POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea Specialistica in Ingegneria Aeronautica

Experimental and numerical analysis of the slag entrapment in Ariane 5 solid rocket motors

Supervisor: L.Galfetti

Co-Supervisor: F.Maggi

VKI Supervisor: J.Anthoine

VKI Co-Supervisor: B.Tóth

Cristóbal Rodríguez 720532

Anno Accademico 2009/10

.

Acknowledgments

En primer lloc, voldria donar gràcies a la meva família que m'ha donat suport durant aquests ja set anys i mig de carrera. La vostra comprensió m'ha servit per a molt tot i que de vegades no ho he sapigut agraïr prou. Gràcies

A voi, i miei amici trovati durante questa più tosto lunga sperienza nel stero. Angelo, tutto quello vissuto assieme sarà difficile da dimenticare: da San Donato fino a Dublino passando per Roma. . 2011 lo dicevamo....Iaaaaaaaaaa; Nicoletta, cosa avrei fatto senza la tua comprensione. Il mio salvataggio nei momenti più bassi vissuti a Milano. Grazie davvero; Matteo, fratello, la tua fiduccia cecca in me mi ha tirato sù sempre. Grazie ; Elena, gracias por tu ayuda compañera de piso. Siempre serás mi terrona favorita; Francesi: Adrien e Julien, questa sperienza non sarebbe stato lo stesso senza di voi... Ci abbiamo aiutato tanto e questa laurea è anche vostra. Allez le bleus! Ceci, non scherzavo quando dicevo che Milano non "aveva senso senza di te".. è una grande verità; Ana, mi soporte a distancia, grácias por tus ánimos, tu confianza casi ciega en mis capacidades y sobretodo por tu paciencia infinita; Marc, sempre portare Maciacchini en el cor. Marco, il mio presidente all'ombra, rispetto!. Diego, hijo mio, un sucesor que me ha superado. E finalmente ad ogni singola persona coinvolta nella mia sperienza in ESN.

Als meus amics que desde la distància m'han entès, escoltat i donat ànims. Gràcies Albert, Jordi i Dani.

Finally I would like to thank the people that have helped me in this thesis. Bálazs I have learnt so much from you and I want to thank your ending less patience with me. To F.Maggi and L.Galfetti for understanding the difficulty to carry on with the work abroad and all the help and indications given.

Abstract

Il problema dell'accumulo di allumina è intrinseco all'utilizzo del grano propellente alluminizzatto nei motori a propulsione solida. Al Von Karman Institute questo problema è stato studiato in una riproduzione in scala del booster usato nell'*Ariane* 5 , modellando l'allumina con acqua e i gas con aria. Il numero di Stokes viene mantenuto.

Dopo il studio parametrico fatto da Tóth si è giunti alla conclusione che è possibile correlare il comportamento dell'accumulo di acqua nella cavità in prossimità dell'ugello in funzione della distanza orizzontale tra la punta dell'anello termico (inibitore) e l'ugello medesimo ($OT2NT$). Si è osservato che lo sgoccialamento dall'inibitore può influenzare pesantemente i risultati: i residui presenti erano il doppio rispetto ai valori reali.

Il primo obbiettivo della parte sperimentale è stato ridurre lo sgocciolamento con un nuovo disegno per l'inibitore. Il disegno scelto non usa materiali assorbenti ma ha un sistema di trappole ricoperte da una lamina di 24 mm. A posteriori è stato realizzato uno studio parametrico riscontrando che $OT2NT$ è ancora il parametro dominante nel fenomeno, ma lo sgocciolamento si è dimezzato avvicinandosi ai risultati noti dal caso reale.

Per quanto riguarda la parte numerica si è applicato il risolutore $twoPhase-EulerFoam$, parte di $OpenFoam$, basato sul modello dei volumi finiti. A posteriori, questo modello è stato modificato per essere capace di gestire 5 fasi disperse con le loro rispettive caratteristiche ($sixPhaseEulerFoam$). I codici elaborati sono stati verificatii confrontandoli direttamente con risultati di pubblicazioni esistenti. Si è inoltre effettuata una analisi di sensibilità per diversi parametri numerici. Per quanto riguarda la $twoPhaseEulerFoam$ si è ricavato che i risultati sono molto simili a quelli già presenti in letteratura, mentre il modello $sixPhaseEulerFoam$ presenta l'andamento attesso delle diverse grandezze ma i relativi valori sono sovrastimati ed quindi serve esclusivamente per dare una risposta qualitativa.

Per finire, i codici di calcolo presentati sono stati applicati al problema in studio. Notando che in precedenza si era osservato che il parametro più importante era $OT2NT$, 3 configurazioni geometriche per valori di $OT2NT$ di 25.69,-4.31 e -25.91 mm sono state simulate. Si è trovato che il valore dell'accumulazione

trovato numericamente mostra lo stesso comportamento che quello trovato sperimentalmente: per valori positivi di $OT2NT$, con l'ugello al di sopra della punta dell'inibitore, l'accumulazione cresce mentre per valori negativi decresce.

Abstract

The accumulation of alumina is an inherent problem in Solid Rocket Motors (*SRM*). In the *Von Karman Institute* this problem have been studied using a scale model of *Ariane* 5 booster where alumina is modeled with water. The Stokes number is maintained. After a parametric test, the relevant parameter found to be responsible for the entrapment of particles, was the horizontal distance between the tip of the isolation ring (inhibitor) and the nozzle tip. This parameter is named as Obstacle tip to Nozzle tip (*OT*2*NT*). The problem found during Tóth investigation was that a large part of the water accumulation in the cavity was due to the dripping of the inhibitor.

The first objective of the experimental part have been to reduce the dripping modifying the inhibitor design. The design found to minimize the dripping is the one with no extra absorbing material and a wider top plate (24*mm*). A posteriori, a parametric tests was done. *OT*2*NT* has been confirmed to be the main responsible while the entrapment for the new system is around a half of Tóth results.

The aim of the numerical part have been to apply an existent code, *Open−Foam twoPhaseEulerFoam* library, based on the finite volumes. Then this code have been modified in order to handle with five dispersed phases with different properties, named as *sixPhaseEulerFoam*. The codes have been verificated through a direct comparison with existing publications. Then a sensitivity study to numerical parameters has been done. For *twoPhaseEulerFoam* it has been found than the results are close to the existing ones while for the *sixPhaseEulerFoam* although the behavior of the solution is the one expected, its values are overestimated giving only a qualitatively valid answer to the problem.

Finally, *twophaseEulerFoam* is applied to the booster geometry studied in the experimental part. Due to the fact that *OT*2*NT* has been determined as the main responsible for the entrapment, three different geometries showing three different *OT*2*NT* (25.69, −4.31 and −25.91 *mm*) values have been tested. It was found that for positive values, nozzle over the inhibitor tip, the entrapment increases while it decreases for negative values (the same behavior than found in the experiments).

Sommario

La prima parte di questa tesi verte sugli aspetti teorici della problematica dell'accumulo di allumina nei motori a propellente solido, principalmente basandosi su una ricerca bibliografica dei principali ricercatori nella materia. L'evoluzione del propellente all'interno del razzo viene divisa in tre fasi: agglomerazione in superficie, evoluzione in prossimità della superficie e evoluzione lungo il razzo fino l'uscita attraverso l'ugello.

Nella prima fase si studia l'agglomerazione sulla superficie del grano propellente: si studia l'influenza della pressione, le dimensioni caratteristiche delle particelle e l'andamento della velocità di regressione. Per la seconda fase, si presentano le interazioni degli agglomerati con il gas usando il modello pubblicato da Babuk. Infine per la terza fase si presentano le interazioni basiche di agglomerati con la geometria interna del razzo e cioè con gli anelli termici isolanti (inibitori) e la cavità attorno l'ugello. Questa ultima fase riguarderà lo studio numerico e sperimentale svolto a posteriori.

La seconda parte della tesi è centrata sulle prove sperimentali realizzate al Von Karman Institute a Bruxelles durante uno stage di 14 settimane. Le prove sono state svolte nella galleria $VKI-L11$. La sezione di test rappresenta l'ultimo stage del booster del Ariane 5 rispettando il tipo di interazione tra fase dispersa e continua basata sul numero di Stokes. L'allumina è modellata con goccia d'acqua mentre i gas sono stati modellati con aria. Il primo obiettivo sperimentale è stato migliorare la galleria esistente. Negli studi precedenti fatti da Tóth si era osservato che lo studio dell'intrappolamento poteva essere falsato per lo sgocciolamento della punta dell'inibitore termico. Per evitare ciò, si sono studiati diverse nuove configurazioni e disegni per il modello di questo componente.

Una volta trovata la soluzione migliore che riduce lo sgocciolamento, si è realizzato uno studio parametrico delle diverse caratteristiche geometriche, oltre alla velocità, che potrebbero influenzare l'accumulo di fase dispersa nella cavità attorno l'ugello. Le analisi hanno evidenziato che il parametro più importante è la distanza orizzontale tra la punta dell'inibitore all'ugello, chiamato $OT2NT$. I valori positivi indicano che l'ugello è posizionato al di sopra della punta dell'inibitore. Maggiore è $OT2NT$ (valori positivi), maggiore è l'intrappolamento.

La terza parte è quella numerica, dove si cerca di confrontare un modello numerico con i risultati trovati sperimentalmente. Il primo passo è stato validare un codice già esistente all'interno di *OpenFoam*, chiamato *twoPhaseEulerFoam*, che riesce a trovare una soluzione per il flusso con una fase dispersa e una continua basato sull'approccio a volumi finiti euleriano. Questo codice è stato modificato per riuscire a gestire fino a cinque fase disperse con le relative caratteristiche (viscosità, diametro e densità). Il nuovo codice verrà chiamato *sixPhaseEulerFoam*. Come punto previo alla validazione del codice si è studiata la sensibilità di entrambi codici alla griglia di calcolo e al numero di Courant. Il modello di verifica è un gradino dove la fase dispersa è acqua mentre che il continuo è olio. Il numero di Reynolds , considerando come grandezza caratteristica l'altezza del gradino, è 500.

Per il risolutore *twoPhaseEulerFoam* si è trovato che i risultati ottenuti sono molto simili a quelli trovati nelle pubblicazione già esistenti. Per *sixPhaseEulerFoam* invece si è evinto un andamento coerente delle diverse proprietà calcolate (frazione volumetrica, pressione e velocità), ma con dei valori sovrastimati rispetto a ciò che dovrebbe essere. Si è concluso quindi che *sixPhaseEulerFoam* potrebbe essere usato esclusivamente in modo qualitativo.

Come parte della validazione si è anche effettuato un studio di sensibilità di entrambi i codici ai coefficienti di portanza e di massa virtuale. Si è ricavato che nella vicinanza del gradino la frazione volumetrica è influenzata fortemente del valore di questi coefficienti mentre la pressione e le velocità ne vengono influenziate debolmente. Per i punti a valle del gradino e vicini alla fine del dominio computazionale non si trovano forti differenze, *SixPhaseEulerFoam* mostra però un comportamento divergente quando vengono impostati alti valori di questo coefficiente (uguali a uno).

Infine, *twoPhaseEulerFoam* viene applicato al problema dell'accumulo di allumina. Si prende come dominio computazionale la sezione di test presentata nella parte sperimentale per tre diversi *OT2NT* (−25.91,−4.31,−25.91 *mm*) visto che era stato rilevato come il parametro più importante. Il flusso viene considerato laminare. I risultati numerici mostrano lo stesso comportamento evidenziato sperimentalmente ma con un valore diverso per una delle tre configurazioni studiate. Questo è giustificabile con il fatto che il sistema di valutazione lo sgocciolamento nella parte sperimentale potrebbe falsare i risultati. Anche la mancanza di un modello di turbolenza e di un modello di accoppiamento aero-acustico potrebbero modificare i risultati presentati.

List of Figures

List of Tables

Contents

Part I

The slag entrapment problem in solid rocket motors

Nomenclature

Al Aluminum

AP Ammonium perchlorate (NH_4ClO_4)

C/F Coarse to fine ratio

DNS Direct Numerical Simulation

D_{43} Mass medium diameter of agglomerates (particles bigger than 43um)

D_{43}^{AP} Mass medium diameter of AP particles

d_{43} Mass medium diameter of smoke oxide particles

$fm(D)$ mass function of agglomerate size distribution density

L_s Length for quenching

$KLLEF$ Kinetically limited leading edge flame

P Pressure

P^{**} Agglomeration critical pressure

R_b Burning rate

SL Skeleton Layer

SOP Smoke Oxide Particles

SRM Solid Rocket Motor

T Temperature

Z_m^a Share of initial Al used to form agglomerates

Z_m^{ox} Share of initial Al used to form oxide on the agglomerates

Z_m^{ox} Share of initial Al used to form SOP

Z_m Share of initial Al used to form unburned Al in the agglomerates

Chapter 1

Statement of the problem

Every year an increasing number of satellites are launched to the space. Thus there is a growing need to have reliable launch systems, which are able to carry the maximum possible payload. From 1996, the European Space Agency (*ESA*) is using the *Ariane* 5 launcher (Fig.1.1). It has been upgraded since the first launch in successive versions: *G*, *G*+, *GS*, *ECA* and from March 2008 the *ES* version.

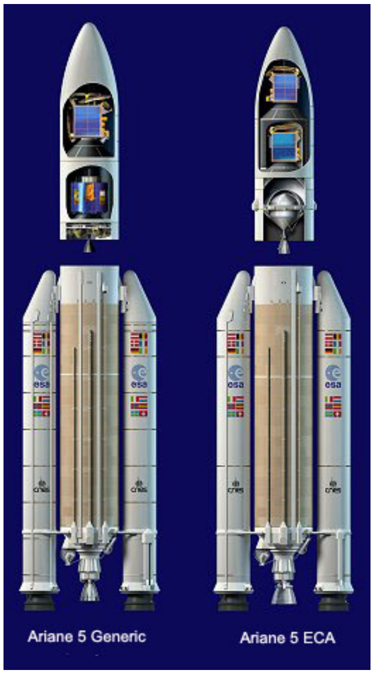


Figure 1.1: ARIANE 5 LAYOUT [21]

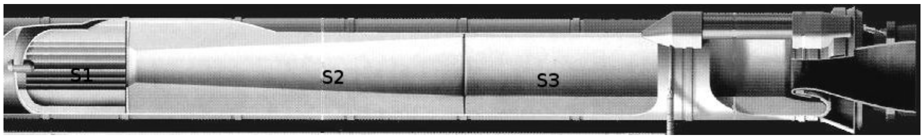


Figure 1.2: MPS P230 INTERNAL LAYOUT [20]

The first stage of the *Ariane* 5 launch system versions is composed of two main parts. The first one is called Cryogenic Main Stage (*EPC*), while the second main part is composed of two solid rocket motors (*SRM* or *EAP*) on both sides. From now on, the technical data will reflect the *ECA* version.

The *SRMs* are 3 meters in diameter and 31 meters tall. Each one weights 37 ton when empty, contains about 230 ton of solid propellant. It is capable of giving 6470 *KN* with a specific impulse of 275 *s* and a burn time of 129 *s*. Working together, the two *SRM* are giving the 92% of the total trust at lift off and can be equipped with parachutes, in order to recover them from the sea for examination porpoises. To shorten the overall length, the nozzle of the rocket is submerged in the last part of the solid propellant. For the *Ariane* launch system the *SRM* is also known as *MPS P*230. The internal layout for the *MPS P*230 can be seen in Fig.1.2.

As it can be seen, the *SRM* is divided in three segments of propellant grains (*S*1, *S*2, *S*3). *S*1 has a star section in order to enlarge its surface and provide an additional thrust during lift-off. This segment operates for the first 25 *s*. *S*2 and *S*3 are conical segments that burn radially increasing their surface. *S*2 and *S*3 burn for 130 *s*.

Between the different segments, there are thermal protection rings (from now on called, inhibitors). The propellant material contains aluminum powder (18%) in order to increase the enthalpy of the combustion. During the chemical reaction liquefied alumina Al_2O_3 droplets are formed. This phenomena will be specifically explained in the following sections.

During the combustion the regression of the solid propellant surrounding the nozzle leads to a formation of a cavity (Fig. 1.3). On the other hand, alumina droplets that are generated during the combustion may accumulate in the cavity next to the nozzle producing alumina puddle (slag), as it can be seen in Fig.1.4, which creates dead weight that reduces thrust performance.

The aim of this project is to study and discuss about the so called slag accumulation of the *MPS P*230.

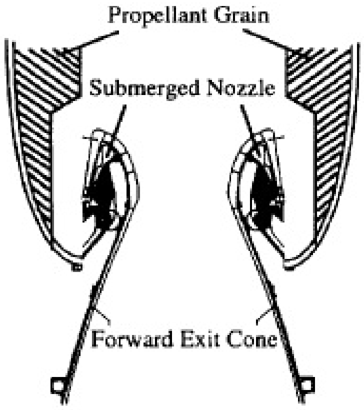


Figure 1.3: CAVITY NEXT TO THE NOZZLE [23]

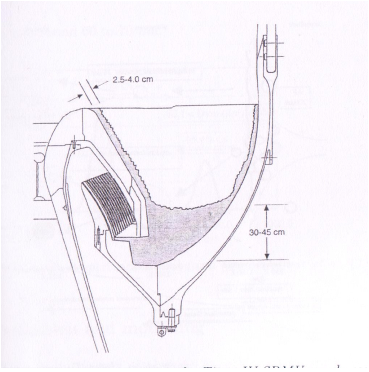


Figure 1.4: SLAG ACCUMULATION IN TITAN IV SRM [22]

Chapter 2

Physical phenomena involved

In order to increase the enthalpy of combustion in Solid Rocket Motor (SRM), metal particles are added to the propellant. The metal introduced into the propellant formulation is Aluminum. In table. 2.1 it can be seen the composition of the propellant for $Ariane$ 5 SRM used by ESA .

Model	Binder	Metal (Al)	Oxidizer (AP)
$ARIANE$ 5 230MPS	14	18	68

Table 2.1: FORMULATION OF PROPELLANT FOR SRM

As a result of using Aluminum, drops of alumina are produced. These could be entrapped in the cavity surrounding the nozzle or exhausted forming what is known as slag. The slag could produce thrust oscillations and a reduction in the performance of the rocket.

The objective of this chapter is to present the phenomena of formation of condensed products on the propellant surface (phase 1), its following behavior on the gas phase nearby the surface (phase 2) and its evolution on the gas flow along the SRM (phase 3). In order to do that, information have been obtained by different papers presented during the last decades of research of these phenomena. These studies are both experimental and numerical. This chapter have been aimed to be a starting point, a presentation of the current physical phenomena that will be further discussed.

In Fig.2.1 a graphical explanation of the process in phase 1 and 2 presented by Duterque in [1] is shown. On the lower part there is the temperature (T) distribution.

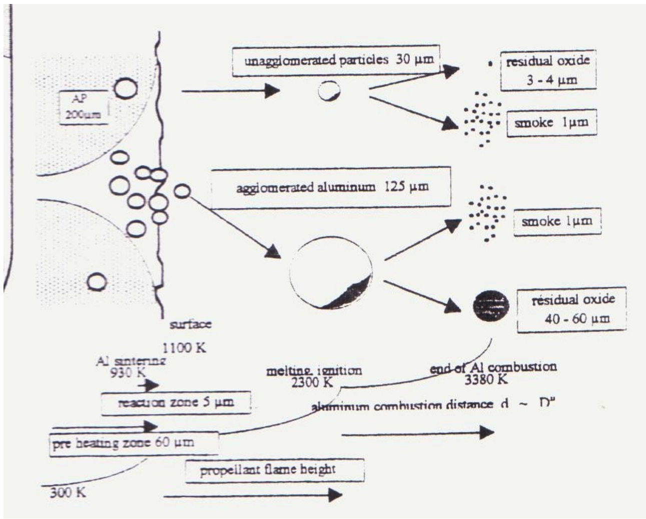


Figure 2.1: SCHEME OF AL AGGLOMERATION AND AL COMBUSTION [1].

2.1 Agglomeration on propellant surface

The propellant is a matrix of an inert binder (polybutadiene), and oxidizer (AP) and a metal phase (Al). During the propellant combustion some Al particles agglomerate with other particles while others will leave the surface without agglomerating. There is a list of parameters that influence whether the particles agglomerate or not. These will be discussed afterward. Once Aluminum, either agglomerated or not, leave the surface, it burns in the gas phase. The agglomerate particles are formed by Al and Alumina (Al_2O_3)

The first point is to present an analytical model for the agglomeration of Al on surface. Cohen [2] had presented the named "pocket model".

2.1.1 Pocket model

Large AP particles are distributed in the propellant matrix. The spaces or "pockets" between the largeAP particles are filled with binder, finer AP particles and Al. These spaces define "free volumes". Once the propellant formulation is known it is possible to define the Aluminum mass in each free volume. The main hypothesis of the pockets model is that the fraction agglomerated is proportional to the amount of Al that melts.

Each Al particle goes across the thermal gradient in the binder. The temperature of the Al particle differs from the binder one. The Al suffers from a lag when reaching the binder temperature. Due to conduction, this lag time is function of the particle size (proportional to the square diameter). This time has to be compared with the residence time of the particle on surface. Al particles that does not stay on surface (residence time) enough time does not melt and could not agglomerate; if the contrary happens, condensed products or agglomerates will be found on surface.

The main problem of using this model is the lack of understanding of *Al* in the propellant. So this is a qualitative model.

2.1.2 Interpocket correction

Sambamurthi et al. [3] in their experiments show that the agglomerates sizes were ofter larger than the one resulting from the combination of all the pockets individually. So they defend that the Cohen model is not enough to explain the accumulation process: the pockets defined by the array of *AP* are connected, the pockets are not closed. This phenomena will produce "super-agglomerates" or bigger than usual condensed products.

2.1.3 Skeleton layer

Experimentally, it has been found that *Al* particles leave the burning propellant surface just after being ignited. So there is the need to model the upper part of the propellant while burning. This model will complement the pocket assumption by Cohen.

Babuk et al [4] state that it is possible to distinguish a condensed phase zone where it is possible to find *Al*, oxide and carbonic elements (these elements are a result of the binder combustion). The name of this zone is the Skeleton Layer. For *AP* based propellants, the skeleton layer consists of the melted original *Al* particles coated with solid oxide films and bounds together where this film is broken. In Fig.2.2 the skeleton layer surface layer can be graphically seen. It is stated that the *Sl* formation takes place only within pockets. The inter pockets bridges supply into the gas phase *Al* particles not involved in agglomeration.

2.1.4 Flamelets

Once it has been explained the formation of agglomeration on surface that happens between large *AP* particles, there is the need to explain what happens on them.

Sambamurthi and Price [3] presented a model to explain the flamelets and its structure is function of the *AP* size. It is based on the fact that on each *AP* particle a stoichiemtric surface grows into the mixing flow. For the larger oxidizer particles, the size assures to form a diffusion flame. This fact contributes to the buring of condensed products on surface. Instead, the mechanism is more complex for finer oxidizer particles.

For the finer *AP* particles, it has been found that the flamelets close over them. This closing region is named as stoichiemetric tip.

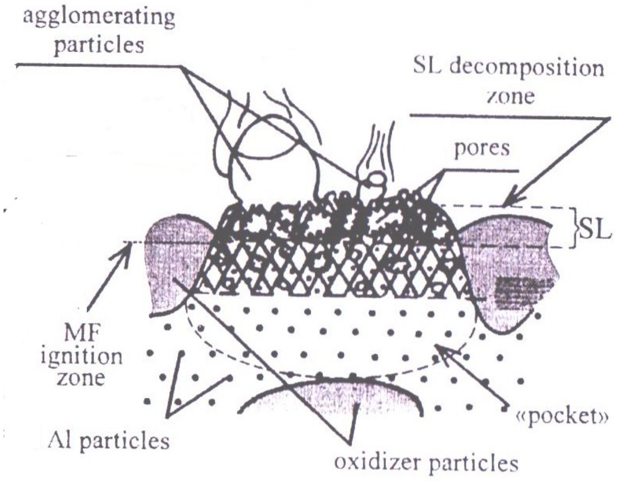


Figure 2.2: SKELETON LAYER LAYOUT [4].

2.1.5 Parametric study for surface agglomeration

Once the phenomena on surface have been described qualitatively, the effect of different parameters on the formation of condensed products will be presented. The main parameter studied by experts in different publications have been: pressure, propellant composition (Al and AP particles particles size distribution), burning rate. A possible physical explanation for each behavior will be given.

Generally speaking, the experimental technique in the papers considered consists in burning a sample of the propellant inside a Constant Volume Bomb (CVB) and then quenching the results of combustion with an inert liquid such as alcohol or an inert gas . The phenomena that is studied is the agglomeration on surface but as long as the particles travel though the environment before being quenched, an evolution of the surface agglomerates exists. So there is a coexistence of the first phase (agglomeration on surface) of the problem with the second one (near surface gas phase) in the studies. This could result in obtaining less agglomerates than expected, because they can be burnt after leaving the surface.

An example of experimental technique for the surface agglomeration is shown in Fig. 2.3 .

2.1.5.1 Pressure

Babuk et al. [4] defined a general behavior for the parameters of agglomeration in function of pressure. When the pressure exceed a certain value, $P**$, there is a rapid decreasing of Z_m^a (Fig. 2.4). The mass medium diameter of agglomerates

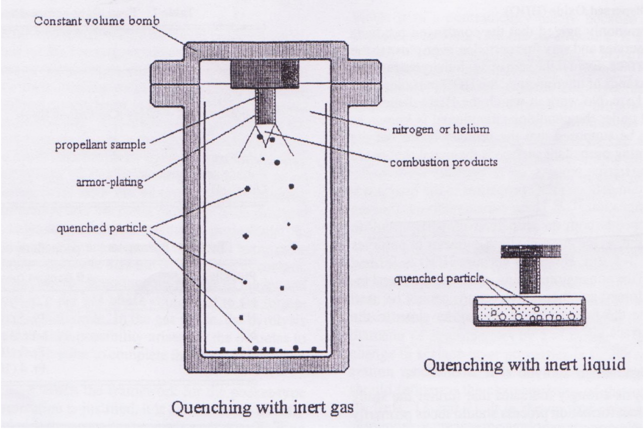


Figure 2.3: Experimental method used in [4].

follow the behavior presented in Fig.2.5 .

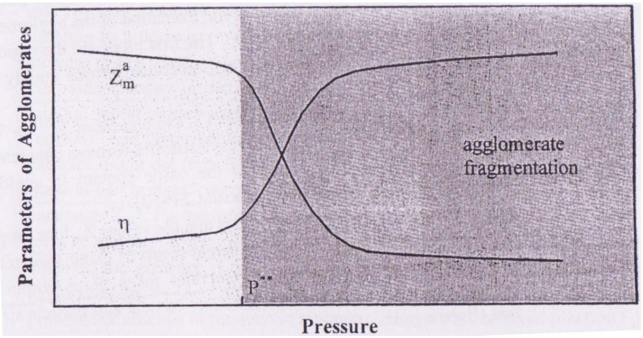


Figure 2.4: generalized pressure dependence [4].

At low pressure, the main phenomena of formation is Pocket agglomeration β, the Cohen model for agglomeration on surface. For pressures higher than the critic one, a multimodal phenomenon has been found. The F branch, mode γ, is formed by the inter pocket-bridge model. Finally E branch, mode α, is due to the pre pocket agglomeration. The concept of pre-pocket have not been presented before. It is an intermediary step in the pocket model. It consists on the particles that feed the gas phase before the pockets they belong top are exhausted.

Experimental results are shown in Fig. 2.6. It can be seen that there is a decreasing behavior of the diameter of the agglomerates with the pressure. For the lower pressure values, there is not flamelet established by the finer particles of AP, so the Al accumulated on the surface could not be burnt. Instead, for

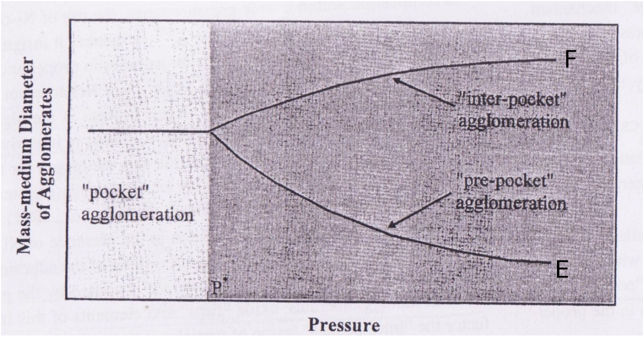

Mass-medium Diameter of Agglomerates
F
"inter-pocket" agglomeration
"pocket" agglomeration
"pre-pocket" agglomeration
E
P*
Pressure

Figure 2.5: DEPENDENCE OF D43 WITH PRESSURE AND FORMATION MECHANISM [4].

the higher values of pressure finer particles of AP could establish a flamelet and then burn the accumulated Al on surface.

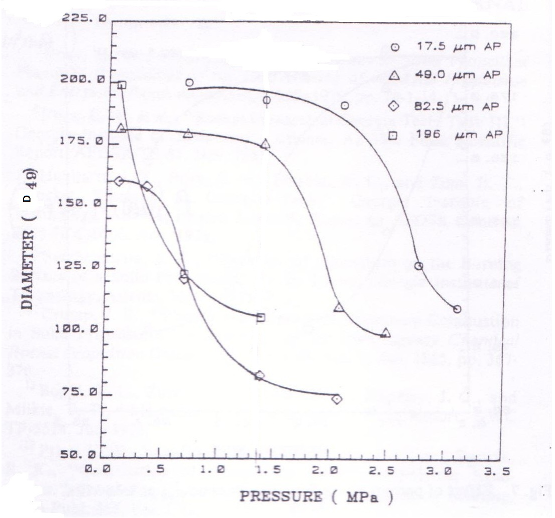


Figure 2.6: EFFECT OF PRESSURE ON MASS AVERAGE AGGLOMERATE SIZE [3]

2.1.5.2 Aluminum particle size

In [6], Dokhan et al. determined experimentally the effects of Al particle size on the burning rate. In following paragraphs the effect of the burning rate on accumulation will be explained widely, but as a way of introduction, the residence time is inversely proportional to the burning rate. So the longer the burning rate is, the smaller the accumulated particle is because the particle does not have enough time to agglomerate.

For unimodal Al distributions the experiments show that the smaller the *Al* particle is, the faster is the burning rate (for constant bi-modal *AP* formulation). For bimodal *Al* distributions (Coarse to ultrafine *Al* ratio), it was found that an increase of *Al* increases the burning rates. So, it could be concluded that to obtain smaller agglomerates it is interesting to use finer Aluminum particles. But Dokhan et al remarked the problems of finer *Al*:

1. Fine *Al* is expensive.
2. Propellant processing with finer *Al* is more difficult: mechanical characteristics are usually poor.

The physical explanation for the *Al* behavior is due to the fact that using finer *Al* increases the heat exchange surface and enhances near surface heat release.

2.1.5.3 Oxidizer particle size

In order to study the effect of the oxidizer formulation in the accumulation, it is important to remark the correlation of this formulation with pressure. Considering a constant *AP* formulation (coarse to fine *C/F AP* ratio), for lower pressures the agglomerates will be bigger due to the impossibility to establish flamelets on the finer *AP* particles.

In Fig.2.6 the *C/F AP* is 8/2. Considering a constant pressure, it is found that the smaller the finer *AP* is, the bigger the diameter of the agglomerate is. If the parameter is the percent of fine *AP*, results in Fig.2.7 are shown.

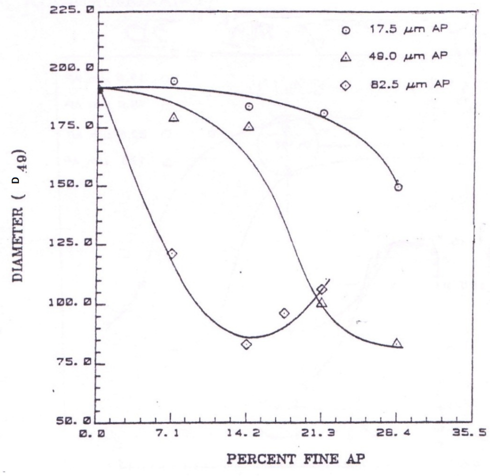


Figure 2.7: EFFECT OF PERCENT FINE AP IN PROPELLANT IN AGGLOMERATED MASS AVERAGE.[3]

It can be seen that there is a decreasing behavior of D_{49} with the percent of fine AP except for the 82.5 μm fine AP sample. This fact have been explained in [3] as a reduction of fuel excess in environment when increasing the percent of fine AP Fig.2.8. This reduction brings the diffusion flamelets closer to burning surface so it is easier for the agglomerates to be burnt on surface. For the 82.5μm fine AP increasing behavior of D_{49} when increasing percent of fine AP, kinetically limited leading edge flame ($KLLEF$) position is due to retract abruptly to a more remote location causing less favorable condition for surface early ignition of agglomerates.

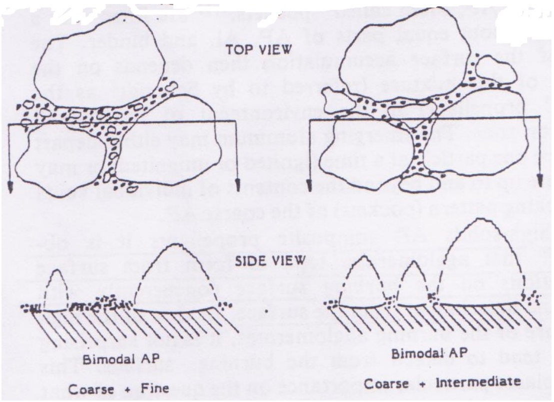


Figure 2.8: EFFECT OF THE AP PARTICLE SIZE IN THE AGGLOMERATES IGNITION [3].

2.1.5.4 Burning rate

In [1] Duterque present a experimental study of Aluminum agglomeration in SRM using two propellant samples. The difference between them is the distribution of C/F AP and the test P . There is a clear inverse behavior (Fig. 2.9) between the burning rate (R_b) and the residence time and agglomerate diameter.

The slower the burning develops , the bigger the residence time is. The globules or agglomerates will have more time to merge together and form bigger agglomerates.

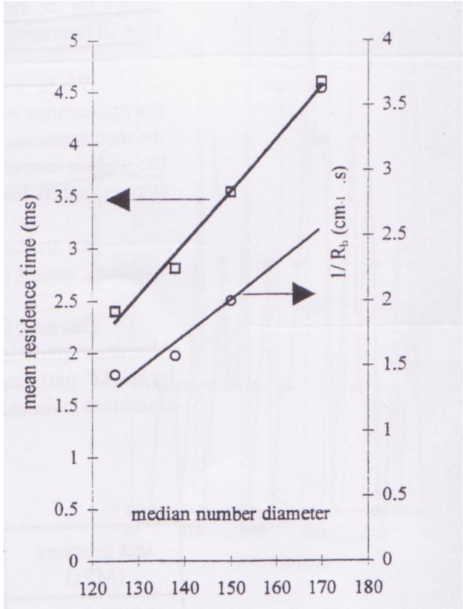


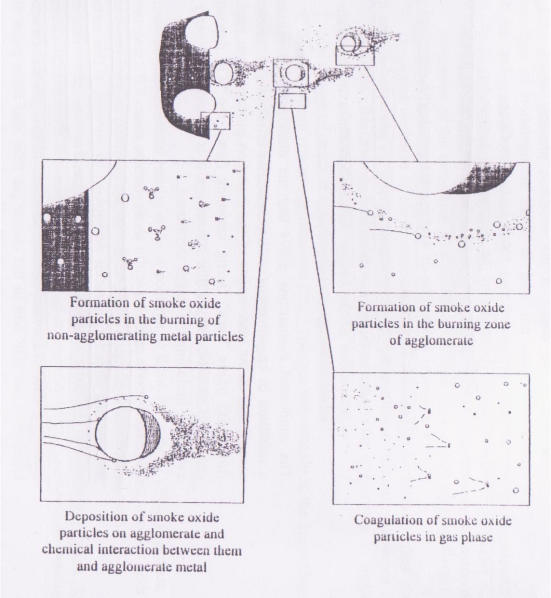
FIGURE 2.9: BEHAVIOR OF DIAMETER WITH RESIDENCE TIME AND BURNING RATE [1].

2.2 Evolution of agglomerates near the surface

On the surface, free Al particles and agglomerates (Al $+Al_2O_3$) can be found. Free Al burns and leaves the surface as a residual oxide (Alumina) or as a smoke particles (SOP). Instead the agglomerates leave the surface and suffer from a combination of physical and chemical transformation. The main point of this section is explaining the evolution of condensed products in the gas phase. Characteristic sizes and temperatures can be seen in Fig. 2.1.

In [7] an explanation for the formation of SOP and evolution of agglomerates in the gas phase is given. As for the first phase studies, the experimental technique was based in a CVB. To simulate the flow, the propellant samples were burnt inside cylinder tubes. Products were quenched at different positions in order to obtain their evolution. So the main parameter during the experiments is the length on quenching (L_s). In order to remove completely agglomerates from the surface an active plate was used to hold the propellant sample. In order to avoid the deposition of products in gas phase on the walls of the cylinder, they were treated. Finally, quenched samples are chemically analyzed.

It was found that the share of Al used to form SOP increases with L_s while the share of Al and share of Al used to form oxide in the agglomerates (Z_m and Z_m^{ox}) decrease. The mass share of oxide in agglomerates increases. Finally size of agglomeratesD_{43} is constant with L_s. Instead SOP particles grow along the flow. The conclusion presented by Babuk et al. in [7] will be now introduced. In order to support the explanation Fig. 2.10 and Fig. 2.11 are used.



Formation of smoke oxide particles in the burning of non-agglomerating metal particles

Formation of smoke oxide particles in the burning zone of agglomerate

Deposition of smoke oxide particles on agglomerate and chemical interaction between them and agglomerate metal

Coagulation of smoke oxide particles in gas phase

Figure 2.10: DIAGRAM OF EVOLUTION FOR SOP [7].

The sources for *SOP* are the products of non agglomerated *Al* and the combustion of agglomerate particles in the gas phase. But *SOP* particles could also deposit on the surface of bigger agglomerates or coagulate among other *SOP* particles. The characteristic size for SOP is less than 1 μm.

Instead for the evolution of agglomerates it has been found the existence of a *SOP* trail. The *SOP* travelling free can deposit on the agglomerate surface. In the meantime the chemical reaction between *Al* and Al_2O_3 goes on.

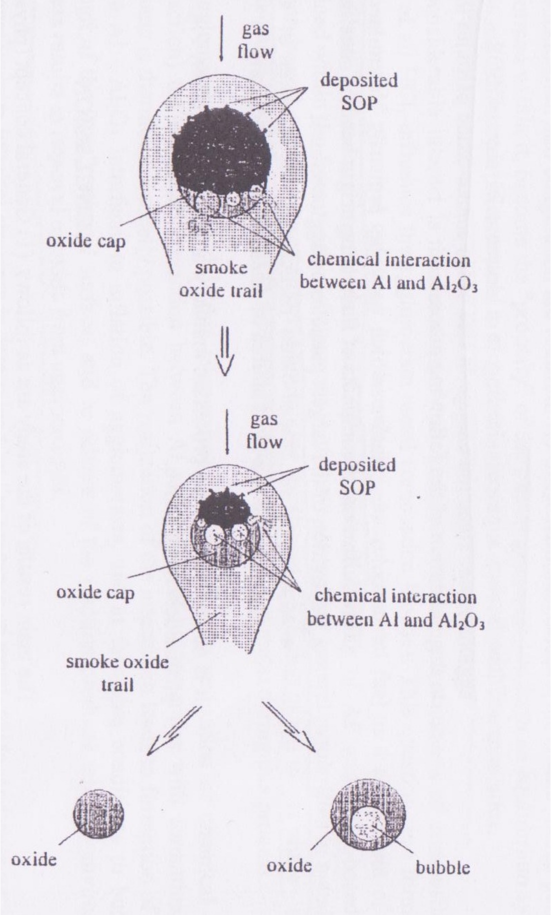


Figure 2.11: DIAGRAM OF AGGLOMERATES EVOLUTION [7].

2.3 Two phase flow along the SRM

Once the particles are released from surface (phase 1) and evolute nearby the surface (phase 2), particles travel through the *SRM* interior towards the exit or nozzle. The flow main characteristic is the two phase flow nature. While traveling towards the exit, particles may interact with the environment. The main interactions are with:

1. inhibitor
2. cavity

The flow field changes abruptly due to an obstacle: the inhibitor. It leads to the formation of vortex that are released from the tip of the inhibitor. This vortex

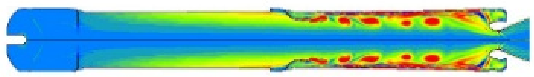


Figure 2.12: inhibitor unstationary instabilities for a mps 230 [20]

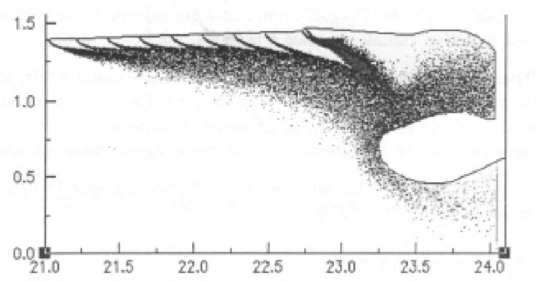


Figure 2.13: particles behavior next to the cavity 95 s after firing[8]

could interact with the acoustic modes producing acoustic instabilities in the pressure that could lead to failures (Fig. 2.12). This phenomena can be called vortex shedding-acoustic coupling.

Instead, for the cavity discontinuities some drops of Alumina could be entrapped by the flow field and carried to the cavity that is being formed during the regression of combustion (Fig. 2.13). As presented before, the accumulation of slag produces a loose in the thrust to weight ratio and then the rocket performance. During the firing tests on $MPS\ P230$, it was found that the slag mass 2070 kg for each rocket [13].

Different studies have shown that this two phenomena could be coupled and are the main source of discontinuities and irregularities in the SRM. These phenomena will be further discussed in the following points.

2.3.1 Vortex shedding and acoustic interaction

Various numerical and cold flow experimental studies have been presented to model the interaction between the vortex released from the inhibitor and its interaction with the different acoustic modes. The main objective of both kind of studies is to determine the fluctuating flow field due to the unstationary nature of this phenomena.

In [24] Fabignon and Gailliègue present a numerical algorithm using two approaches. The first one is based in solving the unsteady compressible Navier Stokes equations with a space averaging procedure while the second approach is based in solving the unsteady compressible averaged Navier Stokes equations with a low Reynolds turbulent model. The results pointed that the level of coherent structures coming from the vortex shedding is greater than the turbulent fluctuations. It was also found a strong dependency between the turbulent boundary conditions and the numerical solution.

In [25] Lupoglazoff and Vuillot present a model to solve a Direct Numerical Simulation based on the geometry of a cold-flow set up. This model presents the porous injecting surface and it was found that this injection was able to modify the acoustic oscillatory modes.

The past two studies did not consider geometry including the cavity next to the nozzle. The effects of the nozzle cavity have been studied, numerically and experimentally, by Anthoine, Buchlin and Guéry in [26]. With the cavity, the vortices released from the inhibitor tip interact at the cavity with the velocity fluctuations induced by the cavity volume producing large pressure oscillations.

2.3.2 Slag cavity entrapment

At this point, the state of the art of the study of the slag cavity entrapment will be presented.

Using quasi-steady Eulerian Lagrangian methods, neglecting the combustion and the coupling between phases is not considered, Cesco et al. (1997) [8] highlighted the importance of the applied capture rules, the droplet diameter and the turbulence level.

Several numerical simulations were done to asses about the effect of the flow field on the Alumina droplet combustion: Orlandi and Fabignon (2000) [9]] and Lupoglazoff et al. (2000) [10].

The Capture rules were determined experimentally by Vardelle et al. (2000) [11]. The tests consisted of high speed imagining of alumina droplets impinging on a hot plate and on a liquid alumina film. This phenomena was modeled numerically by Zaleski and Gueyffier (1998) [12] using $2DDNS$ simulations.

The second phenomenon to be studied was the formation and evolution of the slag. Reynolds Averaged Navier Stokes ($RANS$) simulations were carried out by Pevergne and Le Helley (1998) [13] and Cesco et al. (1997)[8]. The methodology was based on calculating the steady flow and the turbulence levels. Then, alumina droplets were injected at discrete locations. Using capture rules, it was possible to determine whether the droplet was being accumulated in the slag pool or escaped through the nozzle.

Unsteady simulations were performed by Godfroy and Guery (1997) [14] concluded that the vortices of the flow could strongly influence the slag accumulation process due to the centrifugal forces acting on the droplets inside the core. The larger droplets are carried at the periphery of the rotating structures. After repeating the simulations for different droplets diameters (d_p), the slag accumulation was found to be proportional to the square of the droplet diameter.

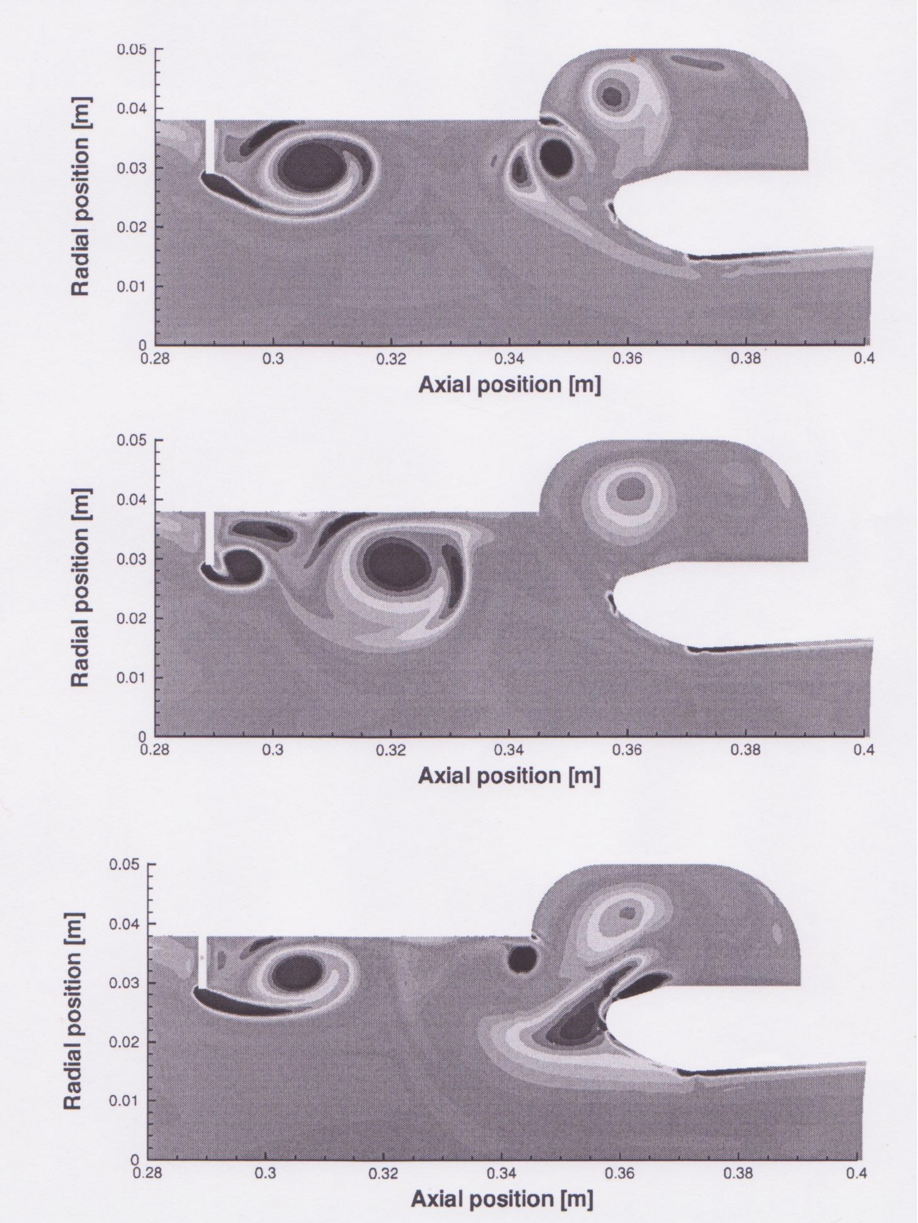


Figure 2.14: SEQUENCE OF VORTICITY FIELDS AT DIFFERENT TIMES[26]

Le Helley et al. (2002)[15] performed numerical simulations with a more refined model, where the pressure oscillations in the motor were considered. It was found that the oscillations affected strongly the accumulation. A priory, it was though that the 100% of the slag comes from the last segment, while this simulation showed that a 23% of the slag may come from the second segment. That proves the suction effect originated by the vortex.

Finally in 2002, Dupays [16] pointed out that the slag accumulation process and the instability of the internal flow field are strongly dependent.

Nowadays, recent numerical tools are being used in order to analyze the slag accumulation phenomenon. Wirzberger et al. (2005) [17] used a similar methodology to the one used by Pevergne and Le Helley (1998) [15]. The capture rule was simplified: all the droplets that were not leaving through the nozzle were considered to be entrapped in the cavity. With this simplification, the slag accumulation rate is overestimated with respect to the static ring tests. However, the tendency was predicted correctly.

Other simulations were carried out for the $RSRM$ Shuttle Booster by Najjar et al. (2006) [18] using a $3D$ Euler type with LES turbulence modeling. The dynamics of the combustion was fully considered. The droplets were tracked in a Lagrangian way and as long as the combustion process is modeled the alumina smoke is produced. With the $2D$ analysis at the moment $t = 100\ s$ it was found the existence of recirculation zones in the head end, the cavity and between the nozzle head and the rearward inhibitor. Impacts of the largest droplets ($dp > 100\,\mu m$) were reported on the nozzle exit cone while the impacts of the smaller ones ($d_p < 25\ \mu m$) were reported to happen on the nozzle exit. The limited integration time during that simulation prevents to make a detailed analysis of the droplet behavior.

As an advise for future simulations, Anthoine et al. (2002) [19] recommend $3D$ unsteady simulations of the SRM flow, which allows to consider the pressure oscillations phenomenon as well.

Nowadays, the slag accumulation study got a lower priority due to the fact that the pressures oscillations were considered to have a higher risk. Despite all the studies that have been carried out about the slag, there is still a need to fully understand the phenomenon.

Balazs Tóth (2008)[20] further investigated the slag accumulation process at the Von Karman Institute (VKI) facilities. His aim was to characterize the slag accumulation process in a simplified model of the MPS P230 SRM using primarily optical experimental techniques : Level Detection and Correction ($LeDaR$) and Particle Image Velocimetry (PIV).

Parametric tests were done with $LeDaR$ to determine the most important factors during the slag formation. It was found that the most relevant parameter was the transverse distance between the inhibitor tip and the nozzle tip ($OT2NT$). Others relevant parameters were the air velocity, the inhibitor size and the nozzle throat opening.

Statistical and instantaneous flow properties were studied with PIV measurements and after compared with an unsteady LES numerical method.

The consequence of the droplet diameter (d_p) was studied through PIV measurements and CFD simulations. The conclusion was that the smaller droplets manage to follow the main stream flow (air-phase) while the bigger ones are not able to follow the flow, due to their inertia, and are carried into the cavity.

Another factor was identified to contribute to the accumulation process: the dripping from the inhibitor. In the tests that were carried out the most part of the slag come from the inhibitor. Numerical simulations with more real geometry were performed. Their results suggested that this dripping might not be the main mechanism in the real configuration.

Chapter 3

Objective

The objective of this thesis will be to study the slag entrapment problem in the cavity next to the nozzle in the solid rocket motors based in the case of the *Ariane* 5 *MPS P*230. This study will be both experimental than numerical, in order to give a global view of the problem.

A parametric study for the entrapment will be presented using a cold-flow bi-phase wind gallery property of the Von Karman Institute. These experiments are the continuation of B. Tóth previous studies.

A numerical model will be implemented in *OpenFOAM CFD* Open source software. The two phase flow will be modeled modifying the solvers currently available inside the software distribution.

Part II

Experiments

Nomenclature

α Volume fraction

ρ Density

d_p Droplet diameter

f Focal length of the camera optics

$f\#$ Focal number (speed/stop number) of the camera optics

h Inhibitor height

L_i Inhibitor position

$O2NR$ Obstacle to Nozzle Ratio Li/h

$OT2NT$ Obstacle tip to nozzle tip

o Nozzle opening

w Cavity width

HSC High Speed Camera

$Meas$ Measurement

SRM Solid Rocket Motor

ST Stokes number

U_0 Nominal velocity

VKI Von Karman Institute for Fluid Dynamics

Chapter 4

Test facilities

4.0.3 Test Section

The objective will be to represent the geometry, respecting the section ratios, and characteristics, as the presence of the thermal insulators, of the SRM P $230/MPS$ of the $Ariane$ 5. As long as the SRM is continuously changing geometry due to the propellant burnt, it was decided to model the motor when the fifty percent of the propelant has been burnt. So only the second and the third stage have been considered.

The thermal insulation between stages, the inhibitor, is considered in the model. It is pressumed to be responsible of the vortex formation and shedding in the flow. This element will be further presented.

At the lower part of the test section (Fig. 4.1), there are the nozzle and the cavity. Its geometry could be modified by sliding its walls. In order to avoid leaks, silicone has been used in the joints.

There is one element of the test section that does not exist in the real case: the splitting plate. It is used to create a $2D$ flow pattern. Due the possible interaction between the inhibitors and this plate, the inhibitors height is slightly smaller than it should be. The main characteristics of the test section are compared with the real case in the following table 4.1 .

The test section is attached to the convergent of the $Von\ Karman\ Insitute$ $L-11$ wind tunnel (Fig. 4.2), so its section is $200x200$ mm^2 . The main body

	Model	Real case
Mach number	0.04	0.1
Flow pattern	axial $2D$	axisymmetric $3D$
Chemical reaction	Not considered. Cold gas facility	Hot gas phenomena

Table 4.1: TEST SECTION VS REAL CASE

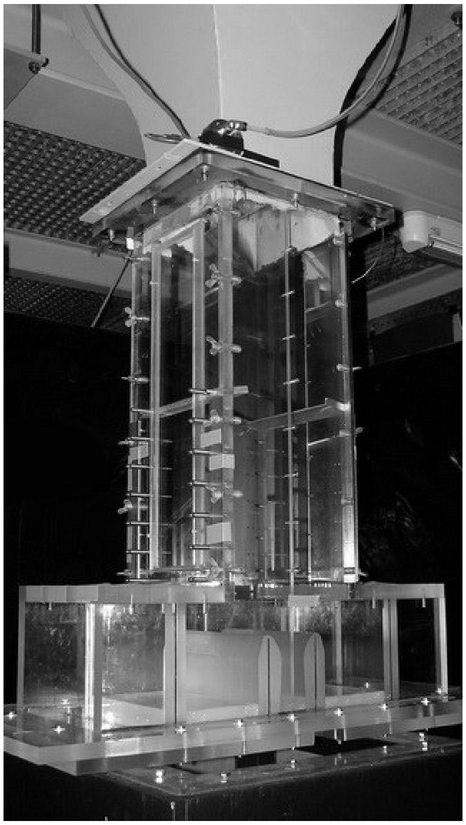


Figure 4.1: TEST SECTION

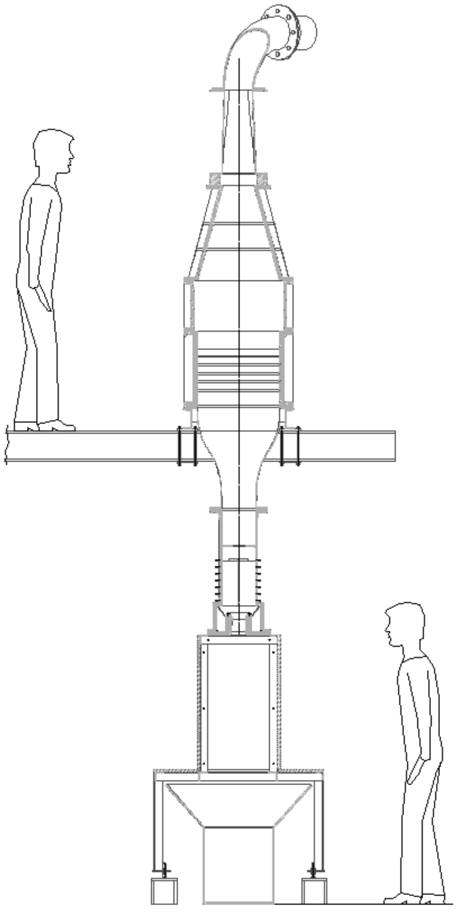


Figure 4.2: VKI L-11 WIND TUNNEL

and one of the sidewalls have been made with *Plexyglass* while the remaining sidewall has been made of steel.

4.0.4 The wind tunnel

The wind tunnel used is the $Von\ Karman\ Insitute\ L-11$ facility (Fig. 4.2), installed vertically. The facility is divided in two floors. The stagnation chamber and a part of the convergent is upstairs while the test section is downstairs. The tube that is connecting the fan with the stagnation camber is fixed in a permanent position, in order not to change the flow behavior.

The air is being blown by a fan which has the following specifications shown in table 4.2.

In the stagnation chamber there is installed the spray that feeds with water

Manufacturer	$Ingersoll - Rand$
Horse power	130 hp
RPM	3800

Table 4.2: FAN SPECS

the test section. This water represents the alumina droplets. The position of the spray is perpendicular to the stream. The spray is fed with a pump installed downstairs. Changing the aperture of the bypass valve it is possible to control the water flow rate to the spray. The pump is fed with a 200 $liters$ tank installed on the upstairs floor.

The criteria to choose the spray had been the conservation of the Stokes number (St) and the volume fraction (α). The Stokes number defines the coupling between the dispersed phase and the continuum (eq. 4.1) andis defined as the ratio between the characteristic droplet response time (τ_p) and the turbulent time scale of the flow (τ_f). Depending on St and α the characterization of the interaction between and dispersed phase can be defined (Fig. 4.3)

$$St = \frac{\tau_p}{\tau_f} = \frac{\rho_p d_p U_0}{18\mu_f L_0} \tag{4.1}$$

where ρ_p is the density of the dispersed phase, d_p the diameter of the dispersed phase, U_0 the nominal speed, the μ_f flow dynamic viscosity and L_0 a characteristic length. For the studied case, L_0 is the height of the inhibitor or h. The liquid chosen has been the water for its manipulation safety and simplicity. For the test facility, the nominal speed is 10 m/s

For the original MPS $P230$ engine the St number is 6.01 and α is equal to $2.2 \cdot 10^{-4}$. If we plot this point in Fig. 4.3 it lays on the two way coupling-enhancing turbulence zone. So, in order to model correctly the phenomena, the chosen spray will have to lay in this zone. The final spray has a St number of 10.36 and a volume fraction of $1.5 \cdot 10^{-4}$.

4.0.4.1 Calibration of spray flow-rate

A pressure transducer is used to measure the flow-rate of the pump. It is connected to a Venturi tube and gives an electrical signal proportional to the instantaneous flow rate. The signal is red via a multimeter.

In order to calibrate the transducer, the amount of water exiting through the spray during a 90s run is collected in a bucket and then measured. Using the mean voltage signal it is possible to find the relation between voltage and flow-rate. This mean voltage is the mean obtained by an acquisition system connected to the pressure transducer. Eleven runs have been done for different by-pass valve aperture. The values obtained can by observed in Fig. 4.5 .

The main problem found during the calibration is the fluctuation of the electrical signal. So the variation (ΔV) between the zero voltage and the mean

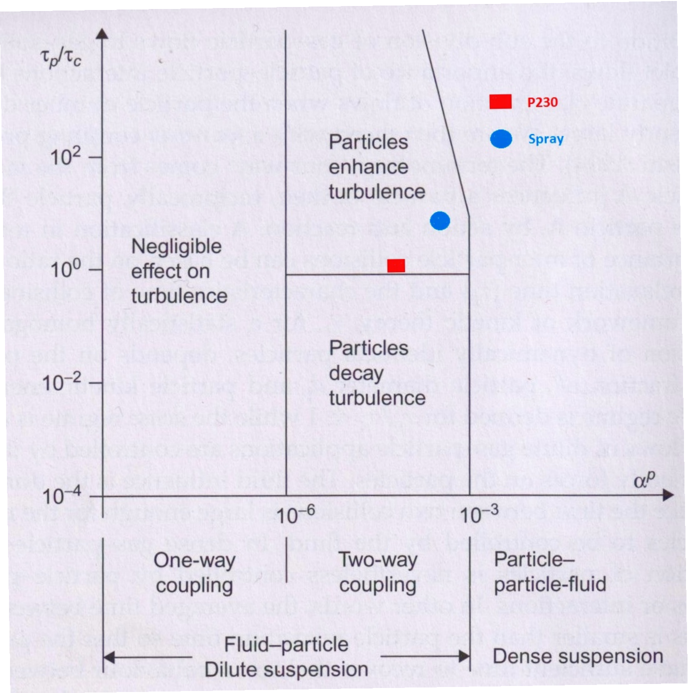


Figure 4.3: Particle interaction f(St,α).

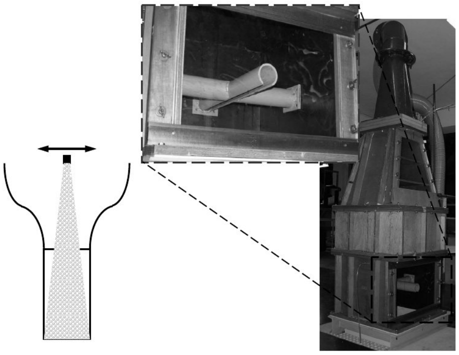


Figure 4.4: spray

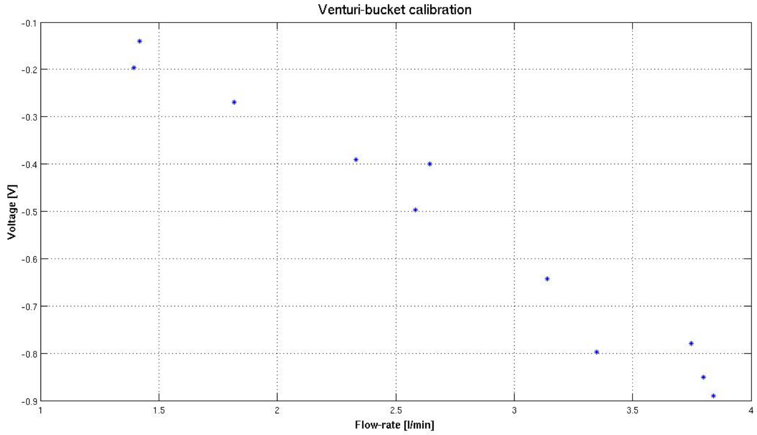


Figure 4.5: Calibration points

voltage has been used. Considering the second order interpolation, the relation between electrical signal and flow rate (F) is obtained in eq. 4.2

$$\triangle V = 1.4522F^{\mathbf{2}} - 4.8408F + 0.66673\ [V] \tag{4.2}$$

In order to assure the correct values, the flows values are checked regularly through a bucket calibration between tests.

4.0.4.2 Flow velocity measurement

In order to determine the flow velocity entering the test section (U_0), a pressure transducer is used. It determines the difference of pressure (Δp) between the stagnation chamber (S) and the test section (Fig 4.6). A water column manometer has been used to obtain the relation between the output of the transducer, an electrical signal, and the pressure difference. Finally, the relation between the pressure and the velocity is obtained through the Bernoulli (eq. 4.3) and the continuity equation (eq. 4.4).

$$\frac{U_o^2}{2} = \frac{\triangle p}{\rho_{air}} + \frac{U_s^2}{2} \tag{4.3}$$

$$U_0 S = U_S S_s \tag{4.4}$$

The air density has been considered constant during the experiments and equal to 1.12 Kg/m^3

.

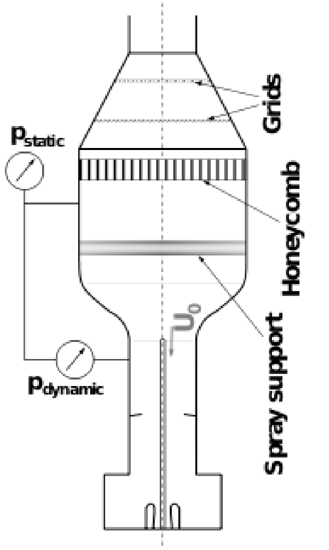


Figure 4.6: Pressure transducer in the wind tunnel

4.0.5 Inhibitor improvement

In Tóth dissertation [20], the thermal insulation inhibitors were modelled as a rectangular steel plate (Fig. 4.7). In order to avoid a massive dripping from water to the cavity, they were attached to a vaccum system (Fig. 4.8) that removes a part of the water accumulated on them.

It was determined that the experimental dripping was higher than the one expected considering the numerical results. So, another inhibitor configuration should be found in order to reduce the importance of dripping.

In order to improve the suction phenomena, the perfored tube has been changed to a trapping system (Fig. 4.9), formed by a superior plate and several holes. Two different width cover plates will be tested (19 and 24 mm). The distance between the plate and the base of the inhibitor will be considered also as a parameter (no gap, 1 and $3mm$).

Different materials for the inhibitor surface have been tested (Fig. 4.10). The reason of changing the surface material, is to obtain an extra porosity in order to retain the maximum amount of water. The surface to cover is the gap between the trapping system and the inhibitor tip and sidewalls. The materials used have been a porous black foam and a *Scotch Brite* foam.

Finally, a grid with a characteristic dimension of 1 mm has been considered. This grid is though to retain the largest drops of water arriving to the inhibitor surface.

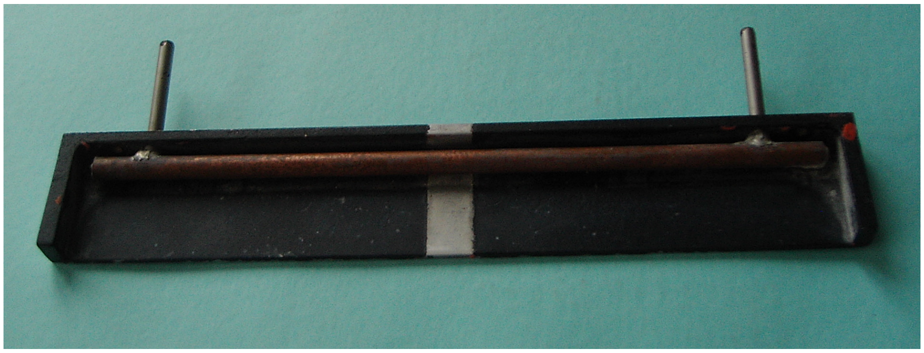


Figure 4.7: FIRST INHIBITOR DESIGN

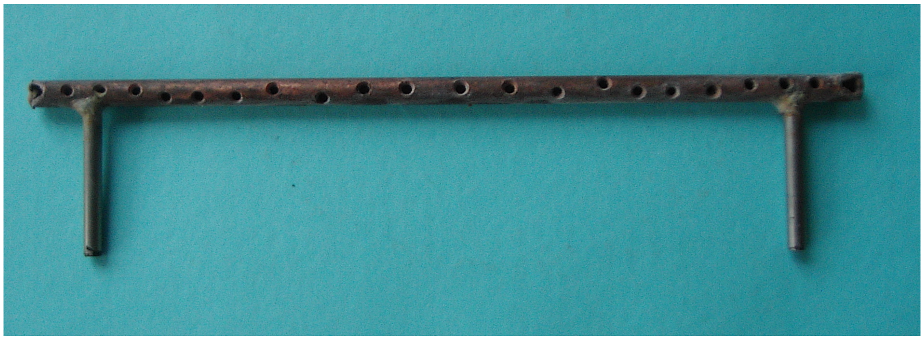


Figure 4.8: ASPIRATION MECHANISM

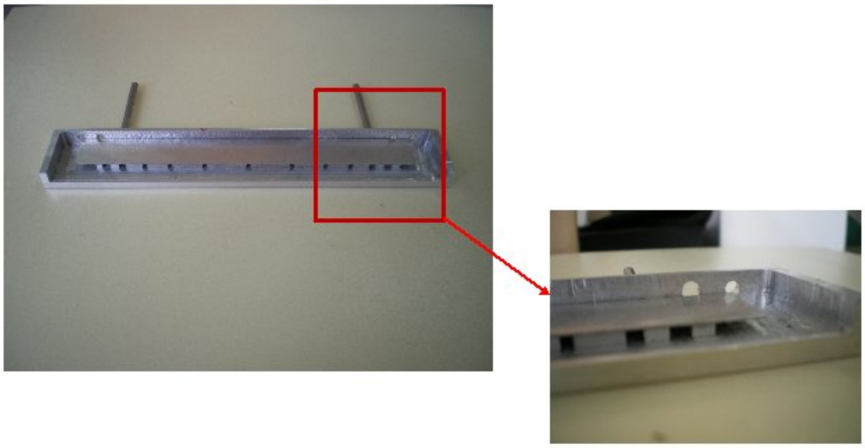


Figure 4.9: NEW TRAPPING SYSTEM

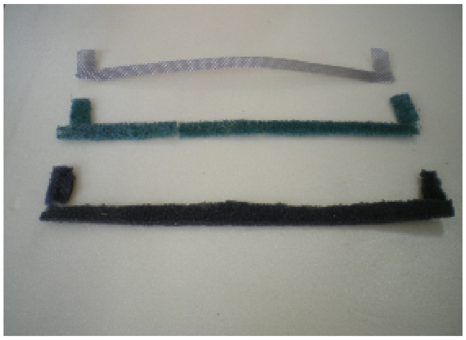


Figure 4.10: GRID, SCOTCH BRITTE AND FOAM

Setup	Parameters
0	Previous Tóth design
1	New Design
2	Plate + 1 *mm* gap
3	Plate + Scotch Britte
4	Plate + Scotch Britte + grid
5	Plate + grid
6	Plate + black foam
7	Plate + 3 *mm* gap
8	wide plate
9	wide plate + 3 *mm* gap

Table 4.3: SETUPS FOR INHIBITOR IMPROVEMENT

The different inhibitors setup used and their characteristic parameters could be found in table 4.3 .

The setup named as new design refers to the inhibitor with the 19 *mm* width plate.

In order to determine which is the optimal configuration, the quantity to be minimized has been the ratio dripping / entrapment. Dripping refers to the amount of water coming from the inhibitor while entrapment is the amount of water found in the cavity minus the Dripping. The sum of Dripping and Entrapment is named as accumulation and is the amount of water that is found in the cavity at the end of each test run.

The first point is to determine the experimental way of calculating the dripping and entrapment (Fig 4.11). A High Speed Camera (*HSC*) will be used to record the test. It is fixed perpendicular to the right side wall at a distance of approx. 100 *cm*. It is focusing the lower part of the inhibitors. In order to obtain a clear image, the inhibitor is illuminated with a lamp. The lamp illuminating field is perpendicular to the camera vision. The characteristics of

Element	Specs
Camera	*Phantonv*7.1 High Speed *CMOS* 512*x*124 resolution
Lens	*AF Nikkoer* $f = 50mm$ at $f\# = 1.8$
Lamp	1 *kW* halogenous photographic lamp

Table 4.4: Optic elements for dripping visualization

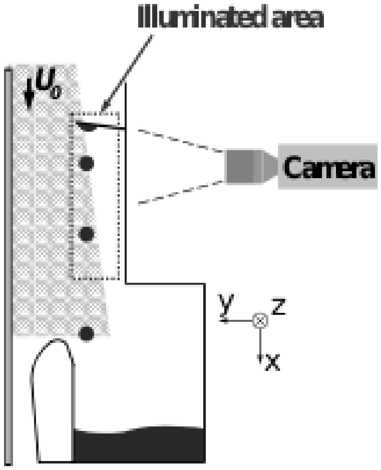


Figure 4.11: dripping visualization arrangement

the *HSC* camera, the lens, and the lamp can be viewed in table 4.4.

Each test run lasts for 12 *min* and 2 *min* are recorded. The amount of drops (dripping) and their sizes are obtained from the video. The size is obtained as the average of the sample of 5 drops. Finally, the accumulation is obtained from measuring the water that is trapped in the cavity. These values are corrected by a factor determining the asymmetry of the spray. The results of dripping / entrapment for the different setups can be seen in Fig 4.12 .

The results show that the setup that optimizes the dripping is the eighth one with a value of 20 percent. So the inhibitor with no additional material with the 24 *mm* plate and no gap will be considered as the nominal one for the parametric test.

The results for the setups with foam and grids that were expected to be the most efficient ones show the highest values. One of the possible reasons is that there may be a blockage in the trapping system. The combination of *Scotch Brite* and grid appear to be the less effective setup.

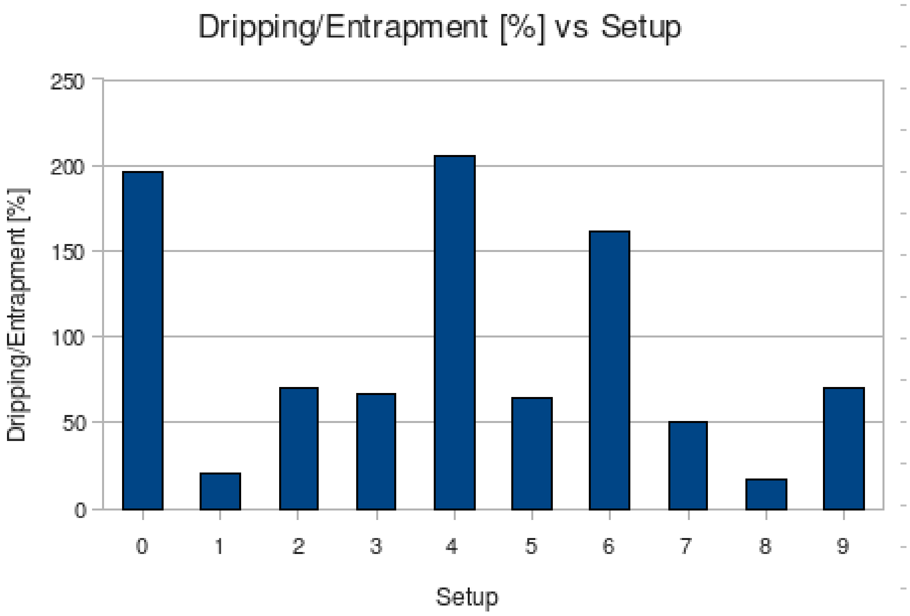


Figure 4.12: DRIPPING / ENTRAPMENT VS SETUP

Chapter 5

Parametric test

In this chapter, the phenomena of the slag accumulation will be discussed in function of different parameters, both geometrical and flow characteristics ones. A physical explanation for the different behaviors will be given. In Fig. 5.1 , the geometrical parameters are presented .

The inhibitor size, h, is the perpendicular distance from the wall to the tip of the inhibitor. It is worth to remark that the inhibitor has an inclination of $10\ degrees$.

The inhibitor position, L_i, is the distance, on x axis, from the inhibitor tip to the nozzle tip.

The Obstacle to Nozzle ratio, $02NR$, is a combination of the anterior two parameters. It is defined as the inhibitor position normalized by the inhibitor size.

The cavity width, w, is the distance, on y axis, between the interior walls of the cavity.

The nozzle throat opening, o, is the distance, on y axis, between the most interior point of the nozzle and the splitting plate surface.

Finally there is the Obstacle Tip to Nozzle Tip distance, $OT2NT$, defined as the distance, on y axis, between the tip and the nozzle and the inhibitor tip. It is defined positive when the nozzle is over (nearer to the splitting plate) the inhibitor tip.

The unique parameter regarding the flow characteristics is the air velocity, U_0 .

It will be helpful to remember the definition of some quantities. First of all, the dripping , Dri, is the amount of water falling to the cavity from the inhibitors. Secondly, the accumulation, Acc, is the amount of water found in the cavity at the end of one test. Finally, the entrapment, Ent, is defined as the amount of water found in the cavity minus the dripping. It can also be seen as

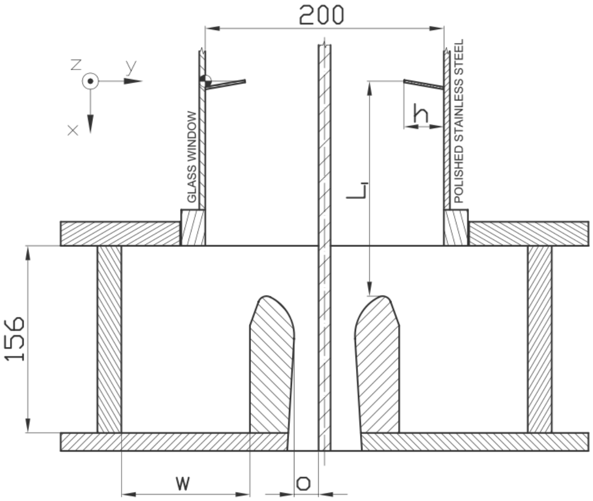


Figure 5.1: GEOMETRICAL PARAMETERS

the amount of liquid carried by the flow. The test facility is the same than the one presented in the inhibitor choice section.

Eighteen different configurations have been tested and 19 measurements realized. The corrected rate of accumulation and entrapment ratio can be found in table 5.2 .

The quantity that will be mainly discussed is the entrapment due to the fact that has been possible to try to explain physically its behavior. Instead, it has not been possible to find a valid explanation for the dripping phenomena.

5.1 Experimental results

5.1.1 Flow velocity

In Fig. 5.3, the entrapment is plotted in function of the free stream velocity, U_0. It is possible to observe that for a fixed geometrical configuration, $h = 33.5$ mm, but changing the velocity, the entrapment remains almost constant.

The main reason for this constant behavior may be the fact that the recirculation zone after the inhibitor does not change enough while changing the velocity. This fact was observed by Lema.M [28]. His simulation of the two phase flow of a geometrical analog case, cavity airflow, concluded that for different velocities (3 m/s and 7 m/s) the main recirculation zone moves and changes minimally. So when the unique changing factor is the velocity, the drops coming from the spray are affected by a similar flow field.

If the plot of Accumulation in front of the flow velocity is considered (Fig.

$Meas.$	U_o	h	L_i	$O2NR$	w	o	$OT2NT$	Acc	Ent
#	$[m/s]$	$[mm]$	$[mm]$	[]	$[mm]$	$[mm]$	$[mm]$	[%]	[%]
1	10	33.5	310	9.25	107.2	15.4	25.69	5.46	4.62
2	10	33.5	310	9.25	107.2	15.4	25.69	5.12	3.90
3	6	33.5	310	9.25	107.2	15.4	25.69	5.02	3.91
4	8	33.5	310	9.25	107.2	15.4	25.69	4.29	3.24
5	14	33.5	310	9.25	107.2	15.4	25.69	4.71	3.29
6	10	33.5	110	2.09	107.2	15.4	25.69	5.19	4.76
7	10	33.5	150	4.48	107.2	15.4	25.69	4.54	3.90
8	10	33.5	230	6.87	107.2	15.4	25.69	4.26	3.31
9	10	23.5	150	6.38	107.2	15.4	35.69	7.46	6.17
10	10	23.5	310	13.19	107.2	15.4	35.69	7.31	7.08
11	10	$-$	$-$	$-$	107.2	15.4	-	14.54	14.54
12	10	50	310	6.2	107.2	15.4	9.19	5.47	3.63
13	10	33.5	310	9.25	107.2	25.4	15.69	2.8	2.3
14	10	33.5	310	9.25	107.2	45.4	-4.31	1.72	1.36
15	10	33.5	310	9.25	107.2	67	-25.91	1.65	1.09
16	10	33.5	310	9.25	150	67	25.69	4.81	4.06
17	10	33.5	310	9.25	73	67	25.69	4.58	4.04
18	10	33.5	310	9.25	48	67	25.69	3.96	3.52
19	10	23.5/33.5	310/110	9.25	107.2	67	25.69	4.58	3.46

Figure 5.2: TEST MATRIX AND EXPERIMENTAL RESULTS

5.4), it is observed the same tendency than in Fig.5.3 It is also possible to conclude that the main part of the accumulation is due to the entrapment.

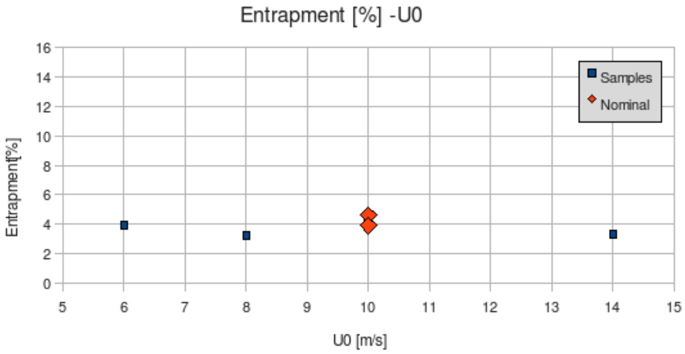


Figure 5.3: Entrapment vs U_o

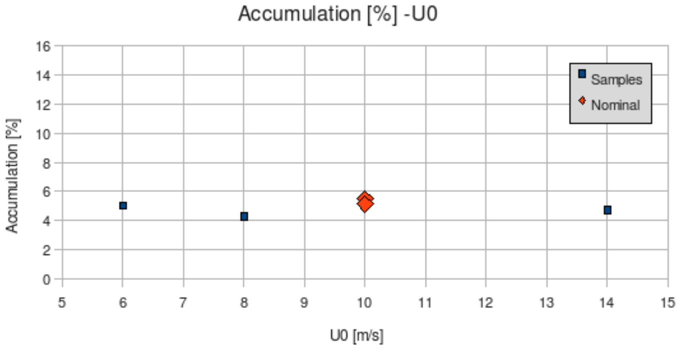


Figure 5.4: Accumulation vs U_o

This fact presents a discrepancy with previous studies. Tóth [20] determined that the accumulation in function of velocity presents a decreasing behavior with a factor of 0.294. Instead, in the current study a constant behavior has been found. The main reason of this difference may be due to the improved inhibitor suction system. As showed previously, the dripping, which was considered the main factor of the accumulation, is drastically reduced.

5.1.2 Inhibitor size

The entrapment in function of the inhibitor size, h, is showed in Fig 5.5. Two different zones can be observed. The first one, before $h = 33.5\ mm$ shows a decreasing behavior. Instead, the zone for h greater than 33.5 shows a constant behavior.

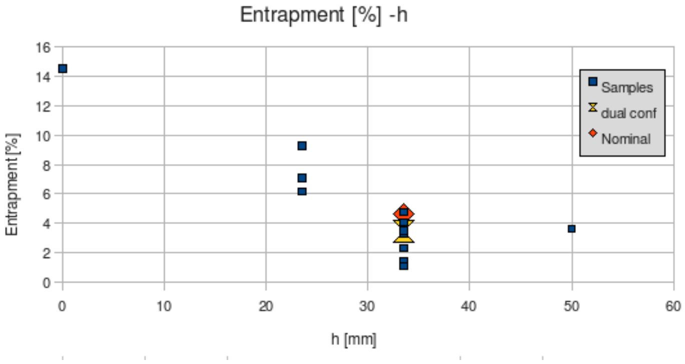


Figure 5.5: ENTRAPMENT VS h

In the first zone, the smaller inhibitor is used. These obstacles create a recirculation zone downstream them and also a blockage of the flow passing section. The recirculation zone may not be as big and strong enough to capture a large amount of the smaller drops but the blockage added may reduce the amount of drops falling in the cavity. Once the inhibitor is big enough (between 23.5 and 33.5 mm) the recirculation zone may be energetic enough to capture the smaller drops. So the decrease due to the blockage could be compensated by this increase in the recirculation zone. Probably, the vortex shedding may produce also more energetic vortexes.

The inhibitor could be seen also as a filter for the bigger drops, the ones that are not disturbed by the recirculation bubble. If this point of view is accepted, the bigger drops may be the main responsible for the entrapment.

There is a point that needs a further explanation. The dual configuration has the nominal inhibitor, $h = 33.5$ at a position of 310 mm and the smaller, $h = 23.5\ mm$, at the lower position $L_i = 110\ mm$. This points has approximately the same value for the entrapment than the nominal, where no additional inhibitor is added.

Adding an additional inhibitor in lower positions may not change the entrapment rate. One possible explanation is that the lower inhibitor may be at the end of the recirculation zone downstream the first inhibitor. In this zone Lema.M *LES* and laminar simulations [28]showed that the streamlines tend to go towards the nozzle and the smaller drops avoid to enter the cavity. The presence of this second inhibitor may change the streamlines curvature but they may finish in the nozzle anyway.

5.1.3 Inhibitor position

In Fig 5.6, the rate of entrapment in function of the inhibitor position, L_i , can be seen. It is not possible to realize a joint analysis, so the results will be commented fixing some parameters.

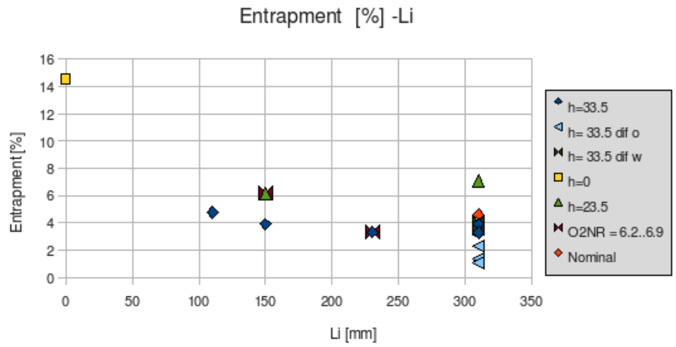


Figure 5.6: Entrapment vs L_i

Considering constant Inhibitor size (23.5 mm and 33.5 mm) and fixing the cavity parameters it can be seen that the entrapment remains almost constant for the different positions. Lema.M realized LES simulations for the booster [28] considering different inhibitor positions, $5h$, $9h$ and $10.8h$. It was found that the recirculation bubble and the streamlines for the $9h$ and $10.8h$ are similar and finishing in the nozzle. For the $5h$ position the flow structure inside the cavity is affected by the recirculation bubble.

As long as that the inhibitor size is constant, the blockage of the flow passing section is the same. The bigger drops unaffected by the streamlines find the same obstacle or filter. Instead, the smaller drops are affected by a similar flow structure. This may be one possible explanation for this constant behavior.

5.1.4 Obstacle to nozzle ratio

In Fig.5.7 , the entrapment is showed in function of the normalized inhibitor position or Obstacle to nozzle ratio, $O2NR$, for a fixed L_i . This ratio is calculated as Li/h. Previously, it has been shown than the entrapment seems to be independent to the inhibitor position. Considering L_i constant this parameter serve to show that the important parameter is h and not L_i .

5.1.5 Cavity width

In Fig 5.8, the entrapment is plotted in function of the cavity width. It has been found that for a same L_i and h the entrapment remains constant. For $w = 107$ mm so many different configurations have been tested so nothing can be concluded at this cavity width.

Changing the cavity width, the recirculation bubble inside the cavity get enlarged but it may not affect the recirculation zone outside the cavity.

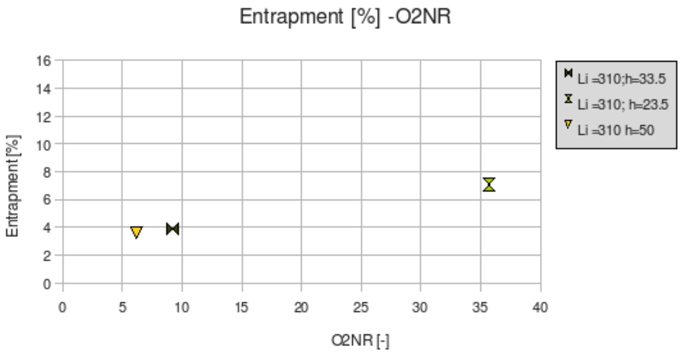


Figure 5.7: ENTRAPMENT VS $O2NR$

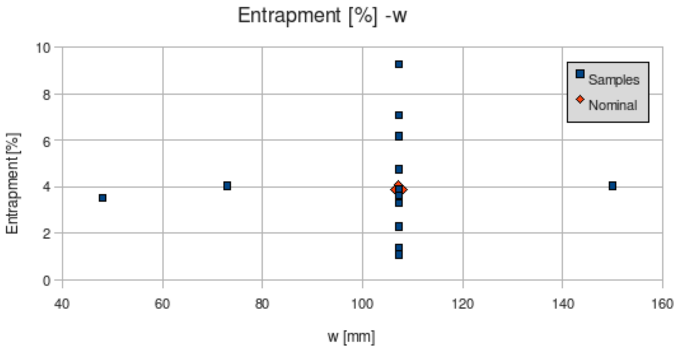


Figure 5.8: ENTRAPMENT VS w

5.1.6 Nozzle throat opening

The behavior of the entrapment in function of the nozzle throat opening, o, can be found in Fig.5.9 . A quasi-hyperbolic decreasing behavior has been found.

Changing the opening, the velocity in the nozzle changes. As a consequence the turbulence intensity and the streamlines also will suffer of a strong change. In order to find a physical explanation to his behavior, further experiences should be carried on.

5.1.7 Obstacle tip to nozzle tip distance

The Obstacle to tip distance could be also seen as the distance by which a droplet passing near the tip of the inhibitor is deviated in order not to be entrapped in the cavity. It is defined positive when the nozzle is in nearer to the splitting plate than the tip of the inhibitor. In Fig 5.10, the plot of the entrapment against the $OT2NT$ can be found.

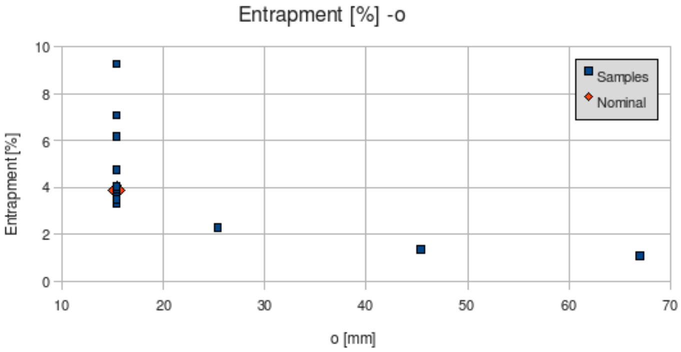


Figure 5.9: Entrapment vs o

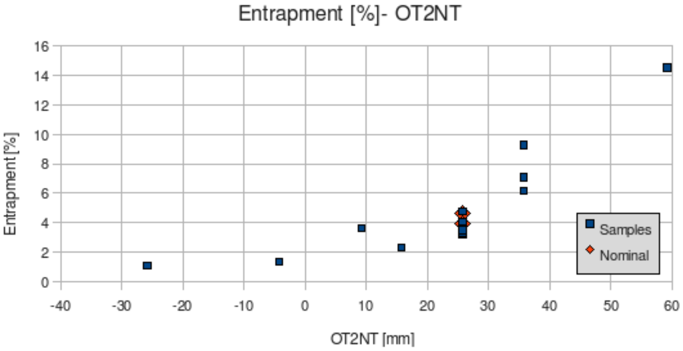


Figure 5.10: Entrapment vs $OT2NT$

Two main zones can be observed. In the first one, for negative $OT2TN$ a costant behavior has been observed while in the second there is a linear increasing behavior.

When the tip of the inhibitor in nearer the splitting plate, the biggest drops may go towards the nozzle throat. Only the smaller drops may be attracted towards the cavity. Instead, when the $OT2NT$ raises with positive values, not only the velocity towards the nozzle may increase but also the passing section towards the cavity. The bigger drops could also fall directly in the cavity. This phenomena seems to follow the inertia driven entrapment explanation, where the most part of the entrapment may be due to the bigger drops.

The $OT2NT$ turns to be the most relevant parameter, where its variations produce larger variations in the entrapment. It is worth to remind than it is a combination of o and h.

Chapter 6

Experiments conclusions

The first objective has been to improve the suction system designed by Tóth in [20]. The main goal has been to reduce the amount of dripping. A new inhibitor has been designed where a trapping system has been implemented. In order to increase the amount of water captured, different materials have been used to cover the inhibitor surface (a grid, a black porous foam and *Scotch Brite* foam). Also the distance between the base of the inhibitor and the plate covering the trapping system has been modified (no gap, 1 mm or 3 mm). There has been two covering plates width considered (19 and 24 mm).

The configuration that has shown a smaller ratio of Dripping / Entrapment has been the new design of inhibitor with the 24 mm plate, no gap and no additional material on the surface. This configuration has been considered as the nominal for the parametric test.

The second objective has been to carry out a parametric test in order to determine how influenced is the flow by geometrical (inhibitor size, inhibitor position, Obstacle to Nozzle ratio, cavity width, nozzle opening and Obstacle Tip to Nozzle tip) and flow parameters (Air velocity).

It has been found that the most relevant parameter is the Obstacle Tip to Nozzle Tip which mainly determines the distance by which a droplet passing near the tip of the inhibitor is deviated in order not to be entrapped in the cavity. It is defined positive for the nozzle located over the inhibitor tip.

The smaller drops are the ones affected by the flow and the recirculation bubble after the inhibitor. Instead, the bigger drops are not affected by the flow. For positive $OT2NT$, the cavity nozzle is in front of the inhibitor tip. Some of the bigger drops fall directly to the cavity. This phenomenon is not happening when the nozzle tip is situated rear the inhibitor tip (negative $OT2NT$). So the entrapment may be a inertia driven phenomenon where the most part of it is due to the bigger drops.

In order to complete the existent experiences, an axisymmetric model should be considered. As long as the dripping has been reduced, the next step will be to consider flexible inhibitors as in the real case. In this way, the coupling between the structure and the shedding could be studied. Also a futher investigation of the coupling between vortex shedding and entrapment should be done.

Part III

Numerical simulations

Nomenclature

α_φ Volume fraction for phase φ

β Volume fraction for phase b ($sixPhaseEulerFoam$)

ρ_φ Density of phase φ

ϕ Volumetric mixture flux

$\phi_{r,a_{ij}}$ Relative flux of phase i with phase j

γ Blendind differencing coefficient

k Turbulent kinetic Energy

ε Turbulent dissipation

φ Phase

χ_φ Conditional indicator function

ν^t Turbulent viscosity

ν_φ^{eff} Effective turbulent viscosity

a Dispersed phase

a_i Dispersed phase i

b Continuum phase

C_d Drag coefficient

C_l Lift coefficient

C_o Courant number

C_{vm} Virtual mass coefficient

C_t Response coefficient

lhs left hand side

L_e Eddy dimensional scale

K_φ Drag coefficient reformulation for phase φ

$\overline{M}_\varphi$ Inter phase momentum transfer for φ

NS Navier Stokes

P_φ Production of kinetic energy

rhs Right hand side

$\overline{R}_{\varphi}^{eff}$ Effective Reynolds stresses

RS Reynolds Stresses

Re Reynolds adimensional number

Re_t Reynolds adimensional number based in eddy scale

S_k Dispersed phase source term for kinetic turbulent

S_ϵ Dispersed phase source term for turbulent dissipation

$\bar{U}_r$ Relative velocity between phases

$\bar{U}_\varphi$ Velocity of phase φ

$\bar{U}_\varphi^{'}$ Turbulent fluctuation velocity

Chapter 7

Introduction. Simulation of multiphase flows

When simulating numerically multiphase dispersed flows, different methodologies can be used depending on the formulation of the conservation equations of the dispersed phase: Lagrangian or Eulerian.

7.1 Euler-Lagrangian

In Euler-Lagrange, each dispersed particle is tracked during the simulation domain. Doing so, each particle has its own equation describing its motion. The main advantages of this approach are that the particles could be defined geometrically alone and that phenomena such as break up or agglomeration could be easily modeled. There are two main drawbacks. The first is the numerical complexity when the dispersed phase fraction grows. The second one is related to the coupling between phases due to the fact that the continuous one is treated in the Eulerian frame, so values and fields are referred to a fix coordinate frame. The simulations carried out with this approach are transient

As presented in the Introduction (Part *I*), the Euler-Lagrangian approach have been largely used when simulating the slag entrapment in solid rocket motors (*SRM*).

7.2 Euler-Euler. Two fluid approach

In the second way, known as Euler-Euler or two fluid methodology, both phases are treated as continuous. The conservation equations are considered macroscopically using the average operation. The effect of particles are averaged within computational cells. Due to the averaging procedure extra terms appear. They are known as Reynolds stresses and inter-phase momentum exchange. The main advantage is the reduced numerical complexity compared to the Euler Lagrange models. The main difficulties are represented by the modelling of the inter-phase momentum and the introduction of particle distributions that leads to a statistical approach. The simulations done with this approach could be either transient or stationary.

When simulating the slag entrapment in *SRM* the two fluid approach have not been largely used. Lagrangian tracking have been mostly preferred.

7.3 Differences

When comparing both approaches, the main difference relies on the numerical complexity. Referring to Euler-Euler:

1. the computational domain, mesh, can be the same as for a single phase simulation;
2. the modeling of the inter-phase interaction becomes easier;
3. it is possible to run steady state simulations;
4. less complex boundary conditions for the dispersed phase.
5. it is numerically more stable.

The objective of the numerical part of this work will be to obtain a simple model, affordable with standard performance *PCs*, that will complement the studies already carried out with Eulerian-Lagrange and Large Eddy Simulations, using *OpenFoam* as starting points.

This model will be compared with the result from the experimental tests in order to study the slag entrapment phenomena.

Chapter 8

Two fluid approach: *twoPhaseEulerFoam*

In this chapter the theoretical basis of the two fluid approach and its application to *OpenFoam* software will be presented. The algorithm that will be used and further on modified is named as *twoPhaseEulerFoam* and it grounds on the following hypotheses :

1. Biphasic flows.
2. Eulerian-Eulerian approach.
3. The dispersed phase (a) cannot become the continuum phase (b). The roles between phases remain unchanged
4. There is no agglomeration or breakup phenomena.
5. The geometrical characteristics of the dispersed particles are constant

The theoretical algorithm used to implement this solver derives from the *PhD* theses of Rusche [29] and Hill [30].

8.1 Governing equations

In order to solve the Navier Stokes equations (NS) and to reduce its numerical complexity, a mean flow is considered. Doing so, the possibly irrelevant microscopic information is not considered and only macroscopic data is studied. This is an engineer approach and its widely used in the industry.

$$U = \overline{U} + U' \tag{8.1}$$

The mean NS equations are also named Reynolds equations. The flow is characterized by a mean velocity and a turbulent fluctuation (eq.8.1), so a macroscopically point of view is considered. When averaging the NS equations, an extra term appear due to the statical covariance of turbulence fluctuation velocities ($< U_i' U_j' >$). This term is named as Reynolds Stress (RS). The unknowns of the Averaged NS equations are the field of velocities and pressure [31]. At this point the problem is unclosed, RS must be modeled through a turbulence model.

As stated previously, the two fluid approach will be used so the momentum averaged and the continuity equation will be considered for each phase. An extra term taking account of the transfer of momentum between phases will appear in the momentum equation.

The equations are also averaged conditionally depending of the presence of the phase (φ), so first the equations should be multiplied by an indicator function (χ_φ) that is 0 when the considered phase is not present and is 1 otherwise [30]. Then the equations are averaged. In order to avoid rewriting, conditionally averaged quantities ($\overline{\phi}$) will be considered :

$$\overline{\phi} = \frac{\overline{\chi_\varphi \phi}}{\alpha_\varphi} \tag{8.2}$$

- where α_φ is the volume fraction of the phase φ. Only two phases will be considered. a will be the dispersed phase while b will be the continuous one.

In the following section, the conditionally averaged two fluid equation will be presented. Different models for the extra terms, Reynolds stresses and inter-phase momentum exchange will be given. Finally, the equations will be discretised and the $OpenFoam$ solver will be presented.

8.1.1 Momentum conditionally averaged equation

The momentum conditionally averaged equation for phase φ is:

$$\frac{\partial \alpha_\varphi \overline{U}_\varphi}{\partial t} + \nabla \cdot \left(\alpha_\varphi \overline{U}_\varphi \overline{U}_\varphi \right) + \nabla \cdot \left(\alpha_\varphi \overline{R}_\varphi^{eff} \right) = -\frac{\alpha_\varphi}{\rho_\varphi} \nabla \cdot (\overline{p}) + \alpha_\varphi g + \frac{\overline{M}_\varphi}{\rho_\varphi} \tag{8.3}$$

Where $\overline{R}_\varphi^{eff}$ is the effective Reynolds and viscous stresses and $\overline{M}_\varphi$ is the inter-phase momentum transfer term. As stated before, these terms need to be modeled in order to give a closure to the model.

Inter-phase momentum

Interphase momentum exhange term takes account of the momentum exchanged between the different phases. Considering a biphasic system, there is only one

Contribution	Expression	Generic expression
Drag	$+\alpha_a C_d \frac{3}{4} \frac{\rho_b}{d_a} \overline{U}_r \overline{U}_r$	$+A_d \overline{U}_r$
Lift	$+\alpha_a C_l \rho_b \overline{U}_r \times \left(\nabla \times \overline{U}_b\right)$	$+A_l \overline{U}_r \times \left(\nabla \times \overline{U}_b\right)$
Virtual Mass	$+\alpha_a C_{vm} \rho_b \left(\frac{D_b \overline{U_b}}{Dt} - \frac{D_b \overline{U_b}}{Dt}\right)$	$+A_{vm} \left(\frac{D_b \overline{U_b}}{Dt} - \frac{D_b \overline{U_b}}{Dt}\right)$
Turbulent Drag	$-C_d \frac{3}{4} \frac{\rho_b}{d} \frac{\nu_t}{\sigma_\alpha} \overline{U}_r \nabla \alpha_a$	$A_\alpha \nabla \alpha$

Table 8.1: Inter-phase momentum term contributions.

term of momentum transfer due to conservation. It will be calculated for the dispersed phase (a). The momentum exchanged by the continuum phase is the complementary of the contribute for the dispersed phase.

The inter-phase momentum term is formed by different contributions. The dispersed particle is subjected to the following forces:

1. Drag. When averaging the equations and additional term of drag appear. This is named as turbulent drag and is the drag produced by the turbulent fluctuation. Only the drag due to a force of resistance will be considered.
2. Lift
3. Virtual mass

An analytical expression for these contributes can be seen in table 8.1.

Where $\overline{U}_r$ is the averaged relative velocity between the continuum (a) and the dispersed phase (b):

$$\overline{U}_r = \overline{U}_b - \overline{U}_a \tag{8.4}$$

There are different empirical expressions for the calculation of the drag coefficient. The one used is Schiller and Naumann solid particle drag model (eq. 8.5) [38] .

$$C_d = \frac{24}{Re}\left(1 + 0.15 Re^{0.687}\right) \tag{8.5}$$

Where the number of Reynolds is defined as:

$$Re = \frac{\rho_b U_r d}{\mu_b} \tag{8.6}$$

C_l and C_{vm} are defined empirically and are functions of the nature of the dispersed phase. Their values will be further discussed.

In order to simplify the notation, the contributions are reformulated using A_d, A_l and A_{vm} (table 8.1).

Reynolds Stress

Reynolds stress appeared due to the operations of averaging the equations. In order to give a closure to the problem, they need to be modeled. They could be modeled through the simpler turbulence viscosity hypothesis or the more complex Reynolds stresses transport equations. In previous studies on multiphase flows summarized in [29] turbulence viscosity hypothesis was able to give satisfactory results for a smaller numerical complexity.

The turbulence viscosity hypothesis , or Boussinesq hypothesis, is based on the idea that the deviatronic Reynolds combined stresses are proportional to the the mean rate of strain. This hypothesis leads to a a close expression for the RS tensor:

$$\overline{R}_{\varphi}^{eff} = -\nu_{\varphi}^{eff}\left(\nabla\overline{U}_{\varphi} + \nabla\overline{U}_{\varphi}^{T} - \frac{2}{3}I\nabla\cdot\overline{U}_{\varphi}\right) + \frac{2}{3}Ik_{\varphi} \tag{8.7}$$

where k_{φ} is turbulent kinetic energy of the phase φ and ν_{φ}^{eff} is the effective viscosity defined as $\nu_{\varphi}^{eff} = \nu_{\varphi} + \nu^{t}$ and ν^{t} is the turbulent viscosity. An expression for ν^{t} and a transport equation for k will be given when presenting the turbulence model. Combined Reynolds stress contains both turbulent and viscous stresses.

Phase Intensive Momentum expression

In order to avoid numerical problems when the dispersed phase is Small ($\alpha_a \to 0$), eq 8.3 can be reformulated as:

$$\frac{\partial\overline{U}_{\varphi}}{\partial t} + \nabla\cdot\left(\overline{U}_{\varphi}\overline{U}_{\varphi}\right) + \frac{\nabla\cdot(\alpha_{\varphi})}{\alpha_{\varphi}}\cdot\overline{R}_{\varphi}^{eff} = -\frac{1}{\rho_{\varphi}}\nabla\cdot(\overline{p}) + g + \frac{\overline{M}_{\varphi}}{\alpha_{\varphi}\rho_{\varphi}} \tag{8.8}$$

There is one term in eq. 8.8, $\frac{\nabla\cdot(\alpha_{\varphi})}{\alpha_{\varphi}}\cdot\overline{R}_{\varphi}^{eff}$ that still is divided by α_{φ}. This could cause stability problems. In [34] Weller proved that he ratio $\frac{\nabla\cdot(\alpha_{\varphi})}{\alpha_{\varphi}}$ tend to zero when $\alpha_a \to 0$.

8.1.2 Continuity conditionally averaged equation

The continuity conditionally averaged equation for each phase φ is:

$$\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot \left(\overline{U}_\varphi\right) = 0 \tag{8.9}$$

When considerding the previously presented biphasic system, the volumetric equation of the mixture is defined as:

$$\frac{\partial \alpha_a}{\partial t} + \nabla \cdot \left(\overline{U}_a \alpha_a\right) + \frac{\partial \alpha_b}{\partial t} + \nabla \cdot \left(\overline{U}_b \alpha_b\right) = 0 \tag{8.10}$$

8.1.3 Turbulence model

Once Boussinesq hypothesis is assumed, Reynolds stress can not still be computed. There is still the need to define the value of the kinetic turbulent energy for both phases. In previous studies of biphasic flows using Eulereian formulation, the $k - \varepsilon$ models has been chosen [29] mainly due to its results for a moderate computational cost.

$k - \varepsilon$ model is a two equation model where two turbulent quantities are considered: the turbulent kinetic energy (k) and the turbulent dissipation rate (ε). The mathematical formulation of the model includes a transport equation for k and ϵ and a definition for the turbulent viscosity.

In order to apply this method to a biphasic system the dispersed phase RS have to be included in the transport equation for the turbulent quantities. The standard transport equation for k and ε for the continuum are modified with an extra term that takes account of the effects of the influence of the dispersed phase (S_k and S_ϵ). These terms raise from the non isotropic disturbances at the interface between continuum and dispersed phases.

$$\frac{\partial k_b}{\partial t} + \left(\overline{U}_b \cdot \nabla\right) k_b - \nabla \cdot \left(\frac{\nu_b^{eff}}{\sigma_k} \nabla k_b\right) = P_b - \epsilon_b - S_k \tag{8.11}$$

$$\frac{\partial \varepsilon_b}{\partial t} + \left(\overline{U}_b \cdot \nabla\right) \varepsilon_b - \nabla \cdot \left(\frac{\nu_b^{eff}}{\sigma_\epsilon} \nabla \varepsilon_b\right) = \frac{\varepsilon_b}{k_b}\left(C_1 P_b - C_2 \varepsilon_b\right) - S_\varepsilon \tag{8.12}$$

where P_b is the production of turbulent kinetic energy defined as $P_b = 2\nu_b^{eff}\left(\nabla \overline{U}_b\right) \cdot \left(\nabla \overline{U}_b + \left(\nabla \overline{U}_b\right)^T\right)$, ν_b^{eff} is the effective viscosity of the continuum and ν^t is the turbulent viscosity defined as $\nu^t = C_\mu \frac{k_b^2}{\varepsilon_b}$. The extra terms are shown in eq. 8.13 and eq. 8.14 .

$$S_k = \frac{2k_b \alpha_a A_d (C_t C_t - 1)}{\rho_b} + \frac{A_d \nu^t}{\rho_b \sigma_\alpha} \frac{\nabla \alpha_a}{\alpha_b} \cdot \overline{U}_r \tag{8.13}$$

$$S_\varepsilon = \frac{2C_3 \varepsilon_b \alpha_a A_d (C_t C_t - 1)}{\rho_b} \tag{8.14}$$

There is the need to give a definition for the response coefficient (C_t). This coefficient was first defined by Politis [35] in order to relate the continuum phase Reynolds stress with the dispersed phase when averaging by time and volume. Then it was generalized by Hill in order to be applied to the joint averaging applied also to Rusche's model. Issa gave an empirical formulation for this term (eq. 8.15)

$$C_t = \frac{U_a^{'}}{U_b^{'}} \rightarrow \frac{3 + \beta}{1 + \beta + 2\frac{\rho_d}{\rho_b}} \tag{8.15}$$

where $\beta = \frac{2A_d L_e^2}{\rho_b \nu_b Re_t}$, $Re_t = \frac{U_b^{'} L_e}{\nu_b}$ is the number of Reynolds based to the eddy scale (L_e) defined as $L_e = C_\mu \frac{k^{1.5}}{\varepsilon_b}$,ν_b is the viscosity of the continuum, $U_b^{'}$ the fluctuation of the continuum defined as $U_b^{'} = \sqrt{\frac{2k_b}{3}}$.

Considering the written above expressions it is possible to define the turbulent kinetic energy and the effective viscosity for the dispersed phase as shown in eq.8.16 and eq. 8.17

$$k_a = C_t^2 k_b \tag{8.16}$$

$$\nu_a^{eff} = \nu_a + C_t^2 \nu^t \tag{8.17}$$

The original values for the coefficients for $k - \varepsilon$ model are shown in table 8.2

Coefficient	Value
C_μ	0.09
C_1	1.44
C_2	1.92
C_3	1.0
σ_k	1.0
σ_ϵ	1.3
σ_α	1.0

Table 8.2: Original coefficient values for k-e turbulence model. [39]

The turbulence model has been implemented but not verificated so it should be considered as a starting point for future works.

8.1.4 Discretisation

In order to solve the above equations, the solution domain has to be discretized in time and in space.

The discretisation in time consist of dividing the time in sub-steps. This discretisation is used for transient simulations.

The discretisation in space consist in subdividing the domain in cells. The main topology conditions are that the cells do not overlap and the sum of them fill the domain. The dependant variables can be stored either at the cell centers (collocated) than in the face or vertices (staggered). In [29], Rusche decided to used the collocated arrangement because it is numerically simpler and more efficient altough it could present problems when dealing with the velocity-pressure coupling or the presence of oscillations. Once the variable storage method have been chosen, there is the need to find a method to interpolate the stored values (at the cell center) to the cell faces.

First, the concept of flux need to be presented. The numerical face Flux (F_f) of the variable ϕ is the quantity of ϕ that passes through the face of Surface (S_f). The total flux (F) is the sum for all the element faces. For a Finite volume element formed of 6 quadrilateral faces the graphical and numerical expression of is presented in fig.8.1. The normal vector is positive when it goes towards the exterior.

$$F = \sum_{f=1}^{nf} \phi_f \cdot (n \cdot S)_f \tag{8.18}$$

The central differencing is a second order accurate method. It is defined as:

$$\phi_{f,CD} = f_x \phi_p + (1 - f_x)\phi_n \tag{8.19}$$

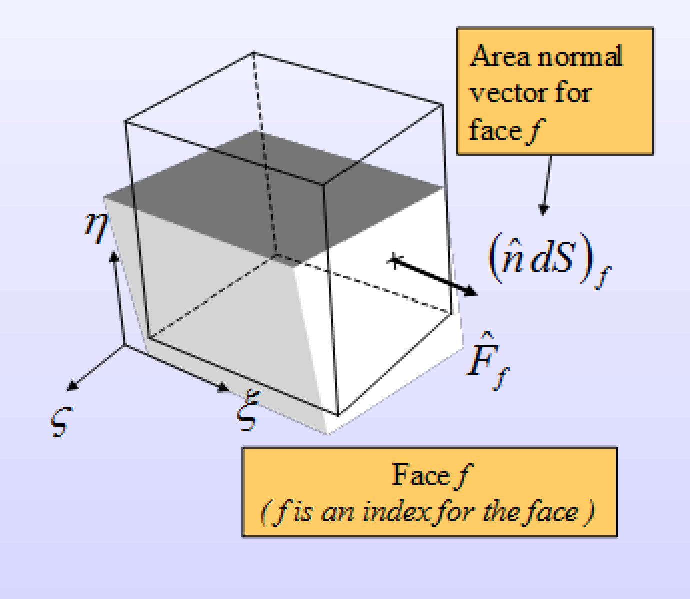

Area normal vector for face f
$(\hat{n}\,dS)_f$
η
$\hat{F}_f$
ς
ξ
Face f
(f is an index for the face)

Figure 8.1: Flux definition for a 6 faces element

where f_x is the distance normal value defined as $\frac{|x_f - x_n|}{|x_f - x_n| + |x_f - x_p|}$

The Upwind differencing is based on the direction of the flow. It is defined as:

$$\phi_{f,UD} = \begin{array}{ll} \phi_P & F \geq 0 \\ \phi_N & F < 0 \end{array} \tag{8.20}$$

The Blended differencing is the combination of the past two schemes. It is defined as:

$$\phi_{f,BD} = (1 - \gamma)\phi_{f,UD} + \gamma\phi_{f,CD} \tag{8.21}$$

Where γ is a parameter taking into account the numerical diffusion.

8.1.4.1 Discretisation of equations

In order to present the numerical treatments of the equation without applying any particular treatment, Weller [34] introduced a finite volume notation. This formulation will be used in the presentation of the solution procedure.

In Weller's notation, an implicit operator is written as $[[\mathcal{L}\,[\phi]]]$ where ϕ is the dependent variable. Instead, the explicit operator is written without brackets.

Phase Momentum Equation

For numerical purposes, it will be convenient to decompose the Reynolds stress in a diffusive term and a correction term (eq. 8.22). Practically, a term is added and will be later substracted in order to modify the terms in the Boussinesq expression (eq. 8.23).

$$\overline{R}_{\varphi}^{eff} = \overline{R}_{\varphi}^{effD} + \overline{R}_{\varphi}^{effC} \tag{8.22}$$

where $\overline{R}_{\varphi}^{effD} = -\nu_{\varphi}^{eff}\left(\nabla\overline{U}_{\varphi}\right)$ and $\overline{R}_{\varphi}^{effD} = \overline{R}_{\varphi}^{eff} + \nu_{\varphi}^{eff}\left(\nabla\overline{U}_{\varphi}\right)$, according to:

$$\overline{R}_{\varphi}^{eff} = -\nu_{\varphi}^{eff}\left(\nabla\overline{U}_{\varphi}^{T} - \frac{2}{3}I\nabla\cdot\overline{U}_{\varphi}\right) + \frac{2}{3}Ik_{\varphi} \tag{8.23}$$

Considering the left hand of eq.8.8:

$$\frac{\partial\overline{U}_{\varphi}}{\partial t} + \overline{U}_{\varphi}^{\tau}\cdot\nabla\overline{U}_{\varphi} - \nabla\cdot(\nu_{\varphi}^{eff}\nabla\overline{U}_{\varphi}) + \nabla\cdot\overline{R}_{\varphi}^{effC} + \frac{\nabla\alpha_{\varphi}}{\alpha_{\varphi}}\cdot\overline{R}_{\varphi}^{effC} \tag{8.24}$$

where $\overline{U}_{\varphi}^{\tau}$ is the total phase velocity and is defined as : $\overline{U}_{\varphi}^{\tau} = \overline{U}_{\varphi} - \nu_{\varphi}^{eff}\frac{\nabla\alpha_{\varphi}}{\alpha_{\varphi}}$

Applying the discretisation rules of Weller, the l.h.s discretized term of eq.8.24 is:

$$\begin{aligned}\tau_{\varphi} = \left[\left[\frac{\partial[\overline{U}_{\varphi}]}{\partial t}\right]\right] + \left[\left[\nabla\cdot(\phi_{\varphi}^{\tau}\left[\overline{U}_{\varphi}\right])_{f(\phi_{\varphi}^{\tau},S)}\right]\right] - \left[\left[\nabla\cdot(\phi_{\varphi}^{\tau}\left[\overline{U}_{\varphi}\right])\right]\right] \\ - \left[\left[\nabla\cdot(\nu_{\varphi}^{eff}\nabla\left[\overline{U}_{\varphi}\right]) + \nabla\cdot\overline{R}_{\varphi}^{effC} + \tfrac{\nabla\alpha_{\varphi}}{<\alpha_{\varphi}>+\delta}\cdot\overline{R}_{\varphi}^{effC}\right]\right]\end{aligned} \tag{8.25}$$

where ϕ_{φ}^{τ} is the total phase flux and is defined in eq. 8.26. The operator $\nabla_{f}^{\perp}$ is the product between face normal and and face gradient : $\nabla_{f}^{\perp}\phi = n\cdot\nabla_{f}\phi$

$$\phi_{\varphi}^{\tau} = \phi - \nu_{\varphi}^{eff}\frac{\nabla_{f}^{\perp}\alpha_{\varphi}}{\alpha_{\varphi f} + \delta} \tag{8.26}$$

In the r.h.s. of eq. 8.8 there are two terms that need further modeling before being discretised: the inter-phase term and the pressure gradient.

The discretised interphase momentum transfer for the dispersed phase ($\mathcal{M}_{\mathrm{a}}$) is formed by different terms as presented in table 8.1.

The discretisation of the drag term ($\mathcal{M}_{\mathrm{a}}^{\mathrm{d}}$) is treated semi-implicitly as shown in eq .8.27. The implicit part of the following expression is the buoyancy term

$$\mathcal{M}_{\mathrm{a}}^{\mathrm{d}} = \mathcal{M}_{\mathrm{a}}^{\mathrm{d_{imp}}} + \mathrm{M}_{\mathrm{a}}^{\mathrm{d_{exp}}} = \frac{\mathrm{A_d}}{\rho_{\mathrm{a}}}(\overline{\mathrm{U}}_{\mathrm{b}} - [[\overline{\mathrm{U}}_{\mathrm{a}}]]) = \frac{\mathrm{K_a}}{\alpha_{\mathrm{a}}}(\overline{\mathrm{U}}_{\mathrm{b}} - [[\overline{\mathrm{U}}_{\mathrm{a}}]]) \tag{8.27}$$

The discretisation for the virtual mass term ($\mathcal{M}_{\mathrm{a}}^{\mathrm{vm}}$) is also treated semi-implicit as shown in eq. 8.28

$$\mathcal{M}_{\mathrm{a}}^{\mathrm{vm}} = \frac{\mathrm{A_{vm}}}{\rho_{\mathrm{a}}}(\frac{\mathrm{D}\overline{\mathrm{U}}_{\mathrm{b}}}{\mathrm{Dt}} - \left[\left[\frac{\mathrm{D}[\overline{\mathrm{U}}_{\mathrm{a}}]}{\mathrm{Dt}}\right]\right]) \tag{8.28}$$

where the explicit substantive derivate is defined as shown in eq.8.29.

$$\frac{D\overline{U}_b}{Dt} = \frac{\partial \overline{U}_b}{\partial t} + \nabla\cdot(\phi_b U_{bf}) - \overline{U}_b \nabla\cdot(\phi_b) \tag{8.29}$$

The lift term is treated explicitly as presented in eq.8.30 .

$$\mathcal{M}_{\mathrm{a}}^{\mathrm{l}} = \frac{A_{lb}}{\rho_a}\overline{U}_r \times (\nabla \times \overline{U}_b) \tag{8.30}$$

The discretised momentum equation is written as:

$$\tau_\varphi = -\frac{\nabla \overline{p}}{\rho_\varphi} + g + \mathcal{M}_\varphi \tag{8.31}$$

The term for the pressure gradient is treated implicitly, but as a part of the pressure equation that is introduced in the following paragraph.

In order to improve convergence a predictor-corrector method is used. A corrector for the velocity need to be derived from the momentum equation. In [30] Hill introduced and expression for the average phase velocity correction . This term is obatained by dividing the momentum equation in linear terms and non linear ones (gradient of pressure and phase fraction).

$$\overline{U}_\varphi = \frac{(\mathcal{A}_\varphi)_H}{(\mathcal{A}_\varphi)_D} - \frac{\nabla \overline{p}}{\rho_\varphi (\mathcal{A}_\varphi)_D} + \frac{K_\varphi}{\rho_\varphi (\mathcal{A}_\varphi)_D} + \frac{g}{(\mathcal{A}_\varphi)_D} \tag{8.32}$$

where $\mathcal{A}_\varphi$ contains the terms of buoyancy drag ($M^{\mathrm{d_{imp}}}$), lift ($\mathcal{M}^{\mathrm{l}}$) and virtual mass ($\mathcal{M}^{\mathrm{vm}}$). $()_D$ contains the diagonal terms whereas $()_H$ contains off diagonal terms.

Volumetric Continuity equation. The pressure equation

The volume equation of the mixture for a biphasic system is eq. 8.10 . It will be used in order to derive an implicit equation for the pressure.

The continuity equation is formulated at the cell faces:

$$\nabla\cdot(\alpha_{af}\phi_a + \alpha_{af}\phi_a) \tag{8.33}$$

The generic face flux is defined as:

$$\phi_\varphi = \phi^*_\varphi - \left(\frac{1}{\rho_\varphi \left(\mathcal{A}_\varphi\right)_D}\right)_f \mid S \mid \nabla^\perp_f \overline{p} \tag{8.34}$$

The ϕ^*_φ is the flux corrector prediction and is defined as:

$$\phi^*_\varphi = \left(\frac{\left(\mathcal{A}_\varphi\right)_H}{\left(\mathcal{A}_\varphi\right)_D}\right)_f \cdot S - \left(\frac{1}{\rho_\varphi \left(\mathcal{A}_\varphi\right)_D}\right)_f \frac{A_{\alpha f} \mid S \mid \nabla^\perp_f \overline{p}}{\alpha_{\varphi f}} \tag{8.35}$$

By combining the above equations, a implicit expression for the pressure gradient is found:

$$\left[\left[\nabla\cdot\left(\left(\alpha_{af}\left(\frac{1}{\rho_a \left(\mathcal{A}_\mathrm{a}\right)_D}\right)_f + \alpha_{bf}\left(\frac{1}{\rho_b \left(\mathcal{A}_\mathrm{b}\right)_D}\right)_f \nabla[\overline{p}]\right)\right)\right]\right] = \nabla\cdot(\alpha_{af}\phi^*_a + \alpha_{af}\phi^*_a) \tag{8.36}$$

It is important to remark that this reformulation (eq. 8.36) for the volumetric mixture equation not have to obey to continuity since this is ensured by considering each phase continuity equation that will be presented in the following paragraph.

8.1.4.2 Phase continuity

The approach used to discretise the phase continuity equation by Rusche derives from the one presented by Weller in [34]. Eq. 8.9 is reformulated in function of the dispersed phase fraction:

$$\frac{\partial \alpha_a}{\partial t} + \nabla\cdot(\overline{U}\alpha_a) + \nabla\cdot(\overline{U}_r\alpha_a(1-\alpha_a)) = 0 \tag{8.37}$$

where $\overline{U}$ is the mixture average velocity, defined as $\overline{U} = \alpha_a\overline{U}_a + \alpha_b\overline{U}_b$. The discretisation of eq. 8.37 leads to :

$$\left[\left[\frac{\partial\left[\alpha_a\right]}{\partial t}\right]\right] + \left[\left[\nabla\cdot(\phi[\alpha_a]_{f(\phi,S)})\right]\right] + \left[\left[\nabla\cdot\left(\phi_{ra}[\alpha_a]_{f(\phi_{ra},S)}\right)\right]\right] = 0 \tag{8.38}$$

where $\phi_{ra} = \alpha_{bf(-\phi_r,S)}\phi_r$ and $\phi_r = \phi_a - \phi_b$

Turbulence model

In order to assure the diagonal dominance of the matrix, the positive term in the production and source terms are discretised explicitly whereas the negative are discretised implicitly as shown in eq. 8.39 and eq. 8.40.

$$\begin{gathered}\left[\left[\tfrac{\partial k_b}{\partial t}\right]\right] + \left[\left[\nabla\cdot\left(\phi_b\left[k_b\right]_f\right)\right]\right] - \left[\left[\nabla\cdot\left(\phi_b\left[k_b\right]\right)\right]\right] - \left[\left[\nabla\cdot\left(\frac{\nu_b^{eff}}{\sigma_k}\nabla\left[k_b\right]\right)\right]\right] = \\ P_b - \left[\left[\tfrac{\epsilon_b}{k_b}\left[k_b\right]\right]\right] + \left[\left[\tfrac{2\alpha_a A_d(Ct^2-1)}{\rho_b}\left[k_b\right]\right]\right] + \tfrac{A_d\nu^t}{\rho_b\sigma_\alpha}\tfrac{\nabla\alpha_a\cdot\overline{U}_r}{\alpha_b}\end{gathered} \tag{8.39}$$

$$\begin{gathered}\left[\left[\tfrac{\partial\varepsilon_b}{\partial t}\right]\right] + \left[\left[\nabla\cdot\left(\phi_b\left[\varepsilon_b\right]_{f(\phi_b,S)}\right)\right]\right] - \left[\left[\nabla\cdot\left(\phi_b\left[\varepsilon_b\right]\right)\right]\right] - \left[\left[\nabla\cdot\left(\frac{\nu_b^{eff}}{\sigma_\epsilon}\nabla\left[\varepsilon_b\right]\right)\right]\right] = \\ P_b C_1\tfrac{\epsilon_b}{k_b} - \left[\left[C_2\tfrac{\varepsilon_b}{k_b}\left[\varepsilon_b\right]\right]\right] + \left[\left[\tfrac{2\alpha_a A_d C_3(Ct_2-1)}{\rho_b}\left[k_b\right]\right]\right]\end{gathered} \tag{8.40}$$

8.2 Solution Procedure

Once the equations have been presented and discretized, an algorithm have to be used in order to solve the mathematical problem for the two fluid methodology. In the implementation of *twoPhaseEulerFoam*, the mathematical procedure chosen by Rusche was the one created by Politis [35] and named as $PISO-2P$. This procedure is also the one applied for the six phase solver that will be later introduced.

$PISO-2P$ is an evolution of the Predictor corrector solver (PEC) , named as $PISO$, implemented for single phase simulation by Issa. It consist in:

1. Velocity prediction
2. Flux prediction
3. Solution of the Pressure equation
4. Correction of fluxes
5. Correction of velocities

The final solution procedure, including coefficients calculation, initialization of $PISO$ loop, $PISO-Loop$, solution of the turbulence model and phase fraction final calculation and its equations are summarized in 6 steps:

1. Calculation of inter-phase momentum exhange coefficients: A_d, A_l and A_vm (table 8.1)
2. Solution of the discretized momentum equation. Initial estimation for phase velocities (eq. 8.31)
3. $PISO$ loop
   (a) Prediction of fluxes (eq. 9.19).
   (b) Solution of the discretized pressure equation (eq. 8.36).
   (c) Correction of fluxes (eq. 8.34).
   (d) Correction of velocities (eq.8.32)
4. Correction of virtual Mass interphase transfer term explicit derivative (eq. 8.29).
5. Solution of discretised dispersed phase equation (eq. 8.39 and eq. 8.40).

8.3 Implementation in OpenFoam

8.3.1 OpenFoam

The $OpenFoam$ software in an open source CFD package made by $Nabla\ Ltd.$ It is based in the $C++$ programming language and it is divided in libraries. Doing so, it is open for the user to add self developed applications into the package. The version used for this thesis is the 1.6 OpenFoam package.

By default, $OpenFoam$ contains different applications known as solvers. Each solver is used to solve the flow for different physical situations. As a way of example, there are solvers able to simulate the Diesel injection ($DieselEngineFoam$), incompressible flows ($simpleFoam$) or even multiphase flows ($twoPhaseEulerFoam$).

In the original package of Openfoam there is also third party software used for the preprocessing and the postprocessing the solution. $ParaView$ software will be used in this thesis to carry out the postprocessing.

The core of the Openfoam libraries are the group of functions fvm and fvc. fvm is used to calculate implicit derivatives while fvc is used for explicits derivatives. The comparison of this method with the Finite Volume notation of Weller can be seen in table 8.3 for time derivatives (ddt OpenFoam function).

Operation	FV notation	OpenFoam functions
implicit derivation	$\left[\left[\frac{\partial[\phi]}{\partial t}\right]\right]$	$fvm :: ddt(phi)$
explicit derivation	$\frac{\partial\phi}{\partial t}$	$fvc :: ddt(phi)$

Table 8.3: EXAMPLE OF THE USAGE OF fvc AND fvm

For extra information of the fvc and fvm group of functions please refer to the OpenFoam user and programmer guide [33].

Once the solver has been chosen, it is applied to solve the flow fields for a discretized case. In the following sections, the structure of the solver, $twoPhaseEulerFoam$ will be presented. Afterwards, the structure of a generic case folder will be summarized.

8.3.2 $twoPhaseEulerFoam$ Solver files

Generally, an OpenFoam solver is formed by a core $C++$ file ($.C$) that calls other $C++$ files or libraries ($.H$). In the current case, the core is $twoPhas-EulerFoam.C$. This file contains the structure of the Solution procedure section above presented and to the libraries containing the functions used to calculate the different variables

There are also other folders in the solver main directory (table 8.5).

By default there are other implemented C_d models other than Schiller Naumann [38], but they will not be used in this report.

The implemented $OpenFoam$ solver is able to simulate flows where the dispersed phase becomes the continuum phase using the phase fraction interpolation modifier function weighting the phase (f_a) first proposed by Weller [34] and then developed by Rusche . This is not relevant for the simulations (so $f_a = 0$) that will be carried out. In the code this is avoided by not defining the dispersed phase as *blended*.

File ($*.H$)	Description	Rel. equations (eq.)
$alphaEqn$	Resolution of phase continuity equation	8.38
$CourantNos$	Computation of the Courant number	-
$createFileds$	Creation of the vector and scalar fields	-
$DDtU$	Virtual mass explicit derivate	8.29
$liftDragCoeffs$	Calculation of inter phase term for lift	8.30
$packingLimiter$	Neighboring cells contributions for α	-
$pEqn$	Resolution of the pressure equation	8.36
$readPPproperties$	Particle-Particle Interaction	-
$readTwoPhaseEulerFoamControls$	PISO numerical parameters	-
$UEqns$	Resolution of momentum	8.31
$write$	Output writing controls	-

Table 8.4: $C++$LIBRARY FILES IN THE $twoPhaseEulerFoam$ SOLVER.

Folder	Description	Related Equation (eq.)
interfacialModels	Drag models	8.5
kineticTheoryModels	Kinetic theory (set off)	-
phaseModel	Definition of phase characteristics	-
Make	Compilation system files	-

Table 8.5: SUBFOLDERS IN $twoPhaseEulerFoam$

8.3.3 Case files

A generic case is stored in a folder that is named with the name of the case. In the pre processing phase, this folder contains three subfolders:

1. *system*. It contains the nature of the simulation (steady/unsteady), the numerical schemes to be applied and the time discretisation and other solution management parameters.

2. *constant*. It contains the geometry (in $/polyMesh$) and the definition of the parameters that will be be maintained constant through the simulation, for example, the values of the coefficient of the turbulence model or the value of gravity.

3. 0. It contains a file each variable where there are specified the boundary conditions at time 0.

This structure is common for all OpenFoam cases.

8.4 Code verification

In order to understand and verify the code, a simulated model will be compared with existing published simulations for two phase eulerian-eulerian treated flows.

In [40] Silva and Lage carried out the implementation of a modification in $twophaseEulerFoam$ introducing a code that was able to handle up to five different phases with constant properties.

The studied geometry was a transient bidimensional backward facing step (BFS) which is shown in Fig. 8.2. The continuous phase was oil while the dispersed one is water. The physical properties can be seen in table. 8.6.

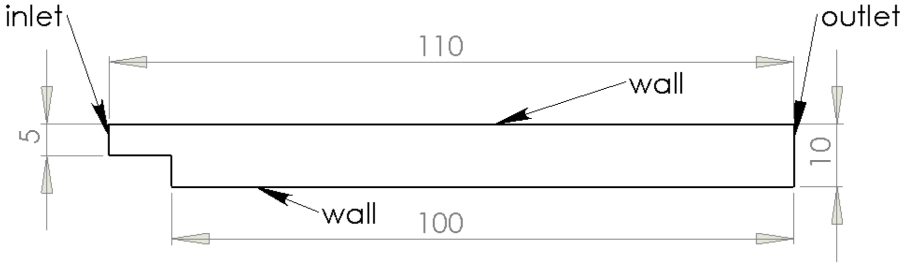


Figure 8.2: Dimensional BFS (dimensions in mm)

Physical Properties	Unit	Oil	Water
density ρ	kg/m^3	900	1000
Kinematic viscosity ν	$\frac{m}{s}^2$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-6}$

Table 8.6: Physical Properties.

The boundaries of the computational domain are: *inlet*, *wall* and *outlet*. The zone that is inside the boundary will be named ad $default\, interior$. Two cases were proposed. The boundary conditions at starting time ($t = 0$) for each case can be seen in table 8.7 and 8.8. The main difference between them is the phase fraction for $default\, interior$: in the second case there will be present stronger gradients testing the numerical stability.

Zone	$p\ [Pa]$	$U_{oil}\ [m/s]$	$U_{water}\ [m/s]$	α_{oil}	α_{water}	k	ε
$inlet$	$\nabla p = 0$	1	1	0.9	0.1	0	-
$outlet$	0	$\nabla U_b = 0$	$\nabla U_a = 0$	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	0	-
$wall$	$\nabla p = 0$	0	0	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	0	-
$default\,interior$	0	0	0	0.9	0.1	0	-

Table 8.7: Boundary conditions t=0 for CASE I.

Zone	$p\ [Pa]$	$U_{oil}\ [m/s]$	$U_{water}\ [m/s]$	α_{oil}	α_{water}	k	ε
$inlet$	$\nabla p = 0$	1	1	0.9	0.1	-	-
$outlet$	0	$\nabla U_b = 0$	$\nabla U_a = 0$	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	-	-
$wall$	$\nabla p = 0$	0	0	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	-	-
$default\,interior$	0	0	0	0	0	-	-

Table 8.8: Boundary conditions t=0 for CASE II.

The simulated time for $Case\,I$ is 0.01 s while it is s for $Case\,II$. The considered times enable the solution to reach steady state. The Reynolds number, considering as characteristic length the step height, is 500. The flow is laminar.

The first step has been to determine the mesh to use for the validation and the sensibility studies. Four different structured meshes have been considered: 2200, 4200, 10500 and 13860 hexaedral elements. The characterisitcal size of the mesh (element length) is greater than the particle size for all the meshes. Two control points haven been determined , where pressure and volumetric fraction will be plotted. The first control point is at $x = 0.0125m$, corresponding to the recirculation bubble. The second point is at $x = 0.075m$ (70% of the total channel length). The test case used is the first one and the Courant maximum number , used to establish the time step, was set to 0.3 .

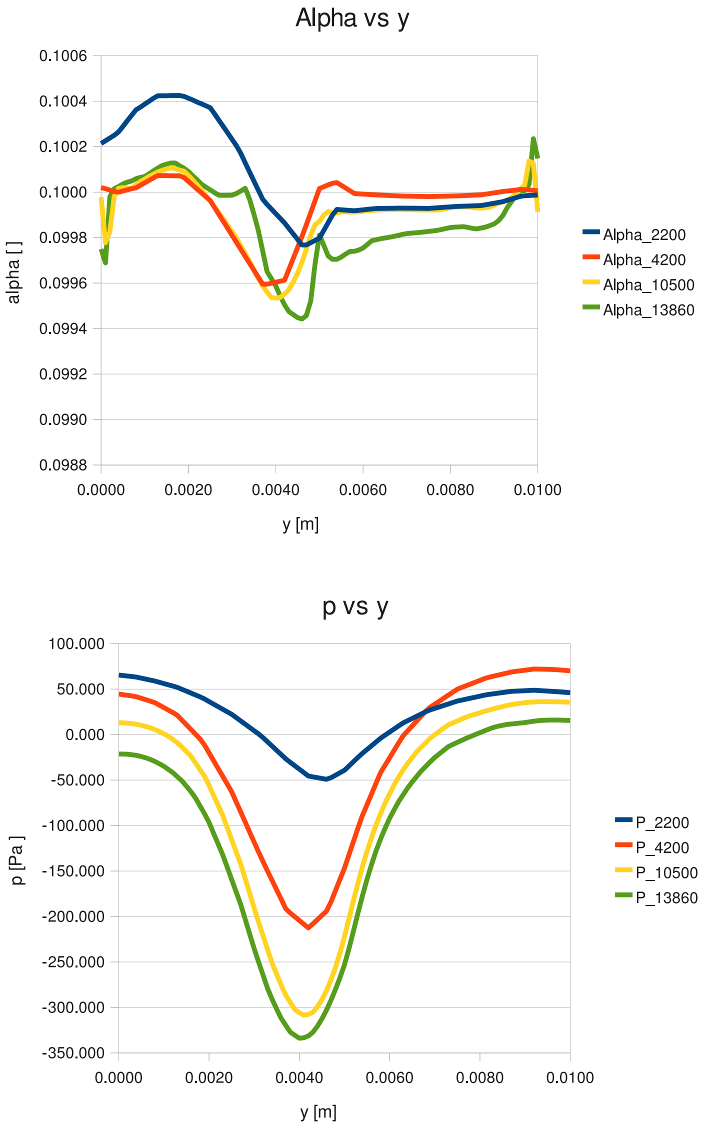


Figure 8.3: P AND ALPHA AT POINT I IN MESH CONVERGENCE.

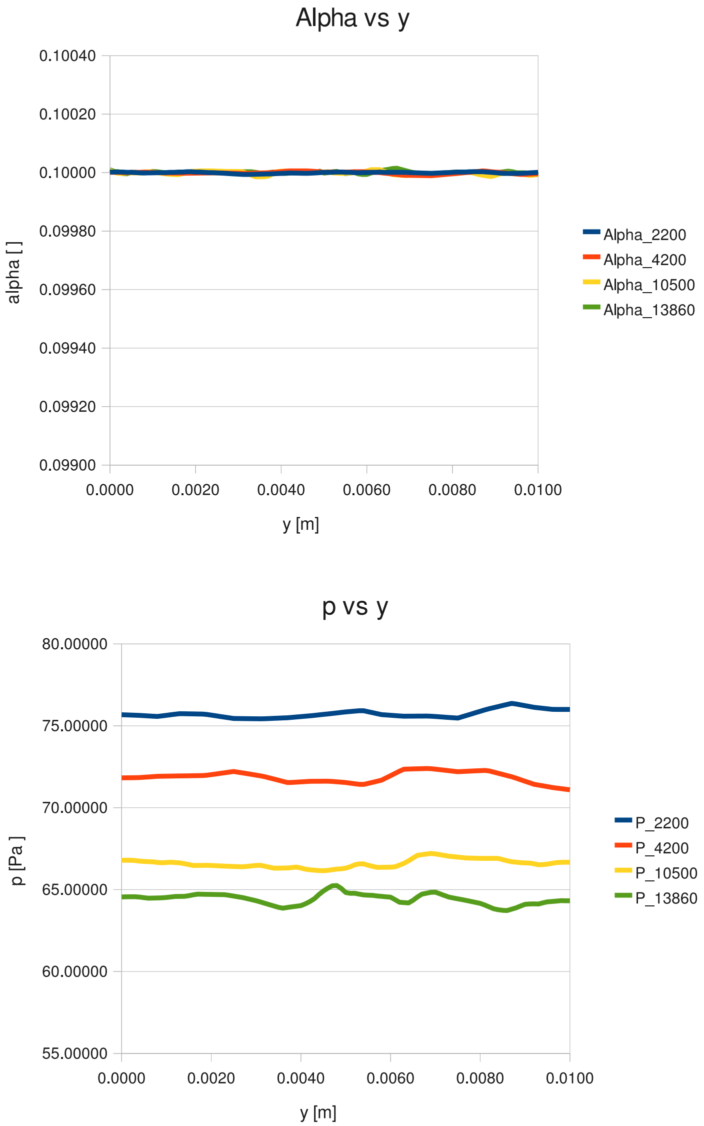


Figure 8.4: P AND ALPHA AT POINT 2 IN MESH CONVERGENCE.

For point 1 (Fig. 8.3), *alpha* varies less than 0.2% from the nominal value for meshes with more than 4200 elements. Instead, the pressure shows a stronger variation as function of the mesh elements. The increase of maximum p between 10500 and 13860 elements is 8.10%. Instead, for point two (Fig. 8.4) there is no variation for *alpha* and the change for p between 10500 and 13860 elements is 3.6% .

The computing time for the meshes are shown in table 8.9 .

Mesh Elements	Computing time [s]
2200	70
4200	172
10500	640
13860	1616

Table 8.9: COMPUTING TIME FOR MESH CONVERGENCE (CASE I).

Considering the relation between accuracy and computational cost, it has been decided to do the verification with the 10500 elements test.

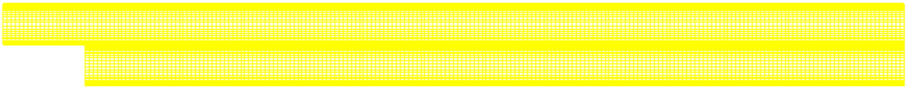


Figure 8.5: 10500 ELEMENTS MESH.

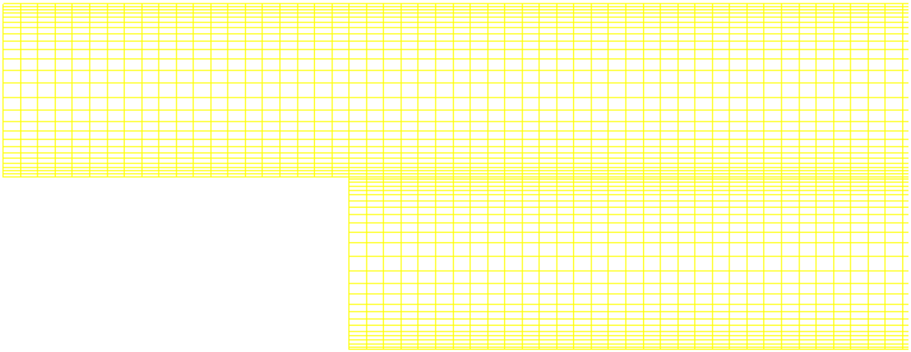


Figure 8.6: 10500 ELEMENTS MESH STEP DETAIL.

The second part regarded has been to study the influence of the Courant (Co) number in the simulation (Fig 8.7). Lower Co number assure stability but could induce numerical problems and also an increase in the computing time.

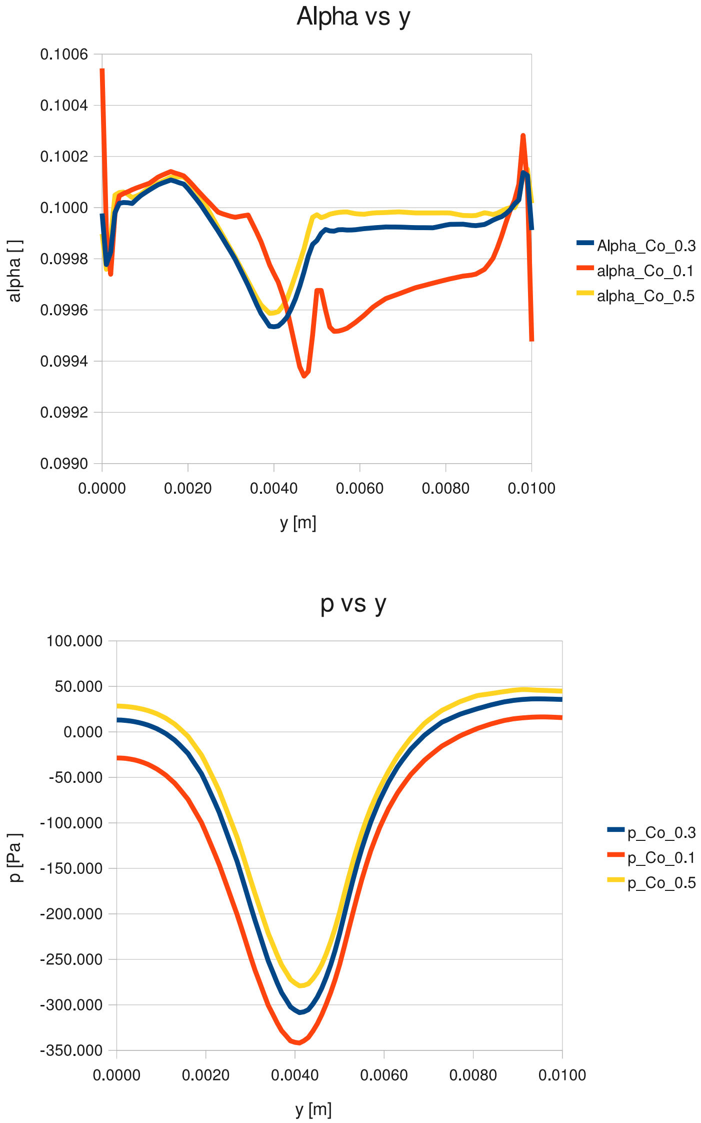
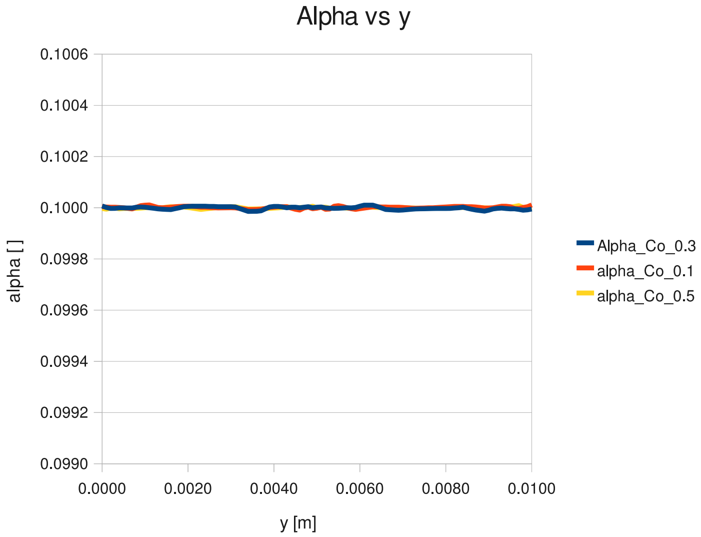


Figure 8.7: p and alpha at point 1 in Courant sensitivity.



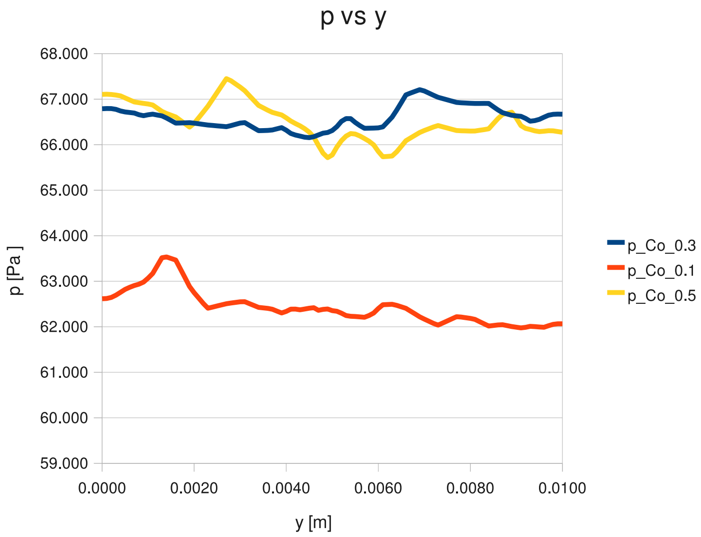


Figure 8.8: P AND ALPHA AT POINT 2 IN COURANT SENSITIVITY.

The evolution of the magnitudes when decreasing the *Co* number follows the same behavior than increasing the number of mesh elements. Considering the accuracy versus the time calculation, a *Co* of 0.3 will be used for the simulation. In [40] also a *Co* number of 0.3 has been used.

The final step of the verification has been to simulate the previous two simulation cases and compare them with the results from [40].

CASE I

The results for simulation for *case I* shows the same evolution for *p* and *alpha* than the one presented in [40] (Fig 8.9 and 8.10). Notwithstanding, the minimum obtained through simulation have not the same values than the ones from Silva. This have been explained to the lack of refinement of the mesh in y direction and the lack of information of the numerical schemes and parameters used by Silva. In [40] a 16000 elements mesh have been used but there is not any image published of that in order to compare the mesh refinement.

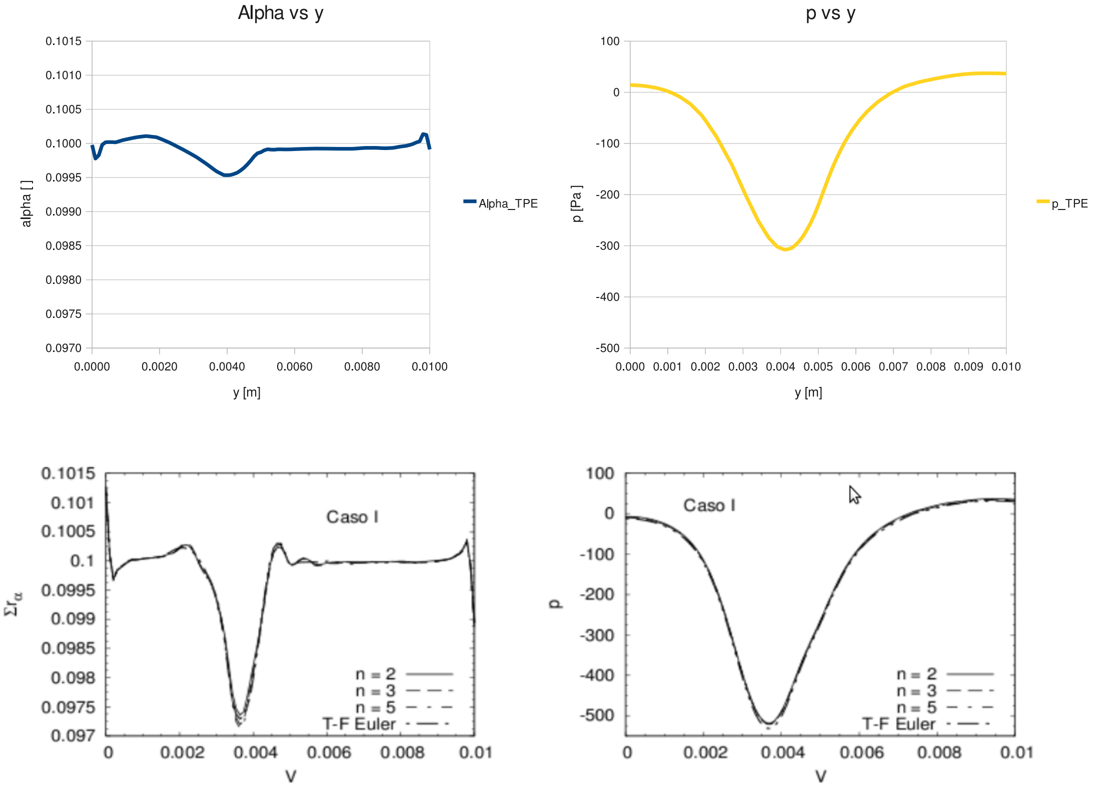


Figure 8.9: Case I results at point 1. Top: current results. Bottom: silva's results [40]

The *twoPhaseEulerFoam* solver for *case I* have been also compared to a single phase laminar solver named as *simpleFoam*. The pressure is not plotted because for simpleFoam shows an almost constant behavior around $-20\ Pa$. The velocity profiles for both solvers are reported in 8.10. For the two phase case, the velocity refers to the mixture velocity.

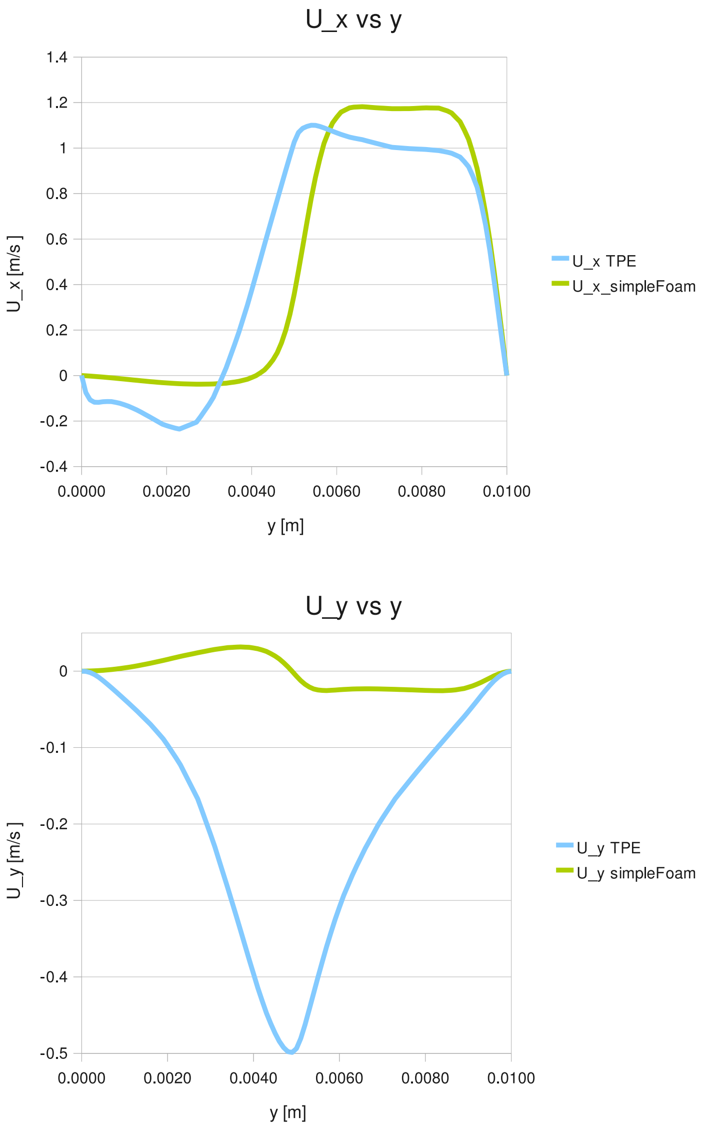


Figure 8.10: Velocity profiles comparison at point 1 case I .

It is concluded that the presence of the dispersed phased enhaces the recirculation zone.

Case II

This case shows stronger gradients in the boundary conditions (table 8.8) as long as the initial conditions for the internal volume is set to 0. The results are shown in Fig. 8.11 and 8.12 .

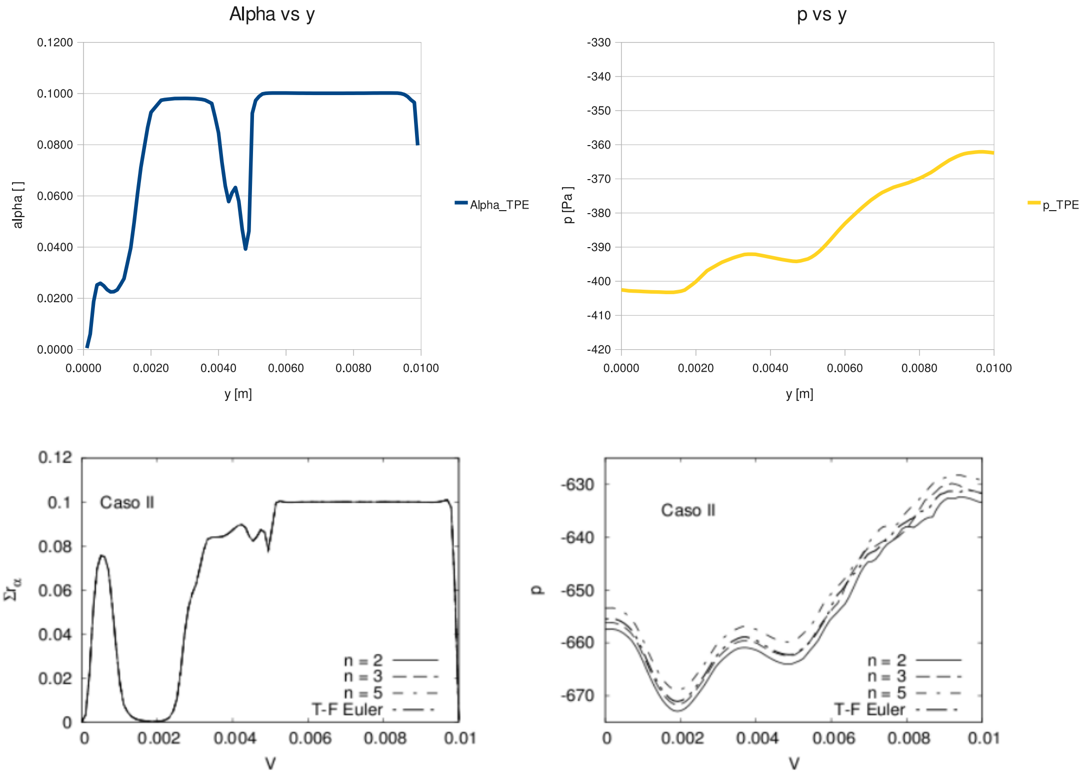


Figure 8.11: CASE II RESULTS. TOP:CURRENT RESULTS. BOTTOM: SILVA'S RESULTS [40]

The *alpha* distribution does not follow the behavior of the results from Silva. It has been explained with the same reason than in *case I*: lack of refinement of the mesh. For pressure it has been obtained the same behavior, a similar difference between maximum and minimum (40 Pa) but not its magnitude. It is explained by the fact that is it unknown is a turbulent kinetic energy (k) has been imposed as boundary condition or the not knowing of all the numerical schemes used in [40].

For the velocity profiles a comparison with $simpleFoam$ will be also presented (Fig. 8.12).

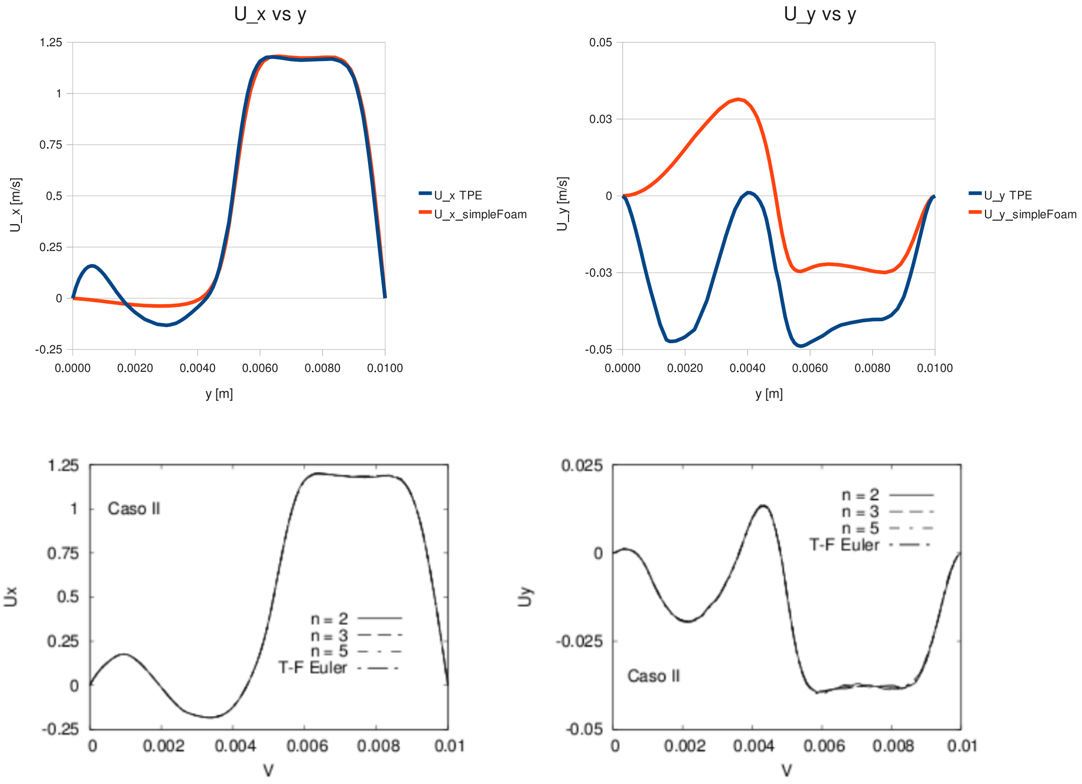


Figure 8.12: Velocity profiles at point 1 for case II.[40]

When comparing the results to Silva's ones it is found that the is a full concordance in values and behavior. Instead, the comparison with *simpleFoam* results it is found than the vorticity is increased in the two phase code.

8.4.1 Lift coefficient sensibility

As presented previously in this chapter, the lift force could be considered in the dispersed phase. The influence of lift coefficient in *alpha* and *p* will be studied. The mesh is the one with the 10500 elements mesh and the *case I* boundary conditions will be considered. The same control points than for the mesh and *Co* convergence have been considered.

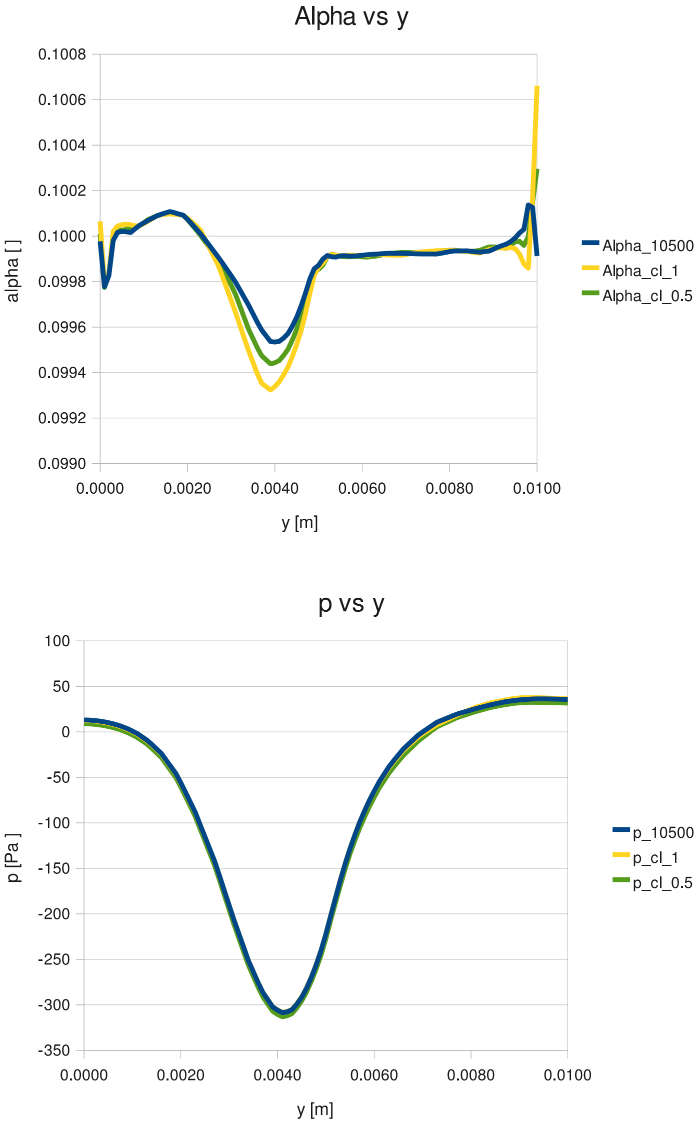


Figure 8.13: p and alpha in point 1 for cl sensitivity study.

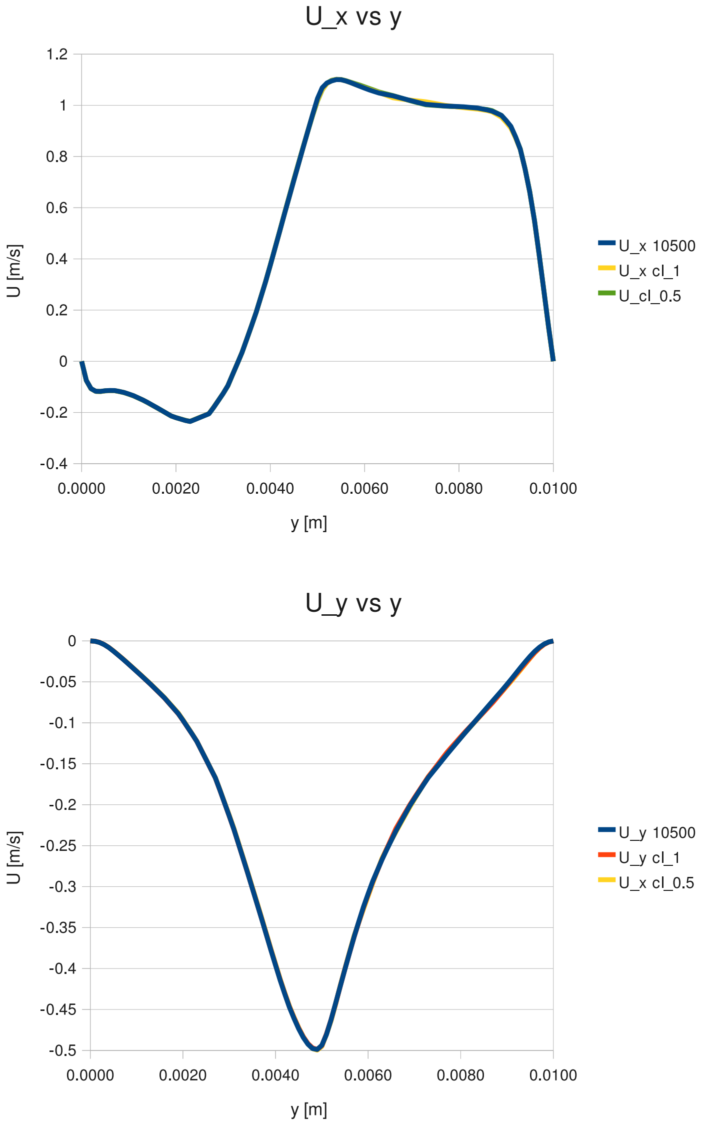


Figure 8.14: U_X AND U_Y AT POINT 1 FOR CL SENSITIVITY .

For point 1(Fig. 8.13 and 8.14), the pressure and velocity profiles are the same but there is a change of the behavior in *alpha* distribution: the higher C_l is, the lower *alpha* value is presented in the recirculation zone.

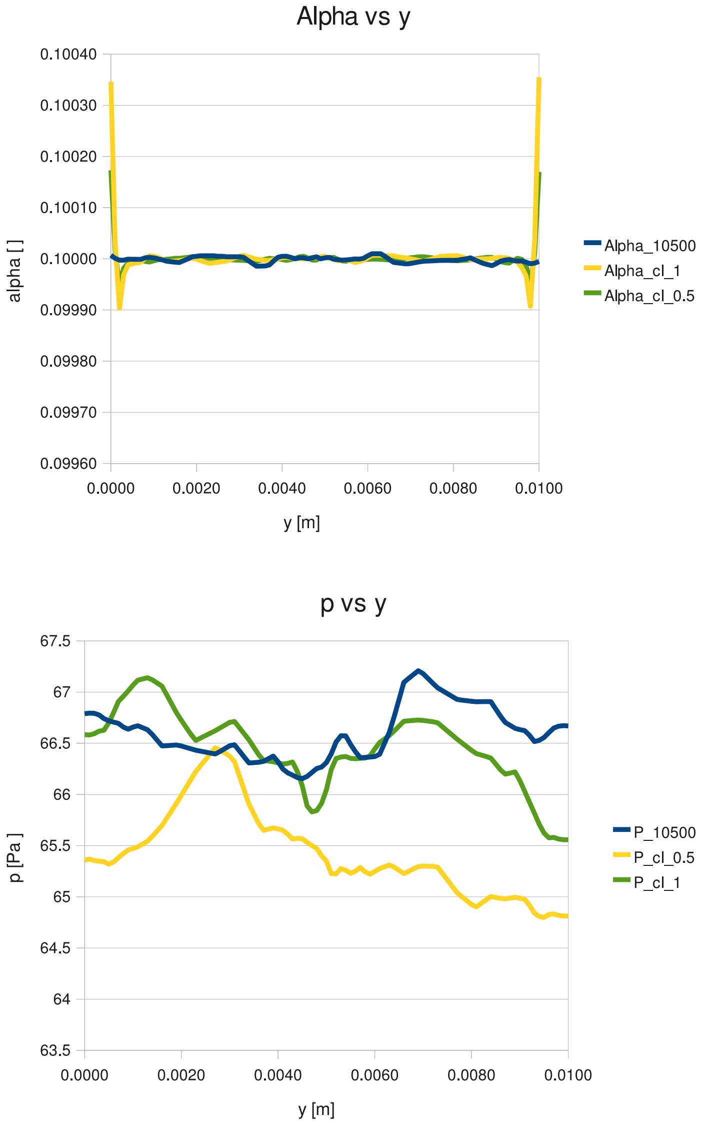


Figure 8.15: P AND ALPHA IN POINT 2 FOR CL SENSITIVITY STUDY.

Instead, in the second point (Fig. 8.15) for pressure and velocity there is no significant difference but for the higher C_l leads to an accumulation of dispersed phase close to the wall.

8.4.2 Virtual mass coefficient sensibility

The inter-phase momentum exchange also can take into account the virtual mass effect. While for the validation virtual mass coefficient (C_{vm}) was set to 0, now its effects in the solution will be introduced.

For Point 1 (Fig. 8.16), the higher C_{vm} is the more dispersed fraction is accumulated in the recirculation bubble. On the contrary, the higher this value is, the lower the pressure is in the recirculation bubble core. It has been found no difference for the velocity.

For point 2 (Fig 8.17), the only remarkable difference is found for the pressure near the upper wall for $C_{vm} = 1$ that shows a decreasing behavior.

Phisically, the virtual mass is the inertia transferred between phases due to the relative acceleration between phases. So for the recirculation the higher C_{vm} is more energy is transferred between continuum and dispersed phase. This could result in an accumulation of alpha in the recirculation core.

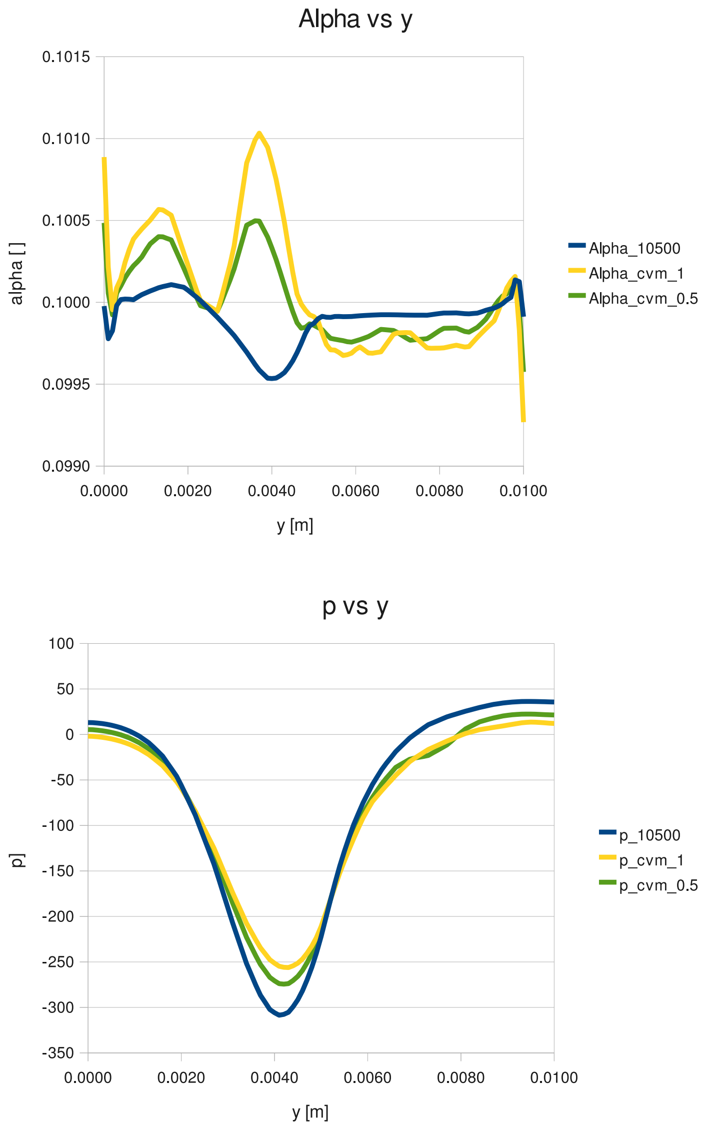


Figure 8.16: p and alpha at point 1 for cvm sensitivity.

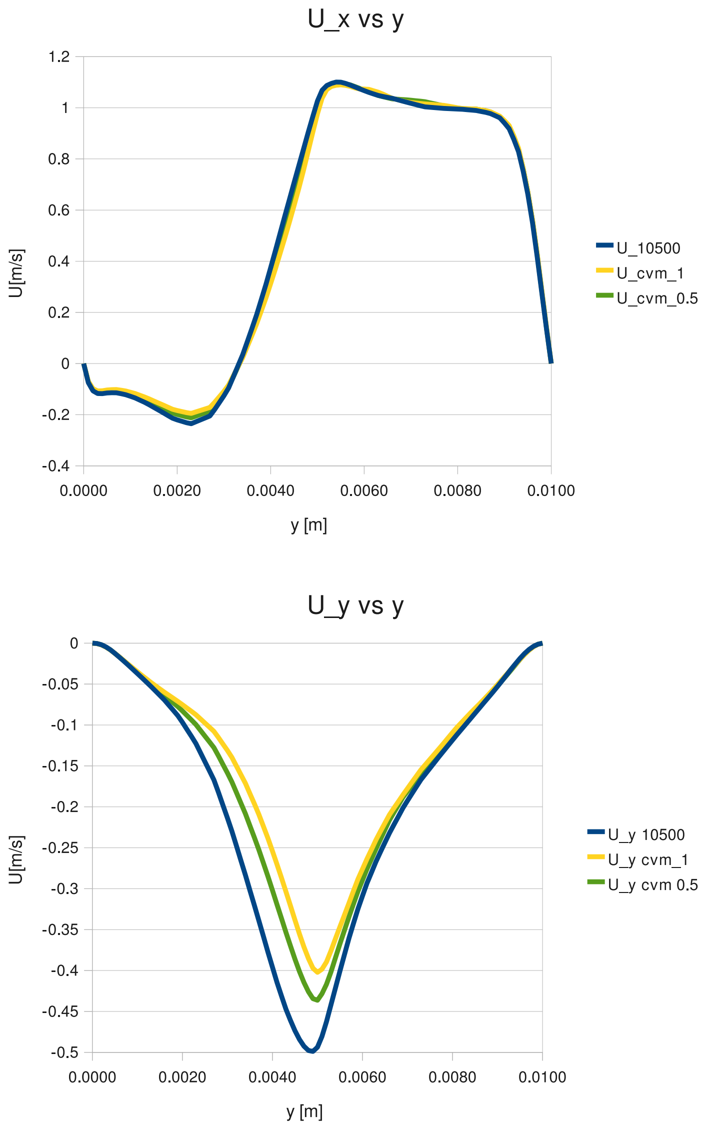


Figure 8.17: U_X AND U_Y AT POINT 1 FOR CVM SENSITIVITY.

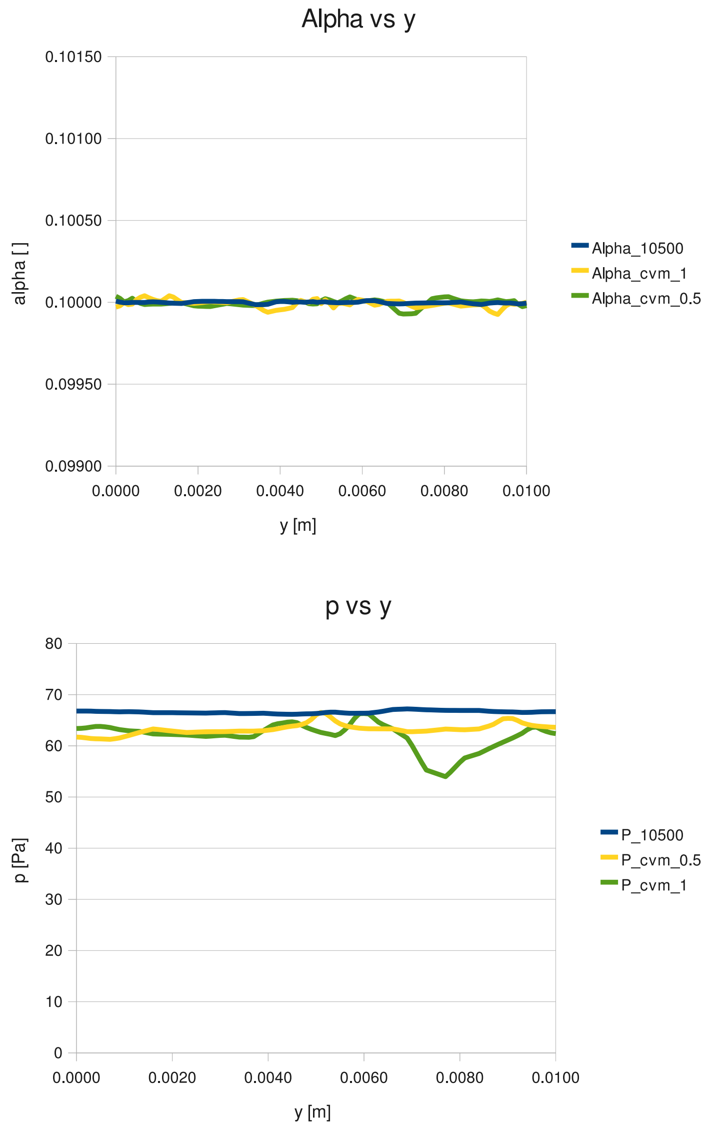


Figure 8.18: cvm sensitivity at point 2.

Chapter 9

$SixPhaseEulerFoam$

9.1 Governing Equations

The equations governing *sixPhaseEulerFoam* are an evolution of the ones presented previously for the *twoPhaseEulerFoam* in the previous chapter. The main difference is found in the equations for the continuum phase (b) as long as the interphase momentum exhange is not the same in magnitude of a single dispersed phase but the sum of all of them. This fact reports problems when calculating the solution due to the bounding of the volumetric fraction to 0 and 1. Another difference is reported when using adaptive time step based on the Co number, where the solution converges for a higher value of it. The third difference is found during the definition of the fluxes taking part in the pressure implicit equation, where the fluxes of the six phases are involved.

In the following sections, the theoretical bases for *sixPhaseEulerFoam* are presented, taking special attention to the main differences and problems when confronted to *twoPhaseEulerFoam*.

9.1.1 Momentum conditionally averaged equation

Contrasting the two phase case, the momentum conditionally averaged equation will be different for dispersed phases (eq. 9.1), named as a_1, a_2 ,a_3, a_4 and a_5; and the continuous phase named as b (eq. 9.2).

$$\frac{\partial \alpha_{a_i}\overline{U}_{a_i}}{\partial t} + \nabla\cdot\left(\alpha_{a_i}\overline{U}_{a_i}\overline{U}_{a_i}\right) + \nabla\cdot\left(\alpha_{a_i}\overline{R}^{eff}_{a_i}\right) = -\frac{\alpha_{a_i}}{\rho_{a_i}}\nabla\cdot(\overline{p}) + \alpha_{a_i} g + \frac{\overline{M}_{a_i}}{\rho_{a_i}} \quad (9.1)$$

$$\frac{\partial \beta\overline{U}_{b}}{\partial t} + \nabla\cdot\left(\beta\overline{U}_{b}\overline{U}_{b}\right) + \nabla\cdot\left(\beta\overline{R}^{eff}_{b}\right) = -\frac{\beta}{\rho_{b}}\nabla\cdot(\overline{p}) + \beta g - \sum_{a_i=a_1}^{a_5}\frac{\overline{M}_{a_i}}{\rho_{a_i}} \quad (9.2)$$

Following the same strategy than in the two phase case, in order to avoid discontinuity when the volumetric fraction is low the momentum equation is considered in its intensive formulation (eq. 9.3 and 9.4).

$$\frac{\partial \overline{U}_{a_i}}{\partial t} + \nabla \cdot \left(\overline{U}_{a_i} \overline{U}_{a_i} \right) + \frac{\nabla \cdot \left(\alpha_{a_i} \overline{R}_{a_i}^{eff} \right)}{\alpha_{a_i}} = -\frac{1}{\rho_{a_i}} \nabla \cdot (\overline{p}) + g + \frac{\overline{M}_{a_i}}{\alpha_{a_i} \rho_{a_i}} \tag{9.3}$$

$$\frac{\partial \overline{U}_b}{\partial t} + \nabla \cdot \left(\overline{U}_b \overline{U}_b \right) + \frac{\nabla \cdot \left(\beta \overline{R}_b^{eff} \right)}{\beta} = -\frac{1}{\rho_b} \nabla \cdot (\overline{p}) + g - \sum_{a_i = a_1}^{a_5} \frac{\overline{M}_{a_i}}{\alpha_{a_i} \rho_{a_i}} \tag{9.4}$$

9.1.2 Continuity conditionally averaged equation

The velocity of the dispersed phase a_i is defined as ($\bar{U}$ refers to the mixture velocity):

$$\overline{U}_{a_i} = \bar{U} + \alpha_{a_i} \bar{U}_{ri,b} + \sum_{a_j = a_1; a_j \neq a_i}^{a_5} \alpha_j \bar{U}_{ri,,j} \tag{9.5}$$

Where:

$$\overline{U}_{ri,b} = \bar{U}_{\alpha_i} - \bar{U}_b \tag{9.6}$$

$$\bar{U}_{ri,j} = \overline{U}_{\alpha_i} - \overline{U}_{\alpha_j} \tag{9.7}$$

The equation for the volumetric fraction of dispersed phase i is presented in eq. 9.8 .

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\overline{U} \alpha_i \right) + \nabla \cdot \left(\beta \bar{U}_{ri,b} \alpha_i \right) + \nabla \cdot \left(\sum_{a_j = a_1; a_j \neq a_i}^{a_5} \alpha_j \bar{U}_{ri,j} \alpha_i \right) = 0 \tag{9.8}$$

There is no equation for β because it is obtained by direct subtraction of α (eq. 9.9).

$$\beta = 1 - \sum_{\alpha_i = \alpha_1}^{\alpha_5} \alpha_i \tag{9.9}$$

9.1.3 Turbulence model

The velocity and the viscosities involved during the validation and the booster case application of the code makes the flows considered laminar. As for the two phase code, the turbulence model has been implemented but not verificated so it should be considered as a starting point for future works.

The standard equations of transport (eq.9.10 and 9.11) for k and ε are complemented with extra terms (eq.9.12 and 9.13) taking into account the influence of each dispersed phase α_i ($S_{k_{\alpha_i}}$ and $S_{\varepsilon_{\alpha_i}}$).

$$\frac{\partial k_b}{\partial t} + \left(\overline{U}_b \cdot \nabla\right) k_b - \nabla \cdot \left(\frac{\nu_b^{eff}}{\sigma_k} \nabla k_b \right) = P_b - \epsilon_b - \sum_{a_i=a_1}^{a_5} S_{k_{\alpha_i}} \tag{9.10}$$

$$\frac{\partial \epsilon_b}{\partial t} + \left(\overline{U}_b \cdot \nabla\right) \epsilon_b - \nabla \cdot \left(\frac{\nu_b^{eff}}{\sigma_\epsilon} \nabla \epsilon_b \right) = \frac{\epsilon_b}{k_b} \left(C_1 P_b - C_2 \epsilon_b\right) - \sum_{a_i=a_1}^{a_5} S_{\epsilon_{\alpha_i}} \tag{9.11}$$

where P_b is the production of turbulent kinetic energy defined as $P_b = 2\nu_b^{eff} \left(\nabla \overline{U}_b\right) \cdot \left(\nabla \overline{U}_b + \left(\nabla \overline{U}_b\right)^T\right)$, ν_b^{eff} is the effective viscosity of the continuum and ν^t is the turbulent viscosity defined as $\nu^t = C_\mu \frac{k_b^2}{\epsilon_b}$.

$$S_k = \frac{2 k_b \alpha_{a_i} A_d (C_t C_t - 1)}{\rho_b} + \frac{A_d \nu^t}{\rho_b \sigma_\alpha} \frac{\nabla \alpha_{a_i}}{\beta} \cdot \overline{U}_{r_i,b} \tag{9.12}$$

$$S_\epsilon = \frac{2 C_3 \epsilon_b \alpha_{a_i} A_d (C_t C_t - 1)}{\rho_b} \tag{9.13}$$

The response coefficient (C_t, eq. 8.15) is assumed to be the same for all the phases as long as the dispersed phases are though to have characteristics in the same scale order. If it was not the case, a C_t coefficient for each dispersed phases may be considered.

The problem found with the implementation of $k{-}\varepsilon$ model in either *sixPhase−EulerFoam* and *twoPhaseEulerFoam*, have been the existent implementation of wall functions. In constrast with other OpenFoam solver, the wall functions are implemented directly in the turbulence library, taking away the possibility of changing its functions without a deep change in the code.

It has ben found that $k{-}\varepsilon$ gives a phisically reasonable solution when the Re number is still low but once the velocity increases there is a immediate diverging and non physical meaning in the solution. As stated before, it is an open door for future developments in the implementation of the multiphase turbulence code, specially in the definition and adequacy of the wall functions to be used.

9.1.4 Discretisation

The principles of spatial and time discretisation have been presented previously is 8.1.4 section and are the sames used for the six phase code.

9.1.4.1 Discretisation of equations

Phase momentum equation

The discretisation for the dispersed phase momentum equation is the same than presented before in section 8.1.4.1 for the two phase case. The index φ should be replaced by a_i. Instead, the expression for the continuum phase (b) have to be reformulated due to the interphase momentum term ($\vec{\mathcal{M}}$), in particular, its drag term.

$$\begin{aligned}\tau_b = &\left[\left[\frac{\partial\left[\overline{U}_b\right]}{\partial t}\right]\right] + \left[\left[\nabla\cdot(\phi_b^\tau \left[\overline{U}_b\right])_f\right]\right] - \left[\left[\nabla\cdot(\phi_{b_i}^\tau \left[\overline{U}_b\right])\right]\right] \\ &- \left[\left[\nabla\cdot(\nu_b^{eff}\nabla\left[\overline{U}_b\right]) + \nabla\cdot\overline{R}_b^{effC} + \tfrac{\nabla\beta}{<\beta>+\delta}\cdot\overline{R}_b^{effC}\right]\right]\end{aligned} \tag{9.14}$$

where $\phi_{a_i}^\tau$ is the total phase flux and is defined as $\phi_{b_i}^\tau = \phi_b - \nu_b^{eff}\frac{\nabla_f^\perp\beta}{\beta_f+\delta}$. $\overline{U}_b^\tau$ is the total phase velocity and is defined as : $\overline{U}_b^\tau = \overline{U}_b - \nu_b^{eff}\frac{\nabla\beta}{\beta}$

$$\tau_b = -\frac{\nabla\overline{p}}{\beta} + g + \frac{1}{\rho_b\beta}\sum_{a_i=a_1}^{a_5}\alpha_i K_\alpha \bar{U}_{a_i} \tag{9.15}$$

the momentum correction equation for the continuum is shown in eq. 9.16 .

$$\overline{U}_b = \frac{(\mathcal{A}_b)_H}{(\mathcal{A}_b)_D} - \frac{\nabla\overline{p}}{\rho_b(\mathcal{A}_b)_D} - \frac{1}{\rho_b\beta(\mathcal{A}_b)_D}\sum_{a_i=a_1}^{a_5}\alpha_{a_i}K_{a_i}\bar{U}_{a_i} \tag{9.16}$$

Volumetric continuity equation . The pressure equation.

If the continuity equation is formulated at face centers:

$$\nabla\cdot\left(\beta_f\phi_b + \sum_{a_i=a_1}^{a_5}\alpha_i\phi_{a_i}\right) \tag{9.17}$$

The generic face flux is defined by interpolating the momentum correction to the face centers. The flux for for the continuum in eq. 9.18 :

$$\phi_{a_i} = \phi_{a_i}^* - \left(\frac{1}{\rho_{a_i}\left(\mathcal{A}_{\mathrm{a_i}}\right)_D}\right)_f \mid S \mid \nabla_f^\perp\overline{p} \tag{9.18}$$

The ϕ_φ^* is the flux corrector prediction and is defined as:

$$\phi_a^* = \left(\frac{(\mathcal{A}_{\mathrm{a_i}})_H}{(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f \cdot S - \left(\frac{K_{a_i}}{\rho_{a_i}\,(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f \phi_b + \left(\frac{1}{(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f g{\cdot}\mathrm{S} \tag{9.19}$$

For the continuous phase b, the flux corrector expression is:

$$\phi_b^* = \left(\frac{(\mathcal{A}_{\mathrm{a_i}})_H}{(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f \cdot S - \left(\frac{K_{a_i}}{\rho_{a_i}\,(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f \phi_b + \left(\frac{1}{(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f g{\cdot}\mathrm{S} \tag{9.20}$$

By combining the above equations, a implicit expression for the pressure gradient is found:

$$\begin{gathered}\left[\left[\nabla\cdot\left(\left(\beta_f \left(\tfrac{1}{\rho_b(\mathcal{A}_{\mathrm{b}})_D}\right)_f + \textstyle\sum_{a_i=a_1}^{a_5} \alpha_{a_i f} \left(\tfrac{1}{\rho_{a_i}(\mathcal{A}_{\mathrm{a_i}})_D}\right)_f \nabla[\overline{p}]\right)\right)\right]\right] = \\ \nabla\cdot(\alpha_{bf}\phi_b^* + \textstyle\sum_{a_i=a_1}^{a_5} \alpha_{a_i f}\phi_{a_i}^*)\end{gathered} \tag{9.21}$$

Phase continuity

The discretisation of eq. 8.10 is presented in eq. 9.22. The terms for the volumetric fraction are treated implicitly.

$$\left[\left[\frac{\partial\left[\alpha_{a_i}\right]}{\partial t}\right]\right] + [[\nabla\cdot(\phi[\alpha_{a_i}])]] + [[\nabla\cdot\left(\alpha_{bf}\phi_{ri,b}[\alpha_{a_i}]\right)]] + \sum_{a_{j=a_i}}^{a_5}\left[\nabla\cdot\left(\alpha_{a_i f}\phi_{ri,a_j}[\alpha_{a_i}]\right)\right] = 0 \tag{9.22}$$

ϕ, the mixture flux, that appears in the second term satisfies the continuity equation, so α_i is bounded by 1 as well as 0. Instead for the third and fourth terms using ϕ_{ri,a_j} could produce a non bounded solution. The bounding of the volumetric fraction is assured manually by the scheme presented in eq. 9.23. For the $twoPhaseEulerFoam$, this problem was sorted out by using $\phi_{i,r}$ in the convection scheme to interpolate α_a to the face and $-\phi_{i,r}$ in the face interpolation of α_b. This could not be applying as long as there are more than two phases involved.

$$\alpha_{a_i} = max(\delta, \alpha_{a_i}) \tag{9.23}$$

Then β is obtained explicitly by the α_i obtained implicitly. This is done by using eq. 9.9.

Turbulence model

The transport equation for turbulent quantities are discretized as shown in eq. 9.24and eq. 9.25. k and ε are treated implicitly.

$$\begin{gathered}\left[\left[\frac{\partial k_b}{\partial t}\right]\right] + \left[\left[\nabla\cdot\left(\phi_b \left[k_b\right]_f\right)\right]\right] - \left[\left[\nabla\cdot\left(\phi_b \left[k_b\right]\right)\right]\right] - \left[\left[\nabla\cdot\left(\frac{\nu_b^{eff}}{\sigma_k}\nabla\left[k_b\right]\right)\right]\right] = \\ P_b - \left[\left[\frac{\epsilon_b}{k_b}\left[k_b\right]\right]\right] + \sum_{a_i=a_1}^{a_5}\left(\left[\left[\frac{2\alpha_{a_i}A_d(Ct^2-1)}{\rho_b}\left[k_b\right]\right]\right] + \frac{A_d\nu^t}{\rho_b\sigma_\alpha}\frac{\nabla\alpha_{a_i}\cdot\overline{U}_{r,i}}{\beta}\right)\end{gathered} \tag{9.24}$$

$$\begin{gathered}\left[\left[\frac{\partial \epsilon_b}{\partial t}\right]\right] + \left[\left[\nabla\cdot\left(\phi_b \left[\epsilon_b\right]_f\right)\right]\right] - \left[\left[\nabla\cdot\left(\phi_b \left[\varepsilon_b\right]\right)\right]\right] - \left[\left[\nabla\cdot\left(\frac{\nu_b^{eff}}{\sigma_\epsilon}\nabla\left[\epsilon_b\right]\right)\right]\right] = \\ P_bC_1\frac{\epsilon_b}{k_b} - \left[\left[C_2\frac{\epsilon_b}{k_b}\left[\epsilon_b\right]\right]\right] + \sum_{a_i=a_1}^{a_5}\left(\left[\left[\frac{2\alpha_{a_i}A_dC_3(Ct_2-1)}{\rho_b}\left[k_b\right]\right]\right]\right)\end{gathered} \tag{9.25}$$

9.1.5 SixPhaseEulerFoam solver Files

In table 9.1 the main libraries of the code are presented. In table 9.2 the subfolders and its contents are listed. The main difference when compared to *twoPhaseEulerFoam* is the fact has *sixPhaseEulerFoam* has the $k-\varepsilon$ code implemented in an own file and the files for particle interaction and kinetic theory have been removed from the solver folder. The content of the listed files for the six phase solver is different from the two phase code. At the end of the report, the full code will be included.

File ($*.H$)	Description	Rel. equations (eq.)
alphaEqn	Resolution of phase continuity equation	9.22
CourantNos	Co number for b	-
CourantNo	Co number for a_i	-
createFileds	Creation of the vector and scalar fields	-
DDtU	Virtual mass explicit derivate	8.29
liftDragCoeffs	Calculation of inter phase term for lift	8.30
ke	Implementation of 6 phases $k-\varepsilon$	9.24 and 9.25
pEqn	Resolution of the pressure equation	9.21
turbulence	Coefficients and filed creation for $k-\varepsilon$	table 8.2
readSixPhaseEulerFoamControls	PISO numerical parameters	-
UEqns	Resolution of momentum	9.15
write	Output writing controls	-

Table 9.1: $C++$LIBRARY FILES IN THE *sixPhaseEulerFoam* SOLVER.

There are also other folders in the solver main directory (table 9.2).

Folder	Description	Rel. equations (eq.)
interfacialModels	Drag models	8.5
phaseModel	Definition of phase characteristics	-
Make	Compilation system files	-

Table 9.2: Subfolders in *sixPhaseEulerFoam*

9.2 Code verification

The solver code will be validated through simulating the same cases than for *twoPhaseEulerFoamcase*, then the results will be compared to Silva's ones. There will be 5 dispersed phases, equal in physical characteristics, but with a boundary condition for *alpha* of one fifth of the original used in [40]. This is shown in table 9.3 and 9.4 .

Zone	$p\ [Pa]$	$U_{oil}\ [m/s]$	$U_{water}\ [m/s]$	α_{oil}	$\alpha_{a_i}\ [i = 1, .., 5]$	k	ε
inlet	$\nabla p = 0$	1	1	0.9	0.02	0	-
outlet	0	$\nabla U_b = 0$	$\nabla U_a = 0$	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	0	-
wall	$\nabla p = 0$	0	0	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	0	-
$default\,interior$	0	0	0	0.9	0.02	0	-

Table 9.3: Boundary conditions t=0 for CASE I *sixPhaseEulerFoam*.

Zone	p $[Pa]$	U_{oil} $[m/s]$	U_{water} $[m/s]$	α_{oil}	α_{a_i} $[i=1,..,5]$	k	ε
$inlet$	$\nabla p = 0$	1	1	0.9	0.02	0	-
$outlet$	0	$\nabla U_b = 0$	$\nabla U_a = 0$	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	0	-
$wall$	$\nabla p = 0$	0	0	$\nabla \alpha_b = 0$	$\nabla \alpha_a = 0$	0	-
$default\,interior$	0	0	0	0	0	0	-

Table 9.4: Boundary conditions t=0 for CASE II *sixPhaseEulerFoam*.

The computational meshes considered for the mesh convergence have been the sames than the ones used previously, with: 2200, 4400, 10500 and 13860 elements. Two meshes have to be discarted *a priori*. The 2200 elements mesh causes a fast divergence in the solution due to the mesh coarseness. Instead the 13860 elements mesh shows unphysical solutions for the pressure. The Co number for mesh testing have been 0.3. In Fig. 9.1 the simulation results for mesh convergence at point 1 are shown.

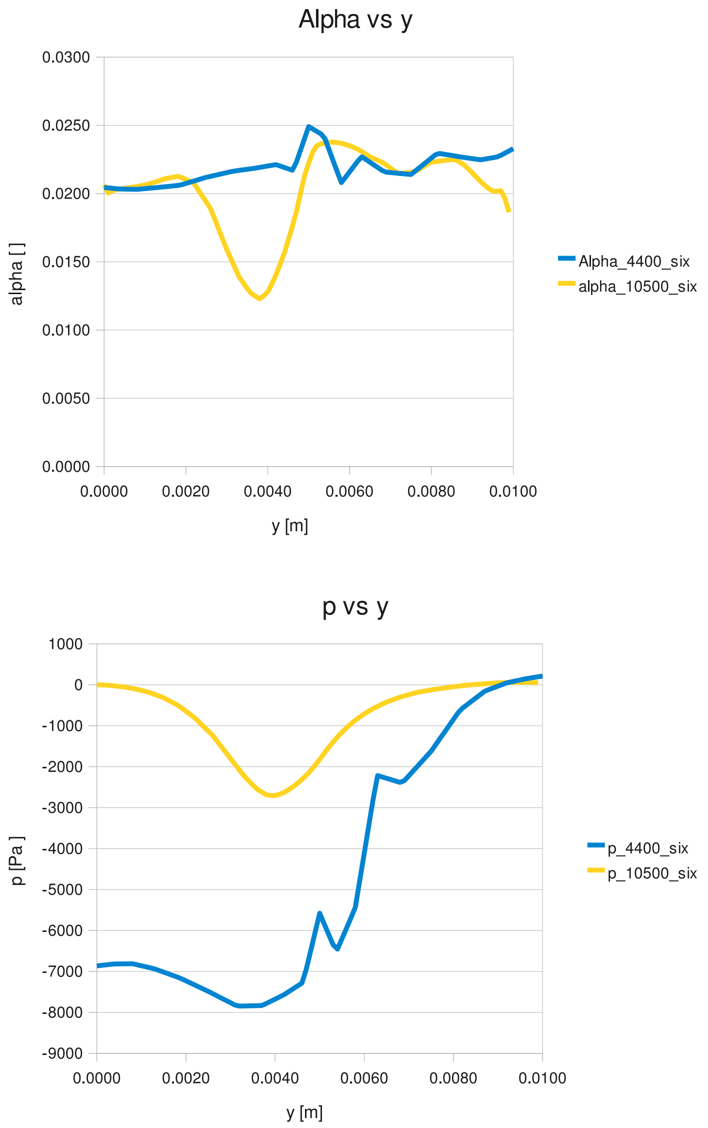


Figure 9.1: MESH CONVERGENCE AT POINT 1.

Considering the relation between accuracy and computational time (table 9.5) the mesh that will be used for the rest of the validation is the one that has 10500 elements.

Mesh Elements	Computing time [s]
2200	NA
4200	246
10500	869
13860	NA

Table 9.5: COMPUTING TIME FOR MESH CONVERGENCE (CASE I) .

For the Courant number sensibility, it has been simulated for *Co*=0.1, 0.3 and 0.5 (Fig. 9.2). The *alpha* diagram refers only to one of the phases as long as all the distributions for the 5 dispersed phases are the same.

As for *twoPhaseSolver* it is observed that the for $Co = 0.5$ and 0.3 converge both for *alpha* and p. Instead, for $Co = 0.1$ the result diverge from the other results. The Courant number chosen for the simulation will be 0.5 due to the elevated computational cost when using a lower *Co* and due to the fact that there is no difference in the pressure between 0.5 and 0.3 and the difference with 0.1 stays within 15% for p and *alpha*.

The final step of the validation has been to simulate the previous two simulation cases and compare them with the results from [40]. A comparison with *twoPhaseEulerFoam* and *simpleFoam* will be also reported.

CASE I

Considering the results shown in Fig. 9.3 and 9.4, it is evident that, although the trent is correct, *sixPhaseEulerFoam* overestimates the values for p and *alpha*. Instead it presents and exact behavior for tangential velocity (U_x).

A further investigation has been carried out in order to determine the nature of the problem.

The mesh sensibility has been presented before so this error is not caused by the computational domain itself. Although, *sixPhaseEulerFoam* presented the incapability to solve in very coarse mesh where *twoPhaseEulerFoam* managed to obtain a phisically reasonable solution.

Several numerical schemes for the divergence terms ($div(phia_i, Ua_i)$), such as Gauss Gamma, Gauss linear or Gauss corrected have been used. There was no significant change in the solution.

Finally the number of correctors for p and *alpha* have been assessed (fig. 9.5) . By default, the code uses two correctors for the *PISO* algorithm and two for alpha. Using one corrector for alpha closes *sixEulerFoam* the solution to Silva's results while for pressure it is found that the optimum value is two.

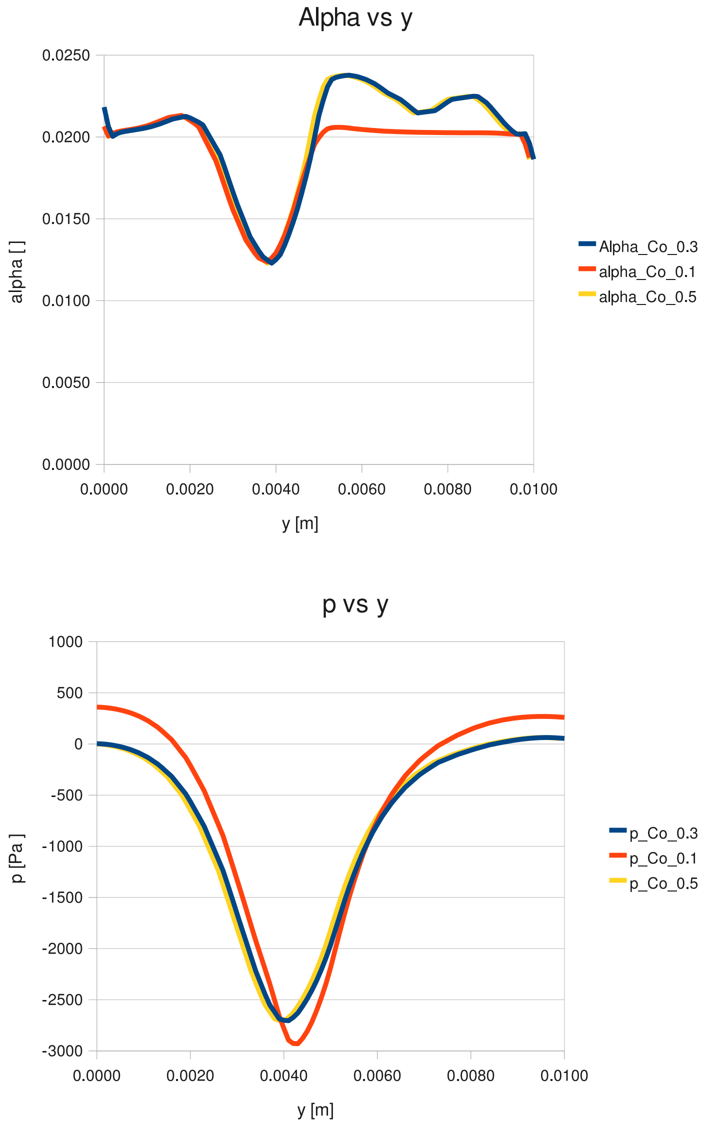


Figure 9.2: COURANT SENSITIVITY AT POINT 1.

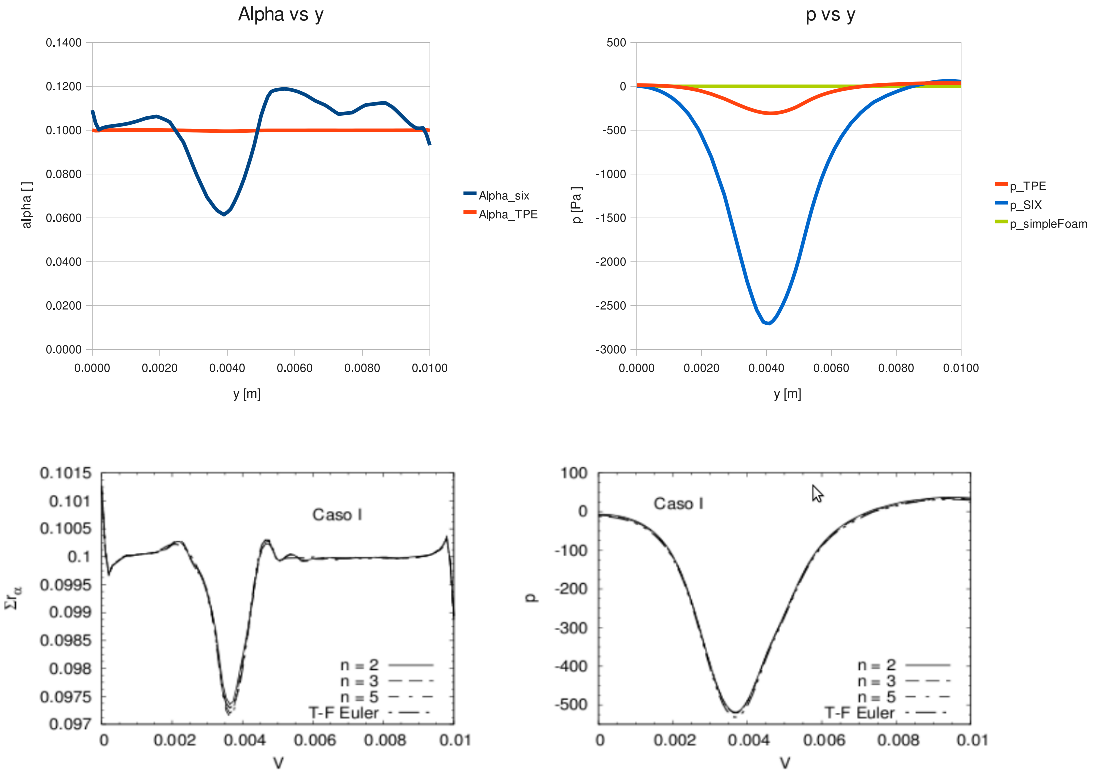


Figure 9.3: Case I results at point 1. top: current results. bottom: silva's results [40]

It is concluded that *sixPhaseEulerFoam* give an exact behavior for the solution but overestimates the solution values so it will be only useful qualitatively. The final reason for the numerical difference has not been found athough several sensitivity studies have been held.

CASE II

For case II a computational problem has been found that has made impossible the application of *sixPhaseEulerFoam*. Due to the strong gradients found in the boundary conditions of *alpha* (set as 0 for the interior of the computational domain), the temporal step have to be low in order to assure stability. It has been found that this value gets stabilized under $1 \cdot 10^{-8} s$. It can be concluded that the code is unable to handle with strong gradients in the boundary conditions. A future development should be the change of the algorithm applied the code in order to make the simulation for this kind of cases feasible.

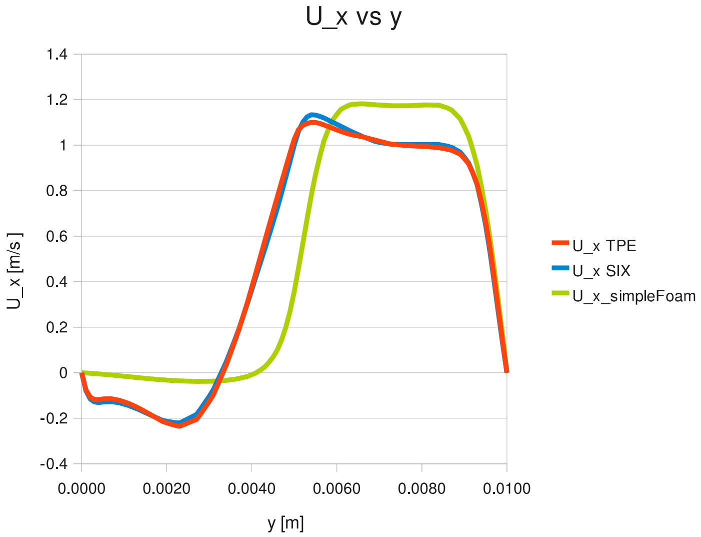


Figure 9.4: Velocity profiles comparison at point 1 for case I.

9.2.1 Lift coefficient sensibility

C_l sensibility results is showed in Fig. 9.6 and 9.8. For point one the expected behavior is found for *alpha* and velocity but the results for pressure seem to be phisically meaningless. It was expected a close results for p , particularly near the wall, and instead they seem to be diverging.

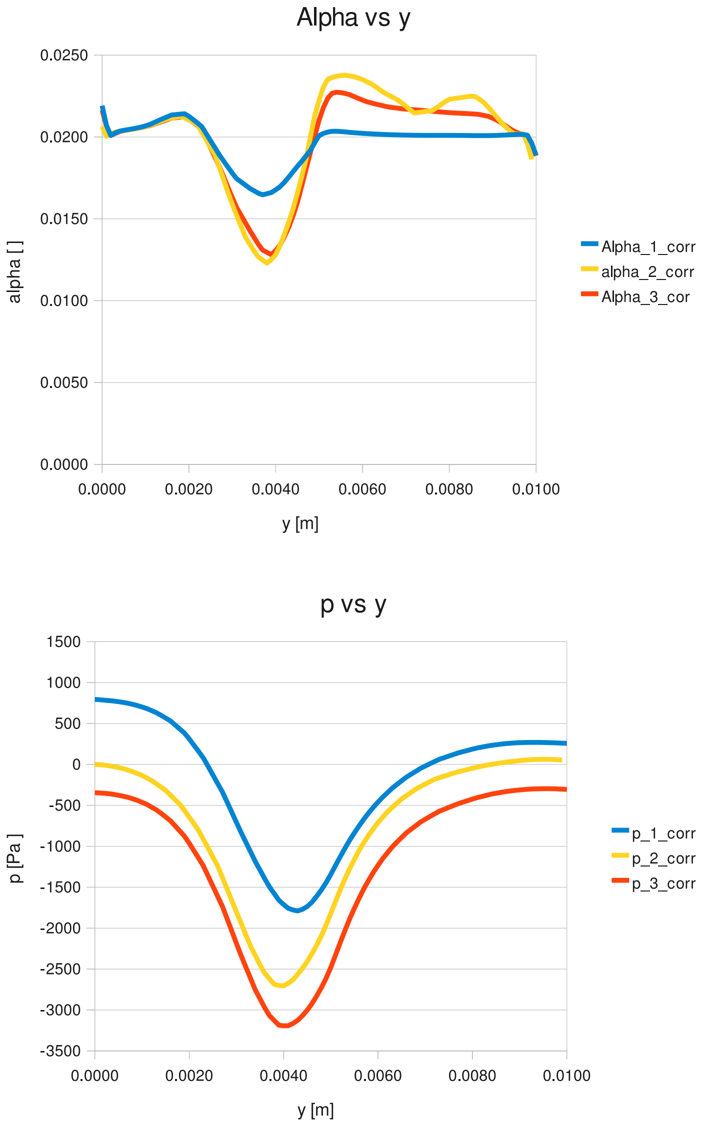


Figure 9.5: CORRECTORS SENSITIVITY AT POINT 1.

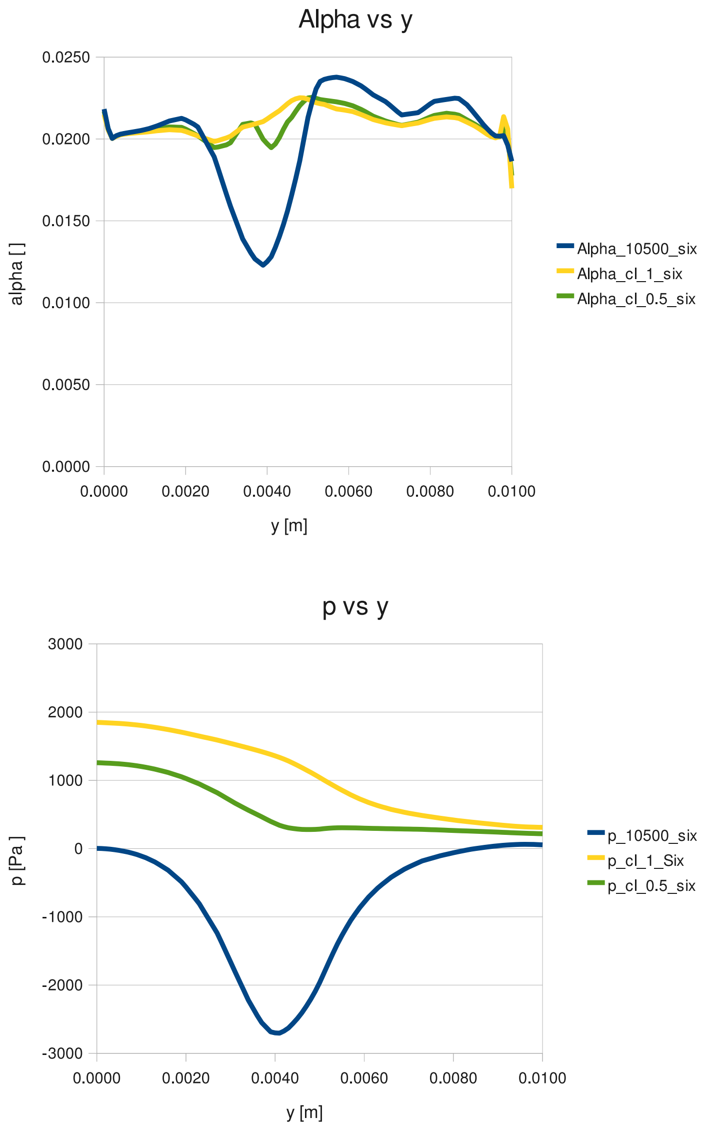


Figure 9.6: CL SENSITIVITY AT POINT 1.

For point 2, it is found exactly the same behavior than for *twoPhaseEulerFoam*. (Fig. 9.8 and 9.9)

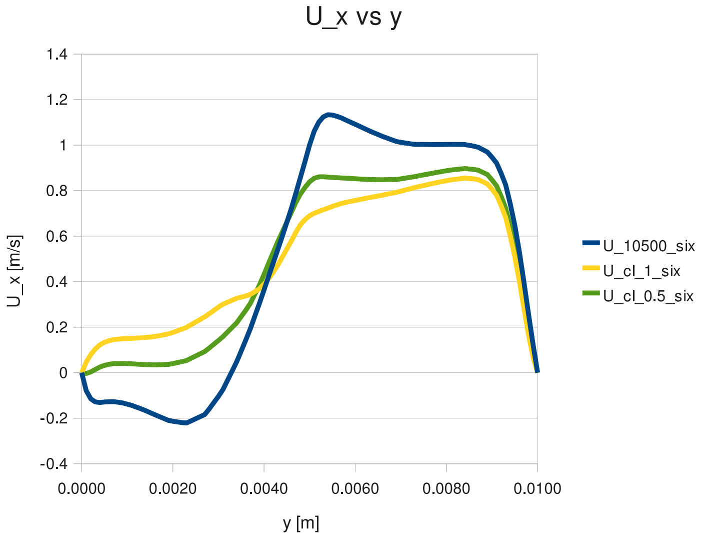


Figure 9.7: U_X AT POINT 1 FOR CL SENSITIVITY STUDY.

9.2.2 Virtual mass coefficient sensibility

The results for virtual mass stability are shown in Fig. 9.10 and 9.12. For point 1, it is found that $sixPhaseEulerFoam$ is not able to handle with the higher value for C_{vm} due to an overestimation of pressure at the recirculation bubble when compared to $twoPhaseEulerFoam$. Instead, for intermediate values of C_{vm} it follows the same behavior for p and $alpha$.

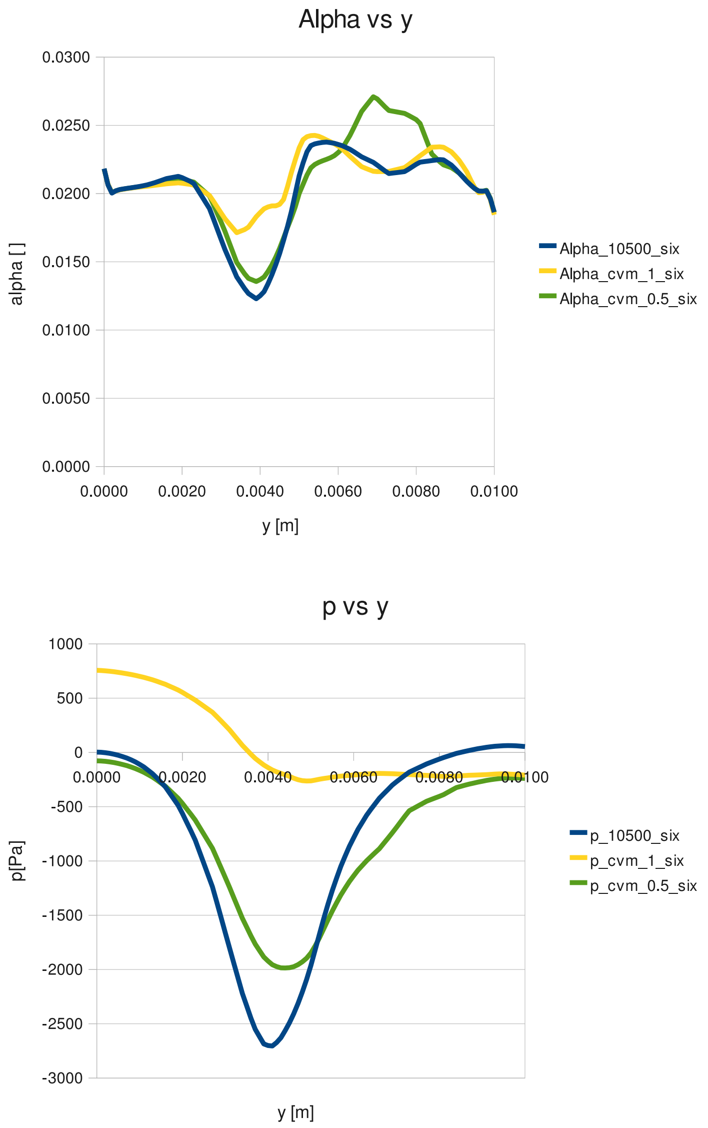


Figure 9.10: P AND ALPHA CVM SENSITIVITY STUDY IN POINT 1 SIXPHASEEULERFOAM.

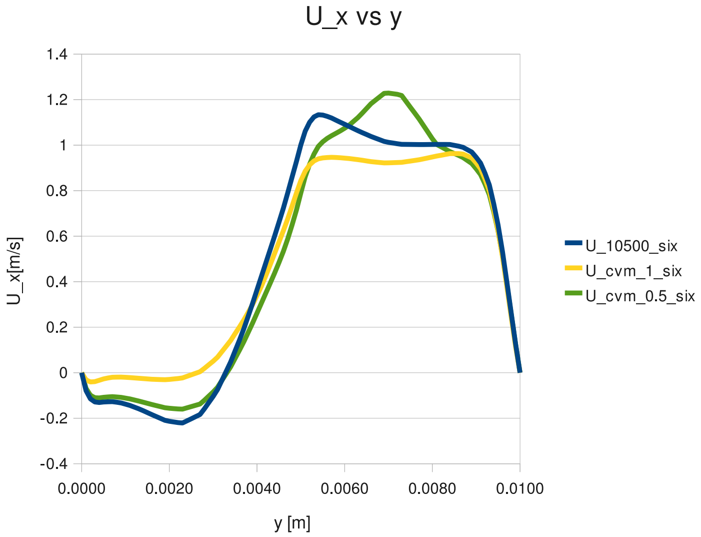


Figure 9.11: U_X AT POINT 1 CVM SENSITIVITY.

For point 2 (Fig. 9.12), no difference for *alpha* and U_x, and the minimal variations in pressure also found for *twoPhaseEulerFoam* .

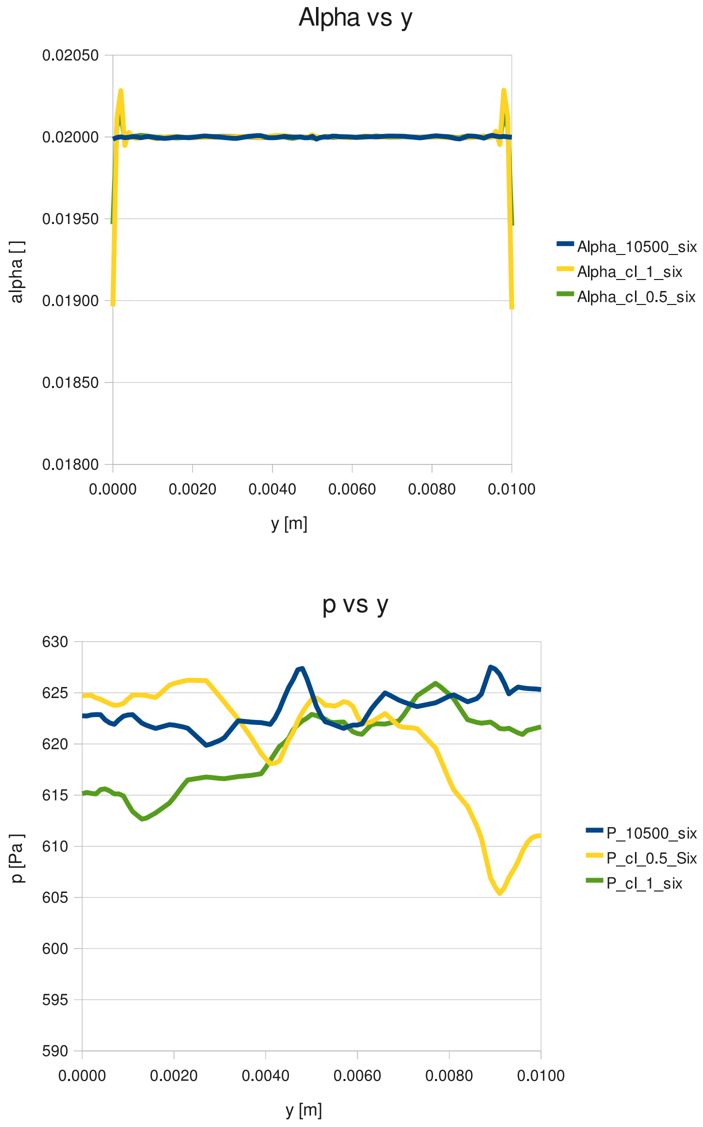


Figure 9.8: cl sensitivity study at point 2.

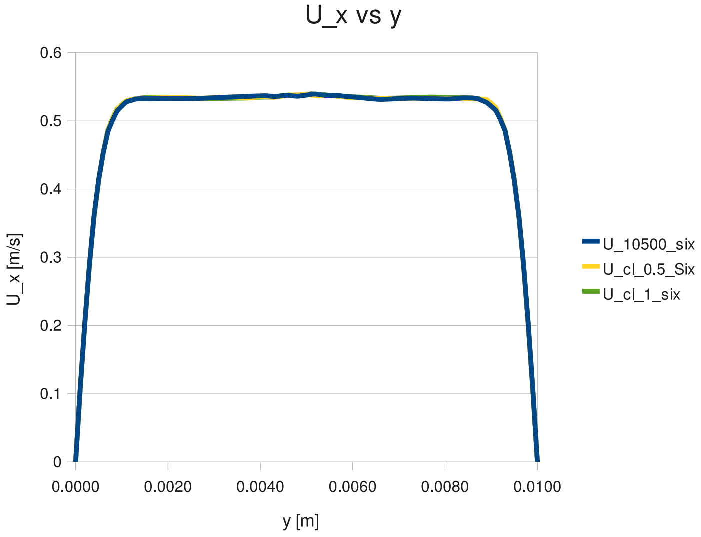


Figure 9.9: U_X AT POINT 2 CL SENSITIVITY.

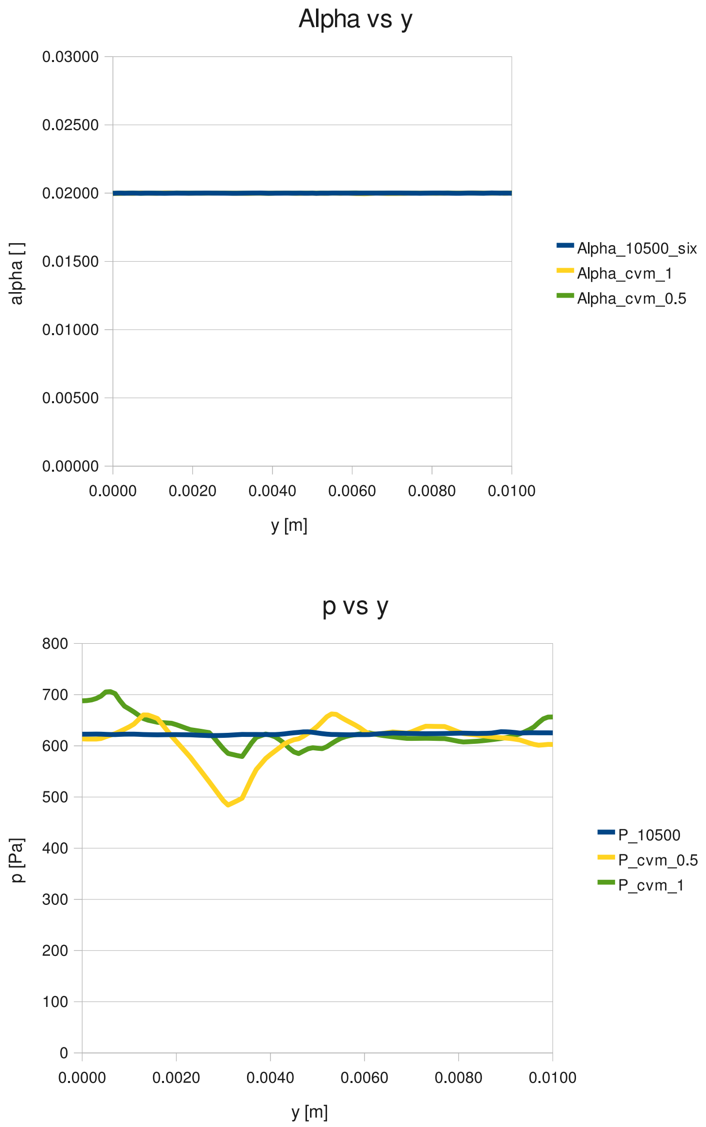


Figure 9.12: CVM SENSITIVITY AT POINT 2.

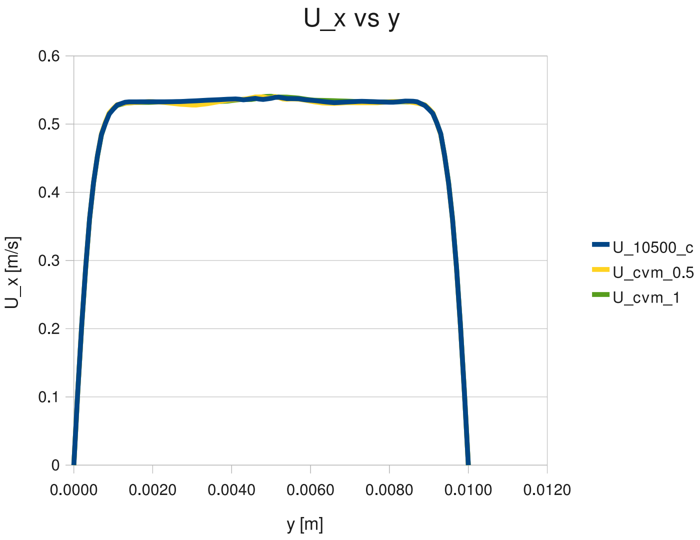


Figure 9.13: U_X AT POINT 2 FOR CVM SENSITIVITY.

Chapter 10

Booster simulation

After having presented and evaluated the simulation code $twoPhaseEulerFoam$, it will be applied to the experimental setup presented in part II. The main objective of the chapter will be to study the entrapment of the dispersed phase in the cavity next to the nozzle varying the Obstacle tip to nozzle tip parameter ($OT2NT$). $SixPhaseEulerFoam$ will not be applied because, as shown previosuly, is not numerical accurate.

10.1 Computational domain

10.1.1 Geometry

The geometry will be Von Karman Institute $L-11$ wind tunnel test section presented previously (Fig. 5.1). The parameter to be modified will be the throat opening (o) in order to obtain different $OT2NT$.

In part II , 18 different configurations were tested (table 5.2). The most representative ones related to $OT2NT$ have been chosen, covering from the maximum positive value of $OT2NT$ to the minimum negative one. They are presented in table 10.1.

Configuration	Measurement	h $[mm]$	L_i $[mm]$	w $[mm]$	o $[mm]$	$OT2NT$ $[mm]$
1	$1-2$	33.5	310	107.2	15.4	25.69
2	14	33.5	310	107.2	45.4	-4.31
3	15	33.5	310	107.2	67	-25.91

Table 10.1: GEOMETRY FOR NUMERICALLY SIMULATED CONFIGURATIONS.

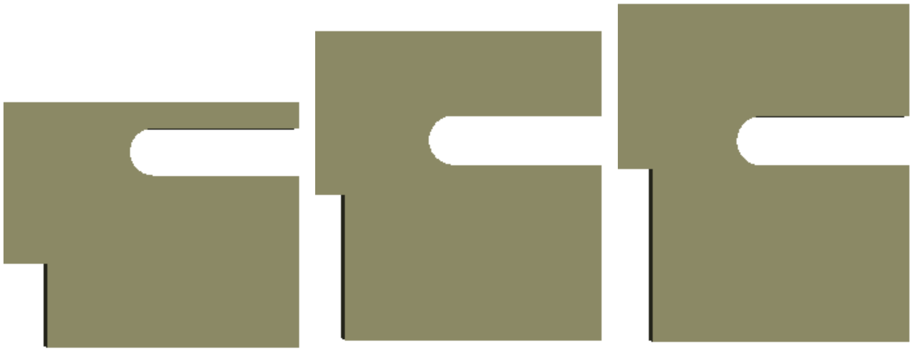


Figure 10.1: DETAIL OF THE NOZZLE AND CAVITY FOR CONFIGURATION 1, 2 AND 3 (LEFT TO RIGHT).

10.1.2 Boundary conditions

First, the different zones of the booster geometry will be presented in Fig. 10.2 .

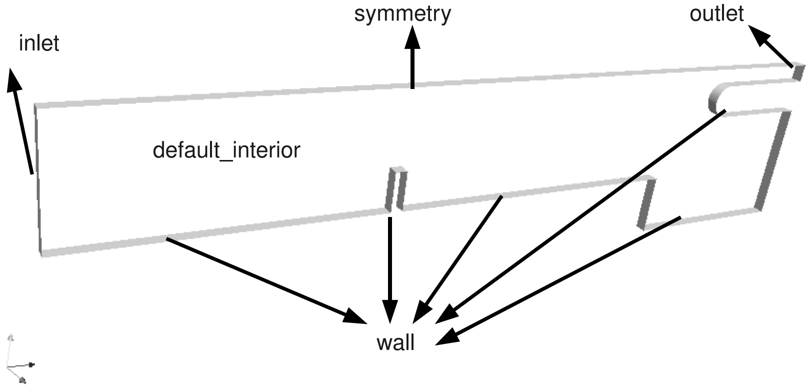


Figure 10.2: BOOSTER COMPUTATIONAL ZONES.

In table 10.2 the boundary conditions are presented. The volumetric fraction for the water have been obtained from the spray characterization done in [20].

Zone	p $[Pa]$	U_{air} $[m/s]$	U_{water} $[m/s]$	β_{air}	α_{water}	k	ε
inlet	$\nabla p = 0$	10	10	0.9985	0.00015	0	-
outlet	0	$\nabla U_{air} = 0$	$\nabla U_{water} = 0$	$\nabla \beta_{air} = 0$	$\nabla \alpha_{water} = 0$	0	-
symmetry	$-$	$-$	$-$	$-$	$-$	-	-
wall	$\nabla p = 0$	0	0	$\nabla \beta_{air} = 0$	$\nabla \alpha_{water} = 0$	0	-
default interior	0	0	0	1	0	0	-

Table 10.2: Boundary conditions for booster case.

The physical properties of the phases involved are shown in table 10.3 .

Physical Properties	Unit	Air	Water
density ρ	kg/m^3	1.20	1000
particle diameter	μm	100	-
Kinematic viscosity ν	$\frac{m}{s}^2$	$1.6 \cdot 10^{-5}$	$1 \cdot 10^{-6}$

Table 10.3: Physical Properties.

Considering the above properties, the Reynolds number based in the inhibitor height is 17500. So, the flow should be solved using turbulent models. The laminar model is as a qualitative approach to the solution. This procedure was also adoptated by Lema for Reynolds up to 15000 [28].

10.1.3 Mesh

Non orthogonal mesh sensitivity

The mesh is structured except in the zone in front of the nozzle rounded edge (Fig. 10.4). In this zone non orthogonal elements have to be used. This fact establishes a difference with the mesh used during the code evaluation. The concept of non orthogonality is related to the skewness that will be now introduced.

The skewness is the lost in accuracy due to the non orthogonality of neighbors cells. As shown in fig. 10.4 the value at the face is evaluated by interpolating the cell center values (P and N). Instead of obtaining the actual face centered value (f), a non centered value (f') is found. This creates an error (m). In order to preserve accuracy skewness should be maintained as low as possible. The skewness problem is widely treated by Jasak in [41]. The mathematical definition for skewness of a quad element applied in this report is presented in eq. 10.1.

$$skewness = max\left[\frac{\theta_{max} - 90}{90}, \frac{90 - \theta_{min}}{90},\right] \tag{10.1}$$

Where θ refer to the angle between element sides.

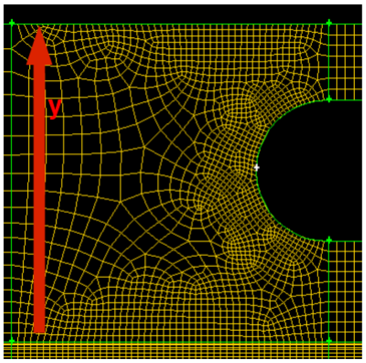


Figure 10.3: DETAILS OF NON ORTHOGONAL MESH ZONE.

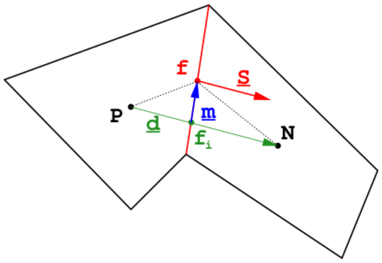


Figure 10.4: SKEWNESS DEFINITION. [41]

In order to evaluate the code sensitivity to skewness, 3 different meshes for the non orthogonal zone will be used, from coarser to finer (table 10.4). The geometrical configuration used will the the first one defined in table 10.1. The simulation time will be 0.2 seconds. Adding more elements to the mesh, increases the skewness because the elements will be more deformed. Therefore, for the considered geometry the skewness increases when increasing the number of elements.

Configuration	Non orthogonal elements	Skewness
Conf1_fine_1	930	0.5
Conf1_fine_2	1326	0.8
Conf1_fine_3	1645	1.0

Table 10.4: NON ORTHOGONAL MESH ELEMENTS SENSITIVITY.

There is not a closed criterion based in the skewness in order to distinguish “good” and “poor” meshes but usually, the skewness should be kept under 1 and preferably under 0.80. p, $alpha$, u_x and u_y will be monitored in the non orthogonal zone (Fig. 10.3) . In the following graphics y refers to the vertical direction (Fig. 10.3).

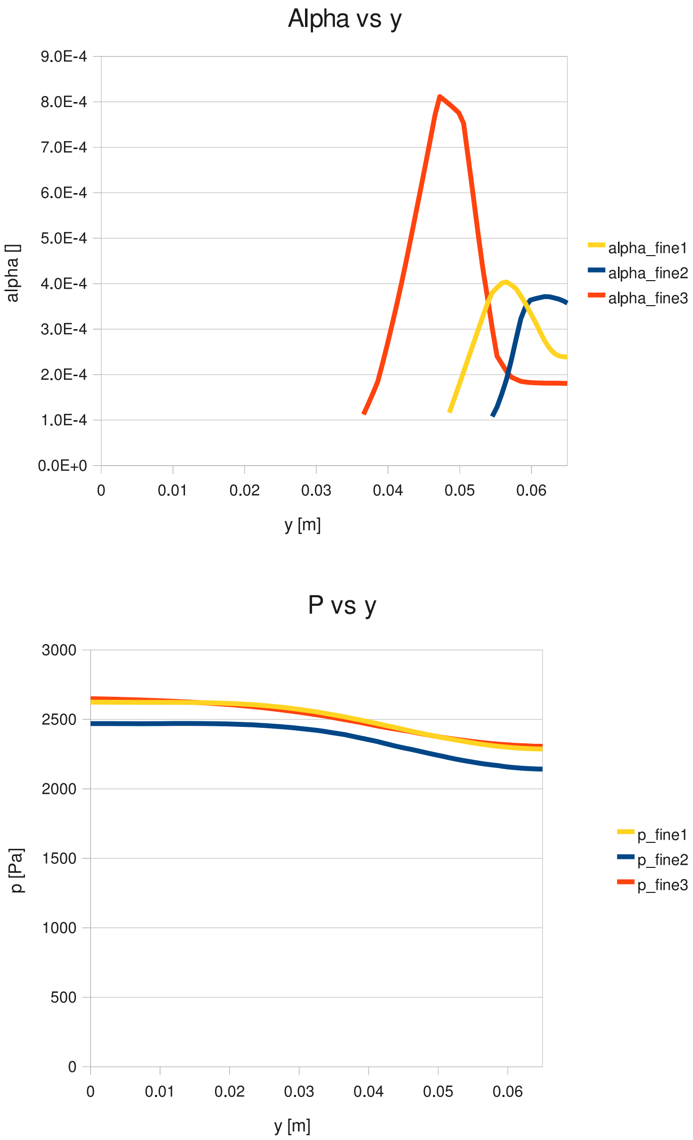


Figure 10.5: MESH SENSITIVITY FOR NON ORTHOGONAL ZONE .

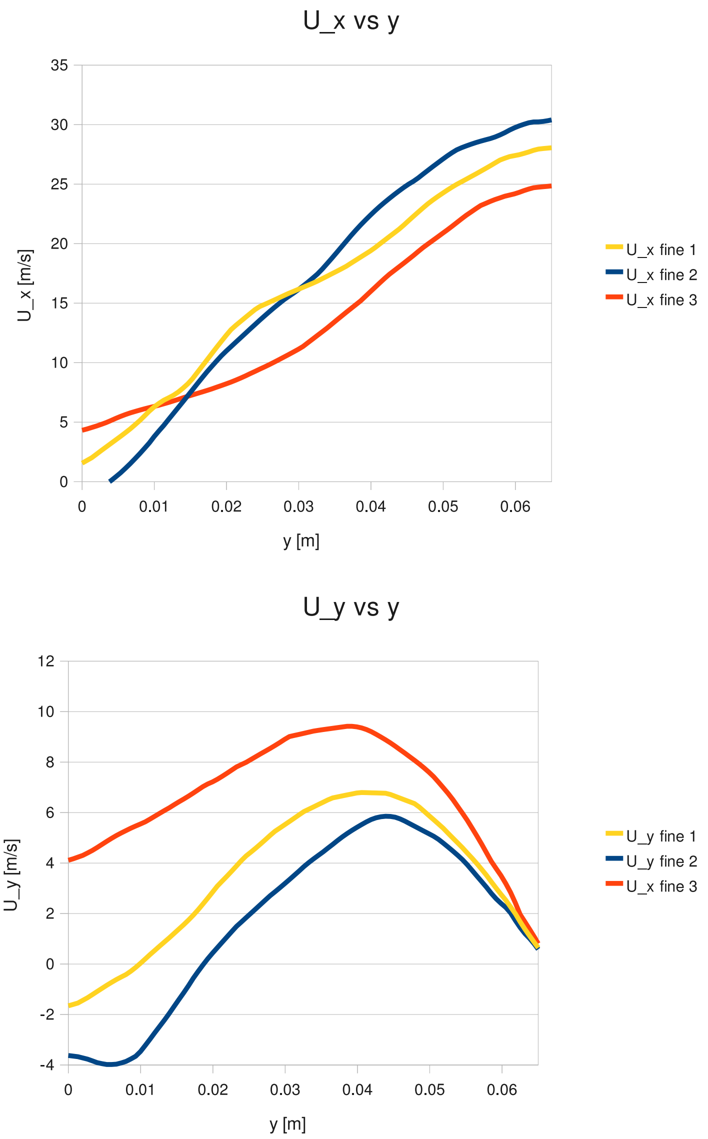


Figure 10.6: U_X AND U_Y FOR NON ORTHOGONAL ZONE IN MESH SENSITIVITY.

Considering the results shown in Fig. 10.5 and 10.6, it is found that for *alpha* and velocities the mesh with major skewness (fine_3) shows a different behavior than the rest of configurations. Instead for *p* there is no major difference.

Considering the convergence for $alpha$ and velocity and the lack of difference for p, the value of 0.8 for maximun skewness will be taken as the maximum allowed.

$twoPhaseFoam$ offers the possibility to apply non orthogonal correctors to the calculation [33]. In Fig. 10.7, the result for a number of non orthogonal correctors sensitivity is shown. The use of 3 corrector will be discarted because for alpha it is observed a non reasonable peak for lower values of y . For 1 corrector u_y shows discrepancy for lower values of y. There is no difference for all three p and u_x . Considering the lack of significant differences for 0 and 3 correctors, 0 correctors will be applied reducing the computational effort.

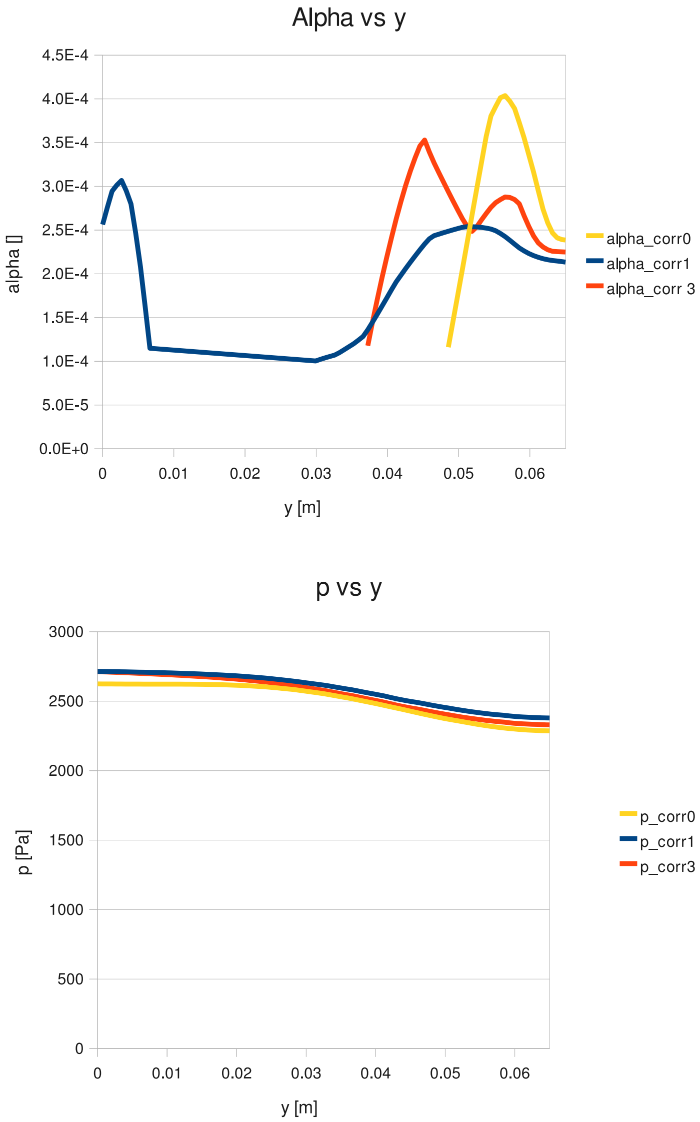
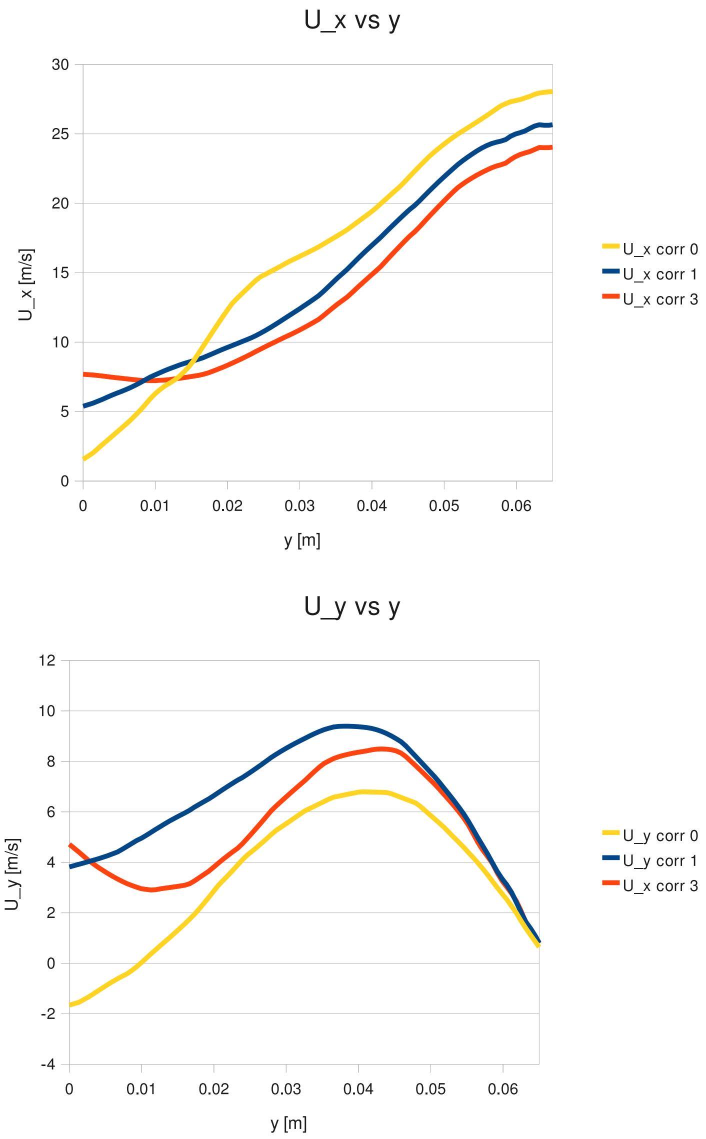


Figure 10.7: p and alpha for non orthogonal correctors sensitivity.


U_x vs y

U_y vs y

Figure 10.8: U_X AND U_Y FOR NON ORTHOGONAL CORRECTORS SENSITIVITY.

Once the mesh sensitivity have been studied both for skewness and non orthogonal correction, a mesh have been obtained for each of the geometrical configurations. Their characteristics are shown in table. 10.5 As a way of example, the mesh for the second configuration is shown in Fig. 10.9

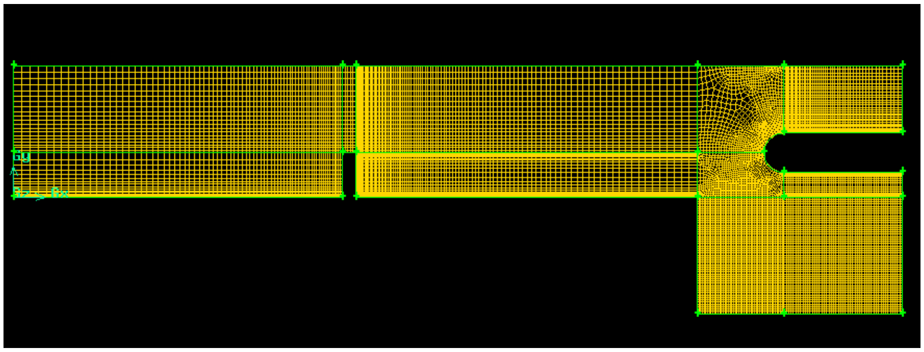


Figure 10.9: MESH FOR CONFIGURATION 2.

Configuration	Mesh elements	Max skewness
1	14401	0.8
2	16482	0.6
3	15134	0.8

Table 10.5: MESH CHARACTERISTICS.

The minimum lengths for the mesh elements are shown in table. 10.6 . These values should be larger than the water particle diameter ($100 \mu m$) . This condition is accomplished by all three configurations.

Configuration	Min. element length
1	0.00047
2	0.00047
3	0.00039

Table 10.6: MINIMUM MESH ELEMENT LENGTH.

10.1.4 Booster solution for $twoPhaseEulerFoam$

The objective of this section is to determine the amount of water droplets trapped in the cavity for the three different $0T2NT$ configurations presented previously. The amount of water droplets is measured by the entrapment. Experimentally, it was determined by the percentage of water found in the cavity minus the percentage of water due to dripping from the inhibitor tip. The percentage is calculated in base to the total injected water. Numerically, entrapment will be determined by integrating the quantity of water ($\rho_{water} \cdot \alpha_{water}$) found in the cavity divided by the amount of injected water (Fig. 10.10) after a 0.5 second simulation run. The time is large enough for the variables to reach steady state.

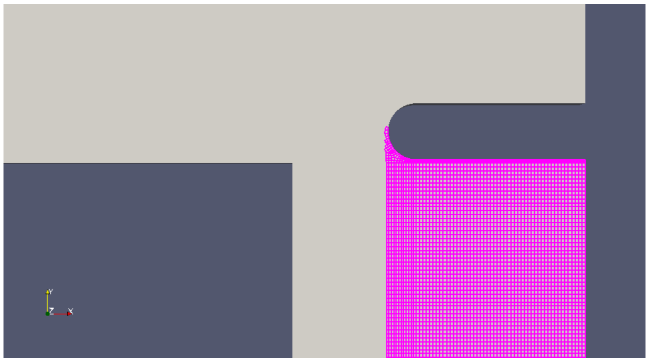


Figure 10.10: entrapment integration region.

The simulation results are shown in table. 10.11. There is a good agreement between experiments and numerical simulation for the first and the third configuration. The numerical trend found is the same than the one found experimentally. The configuration 1, where the nozzle is over the tip of the inhibitor shows the higher value of entrapment. The lower the inhibitor is situated below the inhibitor tip, the lower value of entrapment is obtained. It is worth to remind that during the static firing tests carried out by the ESA, the entrapment had an approximately value of 4%.

Configuration	$0T2NT$	Exp. Entrapmen [%]	Num.Entrapment [%]	Difference [%]
1	25.69	4.62	4.2	-9.1
2	-4.31	1.36	2.1	54.4
3	-23.91	1.09	1.1	0.9

Figure 10.11: twoPhaseEulerFoam booster results.

In the simulation no dripping from the inhibitor tip have been observed. So, the problem found experimentally due to the excess of dripping is not found numerically. This fact could be one of the reasons explaining the difference between computed and experimental results for the second configuration.

The second reason could be the turbulence model applied. The Reynolds number for this problem is 17500 but a laminar model have been applied. The development of a turbulent model, such as $k - \varepsilon$ may lead into a different computed quantity for the entrapment.

Chapter 11

Numerical vs Experimental Conclusions

In the numerical part, the application of an Eulerian model to a biphasic problem have been studied. The studied code was already implemented in the *OpenFoam* distribution and named as *twoPhaseEulerFoam*. Then this code have been developed to be able to handle up to 5 dispersed phases. The modified code has been named as *sixPhaseEulerFoam*

The first step has been to validate both codes and to study their sensitivity to the mesh, Courant number, lift coefficient and virtual mass coefficient. Both solver have been evaluated for laminar flows.

TwoPhaseEulerFoam has shown close results to the published ones taken as reference. The Courant number for the simulation have been a compromise between time calculation and stability and has been set to 0.3. The volumetric fraction is sensitive to the lift coefficient and the virtual mass coefficient while pressure and velocity are insensitive to these coefficients.

For *sixPhaseEulerFoam*, it has been found that the code is much sensitive to mesh quality. The convergence tests have shown that there was almost no difference between using a Co number of 0.3 and 0.5. When compared to published cases it has been found that the correct trend is found although the solver overstimes the solution, so the solver could only be used qualitatively. The solver shows the same sensitivity for lift and virtual coefficient although it can not handle extreme values for them.

TwopPhaseEulerFoam is applied to the slag accumulation problem studied in the experimental part. Previously it was found that the relevant parameter was the Obstacle tip to nozzle tip ($OT2NT$). The solver is applied to three $OT2NT$ values geometries: 25.69, -4.31 and -25.91 mm. The first thing to solve is the non orthogonality problems due the non structured mesh. It has been found that there is no need to apply non orthogonal correction and that the important fact was to keep the skewness bounded to 0.8.

It has been found the same trend than in the experiments: the positive values of $OT2NT$ (nozzle over the inhibitor tip) increase the entrapment while for negative values of $OT2NT$ the lower the nozzle is, the less entrapment is found.

There is a good agreement between experiments and numerical simulation for the first and the third configuration but there is a big discrepancy for the second one. This can be explained due to the lack of the application of a turbulence model, the mechanism to evaluate the dripping experimentally or the acoustic coupling.

The future developments for the numerical part would be the implementation and validation of a turbulence model, including wall functions, for $twoPhase-EulerFoam$ and $sixPhaseeulerFoam$ that does not exist nowadays. It would be also interesting to turn $sixPhaseEulerFoam$ from qualitative solver to an accurate one by solving its accuracy problems. Doing so, a mutidispersed spray could be applied to the experimental geometry.

Bibliography

[1] Duterque J., " Experimental study of Aluminum agglomeration in Solid Rocket Motors", International Journal of Energetic Materials and Chemical Propulsion , Vol 4 Issue $1-6$, pages $693-705$, 1997. In the Fourth International Symposium on special Topics in Chemical Propulsion , $27-31$ May 1996. Stockholm, Sweden.

[2] Cohen N.S, " A pocket model for Aluminum Agglomeration in Composite Propellants", AIAA Journal, Vol 21 $N^{\underline{o}}5$, 1983.

[3] Jayaraman K.Sambamurthi, Price Edward W. and Sigman Robert K., " Aluminum Agglomeration in Solid Propellant Combustion", ,AIAA Journal, Vol 22 $N^{\underline{o}}8$, p.1132-1138, 1984.

[4] Babuk V.A., Vasilyev V.A. and Sviridov V.V., " Propellant Formulation Factors and Metal Agglomeration in Combustion of Aluminized Solid Rocket Propellant", Combustion Science and Technology, Vol 163, p.$261-289$, 2001.

[5] Babuk V.A., Vasilyev and Malakhov M.S, " Condensed Combustion Products at the Burning Surface of Aluminized Solid Propellant", Journal of Propulsion and Power, Vol. 15 $N^{\underline{o}}6$, p. $783-793$, November-December 1999.

[6] Dokhan A., Price E.W. and Seitzman J.M., " The effects of Al particle size on the burning rate and residual oxide in aluminized propellants", AIAA Paper $2001-3581$. In 37^{th} AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit.

[7] Babuk V.A., Vassiliev V.A. and Naslednikov P.A., " Experimental study of evolution of Condensed Products in gas phase of burning Solid Rocket Propellant", International Journal of Energetic Materials and Chemical Propulsion, Issue $1-6$, p.$412-42$, 2002.

[8] Cesco N., L.Dumas, A.Hulin, T.Pevergne and Y.Fabignon. "Stochastic models to the investigation of slag accumulation in a large solid rocket motor" . AIAA Paper $1997-3118$. In 3^{rd} AIAA/ASME/SAE/ASEE Joint Propulsion conference & Exhibit, July $6-9$, 1997, Seattle (WA), USA.

[9] Orlandi, o and Y.Fabignon, " Numerical simulation of a single aluminium particle in a propellant gas". In 2^{nd} European Conference on Launcher Technology, $21-24$ November 2000. Rome, Italy.

[10] Lupoglazoff, N and F.Vuillot . " Numerical simulation of the unsteady ow inside Ariane 5 P230 SRM booster with burning aluminium particles". In 2^{nd} European Conference on Launcher Technology, 21 - 24 November 2000. Rome, Italy.

[11] Vardelle, M., C.Escure, P.Fauchais, B.Platet, and G.Lavergne, "Impact of alumina droplets on cold and hot surfaces. In 2nd European Conference on Launcher Technology" , $21-24$ November 2000. Rome, Italy.

[12] Zaleski S., and Gueyer D., "Simulation de l'impact de gouttes sur des lms liquides". In 3^{eme} Colloque RT a l'ENSMA, $25-27$ March 1998. Poiters, France.

[13] Pevergne T., and Le Halley P., "Simulation numerique des impacts de particules et du depot dans le $MPS230$". In 3^{eme} Colloque RT a l'ENSMA, $25-27$ March 1998. Poiters, France.

[14] Godfroy F. and Guery J.F., "Unsteady eulerian ow analysis of solid rocket motor slag", AIAA Paper $1997-2859$. In 33^{rd} AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, $6-9$ July. Seattle (WA), USA.

[15] Le Helley P., Coste S. and Pevergne T., "Influence of the unsteady ow on the alumina impigning the nozzle and the aft dome of the Ariane 5 boster". In 2^{nd} European Conference on Launcher Technology, $21-24$ November 2000. Rome, Italy.

[16] Dupays J., Fabignon Y., Orlandi O. and Trubert J.F., "Combustion of aluminum particles in solid rocket motors". In proceedings of the ODAS 2000, ONERA/DLR Aerospace symposium. Berlin, Germany.

[17] Wirzberger H., Macales Y. and S.Yaniv, "Prediction of slag formation in a solid rocket motor". AIAA Paper $2005-4496$. In 41^{st} AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, $10-13$ July, 2005. Tucson (AZ), USA.

[18] Najjar F.M., Haselbacher A., Balachandar S. and Moser R.D, "Simulations of droplet nozzle impact and slag accumulation in the RSRM". AIAA Paper $2006-4588$. In 42^{nd} $AIAA/ASME/SAE/ASEE$ Joint Propulsion Conference & Exhibit, $9-12$ July, 2006, Sacramento (CA), USA.

[19] Anthoine J., Buchlin J.M. and Hirschberg A., "Effect of nozzle cavity on ressonance in large SRM: theorical modeling". Journal of Propulsion and Power 18(2), p.$304-311$.

[20] Toth B., “Two phase flow investigation in a Cold gas Solid Rocket Motor Model through the Study of the Slag accumulation Process”. PhD Dissertation Von Karman Institute for Fluid Dynamics and Universite Libre de Bruxelles, 2008.

[21] European Space Agency, http://www.esa.int

[22] Salita M., “Deficiencies and requirements in modelling of slag generation in solid rocket motors”. Journal of propulsion and Power $N^{\underline{o}}$11, p $10 - 23$, 1995

[23] Sambamurthi J.K, Alvarado A and Mathias E.C, “ Correlation of Slag Expulsion with Ballistic Anomalies in shuttle Solid Rocket Motors”. Journal of Propulsion and Power, Vol. $N^{\underline{o}}$12, p. $625 - 631$, 1996.

[24] Schmeisser-Gailliegue, K. and Fabignon, Y., “Numerical simulations of vortex shedding phenomenon in a SRM using two approaches”. AIAA Paper $1996 - 3246$. In 43^{nd} AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit.

[25] Lupoglazoff, N. and Vuillot, F., “Numerical simulation of parietal vortex shedding phenomenon in a cold flow set up”. AIAA Paper $1998 - 3220$. In 34^{th} $AIAA/ASME/SAE/ASEE$ Joint Propulsion Conference & Exhibit, $13 - 15$ July, 1998, Cleveland (OH), USA.

[26] Anthoine, J., Buchlin, J-M. and Guéry, J-F., “ Effect of nozzle cavity on resonance in large SRM: Numerical Simulations”. Journal of propulsion and Power, Vol, $N^{\underline{o}}$19, $N^{\underline{o}}$3, p. $372 - 384$, 2003.

[27] DiBiase (2005) Short Training program report, “Experimental study of the internal flow of SRM”. $VKI\ SR$ 2005−39

[28] Lema. M, Numerical Study of the Entrapment and Accumulation of Droplets in Solid Rocket Motors (SRM). ESA Spanish Trainee Program 2275.

[29] Rusche H., “ Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions ” PhD thesis, Imperial College of Science, Technology & Medicine Department of Mechanical Engineering , 2002.

[30] Hill D.P., “ The Computer Simulation of dispersed Two-Phase Flows ” PhD thesis, Imperial College of Science, Technology & Medicine Department of Mechanical Engineering , 1998.

[31] Pope Stephen.B., “ Turbulent Flows”. Cambridge University Press

[33] Nabla Ltd., ”OpenFoam Programmers guide “, 2009.

[34] Weller H. G., “Derivation, modelling and solution of the conditionally averaged two-phase flow equations “ Technical Report, $TR/HGW/02$, Nabla Ltd., 2002.

[35] Politis S., “Prediction of Two-Phase Solid-Liquid Turbulent Flow Stirred Vessels” *PhD* thesis, Imperial College, University of London, 1989.

[38] Schiller L., Naumann A., “Uber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung”. Z. Vereins deutscher Ing., Vol $N^{\underline{o}}$77, p. 318–320, 1933.

[39] Yeoh G.H, Tu J., “Computational techniques for multi-phase flows”, $ELSEVIER$. First edition , 2010

[40] Silva L.F.R, Lage P.L.C, “Implementation of an Eulerian Multi-phase Model in OpenFoam and its application to Polydisperse Two-Phase Flows”

[41] Hasak, H. , “Error analysis and estimation for the finite volume method with application to fluid flows”. PhD Dissertation, Imperial College London

Appendix A. *SixPhaseEulerFoam* code

AlphaEqn.H

```
{   word scheme1("div(phi,alpha1)");
    word schemer1("div(phir1,alpha1)");
    word scheme2("div(phi,alpha2)");
    word schemer2("div(phir2,alpha2)");
    word scheme3("div(phi,alpha3)");
    word schemer3("div(phir3,alpha3)");
    word scheme4("div(phi,alpha4)");
    word schemer4("div(phir4,alpha4)");
    word scheme5("div(phi,alpha5)");
    word schemer5("div(phir5,alpha5)");

    surfaceScalarField phic = phi;
    surfaceScalarField phir1 = phia1 - phib;
    surfaceScalarField phir2 = phia2 - phib;
    surfaceScalarField phir3 = phia3 - phib;
    surfaceScalarField phir4 = phia4 - phib;
    surfaceScalarField phir5 = phia5 - phib;
    surfaceScalarField alphaf1 = fvc::interpolate(alpha1);
    surfaceScalarField alphaf2 = fvc::interpolate(alpha2);
    surfaceScalarField alphaf3 = fvc::interpolate(alpha3);
    surfaceScalarField alphaf4 = fvc::interpolate(alpha4);
    surfaceScalarField alphaf5 = fvc::interpolate(alpha5);
    surfaceScalarField betaf = scalar(1) - alphaf1 - alphaf2
    - alphaf3 - alphaf4 - alphaf5;

    for (int acorr=0; acorr<nAlphaCorr; acorr++)
    {
        fvScalarMatrix alphaEqn1
        (
             fvm::ddt(alpha1)
           + fvm::div(phic, alpha1, scheme1)
           + fvm::div(-phir1*betaf, alpha1, scheme1)
           + fvm::div(-(phia1-phia2)*alphaf2,alpha1, schemer1)
           + fvm::div(-(phia1-phia3)*alphaf3,alpha1, schemer1)
           + fvm::div(-(phia1-phia4)*alphaf4,alpha1, schemer1)
           + fvm::div(-(phia1-phia5)*alphaf5,alpha1, schemer1)
        );
        alphaEqn1.relax();
        alphaEqn1.solve();
        alpha1 = max(scalar(0.00000001), alpha1);

        fvScalarMatrix alphaEqn2
```

```
 (
      fvm::ddt(alpha2)
    + fvm::div(phic, alpha2, scheme2)
    + fvm::div(-phir2*betaf, alpha2, scheme2)
    + fvm::div(-(phia2-phia1)*alphaf1,alpha2, schemer2)
    + fvm::div(-(phia2-phia3)*alphaf3,alpha2, schemer2)
    + fvm::div(-(phia2-phia4)*alphaf4,alpha2, schemer2)
    + fvm::div(-(phia2-phia5)*alphaf5,alpha2, schemer2)
 );
alphaEqn2.relax();
alphaEqn2.solve();
alpha2 = max(scalar(0.00000001), alpha2);

 fvScalarMatrix alphaEqn3
 (
      fvm::ddt(alpha3)
    + fvm::div(phic, alpha3, scheme3)
    + fvm::div(-phir3*betaf, alpha3, scheme3)
    + fvm::div(-(phia3-phia1)*alphaf1,alpha3, schemer3)
    + fvm::div(-(phia3-phia2)*alphaf2,alpha3, schemer3)
    + fvm::div(-(phia3-phia4)*alphaf4,alpha3, schemer3)
    + fvm::div(-(phia3-phia5)*alphaf5,alpha3, schemer3)
 );
 alphaEqn3.relax();
 alphaEqn3.solve();
 alpha3 = max(scalar(0.00000001), alpha3);

 fvScalarMatrix alphaEqn4
 (
      fvm::ddt(alpha4)
    + fvm::div(phic, alpha4, scheme4)
    + fvm::div(-phir4*betaf, alpha4, scheme4)
    + fvm::div(-(phia4-phia1)*alphaf1,alpha4, schemer4)
    + fvm::div(-(phia4-phia2)*alphaf2,alpha4, schemer4)
    + fvm::div(-(phia4-phia3)*alphaf3,alpha4, schemer4)
    + fvm::div(-(phia4-phia5)*alphaf5,alpha4, schemer4)
 );
 alphaEqn4.relax();
 alphaEqn4.solve();
 alpha4 = max(scalar(0.00000001), alpha4);

 fvScalarMatrix alphaEqn5
 (
      fvm::ddt(alpha5)
    + fvm::div(phic, alpha5, scheme5)
    + fvm::div(-phir5*betaf, alpha5, scheme5)
    + fvm::div(-(phia5-phia1)*alphaf1,alpha5, schemer5)
    + fvm::div(-(phia5-phia2)*alphaf2,alpha5, schemer5)
    + fvm::div(-(phia5-phia3)*alphaf3,alpha5, schemer5)
    + fvm::div(-(phia5-phia4)*alphaf4,alpha5, schemer5)
);
alphaEqn5.relax();
alphaEqn5.solve();
alpha5 = max(scalar(0.00000001), alpha5);

beta = scalar(1) - alpha1 - alpha2 - alpha3 - alpha4 - alpha5;

Info<< "Dispersed phase volume fraction for dispersed phase 1= "
    << alpha1.weightedAverage(mesh.V()).value()
    << "  Min(alpha1) = " << min(alpha1).value()
    << "  Max(alpha1) = " << max(alpha1).value()
    << endl;

Info<< "Dispersed phase volume fraction for dispersed phase 2= "
    << alpha2.weightedAverage(mesh.V()).value()
    << "  Min(alpha2) = " << min(alpha2).value()
    << "  Max(alpha2) = " << max(alpha2).value()
    << endl;

Info<< "Dispersed phase volume fraction for dispersed phase 3= "
    << alpha3.weightedAverage(mesh.V()).value()
    << "  Min(alpha3) = " << min(alpha3).value()
    << "  Max(alpha3) = " << max(alpha3).value()
    << endl;

Info<< "Dispersed phase volume fraction for dispersed phase 4= "
    << alpha4.weightedAverage(mesh.V()).value()
    << "  Min(alpha4) = " << min(alpha4).value()
    << "  Max(alpha4) = " << max(alpha4).value()
    << endl;

Info<< "Dispersed phase volume fraction for dispersed phase 5 = "
    << alpha5.weightedAverage(mesh.V()).value()
    << "  Min(alpha5) = " << min(alpha5).value()
    << "  Max(alpha5) = " << max(alpha5).value()
    << endl;

Info<< "Dispersed phase volume fraction for beta = "
    << beta.weightedAverage(mesh.V()).value()
    << "  Min(beta) = " << min(beta).value()
    << "  Max(beta) = " << max(beta).value()
    << endl;
    }
}

rho = alpha1*rhoa1 +  alpha2*rhoa2 + alpha3*rhoa3
+ alpha4*rhoa4 + alpha5*rhoa5 + beta*rhob;
```

pEqn.H

```
{
    surfaceScalarField alphaf1 = fvc::interpolate(alpha1);
    surfaceScalarField alphaf2 = fvc::interpolate(alpha2);
    surfaceScalarField alphaf3 = fvc::interpolate(alpha3);
    surfaceScalarField alphaf4 = fvc::interpolate(alpha4);
    surfaceScalarField alphaf5 = fvc::interpolate(alpha5);
    surfaceScalarField betaf = scalar(1) - alphaf1 -
    alphaf2 - alphaf3 - alphaf4 - alphaf5;

    volScalarField rUa1A = 1.0/UaEqn1.A();
    volScalarField rUa2A = 1.0/UaEqn2.A();
    volScalarField rUa3A = 1.0/UaEqn3.A();
    volScalarField rUa4A = 1.0/UaEqn4.A();
    volScalarField rUa5A = 1.0/UaEqn5.A();
    volScalarField rUbA = 1.0/UbEqn.A();

    rUa1Af = fvc::interpolate(rUa1A);
    rUa2Af = fvc::interpolate(rUa2A);
    rUa3Af = fvc::interpolate(rUa3A);
    rUa4Af = fvc::interpolate(rUa4A);
    rUa5Af = fvc::interpolate(rUa5A);

    surfaceScalarField rUbA1f = fvc::interpolate(rUbA);
    Ua1 = rUa1A*UaEqn1.H();
    Ua2 = rUa2A*UaEqn2.H();
    Ua3 = rUa3A*UaEqn3.H();
    Ua4 = rUa4A*UaEqn4.H();
    Ua5 = rUa5A*UaEqn5.H();
    Ub = rUbA*UbEqn.H();

    surfaceScalarField phiDraga1 =
        fvc::interpolate(beta*Ka1*rUa1A/rhoa1)*phib
                + rUa1Af*(g & mesh.Sf());
    surfaceScalarField phiDraga2 =
        fvc::interpolate(beta*Ka2*rUa2A/rhoa2)*phib
                + rUa2Af*(g & mesh.Sf());
    surfaceScalarField phiDraga3 =
        fvc::interpolate(beta*Ka3*rUa3A/rhoa3)*phib
                + rUa3Af*(g & mesh.Sf());
    surfaceScalarField phiDraga4 =
        fvc::interpolate(beta*Ka4*rUa4A/rhoa4)*phib
                + rUa4Af*(g & mesh.Sf());
    surfaceScalarField phiDraga5 =
        fvc::interpolate(beta*Ka5*rUa5A/rhoa5)*phib
                + rUa5Af*(g & mesh.Sf());

    surfaceScalarField phiDragb =
        fvc::interpolate(alpha1*Ka1*rUbA/(rhob))*phia1 +
        fvc::interpolate(alpha2*Ka2*rUbA/(rhob))*phia2 +
        fvc::interpolate(alpha3*Ka3*rUbA/(rhob))*phia3 +
        fvc::interpolate(alpha4*Ka4*rUbA/(rhob))*phia4 +
        fvc::interpolate(alpha5*Ka5*rUbA/(rhob))*phia5 +
                rUbAf*(g& mesh.Sf());

    // Fix for gravity on outlet boundary.
    forAll(p.boundaryField(), patchi)
    {
if
(isA<zeroGradientFvPatchScalarField>(p.boundaryField()[patchi]))
        {
            phiDraga1.boundaryField()[patchi] = 0.0;
            phiDraga2.boundaryField()[patchi] = 0.0;
            phiDraga3.boundaryField()[patchi] = 0.0;
            phiDraga4.boundaryField()[patchi] = 0.0;
            phiDraga5.boundaryField()[patchi] = 0.0;
            phiDragb.boundaryField()[patchi] = 0.0;
        }
    }

    phia1 = (fvc::interpolate(Ua1) & mesh.Sf()) +
        fvc::ddtPhiCorr(rUa1A, Ua1, phia1)          +
        phiDraga1;
    phia2 = (fvc::interpolate(Ua2) & mesh.Sf()) +
        fvc::ddtPhiCorr(rUa2A, Ua2, phia2)
    + phiDraga2;
    phia3 = (fvc::interpolate(Ua3) & mesh.Sf()) +
        fvc::ddtPhiCorr(rUa3A, Ua3, phia3)
    + phiDraga3;
    phia4 = (fvc::interpolate(Ua4) & mesh.Sf()) +
        fvc::ddtPhiCorr(rUa4A, Ua4, phia4)
    + phiDraga4;
    phia5 = (fvc::interpolate(Ua5) & mesh.Sf()) +
        fvc::ddtPhiCorr(rUa5A, Ua5, phia5)
    + phiDraga5;
    phib = (fvc::interpolate(Ub) & mesh.Sf()) +
        fvc::ddtPhiCorr(rUbA, Ub, phib)
    + phiDragb;

        phi = alphaf1*phia1 + alphaf2*phia2 + alphaf3*phia3
    + alphaf4*phia4 + alphaf5*phia5 + betaf*phib;

surfaceScalarField Dp("(rho*(1|A(U)))", alphaf1*rUa1Af/rhoa1 +
    alphaf2*rUa2Af/rhoa2
    + alphaf3*rUa3Af/rhoa3
    + alphaf4*rUa4Af/rhoa4
    + alphaf5*rUa5Af/rhoa5
    +betaf*rUbAf/rhob);

for(int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
{
    fvScalarMatrix pEqn
    (
        fvm::laplacian(Dp, p) == fvc::div(phi)
    );

    pEqn.setReference(pRefCell, pRefValue);
    pEqn.solve();
    if (nonOrth == nNonOrthCorr)
    {
        surfaceScalarField SfGradp = pEqn.flux()/Dp;
        phia1 -= rUa1Af*SfGradp/rhoa1;
        phia2 -= rUa2Af*SfGradp/rhoa2;
        phia3 -= rUa3Af*SfGradp/rhoa3;
        phia4 -= rUa4Af*SfGradp/rhoa4;
        phia5 -= rUa5Af*SfGradp/rhoa5;
        phib -= rUbAf*SfGradp/rhob;
        phi = alphaf1*phia1 + alphaf2*phia2 + alphaf3*phia3 +
        alphaf4*phia4 +       alphaf5*phia5 + betaf*phib;

        p.relax();
        SfGradp = pEqn.flux()/Dp;
        Ua1 += fvc::reconstruct(phiDraga1 - rUa1Af*SfGradp/rhoa1);
        Ua1.correctBoundaryConditions();
        Ua2 += fvc::reconstruct(phiDraga2 - rUa2Af*SfGradp/rhoa2);
        Ua2.correctBoundaryConditions();
        Ua3 += fvc::reconstruct(phiDraga3 - rUa3Af*SfGradp/rhoa3);
        Ua3.correctBoundaryConditions();
        Ua4 += fvc::reconstruct(phiDraga4 - rUa4Af*SfGradp/rhoa4);
        Ua4.correctBoundaryConditions();
        Ua5 += fvc::reconstruct(phiDraga5 - rUa5Af*SfGradp/rhoa5);
        Ua5.correctBoundaryConditions();
        Ub += fvc::reconstruct(phiDragb - rUbAf*SfGradp/rhob);
        Ub.correctBoundaryConditions();
        U = alpha1*Ua1 + alpha2*Ua2 + alpha3*Ua3 + alpha4*Ua4
        + alpha5*Ua5 + beta*Ub;
```

```
            }
      }
}
#include "continuityErrs.H"
```

UEqns.H

```
fvVectorMatrix UaEqn1(Ua1, Ua1.dimensions()*dimVol/dimTime);
fvVectorMatrix UaEqn2(Ua2, Ua2.dimensions()*dimVol/dimTime);
fvVectorMatrix UaEqn3(Ua3, Ua3.dimensions()*dimVol/dimTime);
fvVectorMatrix UaEqn4(Ua4, Ua4.dimensions()*dimVol/dimTime);
fvVectorMatrix UaEqn5(Ua5, Ua5.dimensions()*dimVol/dimTime);
fvVectorMatrix UbEqn(Ub, Ub.dimensions()*dimVol/dimTime);

{
        {
        volTensorField gradUa1T = fvc::grad(Ua1)().T();
        volTensorField Rca1
        (
            "Rca1",
            ((2.0/3.0)*I)*(sqr(Ct)*k + nuEffa1*tr(gradUa1T))
                        - nuEffa1*gradUa1T
       );

        surfaceScalarField phiRa1 =
            -fvc::interpolate(nuEffa1)*mesh.magSf()*fvc::snGrad(alpha1)
            /fvc::interpolate(alpha1 + scalar(0.001));

        UaEqn1 =
        (
            (scalar(1) + Cvm*rhob*beta/rhoa1)*
            (
                fvm::ddt(Ua1)
              + fvm::div(phia1, Ua1, "div(phia1,Ua1)")
              - fvm::Sp(fvc::div(phia1), Ua1)
            )
          - fvm::laplacian(nuEffa1, Ua1)
          + fvc::div(Rca1)
          + fvm::div(phiRa1, Ua1, "div(phia1,Ua1)")
          - fvm::Sp(fvc::div(phiRa1), Ua1)
          + (fvc::grad(alpha1)/(fvc::average(alpha1)
                  + scalar(0.001)) & Rca1)
         ==
        //   g
                // Buoyancy term transfered to p-equation
          - fvm::Sp(beta/rhoa1*Ka1, Ua1)
        //+ beta/rhoa*K*Ub
        // Explicit drag transfered to p-equation
          - beta/rhoa1*(liftCoeff1 - Cvm*rhob*DDtUb)
        );
        UaEqn1.relax();
```

```
}
    {
    volTensorField gradUa2T = fvc::grad(Ua2)().T();
    volTensorField Rca2
    (
        "Rca2",
        ((2.0/3.0)*I)*(sqr(Ct)*k + nuEffa2*tr(gradUa2T))
                    - nuEffa2*gradUa2T
    );

    surfaceScalarField phiRa2 =
        -fvc::interpolate(nuEffa2)*mesh.magSf()*fvc::snGrad(alpha2)
        /fvc::interpolate(alpha2 + scalar(0.001));

    UaEqn2 =
    (
        (scalar(1) + Cvm*rhob*beta/rhoa2)*
        (
            fvm::ddt(Ua2)
          + fvm::div(phia2, Ua2, "div(phia2,Ua2)")
          - fvm::Sp(fvc::div(phia2), Ua2)
        )
      - fvm::laplacian(nuEffa2, Ua2)
      + fvc::div(Rca2)
      + fvm::div(phiRa2, Ua2, "div(phia2,Ua2)")
      - fvm::Sp(fvc::div(phiRa2), Ua2)
      + (fvc::grad(alpha2)/(fvc::average(alpha2)
              + scalar(0.001)) & Rca2)
     ==
    //  g
            // Buoyancy term transfered to p-equation
      - fvm::Sp(beta/rhoa2*Ka2, Ua2)
    //+ beta/rhoa*K*Ub
    // Explicit drag transfered to p-equation
      - beta/rhoa2*(liftCoeff2 - Cvm*rhob*DDtUb)
    );
    UaEqn2.relax();
}
    {
    volTensorField gradUa3T = fvc::grad(Ua3)().T();
    volTensorField Rca3
    (
        "Rca3",
        ((2.0/3.0)*I)*(sqr(Ct)*k + nuEffa3*tr(gradUa3T))
                    - nuEffa3*gradUa3T
    );
```

```
        surfaceScalarField phiRa3 =
            -fvc::interpolate(nuEffa3)*mesh.magSf()*fvc::snGrad(alpha3)
            /fvc::interpolate(alpha3 + scalar(0.001));

        UaEqn3 =
        (
            (scalar(1) + Cvm*rhob*beta/rhoa3)*
            (
                fvm::ddt(Ua3)
              + fvm::div(phia3, Ua3, "div(phia3,Ua3)")
              - fvm::Sp(fvc::div(phia3), Ua3)
            )
          - fvm::laplacian(nuEffa3, Ua3)
          + fvc::div(Rca3)
          + fvm::div(phiRa3, Ua3, "div(phia3,Ua3)")
          - fvm::Sp(fvc::div(phiRa3), Ua3)
          + (fvc::grad(alpha3)/(fvc::average(alpha3)
                  + scalar(0.001)) & Rca3)
         ==
        //   g
                // Buoyancy term transfered to p-equation
         - fvm::Sp(beta/rhoa3*Ka3, Ua3)
        //+ beta/rhoa*K*Ub
            // Explicit drag transfered to p-equation
          - beta/rhoa3*(liftCoeff3 - Cvm*rhob*DDtUb)
        );
        UaEqn3.relax();
    }
        {
        volTensorField gradUa4T = fvc::grad(Ua4)().T();
        volTensorField Rca4
        (
            "Rca4",
            ((2.0/3.0)*I)*(sqr(Ct)*k + nuEffa4*tr(gradUa4T))
                        - nuEffa4*gradUa4T
        );

        surfaceScalarField phiRa4 =
            -fvc::interpolate(nuEffa4)*mesh.magSf()*fvc::snGrad(alpha4)
            /fvc::interpolate(alpha4 + scalar(0.001));

        UaEqn4 =
        (
            (scalar(1) + Cvm*rhob*beta/rhoa4)*
            (
                fvm::ddt(Ua4)
          + fvm::div(phia3, Ua4, "div(phia4,Ua4)")
          - fvm::Sp(fvc::div(phia4), Ua4)
        )
      - fvm::laplacian(nuEffa4, Ua4)
      + fvc::div(Rca4)
      + fvm::div(phiRa4, Ua4, "div(phia4,Ua4)")
      - fvm::Sp(fvc::div(phiRa4), Ua4)
      + (fvc::grad(alpha4)/(fvc::average(alpha4)
              + scalar(0.001)) & Rca4)
     ==
    //  g
    // Buoyancy term transfered to p-equation
      - fvm::Sp(beta/rhoa4*Ka4, Ua4)
    //+ beta/rhoa*K*Ub
        // Explicit drag transfered to p-equation
      - beta/rhoa4*(liftCoeff4 - Cvm*rhob*DDtUb)
    );
    UaEqn4.relax();
}
    {
    volTensorField gradUa5T = fvc::grad(Ua5)().T();
    volTensorField Rca5
    (
        "Rca5",
        ((2.0/3.0)*I)*(sqr(Ct)*k + nuEffa5*tr(gradUa5T))
                    - nuEffa5*gradUa5T
    );
    surfaceScalarField phiRa5 =
        -fvc::interpolate(nuEffa5)*mesh.magSf()*fvc::snGrad(alpha5)
        /fvc::interpolate(alpha5 + scalar(0.001));
    UaEqn5 =
    (
        (scalar(1) + Cvm*rhob*beta/rhoa5)*
        (
            fvm::ddt(Ua5)
          + fvm::div(phia5, Ua5, "div(phia5,Ua5)")
          - fvm::Sp(fvc::div(phia5), Ua5)
        )
      - fvm::laplacian(nuEffa5, Ua5)
      + fvc::div(Rca5)

      + fvm::div(phiRa5, Ua5, "div(phia5,Ua5)")
      - fvm::Sp(fvc::div(phiRa5), Ua5)
      + (fvc::grad(alpha5)/(fvc::average(alpha5)
              + scalar(0.001)) & Rca5)
     ==
        //  g
        // Buoyancy term transfered to p-equation
          - fvm::Sp(beta/rhoa5*Ka5, Ua5)
        //+ beta/rhoa*K*Ub
            // Explicit drag transfered to p-equation
          - beta/rhoa5*(liftCoeff5 - Cvm*rhob*DDtUb)
        );
        UaEqn5.relax();
    }

    {
        volTensorField gradUbT = fvc::grad(Ub)().T();
        volTensorField Rcb
        (
           "Rcb",
            ((2.0/3.0)*I)*(k + nuEffb*tr(gradUbT))
                        - nuEffb*gradUbT
        );
        surfaceScalarField phiRb =
            -fvc::interpolate(nuEffb)*mesh.magSf()*fvc::snGrad(beta)
          /fvc::interpolate(beta + scalar(0.001));

        UbEqn =
         (
            (scalar(1) + Cvm*(alpha1 + alpha2 + alpha3 + alpha4 + alpha5))*
            (
                fvm::ddt(Ub)
              + fvm::div(phib, Ub, "div(phib,Ub)")
              - fvm::Sp(fvc::div(phib), Ub)
            )
          - fvm::laplacian(nuEffb, Ub)
          + fvc::div(Rcb)
          + fvm::div(phiRb, Ub, "div(phib,Ub)")
          - fvm::Sp(fvc::div(phiRb), Ub)
          + (fvc::grad(beta)/(fvc::average(beta)
                + scalar(0.001)) & Rcb)
         ==
        //  g
                         // Buoyancy term transfered to p-equation
          - fvm::Sp(alpha1/rhob*Ka1, Ub)

          - fvm::Sp(alpha2/rhob*Ka2, Ub)
          - fvm::Sp(alpha3/rhob*Ka3, Ub)
          - fvm::Sp(alpha4/rhob*Ka4, Ub)
          - fvm::Sp(alpha5/rhob*Ka5, Ub)
        //+ alpha/rhob*K*Ua
```

```
            // Explicit drag transfered to p-equation
          + alpha1/rhob*(liftCoeff1 + Cvm*rhob*DDtUa1)
          + alpha2/rhob*(liftCoeff2 + Cvm*rhob*DDtUa2)
          + alpha3/rhob*(liftCoeff3 + Cvm*rhob*DDtUa3)
          + alpha4/rhob*(liftCoeff4 + Cvm*rhob*DDtUa4)
          + alpha5/rhob*(liftCoeff5 + Cvm*rhob*DDtUa5)
        );
        UbEqn.relax();
    }
}
```

Courantnos.H

```
#   include "CourantNo.H"
{
    scalar Ur1CoNum = max
    (
        mesh.surfaceInterpolation::deltaCoeffs()*mag(phia1 - phib)
      /mesh.magSf()
    ).value()*runTime.deltaT().value();

    Info<< "Max Ur Courant Number phase 1 = " << Ur1CoNum << endl;

    scalar Ur2CoNum = max
    (
        mesh.surfaceInterpolation::deltaCoeffs()*mag(phia2 - phib)
      /mesh.magSf()
    ).value()*runTime.deltaT().value();

    Info<< "Max Ur Courant Number phase 2 = " << Ur2CoNum << endl;

    scalar Ur3CoNum = max
    (
        mesh.surfaceInterpolation::deltaCoeffs()*mag(phia3 - phib)
      /mesh.magSf()
    ).value()*runTime.deltaT().value();

    Info<< "Max Ur Courant Number phase 3 = " << Ur3CoNum << endl;

    scalar Ur4CoNum = max
    (
        mesh.surfaceInterpolation::deltaCoeffs()*mag(phia4 - phib)
      /mesh.magSf()
    ).value()*runTime.deltaT().value();

    Info<< "Max Ur Courant Number phase 4 = " << Ur4CoNum << endl;

    scalar Ur5CoNum = max
    (
        mesh.surfaceInterpolation::deltaCoeffs()*mag(phia5 - phib)
      /mesh.magSf()       ).value()*runTime.deltaT().value();

    Info<< "Max Ur Courant Number phase 5 = " << Ur5CoNum << endl;

CoNum = max(CoNum, Ur1CoNum);
CoNum = max(CoNum, Ur2CoNum);
CoNum = max(CoNum, Ur3CoNum);
CoNum = max(CoNum, Ur4CoNum);
CoNum = max(CoNum, Ur5CoNum);
}
```

ke.H

```
if(turbulence)
{
    if (mesh.changing())
    {
        y.correct();
    }

    tmp<volTensorField> tgradUb = fvc::grad(Ub);
    volScalarField G = 2*nutb*(tgradUb() && dev(symm(tgradUb())));
    tgradUb.clear();

    #include "wallFunctions.H"

    // Dissipation equation
    fvScalarMatrix epsEqn
    (
        fvm::ddt(beta, epsilon)
      + fvm::div(phib, epsilon)
      - fvm::laplacian
        (
            alphaEps*nuEffb, epsilon,
            "laplacian(DepsilonEff,epsilon)"
        )
      ==
         C1*beta*G*epsilon/k
       - fvm::Sp(C2*beta*epsilon/k, epsilon)
    );

    #include "wallDissipation.H"

    epsEqn.relax();
    epsEqn.solve();

    epsilon.max(dimensionedScalar("zero", epsilon.dimensions(), 1.0e-15));


    // Turbulent kinetic energy equation
    fvScalarMatrix kEqn
    (
        fvm::ddt(beta, k)
      + fvm::div(phib, k)
      - fvm::laplacian
        (
            alphak*nuEffb, k,
            "laplacian(DkEff,k)"
        )
      ==
        beta*G
      - fvm::Sp(beta*epsilon/k, k)
    );
    kEqn.relax();
    kEqn.solve();

    k.max(dimensionedScalar("zero", k.dimensions(), 1.0e-8));

    //- Re-calculate turbulence viscosity
    nutb = Cmu*sqr(k)/epsilon;

    #include "wallViscosity.H"
}

nuEffa1 = sqr(Ct)*nutb + nua1;
nuEffa2 = sqr(Ct)*nutb + nua2;
nuEffa3 = sqr(Ct)*nutb + nua3;
nuEffa4 = sqr(Ct)*nutb + nua4;
nuEffa5 = sqr(Ct)*nutb + nua5;

nuEffb = nutb + nub;
```

sixPhaseEulerFoam.C

```
Application
    sixPhaseEulerFoam

Description
    Solver for a system of 6 incompressible fluid phases with five phases
    dispersed

\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "nearWallDist.H"
#include "wallFvPatch.H"
#include "Switch.H"

#include "IFstream.H"
#include "OFstream.H"

#include "dragModel.H"
#include "phaseModel.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "readGravitationalAcceleration.H"
    #include "createFields.H"
    #include "initContinuityErrs.H"
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    Info<< "\nStarting time loop\n" << endl;

    while (runTime.run())
    {
        #include "readsixPhaseEulerFoamControls.H"
        #include "CourantNos.H"
        #include "setDeltaT.H"
        runTime++;
        Info<< "Time = " << runTime.timeName() << nl << endl;

        #include "alphaEqn.H"

        #include "liftDragCoeffs.H"

        #include "UEqns.H"

        // --- PISO loop
        for (int corr=0; corr<nCorr; corr++)
        {
            #include "pEqn.H"

            if (correctAlpha && corr<nCorr-1)
            {
                #include "alphaEqn.H"
            }
        }

        #include "DDtU.H"

        #include "ke.H"

        #include "write.H"

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }

    Info<< "End\n" << endl;

    return 0;
}
```