, 10 and 15. Note: the near wall center cell value is the point closest to the wall; we must check only if it lies or not on experimental profile.

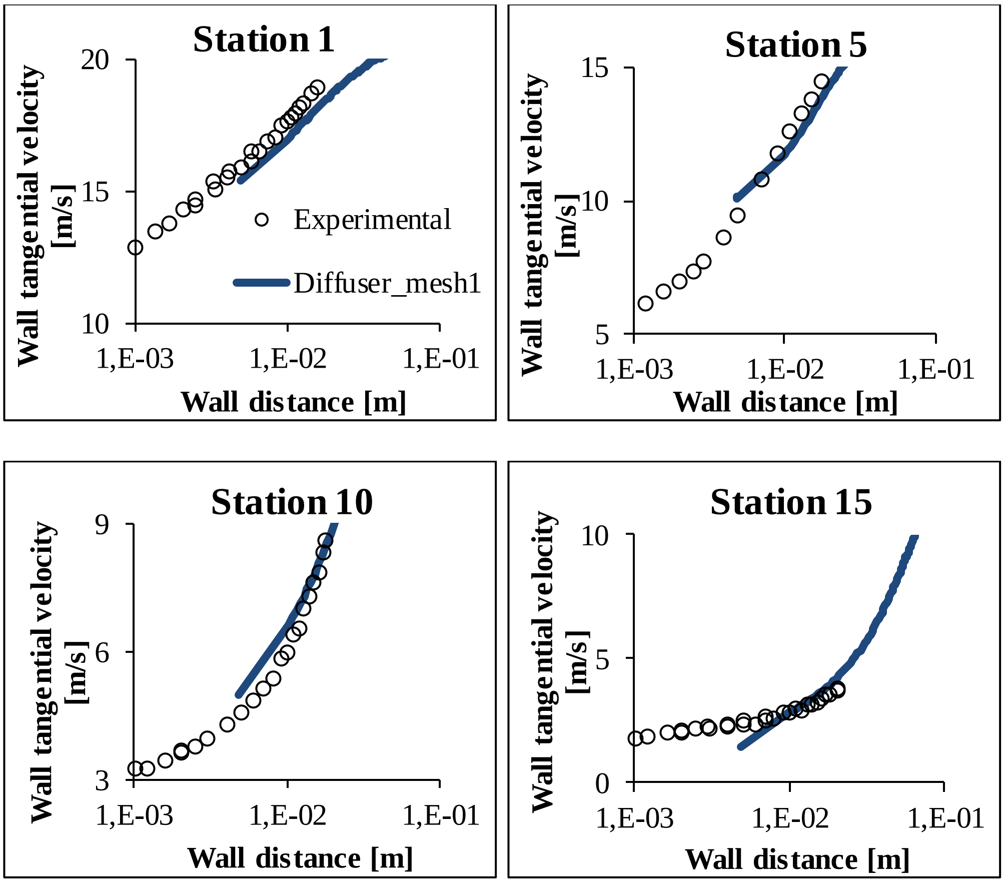

Figure 7.14. Dimensional wall tangential velocity profiles at Station 1, 5, 10 and 15: experimental data and *Diffuser_mesh1* results

We can notice that, for stations 5 and 10, wall tangential velocity at near wall cells is overestimated. For station 1 this quantity is slightly underestimated while for station 15 the underestimation is bigger. In the first part of the diffuser, since the law of the wall is still similar to SWF log law, the correct estimation of wall shear stress is explained. Since wall tangential dimensional velocity, wall shear stress and the law of the wall are in good accordance with experimental data, turbulent kinetic energy at near wall center cells can be considered correct. Moving more inside the diffuser, the other stations show discordance between experimental wall shear stress and the same quantity from *Diffuser_mesh1*. One reason is the overestimation of dimensional wall tangential velocity; moreover the flow doesn't comply with SWF log law. In the final part of the diffuser wall shear stress become closer to experimental data. Since wall tangential velocity is underestimated (see station 15 in Figure 7.14) we can guess that there are compensation effects between the quantities of expression (5.13).

It is interesting to compare all the experimental quantities measured with the correspondent quantities from *Diffuser_mesh1*. The first quantity which is interesting to compare is axial velocity on the axis of the diffuser.

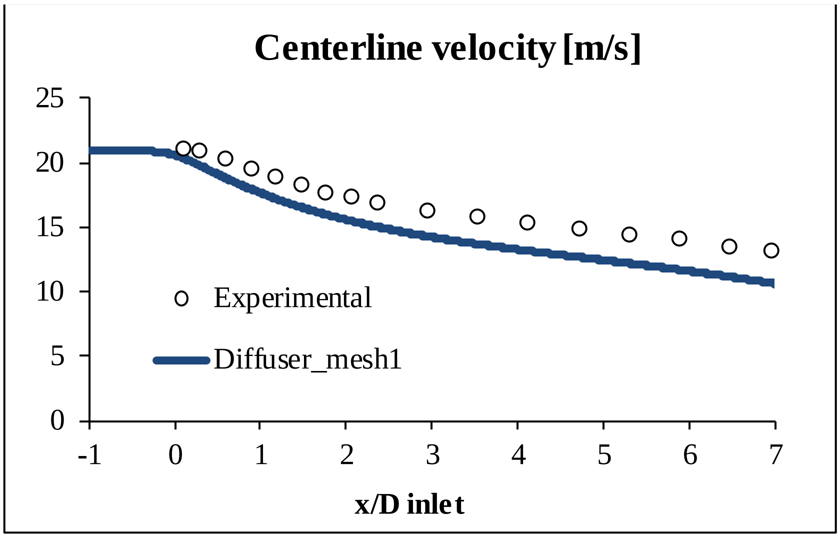

Figure 7.15. Centerline velocity comparison between *Diffuser_mesh1* and experimental data

The dimensional data sets show a big discordance between each other; the discordance grows up with the axial coordinate x. *Diffuser_mesh1* provides lower values of axial velocity on the axis of the diffuser. We are imposing an axial velocity profile at inlet which ensures that the CFD mass flow rate inside the diffuser is equal to experimental one. The analysis of these data sets shows that *Diffuser_mesh1* should provide a flatter velocity profile. Indeed, the integral of axial velocity profile defined by equation (7.10) provides the value of volumetric flow rate per radiant.

$$\int_{\text{wall}}^{\text{axis}} U(y)dy = \frac{\dot{m}}{\rho 2\pi} \tag{7.10}$$

Since this value is equal for the experiment and for CFD simulation, and since the velocity is null at the wall, if the centerline value of axial velocity for CFD simulation is lower than experimental one, there must be a region where the CFD profile is greater than experimental one. This means that in the first part of the diffuser the two profiles should show a good accordance, but the more we enter into the diffuser the more the dimensional CFD profile differs from experimental one. However, the discordance can be caused by different parameters, and the next sensitivity analyses will show how a single parameter affects this quantity. It is interesting to check the effect of the Wall Functions adopted on this quantity, in order to check how the Wall Functions influence the results in the entire domain.

Another quantity, which has been measured by experiment authors, we are going to compare to results from *Diffuser_mesh1* is dimensionless static pressure at the wall. The values are dimensionless because these are divided by the kinetic energy evaluated with inlet bulk velocity.

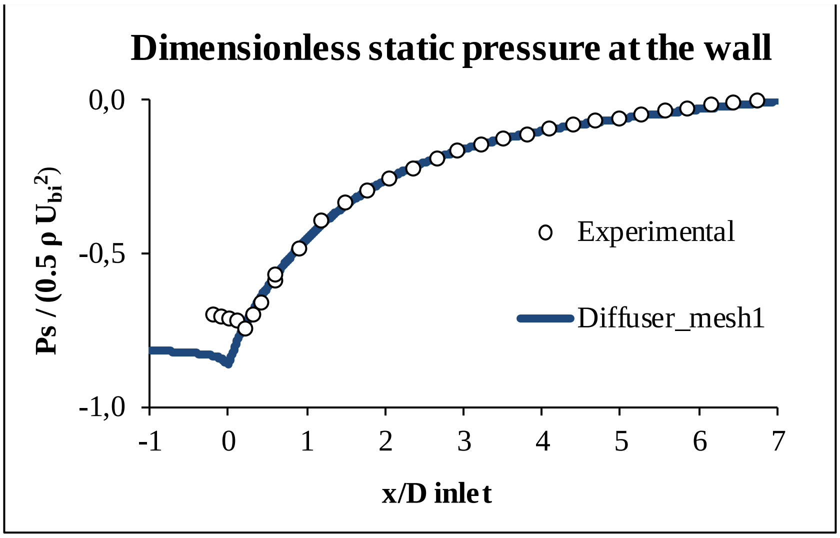

Figure 7.16. Dimensionless static pressure at the wall: comparison between *Diffuser_mesh1* and experimental data

First thing to point out is that all the pressure values are taken with respect to outlet pressure, which is the reference value and has a default value of zero. That's the reason of negative values. Since the turbulent kinetic energy evaluated with inlet bulk velocity is constant for both experimental data and *Diffuser_mesh1*, it's like if the comparison is made between dimensional static wall pressure values. Inside the pipe both the data set show a decreasing of pressure due to pressure drop. Inside the diffuser, on the other hand, the pressure drop is contrasted by pressure recovery (due to the increasing of the areas), which is prevalent. As we can notice from Figure 7.16, the accordance between experiment and CFD inside the diffuser is very good, i.e. the values of wall pressure obtained from *Diffuser_mesh1* are equal to experimental one. In the first part of the diffuser, *Diffuser_mesh1* over predicts pressure recovery, i.e. the pressure at the diffuser inlet obtained with *Diffuser_mesh1* is lower than experimental value. Experimental pressure values are provided also in two different dimensionless forms, i.e.

$$\alpha = \text{kinematic (axial static) pressure gradient} = \frac{1}{\rho}\frac{\partial P}{\partial x} \tag{7.11}$$

$$\Delta = \text{dimensionless pressure gradient parameter} = \frac{\alpha \upsilon}{u_\tau^3} \tag{7.12}$$

Δ can be considered a more meaningful quantity because its definition contains friction velocity, while α have the same characteristics of static pressure at the wall (it is obtained only by deriving axial static pressure with respect to *x* direction.

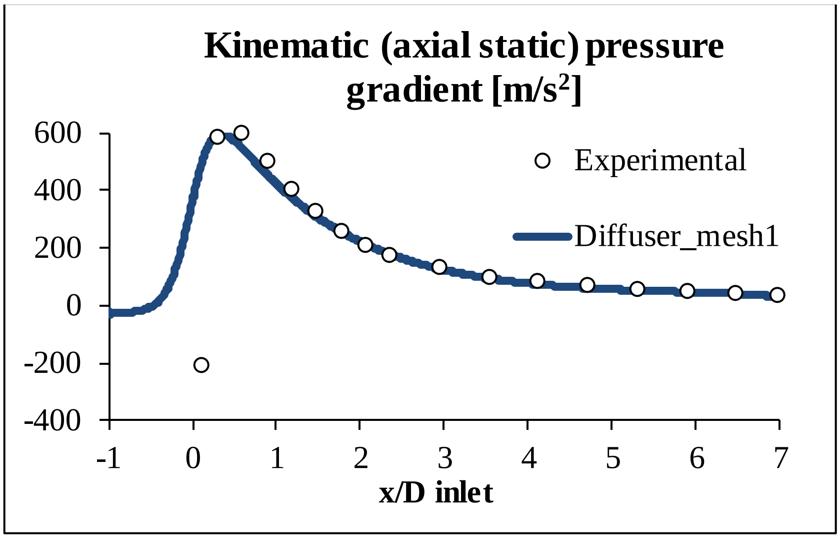

Figure 7.17. Kinematic (axial static) pressure gradient: comparison between *Diffuser_mesh1* and experimental data

As expected, an analysis of Figure 7.17 shows that the two sets of data are in good accordance. We can notice that *Diffuser_mesh1* does not provide a negative derivative of pressure after the inlet of the diffuser.

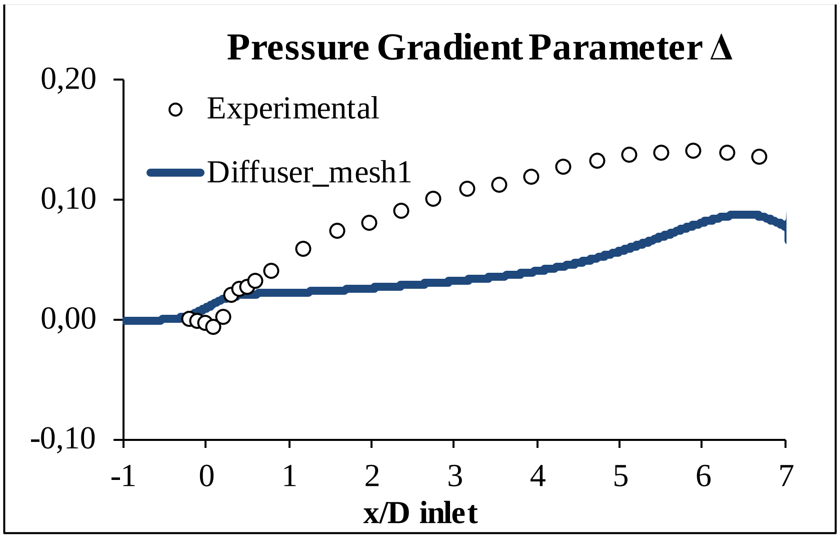

Figure 7.18. Pressure gradient parameter: comparison between *Diffuser_mesh1* and experimental data

The differences arise if we look at the Figure 7.18. Here the profile obtained with *Diffuser_mesh1* is completely different from the experimental one. The difference is not only related to the quantitative values (underestimation of it) but it is also related to the concavity of the two curves. The underestimation can be explained by the fact that friction velocity (which is at the denominator of Δ) is overestimated.

Last quantities to compare are the dimensionless velocity profile. As did for dimensional wall tangential velocity, instead of comparing all the 17 stations, only 5 stations have been chosen for the comparison. These stations are:

- Feeding pipe: station placed half a diameter before the inlet of diffuser; however, there is no need to present results of this station, since the profile here is be equal to the one given at the inlet, which has already proven to be correct (paragraph 7.3)
- Station 1: the first station inside the diffuser; it is useful because we want to analyze the first effects that the diffuser causes to the flow
- Station 5: where, for the authors of the experiment, the experimental departure from the standard law of the wall becomes significant
- Station 10: it is placed almost in the middle of the diffuser
- Station 15: it is close to the diffuser outlet; it is the region where a significant logarithmic region reappears and where the boundary layer becomes similar to an equilibrium type again

The dimensionless wall tangential velocity profiles for these stations are here presented.

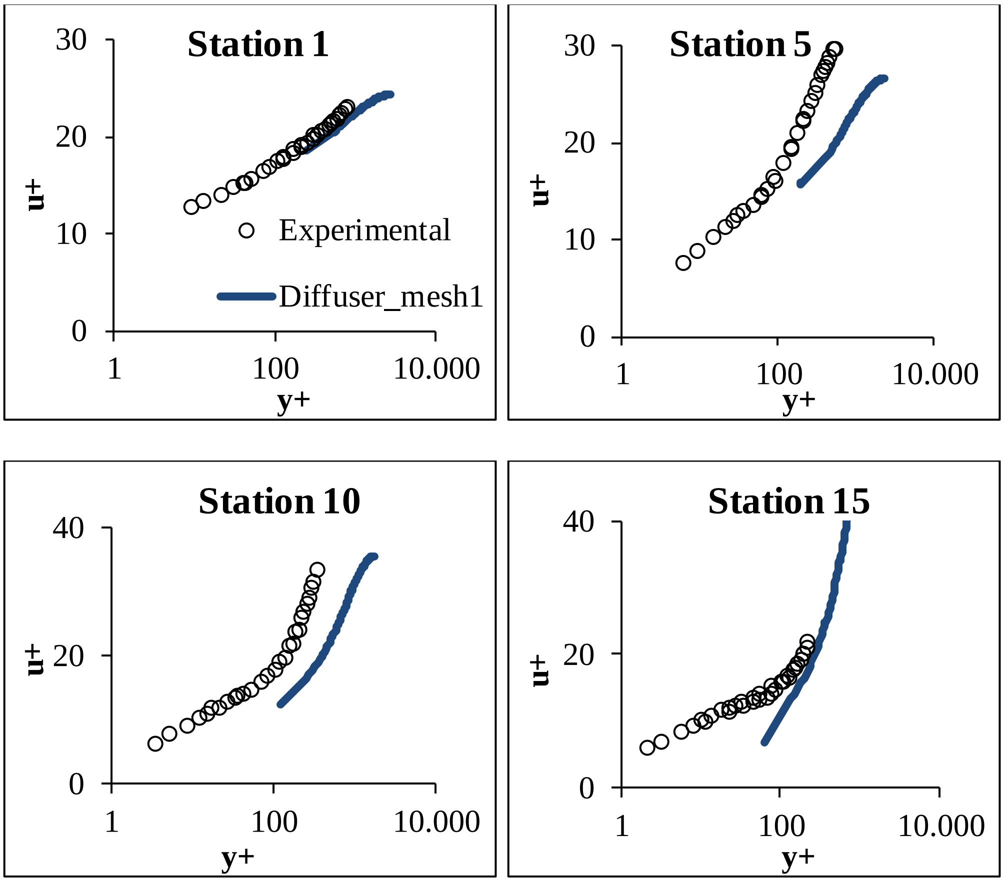

Figure 7.19. Dimensionless velocity profiles at stations 1, 5, 10 and 15: comparison between *Diffuser_mesh1* and experimental data

Analyzing the plots of Figure 7.19, it can easily be noticed that, starting from station 1 the dimensionless velocity profiles obtained from *Diffuser_mesh1* differ from experimental ones, and that the difference increases for station 5 and 10. This must not be surprising because we have already discussed the difference between experimental wall shear stress and CFD one. An important thing to notice is that the point closer to the wall doesn't lie on the experimental profile for all the stations presented (except station 1). This is another proof that near wall flow behavior is badly estimated by *Diffuser_mesh1*.

We can conclude that the results obtained with the mesh presented (boundary layer and core mesh), with standard k-epsilon turbulence model and with SWF are not satisfactory. But we still don't know the weight of the Wall Functions adopted on the bad quality of results. For example SWF can provide good results, but the results in this simulation are wrong also because the boundary layer mesh is not proper, or because core mesh is not thick enough or because the turbulence model adopted is not suitable to this kind of flow. Since the purpose if this thesis is to evaluate if GWF can provide better results compared to SWF and to NEWF (substantially we want to evaluate the goodness of the Wall Functions approach itself), it is good to minimize the effect on the results of the parameters which are not part of the goal of the work.

7.6 Boundary Layer Mesh Sensitivity Analysis

The first parameter taken into account for a sensitivity analysis is the boundary layer mesh (to be more precise the parameter is the constant thickness of near wall cells). The experimental data set shows that the logarithmic region inside the diffuser exists but its width is smaller than the range proposed by Fluent 6.3 User Guide for SWF. The lower and upper limit of this region is provided by the authors and the values have been already discussed (see Figure 7.4). Since from the analysis of y^+chart for *Diffuser_mesh1* (Figure 7.12) we noticed that all the dimensionless values of wall distance are higher than the upper limit of the experimental logarithmic region range, we can try to reduce the thickness of boundary layer mesh in order to make the near wall center cells fall into experimental logarithmic range. After that we will observe how the results change. Two new meshes have been created, where the only parameter changed is the thickness of boundary layer: the first has a boundary layer thickness which is halved respect to the boundary layer of the mesh of *Diffuser_mesh1*, and the second has a boundary layer thickness equal to one quarter of the thickness of the boundary layer of *Diffuser_mesh1*. Table 7.12 summarizes the characteristics and the name of the two meshes created. Note: this sensitivity analysis has been carried out using standard k-ε turbulence model and SWF.

Table 7.12. Parameters of the other two meshes adopted

Name of simulation	*Diffuser_mesh2*		*Diffuser_mesh3*	
Thickness of boundary layer mesh	4,87E-03	m	2,43E-03	m
Core mesh characteristic length	0,001016	m	0,001016	m
Number of cells	77856		79478	

The simulations obtained with the new two meshes have been called respectively *Diffuser_mesh2* and *Diffuser_mesh3*. In all the legends of the charts this simulations have been referred with these names. The next three figures show a particular of the three different meshes. As we can see, core mesh is always the same, while the thickness of near wall cells change significantly.

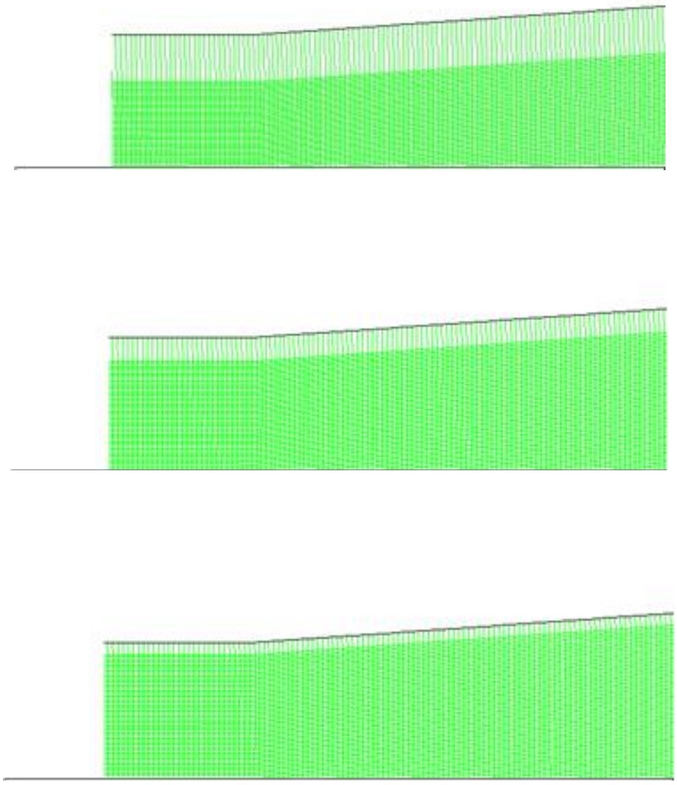

Figure 7.20. Different boundary layer meshes adopted: respectively *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3*

These simulations have been carried out with a double precision solver and adopting a second order discretization scheme for all the advective terms. The first quantity which is monitored is dimensionless wall distance y^+ for the near wall center cells. From an a priori analysis we expect that the values of y^+ for *Diffuser_mesh2* and *Diffuser_mesh3* will be respectively one half and one fourth of the values obtained from the first mesh. This will be rigorously true only if friction velocity remains equal for all the simulations. But, if friction velocity remains constant the effect of thickness of boundary layer mesh on the results will be null. We don't expect that the influence of this parameter is null. However we can guess the effect of boundary layer mesh thickness on wall shear stress values. Since we are trying to make the near center cells fall into experimental logarithmic region we expect a better agreement with experimental data for *Diffuser_mesh2* and *Diffuser_mesh3*. We expect the values of wall shear stress obtained for *Diffuser_mesh2* and *Diffuser_mesh3* lower than the ones obtained with *Diffuser_mesh1* and at the same time we expect an improvement of the agreement with experimental data.

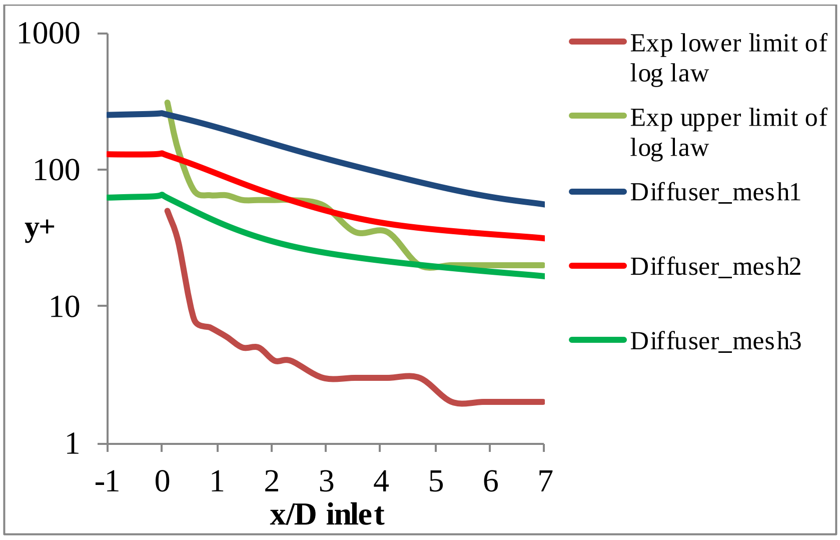

Figure 7.21.. y+ values of the near wall cells obtained with *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* compared with the lower and upper limit of experimental log law

Reducing the thickness of boundary layer mesh, as expected, the values of y^+ are lower. While with *Diffuser_mesh2* there are only two narrow regions where y^+ falls into experimental logarithmic range (i.e. in the first part of the diffuser and around 3 diameters of feeding pipe from the inlet), with *Diffuser_mesh3* all the near wall center cells fall into the experimental logarithmic region. We expect then to obtain better results with *Diffuser_mesh3* compared to *Diffuser_mesh2*. Moreover, for *Diffuser_mesh3* no cells have the center which falls into sub-viscous layer (note: dimensionless threshold value is 11.225). If some cells fell into sub-viscous layer, it would have been a problem because the law of the wall assumed would have been linear and not logarithmic. We would have set a new threshold value for the dimensional thickness of sub-viscous layer.

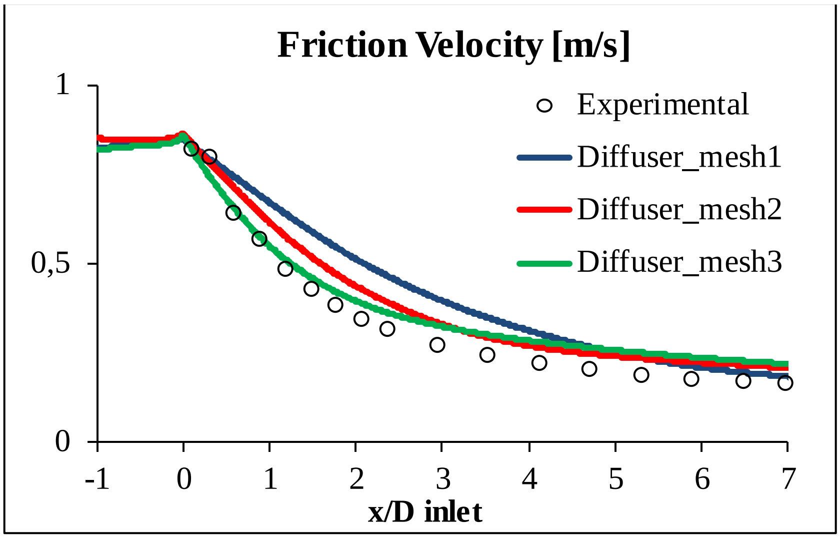

Figure 7.22. Friction velocity obtained with the simulation performed with *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* compared to the experimental values of friction velocity

As expected, *Diffuser_mesh2* and *Diffuser_mesh3* give a better agreement with experimental data. In the first part of the diffuser, a reduction of the thickness of near wall cells makes wall shear stress values lower and closer to experimental data. However, in the second region the effect is the opposite, i.e. a reduction of the thickness makes wall shear stress higher and consequently the accordance with experimental data is worse. The division point between the regions just discussed moves closer to the inlet when reducing the thickness of boundary layer mesh. For example between *Diffuser_mesh1* and *Diffuser_mesh2* the division point is placed around 5.5 feeding pipe diameters from the inlet of the diffuser while between *Diffuser_mesh2* and *Diffuser_mesh3* the point is placed around 3 feeding pipe diameters from the inlet of the diffuser. One thing must be pointed out, i.e. it seems that in the first region the gain in quality of the results is much higher than the loss in quality we notice in the second region. Last, moving from *Diffuser_mesh1* and *Diffuser_mesh3* the prediction of wall shear stress right after the inlet of the diffuser worsens (to be more precise the value obtained with *Diffuser_mesh2* is still quite correct while with *Diffuser_mesh3* the value of wall shear stress is underestimated. As did in paragraph 7.5, we must compare the quantities the computation of wall shear stress is related to (see equation (7.9)). First of all, dimensional wall tangential velocity profiles for stations 1, 5, 10 and 15 have been compared with experimental profiles. Second, velocity scale u_k obtained at all the near wall center cells for every case has been plotted in Figure 7.24.

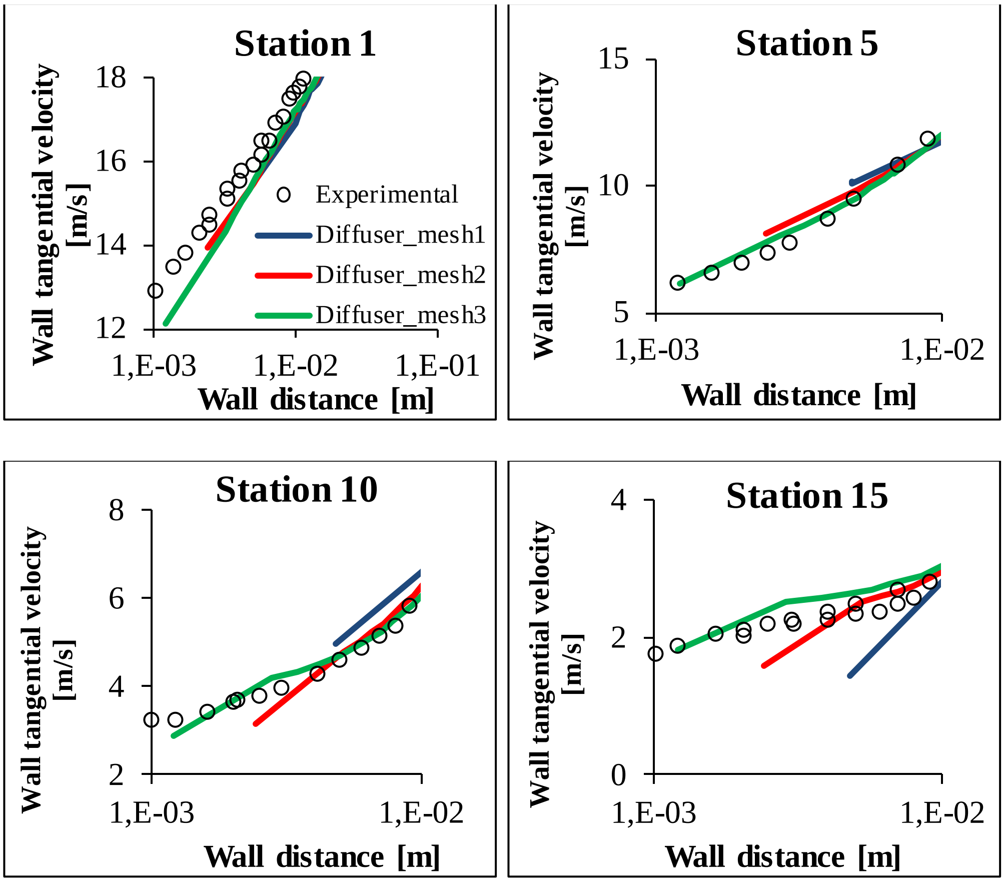

Figure 7.23. Dimensional wall tangential velocity profiles for Station 1, 5, 10, 15 for *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* compared with experimental data

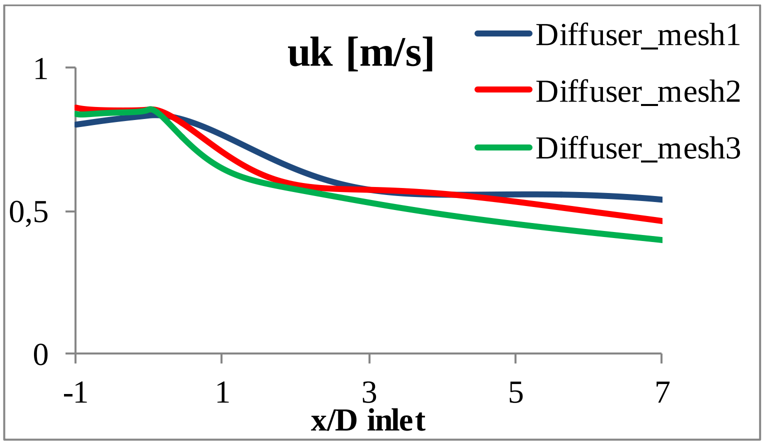

Figure 7.24. Velocity scale u_k obtained with *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3*

As we can see, the estimation of wall tangential velocity at near wall cells for *Diffuser_mesh3* is better compared to *Diffuser_mesh1* and *Diffuser_mesh2*, except for station 1, where *Diffuser_mesh1* have the best estimation. Regarding *Diffuser_mesh3* (and aside station 1), the accordance with experimental data is almost correct, except for station 10, where there is still a slight difference with the experimental datum. Therefore the errors of wall shear stress obtained for *Diffuser_mesh3* can be due to a wrong estimation of turbulent kinetic energy at near wall cells and to a wrong law of the wall assumed (see equation (7.9)). Regarding last cause, since with the boundary layer mesh of *Diffuser_mesh3* all the near wall center cells dimensionless wall distances are falling into the logarithmic region (Figure 7.21), possible errors related to the law of the wall assumed are related to a wrong slope of log law. In our situation the inverse of the slope of SWF log law is constant (Table 3.1) and it is higher than the inverse of the slopes of the experimental log laws inside the diffuser (Table 7.4). Therefore we expect to obtain higher values of wall shear stress, like it really happens (Figure 7.22). Regarding turbulent kinetic energy (or equivalently u_k) it decreases all along the wall when moving from *Diffuser_mesh1* to *Diffuser_mesh2 and Diffuser_mesh3*. The computation of wall shear stress is lower in the first part of the diffuser because there is both a decrease of the error in the computation of wall tangential velocity and there is a decrease of the velocity scale u_k. In the last part of the diffuser the higher values obtained with *Diffuser_mesh2* and *Diffuser_mesh3* can be explained by the fact that wall tangential velocity increases (Figure 7.23) from 1,43 m/s to 1,58 m/s to 1,83 m/s. Regarding station 1 the different trend for wall tangential velocity at near wall cell (the simulation with *Diffuser_mesh3* underestimates it) is the reason of the underestimation of wall shear stress with *Diffuser_mesh3*.

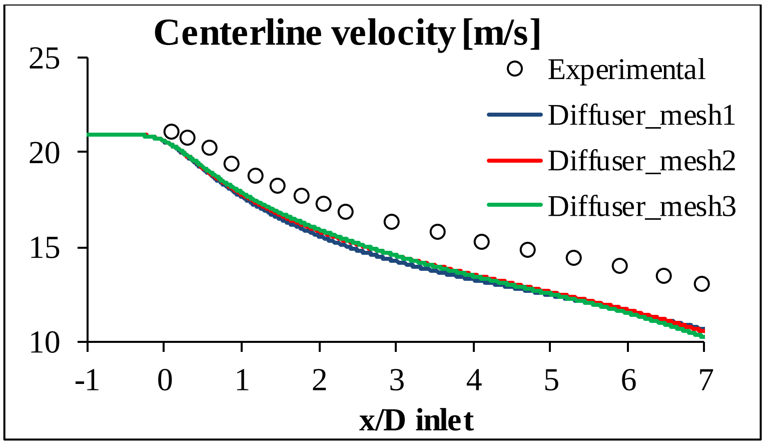

Figure 7.25. Centerline velocity: *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* and experimental data

Regarding centerline velocity, no remarkable changes can be noticed. Since the turbulence model adopted is not changed, we can conclude that the thickness of boundary layer mesh does not have any influence on the flow field on the axis of the diffuser.

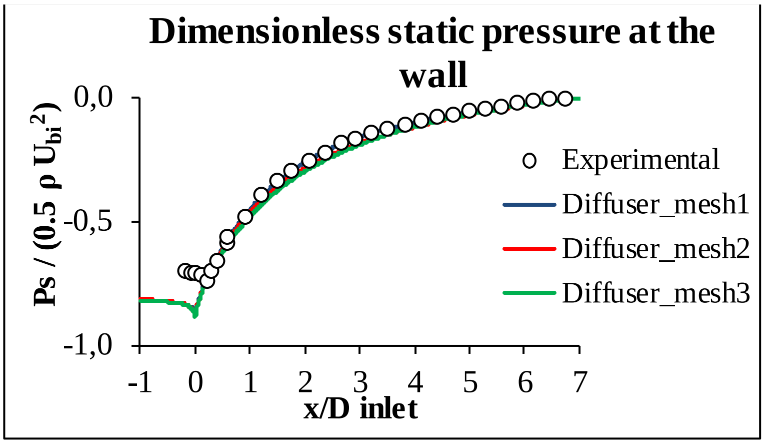

Figure 7.26. Dimensionless static pressure at the wall: *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* and experimental data

The dimensionless static wall pressure chart (Figure 7.26) does not allow to notice remarkable differences in the results too. We can make the same observation we made for centerline velocity, i.e. thickness of boundary layer mesh does not significantly affect the pressure field.

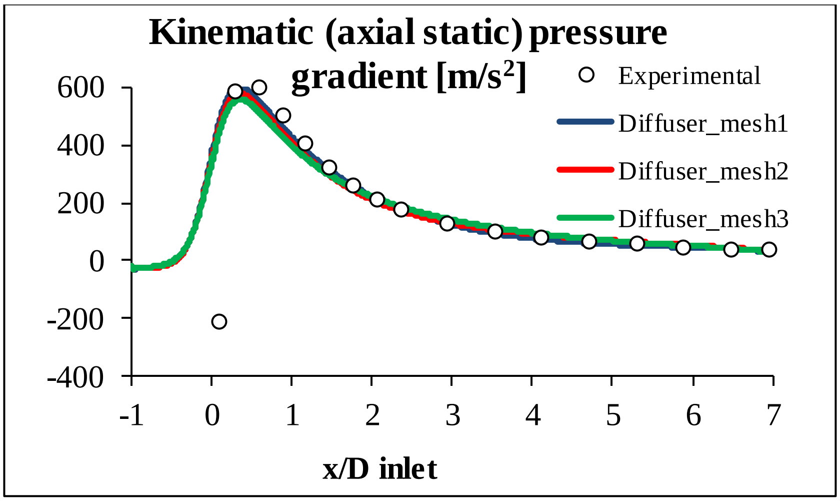

Figure 7.27. Kinematic (axial static) pressure gradient: *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* and experimental data

Also kinematic pressure gradient chart doesn't show differences in the results. As already stated this value is strongly related to the pressure (it is just the derivative of the pressure on the axis divided by the density, which is a constant). Therefore it's not surprisingly if there is not difference in the results, since there was no difference in the static wall pressure chart.

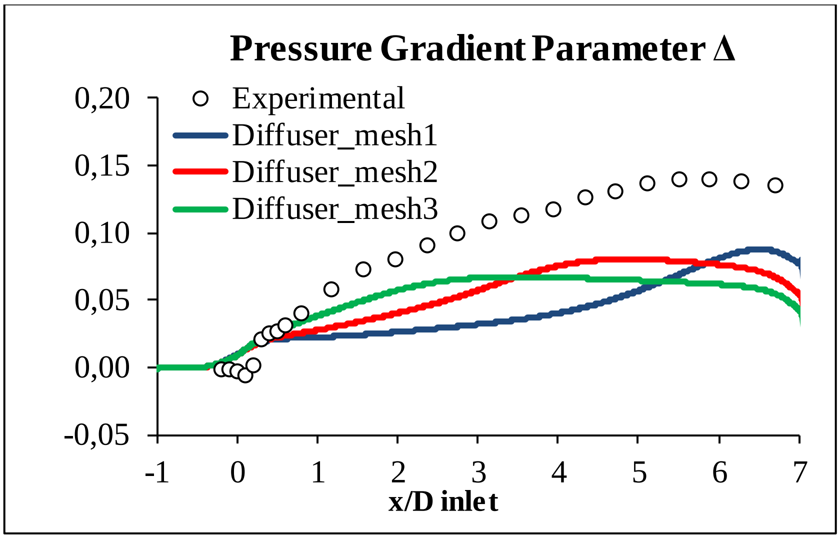

Figure 7.28. Pressure gradient parameter: *Diffuser_mesh1*, *Diffuser_mesh2* and *Diffuser_mesh3* and experimental data

Pressure gradient parameter, whose chart is represented in Figure 7.28, on the other hand, shows a big difference in the results. First thing we can say is that with *Diffuser_mesh2 and Diffuser_mesh3*, even if the accordance with experimental data is not perfect, the trend of the CFD data is similar to experimental one, i.e. the two profiles show a region where the second derivative is negative, like the experimental profile. The part with the positive concavity is confined to the first part of the diffuser; profile for *Diffuser_mesh2* has a wider region where the second derivative is positive (from about 0 to 4 feeding pipe diameters from the inlet of the diffuser) while for *Diffuser_mesh3* the second derivative is negative all along the wall.

We want to remember that the purposes of this sensitivity analysis are, on one hand to evaluate how the results changes with the thickness of boundary layer mesh, on the other hand to find the boundary layer mesh that gives the results closer to experimental data. Last purpose is pursued in order to minimize the effects of boundary layer mesh on the results. Since the boundary layer mesh that gives better results will be used for further sensitivity analysis and for the implementation of GWF, in order to make a good choice we must check also dimensionless velocity profiles.

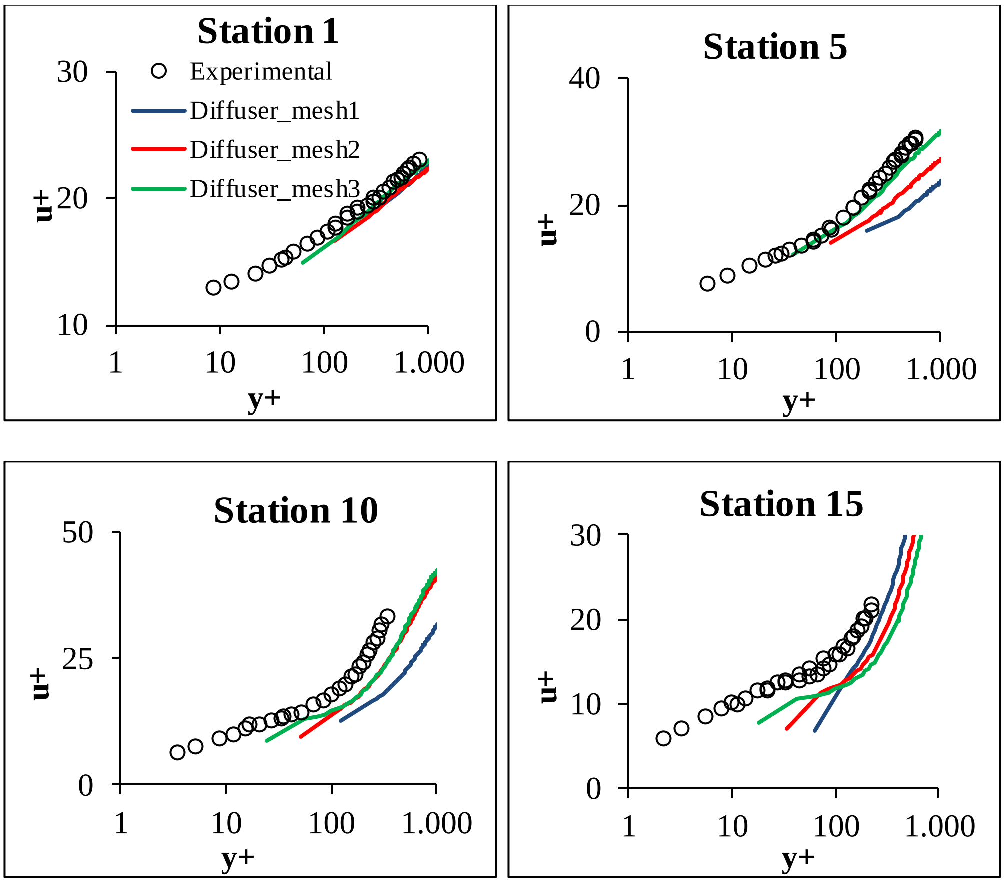

Figure 7.29. Dimensionless velocity profiles for Station 1, 5, 10 and 15: *Diffuser_mesh1, Diffuser_mesh2* and *Diffuser_mesh3* and experimental data

Looking at the dimensionless velocity profiles of Figure 7.29, we can get the following observations. For station 1 the results are very similar each other and there is a good agreement with the experimental profile. This is right because for station 1 SWF can be used with a negligible error. From station 5 the profiles become remarkably different. Profile for *Diffuser_mesh3* has a very good accordance with the experimental profile. For station 10, the difference between *Diffuser_mesh2* and *Diffuser_mesh3* is small, but both the profiles are closer to experimental one than *Diffuser_mesh1*. However, the near wall point of *Diffuser_mesh3* is the closest to the experimental profile. For station 15, even if profile with *Diffuser_mesh1* is the closest to experimental profile for y^+ higher than 100, the trends of the three profiles are not similar. *Diffuser_mesh1* has the first derivative higher than the first derivative of experimental profile. *Diffuser_mesh2* gives a worse accordance than *Diffuser_mesh1*, and *Diffuser_mesh3* gives a worse accordance than *Diffuser_mesh2* for y^+ greater than 100. However, the first and second derivatives of *Diffuser_mesh3* profile are in better accordance with experimental first and second derivative. Moreover, by looking at first computational node point, we can conclude that *Diffuser_mesh3* provides the better result.

The choice of the best thickness of boundary layer mesh has then fallen then on the one adopted by *Diffuser_mesh3*. The reasons at the base of the choice of this boundary layer mesh are here summarized:

- All the values of near wall center cell dimensionless wall distance fall into experimental logarithmic range (Figure 7.21)
- Near wall tangential velocity is correctly estimated, except for station 1 (see Figure 7.23)
- The chart of friction velocity for *Diffuser_mesh3* shows a good accordance with experimental data in the first part of the diffuser; in the second part the accordance is worse than *Diffuser_mesh1* and *Diffuser_mesh2*, but the difference of *Diffuser_mesh3* from experimental data is not so high. Moreover the trend of *Diffuser_mesh3* profile seems to be similar to experimental one (Figure 7.22). Since wall tangential velocity at near wall center cells is correctly estimated and since first computational nodes fall into the experimental logarithmic range, the error of wall shear stress is due to the slope of the law of the wall and to the turbulent kinetic energy (these quantities are directly related to the Wall Functions adopted)
- From the dimensionless velocity profiles analysis (Figure 7.29) we notice that for *Diffuser_mesh3* the profiles for feeding pipe, station 1 and station 5 are closer to experimental one. The last two stations considered show that the trend of *Diffuser_mesh3* profiles is similar to experimental ones; the value of the first computational node is the closest to experimental profile even if the rest of the profile is not the closest to the experimental one.

7.7 Core Mesh Sensitivity Analysis

Using the thickness of boundary layer mesh adopted by *Diffuser_mesh3*, a core mesh sensitivity analysis must be carried out in order to evaluate the influence of the core mesh on the results. The parameter taken into account for the analysis is the characteristic length of quadrilateral element of core mesh. Three different core meshes have been created. The first one is that one used in the previous three simulations and it is called *medium*. The other two are obtained respectively doubling and halving the parameter just discussed in order to obtain a *coarse* and a *fine* mesh. Simulations are carried out using the same conditions adopted for the previous simulations. To be more precise: turbulence model used is standard k-ε, SWF are adopted, the solver has been launched in double precision mode and a second order scheme of discretization for all the advective terms has been used.

Table 7.13. Parameters of the meshes coarse, medium and fine

	Thickness of boundary layer [m]	Core mesh characteristic length [m]	Nr of elements	Name of the simulation
Medium	2,43E-03	0,001016	79478	*Diffuser_mesh3*
Coarse	2,43E-03	0,002032	20250	*Diffuser_mesh3_coremesh2*
Fine	2,43E-03	0,000508	316095	*Diffuser_mesh3_coremesh3*

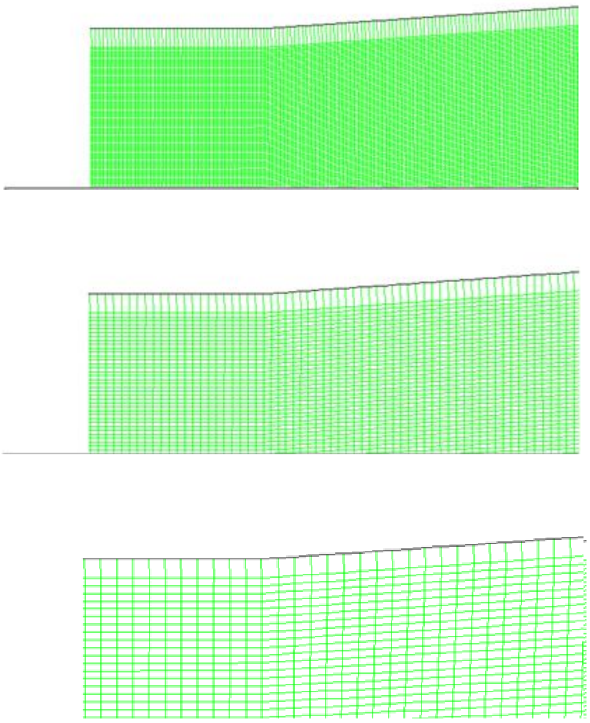

Figure 7.30. Particulars of (respectively) mesh fine, mesh medium, mesh coarse

When we'll choose the best core mesh we must take into account also the trade-off between the improved quality of the results and the increased number of cells (quantity related to computational time). Therefore, for example, if mesh fine gives results which are closer to experimental data than mesh medium, but the difference between the two is not so high, we must consider if the slight increase of the quality of the results justifies the increased number of cells (which is about four times higher), and consequently the increased computational time. We are going to compare now the results obtained with mesh coarse, medium and fine.

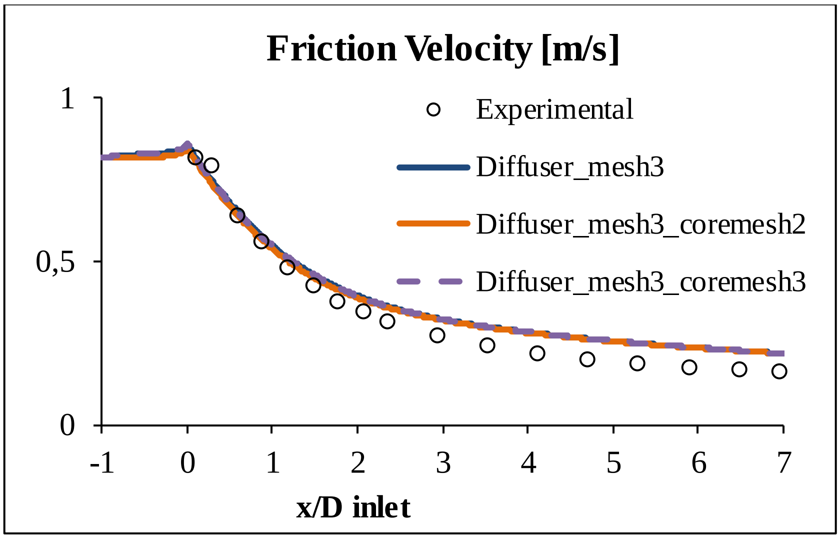

Figure 7.31. Friction velocity: core mesh sensitivity

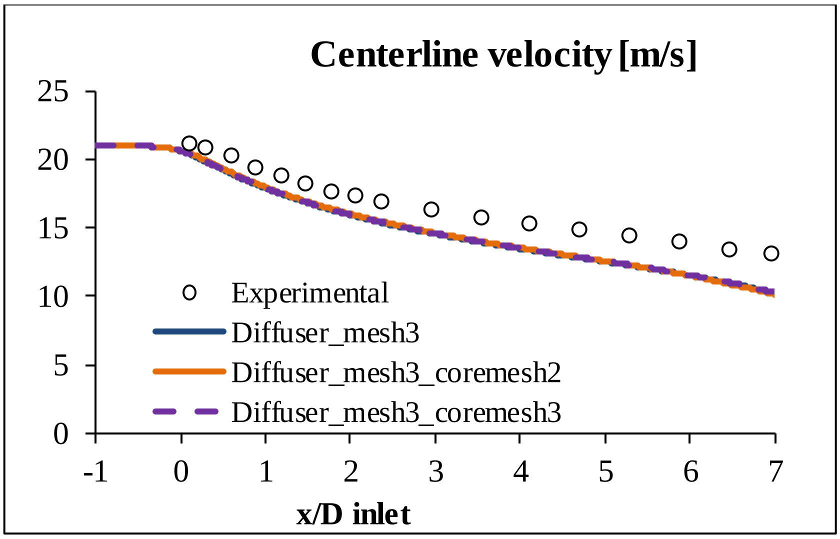

Figure 7.32. Centerline velocity: core mesh sensitivity

The plot of friction velocity values obtained with the three different core meshes does not show significant difference in the results (Figure 7.31). Therefore core mesh does not have a significant influence on wall shear stress. Before we can state that the influence on the whole solution of core mesh is negligible we must check also some other quantities, for example centerline velocity. The influence of core mesh has resulted to be null also for centerline velocity (Figure 7.32). Another quantity that can be checked is pressure gradient parameter.

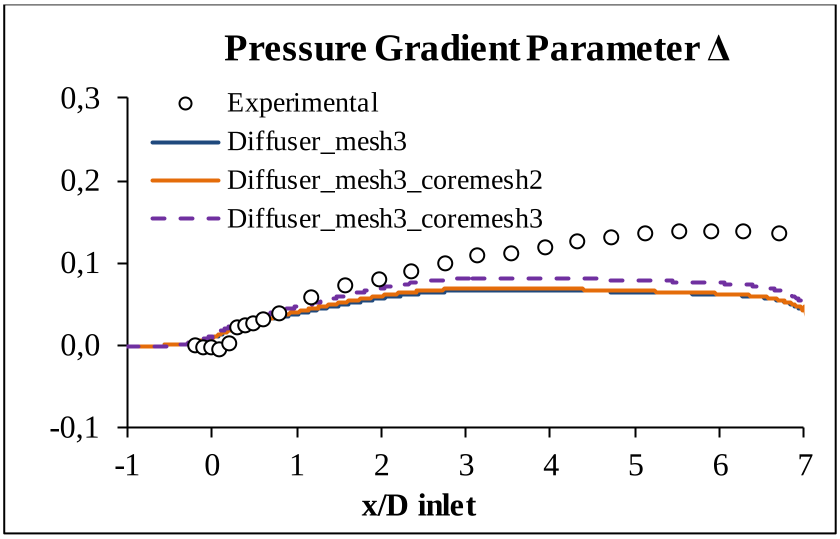

Figure 7.33. Pressure gradient parameter: core mesh sensitivity

The only significant difference in the results can be noticed for the quantity Δ (Figure 7.33), which is dimensionless. The chart of Δ shows that the mesh fine provides better results. But now we have to take into account the tradeoff between quality of the results and number of cells. Since there is no significant difference in the results obtained with the different core meshes, except for pressure gradient parameter, the choice of mesh fine will be not smart, because along a slight improvement of the results the computational time increases by four times. Therefore we are driven to choose the mesh medium for the further simulations.

7.7.1 GCI procedure

Using data obtained from these three core meshes it is possible to obtain an estimated value of the error due to spatial discretization of the domain, also known as Grid Convergence Index (GCI) [5], [6], [29]. This quantity is based on the concept that using three different meshes with a refinement ratio constant makes possible to extrapolate the solution for a mesh with an infinite number of cells. Therefore it is possible to find a number which represents the uncertainty of the results caused by the only grid. We want to evaluate the error caused by the grid on wall shear stress, which is one of the quantities set as the target for the evaluation of the goodness of the Wall Functions adopted. The quantity chosen for the evaluation of the GCI is the area weighted averaged wall shear stress. This value is computed starting from the following equation (which is the statement of the Mean Value Theorem):

$$\int \tau_w dA = \overline{\tau_w} \int dA = \overline{\tau_w} A \tag{7.13}$$

Therefore the value reported in the table and used for GCI is computed as

$$\overline{\tau_w} = \frac{\int \tau_w dA}{A} \tag{7.14}$$

Table 7.14. Values of averaged wall shear stress obtained with meshes coarse, medium and fine

Mesh	Averaged wall shear stress [Pa]	Number of elements
Coarse (3)	0,2165	20250
Medium (2)	0,2215	79478
Fine (1)	0,2220	316095

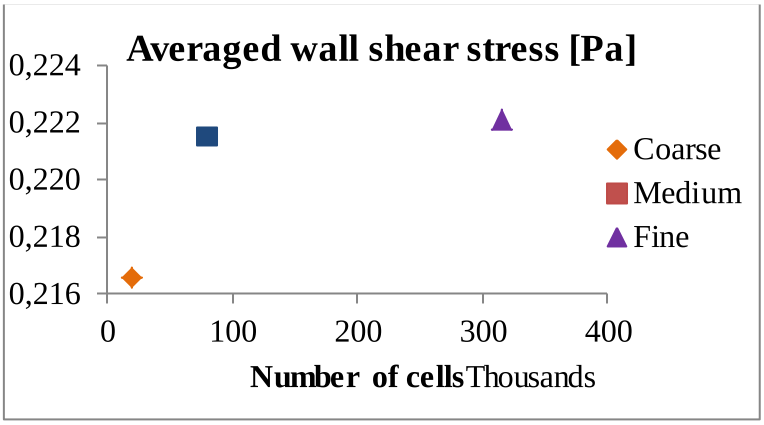

Figure 7.34. Chart presenting the values of area weighted wall shear stress along the wall obtained with meshes coarse, medium and fine

As we can notice from this figure, the trend of averaged wall shear stress tends to a horizontal asymptote when the number of cells of the mesh tends to an infinite number. This means that the number will converge to a finite number, even if we don’t know the real value of the asymptote (because we cannot create a grid with an infinite number of elements). We can just find an uncertainty range where the value obtained with a mesh with an infinite number of elements will fall into.

Table 7.15 reports the procedure for evaluation of Grid Convergence Index.

Table 7.15. GCI procedure

Procedure for GCI evaluation		
f3-f2	-0,0050	Pa
f2-f1	-0,0005	Pa
((f3-f2)/f2-f1))	9,876	
ln((f3-f2)/f2-f1))	2,29	
	GCI_{12}	GCI_{23}
r	1,981121	1,994276
p	3,349734	3,317616
ε	-0,00229	-0,02265
r^p-1	8,875812	8,875812
E1	-0,00026	-0,00255
FS	3	3
GCI	0,000774	0,007657
r^p GCI_{12}	0,007639	
f*	0,2221	[Pa]
f* lower limit	0,2222	[Pa]
f* upper limit [Pa]	0,2219	[Pa]

The procedure presented in Table 7.15 is here briefly described. The starting point is the so-called Richardson extrapolation, which states that the discrete solution can be a function of the exact solution (unknown) and the grid spacing h.

$$f = f_{\text{exact}} + g_1 h^1 + g_2 h^2 + \cdots + g_n h^n \qquad \textbf{(7.15)}$$

By using two grids with two different grid spacing $h1$ (finer) and $h2$ (coarser), where their ratio is called grid refinement r, it is possible to write two similar expressions similar to (7.15).

$$r = \frac{h_2}{h_1} \qquad \textbf{(7.16)}$$

Note: (7.16) is strictly valid only for a structured grid. For unstructured grid r can be computed as:

$$r = \left(\frac{N_1}{N_2}\right)^{\frac{1}{D}} \tag{7.17}$$

In the expression (7.17) N_1 and N_2 are the number of cells of the two grids, while D is the dimensionality of the problem.

$$f_1 = f_{\text{exact}} + g_p {h_1}^p + o({h_1}^{p+1}) \tag{7.18}$$

$$f_2 = f_{\text{exact}} + g_p (rh_1)^p + o((rh_1)^{p+1}) \tag{7.19}$$

It is possible to obtain a better estimation of the solution by using a linear combination of f_1 and f_2.

$$f^* = f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{7.20}$$

In order to evaluate the order of the method p, three grids are required (called respectively coarse, medium and fine).

$$p = \frac{\log(\frac{f_3 - f_2}{f_2 - f_1})}{\log(r)} \tag{7.21}$$

From this point, it is possible to evaluate the GCI obtained from grid 1 (fine) and grid 2 (medium), called GCI_{12}. The relative error between the two grid is defined as:

$$\varepsilon = \frac{|f_2 - f_1|}{f_1} \tag{7.22}$$

Then, the estimated fractional error called E_1 can be computed.

$$E_1 = \frac{\varepsilon}{r^p - 1} \tag{7.23}$$

So far, all the data needed for the computation of the GCI_{12}have been obtained.

$$GCI_{12} = FS\ E_1 \tag{7.24}$$

FS may be seen as a factor of safety over the Richardson extrapolation.

At this point, one constraint must be verified. If the constraint is verified, therefore all the assumptions behind the GCI procedure here exposed are correct. The constraint is:

$$GCI_{23} = r^p\ GCI_{12} \tag{7.25}$$

Where GCI_{23} is the value of GCI obtained from the grid 2 (medium) and 3 (coarse). Its computation follows the same procedure shown for GCI_{12}. As we can see from the Table 7.15, the constraint (7.25) is verified (the relative percentage error is around 0,23%).

We must point out that the result of the whole procedure is the GCI_{12} value. This value is useful to create an uncertainty bar for the fine solution, in the meaning of:

$$f_{\mathrm{exact}} = f^* \pm f^* GCI_{12} \tag{7.26}$$

$$f^* = f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{7.27}$$

The uncertainty bar is shown in Figure 7.35.

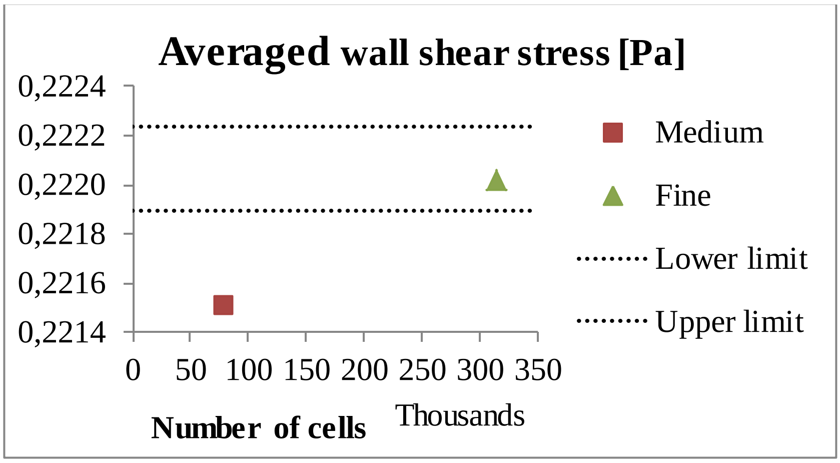

Figure 7.35. Result of GCI procedure: uncertainty bar for the fine solution

One thing we can point out is that the number of GCI_{12} obtained is low (the percentage value is about 7E-2%). Therefore the influence on this result of core mesh is relatively low too. This is a numerical proof that strengthens the observation made for Figure 7.31. Therefor the mesh medium is chosen for the next works.

7.8 Turbulence Model Sensitivity Analysis

One parameter which has not taken into account so far is turbulence model. Since GWF are defined for the k-ε turbulence models family, it is interesting to look at the results obtained with the other two models available, i.e. Realizable and RNG. The turbulence model adopted doesn't change the law of the wall of the wall function adopted, but it has an effect on the computation of wall boundary conditions. Indeed, even if the law of the wall does not depend on the turbulence model at all, the expressions used by the code to compute wall boundary conditions contain quantities directly affected by the turbulence model chosen. One quantity directly affected by turbulence model is turbulent kinetic energy. Even if the equation for k does not change significantly with the turbulence model (this can be demonstrated if we look at (2.18)). Diffusion of k in wall perpendicular direction is set to zero, this means that there can be diffusion of k only between near wall cells. The way to compute ε, which changes with the turbulence model, is defined by the wall function for the near wall cells. The only difference resides in the way turbulent viscosity is computed. Moreover, turbulent kinetic energy can be transported into near wall cells by advection; the value of k for adjacent near wall cells is affected by the turbulence model chosen. Turbulence model sensitivity analysis has been carried out using the mesh adopted for *Diffuser_mesh3*. Double precision solver and second order discretization scheme for the advective terms have been used.

Table 7.16. Simulations made for turbulence model sensitivity analysis

Name of the case	Turbulence model	Wall functions
Diffuser_mesh3	k-epsilon standard	SWF
Diffuser_mesh3_rea	k-epsilon realizable	SWF
Diffuser_mesh3_rng	k-epsilon RNG	SWF

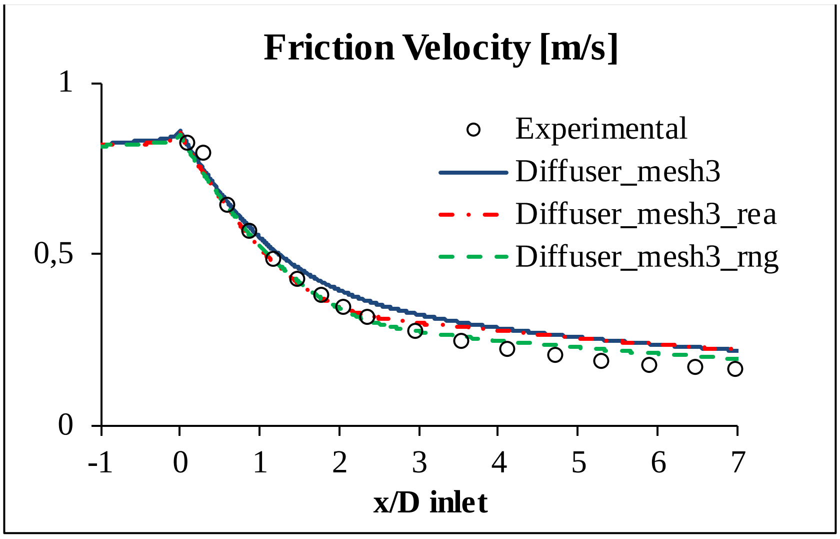

Figure 7.36. Friction velocity chart: turbulence model sensitivity analysis

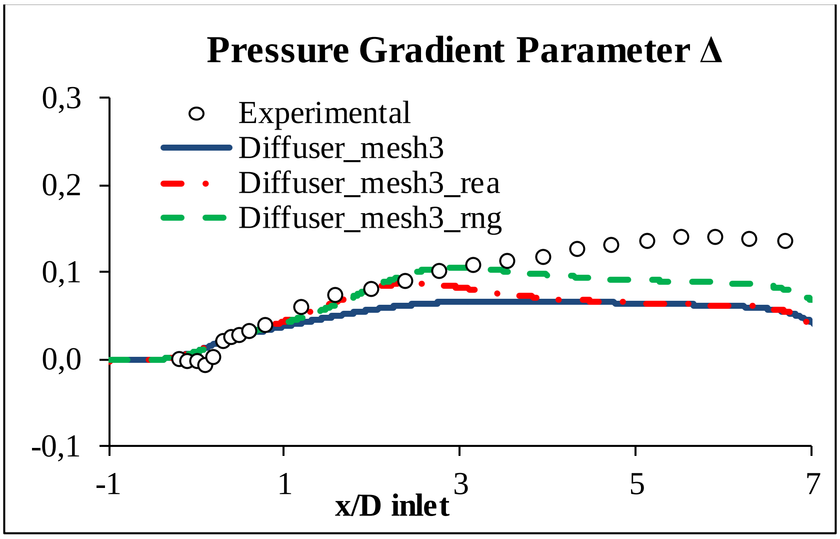

Figure 7.37. Pressure gradient parameter: turbulence model sensitivity analysis

The other two turbulence models of k-ε family (Realizable and RNG) provide better results even with SWF as wall treatment. If we look at Figure 7.36 we can notice that the Realizable model can predict the correct values of friction velocity until around 2 feeding pipe diameter from the inlet of the diffuser. The RNG model can predict correct results for a wider range, i.e. until around 3 feeding diameter from the inlet. However, these two models are not able to give good agreement in the last part of the diffuser. Realizable provides values higher and these values become equal to the values provided by Standard model. RNG overestimates friction velocity too, but the difference from experimental data is lower. Pressure gradient parameter chart shows a similar trend, but in this case the values are underestimated. After the analysis of these charts it seems that Realizable and RNG model provides good results. However we must check dimensional velocity profile in order to evaluate if wall tangential velocity at near wall cells is correctly estimated. If not, this means that values of wall shear stress are in accordance with experimental data only because of compensation error.

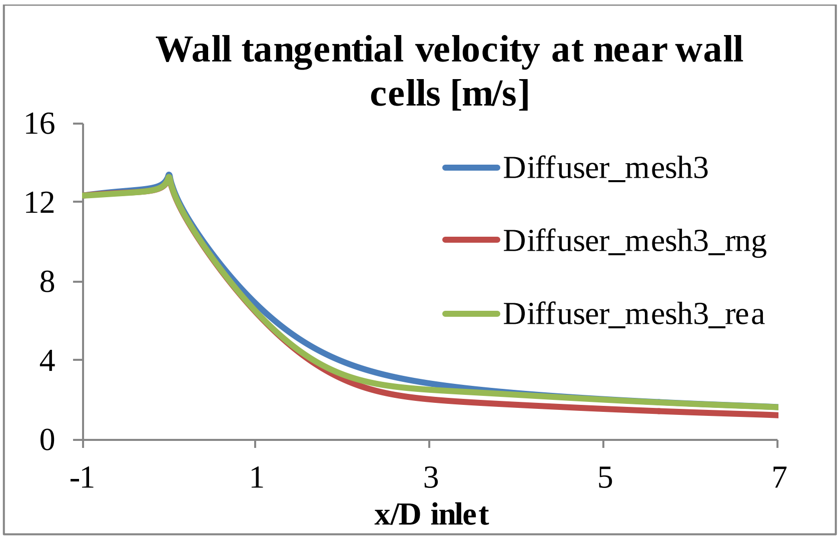

Figure 7.38. Wall tangential velocity at near wall center cells: turbulence model sensitivity analysis

Looking at expression (5.17), we can explain the trend of wall shear stress when changing turbulence model. First thing to say is that the results obtained with standard model allows to obtain good values of wall tangential velocity at near wall center cells for stations 5, 10 and 15 (Figure 7.23). The value at station 1 is underestimated. Instead, numerical values of wall tangential velocity at near wall cells obtained with RNG and Realizable are lower (as we can see from Figure 7.36). Therefore the estimation of this quantity with Realizable and RNG model is wrong. For example we can look at dimensional velocity profiles at station 5 obtained with standard, realizable and RNG models (Figure 7.39).

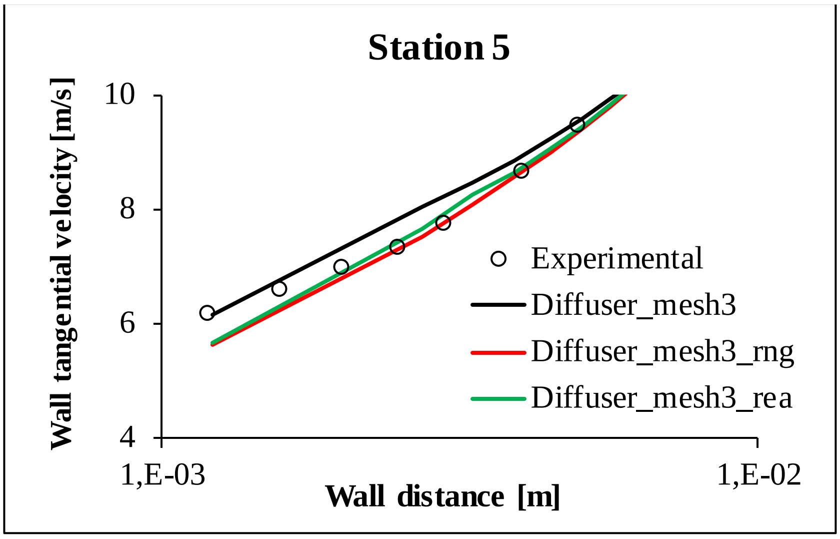

Figure 7.39. Dimensional velocity profile at station 5: turbulence model sensitivity analysis

It can be noticed that wall tangential velocity at near wall center cells is correctly estimated with Standard model, while the other two models underestimate it. even if they predict better the dimensional profile far away from the wall. However, the purpose of this work is to evaluate the goodness of Wall Functions approach. Therefore we want to find the turbulence model that allows to predict correctly the near wall flow behavior, even if it doesn't predict perfectly the core flow behavior.

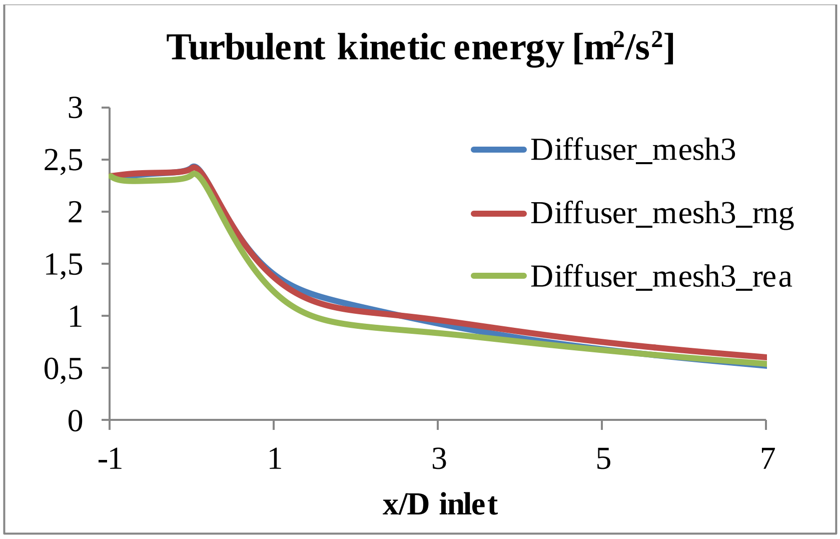

Figure 7.40. Turbulent kinetic energy at near wall center cells: turbulence model sensitivity analysis

Figure 7.40 shows the values of turbulent kinetic energy for the near wall center cells obtained with the three different turbulence models.
We know that velocity profile doesn't follow the SWF log law. To be more precise the real profiles follow a log law but with a different slope with respect to the one adopted by SWF. Since Realizable and RNG turbulence models don't allow to obtain good values of wall tangential velocity at near wall center cell (while Standard model allows to) then the use of these two models for the implementation of GWF is not recommended (we want to avoid compensation effects).

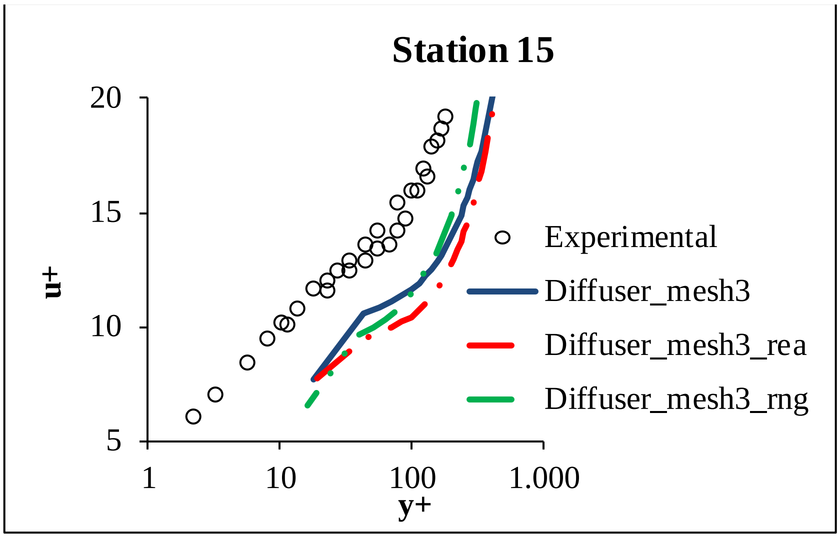

Figure 7.41. Dimensionless velocity profile for station 15: turbulence model sensitivity analysis

An analysis of dimensionless velocity profile at station 15 (Figure 7.41) shows that all the models adopted cannot predict correctly the last part of the diffuser, even if the switch from Standard to Realizable to RNG model involves an improvement of the results in terms of wall shear stress (even if it is driven by compensation effects, as already stated). As conclusion of this paragraph, standard k-ε turbulence model has been chosen for being used for next works, which involve the evaluation of the results obtained with GWF.

7.9 Implementation of GWF and Evaluation of the Results

Thanks to all the considerations reported in the previous paragraphs (7.5, 7.6, 7.7 and 7.8), the mesh and the turbulence model suitable for a simulation with the implemented GWF (standard k-ε) have been defined. Table 7.17 summarizes the parameters of the mesh adopted.

Table 7.17. Parameters of the mesh used for the implementation of GWF

Boundary layer thickness	0,0024325	m
Characteristic length of quadrilateral element of core mesh	0,001016	m
Number of cells	79478	

Before proceeding with the work, it is necessary to focus on the way the non-equilibrium function Ψ is defined. First of all, the first expression of Ψ given by Popovac and Hanjalic (3.72) can be reformulated in order to make Ψ function of only dimensionless quantities (3.74). However, in order to avoid singularities when wall shear stress tends to zero, the authors propose to modify the definition (3.72) by substituting u_τ with u_k. This substitution leads to definition (3.81), which can be written with only dimensionless quantities too (definition (3.82)). It is good to recall definition (3.82).

$$\Psi = 1 - \frac{y^*}{u^* \kappa} C_U \frac{\mu}{\rho^2 u_k u_\tau^2} \tag{7.28}$$

$$C_U^* = C_U \, adim = C_U \frac{\mu}{\rho^2 u_k u_\tau^2} \tag{7.29}$$

Expression (7.28) represents a rigorous definition of non-equilibrium function Ψ, in the meaning that it is equivalent to the definition of Ψ given by GWF authors. However, the term *adim* contains the square of friction velocity at the denominator. It has been noticed that, if the wall shear stress is small (and consequently also u_τ is small) the convergence of the calculation is not predictable. Indeed our study case (axial symmetric conical diffuser) presents a wall shear stress which is decreasing with the axial coordinate; this is a problem because, for example, if the CFD simulation predicts a separating and reattaching flow the wall shear stress goes to zero in that region, and therefore the denominator of (7.29) goes to zero, causing an error. Moreover, convergence problems have been found out also if the CFD simulation does not predict a separating flow. For the reason just explained, we decided to work with the following logical path:

- The first implementation of GWF has been made computing the *adim* term only with u_k as velocity scale, i.e.

$$adim = \frac{\mu}{\rho^2 u_k^3} \tag{7.30}$$

We must point out that the substitution of friction velocity with u_k is a common practice in Wall Functions approach (see for example the passage that leads from the simplest implementation to the standard implementation of SWF, paragraph 3.6.2). Therefore the substitution of the velocity scale inside the term $adim$, even if it is not rigorous, can be considered acceptable. This substitution has been made in order to have a bigger denominator of (7.29); indeed it has been found out that the decrement of turbulent kinetic energy for near wall center cells inside the diffuser is not as big as the decrement of wall shear stress.

- If the calculation converges, data are post-processed and their quality can be discussed.
- Next implementations of GWF are made using respectively $u_k^2 u_\tau$, $u_\tau^2 u_k$ and u_τ^3 as the velocity scale inside $adim$. This work is presented in paragraph 7.10. Note: if the calculation doesn't converge with a velocity scale, it is expectable that the calculation doesn't converge also when using the next velocity scales.

The calculation with the first GWF implemented has been found out to converge. The simulation is named *Diffuser_mesh3_gwf*. The results obtained can be post-processed and compared with both experimental data and results obtained from the simulation which adopts SWF (results discussed in the paragraph 7.6; the simulation is named *Diffuser_mesh3*) and NEWF (the simulation is named *Diffuser_mesh3_newf*). Simulations have been carried out with a double precision solver and with second order discretization schemes for all the advective terms.

Table 7.18. Parameters of the simulations made with NEWF and with GWF

Name of the case	Turbulence model	Wall functions
Diffuser_mesh3_newf	k-epsilon standard	NEWF
Diffuser_mesh3_gwf	k-epsilon standard	GWF

First of all, an analysis of the trend of the non-equilibrium Ψ function and also of the single terms which compose Ψ is required.

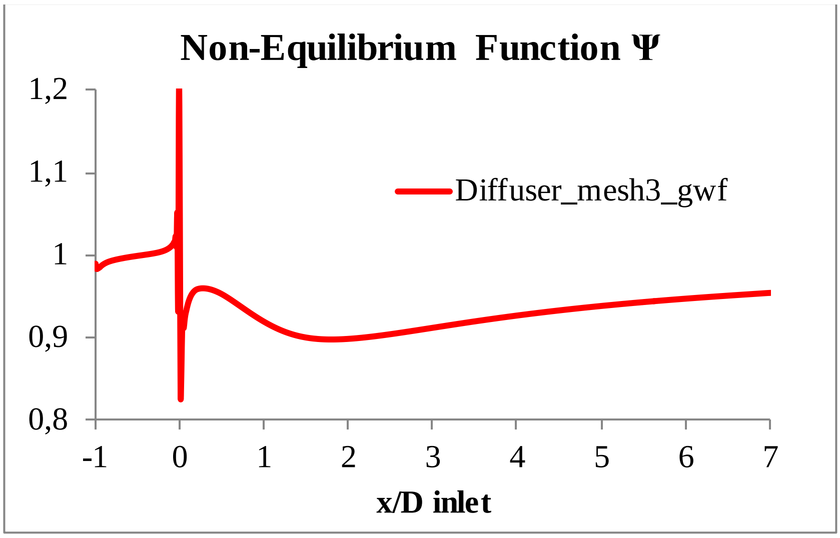

Figure 7.42. Chart of Non-Equilibrium function Ψ obtained for the first implementation of GWF

As stated before, Ψ is a function which represents how much a boundary layer is a non-equilibrium one. When dealing with an equilibrium boundary layer Ψ is equal to one, otherwise Ψ can be greater or smaller than 1. The trend of this function obtained for our geometry states that:

- Inside the feeding pipe the values are closer to 1. This is in accordance with the fact that a pipe flow presents an equilibrium boundary layer. The difference between SWF and GWF should be then negligible in this region. This is correct because SWF are a suitable wall treatment for this kind of flow and the goodness of the results obtained with SWF in this region has already been proved (see for example paragraph 7.3). Moreover the fact that GWF are equal to SWF is a good thing for the general validity of GWF, i.e. GWF provides good results also for equilibrium boundary layer flows.
- In a narrow region close to the diffuser inlet, the trend is strange, i.e. there is a high peak followed by a trough. This strange trend can be explained by the sudden change of the wall inclination when entering the diffuser. In the reality the change of inclination is not confined to a point, but it happens in a finite region, even if very narrow. Therefore, if the change happens in one point, the sudden different way to compute C_{Uw} (it happens because the wall tangential direction has a sudden change) can be one cause of this strange trend
- Inside the diffuser, all the Ψ values are lower than 1. In the first part the first derivative of the function is negative and the function reaches his local minimum value around 2 feeding pipe diameters from the inlet of

the diffuser (where Ψ is equal to about 0.9). After the minimum, there is a region that lasts until the outlet where the first derivative is positive and Ψ reaches the value of about 0.95 at the outlet. This trend shows a good accordance with what stated by the authors of the experiment. Indeed they stated that the irreversibility is confined in the first part of the diffuser while the boundary layer tends to reassume an equilibrium form when approaching the outlet of the diffuser.

The function Ψ we have just discussed about is composed by different terms. We want to check the weights of the single terms on the whole function.

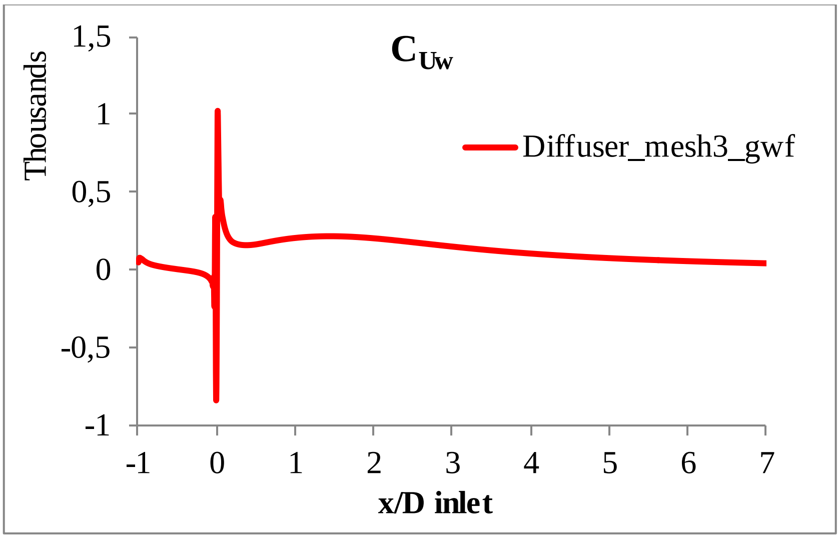

Figure 7.43. Chart representing C_{Uw} trend obtained for the first implementation of GWF

Since Ψ is smaller than 1, C_{Uw} is positive (indeed all the other terms that appear inside Ψ are always positive, see (7.28)). C_{Uw} represents how much a boundary layer is a non-equilibrium one; to be more precise it contains the dimensional non-equilibrium terms, i.e. pressure gradient and advective terms. Excluding from the analysis the narrow region placed around the inlet of the diffuser, the trend of this quantity is similar to the trend of function Ψ. We notice that the biggest values of C_{Uw} are placed around 2 feeding pipe diameter from the inlet and close to the outlet this term goes to zero. Therefore, as we stated about Ψ, the boundary layer presents a strong non-equilibrium region right after the inlet of the diffuser; moving along the wall, the non-equilibrium region is weaker and it shows a trend asymptotical to an equilibrium boundary layer. After analyzing the term C_{Uw} in its entirety, it's interesting to check the weight of the three different terms that are summed in order to obtain the whole C_{Uw} (definition (3.63) and chapter 6). The three terms are:

$$\rho U_w \frac{\partial U_w}{\partial \boldsymbol{w}} \qquad \textbf{(7.31)}$$

$$\rho U_z \frac{\partial U_w}{\partial \boldsymbol{z}} \qquad \textbf{(7.32)}$$

$$\frac{\partial P}{\partial \boldsymbol{w}} \qquad \textbf{(7.33)}$$

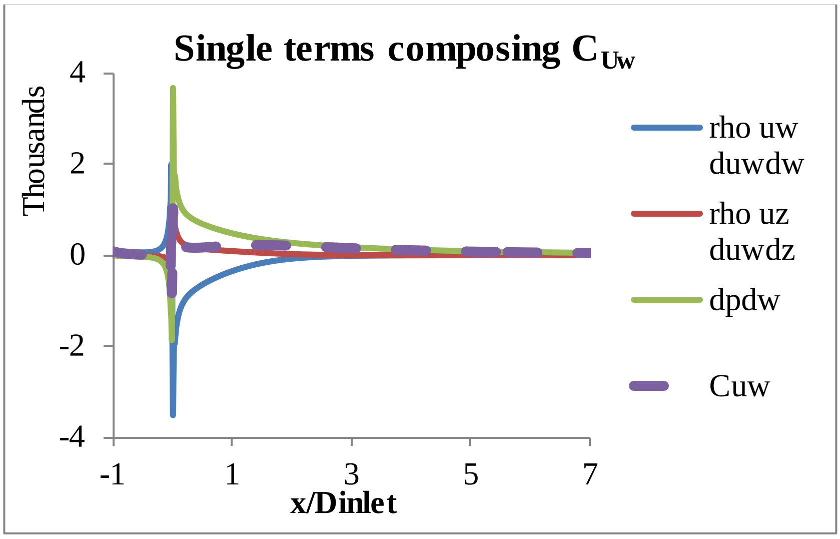

Figure 7.44. Chart showing the trend of C_{Uw} and of the terms which compose C_{Uw}. Lines blue, red and green correspond respectively to terms defined by (7.31), (7.32) and (7.33)

The second term $\rho U_z \frac{\partial U_w}{\partial z}$ (red line) is the smallest of the three and it is always positive, while the terms $\rho U_w \frac{\partial U_w}{\partial w}$ (blue line) and $\frac{\partial P}{\partial w}$ (green line) have opposite sign and a similar trend (to be more precise $\rho U_w \frac{\partial U_w}{\partial w}$ is always negative and $\frac{\partial P}{\partial w}$ is always positive – as expected for an adverse pressure gradient flow). C_{Uw} is positive; therefore it is the pressure gradient sign which determines the sign of the whole term. We can conclude that the first and the third terms are in contrast each other and that the first one has the goal to minimize the effect of the third. If we make a rough comparison with NEWF, we can state that, even if the way NEWF are written is not concerning the non-equilibrium function Ψ at all, since its law of the wall considers only pressure gradient we can expect equivalent values of Ψ smaller and then small values of wall shear stress obtained (3.84).

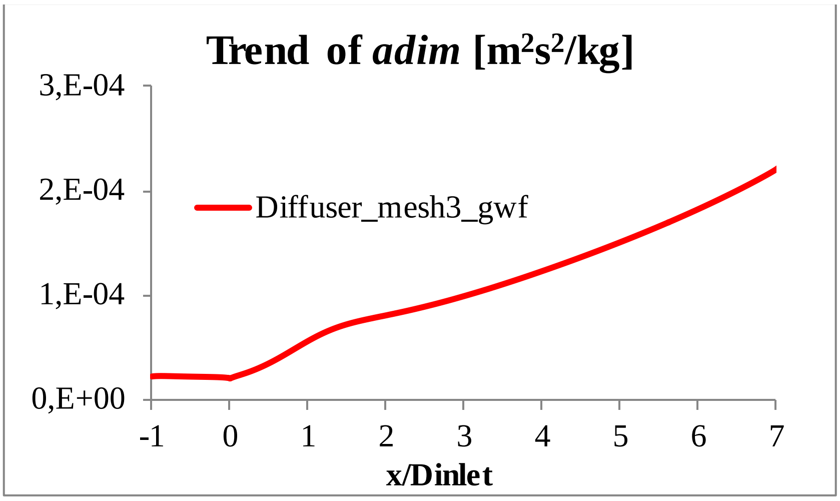

Figure 7.45. Trend of the term *adim* obtained with the first implementation of GWF

The chart with the values of the term $adim$ is shown in Figure 7.45. It is only dependent on the velocity scale which is placed at the denominator of the function and it is always positive. The product of C_{Uw} by $adim$ gives the term C^*_{Uw}, which is practically the one to put inside Ψ function. If $adim$ is small, the just cited product will be small and therefore Ψ will be closer to 1. On one hand this seems to be good for the convergence, but on the other hand Ψ closer to one means less difference with SWF. The correctness must be carefully checked with a data post-processing.

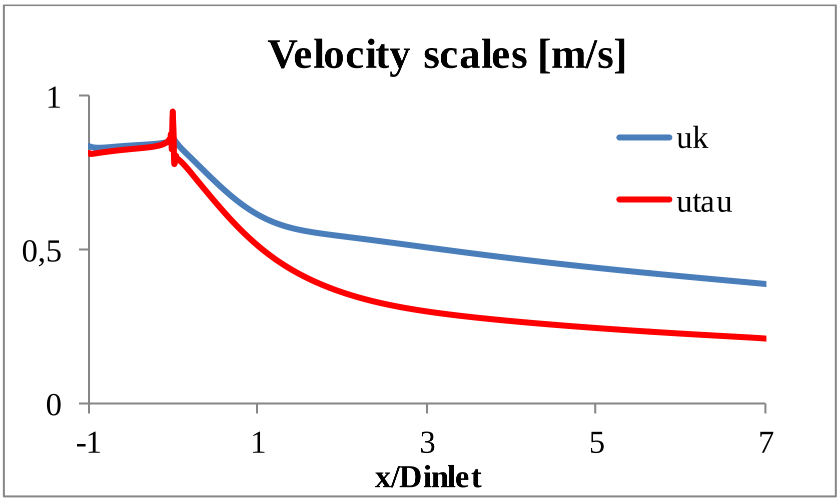

Figure 7.46. Velocity scales u_k and u_τ obtained with the first implementation of GWF

The trend of $adim$ is justified by looking at Figure 7.46, which reports the values of the two velocity scales. Friction velocity is always smaller than u_k all along the wall. This means that if we use $u_\tau^2 u_k$ as velocity scale inside the term $adim$, the cited term will be higher and then Ψ will be smaller (since C_{U_w} is positive Ψ is much smaller than 1). Ψ smaller means that there will be more difference between SWF and GWF but then the convergence of the iterations cannot be guaranteed “a priori”.

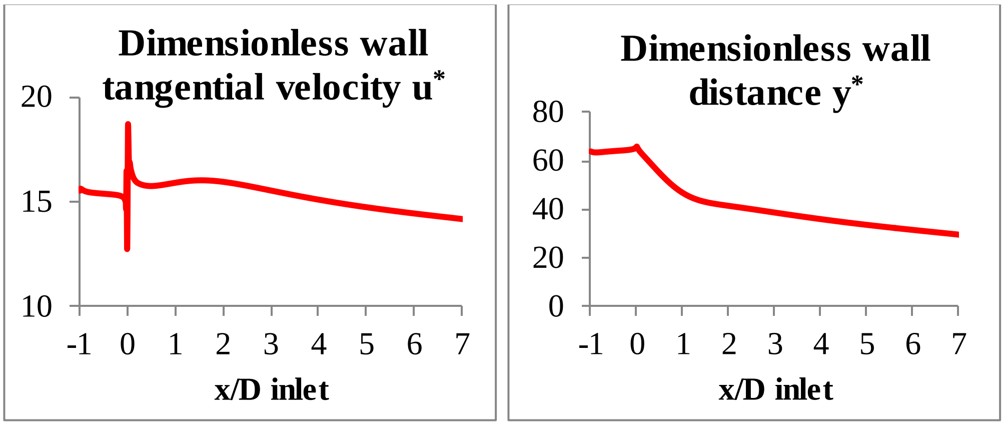

Figure 7.47. u* and y* obtained with the first implementation of GWF. Note: y* is not the real Fluent value but it is computed using a constant wall distance (see paragraph 4.3)

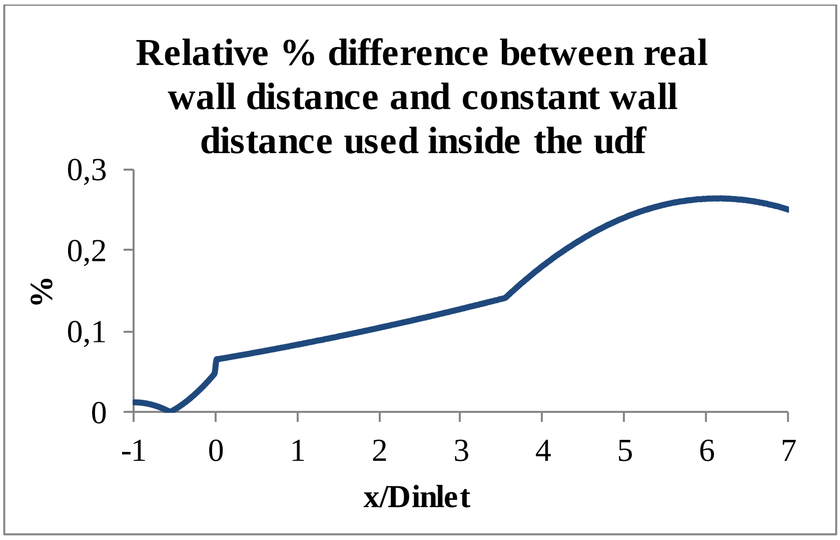

Figure 7.48. % Relative % difference between real wall distance and constant wall distance used inside the udf; note: real wall distance is adopted as reference value

Last thing to be checked is the error introduced by the use of a constant wall distance introduced by the user for the calculation of y*. Indeed, even if the thickness of the boundary layer mesh adopted is constant, there are no proofs that also the wall distance of the near wall center cells is strictly constant. If the error is small it can be neglected since its influence on the function Ψ will be small too. Figure 7.48 reports the relative percentage difference between y^* obtained from Fluent post-processing and the y^* computed inside the udf and adopted for the computation of Ψ. Since the maximum difference is on the order of 0.3%, the error introduced by the approximation can be reasonably neglected. Now, we can move on the comparison of the results obtained with the first implementation of GWF with the experimental data and with results obtained with NEWF.

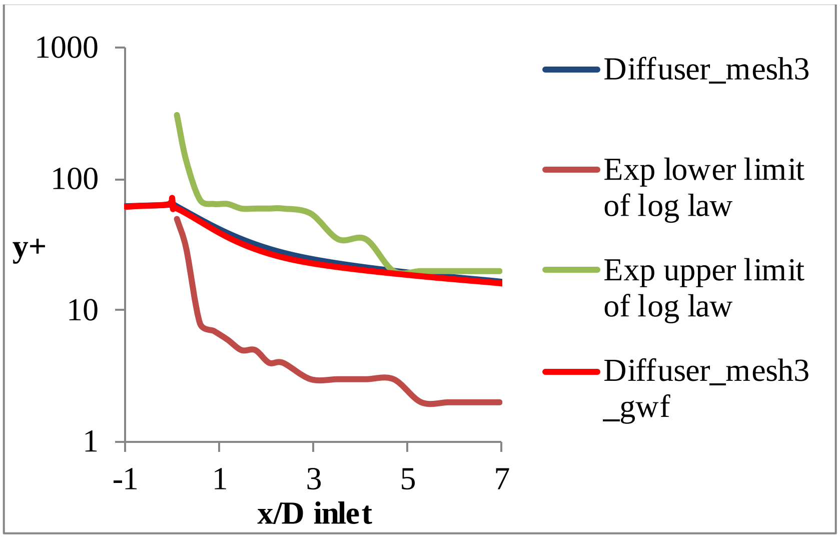

Figure 7.49. Dimensionless values of wall distance for near wall center cells: values obtained with SWF and with the first implementation of GWF compared with experimental range of log law

We can notice that all the values of y^+ fall into experimental logarithmic range also when using the first implementation of GWF. Red line (*Diffuser_mesh3_gwf*) is smaller than blue line; therefore we expect lower values of wall shear stress when using GWF.

The use of the first implementation of GWF provides an improvement of the results in terms of wall shear stress (Figure 7.50). Indeed, we expect that, since Ψ is always lower than 1 (we are neglecting the narrow region close to the inlet of the diffuser), also wall shear stress values obtained from *Diffuser_mesh3_gwf* will be lower than same values obtained from *Diffuser_mesh3* (see equation (3.84).

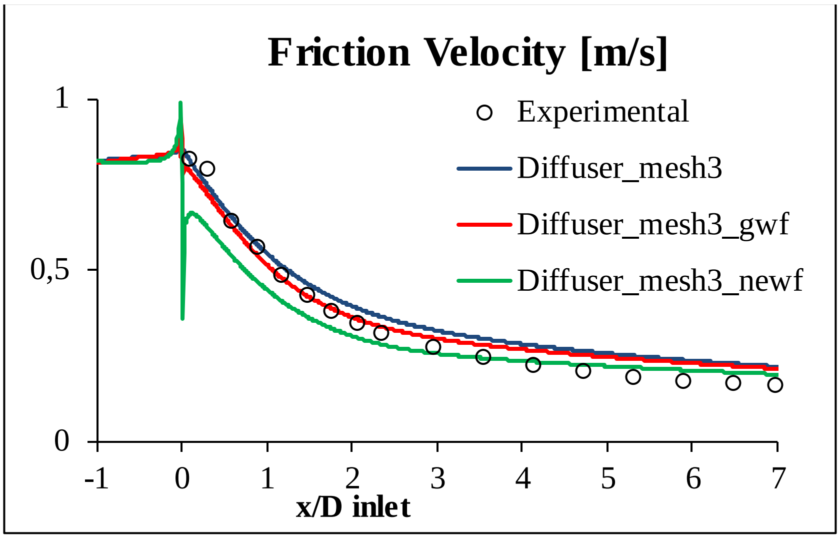

Figure 7.50. Friction velocity: values obtained for *Diffuser_mesh3*, *Diffuser_mesh3_gwf*, *Diffuser_mesh3_newf* compared with experimental data

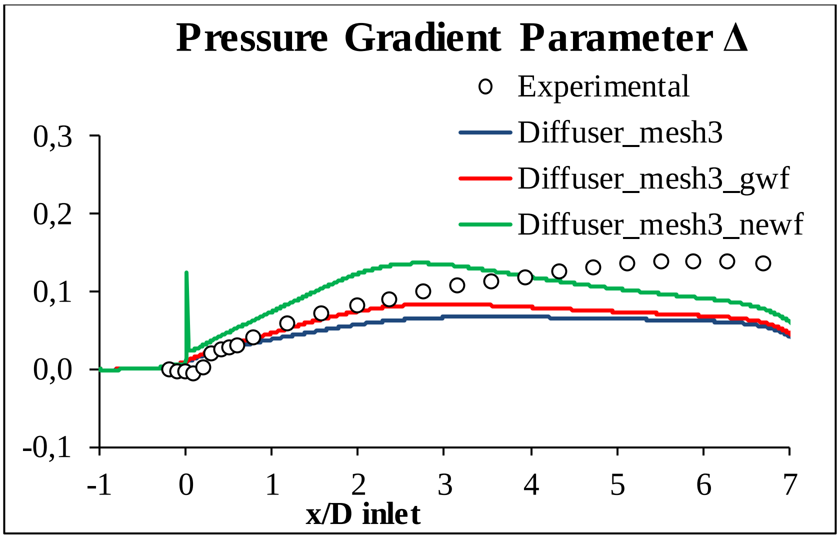

Figure 7.51. Pressure gradient parameter: values obtained for *Diffuser_mesh3*, *Diffuser_mesh3_gwf*, *Diffuser_mesh3_newf* compared with experimental data

In this situation the use of the first implementation of GWF provides better results compared to SWF. To be more precise, there is good accordance for wall shear stress between 1 and 2 feeding pipe diameter from the inlet of the diffuser; greater values the accordance is not perfect but values are closer to experimental data. At the diffuser inlet the values are similar, and both of them predict the wall shear stress to sharply decrease right after entrance in the diffuser, while experimental data show a narrow region inside the diffuser where the wall shear stress is almost constant. However, close to the outlet the difference between *Diffuser_mesh3* and *Diffuser_mesh3_gwf* is negligible, and for the rest of the wall *Diffuser_mesh3_gwf* overestimates wall shear stress too (even if less than *Diffuser_mesh3_gwf*). We can guess then that the values of Ψ are not small enough to guarantee a perfect accordance between CFD and experimental data. Therefore we expect that when the other implementations of GWF with the different velocity scales inside the term *adim* will be implemented, better results will be obtained. Regarding *Diffuser_mesh3_newf* results, the first thing we notice is that the discontinuity at the inlet of the diffuser is significantly higher than *Diffuser_mesh3_gwf*. This can be explained by the fact that GWF have a term which has opposite sign of pressure gradient and it balances the effects of it while NEWF considers only pressure gradient (Figure 7.44). Regarding results inside the diffuser, *Diffuser_mesh3_newf* underestimates values of friction velocity until 3 feeding pipe diameters from the inlet of the diffuser; for higher values the friction velocity is overestimated, but the values are closer to experimental data if compared to *Diffuser_mesh3* and *Diffuser_mesh3_gwf*. We move now to the analysis of the chart of pressure gradient parameter (Figure 7.51). *Diffuser_mesh3_gwf* profile has a region where the accordance with experimental data is good, even though the accordance is not perfect for the entire wall; the accordance between *Diffuser_mesh3_gwf* and experimental data is good until about 2 feeding pipe diameter from the inlet, while when using SWF the accordance can be considered acceptable only until 1 feeding pipe diameter from the inlet. On the other hand *Diffuser_mesh3_newf* results are in discordance with experimental data all along the wall.

As we did in the previous paragraphs, we want to give a rough explanation of the causes of the error related to the computation of wall shear stress. Figure 7.52 shows the experimental values of the inverse of the slopes of the first implementation of GWF log law compared to the inverse of the experimental slopes of log law. The inverse of the slopes of *Diffuser_mesh3_gwf* (which are called A) are taken from the log law for the fully turbulent region computed inside the udf, i.e.

$$A = \kappa\Psi \tag{7.34}$$

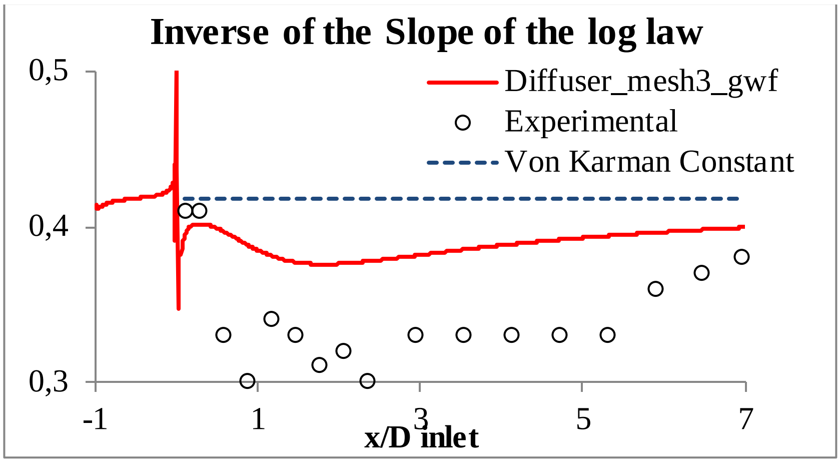

Figure 7.52. Inverse of the slope of the log law: *Diffuser_mesh3_gwf* and experimental values (also Von Karman constant, which is the inverse of the slope of SWF log law, is shown)

It's necessary to point out that the experimental values are obtained for a law of the wall that relates u^+ to y^+, while the ones plotted for *Diffuser_mesh3_gwf* come from a law of the wall that relates u^* to y^*. Moreover, the experimental values have been obtained fitting the experimental data with a logarithmic law. Therefore they cannot be considered truly right. They are useful because they can give a reasonable and a physical range for the values of Ψ, which otherwise cannot have an experimental analogous. The values of the inverse of the slopes for *Diffuser_mesh3_gwf* are higher of the experimental ones all along the wall. Therefore with lower values of Ψ the accordance will be better.

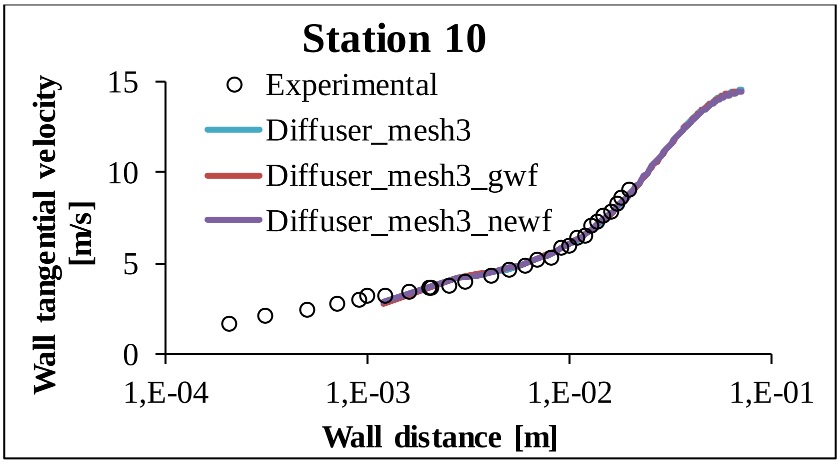

Figure 7.53. Dimensional velocity profile for station 10: comparison of *Diffuser_mesh3*, *Diffuser_mesh3_gwf* and *Diffuser_mesh3_newf* profiles with experimental data

Now we want to check if the estimation of wall tangential velocity at near wall cells is still good. Figure 7.53 show the wall tangential dimensional velocity profile for station 10. We can notice that the influence of Wall Functions adopted is negligible for the dimensional velocity profile, which is therefore correctly estimated with all the three Wall Functions adopted. Same results have been found out for stations 1, 5 and 15 and also for the other dimensional quantities measured by the authors of the experiment.

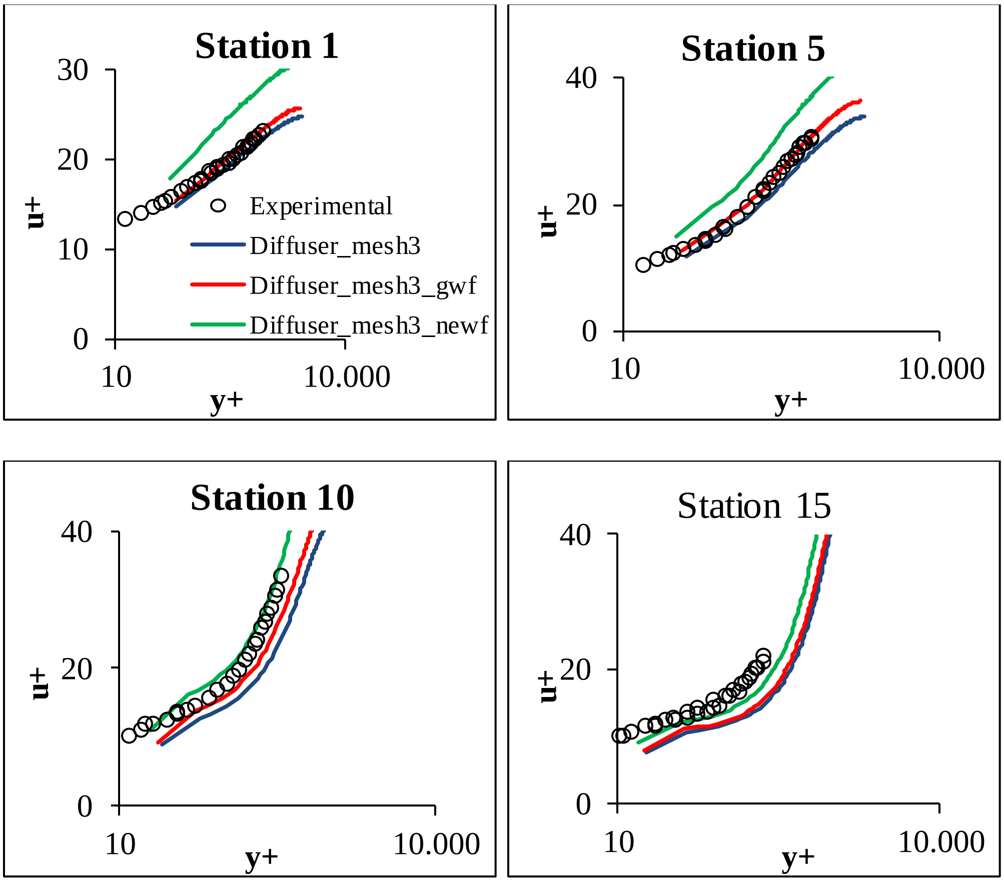

Figure 7.54. Dimensionless velocity profile for stations 1, 5, 10, 15: comparison of *Diffuser_mesh3*, *Diffuser_mesh3_gwf* and *Diffuser_mesh3_newf* with experimental data

The charts of dimensionless velocity profiles (Figure 7.54) bring us to the same conclusions made after the observation of friction velocity chart (Figure 7.50). For stations 1 and 5, *Diffuser_mesh3_newf* underestimates the value of friction velocity and therefore the dimensionless profiles are far from the experimental one. For these two stations *Diffuser_mesh3_gwf* are the Wall Functions that predicts the profiles with the best accordance with experimental data. For stations 10 and 15, on the other hand, the values of friction velocity obtained with *Diffuser_mesh3_gwf* are the closest to experimental data, as the dimensionless velocity profiles is.

One thing which is important to point out is that the way turbulent kinetic production and turbulent dissipation rate for the near wall center cells are not computed rigorously as proposed by Popovac and Hanjalic. The way adopted by the code for the computation of the production of turbulent kinetic energy, when it adopts GWF is:

$$P_k = \frac{\tau_w^2}{\mu}\frac{1}{\psi\kappa y^*} \tag{7.35}$$

$$P_k = \frac{\tau_w^2}{\mu}\frac{1}{\frac{\psi\kappa\rho y u_k}{\mathrm{\mu}}} \tag{7.36}$$

$$P_k = \frac{\tau_w^2}{\mu}\frac{1}{\frac{\psi\kappa\rho y u_k}{\mathrm{\mu}}} \tag{7.37}$$

The way adopted by the code for the computation of the turbulent dissipation rate, when it adopts GWF is:

$$\varepsilon = \frac{u_k^3}{\psi\kappa y} \tag{7.38}$$

This is the way proposed by Popovac and Hanjalic, respectively for the computation of the production of turbulent kinetic energy and for the computation of turbulent dissipation rate:

$$P_k = \rho\frac{u_k^3}{\psi\kappa y_P} \tag{7.39}$$

$$\varepsilon_P = \frac{u_k^3}{\kappa y_P} \tag{7.40}$$

The expressions (7.37) and (7.38) are different from the expressions (7.39) (7.40). The difference is caused by the general expressions used by Fluent for the computation of these quantities when the Wall Functions are implemented through a user defined function (expressions (7.41) and (7.42)).

$$P_k = \frac{\tau_w}{\mu} f'(y^*) \tag{7.41}$$

$$\varepsilon = \frac{u_k^4}{\upsilon} f'(y^*) \tag{7.42}$$

It is impossible to write a law of the wall whose first derivative ensures turbulent kinetic production and turbulent dissipation rate to be computed as proposed by Popovac and Hanjalic. Looking deeply at the different expressions, we notice that the difference for P_k just resides in a different velocity scale used ($u_\tau^4 u_k^{-1}$ versus u_k^3), whereas turbulent dissipation rate becomes a function also of the non-equilibrium function Ψ. The first difference can be considered acceptable for two reasons. The first one is that the substitution of one velocity scale with the other is a common practice in Wall Functions approach. The second is here described. The expression for production of turbulent kinetic energy comes from the general expression compatible with Boussinesq hypothesis, i.e.

$$P_k = \tau_w \frac{\partial U_w}{\partial y_w} \tag{7.43}$$

Using the dimensionless expression for the wall tangential velocity and for the wall distance (using the * *adimensionalization*) we can write the expression (7.43) as:

$$P_k = \tau_w \frac{\partial \frac{u^* u_\tau^2}{u_k}}{\partial \frac{\mu y_w^*}{\rho u_k}} \tag{7.44}$$

And we can take out from the differential the terms which are constants.

$$P_k = \frac{\rho \tau_w u_\tau^2}{\mu} \frac{\partial u^*}{\partial y_w^*} = \frac{\tau_w^2}{\mu} \frac{\partial u^*}{\partial y_w^*} \tag{7.45}$$

$$P_k = \frac{\tau_w^2}{\mu} \frac{\partial u^*}{\partial y_w^*} \tag{7.46}$$

Putting the first derivative of the law of the wall in place of the derivative in the expression we obtain the expression shown in equation (7.35). Therefore it seems that expression (7.35) have a physical consistence. Regarding turbulent dissipation rate expression, on the other hand, we don't have enough elements to verify his physical consistence. By the way, if we compare it to its general expression, we can assume that when using the expression (3.14) the length scale c_l is not constant and equal to $C_\mu^{0.75}\kappa^{-1}$ but it is instead a function of Ψ too, i.e. length scale becomes equal to $C_\mu^{0.75}\kappa^{-1}\psi^{-1}$. Since there is no way we can guarantee both the expression to be equal to Popovac and Hanjalic proposal, we are forced to analyze the results obtained with these expressions for the computations of wall boundary conditions. Figure 7.55 shows the values of turbulent kinetic energy, production of turbulent kinetic energy and turbulent dissipation rate for the near wall center cells obtained for *Diffuser_mesh3* and *Diffuser_mesh3_gwf*.

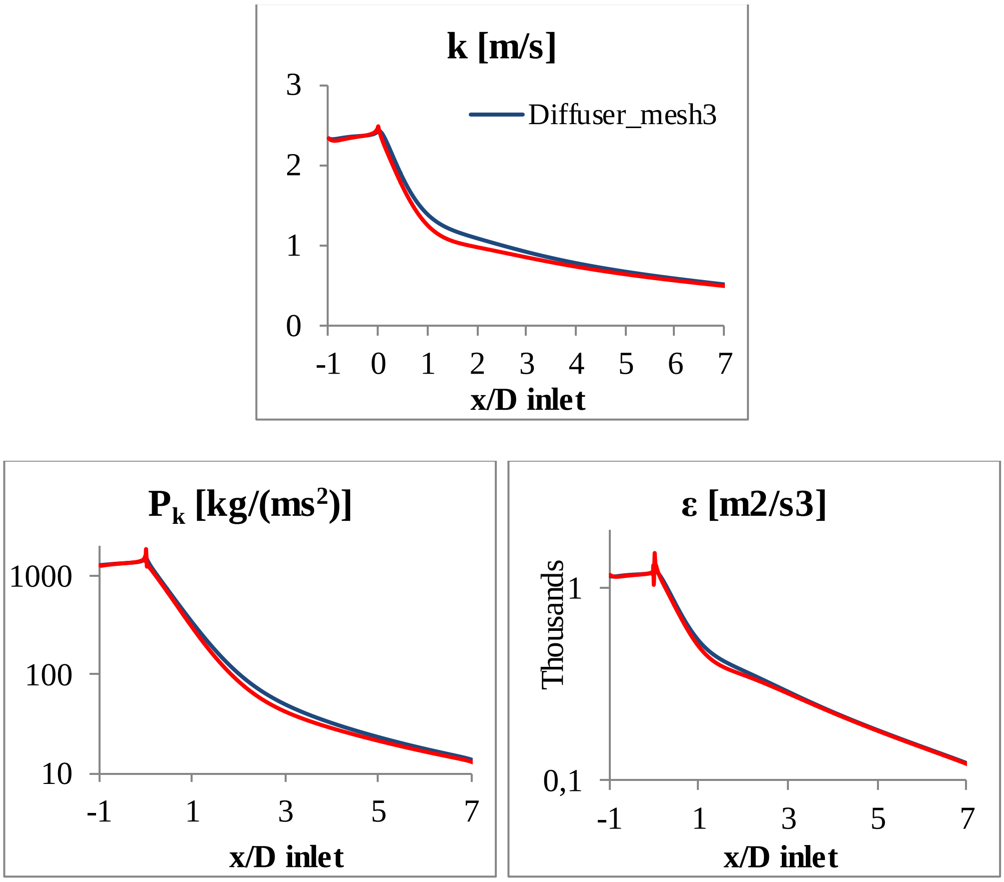

Figure 7.55. Profiles of turbulent quantities (k, P_k and ε) for near wall center cells obtained for *Diffuser_mesh3* and *Diffuser_mesh3_gwf*

The difference in the results between when passing from *Diffuser_mesh3* to *Diffuser_mesh3_gwf* is not very big. The only trend we can notice is a decrease of production of turbulent kinetic energy and of turbulent dissipation rate. The trend P_k is expectable because the expression for its computation has at the numerator the friction velocity to the fourth and Ψ, which are both smaller for *Diffuser_mesh3_gwf*. Regarding ε, on the other hand, its expression states that if Ψ is smaller than 1, then ε should be greater. However, ε is directly proportional to turbulent kinetic energy also, which is smaller for *Diffuser_mesh3_gwf* because of the decreased production of turbulent kinetic energy inside near wall cells. Therefore, inside the expression for ε, it is the decrease of k which prevails over the non-equilibrium function Ψ.

7.10 GWF Velocity Scale Sensitivity Analysis

Purpose of this paragraph is the implementation into Fluent 6.3 of different forms of GWF where the velocity scale inside the term $adim$ is changed. Summarizing, the new GWF implemented use the following velocity scales inside $adim$:

Table 7.19. Velocity scales inside *adim* adopted for GWF velocity scale sensitivity

Name	$adim$	Convergence of the solution
Diffuser_mesh3_gwf_utauuk2	$\frac{\mu}{\rho^2 u_k^2 u_\tau}$	Yes
Diffuser_mesh3_gwf_utau2uk	$\frac{\mu}{\rho^2 u_k u_\tau^2}$	Yes
-	$\frac{\mu}{\rho^2 u_\tau^3}$	No

Note: the case *Diffuser_mesh3_gwf_utau2uk* adopts a velocity scale that makes the term *adim* rigorously equal to the one proposed by the authors of GWF after the substitution of u_τ with u_k in the expression (3.82). Moving from the first to the last case, the term $adim$ increases its value (because u_τ has been already found out to be smaller than u_k). This leads the values of Ψ to become smaller. On one hand this is good because, since the first implementation of GWF overestimates the values of wall shear stress, smaller values of Ψ should improve the quality of the results. On the other hand the iterations may not converge to the solution. Indeed the simulation with the GWF which uses u_τ^3 as velocity scale does not converge.

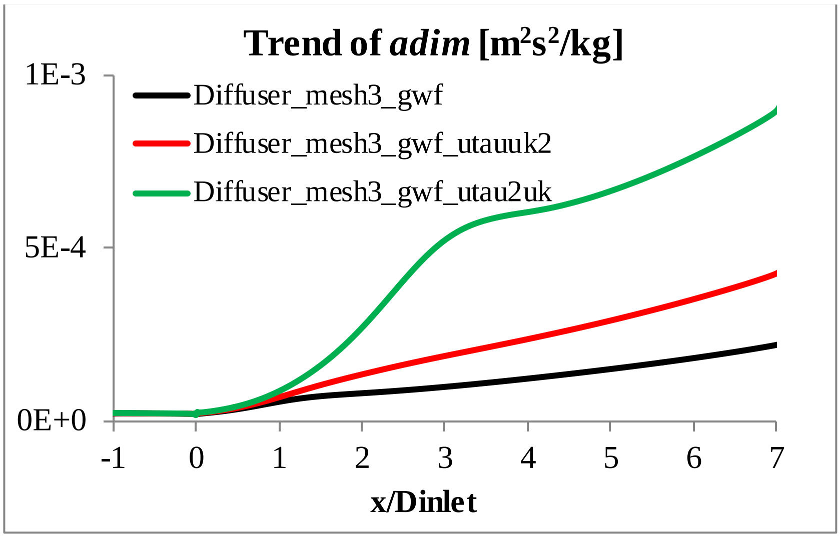

Figure 7.56. Trend of the term *adim*: GWF velocity scale sensitivity analysis

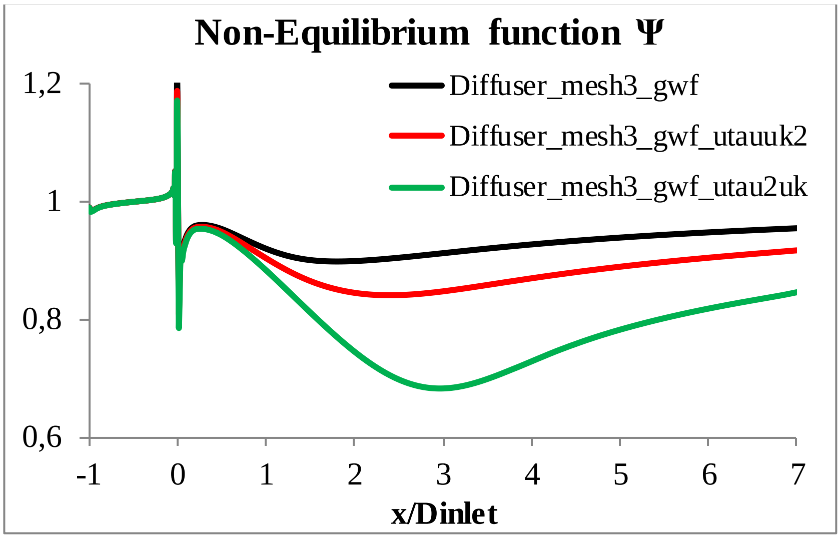

Figure 7.57. Non-Equilibrium function Ψ: GWF velocity scale sensitivity analysis

From an analysis of the quantity C_{Uw} and of Figure 7.56 we notice that the difference in the function Ψ is related only to the change in the term $adim$ (i.e. C_{Uw} chart has been found out not to change). This means that even if the changes in the function Ψ are significant (Figure 7.57), and therefore also the changes in the wall shear stress values are significant, the overall flow field is not influenced too much by it.

Figure 7.58 shows the values of the inverse of the slopes of the logarithmic law of the wall.

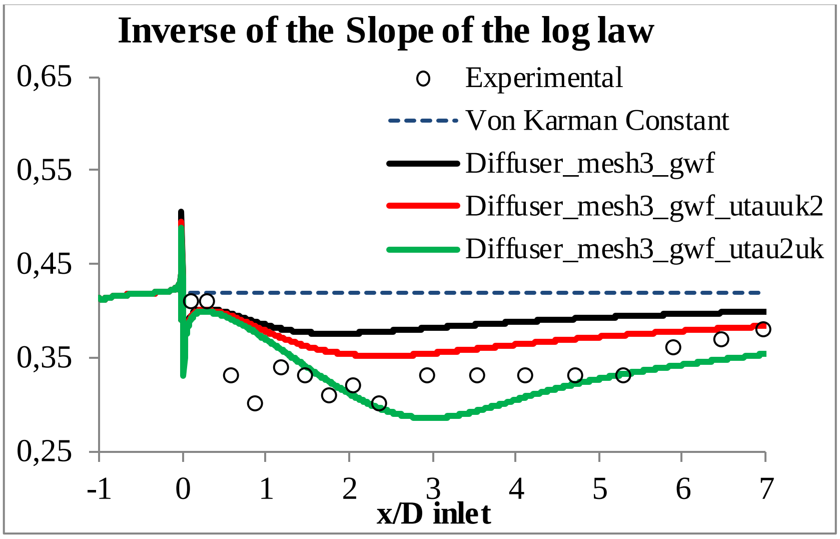

Figure 7.58. Slope of the logarithmic law of the wall: GWF velocity scale sensitivity

When introducing friction velocity inside the term *adim*, the values of the inverse of the slope of the log law decrease and they become closer to experimental data. To be more precise, *Diffuser_mesh3_gwf_utau2uk* has the lowest values and these values almost fall into experimental range around 2 feeding pipe diameters from the inlet of the diffuser. For greater values there is an underestimation of the values of the inverse of the slope, while from 5 feeding pipe diameter until the end the accordance is good. *Diffuser_mesh3_gwf_utauuk2*, on the other hand, predicts an overestimation of the inverse of the slope all along the wall, even if the values are closer to experimental data compared to *Diffuser_mesh3_gwf*.

Figure 7.59 shows the chart of the values of wall shear stress obtained. The conclusions we can made after looking at this chart are compatible with the observations made for the inverse of the slopes of the log law. The results obtained with *Diffuser_mesh3_gwf_utauuk2* are closer to the experimental data all along the wall compared to *Diffuser_mesh3*, and the accordance is very good until about 3.5 feeding pipe diameters from the inlet of the diffuser. *Diffuser_mesh3_gwf_utau2uk* underestimates friction velocity in the region between 1 and 4 feeding pipe diameters from the inlet, while for values greater than 5 feeding pipe diameter values become overestimated but they are the closest to experimental data.

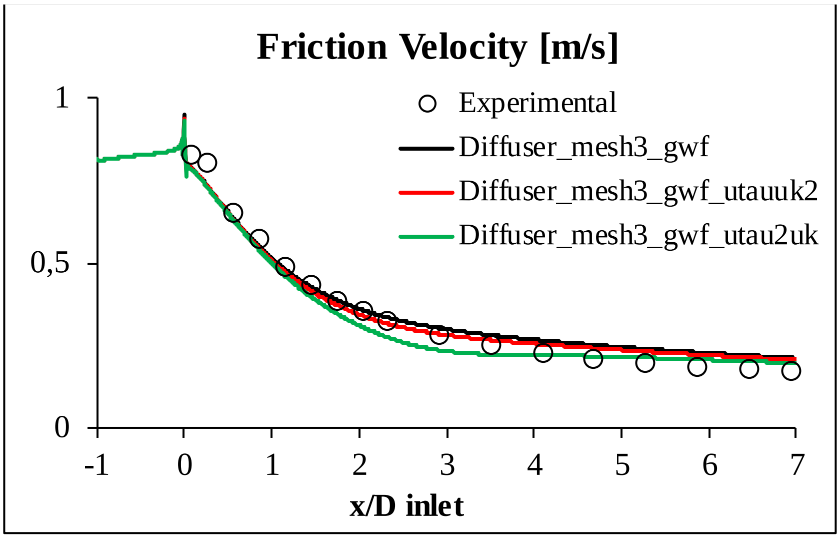

Figure 7.59. Friction velocity: GWF velocity scale sensitivity analysis

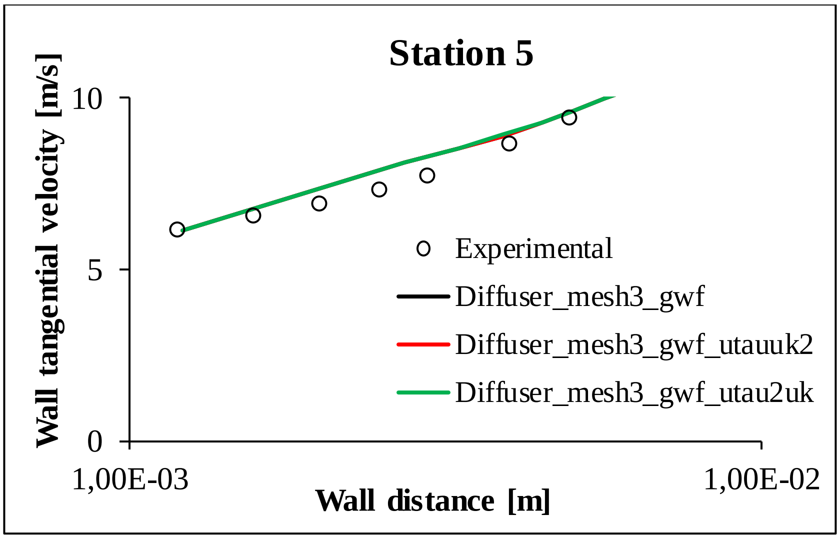

Figure 7.60. Dimensional velocity profile at station 5: GWF velocity scale sensitivity analysis

The analysis of dimensional wall tangential velocity profile at station 5 (Figure 7.60) shows that overall flow field is not significantly affected by Wall Functions. Like already observed in paragraph 7.9, Wall Functions are found out to have a negligible influence on all the dimensional quantities which have been measured by the authors of the experiment.

The chart of pressure gradient parameter can give us more elements for the evaluation of the goodness of the results obtained, and therefore we can evaluate better the performance of GWF.

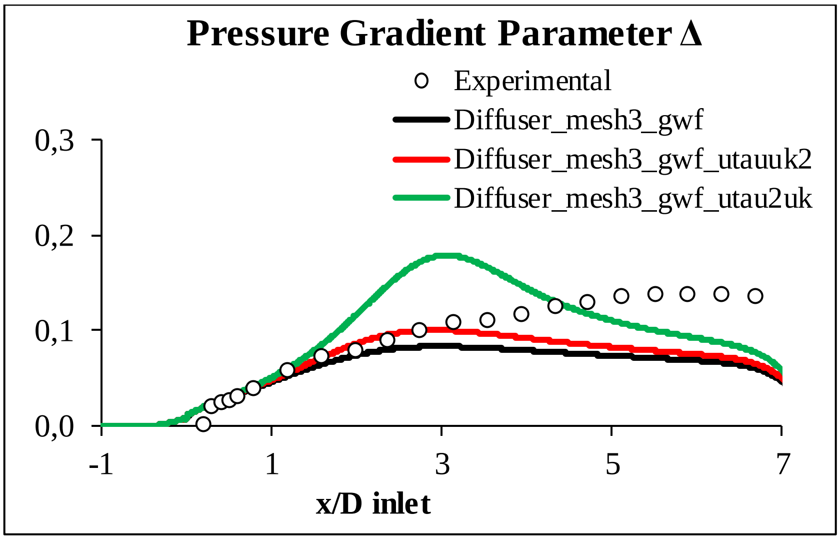

Figure 7.61. Pressure gradient parameter: GWF velocity scale sensitivity

From the analysis of the only friction velocity chart (Figure 7.59) we couldn't decide which one of the three was the GWF which gives the best results, since no one gives a perfect accordance with experimental data and no one gives a trend completely wrong and different from the experimental one. Analyzing the pressure gradient parameter chart (Figure 7.61) it can be noticed *Diffuser_mesh3_gwf_utau2uk* provides a trend of Δ which is different from experimental one. To be more precise, the experimental data are always increasing (except for a narrow region close to the outlet) while the profile for *Diffuser_mesh3_gwf_utau2uk* has a maximum (which is significantly higher than the experimental one) placed around 3 feeding pipe diameters from the inlet of the diffuser; then it decreases steeply (around 5 feeding pipe diameters *Diffuser_mesh3_gwf_utau2uk* underestimates the experimental profile. *Diffuser_mesh3_gwf_utauuk2*, unlike *Diffuser_mesh3_gwf_utau2uk*, shows an improvement concerning the profile of Δ. It has a trend similar to *Diffuser_mesh3_gwf* (and also to *Diffuser_mesh3*) but the accordance with experimental data is verified for a wider range (until 3 feeding pipe diameters from the inlet). For higher values there is an underestimation of the pressure gradient parameter, but the curve is still higher compared to the curve obtained with *Diffuser_mesh3_gwf*. We expect to get to the same conclusion by the observation of the dimensionless profiles.

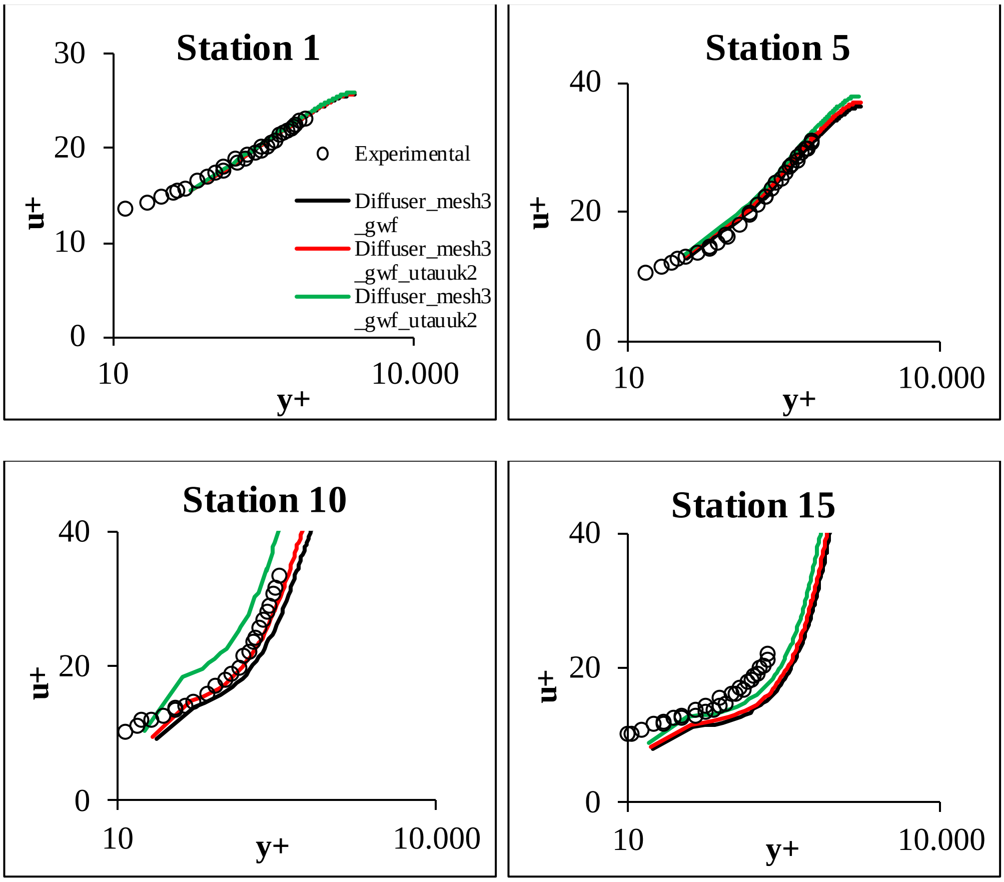

Figure 7.62. Dimensionless velocity profiles at stations 1, 5, 10 and 15: GWF velocity scale sensitivity analysis

Figure 7.63 shows the charts of the turbulent quantities (which are turbulent kinetic energy, production of turbulent kinetic energy and turbulent dissipation rate) at near wall center cells obtained for the cases *Diffuser_mesh3*, *Diffuser_mesh3_gwf_utauuk2* and *Diffuser_mesh3_gwf_utau2uk*. Following observations can be gotten from the analysis of these charts. When moving from *Diffuser_mesh3* to *Diffuser_mesh3_gwf_utau2uk*, turbulent kinetic energy decreases, and the decrement can be explained by the fact that the production of it decreases significantly because it is directly proportional to function Ψ. Also turbulent dissipation rate decreases even if the function Ψ is at the denominator. However this decrement is not high enough to make turbulent kinetic energy increase. Last observation: the decrement of turbulent kinetic energy is another cause of the decrement of the predicted values of wall shear stress all along the wall of the diffuser.

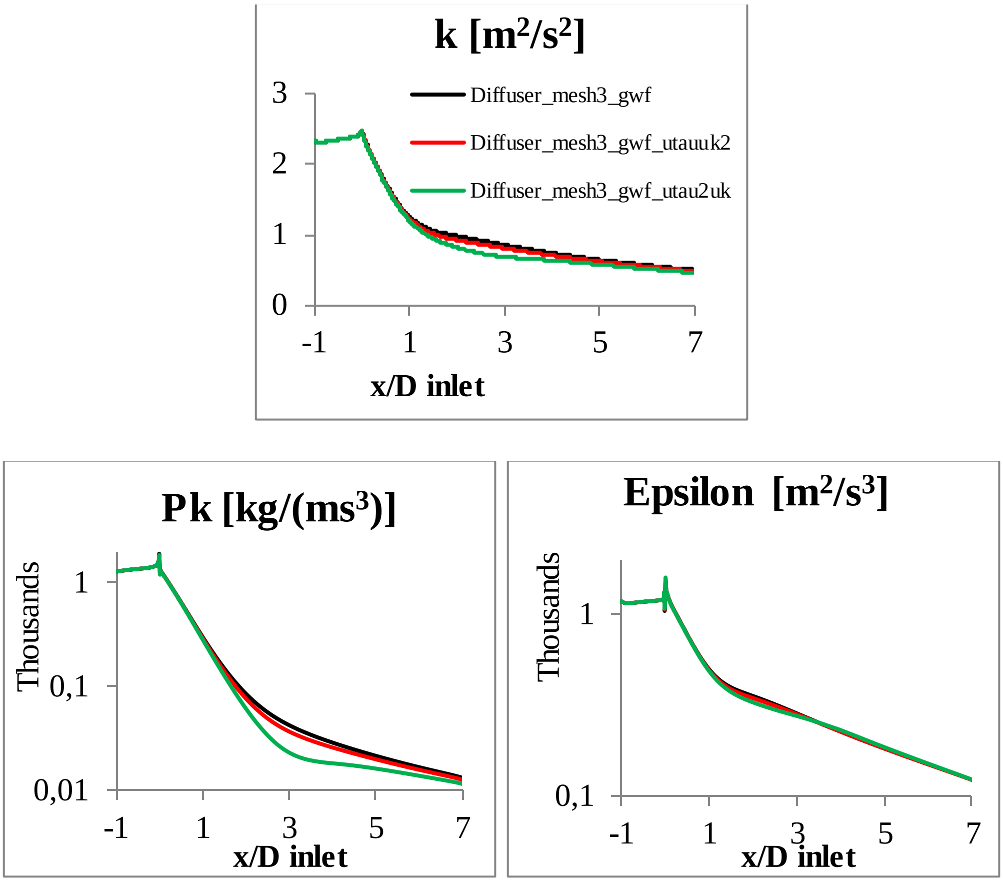

Figure 7.63. Turbulent quantities at the wall: GWF velocity scale sensitivity

The results obtained with *Diffuser_mesh3*, *Diffuser_mesh3_gwf_utauuk2* and *Diffuser_mesh3_gwf_utau2uk* can be summarized as follows:

- Wall tangential velocity at near wall cell is quite correctly estimated (underestimation for station 1 and correct estimation of stations 5, 10 and 15); however, the influence of Wall Functions on this quantity has been found out to be negligible
- The values of Ψ obtained have a physical meaning if we compare the experimental slopes of the log law with the slopes of the different GWF log law; first of all the slopes of GWF change with axial position (while the slope of SWF log law is always constant); second, their values tend to be closer to the experimental ones
- Values of wall shear stress and of pressure gradient parameter Δ tend to be closer to experimental data; *Diffuser_mesh3_gwf_utauuk2* seems to provide results closer to experimental data for both the quantities just mentioned. The error related to the overestimation of wall shear stress

can be caused by a not perfect estimation of turbulent kinetic energy inside near wall center cell or by the values of slope of GWF log law which are not in perfect accordance with experimental values. However, it is not possible to state which GWF provides the best accordance with experimental data.

Conclusions

Purpose of this master thesis work is the evaluation of the goodness of the results obtained with Generalized Wall Functions. These Wall Functions have been written through a user defined function and then this udf has been interpreted into Fluent 6.3, in order to make GWF available for wall treatment inside the code.

First of all, the characteristics of wall bounded turbulent flows, and the challenges in the treatment of near wall region, have been pointed out. Then a brief description of the k-ε family of turbulence models have been presented, just in order to make the reader aware of the general background of turbulence modeling and to make him aware of the turbulence models adopted in this master thesis work. After doing that, the work has moved on a study of the theoretical approach of Wall Functions for the wall treatment of turbulent flows. The general approach which is used by every CFD code when Wall Functions are set as wall treatment has been described. Then, the assumptions behind the two Wall Functions available in Fluent 6.3 (Standard Wall Functions and Non-Equilibrium Wall Functions) have been presented.

Next three chapters have been dedicated to the writing of the udf. To be more precise one chapter is dedicated to the logical approach to be used when writing a udf which concerns the implementation of a user defined wall function. Next chapter is dedicated to the deep study of the macro provided by Fluent 6.3 for the writing of user defined Wall Functions. This macro is called DEFINE_WALL_FUNCTIONS. This work has been carried out because there is no available online documentation about the macro except for a brief description of it inside Fluent User Defined manual. Also an example related to the implementation of Standard Wall Functions inside Fluent through this Macro is presented. The purpose of this chapter is to evaluate, starting from these few information, the complete working of the macro, in terms of variables passed from the solver to it, quantities that the user must provide, macro output and the way adopted by Fluent for the use of this output. Topic of next chapter is the description of a computation method for the non-equilibrium term C_U, which appears inside the GWF log law of the wall. Indeed, this term comes from the writing of wall tangential component of momentum equation; if the wall is not oriented as one of the xy coordinate axis, there is the need to obtain proper values starting from the data available from Fluent 6.3 post-processing. The work made in these three chapters has been useful for the writing of the complete udf for the implementation of GWF.

After that, an axial symmetric conical diffuser has been studied in order to evaluate the performance of GWF. Experimental data by Trupp et al. have been used for the comparison of CFD results. The results obtained are here summarized. Wall shear stress has been taken as the main target parameter to be verified. It has been found out that it depends on: wall tangential velocity and turbulent kinetic energy at near wall center cells, and on the law of the wall assumed by the Wall Functions adopted.
The first simulation has been carried out using a boundary layer mesh which guarantees dimensionless wall distance of near wall center cells to fall into 30 and 300 (which is the range of validity of SWF log law suggested by Fluent 6.3 User Guide). Standard k-epsilon model and SWF have been adopted for this simulation. Wall shear stress has been found out to be overestimated all along the wall. This happens because wall tangential velocity at near wall center cell is overestimated. In the first and last part of the diffuser the error is lower, and we can guess that there are some compensation effects. Moreover the law of the wall assumed with SWF is wrong because near wall center cells fall out of experimental log law region.

Subsequently a boundary layer mesh sensitivity has been carried out. New boundary layer meshes have been created so that the dimensionless wall distances of near wall center cells should fall into experimental log law region. The same turbulence model and Wall Functions (SWF) have been adopted. It has been found out that the mesh that respects this constraint provides wall tangential velocity at near wall cells in good accordance with experimental data. Therefore the error of wall shear stress (which is still overestimated) is due to the wrong slope of the law of the wall and to possible wrong turbulent kinetic energy at near wall center cells (unluckily no quantities regarding turbulence have been measured by the authors of the experiment).

After that, a core mesh sensitivity analysis has been made. No sensible effects of core mesh on the results have been found. GCI procedure has been performed in order to obtain an uncertainty range (related to the grid discretization error) for the wall shear stress.

Next step regarded the use of GWF for the wall treatment. The values of the slope of the GWF log law are found to be closer to experimental values while wall tangential velocity at near wall center cell is found not to change significantly with the Wall Functions adopted. The values of wall shear stress obtained are closer to experimental data, even if the accordance is not still perfect. At least, results are better than the ones obtained with SWF and with NEWF (last ones underestimate too much wall shear stress in the first half of the diffuser). The residual error of wall shear stress for GWF is due to the non-perfect accordance between GWF slope of the law of the wall and experimental slopes, and we can guess also to wrong turbulent kinetic energy in the region close to the wall.

Challenges regarding the convergence of the calculation leaded us to perform a sensitivity analysis for the velocity scale used inside the GWF law of the wall. Values of wall shear stress are improved with the other velocity scales adopted, even if no one gives a perfect accordance with experimental data.

For the characteristics exposed in this work, GWF are a promising way to deal with wall treatment, and their validity field is wider than the one of SWF and NEWF. It is good to point out that GWF are not an improvement of Low Reynolds models at all. Briefly, the fact that Low Reynolds models require a fine near wall mesh is a direct consequence of increased computational times. This is the reason that makes the use of Wall Functions acceptable, even though the results obtained with Wall Functions are known to be poorer compared to Low Reynolds. Between the Wall Functions available, SWF and NEWF are two stable and widespread alternatives and these are already implemented into Fluent 6.3. However, while SWF propose a law of the wall which is always constant and NEWF propose a law of the wall which takes into account only pressure gradient, GWF propose a law of the wall that should take into account all the non-equilibrium effects of the flow which is studied. Therefore this law of the wall is not constant but it is sensitized to the different conditions of the flow which is the subject of the simulation. This is the promising aspect of GWF, along with the fact that for the case of the axial symmetric conical diffuser GWF provide an improved solution compared to SWF and NEWF.

However, there’s still much work to do to improve them. First of all a comparison of turbulent characteristics at near wall cells with experimental data must be done in order to discover if the way adopted by GWF to compute production of turbulent kinetic energy and turbulent dissipation rate (and consequently turbulent kinetic energy) can be considered acceptable. Then, GWF must be tested for other benchmarks; indeed GWF must provide better results compared to SWF and NEWF for a wide range of situations, in order to become a reliable and stable approach for the wall treatment in CFD. For example, one benchmark which GWF performance can be tested for is a backward facing step [30].

Appendix A – Udf for GWF Implementation

```
/*****************************************************************
**

Udf for the implementation of Generalized Wall Functions - case study: conical
diffuser of Trupp et al.

Definition of DEFINE_ADJUST → calculation & storage of CUw* values

Definition of DEFINE_WALL_FUNCTIONS

*****************************************************************
**/

#include "udf.h"


/* Constant definition */

#define CMU 0.09
#define VKC 0.4187


DEFINE_ADJUST(define_cuw,d)

{

                Thread *tf1, *tc1;
                face_t f1;
                cell_t c1;

                real A[ND_ND], P[ND_ND], wallshear[ND_ND];
                real AMag, PMag;

        /* Definition of two unit vectors that define the new coordinate system
        (rotation of standard coordinate system) & of two vectors used to store
        faces & cells position */

                real Z[ND_ND], W[ND_ND], L[ND_ND], B[ND_ND];
                double delta, alfa, beta, gamma;

                real mu, rho, k, U, V, modV, DUDX, DUDY, DVDX, DVDY,
                DPDX, DPDY;
                real Uw, Uz, DPDW, DUwDX, DUwDY, DUwDW, DUwDZ;
                real uk, utau, ustar, y, ystar, psi;
                real CUW;
                real adim;

                int zone_id1; /* Wall in the geometry */

                zone_id1 = 4;

                tf1 = Lookup_Thread(d, zone_id1);

begin_f_loop(f1, tf1)

        {

                c1 = F_C0(f1,tf1);
                tc1 = THREAD_T0(tf1);

                F_CENTROID(L,f1,tf1);
                C_CENTROID(B,c1,tc1);

                F_AREA(A,f1,tf1);
                AMag=NV_MAG(A);

                NV_V(wallshear,=,F_STORAGE_R_N3V(f1,tf1,SV_WALL_S
                HEAR));

                P[0]=wallshear[0]/AMag;
                P[1]=wallshear[1]/AMag;

                PMag=NV_MAG(P);

                Z[0]=A[0]/AMag;
                Z[1]=A[1]/AMag;

                /* Clockwise rotation of pi/2 of the vector - see the geometry */

                W[0]=Z[1];
                W[1]=-Z[0];
```

```
          delta = acos(Z[0]); /* Angle between horizontal and Z */
          alfa = asin(W[1]); /* Angle between horizontal and W -
          Inclination of the face */

          U = C_U(c1,tc1);
          V = C_V(c1,tc1);
          DPDX = C_P_G(c1,tc1)[0];
          DPDY = C_P_G(c1,tc1)[1];
          DUDX = C_DUDX(c1,tc1);
          DUDY = C_DUDY(c1,tc1);
          DVDX = C_DVDX(c1,tc1);
          DVDY = C_DVDY(c1,tc1);

          beta = atan(V/U); /* Angle between horizontal and velocity
vector */
          gamma = beta - alfa; /* Angle between wall and velocity vector
*/

          modV = sqrt(pow(U,2)+pow(V,2));

          Uw = modV * cos(gamma);
          Uz = modV * sin(gamma);

          DPDW = DPDX * W[0] + DPDY * W[1]; /* From gradient
          theorem */

          DUwDX = DUDX * cos(alfa) + DVDX * sin(alfa); /* From
          decomposition of U and V in W direction */
          DUwDY = DUDY * cos(alfa) + DVDY * sin(alfa);

          DUwDW = DUwDX * W[0] + DUwDY * W[1]; /* From
          gradient theorem */
          DUwDZ = DUwDX * Z[0] + DUwDY * Z[1];

          k = C_K(c1,tc1);
          rho = C_R(c1,tc1);
          mu = C_MU_L(c1,tc1);

          uk = pow(CMU,0.25)* pow(k,0.5);

          utau = sqrt(PMag/rho);
```

```
                CUW = (rho * Uw * DUwDW ) + (rho * Uz * DUwDZ) + (
                DPDW )

                adim = mu/(pow(rho,2)*pow(uk,3)); /* Adimensionalization term
*/

                ustar = Uw*uk/pow(utau,2);

                y = 0.00486; /* This value represents an approximated average
                value of wall center cell wall distance - Must be properly
                substituted with C_WALL_DIST(c1,tc1); but it requires the udf
                to be compiled and not interpreted */

                ystar = y * rho * uk / mu;

                psi = 1- CUW*adim*ystar/(VKC*ustar);

                C_UDMI(c1,tc1,0) = psi;

        }

end_f_loop(f1, tf1)

}


DEFINE_WALL_FUNCTIONS(generalized, f, t, c0, t0, wf_ret, yPlus, Emod)

{

real wf_value;

real M[ND_ND];
real Q[ND_ND];
int a, b;

F_CENTROID(M,f,t);
C_CENTROID(Q,c0,t0);
a=THREAD_ID(t0);
b=THREAD_ID(t);

switch (wf_ret)
```

```
{
            case UPLUS_LAM:
                  wf_value  = yPlus;
            break;
            case UPLUS_TRB:
                  wf_value  = log(Emod*yPlus)/(VKC*C_UDMI(c0,t0,0));
            break;
            case DUPLUS_LAM:
                  wf_value  = 1.0;
            break;
            case DUPLUS_TRB:
                  wf_value  = 1./(VKC*yPlus*C_UDMI(c0,t0,0));
            break;
            case D2UPLUS_TRB:
                  wf_value  = -1./(VKC*yPlus*yPlus*C_UDMI(c0,t0,0));
            break;
default:
            printf("Wall function return value unavailable\n");
}
return wf_value;
}
```

Nomenclature and List of Acronyms

RANS	Reynolds Averaged Navier-Stokes
CFD	Computational Fluid Dynamics
SWF	Standard Wall Functions
NEWF	Non-Equilibrium Wall Functions
GWF	Generalized Wall Functions
AWF	Analytical Wall Functions
k	Turbulent kinetic energy
y^+	Dimensionless wall distance
ε	Turbulent dissipation rate
ε^*	Isotropic turbulent dissipation rate
Re	Reynolds number
$\mathrm{Re_y}$	Local Reynolds number
ρ	Density
v	Core flow velocity
L	Characteristic length of the case
μ	Dynamic viscosity
y	Wall distance
$\mathbf{V}$	Time averaged velocity vector
$\mathbf{w}$	Unit vector parallel to the wall
V_{wall}	Velocity of the wall
$\mathbf{n}$	Unit vector perpendicular to the wall
$\cdot$	Scalar product between two vectors
τ_w	Wall shear stress
u_τ	Friction velocity
u^+	Dimensionless wall tangential velocity
$\tau(y)$	Shear stress evaluated at the wall distance y
A	Constant of the log law
B	Constant of the log law
δ	Boundary layer thickness
$\mathrm{U_{max}}$	Free-stream (or axial) wall tangential velocity
URANS	Unsteady Reynolds Averaged Navier-Stokes
$\nabla \cdot$	Divergence
$\mathbf{v}$	Instantaneous velocity vector
p	Instantaneous pressure
$\boldsymbol{\tau}$	Instantaneous stress tensor
$\nabla \boldsymbol{v}$	Gradient of instantaneous velocity vector
$\nabla \boldsymbol{v}^T$	Transpose of gradient of instantaneous velocity vector

$\boldsymbol{I}$	Identity matrix
$< \phi >$	Time averaged quantity ϕ
$\boldsymbol{V}$	Time averaged velocity vector
P	Time averaged pressure
$\boldsymbol{T}$	Time averaged stress tensor
$\boldsymbol{R}$	Reynolds stress tensor
u’, v’, w’	Turbulent velocity fluctuations in the three directions
μ_t	Turbulent viscosity
$\mathbf{S}$	Mean strain rate tensor
δ_{ij}	Kronecker delta
C_μ	Constant of the expression for μ_t
P_k	Production of turbulent kinetic energy [$MT^{-3}L^{-1}$]
σ_k, σ_ε	Turbulent Prandtl numbers
$C_{1\varepsilon}$, $C_{2\varepsilon}$	Constants of k-ε turbulence models
S	$\sqrt{2\boldsymbol{S}_{ij}\boldsymbol{S}_{ij}}$
α_k, α_ε	Inverse of turbulent Prandtl numbers
μ_{eff}	Effective viscosity (RNG model)
R_ε	Additional source term for ε equation (RNG model)
v	Cinematic viscosity
a_P, a_E, a_W, a_N, a_S	Coefficients of the discretized wall tangential momentum equation
U_P	Wall tangential velocity at center cell P
U_W, U_E, U_S, U_N	Wall tangential velocity at the neighboring cells of cell P
S_U	Source term in the discretized wall tangential momentum equation
Δx	Thickness of the cell in the x direction
u	Equivalent to U
E	Constant of log law
κ	Von Karman constant
c_l, C_l^*	Length scale used for the computation of ε
u_k	Alternative velocity scale for Wall Functions
u^*	Alternative dimensionless wall tangential velocity
y^*	Alternative dimensionless wall distance
y_v	Thickness of sub-viscous layer
y_n	Thickness of near wall cell
$\overline{P_k}$	Production of turbulent kinetic energy averaged on the whole near wall cell
$\bar{\varepsilon}$	Turbulent dissipation rate averaged on the whole near wall cell
y_v^*	Dimensionless thickness of sub-viscous layer
$\hat{U}$	Modified wall tangential velocity of NEWF log law

C_U	Non-equilibrium terms (GWF)
Ψ	Non-equilibrium function (GWF)
C_U^+	Dimensionless value of C_U
A_{face}	Area of the cell face
C_f	Friction factor
z	Wall perpendicular direction
C_{Uw}	C_U defined for a wz coordinate system
U_w	Wall tangential velocity
U_z	Wall perpendicular velocity
α	Angle between x and wall
β	Angle between x and velocity vector
γ	Angle between wall and velocity vector
U_b	Bulk velocity
$f(y^*)$	Dimensionless law of the wall
$\overline{\tau_w}$	Wall shear stress averaged on the whole wall
h	Grid spacing
C_U^*	Alternative dimensionless value of C_U
$adim$	Adimensionalization term for the term C_U

Bibliography and References

[1] A.C. Trupp, R.S. Azad and S.Z. Kassab, "Near-wall velocity distributions within a straight conical diffuser," *Experiments in Fluids*, vol. 4, p. 319.331, 1986.

[2] H.K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics. The finite volume method.*: Longman Scientific & Technical, 1995.

[3] [Online]. http://www.cfd-online.com/Wiki/Introduction_to_CFD

[4] J.P.A.J. van Beeck & C. Benocci, Introduction to Turbulence Modeling, 2004.

[5] E. Colombo, Laboratorio progettuale di CFD - La modellazione della turbolenza, 2010-2011.

[6] G. Comini, G. Croce, E. Nobile, *Fondamenti di termofluidodinamica computazionale.*: S.G.E., 2008.

[7] [Online]. http://www.kxcad.net/STAR-CCM/online/139-kEpsilonTurbulence-28.html

[8] B.E. Launder and D.B. Spalding, "The Numerical Computation of Tubulent Flows," *Computer Methods in Applied Mechanics and Engineering* , vol. 3, pp. 269-289, 1974.

[9] S.-E. Kim, D. Choudhury, "A Near-Wall Treatment Using Wall Functions Sensitized to Pressure Gradient," *ASME FED, Separated and Complex Flow*, vol. 217, 1995.

[10] M. Popovac, K. Hanjalic, "Compound Wall Treatment for RANS Computation of Complex Turbulent Flows and Heat Transfer," *Flow Turbulence Combust*, vol. 78, pp. 177-202, 2007.

[11] T.J. Craft, A.V. Gerasimov, H. Iacovides, B.E. Launder, "Progress in the generalization of wall-function treatments," *International Journal of Heat and Fluid Flow*, vol. 23, p. 148.160, 2002.

[12] E. Karunakaran & V. Ganesan, "Mean flow field measurements in an axisymmetric conical diffuser with and without inlet flow distortion," *Indian Journal of Engineering & Material Sciences*, vol. 16, pp. 211-219, Aug 2009.

[13] C. Osnaghi, *Teoria delle Turbomacchine.*: Società Editrice Esculapio s.r.l., 2006.

[14] [Online]. http://www.eng.wayne.edu/legacy/forms/4/Buckinghamforlect1.pdf

[15] Ansys , FLUENT 6.3 User's Guide, Modeling Turbulence - Near-Wall Treatmens for Wall Bounded Turbulent Flows - Standard Wall Functions.

[16] H.K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics. The finite volume method.*: Longman Scientific & Techinal, 1995.

[17] [Online]. http://www.stanford.edu/class/me469b/handouts/turbulence.pdf

[18] J.O. Hinze, *Turbulence*. New York: McGraw-Hill, 1975.

[19] B.E. Launder and D.B. Spalding, Lectures in Mathematical Models of Turbulence, 1972, Academic Press, London, England.

[20] Ansys, FLUENT 6.3 User's Guide - Modeling Turbulence - Standard, RNG and Realizable k-epsilon Models Theory - Standard k-epsilon model.

[21] D. Choudhury, Introduction to Renormalization Group Method and Turbulence Modeling, 1993, Fluent Inc. Technical Memorandum TM-107.

[22] Ansys, FLUENT 6.3 User's Guide - Modeling Turbulence - Standard, RNG and Realizable k-epsilon Models Theory - RNG k-epsilon Model.

[23] T.-H. Shih, W.W. Liou, A. Shabbir, Z. Yang, J. Zu, "A New k-epsilon Eddy-Viscosity Model for High Reynolds Number Turbulent Flows - Model Development and Validation," *Computers Fluids*, vol. 24, pp. 227-238, 1995.

[24] Ansys, FLUENT 6.3 User's Guide - Modeling Turbulence - Standard, RNG and Realizable k-epsilon Models Theory - Realizable k-epsilon Model.

[25] Ansys, FLUENT 6.3 User's Guide - Modeling turbulence - Near Wall Treatments for Wall-Bounded Turbulent Flows - Non-Equilibrium Wall Functions.

[26] Ansys, FLUENT 6.3 UDF Manual.

[27] P.A. Okwuobi, A.C. Azad, "Turbulence in a conical diffuser with fully developed flow at entry," *Journal of Fluid Mechanics*, vol. 57, pp. 603-622.

[28] [Online]. http://www.engineeringtoolbox.com/entrance-length-flow-d_615.html

[29] P.J. Roache, "Verification of Codes and Calculations," in *26th AIAA Fluid Dynamics Conference*, San Diego, California, USA, 1995, pp. AIAA Paper 95-2224.

[30] D.M. Driver and H.L. Seegmiller, "Features of Reattaching Turbulent Shear Layer in Divergent Channel Flows," *AIAA Journal*, vol. 23, pp. 163-171, February 1985.

[31] H.K. Versteeg and W. Malalasekera, *An introduction to computational fluid*

dynamics. the finite volume method.: Longman Scientific & Technical, 1995.

[32] J.H. Ferziger, M. Peric, *Computational Methods for Fluid Dynamics. 3rd edition.*: Springer, 2002.

[33] D.C. Wilcox, *Turbulence Modeling for CFD.*: DCW Industries, Inc., 1994.

[34] E. Colombo, *Laboratorio Progettuale di CFD.*, 2011.

[35] B. Launder, "Computer Modelling / Simulations for industrial and environmental engineering," in *FAGGIOLATI PUMPS spa - Workshop / Industrial Short Course - Confindustria Macerata*, Macerata, 2007.

[36] K. Hanjalic, "Advanced Wall Functions and Compound Wall Treatment," in *Turbulence Modelling and Simulation Pt 2*, Darmstadt, 2005/2006.

[37] S. Socolofsky, Law of the Wall and the Velocity Defect Law, OCEN 678 - Fluid Dynamics for Ocean and Environmental Engineering.

[38] T.J. Craft, "Wall Functions," in *School of Mechanical Aerospace and Civil Engineering - TPFE MSc Advanced Turbulence Modelling*, Manchester, 2011.

[39] E.Y.K. Ng, H.Y. Tan, H.N. Lim, D. Choi, "Near-wall function for turbulence closure model," *Computational Mechanics*, vol. 29, pp. 178-181, 2002.

[40] H. Iacovides, Current Practice and Recent Developments in Wall Functions I (Conventional Approaches), Department of Mechanical Aerospace and Manifacturing Engineering UMIST.

[41] K. Hanjalic, Turbulence Modelling for industrial appllications: some developments and prospects: Second-moment closure versus Eddy-viscosity models.

[42] K. Hanjalic, Turbulence Modelling for industrial applications: some developments and prospects - SMC-based simpler models (ASM/AFM, EVM/EDM) and treatments of wall boundary conditions.

[43] M. Bramanti, C.D. Pagani, S. Salsa, *Matematica. Calcolo infinitesimale e algebra lineare.*: Zanichelli.

[44] R.S. Azad, R.H. Hummel, "Measurement of intermittency factor in diffuser flow," *J. Physics*, vol. 49, pp. 2917-2930.

[45] N. Kivekas, I. Riipinen, H. Vuollekoski, Near-Wall Treatmens for Wall-Bounded Turbulent flows.

[46] [Online]. http://www.stanford.edu/class/me469b/handouts/turbulence.pdf