ISRN UTH-INGUTB-EX- KKI-2011/08-SE

Examensarbete 15 hp September 2011

Analys av turbulensmodeller för CFD

Johan Erlandsson Patrik Berg



Teknisk- naturvetenskaplig fakultet UTH-enheten

Besöksadress: Ångströmlaboratoriet Lägerhyddsvägen 1 Hus 4, Plan 0

Postadress: Box 536 751 21 Uppsala

Telefon: 018 - 471 30 03

Telefax: 018 - 471 30 00

Hemsida: http://www.teknat.uu.se/student

Abstract

Analys av turbulensmodeller för CFD

Analysis of turbulens models for CFD

Patrik Berg, Johan Erlandsson

This thesis has been a part of Forsmarks Kraftgrupp AB's evaluation of a turbulence model used in simulation of turbulent flow called PRNS (Partially Resolved Numerical Simulation). This model has promising properties and may be of use in saving computational resources. The purpose of this thesis was to analyze this model and compare it with industrially applied models such as k-omega SST and LES (Large Eddy Simulations).

PRNS works as a hybrid of the k-omega SST and DNS (Direct Numerical Simulation) where a constant, RCP (Resolution Control Parameter) with a value between 0 and 1 are selected. This constant is then used in the calculations and determines the behavior of the simulation. When RCP is set to zero the equation are the same as for a DNS simulation and when RCP is set to one the equations for k-omega SST is solved. In this report four different PRNS models have been used, three where RCP was given a constant value (0.1, 0.4 and 0.6). In the fourth model RCP is calculated from the flow field variables

The models have been compared to an experiment from 2008 and simulations have been made to resemble the experiment. In the experiment a Particle Image Velocimeter (PIV) was used as method of measurement. From the experimental report data such as velocity (U), turbulent kinetic energy (k) and standard deviation (URMS) have been obtained and have formed the basis for comparison. The models have been simulated in two different software programs: OpenFOAM and Fluent. The data have thereafter been post processed in the software programs MatLab and ParaView, to be compared with experimental data.

The results of the simulations have shown that PRNS models generally show a good accordance with experimental data. In particular, PRNS models with constant RCP have shown good results, however, there are some discrepancies. The PRNS model with varying RCP has in most cases showed the largest deviation from experimental data but also a deviation from the other models, including the reference models. Due to the design of the mesh (coarse) further evaluation of the PRNS models will be needed. First, simulate with a finer mesh, but also more complex geometries should be simulated in order to sort out PRNS strengths and weaknesses and thus determine if the model can be used in the daily work at Forsmarks Kraftgrupp AB.

Handledare: Nicolas Forsberg Ämnesgranskare: Michael Österlund Examinator: Michael Österlund ISRN UTH-INGUTB-EX- KKI-2011/08-SE

Sammanfattning

Detta examensarbete har ingått som ett delsteg i Forsmarks Kraftgrupp AB:s utvärdering av en turbulensmodell som används vid simulering av turbulentflöde kallad Partially Resolved Numerical Simulations (PRNS). Denna modell har lovande egenskaper och förhoppningen är att beräkningsresurser kan sparas med denna modell. Syftet med detta examensarbete har varit att analysera denna modell och jämföra den med industriellt tillämpade modeller så som k- ω SST och LES (Large Eddy Simulations).

PRNS fungerar som en hybrid mellan k- ω SST och DNS där en konstant (kallad RCP) med ett värde mellan 0 och 1 väljs. Denna konstant används sedan i beräkningarna och bestämmer hur noggrann beräkningen blir. Är RCP = 0 betyder det att ekvationerna blir lika som när en DNS löses och om RCP = 1 löses ekvationerna för k- ω SST. I denna rapport har fyra olika PRNS modeller använts, tre med konstant värde på RCP (0.6, 0.4 och 0.1) samt en modell där RCP har varierats under simuleringens gång för att alltid ha ett optimalt värde.

Modellerna har jämförts mot ett experiment från 2008 där simuleringarna har gjorts för att efterlikna experimentet. Vid experimentet användes en Particle Image Velocimeter (PIV) som mätmetod. Från experimentets rapport har data om hastighet (U), turbulent kinetisk energi (k) och standardavvikelse (U_{RMS}) erhållits vilka har legat till grund för jämförelsen. Modellerna har simulerats i två olika programvaror, OpenFOAM och Fluent. Datan har efterbehandlats i programmen MatLab och paraView för att kunna jämföras med experimentets data.

Resultatet av simuleringarna har visat att PRNS modellerna överlag visar på en god överrännsstämmelse med experimentets data. Särskilt de PRNS modeller med konstant RCP har visat gott resultat, dock finns vissa avvikelser. PRNS modellen med varierande RCP har i de allra flesta fall visat på den största avvikelsen från experimentdata men också på en avvikelse från de andra modellerna, inklusive referensmodellerna.

Pga. utformningen på meshen (relativt grov) krävs dock fortsatt utvärdering av PRNS modellerna. Dels simulera med en finare mesh men även mer komplexa geometrier bör simuleras för att kunna sortera ut PRNS styrkor och svagheter och på så sätt fastställa om modellen kan användas i det dagliga arbetet på Forsmarks Kraftgrupp AB.

Nyckelord

Forsmarks kraftgrupp AB, turbulensmodeller, CFD, PRNS, LES, RANS, Fluent, OpenFOAM

Förord

Detta examensarbete är genomfört på Forsmarks Kraftgrupp AB under avdelningen FTMT och är det avslutande momentet i vår utbildning till högskoleingenjör i kärnkraftteknik vid Uppsala universitet. Examensarbetet omfattar tio veckors arbete och pågick mellan april och juni 2011.

Vi vill särskilt tacka vår handledare på FKA, Nicolas Forsberg men även Hernan Tinoco och Hans Lindqvist för deras stöd och expertis under vårt arbete.

Ett tack går också till våran ämnesgranskare Michael Österlund.

Uppsala, juni 2011

Patrik Berg Johan Erlandsson

Innehållsförteckning

1	INLEDNING	1
1	L.1 BAKGRUND	1
1	L.2 Syfte och mål	1
1	1.3 Metod	2
	1.3.1 <i>OpenFOAM</i>	3
	1.3.2 Fluent	3
1	1.4 Avgränsningar	4
1	L.5 FÖRETAGSPRESENTATION	4
2	TEORI	5
2	2.1 TURBULENS	5
2	2.2 Mesh	7
2	2.3 NAVIER-STOKES EKVATIONER	8
2	2.4 ENERGISPEKTRA	9
2	2.5 DIRECT NUMERICAL SIMULATION (DNS)	
2	2.6 REYNOLD-AVERAGE NAVIER-STOKES (RANS)	11
2	2.7 TURBULENSMODELLER	13
	2.7.1 k-ω SST	
	2.7.2 Large Eddy Simulation (LES)	14
	2.7.3 Partially Resolved Numerical Simulation (PRNS)	
2	2.8 VAL AV TURBULENSMODELL	
2	2.9 COURANTS TAL	
3	UTFÖRANDE	
3	3.1 CFD-simuleringar	
3	3.2 Referensexperiment	20

4 RESUL	ГАТ	
4.1 CF	D-SIMULERING	
4.1.1	Hastighetsprofil OpenFOAM	
4.1.2	Standardavvikelse OpenFOAM	
4.1.3	Turbulent intensitet OpenFOAM	24
4.1.4	Hastighetsprofil Fluent	25
4.1.5	Standardavvikelse Fluent	
4.1.6	Turbulent intensitet Fluent	
4.2 Jän	IFÖRELSE AV OPENFOAM OCH FLUENT	27
4.2.1	Hastighetsprofil	27
4.2.2	Standardavvikelse	
4.3 An	ALYS	
4.3.1	Hastighet, standardavvikelse och turbulent intensitet	29
4.3.2	Villkor för RCP samt k och $oldsymbol{\omega}$	
4.3.3	LES	
4.3.4	Jämförelse mellan Fluent och OpenFOAM	
5 DISKUS	SION	
5.1 Fr.	AMTIDA ARBETE	
5.2 Fei	LKÄLLOR	
6 SLUTSA	\TS	
7 REFERI	ENSER	

Figurförteckning

Figur 2.1	Laminärt och turbulent flöde [5]	5
Figur 2.2	Mesh, geometrin är delad på mitten för att illustrera storleksskillnaden på cellerna	7
Figur 2.3	Energispektrum RANS till DNS [11]	9
Figur 2.4	Vad som löses respektive modelleras vid olika turbulensmodeller	10
Figur 2.5	Fluktuationer kring medelvärdet	11
Figur 2.6	Illustration av liknelsen med högpassfilter, stora svängningar löses och små modeleras	14
Figur 3.1	Fyrkantsrör med en Backward-Facing Step konfiguration, modifierad [3]	20
Figur 3.2	Meshen som används med cellstorlek i meter	21
Figur 4.1	Hastighetsprofil OpenFOAM (exklusive RCP0.1 och RCP0.6)	22
Figur 4.2	Standardavvikelse (U _{RMS}) OpenFOAM	23
Figur 4.3 Tu	rbulent intensitet (u´/U $_0$) för alla modeller	24
Figur 4.4 Ha	stighetsprofil Fluent (exklusive RCP0.1 och RCP0.6)	25
Figur 4.5 Sto	Indardavvikelse (URMS) Fluent	26
Figur 4.6 Ha	stighetsprofil OpenFOAM mot Fluent k- ω SST	27
Figur 4.7 Ha	stighetsprofil OpenFOAM mot Fluent RCP0.4	27
Figur 4.8 Sto	undardavvikelse OpenFOAM mot Fluent RCP0.4	28
Figur 4.9 Jäi	nförelse av vad RCP-värdet borde vara och vad det faktiskt är i RCP-filter	31
Figur 8.1.1 H	Ritning över hela förbrännaren (OBS spegelvänd gemtemot figur 8.1.2)[4]	39
Figur 8.1.2 (Geometrin som simulerades i fall 1	40
Figur 8.1.3 I	lastighetsprofil fall 1	41
Figur 8.1.4 S	Standardavvikelse (U _{RMS}) fall 1	42
Figur 8.3.5 I	lastighet i x-led för OpenFOAM och Fluent	47
Figur 8.3.6 k	för OpenFOAM och Fluent	48
Figur 8.3.7 d	o för OpenFOAM och Fluent	49
Figur 8.3.8 1	Furbulent viskositet för OpenFOAM och Fluent	50
Figur 8.3.9 1	Furbulent viskositet enl. Menter för OpenFOAM och Fluent	51
Figur 8.3.10	Gyllenram Nilssons RCP-villkor nr 2 för OpenFOAM och Fluent	52
Figur 8.3.11	NASA:s RCP-villkor för OpenFOAM och Fluent	53

Nomenklatur

- α = Filterkonstant (3)
- $\beta^* =$ Konstant (0,09)
- $\sqrt[3]{\Delta} = \delta x = \text{Cellängd [m]}$
- k = Turbulent kinetisk energi $[m^2/s^2]$
- ε = Turbulent dissipation [m²/s³]
- ω = Specifik dissipation [1/s]
- L = Längdskala [m]
- μ = Dynamisk viskositet [Pa*s]
- ν = Kinematisk viskositet [m²/s]
- v_t = turbulentviskositet
- $\rho = \text{Densitet } [\text{kg/m}^3]$
- Re = Reynolds tal
- $\delta t = \text{Tidssteg} [s]$
- U = hastighet (x-led) [m/s]
- U_{RMS} = Standardavvikelse [m/s]
- *u* = Hastighetsfluktuation [m/s]
- \bar{u} = Medelhastighet [m/s]
- $\kappa = v$ ågtal [1/m]
- C = Courants tal

1 Inledning

1.1 Bakgrund

CFD (Computational Fluid Dynamics) innebär analyser av system som innehåller beräkningar av vätskors och gasers flöden, men även värmeöverföring och kemiska reaktioner. Metodiken är väldigt kraftfull och användbar i många sammanhang, t.ex. vid konstruktion av flygplan, turbiner, motorer och rörsystem men även i meteorologi och oceanografi. De stora fördelarna med CFD är att det går att testa koncept och föreslå åtgärder då fullskaleexperiment skulle bli för kostsamma.[1]

I en CFD-simulering används Navier-Stokes ekvationer för att beskriva hur flödet beter sig. När ett turbulent flöde ska simuleras så exakt som möjligt används en DNS (Direct numerical simulation). Denna metod är dock mycket resurskrävande och är sällan användbar i industriella tillämpningar. För att minska resursåtgången har turbulensmodeller utvecklats som löser modifierade varianter av ekvationerna, men dessa är dock inte lika exakta. I dagsläget är vanliga grupper av modeller, LES (Large Eddy Simulation) och RANS (Reynold-Average Navier-Strokes). Dessa modeller har olika styrkor respektive svagheter vad gäller beräkningstid och noggrannhet. I takt med att datorers processorkapacitet hela tiden förbättras utvecklas och förbättras turbulensmodeller sändigt. En av de nyare modellerna som är en blandning av DNS och RANS kallas PRNS (Partially Resolved Numerical Simulations) och har visat sig vara en lovande modell. Dessvärre kräver PRNS vidareutveckling, framförallt avseende praktiska erfarenheter samt analyser gentemot andra turbulensmodeller.

1.2 Syfte och mål

Examensarbetet ingår som ett delsteg i Forsmarks utvärdering av PRNS. Ett fall som tydligt visar viktiga parametrar är en stegformad enkelsidig expansion (BFS backward-facing step). Ett experiment på ett BFS där tredimensionella turbulensmätningar gjord med en PIV (particle image velocimeter) har genomförts av en forskargrupp i Finland [2]. Dessa mätningar ligger till grund för PRNS simuleringarna i denna rapport. Arbetet innebär att inrätta, genomföra och utvärdera CFD-simuleringar gjorda med olika turbulensmodeller i två olika program, Fluent och OpenFOAM.

Målet med detta arbete är att ta reda på om de PRNS-modeller som används är tillräckligt noggranna för att kunna användas i det dagliga arbetet på FTMT. Om det visar sig att metoden är tillräckligt noggrann kan både tid och pengar sparas.

1.3 Metod

Två olika fall har studerats.

Fall 1: Pitz Daily; ett fall med turbulens som skapas via en "klack" som simuleras i två dimensioner (Dessa simuleringar kan ses i bilaga 1). [3]

Fall 2: BFS (Referensexperiment); Liknande geometri som i fall 1 men beräkningarna sker i tre dimensioner. [2]

Fallen simuleras med turbulensmodeller från tre olika grupper; PRNS, LES och RANS samt i två olika programvaror, OpenFOAM och Fluent 13.0.0. Mätdata från bestämda punkter samlas in och med data från dessa punkter kan hastighetsprofiler och standardavvikelsekurvor vid bestämda avstånd från origo skapas. För att jämföra de olika modellerna och experimenten med varandra har data sammanställas med specialskrivna programkoder i MatLab.

Kurvor som plottas mot experimentdata är tidsmedelvärdet av hastigheten

$$U = \bar{u} = \frac{1}{n} \sum_{i=1}^{n} u_i \tag{1.1}$$

där U är medelhastighet över tid och

$$U_{rms} = \sqrt{(u - \bar{u})^2},\tag{1.2}$$

där U_{rms} är standardavvikelsen som beskriver hur mycket hastigheten varierar mot medelhastigheten i en punkt. U_{rms} kan även tolkas som ett mått på turbulensintensitet. Värdena till ekvation (1.2) är hämtade från den sista sekunden av simuleringen för att undvika fluktuationer som beror på att startförhållandena skiljer sig från modellernas uppskattning av flödet.

Syftet med rapporten är att analysera och att:

- Jämföra hur bra turbulensmodellerna stämmer överens med referensexperimentet.
- Jämföra villkoren för hur RCP väljs samt.
- Jämföra hur Fluent och OpenFOAM skiljer sig åt.

1.3.1 OpenFOAM

OpenFOAM är en programvara för CFD-simuleringar med öppen källkod utan grafiskt gränssnitt. Programmet kan användas för simulering av flöden med hänsyn tagen till kemiska reaktioner, turbulens och värmeöverföring. OpenFOAM består i grunden av ett antal C++ moduler. Dessa används för att bygga lösare (i detta fall pisoFoam) och efterbehandlare (paraView).[4]

För att köra en simulering behövs tre kataloger med filer, "constant", "system" och "0". I "constant"-katalogen ingår modellens mesh (se kapitel 2.2) samt även ett antal filer som definierar hur simuleringen ska lösas. I "system"-katalogen ingår filer för bl.a. hur simuleringen skall köras, t.ex. hur länge, samt vad simuleringen ska visa (utskrift av data). I den tredje mappen "0" definieras rand- och initialvillkor, d.v.s. vilka värden simuleringen ska utgå ifrån vad gäller t.ex. hastighet.

Under simuleringens gång skriver programmet textfiler med data (hastighet och k-värde) från de punkter i modellen som definierats i systemkatalogen. Dessa efterbehandlas sedan i MatLab. Den inbyggda efterbehandlaren paraView kan sedan grafiskt illustrera vad som skett, se t.ex. figur 4.8.

1.3.2 Fluent

Fluent har till skillnad från OpenFOAM ett grafiskt gränssnitt och är på så sätt mer lättanvänt. Inställningarna görs genom att i olika menyer välja de alternativ som passar bäst eller genom att skriva in värden på parametrar där det finns möjlighet.

Arbetssättet går kortfattat ut på att en mesh läses in, sedan väljs den turbulensmodell som ska användas och de initialvärden som ska gälla för att specificera fallet. När beräkningarna är klara finns det olika verktyg för att bearbeta och visualisera resultatet.

Ibland räcker inte de inställningar som finns i programmet till. Då kan så kallade UDF:er (User Defined Function) användas. UDF är en C-kod som definierar olika funktioner som kan användas i Fluent. Detta är något som slutanvändaren får programmera själv. Koden kompileras och länkas dynamiskt till beräkningarna i Fluent. I detta projekt användes två funktioner, en som bestämmer värdet på PRNS-modellerna och en som skriver ut data (hastighet) i olika fördefinierade punkter till en textfil. Detta är praktiskt då dessa värden därefter enkelt kan läsas in i MatLab för analys av resultatet.[7]

1.4 Avgränsningar

Experimenten som används som referens är inte utförda inom denna rapports ramar och inte heller av rapportens författare. De meshar som används är färdiggenererade av personal på FTMT och har använts utan modifikation. Då den bakomliggande matematiken är väldigt komplicerad kommer ingen större vikt läggas på att förklara denna, endast de mest grundläggande ekvationerna kommer att behandlas.

1.5 Företagspresentation

Vattenfall som är storägare i Forsmarks Kraftgrupp AB (FKA) driver sju av Sveriges tio kärnkraftsreaktorer, tre i Forsmark och fyra i Ringhals, dessa producerar drygt 50 TWh varje år. Kontoret för Mekanisk Integritet (FTM) på FKA är ett specialistkontor och ingår i Forsmarks Teknikavdelning. FTM arbetar med och har ansvaret för anläggningarnas mekaniska integritet. Kontoret består av fyra avdelningar med ansvar för samordning av belastningsunderlag (FTMB), hållfasthet och strukturmekanik (FTMH) material, svets och vibrationer (FTMQ) samt termohydraulik (FTMT). På avdelningen FTMT (där detta examensarbete utfördes) tas underlag fram för att kunna genomföra t ex. hållfasthetsberäkningar. Dessa underlag består i att med hjälp av CFD-simuleringar bestämma hur ång/vattenflödet beter sig i anläggningen. Problem som kan behandlas är t ex. temperatur och tryckgradienter eller oförutbestämda händelser. När beteendet hos flödet är bestämt skickas datan vidare till relevant avdelning för fortsatt analys.

2 Teori

2.1 Turbulens

Ett flöde kan definieras som laminärt eller turbulent. För att få en förståelse för vad detta innebär är röken från en cigarett ett bra exempel (se figur 2.1). Från cigarettens glöd strömmar ett laminärt flöde av rök jämnt och rakt upp utan virvlar. Vid en viss punkt övergår röken till ett flöde fyllt av virvlar, turbulens uppstår. Turbulenta flöden kännetecknas av fluktuerande, oregelbundna tredimensionella rörelser. Turbulens inträffar oftast vid hög hastighet, stora friktionskrafter och då de viskösa krafterna är relativt små jämfört med de kinematiska krafterna. Turbulent flöde är kaotiskt (icke linjärt) vilket betyder att en liten störning kan medföra stora förändringar av flödet. Därför är det svårt att förutspå en exakt hastighet för flödet vid en bestämd tidpunkt. Detta gör att turbulent flöde är väldigt svårt att beräkna exakt och statistiska metoder måste användas.



Figur 2.1 Laminärt och turbulent flöde [5]

När turbulens uppstår (t.ex. om flödet träffar på ett hinder) bildas först stora virvlar som i sin tur "sönderfaller" till fler mindre virvlar (med bevarat totalt energiinnehåll), till slut blir de så små att deras inneboende rörelseenergi kommer att övergå till värme. Detta sker när de viskösa krafterna i vätskan blir större än virvelns inneboende kraft. När dessa två krafter är lika stora har virvlarna uppnått Kolmogorovstorlek, denna storlek är således den minsta en virvel kan ha innan den upplöses.[6]

För att uppskatta om ett flöde är turbulent eller laminärt (och även hur turbulent det är) kan Reynolds tal användas

$$Re = \frac{U*L*\rho}{\mu} = \frac{U*L}{\nu}.$$
(2.1)

Reynolds tal är dimensionslöst och kan ses som kvoten mellan de kinematiska krafterna och de viskösa krafterna. *U* är medelhastigheten på flödet (alternativt bulkhastighet). Vad *L* symboliserar beror på systemet, för en strömningskanal används ofta den hydrauliska diametern [m]. ρ är densiteten på fluiden, μ är den dynamiska viskositeten och ν är den kinematiska viskositeten ($\nu = \mu / \rho$).

Vid flöden i rör uppstår turbulens vanligtvis när Reynoldstalet överstiger ett värde någonstans mellan Re 2300 och Re 4000. Intervallet mellan 2300 och 4000 beror på ett antal faktorer som t.ex. rörets ytfriktion och flödets likformighet vilket gör det svårt att exakt förutspå när turbulens uppstår.[7]

2.2 Mesh

De partiella differentialekvationer som kan användas för att beskriva vätskeflöden och värmeöverföring är vanligtvis inte användbara för analytiska lösningar, utom för mycket enkla fall. Därför, i syfte att analysera flödets rörelser delas geometrin upp i subdomäner (även kallat celler eller element). I varje cell beräknas sedan ekvationerna och resultatet skickas vidare som indata till nästa intilliggande cell. När alla celler satts ihop kallas geometrin för mesh eller nät (se figur 2.2). En mesh kan se ut på flera olika sätt, de kan se ut som kuber eller ha mer oregelbundet utseende. Storleken kan också vara variera övergeometrin. Allt utifrån vilka krav som ställs på beräkningen.[8]



Figur 2.2 Mesh, geometrin är delad på mitten för att illustrera storleksskillnaden på cellerna

2.3 Navier-Stokes ekvationer

För att beskriva hur fluider beter sig när den påverkas utifrån t ex. av olika krafter, tryck och temperaturer kan Navier-Stokes ekvationer användas. Ekvationerna är uppkallade efter dess skapare, Claude-Louis Navier och George Gabriel Stokes som utvecklade den under första halvan av 1800-talet. Ekvationerna grundas på Newtons andra lag om rörelsemängdens bevarande. Om en kraftsummering görs kommer resultatet bli en icke linjär partiell differentialekvation eftersom fluiden kan påverkas i alla fyra dimensioner (rum och tid), vilket ger fyra variabler (x, y, z, t).[9][10]

Ekvationerna ser ut enligt följande.

Massbevarande

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \tag{2.2}$$

och rörelsemängdsbevarande

$$\frac{\partial \rho \, u_i}{\partial t} + \frac{\partial \rho \, u_i u_j}{\partial x_j} = \rho f_i + \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_i}.$$
(2.3)

 au_{ij} är spänningstensorn för linjärt viskösa gaser

$$\tau_{ij} = \left(\mu \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] - \delta_{ij} \left(\frac{2}{3}\mu \frac{\partial u_k}{\partial x_k}\right)\right).$$
(2.4)

Det finns även en energiekvation men den kommer inta att tas hänsyn till i denna rapport då isotermt och inkompressibelt flöde antas.

För att förenkla beräkningarna är det vanligt att vissa fysikaliska variabler antas vara konstanta, t.ex. kan flödet antas vara inkompressibelt, friktionsfritt eller att det har en konstant temperatur. Trotts dessa förenklingar är ekvationerna fortfarande i stort sätt omöjliga att lösa analytiskt fullt ut. När ekvationerna löses med hjälp av datorer (CFD) görs därför en diskretisering för att det ska vara möjligt att nå fram till en lösning. Detta sker genom att ett rutnät (mesh) skapas som motsvarar geometrin där storleken på rutorna (eller cell för tre dimensioner) bestämmer hur noggrann beräkningen blir. I varje cell löses sedan ekvationerna för varje tidssteg och ett approximativt resultat fås.

2.4 Energispektra

Alla virvlar i ett turbulent flöde kan delas upp efter hur mycket energi de innehåller. Enkelt förklarat innehåller de största virvlarna mest energi medan de minsta innehåller minst energi (se figur 2.3). På x-axeln i Figur 2.3 visas κ som är vågtalet [1/m], virvlarna blir mindre längre bort från y-axeln och på y-axeln visas energiinnehåll. För en mer ingående förklaring av figur 2.3 se [11]. För att på ett korrekt sätt beräkna flödets karaktär måste alla virvlar lösas ända ner till Kolmogorovstorleken (de virvlar med lägst energiinnehåll). Detta är i praktiken omöjligt då det krävs oerhört mycket resurser (tid och datakraft) att utföra en sådan beräkning. Det som bestämmer hur noggrann en beräkning blir är hur mycket av energispektrat som löses (med Naiver-Stokes ekvationer) respektive modelleras (med turbulensmodeller).



Figur 2.3 Energispektrum RANS till DNS [11]

För att kunna utföra CFD-simuleringar måste en metod för hur lösningen av ekvationerna ska ske väljas. Några av de vanligaste metoderna är DNS, LES och RANS. DNS är den metod som är mest exakt och löser alla virvlar, oavsett storlek och energiinnehåll. RANS löser bara medelvärdeskomponenter och modellerar istället fluktuationer med hjälp av en turbulensmodell. LES löser de största virvlarna men modellerar de allra minsta, detta gör att den hamnar mellan DNS och RANS i hur noggranna beräkningarna blir (se figur 2.4). Det som menas med att "lösa upp" en virvel är att Navier-Stokes ekvationer löses fullständigt vilket ger det mest exakta resultatet. Begreppet "modellera" betyder att vissa förenklingar görs i Navier-Stokes ekvationer. Empiriska ekvationer kan också användas på variabler där olika samband är kända, det är vad som gör att beräkningarna för de mindre virvlarna blir mindre krävande (de modelleras). En sak som inte får glömmas är att CFD-simulering är ett relativt nytt påfund vilket gör att utveckling av gamla och nya modeller ständigt sker. [8]



Figur 2.4 Vad som löses respektive modelleras vid olika turbulensmodeller

2.5 Direct numerical simulation (DNS)

DNS är den mest upplösta simuleringen. Vid denna simulering löses alla flödesvariationer i Navier-Stokes ekvationer utan att någon modellering sker [12]. DNS kan många gånger användas som ett referensvärde men är extremt resurskrävande och tar mycket lång tid. Hur många celler meshen måste innehålla i en DNS kan uppskattas enligt följande villkor

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

där N är antalet celler. Detta gör att även väldigt enkla geometrier får väldigt många celler då DNS kräver att cellängden ska vara mindre än Kolmogorovstorleken. Då cellängden minskas måste även tidssteget vid beräkningarna minskas för att Courants tal ska hållas under 1 (se kapitel 2.9). Fler tidssteg resulterar i fler beräkningar vilket kräver mer resurser. Detta gör att även simuleringar med förhållandevis låga Re blir väldigt resurskrävande då antalet celler som måste beräknas blir väldigt många och små. Detta gör DNS för närvarande oanvändbar i industriella tillämpningar. [13]

2.6 Reynold-Average Navier-Stokes (RANS)

RANS är en förenkling av Navier-Stokes ekvationer och är en vanlig metod för att simulera turbulent flöde och förekommer i många industriella tillämpningar.



Figur 2.5 Fluktuationer kring medelvärdet

Flödet kan uttrycks i en medelvärdeskomponent \bar{u} och en fluktuationskomponent u' (se figur 2.5), u är momentanhastigheten.

$$u_i = \overline{u_i} + u'_i \tag{2.5}$$

Även andra storheter som tryck, temperatur och densitet delas upp i en medelvärdeskomponent och en fluktuationskomponent, i denna rapport användes endast hastighet och tryck.

Vid RANS löses endast tidsmedelvärdet för flödesparametrarna (\bar{u}) medan den fluktuerande delen (u) modelleras med hjälp av en turbulensmodell. För att ytterligare förenkla beräkningarna antas flödet vara isotermt och inkompresibelt. Genom att kombinera ekvation (2.2) och (2.5) och medelvärdera ekvationen fås motsvarande ekvation för massbevarande i RANS och får följande utseende

$$\rho \frac{\partial \overline{u_i}}{\partial x_i} = 0 \tag{2.6}$$

och om samma sak görs med ekvation (2.3) fås ekvationen för rörelsemängdsbevarande i RANS enligt

$$\rho \frac{\partial \overline{u_i}}{\partial t} + \rho \overline{u_j} \frac{\partial \overline{u_i}}{\partial x_j} = \rho f_i + \frac{\partial}{\partial x_j} \left[-\bar{p} \delta_{ij} + \mu \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) - \rho \overline{u'_i u'_j} \right].$$
(2.7)

Dessa ekvationer har samma generella form som Navier-Stokes ekvationer men representerar nu tidsmedelvärden. Nu framträder även termen $\rho \overline{u'_i u'_j}$ (Reynolds spänningstensor) som måste beräknas för att kunna lösa ekvationen. Denna term representerar den fluktuerande delen av flödet (*u*) och kan inte försummas då medelvärdet inte räcker för att beskriva flödet korrekt. Istället för att lösa den fluktuerande delen (som är fallet vid DNS) används därför Boussinesq ansats för att skapa en modell över den fluktuerande delen

$$\rho \overline{u'_{i}u'_{j}} = \mu_{t} \left(\frac{\partial \overline{u_{i}}}{\partial x_{j}} + \frac{\partial \overline{u_{j}}}{\partial x_{i}} \right) - \frac{2}{3} \rho k \delta_{ij}$$
(2.8)

där μ_t representerar den turbulenta viskositeten som är en icke-fysikalisk variabel.

För att kunna lösa Raynolds spänningstensor måste den turbulenta viskositeten μ_t kunna räknas ut på något sätt. Detta görs genom att använda turbulensmodeller och det är dessa som studeras i denna rapport. Genom att använda turbulensmodeller går förhållandevis snabbt att göra simuleringar trots väldigt höga Reynoldstal. Problem uppstår dock vid komplexa flöden som domineras av stora sammanhängande virvlar.[14]

2.7 Turbulensmodeller

För att lösa ekvationen (2.8) används olika turbulensmodeller, två av de vanligaste modellerna är k- ε och k- ω . Dessa modeller bygger på RANS och ekvation (2.8) och här har två transportekvationer adderats för att nå en lösning. Transportekvationerna används för att räkna ut *k* och ε resp. *k* och ω där *k* är den turbulenta kinetiska energin, ε är den turbulenta dissipationen och ω är specifik dissipation. Kortfattat kan sägas att *k* bestämmer energin i turbulensen och ε eller ω bestämmer längdskalan, dvs. storleken på virvlarna som ska beräknas. I dessa modeller finns även ett antal konstanter som på empirisk väg har fastställts för att ge ett gott resultat [se bilaga 2], [8].

Nackdelen med k- ε är att den har svårt att beskriva beteendet hos flödet nära väggar, särskilt där turbulens uppstår. k- ω modellen är mer tillförlitlig nära väggar än k- ε och fungerar bra vid måttliga tryckgradienter men fungerar inte lika bra när separationen är tryckinducerad. Dock är k- ω inte tillräckligt bra längre ut från väggen där flödet strömmar fritt för att helt kunna ersätta k- ε . För att lösa dessa problem har turbulensmodellen k- ω SST utvecklats [15].

2.7.1 k-ω SST

k- ω SST (Shear Stress Transport) är en av de vanligaste turbulensmodellerna inom industrin och tanken bakom k- ω SST är att kombinera k- ω modellens styrka nära väggar med k- ε modellens fördelar där fluiden strömmar fritt. Beroende på avståndet till väggarna räknar modellen själv ut vilken av k- ω och k- ε som ska användas. I gränsskiktet där k- ε och k- ω möts används en tangenshyperbolicus-funktion för att få en jämn övergång mellan de två modellerna (se bilaga 2). Detta sätt att angripa ekvationerna gör att k- ω SST på ett bra sätt kan förutspå var separationen kommer att inträffa. Metoden är heller inte lika känslig vad det gäller storleken på nätets celler utan ger ett acceptabelt resultat även vid relativt grov mesh. [15]

Den största nackdelen med k- ω SST är att den inte ger tillräckligt med information om turbulensen. Då den modellerar stora delar av fluktuationerna är det svårt att få andra statistiska storheter än tidsmedelvärdet av flödet.

Pga. att k- ω SST är vanlig i industriella tillämpningar gör att den är en bra referensmodell i denna rapport.

2.7.2 Large Eddy Simulation (LES)

För att lösa upp de stora skalorna finns en metod som kallas LES och är i sig ingen turbulensmodell. Likt RANS finns det även här olika modeller som kan användas. LES fungerar som ett högpassfilter, där de små virvlarna modelleras istället för att lösas medan de stora virvlarna löses (se figur 2.6). I stället för att lösa medelvärdet för de beroende variablerna i tiden (hastighet och tryck i detta fall) som i RANS görs i stället en volymfiltrering av variablerna i LES. LES-simuleringar ställer också högre krav på meshen, om den inte är fin nog kommer resultatet bli sämre än motsvarande RANS-simulering. Därför är det väldigt viktigt hur nätet konstrueras vid LES-simuleringar, nätet måste vara väldigt fint närmast väggarna för att kunna lösa de småskaliga virvlar som uppstår där, dessa virvlar är dock större än Kolmogorovstorleken. Detta gör att simuleringen blir mer korrekt men också mer tids och resurskrävande för att få ett fullgott resultat (jämför RANS som använder sig av väggfunktioner). Den största skillnaden mot turbulensmodeller som använder sig av RANS är hur LES räknar ut den turbulenta viskositeten och att den löser virvlar längre ner på energiskalan och således blir mer noggrann. De två LESturbulensmodeller som använts är WALE (Wall-Adapting Local Eddy-viscosity) i Fluent och Local Dynamic One Equation Eddy i OpenFOAM. Skillnaden mellan dessa två är att WALE använder sig av ett algebraiskt uttryck för att lösa den turbulenta viskositeten medan Local Dynamic One Equation Eddy använder sig av en transportekvation för att lösa ut k och där efter räknar ut den turbulenta viskositeten. Pga. att LES kräver ett finare nät tar en LES-simulering generellt längre tid att genomföra än en RANS-simulering som kan använda sig av ett grövre nät. En erfarenhet som FTMT dragit är att LES kan fungera bra trots en grov mesh nära väggarna, särskilt när turbulensen uppstår pga. blandning och inte pga. väggarnas friktion. Dessa egenskaper gör LES till en bra referensmodell som kan visa på hur bra turbulent flöde kan simuleras med CFD. [16]



Figur 2.6 Illustration av liknelsen med högpassfilter, stora svängningar löses och små modeleras

2.7.3 Partially Resolved Numerical Simulation (PRNS)

PRNS är i grunden en k- ω SST men med vissa modifieringar i några ekvationer och är den modell som ska utvärderas och jämföras med k- ω SST och LES.

I Boussinesqs ansats finns en variabel μ_t som är av största vikt då det är μ_t som modifieras i PRNS. För att enklare kunna skriva ut nästkommande ekvationer kommer följande omskrivning att göras.

$$\mu_t = \rho \nu_t \to \nu_t = \frac{\mu_t}{\rho} \tag{2.9}$$

Där v_t är den turbulenta viskositeten dividerat med densiteten.

I både k- ω SST och PRNS räknas v_t fram genom samma ekvation, nämligen.

$$\nu_{t,SST} = \frac{a_1 * k}{\max(a_1 \omega, SF_2)}.$$
(2.10)

Denna ekvation styrs som synes av k och ω , k och ω räknas ut enligt bilaga 2.

Skillnaden mot k- ω SST är att konstanten RCP (Resolution Control Parameter) multipliceras med $v_{t,SST}$ vilket ger

$$v_{t,RCP} = RCP * v_{t,SST}.$$
(2.11)

 $v_{t,RCP}$ används sedan i nästa tidssteg för att lösa Navier-Stokes ekvationer (se ekvation (2.6)-(2.8)).

Beroende på vilket värde på RCP som väljs (måste vara mellan 0 och 1) kommer den turbulenta viskositetens inverkan på Navier-Stokes ekvationer att påverkas (genom Boussinesqs ansats). I praktiken kommer k- ω SST användas om RCP sätts till 1, ty

$$\nu_{t,RCP} = 1 * \nu_{t,SST}. \tag{2.12}$$

Om RCP sätts till 0 kommer denna term försvinna helt och lösningen kommer motsvara en DNS, dock krävs ett mycket finare nät för att det ska bli en korrekt DNS. RCP säger hur mycket som ska lösas och hur mycket som ska modelleras.

Det finns vissa begränsningar för hur värdet på RCP får sättas, dels beror det på hur fint nätet är men också på flödeshastigheten och hur turbulent flödet är. Syftet är att kunna använda ett lite grövre nät men ända kunna närma sig noggrannheten som finns hos LES. I denna rapport används två sätt för att välja RCP. Ett där ett konstant värde ges (0.6, 0.4 och 0.1). Denna metod har utarbetats av bl.a. NASA[17]. Den andra metoden som används är att ha ett värde som varierar (kallad RCP-filter) i tid och rum. Denna metod har utvecklats på Chalmers av Gyllenram.W och Nilsson.H [18]. Båda metoderna går ut på att genom matematiska villkor ta fram det lägsta RCP-värdet som bör/får användas i varje cell. Om RCP sätts lägre finns en risk att resultatet i dessa celler inte blir helt korrekta, cellens storlek kan bli för stor för att klara av de villkor som modellen ställer utifrån hastighet och turbulens.

PRNS-modellerna som kommer att användas i simuleringarna i denna rapport kommer att namnges enlig följande: RCP0.6, RCP0.4, RCP0.1 och RCP-filter

2.7.3.1 Villkor enligt Shih och Liu

Shih och Liu [17] har för NASA utvecklat en metod som går ut på att ett konstant värde sätts på RCP innan simuleringen startas som sedan inte går att ändra. Hur värdet väljs är ofta en fråga om fingertoppskänsla och erfarenhet men viss vägledning kan fås genom ekvation (2.13). Ett problem med NASA:s villkor är att det bl.a. bygger på cellstorlek vilket gör att det rekommenderade RCP värdet varierar med storleken på cellen men RCP varierar inte med cellstorleken.

Villkoret för att fastställa hur RCP enligt NASA baseras på cellstorleken, k och ε enligt

$$RCP \ge \left(\frac{\sqrt[3]{\Delta}}{l_{RANS}}\right)^{4/3} * \left(\frac{k}{k_{RANS}}\right)^{-2}$$
(2.13)

$$l_{RANS} = \left(\frac{\nu_{RANS}^3}{\varepsilon_{RANS}}\right)^{1/4} \tag{2.14}$$

 ε kan räknas om till ω enligt följande samband

$$\varepsilon = \frac{\omega}{\beta^* k}.$$
(2.15)

En nackdel med denna metod är att det är svårt att bestämma ett RCP när cellstorleken inte är konstant över hela meshen. Ett annat problem är de faktorer som är med i villkoret, särskilt de som benämns med beteckningen RANS (se ekvation (2.13) och (2.14)). I denna rapport har dessa faktorer tolkats som om de skulle komma från en motsvarande k- ω SST simulering. Detta betyder att data måste hämtas från en tidigare simulering för att kunna använda sig av villkoret. Detta var inget problem i detta arbete då data från en sådan simulering fanns att tillgå. Detta villkor användes dock inte för att bestämma vilka RCP som skulle användas i denna rapport, utan bara för att utvärdera PRNS modellerna. I de flesta industriella tillämpningar är detta resursslöseri och således används erfarenhet och fingertoppskänsla i de allra flesta fall för att bestämma RCP-värdet.

2.7.3.2 Villkor enligt Gyllenram och Nilsson

För att lösa problemet med oflexibiliteten hos NASA:s modell har en PRNS-modell med varierande värde på RCP utvecklats (RCP-filter)[18]. Med hjälp av ett varierat RCP kan villkoren uppfyllas över hela meshen (och även i varje tidssteg). Tanken med modellen är att lokalt kunna sätta RCP med hänsyn till varierande cellstorlek och lokala flödesförhållanden. Om ett lågt värde sätts på RCP nära väggarna måste ett fint nät användas för att resultatet ska bli fullgott. Detta gör att simuleringen tar längre tid vilket inte var önskvärt i detta fall. På detta sätt kan beräkningsresurser sparas.

Gyllenram-Nilsson har två olika krav som bestämmer RCP. Det krav som får störst värde i varje cell och tidssteg bestämmer vilket RCP som sätts där. Om något villkor skulle ge ett större värde än 1 kommer dock RCP begränsas till 1, annars blir beräkningarna felaktiga.

Villkor 1:
$$RCP \ge \left(\alpha * \frac{|U| * \delta t}{L_t}\right)^{4/3}$$
 (2.16)

Villkor 2:
$$RCP \ge \left(\alpha * \frac{\sqrt[3]{\Delta}}{L_t}\right)^{4/3}$$
 (2.17)

Där
$$L_t = \frac{\sqrt{k}}{\beta^{**\omega}}$$
 (2.18)

 α =filterkonstant, rekommenderat värde av Gyllenram-Nilsson är mellan 2 oh 3, i denna rapport har ett värde på 3 valts.

$$\beta^* = 0.09$$

För de allra flesta fall i de simuleringar som gjorts i denna rapport är det villkor nummer 2 som bestämmer RCP.

Detta då

$$U * \delta t = C * \sqrt[3]{\Delta}$$
 (se likheten med ekvation 2.21) (2.19)

För de simuleringar som gjorts har Courants tal försökt att hållas under 0.3 genom att välja $\delta t = 2.5 * 10^{-5} s$

Vilket i de allra flesta fall ger

$$U * \delta t < \sqrt[3]{\Delta} * 0.3 \tag{2.20}$$

där vänsterledet är villkor 1 och högerledet villkor 2. I de allra flesta fall är det således villkor 2 som begränsar RCP. Men det är den turbulenta längdskalan, L_t (ekvation 2.18), som bestämmer storleken på RCP då den varierar mer än cellängden.

2.8 Val av turbulensmodell

Det som främst styr valet av vilken turbulensmodell som ska användas är hur mycket tid och resurser som finns, men även geometrin har en viss inverkan. För en enkel geometri kan en enklare mesh genereras, vilket leder till kortare beräkningstid. Är geometrin mer komplicerad måste en mesh med ett finare nät skapas, detta ställer högre krav på modellen som använd. Om en LES-modell väljs ställs extra höga krav på meshen, då resultatet i allra högsta grad beror på precisionen i meshen. Därför beror valet på modell både på vilka resurser som finns tillgängliga och på vilken mesh som används. Således finns ingen modell som passar i alla situationer. Beräkningstiden för en cell och tidssteg skiljer sig inte så mycket åt mellan de olika modellerna då de innehåller nästan lika många ekvationer. Det som gör att vissa modeller tar längre tid är att de kräver ett finare nät med fler celler som i sin tur leder till fler beräkningar per tidssteg.

2.9 Courants tal

Courants tal används för att styra simuleringarna så att inte beräkningarna skenar iväg. Courants tal är kvoten mellan strömningshastigheten multiplicerat med tidssteget och längden på cellen.

$$C = \frac{u\delta t}{\delta x} < 1 \tag{2.21}$$

För att få en extra säkerhetsmarginal kan det vara bra att sikta på ett värde under 0.5. Om C>1 betyder det att en flödespartikel kan hinna förflytta sig igenom en hel cell under ett tidssteg. Detta innebär att inga data fås för partikeln i den specifika cellen. Då kan det se ut som om ingenting har hänt i den cellen, fast i själva verket kan en virvel ha passerat. Detta leder till felaktiga beräkningar och i värsta fall kan simuleringen krascha.

I denna rapport har målet varit att ha ett Courants tal lägre än 0.5 (oftast har det legat runt 0.3). Det som har gjorts för att variera Courants tal är att ändra på tidssteget, d.v.s. tiden mellan varje beräkning. Det finns ingen nytta med att ha ett ännu lägre värde då det bara resulterar i att beräkningarna tar längre tid.

3 Utförande

3.1 CFD-simuleringar

Vid en CFD-simulering genomgås tre steg: preprocessing, solving och postprocessing.[9]

Preprocessing

Det första steget som sker vid en CFD-process kallas preprocessing. I detta steg genereras meshen som simuleringen ska utföras på. Detta görs genom att den grundläggande strukturen skapas med ett 3D CAD-program (Computer Aided Design), exempelvis SolidWorks, CATIA eller Pro/E. Nästa steg i CFD processen är att skapa en mesh med god kvalitet, antingen runt eller inuti i CAD-modellen. Om en flygplansvinge ska simuleras skapas meshen utan på geometrin då det är där flödet befinner sig. I denna rapports simuleringar skapades meshen inuti geometrin då flödet befinner sig inuti fyrkantsröret. Meshen kan exempelvis genereras i något av följande program ANSA, GAMBIT, AUTOGRID eller GridZ.

Solving

När meshen är klar läses den in i simuleringsverktyget, i vårt fall Fluent och OpenFOAM där randvillkor och materialdata väljs. Här väljs även vilken turbulensmodell som ska användas. Beroende på noggrannhet på meshen och vilka randvillkor som är satta varierar simuleringstiden. Tiderna för simuleringarna i denna rapport har varierat mellan 10 till 40 timmar och då har 3 sekunder simulerats.

Postprocessing

Den enorma mängd data som fås från simuleringen måste sedan behandlas för att kunna visualiseras. Detta kan göras i specifika postprocessverktyg som paraView eller direkt i Fluent eller läsas in i program som t.ex. MatLab. I rapporten har MatLab använts för att plotta diagram för hastighetsprofiler och standardavvikelse medan paraView har använts för att generera färgglada bilder på momentanvärden.

3.2 Referensexperiment

Referensdatan som turbulensmodellerna jämförs mot kommer från ett experiment som utförts av Piirto M et al. [2]. Experimentet hade som syfte att bl.a. undersöka skillnaden mellan 2D och 3D simuleringar och jämföra dessa mot mätningar utförda av en PIV (particle image velocimeter) [20]. Denna mätmetod har potential att göra mycket exakta mätningar. Metoden är dock väldigt känslig för störningar t.ex. vibrationer. En annan nackdel med PIV är att den inte kan göra mätningar närmare väggen än den laserstråles diameter som utförmätningen. Rapporten visar även kurvor från en förenklad DNS av referensexperimentet som också är av intresse för denna rapports analys. I experimentets rapport finns inget värde för felmarginal från PIV-mätningarna, men enligt personal på FTMT rör sig felmarginalen ofta mellan 5 - 10 %.

Geometrin åskådliggörs i figur 3.1 och visar ett fyrkantsrör med en klack på 10 mm. Denna geometri kallas för Backward-Facing Step, vilket innebär att turbulens påtvingas när flödet passerar klacken. Flödet av vatten passerar geometrin från vänster till höger. Linjerna som är markerade i figur 3.1 symboliserar de positioner där hastighetsprofiler är utmätta med PIV. "h" motsvarar höjden på klacken som är 10 mm, "-3h" betyder således 30 mm innan klacken.



Figur 3.1 Fyrkantsrör med en Backward-Facing Step konfiguration, modifierad [3]

Rapporten definierar inte vilken turbulensintensitet flödet har i inloppet vilket gjorde det lite svårare att få rätt initialvärden till simuleringarna. För att ändå kunna efterlikna experimentet så mycket som möjligt användes experimentets hastighetsprofil vid linjen "-3h" som inloppsprofil i denna rapports simuleringar. Meshen som har funnits till förfogande har genererats av personal på FTMT och har 300 000 celler. Den är gjord med olika storlekar på cellerna med minst cellstorlek närmast väggarna och större cellstorlek i mitten (se figur 3.2).



Figur 3.2 Meshen som används med cellstorlek i meter

Detta fall har simulerats i både OpenFOAM och Fluent med samma inställningar så när som på turbulensintensiteten som i OpenFOAM var 2 % och i Fluent 10 % vid inloppet.

För att matcha experimentet har följande indata valts:

Fluid: vatten $U_0 = 5,5 \text{ m/s}$ $v = 1 * 10^{-6} m^2/s$ Re = 55000 Simulerad tid = 3 s

 $\delta t = 2,5 * 10^{-5} s$

Dessvärre har RCP-filter endast kunnat användas i OpenFOAM då en kod för RCP-filter till Fluent inte fanns tillgänglig. Övriga modeller har simulerats med båda programvaror och turbulensmodellerna k- ω SST, RCP0.1, RCP0.4, RCP0.6, och LES.

4 Resultat

4.1 CFD-simulering

Alla hastighetsprofiler bygger på ett medelvärde över en längre tid (ca 2-3 s). Medelhastighet och standardavvikelse är två viktiga parametrar som kan fås ur den data som samlades in i de olika punkterna. Dessa profiler ligger till grund för analysen av resultatet då de enkelt kan jämföras med motsvarande profiler från referensexperimentet. Hastighetsprofilen visar på medelhastigheten i varje mätpunkt och visar tydligt hur flödet beter sig. Standardavvikelsen är ett mått på hur "intensiv" turbulensen är i en viss punkt, hur mycket hastighetsfluktuationen avviker från medelhastigheten.

4.1.1 Hastighetsprofil OpenFOAM

Hastighetsprofiler för turbulensmodellerna k- ω STT, RCP0.4, RCP-filter och LES samt experimentet visas i figur 4.1. Vid linje 1 uppvisar alla modeller bra överensstämmelse med experimentet. Även vid övriga linjer är överensstämmelsen mellan experimentella data och simuleringarna goda. RCP-modellerna med konstant RCP ligger mellan LES och k- ω SST som överskattar resp. underskattar hastigheten. RCP-filter är den modell som avviker mest från experimentet.



Figur 4.1 Hastighetsprofil OpenFOAM (exklusive RCP0.1 och RCP0.6)

4.1.2 Standardavvikelse OpenFOAM

Standardavvikelsen för turbulensmodellerna RCP0.6, RCP0.4, RCP0.1 och LES samt experimentet visas i figur 4.2. k- ω SST och RCP-filter är ej med i figur 4.2 då k- ω SST och RCP-filters standardavvikelse är mycket liten. Att det blev så för k- ω SST var väntat eftersom den modellerar nästan all fluktuation (ger ett hastighetsmedelvärde). Varför det blev så för RCP-filter är mer osäkert, det kan tyda på att den modellerar mycket av flödet likt k- ω SST men detta är lite motstridigt då RCP för RCP-filter under klacken är ungefär 0.5 - 0.6 (se figur 4.9) vilket borde ses tydligare i standardavvikelsen då RCP0.6 visar sig tydligt.



Figur 4.2 Standardavvikelse (U_{RMS}) OpenFOAM

Ovanför klacken har alla modeller nästan en standardavvikelse på 0, detta återspeglas även i de DNS som återfinns i experimentets rapport [2]. Under klacken stämmer standardavvikelsen bra med experimentet. Ett högre RCP tenderar att ge en störres standardavvikelse, detta syns även i Pitz-Daily (bilaga 1). Detta eftersom ett högre RCP ger en större modellerad del och således en mindre fluktuerande del. Generellt kan sägas att värdet på standardavvikelsen ökar med avståndet från klacken.

4.1.3 Turbulent intensitet OpenFOAM

Turbulensintensiteten räknas fram med hjälp av följande ekvation

$$u' = \sqrt{\frac{2}{3} * k}.$$
 (4.1)

För k- ω STT och RCP-filter som modellerade stora delar av fluktuationen motsvarar detta standardavvikelsen som således ligger "gömd" i den turbulenta kinetiska energin. Men den kan också tillämpas på de övriga modellerna. I figur 4.3 visas dels standardavvikelsen för referensexperimentet, men också ekvation (4.1) plottad vid linje 2-4 för alla turbulensmodeller som simulerades i denna rapport. Värdet på *k* representeras av tidsmedelvärdet av de sista sekunderna av simuleringarna, på samma sätt som hastigheten till hastighetprofilerna.



Figur 3 Turbulent intensitet (u'/U_0) för alla modeller

Profilen för LES har ett utseende som påminner mycket om experimentets standardavvikelse, dock har den över lag ett mindre värde. Den kan inte direkt jämföras med de övriga modellerna då LES använder sig av andra typer av ekvationer. Av de andra modellerna är det k- ω STT som har det högsta värdet vid alla linjer, detta eftersom den modellerar stora delar av fluktuationerna (detta kan jämföras med standardavvikelsen som är mycket liten för k- ω STT). RCP-filter som också hade en mycket låg standardavvikelse är den modell efter k- ω STT som har störst turbulent intensitet, detta beror antagligen på samma orsak som för k- ω STT. RCP0.6, RCP0.4 och RCP0.1 radar upp sig där lägst RCP har lägst turbulent intensitet. Signifikant för dessa tre är att de under klacken har ett lägre värde än ovanför klacken. Detta är omvänt om en jämförelse görs med standardavvikelsen och beror på att mycket av fluktuationen löses under klacken.

4.1.4 Hastighetsprofil Fluent

Hastighetsprofilen för turbulensmodellerna k- ω STT, RCP0.4, RCP-filter och LES samt experimentet visas i figur 4.4. Vid linje 1 överensstämmer modellerna bra med experimentet, likaså vid linje 2 förutom k- ω SST som avviker något under klacken. Vid linje 3 och 4 börjar skillnader utkristallisera sig mellan modellerna. Tendensen är att k- ω SST överskattar medan LES underskattar hastigheten under klacken. Ovanför klacken är det tvärtom, där överdriver LES och k- ω SST underskattar hastigheten. RCP-modellerna ligger överlag mellan k- ω SST och LES vid alla linjer. De "hack" som syns i profilerna beror på att fler mätpunkter användes i Fluent än i OpenFOAM (50 respektive 30). Detta resulterade i att två mätpunkter hamnade i samma cell på vissa ställen, vilket leder till att de får samma hastighet. När grafen plottas hamnar de två mätvärdena på två olika punkter fast de egentligen borde vara i samma punkt i y-led. En annan skillnad är att om inte mätpunkten ligger mitt i cellens centrum interpolerar OpenFOAM storleken på hastigheten, detta med avseende på avståndet från cellens centrum till mätpunkten. Fluent gör inte denna interpolering och därmed blir inte övergången mellan mätpunkterna lika jämn.



Figur 4 Hastighetsprofil Fluent (exklusive RCP0.1 och RCP0.6)

4.1.5 Standardavvikelse Fluent

Standardavvikelsen för turbulensmodellerna RCP0.6, RCP0.4, RCP0.1 och LES samt experimentet visas i figur 4.5.





Standardavvikelsen för modellerna har samma form som experimentet men avviker i storlek. Generellt kan sägas att värdet på standardavvikelsen ökar med avståndet från klacken. Ingen koppling mellan storheten och modellerna verkar finnas då det varierar vilken modell som har störst värde. En viss tendens finns dock till, precis som i OpenFOAM, att standardavvikelsen sjunker med högre RCP. LES tenderar att vara den modell med högst standardavvikelse vilket är väntat då den modellerar minst. Även här går värdet mot noll ovanför klacken, vilket även var fallet i OpenFOAM-simuleringarna. Överlag är standardavvikelsen lägre än motsvarande simuleringar i OpenFOAM .

4.1.6 Turbulent intensitet Fluent

Data för turbulent kinetisk energi (k) som används för att räkna ut den turbulenta intensiteten skrevs ej ut vid simuleringarna i Fluent då detta ej var programmerat i den UDF som användes.

4.2 Jämförelse av OpenFOAM och Fluent

4.2.1 Hastighetsprofil

Enligt figur 4.6 visar k- ω SST hastighetsprofil från Fluent och OpenFOAM överlag på en bra överensstämmelse med varandra. De skiljer sig lite från varandra i mitten av flödet där Fluent visar en något högre hastighet. En annan skillnad syns tydligt i linje 3 och 4 närmast golvet, där avviker OpenFOAM både från Fluent och experimentet.



Figur 6 Hastighetsprofil OpenFOAM mot Fluent k-ω SST

I figur 4.7 visas hastighetsprofilerna för RCP0.4. Tendensen för hur OpenFOAM och Fluent avviker från varandra är den samma som för k- ω SST. Flödet mitt i röret är också något plattare än för k- ω SST vilket stämmer med teorin att profilen blir plattare när turbulensen ökar.



Figur 7 Hastighetsprofil OpenFOAM mot Fluent RCP0.4

4.2.2 Standardavvikelse

I figur 4.8 visas standardavvikelsen för RCP0.4. Här syns tydligt att OpenFOAM s simulering har en högre standardavvikelse under klacken, undantaget är vid linje 4 där båda programmen har i stort samma värde på standardavvikelsen. Anledningen till att ingen standardavvikelse för k- ω SST visas är av samma anledning som nämnts tidigare, nämligen att den är väldigt liten.



Figur 8 Standardavvikelse OpenFOAM mot Fluent RCP0.4

4.3 Analys

De figurer som presenteras i bilaga 3 och som mycket av analysen bygger på är framtagna efter att simuleringarna i Fluent och OpenFoam var klara. I programmen finns en funktion där olika variabler (t ex. ω och k) från sluttillståndet kan användas för att generera dessa bilder. Det finns även en funkton där ekvationer kan skrivas in och på så sätt kan de bilder som visar vad RCP-villkoren ger för rekommenderade värden för de olika turbulensmodellerna.

4.3.1 Hastighet, standardavvikelse och turbulent intensitet

I simuleringarna som gjorts i denna rapport kan små men tydliga skillnader mellan modellerna med olika RCP ses, både vad det gäller hastighet och standardavvikelse. De skiljer sig också med samma tydlighet från referensmodellerna k- ω SST och LES. I Både OpenFOAM och Fluent finns mönster för hur de olika modellerna skiljer sig från varandra, oftast ligger RCP-modellerna mitt i mellan k- ω SST och LES (se figur 2.3 och 2.4). Detta var väntat då RCP-modellerna är en "hybrid" av SST och DNS.

Erfarenheter säger dock att LES borde vara den bättre av modellerna men detta stämmer inte i alla mätpunkter då k- ω SST ligger närmare experimentet i vissa fall, framförallt i Fluent. Detta kan dock härledas till nätets utformning som är för grovt (särskilt nära väggarna) för att LES ska komma till sin fulla rätt.

RCP-filter som på förhand hade höga förväntningar på sig är den modell som i de flesta fall avviker mest från experimentet (överdriver hastigheten). Särskilt mycket avviker RCP-filter i mitten av flödet och i regionen under klacken där mest turbulens finns. Figur 4.9 visar vilket RCP-värde som satts enligt villkoren i ekvation (2.16) och (2.17). Jämförs de punkter där RCP-filter har samma RCP-värde som någon av de modellerna med konstant RCP avviker resultaten ifrån varandra trotts att samma RCP-värde har använts. En anledning till den i överlag överdrivna hastigheten för RCP-filter kan vara att massflödet måste vara konstant och på så sätt blir hastigheten överdriven över hela profilen pga. att massflödet blivit högt i en region och måste då kompenseras i andra regioner.

En annan tendens är att k- ω SST är sämre än de andra modellerna på att efterlikna experimentets data precis intill väggarna (borde bli noll), detta syns särskilt vid linje två och tre och visar sig både i OpenFOAM och i Fluent (se figur 4.1 och 4.4). Detta kan bero på avsaknaden av väggfunktioner (endast i OpenFOAM) som valdes att inte användas vid denna rapports simuleringar.

Vad det gäller standardavvikelsen från simuleringarna gjorda i OpenFOAM finns det ett tydligt mönster (se figur 4.2). När ett lågt RCP-värde används blir standardavvikelsen högre. Detta kan tyda på att de modeller som löser större delen av fluktuationerna (RCP0.1 och LES) får en större fluktuerande del än de modeller som har ett högre RCP, i synnerhet k- ω SST och RCP06. Men det kan också tyda på de "fel" som introduceras i Navier-Stokes ekvationer i och med användandet av PRNS, vilket ger ett "brus" som kan höja standardavvikelsen. k- ω SST har en standardavvikelse som är mycket nära 0 då den modellerar större delen av flödet och således syns ingen fluktuation då den finns "gömd" i k. Anledningen till att standardavvikelsen är nära 0 mitt i flödet tyder på att alla modeller modellerar fluktuationerna i denna region.

Vad det gäller standardavvikelsen för simuleringarna i Fluent (se figur 4.4) är resultatet lite mer tvetydigt. Vid linje 1 och 2 är det svårt att se ett mönster (förutom för LES) och resultatet avviker både från motsvarande OpenFOAM-resultat och teorin. Vid linje 4 visas dock samma tendens som vid OpenFOAM simuleringarna. Vad detta beror på är svårt att säga men en anledning kan vara att Fluent behandlar ekvationerna på ett annorlunda sätt jämfört med OpenFOAM.

Den turbulenta intensiteten som visas i figur 4.3 visar på ett bra sätt hur de olika turbulensmodellerna fungerar. Storleken på k är ett mått på hur mycket som modelleras respektive löses för varje modell och figur 4.3 visar bra överensstämmelse med teorin. Den visar också på en motsatt överensstämmelse med standardavvikelsen vilket också var förväntat. Tyvärr fanns inte motsvarande värden för Fluent att tillgå, vilket hade varit intressant att analysera.

4.3.2 Villkor för RCP samt k och ω

När en jämförelse gjordes mellan OpenFOAM och Fluent gällande Gyllenram-Nilssons villkor nummer 2 (se bilaga 3, figur 8.3.10) finns stora likheter i hur RCP bör anges enligt villkoret. Den enda modellen som avviker är RCP0.6 från OpenFOAM som säger att RCP kunde satts lägre i den bortre regionen under klacken än vad Fluents simulering av RCP0.6 säger. Detta kan bero på att bilderna är ögonblicksbilder vilket gör en direkt jämförelse mellan de olika simuleringarna svår.

Något som är signifikant för modellerna RCP0.6, RCP0.4 och RCP0.1 i både OpenFOAM och Fluent är att Gyllenram-Nilssons villkor säger i stort sätt samma sak. Mitt i flödet kan värdet på RCP närma sig noll i OpenFOAM:s fall medan RCP i Fluent inte bör sättas lägre än 0.15-0.20 i samma region.

I regionen under klacken är de tre modellerna ännu mer samstämmiga, nämligen att värdet för RCP bör närma sig 1. Detta beror delvis på meshen och dess cellstorlek. I och med att cellerna är som störst under klacken (se figur 3.2) blir värdet på $\sqrt[3]{\Delta}$ störst där, vilket enligt ekvation (2.17) ger ett högt RCP. Men det som antagligen gör störst inverkan är *k* och ω . Om en jämförelse görs mellan området ovanför klacken med området under klacken varierar skillnaden på k och ω väldigt mycket (se bilaga 3, figur 8.3.6 och 8.3.7). I området över klacken har *k* värden mellan ~0,3-0,5 m²/s² och under klacken ligger värdena mellan ~0-0,3 m²/s². ω har storleken <100 s⁻¹ ovanför klacken och ligger mellan ~800-1500 s⁻¹ under klacken.

Det är samverkan mellan k och ω och i viss mån även $\sqrt[3]{\Delta}$ som bestämmer vilket RCP som Gyllenram-Nilssons villkor ger. RCP0.6, RCP0.4 och RCP0.1 har fasta värden på RCP och Gyllenram-Nilssons villkor ger en nedre gräns för vad RCP bör sättas till. Detta betyder att dessa tre modeller i både OpenFOAM och Fluent har för lågt satta RCP i regionen under klacken (om villkor nr 2 ska uppfyllas), vilket borde leda till mer oexakta resultat i denna region. I regionen mitt i flödet ligger alla modeller med fasta RCP över det värde som Gyllenram-Nilssons villkor nr 2 anger och därmed bör de ge ett bättre resultat. I figur 4.9 visas till vänster hur RCP borde sättas enligt enligt Gyllenram-Nilssons villkor nr 2 och till höger visas hur RCP sattes vid simuleringen där RCP-filter avvändes. I stort är dessa två identiska och det är bara precis efter klacken som ett RCP på 1 användes (förutom närmast väggarna). I övrigt är RCP<0.6.



Figur 9 Jämförelse av vad RCP-värdet borde vara och vad det faktiskt är i RCP-filter

I bilaga 3, figur 8.3.7 syns att värdet på ω skiljer sig väldigt mycket mellan Fluent och OpenFOAM. Detta kan antingen bero på att ekvationerna löses på olika sätt eller att randvillkoren skiljer sig åt (alternativt olika momentanvärden). Då ω har stor inverkan för hur Gyllenram-Nilssons villkor nr 2 bestämmer RCP kan detta vara en av anledningarna till att RCP-filter avviker från de övriga modellerna. Då ω har ett lågt värde mitt i flödet blir även RCP lågt där (se figur 4.9), detta kan leda till onoggranheter vid RCP-filter simuleringarna.

4.3.3 LES

När meshens struktur och uppbyggnad studeras och jämförs med hur LES-modellens hastighetsprofil ser ut kan vissa slutsatser dras. Meshens cellstorlek verkar inte påverka resultatet i någon större negativ riktning. Dock kan en viss avvikelse från experimentets resultat ses under klacken (där cellerna är som störst) i simuleringen som gjordes i Fluent, framförallt i linje 3 och 4 (se figur 4.4). Detta kan bero på att blandningszonen återfinns där och att det finns virvlar där som är små nog för att inte LES-modellen ska lyckas fånga upp dem pga. det grova nätet. LES-modellen i OpenFOAM lyckas bättre med att efterlikna experimentet vid motsvarande linjer. Detta kan möjligtvis förklaras med att två olika typer av LES-modeller användes i de två programmen (WALE i Fluent och Local Dynamic One Equation Eddy i OpenFOAM).

4.3.4 Jämförelse mellan Fluent och OpenFOAM

En anledning till att Fluent visar en något högre hastighet kan vara att en högre turbulensintensitet valdes vid inloppet (10 % mot 2 %). Att hastigheten skiljer sig närmast golvet kan bero på att inga väggfunktioner användes i OpenFOAM, detta gör att mindre turbulens syns i hastighetsprofilen. Att använda väggfunktioner är annars att rekommendera när en grov mesh används.

Standardavvikelsen för RCP0.4 avviker mycket från varandra vid linje 2 och 3, vid linje 4 har de i stort sätt uppnått samma storhet. Om detta också beror på den högre turbulenta intensiteten som sattes i Fluent är svårt att säga. Möjligen hade det förväntade då varit att Fluents simulering skulle haft det högsta värdet. En annan förklaring kan vara att ekvationerna behandlas på olika sätt.

5 Diskussion

I det stora hela har de flesta modeller presterat bra och visat en god överensstämmelse med referensexperimentet. Om modellerna jämförs inbördes har de också i stort betett sig som förväntat, k- ω SST har varit den modell som överlag har visat på minst upplöst turbulens och LES har oftast visat på mest upplöst turbulens. Mellan dessa två har oftast PRNS modellerna radat upp sig med fallande storlek på RCP.

För att göra en fullständig utvärdering av PRNS bör ett större urval av simuleringar göras än vad som gjorts i denna rapport. Men denna rapport är en bra grund och väcker nya frågeställningar inför fortsatt utvärdering.

En av de viktigaste frågorna som väcks är hur storleken på meshen påverkar resultatet. Med en finare mesh borde de modeller med ett lågt RCP-värde samt RCP-filter överensstämma bättre med experimentet. Detta leder dock till vissa komplikationer. Om cellängden halveras leder det till att antalet celler stiger med en faktor 8 och med ökat antal celler ökar även beräkningstiden. Om cellängden minskas tillräkligt mycket kommer meshen uppnå de kriterier som LES ställer på meshen. Då går syftet med PRNS förlorad då tanken är att den ska prestera bra trots en medelgrov mesh. Frågan är således var gränsen går när det blir olönsamt att minska cellängden ytterligare.

En fråga som också väcks är hur RCP-villkoren (NASA och Gyllenram-Nilsson) påverkas om cellängden minskas? Förhållandet mellan cellängden och villkoren är inte linjära. Om cellängden minskas kommer även k och ω ändras och därmed även villkoren för RCP. Detta kan betyda att om cellängden minskas med t ex. en faktor 2 kan RCP minskas med en större faktor än den faktor som cellängden minskades med. På så sätt kan kanske en större förändring av RCP göras i utbyte mot en mindre förändring av cellängden. Men möjligheten finns att det kan bli tvärt om också (se ekvation (2.17)).

En annan faktor som kan användas för att kunna använda ett lägre RCP är tidssteget (δt) på simuleringarna. I simuleringarna som gjordes i denna rapport var Gyllenram-Nilssons villkor nr 1, ekv. (2.16) underordnad villkor nr 2, ekv. (2.17) i de allra flesta fall. Detta för att Courants tal har haft ett maximum på omkring 0.3, men medelvärdet har legat långt under 0.3. Därför kan ett antagande göras att det endast är ett fåtal celler som gör att Courants tal blir 0.3. Det skulle innebära att det maximala värdet på Courants tal skulle kunna stiga upp mot 1 utan några större förluster i noggrannhet. En annan förhoppning är att villkor nr 1 ska gå in och ta över när Courants tal har ett maximum. Om detta fungerar skulle tidssteget kunna ökas upp mot en faktor tre och beräkningstiden skulle således minskas med samma faktor.

Något som är signifikant för alla PRNS-modeller med konstant RCP är att enligt villkoren (både NASA och Gyllenram-Nilsson) kan ett lågt RCP inte användas under klacken. Det är egentligen bara RCP0.6 i OpenFOAM som tillåter ett lägre värde än 1 och det relativt långt efter klacken (se bilaga 3, figur 2.10). Det vore önskvärt att kunna använda ett lägre RCP under klacken för att på så sätt mer noggrant lösa upp turbulensen. För att lyckas med detta kan resonemang enligt ovan användas, antingen minska cellängden eller minska tidssteget på beräkningarna.

En annan intressant frågeställning är hur lågt RCP kan sättas när ett konstant värde används (men även för RCP-filter). Det finns en viss tendens till att de modeller med ett lågt RCP (RCP0.1 och till viss del även RCP0.4) stämmer mindre bra överens med referensexperimentet. Vad händer om det sätts för lågt och går det att uppskatta felen utan att jämföra med experimentdata?

5.1 Framtida arbete

Ett första steg i en fortsatt utvärdering av PRNS är att konstruera en finare mesh av samma geometri som i referens experiment och där efter genomföra simuleringar med samma turbulensmodeller som i denna rapport. Om någon eller några av PRNS-modellerna visar på bättre överensstämmelse med referensexperimentet visar det på att meshen i denna rapport inte var optimal. Det är även intressant att se hur LES beter sig gentemot PRNS vid en mindre cellstorlek, närmar de sig varandra i noggrannhet eller blir skillnaden större? Det viktiga med detta steg är att utreda hur små cellerna måste göras för att uppnå ett fullgott resultat. Vid en viss storlek är det kanske bättre att använda en LES.

Nästa steg skulle kunna vara att testa några av PRNS-modellerna på andra typer av geometrier. För att verkligen se hur modellerna beter sig då turbulens endast genereras av väggarna skulle en simulering på en fyrkantig kanal kunna göras. Det svåra med simuleringar där turbulens uppstår pga. väggarna är att mycket små virvlar bildas först för att sedan övergå till större virvlar. Till geometrier med denna form finns det gott om experimentdata och DNS att tillgå för jämförelse. Även geometrier där turbulensen till största delen skapas på grund av avlösningar och komplex geometri så som T-stycken och ventiler bör simuleras. Dessa geometrier är även av större intresse för FKA. Men även här är det viktigt att ha tillgång till experimentdata som jämförelse.

Även skillnaden mellan Fluent och OpenFOAM bör utredas. Då särskilt med avseende på inloppets turbulentintensitet och väggfunktioner. Det vore bra att veta om det är någon skillnad på programmen eller om det bara handlar om skillnader på ingångsvärden. Att skriva ut k för en Fluent-simulering och sedan jämföra med en motsvarande OpenFOAM-simulering vore också bra att göra för att se om det finns skillnader på hur programmen behandlar k.

I Gyllenram-Nilssons villkor finns en konstant, α , den så kallade filterkonstanten. Värdet på α ska tolkas som att det behövs α -antal celler för att lösa upp den minsta cellen, i denna rapport användes värdet 3. Men enligt Gyllenram-Nilsson kan ett värde mellan 2-3 användas och om 2 används i stället blir RCP enligt Gyllenram-Nilssons villkor nära en faktor 2 mindre. Detta kan ha stor inverkan på RCP-filters simuleringar och således bör olika värden på α och dess inverkan utredas.

En mer ingående matematisk analys vore också intressant. För att på så sätt utreda dels hur värdet på RCP påverkar k och ω och hur det i sin tur påverkar villkoren för hur RCP bör sättas. Men även villkoren i sig, är de rimligt satta eller kan de förändras på något sätt?

5.2 Felkällor

Det är svårt att peka ut några direkta felkällor från simuleringarna. Som nämnts tidigare i rapporten har indatan för Fluent och OpenFOAM skilt sig något åt. Framförallt två saker, turbulensintensiteten vid inloppet och att inga väggfunktioner användes i OpenFOAM. Utöver det finns det skillnader i hur lösarna behandlar ekvationerna i de olika programmen. Hur stor påverkan dessa faktorer har på resultatet är svårt att uppskatta då t ex. hastighetsprofilerna inte skiljer sig nämnvärt åt.

Angående modellernas överensstämmelse med referensexperimentet finns det vissa faktorer som kan göra att resultaten avviker. Dels finns det en inbyggd felmarginal i mätmetoden (PIV) som kan medverka till avvikelser, det finns också vissa tveksamheter för experimentets inloppsvillkor. I rapporten finns ingen angiven storhet på inloppets turbuletintensitet, men ett antagande som kan göras är att flödet har spolats igenom med ett antal genomflödestider för att minimera fel innan mätningarna startas. I kurvan för referensexperimentets standardavvikelse finns dock en viss tendens till turbulens mitt i bulkflödet, detta kan tyda på viss fluktuation finns redan i inloppet. Dessa faktorer gör det svårt att exakt efterlikna referensexperimentet med simuleringarna fullt ut.

6 Slutsats

Med den enkla geometri och den grova mesh som användes i denna rapport fungerar alla turbulensmodeller som utvärderats relativt bra (möjligen bortsett från RCP-filter) för att återspegla experimentdata. I dagsläget är det dock för tidigt att använda sig av PRNSmodellerna i det dagliga arbetet då alla variabler som påverkar modellerna inte är fullt utredda än.

Hur nätet påverkar modellerna är en av de viktigaste parametrarna att utreda mer. Fler simuleringar bör göras där samma geometri som i detta arbete används men med andra, förslagsvis mindre storlekar på nätet. I detta steg behöver inte alla modeller utvärderas då denna rapport visar hur modellernas beteende ser ut inbördes. k- ω SST och LES kan dock vara bra att som referensmodeller. Detta bör fastställa hur fina näten måste vara. Om nätet blir för fint kan det vara bättre att anpassa nätet till en LES-modell istället för en PRNS-modell.

Därefter bör någon eller några av modellerna simuleras på mer komplex geometri som mer överensstämmer med verkliga förhållanden.

Att göra en matematisk analys av de villkor som satts av NASA och Gyllenram-Nilsson kan också vara till nytta. På så sätt kan anledningen till varför modellerna skiljer sig åt utredas på ett mer teoretiskt plan.

7 Referenser

- [1] Versteeg, H.K., Malalasekera, W., (1995), An introduction to computational fluid dynamics - The finite volume method, Longman Group, Harlow, ISBN 0-582-21884-5.
- [2] Piirto, M., Karvinen, A., Ahlstedt, H., Saarenrinne, P., Karvinen, R., (2007), PIV Measurements in Square Backward-Facing Step, Institute of Energy and Process Engineering, Tampere University of Technology, Tampere.
- [3] Pitz, R.W., Daily, J.W., (1983), Combustion in a Turbulent Mixing Layer Formed at a Rearward-Facing Step, Universivy of Carlifornia, Berkeley.
- [4] http://www.openfoam.com 28/3-2011
- [5] Figure 2.1 http://encyclopedia2.thefreedictionary.com/Chaos+%28greek+god%29
- [6] http://sv.wikipedia.org/wiki/Turbulens 3/5-2011
- [7] Fluent user guide v.6.3 http://my.fit.edu/itresources/manuals/fluent6.3/help/ 8/6-
- 2011
- [8] www.CFD-online.com 6/7-2011
- [9] http://www.cctech.co.in/training/dacfd/DACFDfaq.htm#What_are_Navier-Stokes_Equations 9/6-2011
- [10] http://en.wikipedia.org/wiki/Navier-Stokes_equations 5/5-2011
- [11] Tinoco, H., Lindqvist, H., Frid, W., (2010), Numerical Simulation of industrial Flow, Från Angerman, L., Numerical Simulations, Examples and Applications In Computational Fluid Dynamict
- [12] http://en.wikipedia.org/wiki/Turbulence kinetic energy 4/5-2011
- [13] http://en.wikipedia.org/wiki/Direct numerical simulation 16/5-2011
- Hsieh, K.J., Lien, F.S., Yee, E., (2009), Towards a Unified Turbulence Simulation [14] Approach for Wall-Bounded Flows.
- [15] Menter, F.R., Kuntz M., Langtry R. (2003), Ten Years of Industrial Experience with the SST Turbulence Model Från Hanjalic K., Nagano Y., Tummers M., Turbulence, Heat and Mass Transfer 4.
- [16] Villiers, E., (2006), The Potential of Large Eddy Simulation for the Modeling of Wall Bounded Flows, Department of Mechanical Engineering, Imperial College of Science.
- [17] Shih, T.H., Liu, N.S., (2008), Assessment of the Partially Resolved Numerical Simulation (PRNS) Approach in the National Combustion Code (NCC) for

Turbulent Nonreacting and Reacting Flows, NASA, Glenn Research Center Cleveland, Ohio.

- [18] Gyllenram W., Nilsson H., (2008), Design and Validation of a Scale-Adaptive Filtering Technique for LRN Turbulence Modeling of Unsteady Flow, Department of Applied Mechanics, Chalmers University of Technology, Göteborg.
- [19] Davidsson, L., (2011), *An Introduction to Turbulence Models*. Department of thermo and fluid dynamics, Chalmers University of Technology, Göteborg.
- [20] Prasad A.K, (2000), *Particle Image Velocimetry*, Department of Mechanical Engineering, University of Delaware, USA

Bilaga 1 Experiment 2 - Pitz-Daily

Pitz-Daily fallet baseras på ett experiment utfärdat av Robert W. Pitz och John W. Daily [4]. Detta fall ligger som tutorialfall i OpenFOAM. Experimentet hade som syfte att studera förbränning av en flytande blandning av luft och propan i en turbulent blandningszon vid olika Reynoldstal (15k, 22k och 37k). Anledningen till att detta experiment valdes var för att geometrin är enkel och kan därför enkelt och snabbt simuleras i ett CFD-program. Att simulera Pitz-Daily är även en bra inkörningsport till CFD för författarna. I experimentets rapport finns även bra data (hastighetsprofiler, standardavvikelser) som enkelt kan jämföras med de resultat som fås ut ur CFD-simuleringarna.

Bredden på "röret" är mycket bredare än höjden (ca 3ggr), detta gör att flödet i mitten där data samlas in inte påverkas nämnvärt av väggarna. På så vis kan experimentet ses som ett två-dimensionellt fall. När fallet sätts upp i OpenFOAM är meshen tre-dimensionell, men inställningar görs så beräkningar inte sker i z-led. Då kan beräkningarna i OpenFOAM jämföras med experimentet på ett bra sett.

I figur 8.1.1 ses hela förbrännaren där flödet av luft och propan går från höger till vänster.



Figur 8.1.1 Ritning över hela förbrännaren (OBS spegelvänd gemtemot figur 8.1.2)[4]

I detta fall har simuleringar enbart gjorts i OpenFOAM av den anledningen att det inte har funnits tillgång till en mesh som passar Fluent. LES, k - ω SST, RCP0.6 och RCP0.4 är de turbulensmodeller som används i detta fall. Simuleringarna i denna rapport har bara räknat på den sista delen av stycket se figur 8.1.2 (från klacken till utblåset). I figur 8.7.2 syns också de linjer där data har inhämtats. Avståndet H är höjden på klacken vilken är 25mm.



Gemensamma indata:

 $U_0 = 13,3 \text{m/s}$ Re = 22000 $v = 1,51136 * 10^{-5} m^2/s$ $\delta t = 1 * 10^{-5} s$ Simularingstid = 3 sekunder

Resultat

I figur 8.1.3 visas hastighetsprofilen för simuleringarna av Pitz.Daily. Överlag överensstämmer k- ω SST modellen väl med experimentdatan från Pitz-Daily. RCP0.6 profil överensstämmer bättre med k- ω SST modellens än RCP0.4. LES har en tendens till att överdriva hastigheten (både negativ och positiv hastighet i x-led), framförallt i området under klacken. Detta syns tydligt vid linje 2 och linje 3 men också i linje 5 och 6. Alla modeller utom k- ω SST visar på större avvikelse från experimentet längre ifrån klacken i xled, framför allt nära väggarna. En annan tydlig tendens är också att RCP0.6 och RCP0.4 allt som oftast ligger mellan k- ω SST och LES, under klacken är i stort sätt alltid LES som visar den högsta hastigheten och över klacken är det nästan alltid (förutom i linje 1 och 2 och i viss mån linje 3 där LES visar en högre hastighet) k- ω SST som visar högst hastighet.



Figur 8.1.3 Hastighetsprofil fall 1

Standardavvikelsen som visas i figur 8.1.4 stämmer ganska väl med experimentdatan vid linje 1 och linje 2, dock avviker den mer och mer vid linjerna längre bort från klacken och då framförallt kring väggarna. Vid alla linjer har RCP04 högre standardavvikelse än RCP06. LES har oftast lite lägre värde än de andra två modellerna men det finns mycket fluktuation i standardavvikelsen.



Figur 8.1.4 Standardavvikelse (U_{RMS}) fall 1

Analys

Tyvärr finns inga andra data så som värden för k eller ω för dessa simuleringar som gjorts på Pitz-Daily. Inte heller gjordes några simuleringar med modellerna RCP0.1 eller RCP-filter. Hade dessa data funnit hade djupare analyser kunnat göras.

Den största skillnaden mellan resultatet från simuleringarna i bilaga 1 och de som finns i rapporten är att modellernas hastighetsprofiler avviker mer från varandra i bilaga1. Skillnaden mellan bilaga 1 och simuleringarna i rapporten ligger främst i geometrin men också i meshen. Klackhöjden motsvarar ½ av den totala höjden men bara 1/5 av den totala rörhöjden i rapportens simuleringar. Detta kan vara en av anledningarna till att modellerna avviker mer från varandra i bilaga 1. Att modellerna inte överensstämmer med experimentet kan dels bero på onoggrannheter i experimentets mätdata, men även som nämnts tidigare, meshen. Dessa orsaker kan ligga bakom att k- ω SST är den modell som överensstämmer bäst med experimentdatan.

Den mest intressanta slutsatsen som kan dras från dessa simuleringar är dock hur modellerna förhåller sig till varandra. Likt simuleringarna i rapporten är k- ω SST och LES de två extremerna medan RCP0.6 och RCP0.4 befinner sig mitt i mellan. RCP0.6 något närmre k- ω SST och RCP0.4 något närmre LES, precis som teorin säger.

Anledningen till att standardavvikelsen avviker nära väggarna kan förklaras med att väggfunktioner ej användes. Om detta är hela sanningen är svårt att säga då de inte alls har samma utseende som profilerna i rapporten. Profilerna har inte heller någon tendens att närma sig 0 mitt i flödet (se figur 4.2).

Bilaga 2

Ekvationer

k-ω SST

De ekvationer som skiljer k- ω SST från övriga RANS-baserade turbulensmodeller (så som k- ω och k- ε) ser ut enligt följande,

turbulent kinetisk energi

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right],$$

Specifik dissipation

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta^* \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \sigma_{\omega^2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}$$

och kinematisk viskositet (Eddy viskositet)

$$v_{t,SST} = \frac{a_1 * k}{\max(a_1 \omega, SF_2)}.$$
(2.11)

För att kunna stänga dessa ekvationer krävs också följande konstanter och stängningsekvationer.

$$F_{2} = tanh\left[\left[max\left(\frac{2\sqrt{k}}{\beta^{*}\omega y}, \frac{500v}{y^{2}\omega}\right)\right]^{2}\right],$$

$$P_{k} = min\left(\tau_{ij}\frac{\partial U_{i}}{\partial x_{j}}, 10\beta^{*}k\omega\right),$$

$$F_{1} = tanh\left\{\left\{min\left[max\left(\frac{\sqrt{k}}{\beta^{*}\omega y}, \frac{500v}{y^{2}\omega}\right), \frac{4\sigma_{\omega^{2}}k}{CD_{k\omega}y^{2}}\right]\right\}^{4}\right\},$$

$$CD_{k\omega} = max\left(2\rho\sigma_{\omega^{2}}\frac{1}{\omega}\frac{\partial k}{\partial x_{i}}\frac{\partial \omega}{\partial x_{i}}, 10^{-10}\right),$$

$$\Phi = \Phi_{1}F_{1} + \Phi_{2}(1 - F_{1}),$$

$$\beta_{1} = \frac{3}{40}, \beta_{2} = 0.0828,$$

$$\beta^{*} = \frac{9}{100},$$

$$\sigma_{k1} = 0.85, \sigma_{k2} = 1,$$

$$\sigma_{\omega 1} = 0.5, \sigma_{\omega 2} = 0.856.$$

LES- Wall-adapting local eddy-viscosity (WALE)

När CFD-simuleringarna gjordes i Fluent användes en LES-baserad turbulensmodell kallad WALE och dess ekvationer ser ut som följer.

Filtrerad Navier-Stokes ekvation

$$\frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\left[\nu + \nu_t \right] \frac{\partial \overline{u}_i}{\partial x_j} \right)$$

För WALE modelleras den turbulenta viskositeten enligt

$$\mu_t = \rho \Delta_s^2 \frac{\left(S_{ij}^d S_{ij}^d\right)^{3/2}}{\left(\bar{S}_{ij} \bar{S}_{ij}\right)^{5/2} + \left(S_{ij}^d S_{ij}^d\right)^{5/4}}$$
$$\Delta_s = C_w V^{1/3}$$
$$S_{ij}^d = \frac{1}{2} \left(\bar{g}_{ij}^2 + \bar{g}_{ji}^2\right) - \frac{1}{3} \delta_{ij} \bar{g}_{kk}^2$$
$$\bar{g}_{ij} = \frac{\partial \bar{u}_i}{\partial x_j}$$
$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)$$
$$C_w = 0.325$$

I OpenFOAM användes turbulensmodellen locDynOneEqEddyi men dess ekvationer hittades ej.

Bilaga 3

Bilder

Dessa bilder representerar ett tidssteg i simuleringen vilket betyder att det ej fullt ut går att jämföra med hastighetsprofilerna som visar ett tidsmedelvärde. De visar dock på trender för de olika turbulensmodellerna vilket gör att en inbördes jämförelse mellan modellerna fungerar bra att göra. Alla bilder är tagna ur x-y planet.

Skalan är klippt vid rött och blått och betyder att när det är rött eller blått kan värdena vara större (eller mindre vid blått) än vad som visas i skalan. Detta är valt så för att tydligare kunna åskådliggöra det som är intressant i figurerna.

I figur 8.3.5 visas hastigheten i x-led för både OpenFOAM och fluent för alla modeller utom LES. Dessa kan jämföras med hastighetsprofilerna i rapporten men dessa visar ej tidsmedelvärden. En viss tendens finns till att de modeller med lågt RCP visar på mer negativ hastighet under klacken (turbulens).



Figur 22 Hastighet i x-led för OpenFOAM och Fluent

I figur 8.3.6 visas ett momentanvärde för k för alla modeller utom LES. Fluents modeller visar på ett högre värde vid inloppet. Detta kan möjligen bero på den högre turbulenta intensiteten som sattes i randvillkoret. De modeller med lägre RCP visar på ett större område med lägre värde på k under klacken.



Figur 6 k för OpenFOAM och Fluent

I figur 8.3.7 visas ω för alla modeller utom LES. Simuleringarna i Fluent har ett högre värde på ω i bulken men det avtar något nedströms. En tendens finns att ω ökar under klacken med ett lägre RCP.



Figur 7 ω för OpenFOAM och Fluent



I figur 8.3.8 visas den turbulenta viskositeten v_t för alla modeller utom LES. Ett lägre RCP ger ett lägre μ_t . Notera att värdena för Fluent är lägre då dess skala är lägre.

Figur 8 Turbulent viskositet för OpenFOAM och Fluent

Figur 8.3.9 visar

$$\frac{v_{t,RCP}}{RCP} = v_{t,SST} \tag{2.12}$$

och är inget som användes i någon simulering utan ska ses som ett diskussionsunderlag. Figuren visar att $v_{t,SST}$ varierar för de olika modellerna.



Figur 9 Turbulent viskositet enl. Menter för OpenFOAM och Fluent

I figur 8.3.10 visas Gyllenram-Nilssson RCP-villkor nr 2. Detta användes inte i någon simulering (förutom RCP-filter) utan ska ses som ett diskussionsunderlag. Figuren visar vad RCP borde satts till enligt villkor nr 2.



Figur 10 Gyllenram Nilssons RCP-villkor nr 2 för OpenFOAM och Fluent

I figur 8.3.11 visas NASA:s RCP-villkor. Detta användes inte i någon simulering utan ska ses som ett diskussionsunderlag. Figuren visar vad RCP borde satts till enligt NASA:s villkor.



Figur 11 NASA:s RCP-villkor för OpenFOAM och Fluent