Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Μηχανολόγων Μηχανικών

Εργαστήριο Ατμοκινητήρων και Λεβήτων

Διπλωματική Εργασία με θέμα:

Προσομοιώσεις CFD σε καταλυτικούς αντιδραστήρες σταθερής κλίνης

Κουτσούμπα Ευθυμία-Ιωάννα

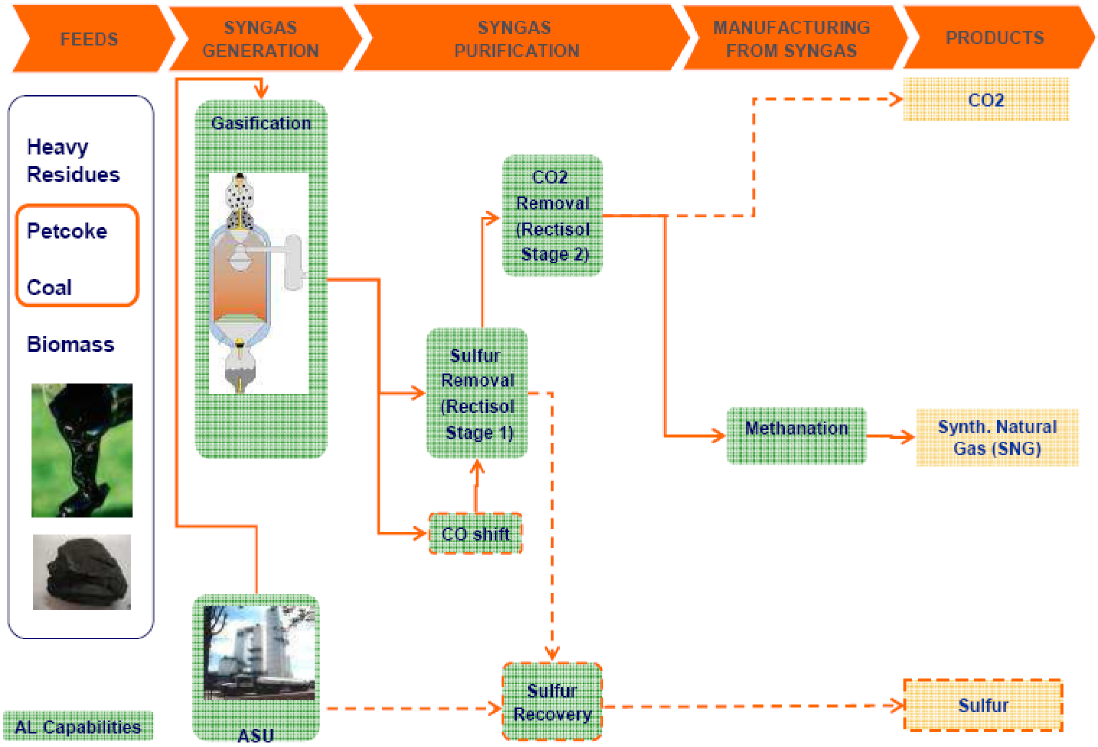
Επιβλέπων Καθηγητής: Δρ. Κακαράς Εμμανουήλ

Αθήνα 2010

ΠΕΡΙΛΗΨΗ

Το φαινόμενο της αύξησης της θερμοκρασίας και η μόλυνση του περιβάλλοντος λόγω της εκπομπής αερίων ρύπων καθιστούν επιτακτική την ανάγκη αποδοτικών και φιλικών προς το περιβάλλον τεχνολογιών. Οι αυξανόμενες ενεργειακές ανάγκες, τα υψηλά κόστη παραγωγής ενέργειας και οι περιορισμένες πηγές ενέργειας έχουν οδηγήσει την αγορά σε εναλλακτικές μορφές ενέργειας. Η μετατροπή των στερεών καυσίμων, όπως ο άνθρακας και ο λιγνίτης καθώς επίσης και βιομάζα, σε υποκατάστατο φυσικό αέριο (to Substitute Natural Gas (SNG)) μέσω της διαδικασίας της αεριοποίησης και μεθανοποίησης αποτελεί μια πολλά υποσχόμενη τεχνολογία.

Τα βασικά βήματα παραγωγής SNG είναι η διαδικασία της αεριοποίησης του στερεού καυσίμου, η διαδικασία καθαρισμού του παραγόμενου αερίου και η μεθανοποίηση. Αναλυτικότερα, το καύσιμο εισέρχεται στον αεριοποιητή (gasifier) όπου και αντιδρά (μερική οξείδωση) παράγοντας ακατέργαστο αέριο υψηλής θερμοκρασίας (syngas). Στη συνέχεια μεταφέρεται για τον καθαρισμό του από στερεά κατάλοιπα και άλλες ανεπιθύμητες προσμίξεις και καταλήγει σε ένα καταλυτικό αντιδραστήρα, όπου πραγματοποιείται η αντίδραση της μεθανοποίησης. Η παραγωγή SNG όχι μόνο συμβάλλει στην τοπική παραγωγή ενέργειας και στην ανεξαρτητοποίηση των ευρωπαϊκών χωρών από το φυσικό αέριο, αλλά και παρέχει παρόμοια θερμογόνο ικανότητα με το φυσικό αέριο χωρίς μείωση της απόδοσης σε σχέση με τις συμβατικές εγκαταστάσεις παραγωγής ενέργειας. Η Εικόνα 1 παρουσιάζει μια επισκόπηση της διαδικασία παραγωγής SNG μετά από αεριοποίηση με οξυγόνο.



Εικόνα 1: Επισκόπηση διαδικασίας παραγωγής SNG (Weiss M.et al 2008)

Η μεθανοποίηση είναι μια καταλυτική αντίδραση που οδηγεί στη μετατροπή του παραγόμενου από την αεριοποίηση αερίου σε μεθάνιο. Η διαδικασία αποτελείται από ένα μηχανισμό πολλαπλών υψηλά εξώθερμων αντιδράσεων με αποτέλεσμα να ανακύπτουν τεχνολογικά εμπόδια και απαιτήσεις για την κατασκευή του αντιδραστήρα μεθανοποίησης. Η αποδοτικότητα ενός αντιδραστήρα μεθανοποίησης εξαρτάται από τη σύνθεση του παραγόμενου αερίου (το ποσοστό H_2/CO, το εναπομένον περιεχόμενο σε CO_2 , του καταλυτικούς ρύπους κ.ά.), τη βέλτιστη θερμοκρασία και πίεση λειτουργίας καθώς και την επίδραση του ατμού στην καταλυτική ενεργητικότητα και σταθερότητα (Eisenlohr, Moeller & Dry 1974). Ο πιο συνηθισμένος τύπος αντιδραστήρα που χρησιμοποιείται σε βιομηχανικές εφαρμογές είναι αυτός όπου τα καταλυτικά σωματίδια διατάσσονται τυχαία μέσα σε κυλινδρικό αντιδραστήρα σε σταθερές θέσεις (σταθερής κλίνης). Η γεωμετρία των καταλυτικών σωματιδίων διαφέρει ανάλογα με την εφαρμογή και τη χρήση του αντιδραστήρα καθώς και ανάλογα με τη χημική σύσταση του παραγόμενου αερίου. Το μέγεθος και το σχήμα των σωματιδίων επηρεάζει την εσωτερική αντίσταση στο φαινόμενο της διάχυσης, την πτώση της πίεσης στον αντιδραστήρα, αλλά και την καταλυτική ενεργητικότητα λόγω της διαθέσιμης καταλυτικής τους επιφάνειας. Η καταλυτική συμπεριφορά των σωματιδίων επηρεάζεται επίσης από τις μηχανικές ιδιότητες των κραμάτων μετάλλων, από τα οποία αποτελούνται τα σωματίδια. Η γεωμετρία του αντιδραστήρα σε συνδυασμό με τη γεωμετρία των σωματιδίων παίζει σημαντικό ρόλο καθώς η διαμόρφωση των πεδίων ροής, τα προφίλ ταχύτητας και θερμοκρασίας του αντιδραστήρα και η πτώση πίεσης είναι άρρηκτα συνδεδεμένα με τις συσχετίσεις μεταξύ του ενός και του άλλου σωματιδίου, καθώς και μεταξύ του σωματιδίου και του τοιχώματος του αντιδραστήρα. Οι συσχετίσεις αυτές συχνά αναπαριστούνται με το μέγεθος που ορίζεται από το πηλίκο της διαμέτρου του αντιδραστήρα προς τη διάμετρο του σωματιδίου.

$$\mathrm{N} = \frac{\text{diameter of the reactor tube}}{\text{diameter of the particle}} = \frac{\mathrm{D}}{\mathrm{d}} \qquad \text{Εξίσωση 1}$$

Το ρευστό ρέει ανάμεσα στα κενά που σχηματίζονται ανάμεσα στα σωματίδια αλλά και ανάμεσα στα σωματίδια και στον αντιδραστήρα. Τα αντιδρώντα συστατικά από τα οποία αποτελείται το ρευστό μεταφέρονται στην επιφάνεια του καταλυτικού σωματιδίου και στη συνέχεια στο πορώδες υλικό τους, όπου λαμβάνει χώρα η χημική αντίδραση. Τα προϊόντα της αντίδρασης μεταφέρονται στη συνέχεια στην κυρίως ροή. Τα φαινόμενα μεταφοράς μάζας και θερμότητας λαμβάνουν χώρα σε όλο τον αντιδραστήρα.

Οι αντιδραστήρες με τυχαία διατεταγμένα καταλυτικά σωματίδια χρησιμοποιούνται ευρέως σε διάφορες βιομηχανικές εφαρμογές. Ο σχεδιασμός και η κατασκευή τους απαιτεί τον καθορισμό των χαρακτηριστικών λειτουργίας τους και τη σωστή διαστασιολόγηση τους. Πειραματικός εξοπλισμός, ενδελεχής έρευνα και μαθηματική μοντελοποίηση επιστρατεύονται για αυτόν τον σκοπό. Το πιο σημαντικό αλλά και επίμαχο θέμα στην μοντελοποίηση τους είναι η ποιοτική και ποσοτική περιγραφή της ροής του ρευστού και η μεταφορά θερμότητας μέσα στον αντιδραστήρα. Οι συσχετίσεις μεταξύ του ενός σωματιδίου και του άλλου, μεταξύ του σωματιδίου και του ρευστού αλλά και μεταξύ του σωματιδίου και του αντιδραστήρα συντελούν στο μακρόχρονο πρόβλημα της πρόβλεψης των φαινόμενων μεταφοράς στον αντιδραστήρα.

Πολλές μελέτες και έρευνες έχουν πραγματοποιηθεί τις περασμένες δεκαετίες αλλά ακριβείς συσχετίσεις δεν έχουν ανακαλυφθεί και πολλές αντιφάσεις προκύπτουν από τη μια έρευνα στην άλλη. Θεωρητικά μοντέλα για μικρής κλίμακας αντιδραστήρες έχουν αναπτυχθεί βασισμένα σε πειραματικά δεδομένα. Η σύνδεση των τοπικών χαρακτηριστικών της ροής σε μικρής κλίμακας αντιδραστήρες με την ολική ροή και τα φαινόμενα που λαμβάνουν χώρα σε ένα βιομηχανικό αντιδραστήρα είναι αρκετά δύσκολη. Η μαθηματική μοντελοποίηση της στατικής και δυναμικής συμπεριφοράς του αντιδραστήρα έχει μελετηθεί εκτενώς (Wei James 1987, Dommeti, Balakotaiah & West 1999, Ioardanidis 2002). Οι εξισώσεις συνέχειας , ορμής και ενέργειας σε χωρικές διαστάσεις μπορεί να διαφέρουν από διαφορικές σε μερικές διαφορικές εξισώσεις ανάλογα τις υποθέσεις και τη χρήση συγκεκριμένων μοντέλων για την πρόβλεψη των φαινόμενων. Η σύνδεση μεταξύ στατικών μεγεθών, φαινόμενων μεταφοράς και κινητικής οδηγούν σε μη γραμμικά μοντέλα εξισώσεων (Dommeti, Balakotaiah & West 1999).

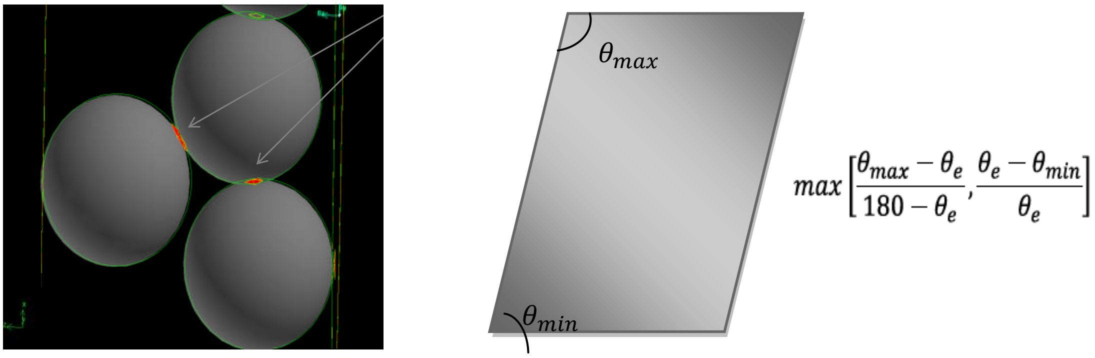
Η αυξανόμενη υπολογιστική δύναμη τις τελευταίες δεκαετίες παρέχει τη δυνατότητα βελτιστοποίησης του σχεδιασμού με τη χρήση της Υπολογιστικής Ρευστομηχανικής (Computational Fluid Dynamics (CFD)). Η αποδοτικότητα και η λειτουργία του μηχανικού εξοπλισμού μπορεί να προβλεφθεί με τη χρήση των CFD προσομοιώσεων.

Πολλές προσεγγίσεις έχουν προταθεί για την προσομοίωση καταλυτικών αντιδραστήρων. Ειδικότερα για την περίπτωση των αντιδραστήρων δυο φάσεων ,αερίου- στερεού, δύο μέθοδοι είναι οι πιο διαδεδομένες: η μέθοδος με ένα ομογενές μέσο και η διακριτή μέθοδος.

Η αποδοτικότητα και η διάρκεια ζωής του καταλυτικού αντιδραστήρα εξαρτάται από το πορώδες υλικό και τη γεωμετρία του σχηματιζόμενου κενού χώρου που σημαίνει τη θέση, το σχήμα, τον προσανατολισμό και το μέγεθος των σωματιδίων που εμπεριέχει. Πολλά προβλήματα προκύπτουν από αυτή την περίπλοκη γεωμετρία κατά την προσομοίωση αλλά και στο σχεδιασμό και το πλέγμα της γεωμετρίας. Αναλυτικότερα, οι γραμμές ροής, η κινηματική των αντιδράσεων, η επιφάνεια όπου λαμβάνουν χώρα οι αντιδράσεις , τα μοντέλα μεταφοράς μάζας και θερμότητας, τα μοντέλα τύρβης αλλά και τα οριακά στρώματα γύρω από τα σωματίδια επηρεάζουν τα αποτελέσματα, αλλά και την αριθμητική σύγκλιση και διάχυση.

Στους βιομηχανικούς αντιδραστήρες όπου το πηλίκο διαμέτρου αντιδραστήρα και σωματιδίου είναι μικρό και ο αριθμός σωματιδίων μεγάλος, η υπολογιστική δύναμη καθώς και ο υπολογιστικός χρόνος ξεπερνούν τις διαθέσιμες υπολογιστικές δυνατότητες. Για το λόγο αυτό η μέθοδος του ενός μέσου που αντιπροσωπεύεται κυρίως από τη μέθοδο προσομοιώσεων πορωδών υλικών πλεονεκτεί και είναι η πιο κοινώς χρησιμοποιούμενη μέθοδος. Για τη χρήση της μεθόδου απαιτείται η εισαγωγή του πλέγματος στο λογισμικό και ο καθορισμός όλων των παραμέτρων για το πορώδη υλικό. Οι πόροι του υλικού διανέμονται μέσω προσεγγιστικών ή θεωρητικών εξισώσεων κατά την οριζόντια και κατακόρυφη διεύθυνση του αντιδραστήρα και δρουν ως αντίσταση στην κυρίως ροή αντιπροσωπεύοντας επιπρόσθετες καταβόθρες ή πηγές ορμής. Η θερμοκρασία, η ταχύτητα, οι συντελεστές μεταφοράς θερμότητας και όλα τα μεγέθη που χαρακτηρίζουν μια συγκεκριμένη θέση αθροίζονται στα σημεία όπου καθορίζονται οι πόροι· και συνεπώς προσεγγιστικές τιμές για τα μεγέθη αυτά χαρακτηρίζουν τη συμπεριφορά του αντιδραστήρα. Η πτώση πίεσης, το προφίλ ταχύτητας και θερμοκρασίας, οι συντελεστές θερμικής αγωγιμότητας και συναγωγιμότητας, οι χημικές αντιδράσεις, το μοντέλο τύρβης και όλα τα μεγέθη επηρεάζονται από την κατανομή των πόρων στο υλικό. Στην παρούσα διπλωματική παρουσιάζονται μελέτες στις οποίες χρησιμοποιήθηκε η μέθοδος αυτή καθώς και άλλες μέθοδοι που θεωρούν το υλικό ως ένα ομογενές μέσο.

Η διακριτή μέθοδος παρότι δεν έχει μελετηθεί εκτενώς και υπάρχει περιορισμένη βιβλιογραφία για τη χρήση της, επιτρέπει την τρισδιάστατη ρεαλιστική γεωμετρία του αντιδραστήρα. Εκτός από τον περιορισμό της διαθέσιμης υπολογιστικής δύναμης, η πολυπλοκότητα της μεθόδου έγκειται και στο γεγονός του σχεδιασμού της γεωμετρίας αλλά κυρίως στη δημιουργία του πλέγματος. Μέχρι την παρούσα χρονική περίοδο συγγραφής, δεν έχει δημιουργηθεί ένα πλέγμα που περιέχει τα ακριβή σημεία επαφής μεταξύ των σωματιδίων και μεταξύ σωματιδίων και αντιδραστήρα καθώς και τον αριθμό σωματιδίων με τις διαστάσεις που χρησιμοποιούνται στις βιομηχανικές εφαρμογές. Οι δυσκολίες στη δημιουργία του τρισδιάστατου πλέγματος για την πολύπλοκη γεωμετρία του αντιδραστήρα δεν κείνται μόνο στις διαθέσιμες υπολογιστικές δυνατότητες του χρήστη λόγω του αριθμού των κελιών, αλλά και στην ποιότητα του δημιουργουμένου πλέγματος. Η διακριτοποίηση του χώρου έχει εμφανή επίδραση στη διακριτοποίηση των εξισώσεων Η αριθμητική σύγκλιση, η σταθερότητα της προσομοίωσης και η αριθμητική διάχυση και επομένως η ποιότητα των αποτελεσμάτων καθορίζονται επί το πλείστον από την ποιότητα του πλέγματος. Το μέγεθος των κελιών, η πυκνότητα του πλέγματος καθώς και η στρεβλότητα τους είναι ενδεικτικοί παράγοντες της ποιότητας του πλέγματος. Κατά τη δημιουργία πλέγματος για αντιδραστήρες που εμπεριέχουν κατά βάση κυλινδρικά ή σφαιρικά σωματίδια, προκύπτουν κελιά με υψηλή στρεβλότητα στα σημεία επαφής μεταξύ των σωματιδίων αλλά και στα σημεία επαφής με τα τοιχώματα του αντιδραστήρα. Η παρουσία αυτών των στρεβλών κελιών μπορεί να οδηγήσει σε αριθμητική διάχυση, σε μη σύγκλιση της αριθμητικής μεθόδου ή ακόμα σε λανθασμένα αποτελέσματα. Στην Εικόνα 2 παρουσιάζονται τα στρεβλά κελιά στα σημεία επαφής σφαιρικών σωματιδίων καθώς και ο ορισμός της στρεβλότητας.


θ_{max}
θ_{min}
$max\left[\frac{\theta_{max}-\theta_e}{180-\theta_e},\frac{\theta_e-\theta_{min}}{\theta_e}\right]$

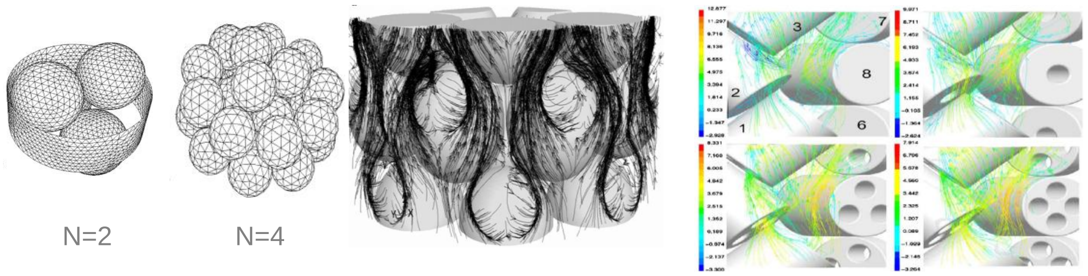
Εικόνα 2: Στρεβλότητα των κελιών

Παρά τα προβλήματα που προκύπτουν στην τρισδιάστατη αναπαράσταση των τυχαία διατεταγμένων καταλυτικών σωματιδίων και τη δημιουργία ενός ποιοτικού πλέγματος, η διακριτή μέθοδος παρουσιάζει ορισμένα πλεονεκτήματα σε σχέση με τη μέθοδο του ομογενούς μέσου. Το πεδίο ροής αντιπροσωπεύεται από ρεαλιστικές γραμμές ροής στον τρισδιάστατο χώρο, χωρίς την εισαγωγή προσεγγιστικών σχέσεων στα μεγέθη για την κατανομή των πόρων. Ο όγκος των σωματιδίων έχει την ακριβή θέση όπου και λύνονται οι διαφορικές εξισώσεις. Η ροή στους καταλυτικούς αντιδραστήρες δεν μπορεί να χαρακτηρισθεί ως στρωτή , μεταβατική ή τυρβώδης καθώς μεταβάλλεται από σημείο σε σημείο. Ο καθορισμός της ταχύτητας στα τοπικά σημεία παίζει σημαντικό ρόλο στο μηχανισμό της μεταφοράς θερμότητας καθώς στάσιμες ροές ευνοούν τη μεταφορά με αγωγή. Η μεταφορά μάζας και θερμότητας είναι επίσης απαλλαγμένη από προσεγγιστικές σχέσεις και τοπικά φαινόμενα διάχυσης, αγωγιμότητας, συναγωγής που συνδέονται με τις πηγές θερμότητας, δηλαδή τις χημικές αντιδράσεις που μπορούν να προβλεφθούν ακριβέστερα στις διακριτές θέσεις.

Η επιλογή της μεθόδου προσομοίωσης εξαρτάται από τις απαιτήσεις του χρήστη, τη διαθέσιμη υπολογιστική δύναμη αλλά και τις υποθέσεις και προσεγγίσεις των μοντέλων που θα χρησιμοποιηθούν. Στην παρούσα διπλωματική εξετάζεται ο σχεδιασμός και η δημιουργία πλέγματος για προσομοιώσεις με τη διακριτή μέθοδο.

Στη διατιθέμενη βιβλιογραφία, με χρήση της διακριτής μεθόδου, το πρόβλημα των στρεβλών κελιών αντιμετωπίζεται με δύο μοντέλα. Το πιο ευρέως διαδεδομένο είναι προσεγγιστικό μοντέλο αστοχίας που αναφέρεται ως “near-miss model”. Το μοντέλο αναπτύχθηκε από τους Dixon & Nijemeisland και ουσιαστικά παραλείπει την αναπαράσταση των σημείων επαφής με μικρή μείωση της ακτίνας των σφαιρικών (της τάξης του 0,5-1%) και κυλινδρικών σωματιδίων (της τάξης του 1%). Το κενό που δημιουργείται ανάμεσα στα σωματίδια είναι ανάλογο της μείωσης της ακτίνας και απαλείφει τη δημιουργία στρεβλών κελιών χωρίς να επιφέρει δραματικές αλλαγές στα αποτελέσματα. Το δεύτερο μοντέλο χρησιμοποιήθηκε από τον Guardo και περιλαμβάνει σωματίδια που επικαλύπτουν το ένα το άλλο αυξάνοντας την επιφάνεια επαφής. Στην παρούσα διπλωματική εξετάστηκε και ένα τρίτο μοντέλο αυτό του προστιθέμενου όγκου σφαίρας ανάμεσα σε σωματίδια που εφάπτονται για την απαλοιφή των στρεβλών κελιών στα σημεία επαφής.

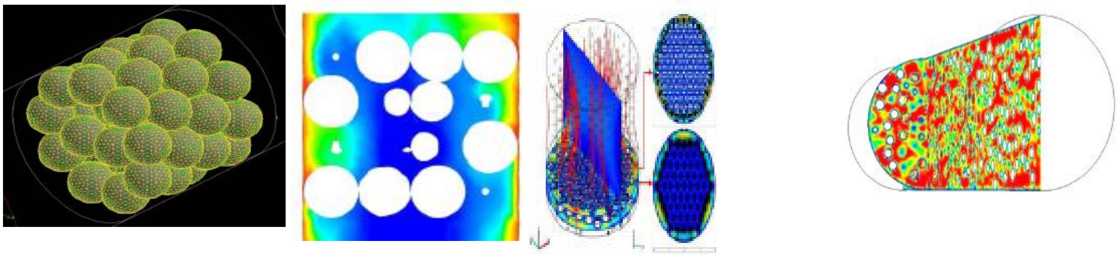
Οι προσομοιώσεις με τη διακριτή μέθοδο λόγω των προαναφερθέντων δυσκολιών άρχισαν πρόσφατα να λαμβάνουν χώρα. Προσομοιώσεις με σφαιρικά σωματίδια έχουν πραγματοποιηθεί από διάφορους ερευνητές με πρωτοπόρους τους Dixon & Nijemeisland. Τα πρώτα επαρκή αποτελέσματα δίνονται με την εισαγωγή του “near-miss model” το 2001 για πηλίκα διαμέτρων N=2 και N=4 και μέγιστο αριθμό σφαιρών 44, επιφέροντας διαφορά θερμοκρασίας μέχρι 2 K σε υψηλούς αριθμούς Reynolds και καλή ποσοτική συμφωνία με τα πειραματικά δεδομένα. Οι έρευνες τους συνεχίστηκαν και επεκτάθηκαν μέχρι το 2008 προβλέποντας υδροδυναμικά και θερμοκρασιακά πεδία για σφαιρικά και κυλινδρικά σωματίδια. Παρά τα μικρό αριθμό σωματιδίων και πηλίκου διαμέτρων που είναι ενδεικτικός της δυσκολίας εφαρμογής της διακριτής μεθόδου, οι έρευνες αυτές παρέχουν πλούσιο υλικό για τα φαινόμενα της ροής και της μεταφοράς μάζας και θερμότητας γύρω από τα καταλυτικά σωματίδια. Τα μοντέλα τύρβης και μεταφοράς θερμότητας που χρησιμοποιήθηκαν καθώς και τα χρήσιμα συμπεράσματα παρατίθενται στα αντίστοιχα κεφάλαια της παρούσας διπλωματικής. Η Εικόνα 3 παρουσιάζει μερικά από τα αποτελέσματα των ερευνών αυτών.



Δημιουργούμενο πλέγμα   Γραμμές ροής N=4   Γραμμές ροής για N=4

Εικόνα 3:CFD προσομοιώσεις για σφαίρες και κυλίνδρους (Dixon A. G. 2001, Nijemeisland M. 2001, Nijemeisland, Dixon 2001, Nijemeisland, Dixon 2004,Nijemeisland, Dixon & Hugh Stitt 2004))

Η διακριτή μέθοδος χρησιμοποιήθηκε και από τους (Romkes S.J.P et al. 2003) για 8÷16 σφαιρικά σωματίδια με N=1÷2 και από τους (Guardo et al. 2005) για 44 σφαίρες και N=3,923. Ενδελεχής έρευνα πραγματοποιήθηκε για το χαρακτηρισμό της ροής, τα μοντέλα τύρβης και τη συσχέτιση του τοπικού αριθμού Nusselt (Nu) με τον αριθμό Reynolds (Re). Στις έρευνες που πραγματοποιήθηκαν από τους (Phavanee N. et al. 2009) και (Jafari et al. 2008) ο αριθμός των σφαιρών φτάνει τα 700 αλλά για την επίτευξη δημιουργίας πλέγματος οι σφαίρες δεν εφάπτονταν μεταξύ τους. Η Εικόνα 4 παρουσιάζει τη σύγκριση μεταξύ των ερευνών τους (Guardo et al. 2005) και τις έρευνες για προσομοιώσεις με περισσότερα σφαιρικά σωματίδια χωρίς σημεία επαφής. Λόγω του μεγάλου αριθμού κελιών που δημιουργήθηκαν για την προσομοίωση ολόκληρου του αντιδραστήρα, οι έρευνες αυτές περιορίστηκαν στην πρόβλεψη μόνο της υδροδυναμικής συμπεριφοράς σε αντίθεση με τις έρευνες για μικρό αριθμό σφαιρών που επεκτείνονται και στη θερμική συμπεριφορά του αντιδραστήρα.



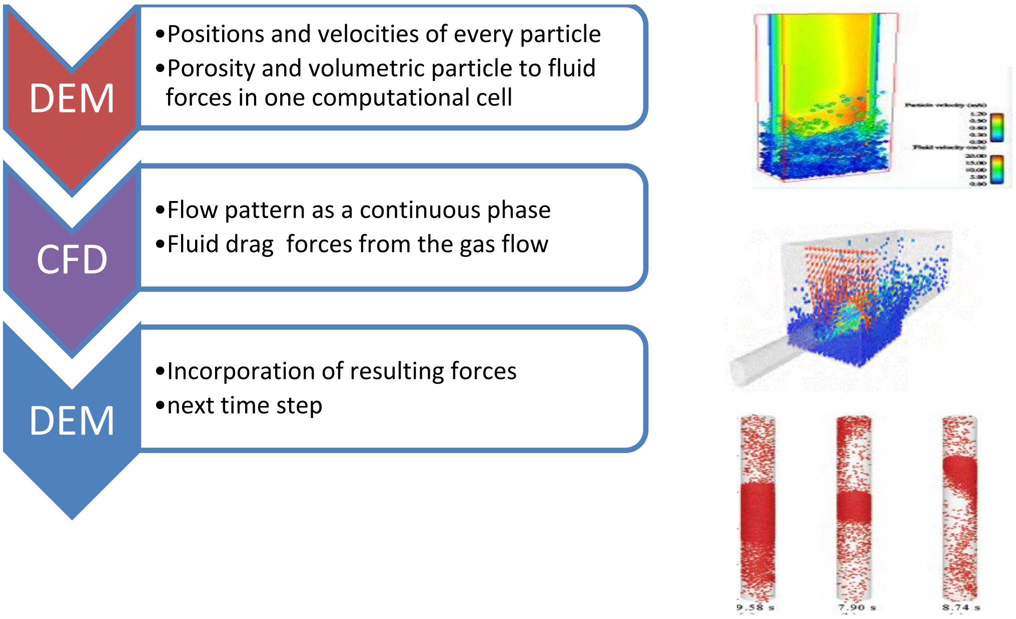
Πλέγμα και θερμοκρασιακό προφίλ για 44 σφαίρες (Guardo et al. 2005)

Προφίλ ταχύτητας από τις έρευνες (Phavanee N. et all 2009) και (Jafari et all 2008) αντίστοιχα

Εικόνα 4: Προσομοιώσεις καταλυτικών αντιδραστήρων

Οι (Motlagh, Hashemabadi 2008) προσομοίωσαν ένα εργαστηριακό αντιδραστήρα με N=2 με 8 κυλινδρικά σωματίδια και συγκρίνανε τα αποτελέσματα με πειραματικά δεδομένα της εγκατάστασης. Η μέθοδος διακριτών στοιχείων DEM (Discrete Element Method) χρησιμοποιήθηκε από τους (Bai et al. 2009) για την πρόβλεψη των τυχαίων θέσεων σφαιρικών και κυλινδρικών σωματιδίων.

Η μέθοδος διακριτών στοιχείων DEM είναι μια αριθμητική μέθοδος που προβλέπει ξεχωριστά την κίνηση πολλαπλών σωματιδίων με διαφορετικά μεγέθη λύνοντας τις εξισώσεις του δεύτερου νόμου κίνησης του Newton. Η μέθοδος αναπτύχθηκε αρχικά από τους (Cundall, Strack 1979). Μέχρι σήμερα διάφορα μοντέλα για τις αλληλεπιδράσεις μεταξύ του ενός και του άλλου σωματιδίου αλλά και μεταξύ του σωματιδίου και ρευστού έχουν προταθεί για την επέκταση της μεθόδου σε περισσότερες εφαρμογές. Στην παρούσα διπλωματική παρατίθενται διάφορα διαθέσιμα λογισμικά με εμπορικούς ή ανοιχτούς κώδικες για τη χρήση της μεθόδου διακριτών στοιχείων DEM. Ο συνδυασμός της μεθόδου με τη CFD μέθοδο έχει αναπτυχθεί πρόσφατα και βρίσκει πολλές τεχνολογικές εφαρμογές. Υπάρχουν δύο τρόποι συνδυασμού των μεθόδων: ο άμεσος τρόπος ή αλλιώς παράλληλη χρήση των δύο μεθόδων και ο έμμεσος τρόπος ή σταδιακός υπολογισμός ή αλλιώς σειριακή χρήση. Στην πρώτη περίπτωση, η DEM μέθοδος υπολογίζει τις θέσεις του σωματιδίου και επιθέτει τις δυνάμεις που ασκούνται σε αυτό από τις αλληλεπιδράσεις με το ρευστό και από τα άλλα σωματίδια σε ένα υπολογιστικό κελί το συγκεκριμένο χρονικό βήμα. Τα αποτελέσματα εισάγονται στον CFD κώδικα όπου υπολογίζονται εκ νέου οι δυνάμεις που ασκούνται στο ρευστό. Στη συνέχεια οι συνιστάμενες δυνάμεις μεταφέρονται στον κώδικα DEM για να ξεκινήσει ο υπολογισμός για το επόμενο χρονικό βήμα. Στην Εικόνα 5 παρουσιάζεται η διαδικασία που ακολουθείται για τον άμεσο συνδυασμό των CFD- DEM μεθόδων καθώς και μερικά παραδείγματα εφαρμογών τους όπως ρευστοποιημένη κλίνη, εισαγωγή κόκκων με δέσμη νερού και πνευματική μεταφορά σωματιδίων.



DEM

- Positions and velocities of every particle
- Porosity and volumetric particle to fluid forces in one computational cell

CFD

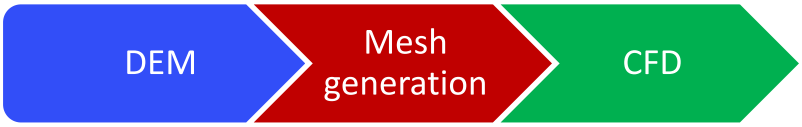
- Flow pattern as a continuous phase
- Fluid drag forces from the gas flow

DEM

- Incorporation of resulting forces
- next time step

Εικόνα 5: Παράλληλη χρήση DEM-CFD και οι εφαρμογές της

Ο έμμεσος τρόπος ή σειριακός επιτρέπει την ξεχωριστή λειτουργία των δύο μεθόδων κατά την οποία τα αποτελέσματα από τη DEM μέθοδο χρησιμοποιούνται για το σχεδιασμό της γεωμετρίας και του πλέγματος που εισέρχονται έπειτα στον CFD κώδικα όπως φαίνεται στην Εικόνα 6.



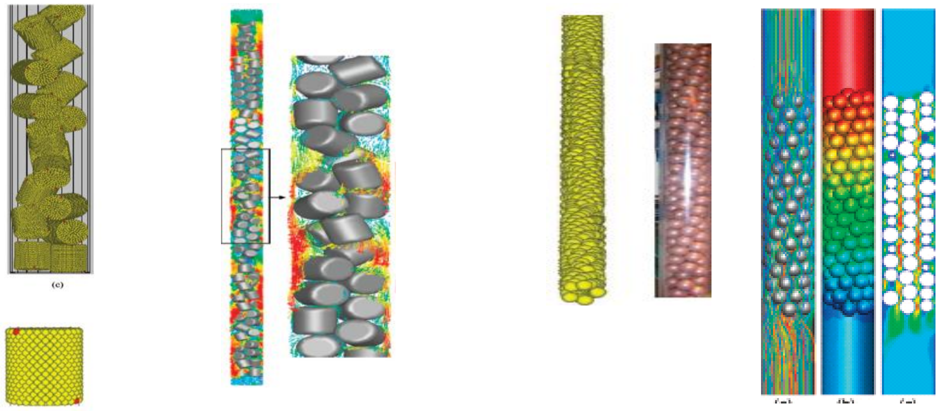
DEM

Mesh generation

CFD

Εικόνα 5: Σειριακή χρήση DEM-CFD

Η σειριακή χρήση CFD-DEM ενδείκνυται για τις προσομοιώσεις αντιδραστήρων με τυχαία διατεταγμένα σωματίδια σε σταθερή κλίνη, διότι τα σωματίδια παραμένουν σε σταθερές θέσεις μέσα στον αντιδραστήρα. Η σειριακή χρήση CFD-DEM χρησιμοποιήθηκε από τους (Bai et al. 2009) για την πρόβλεψη υδροδυναμικής συμπεριφοράς καθ’ όλον το μήκος του αντιδραστήρα με σφαιρικά ή κυλινδρικά σωματίδια για N<4. Το εμπορικό λογισμικό PFC3D χρησιμοποιήθηκε για την πρόβλεψη των θέσεων των σωματιδίων. σωματιδίων. Ειδικότερα για τα κυλινδρικά σωματίδια, χρησιμοποιήθηκαν 1000 σφαίρες για να προσομοιώσουν ένα κύλινδρο καθώς τα περισσότερα λογισμικά λειτουργούν μόνο με σφαιρικά σωματίδια. Στην Εικόνα 6 φαίνονται τα αποτελέσματα των προσομοιώσεων για σφαίρες και κυλίνδρους. Το πλέγμα και για τις δυο περιπτώσεις δημιουργήθηκε με τη χρήση του “near-miss model”.



Κυλινδρικά σωματίδια

Σφαιρικά σωματίδια

Εικόνα 6: Αποτελέσματα CFD-DEM προσομοιώσεων (Bai et al. 2009)

Εκτός από τη βιβλιογραφική έρευνα που πραγματοποιήθηκε και παρουσιάζεται στο κεφάλαιο 2 της παρούσας διπλωματικής, παρατίθεται και μια συγκριτική έρευνα για τα διάφορα μοντέλα που χρησιμοποιήθηκαν στις έρευνες αυτές. Το πρόβλημα της μοντελοποίησης της τύρβης και τα διαφορετικά αποτελέσματα που προκύπτουν με τη χρήση μοντέλων μιας ή δυο εξισώσεων παρουσιάζονται στο τέλος του κεφαλαίου. Η επιρροή της αδιάστατης μεταβλητής y^+ που χαρακτηρίζει την ποιότητα του πλέγματος για την πρόβλεψη των οριακών στρωμάτων στα στερεά όρια, μελετήθηκε από όλους τους ερευνητές για διάφορα εύρη του αριθμού Re. Το μοντέλο τύρβης των Spalart-Allmaras προτάθηκε από τους (Guardo et al. 2005) ενώ οι περισσότεροι ερευνητές καταλήγουν στη χρήση του μοντέλου k- ε RNG. Η σωστή πρόβλεψη του υδροδυναμικού πεδίου γύρω από τα στερεά καταλυτικά σωματίδια επηρεάζει άμεσα το μηχανισμό μεταφοράς θερμότητας. Επομένως η επιλογή του σωστού μοντέλου παίζει καταλυτικό ρόλο. Το k- ε RNG θεωρήθηκε ότι προβλέπει καλύτερα τη συμπεριφορά της ροής γύρω από καμπυλόγραμμα οριακά στρώματα. Τα μοντέλα RSM και LES απαιτούν περισσότερο υπολογιστικό χρόνο και δύναμη και χρησιμοποιήθηκαν για τη σύγκριση των αποτελεσμάτων σε σχέση με τα διάφορα μοντέλα. Για τον αριθμό των κελιών που απαιτείται για προσομοιώσεις με τη διακριτή μέθοδο, το μοντέλο k- ε RNG αποδείχθηκε ότι είναι επαρκές. Η μεταφορά θερμότητας με ακτινοβολία αποδείχθηκε από τους (Nijemeisland, Dixon 2001) ότι δεν παίζει σημαντικό ρόλο και για το λόγο αυτό δεν χρησιμοποιήθηκε σε περαιτέρω έρευνες. Οι συντελεστές της ακτινικής θερμικής αγωγιμότητας, της θερμικής αγωγιμότητας του ρευστού και ο τοπικός αριθμός Nu, αντίθετα, παίζουν σημαντικό ρόλο στη διαμόρφωση του θερμοδυναμικού πεδίου. Οι συντελεστές αυτοί σχετίζονται άμεσα με τον αριθμό Re αλλά και με το πηλίκο διαμέτρων N και διαφοροποιούνται αρκετά με τη μετάβαση της ροής από στρωτή σε τυρβώδη. Η χρήση του “near-miss” model επέφερε την υπερεκτίμηση της θερμοκρασίας κατά 2,5 K για ροές με Re=1922 (Nijemeisland, Dixon 2001). Λόγω της μικρής αυτής διαφοράς, το μοντέλο θεωρήθηκε επαρκές και χρησιμοποιήθηκε από πολλούς ερευνητές. Ο μηχανισμός της μεταφοράς θερμότητας με αγωγή είναι κυρίαρχος όταν η ροή γύρω από τα στερεά όρια είναι στάσιμη. Στα ίδια συμπεράσματα κατέληξαν και οι (Guardo et al. 2006), διατυπώνοντας ότι η μεταφορά θερμότητας από το στερεό καταλύτη στο υγρό παίζει μικρό ρόλο για μικρούς αριθμούς Re. Η μεταφορά θερμότητας από τις χημικές αντιδράσεις μοντελοποιήθηκε με την εισαγωγή πηγών ενέργειας σε ποσοστό της επιφάνειας των καταλυτικών σωματιδίων (Nijemeisland, Dixon & Hugh Stitt 2004). Τα αποτελέσματα έδωσαν διαφορετικά θερμικά προφίλ καθώς και θερμοκρασίες στα τοιχώματα του αντιδραστήρα. Η μελέτη της μεταφορά θερμότητας με αγωγή από τα τοιχώματα του αντιδραστήρα απέδειξε ότι ο μέσος όρος των θερμοκρασιών στα τοιχώματα δεν επηρεάζετα,ι παρά την εξομάλυνση των ακραίων τιμών. Επιπλέον καμία οπτική διαφορά δεν παρατηρήθηκε στα ακτινικά προφίλ θερμοκρασιών (Dixon, Nijemeisland & Stitt 2005). Στην έρευνα των (Romkes S.J.P et al. 2003), η μεταφορά θερμότητας από χημικές αντιδράσεις μοντελοποιήθηκε με την εισαγωγή διαφορετικών θερμοκρασιών στην επιφάνεια ορισμένων καταλυτικών σωματιδίων. Τα αποτελέσματα παρουσιάζουν μια απόκλιση της τάξης του 15% για τις συσχετίσεις μεταξύ των αριθμών Re και Nu ανάμεσα στα τιμές που προέκυψαν από τις προσομοιώσεις και τα πειραματικά δεδομένα. Η μοντελοποίηση με την εισαγωγή διαφορετικών θερμοκρασιών στην επιφάνεια των καταλυτικών σωματιδίων χρησιμοποιήθηκε και στις έρευνες των (Guardo et al. 2005) και (Motlagh, Hashemabadi 2008). Χρήσιμα συμπεράσματα προκύπτουν από την έρευνα των (Motlagh, Hashemabadi 2008) για τη διαφοροποίηση του τοπικού αριθμού Nu ανάλογα με τη θέση και το σχήμα του εκάστοτε σωματιδίου. Στην έρευνα των (Jafari et al. 2008), όπου έγινε χρήση του μοντέλου πρόβλεψης τύρβης, το φαινόμενο της μεταφοράς θερμότητας δε μελετήθηκε λόγω της πολυπλοκότητας του συστήματος εξισώσεων προς επίλυση και της απαιτούμενης υπολογιστικής δύναμης. Επιπρόσθετα στην έρευνα των (Bai et al. 2009) όπου η μέθοδος DEM χρησιμοποιήθηκε για την πρόβλεψη της τυχαίας διάταξης των σωματιδίων σε όλο τον αντιδραστήρα και τη δημιουργία πλέγματος, η μεταφορά θερμότητας δεν περιλήφθηκε στις προσομοιώσεις.

Τα συμπεράσματα που προκύπτουν από τις έρευνες αυτές παρατίθενται περιληπτικά στη συνέχεια. Ο μηχανισμός μετάβασης της ροής από στρωτή σε τυρβώδη δεν μπορεί να προβλεφθεί με ακρίβεια. Η ροή μπορεί να διαφοροποιείται από στρωτή σε τυρβώδη μέσα στον αντιδραστήρα ανάλογα τη γεωμετρία του. Αυτό συμβαίνει λόγω της γεωμετρίας των σωματιδίων, του μεγέθους και του σχήματος που ορίζουν το είδος και τη συμπεριφορά της τοπικής ροής. Μικρές ή μεγάλες δίνες, στάσιμα σημεία ή ροές γύρω από τα σωματίδια ή κοντά στα σημεία επαφής τους είναι τα ιδιαίτερα χαρακτηριστικά της προσομοιωμένης γεωμετρίας. Έρπουσες και στρωτές ροές εντείνουν το μηχανισμό μεταφοράς θερμότητας με αγωγή. Αντίθετα, τυρβώδεις ροές ενισχύουν το φαινόμενο της διασποράς και της διάχυσης και διαφορετικά αποτελέσματα στο υδροδυναμικό και θερμικό προφίλ του αντιδραστήρα προκύπτουν από τη χρήση διαφόρων μοντέλων τύρβης. Οι διακυμάνσεις της ταχύτητας κοντά στα τοιχώματα του αντιδραστήρα παίζουν σημαντικό ρόλο στους μηχανισμούς μεταφοράς θερμότητας. Επιπλέον ο τοπικός αριθμός Nu επηρεάζεται από το σχήμα και τη θέση των σωματιδίων εκτός από την εξάρτηση του από την τοπική ροή και τον αριθμό Re. Καμία γενίκευση ή ακριβής μεθοδολογία δεν μπορεί να εφαρμοσθεί μέχρι στιγμής για τους αντιδραστήρες με τυχαία διατεταγμένα καταλυτικά σωματίδια για τα πολυάριθμα μεγέθη και γεωμετρίες των σωματιδίων και τα διαφορετικά πηλίκα διαμέτρων που μπορεί να προκύπτουν. Ανάλογα το μέγεθος του πλέγματος, την επάρκεια του στα στερεά όρια, το χρησιμοποιούμενο μοντέλο τύρβης και το μηχανισμό μεταφοράς θερμότητας, διαφορετικοί συλλογισμοί πρέπει να εφαρμόζονται στην εκάστοτε προσομοίωση. Η βιβλιογραφική έρευνα που περιλαμβάνεται στην παρούσα διπλωματική, παρουσιάζει μοντέλα που χρησιμοποιήθηκαν δίνοντας επαρκή αποτελέσματα αλλά κάθε εφαρμογή απαιτεί τη διερεύνηση για τα πιο κατάλληλα μοντέλα και τη σύγκριση των αποτελεσμάτων με πειραματικά δεδομένα.

Η DEM μέθοδος χρησιμοποιείται για την πρόβλεψη της κίνησης και της θέσης ξεχωριστών σωματιδίων ή ενός συνόλου αυτών σύμφωνα με το δεύτερο νόμο του Newton. Οι αλληλεπιδράσεις μεταξύ των σωματιδίων ,οι οποίες ορίζουν και τις εφαρμοζόμενες δυνάμεις πάνω στα σωματίδια χρήζουν ιδιαίτερης σημασίας. Ειδικότερα οι εσωτερικές δυνάμεις διαφοροποιούνται από το ένα λογισμικό στο άλλο. Όταν η μέθοδος πρωτοεφαρμόσθηκε από τους (Cundall, Strack 1979) οι δυνάμεις αυτές υπολογίζονταν βασισμένες στο μοντέλο ελατηρίου και στην τριβή ολίσθησης. Σε αυτό το μοντέλο που ονομάζεται μοντέλο "ήπιας επαφής" ή μοντέλο "μετατιθέμενης δύναμης", οι δυνάμεις υπολογίζονται όταν τα στοιχεία ή σωματίδια εισχωρούν ελάχιστα το ένα μέσα στο άλλο. Η εισχώρηση ή επικάλυψη αυτή, παρόλο που δεν είναι ρεαλιστική, εισάγεται ούτως ώστε να λάβει υπόψιν την παραμόρφωση που παρατηρείται στα επιφανειακά στρώματα των στοιχείων (Donzé, Richefeu & Magnier 2009). Εναλλακτικά μοντέλα γνωστά ως "μη-ήπιας επαφής" μπορούν να κατηγοριοποιηθούν σε μεθόδους μηχανικές επαφής ή ομαλές μεθόδους. Οι ομαλές μέθοδοι δεν είναι κατάλληλες για προσομοιώσεις πολλών σωματιδίων που έρχονται σε επαφή το ένα με το άλλο αντίθετα με τις μεθόδους μηχανικής επαφής (Moreau 1999, Luding S. et al. 1996).

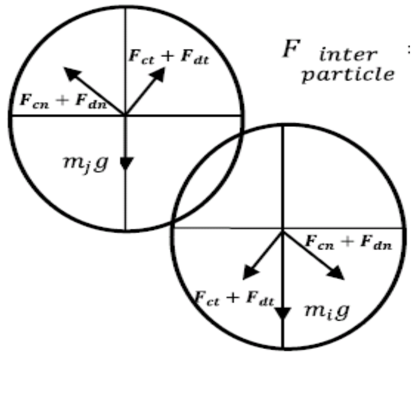
Ο δεύτερος νόμος του Newton περιγράφει την κίνηση των σωματιδίων με την παρακάτω εξίσωση:

$$\mathrm{m_i}\frac{\mathrm{d}\mathbf{v}_\mathrm{i}}{\mathrm{dt}} = \sum_{\mathrm{j=1,j\neq i}}^{\mathrm{k_i}} \left(\mathbf{F}_{\mathrm{c,ij}} + \mathbf{F}_{\mathrm{d,ij}}\right) + \mathrm{m_i}\boldsymbol{g} + \mathbf{F}_\mathrm{i} \qquad (\mathrm{Nt})$$

Εξίσωση 2

όπου m_i και $\mathbf{v}_\mathrm{i}$ η μάζα και η ταχύτητα αντίστοιχα του σωματιδίου i , $\mathbf{F}_{\mathrm{c,ij}}$ η δύναμη επαφής μεταξύ του σωματιδίου i και του σωματιδίου j, $\mathbf{F}_{\mathrm{d,ij}}$ η συνεκτική δύναμη απόσβεσης και $\mathrm{m_i}\boldsymbol{g}$ η δύναμη της βαρύτητας. $\mathbf{F}_\mathrm{i}$ είναι άλλες δυνάμεις που μπορούν να δρουν πάνω στο σωματίδιο συμπεριλαμβανομένου των ηλεκτρομαγνητικών δυνάμεων και των δυνάμεων που προκύπτουν από την αλληλεπίδραση ρευστού και υγρού και θα αμεληθούν στην παρούσα εργασία. Η δύναμη της βαρύτητας εφαρμόζεται στο κέντρο μάζας του σωματιδίου ενώ οι εσωτερικές δυνάμεις επαφής στο σημείο επαφής των σωματιδίων. Στα περισσότερα DEM λογισμικά, χρησιμοποιείται μόνο η γεωμετρία του σφαιρικού σωματιδίου λόγω της απλότητας που παρουσιάζει. Η γεωμετρία της σφαίρας απαιτεί μόνο τον ορισμό της ακτίνας και επιτρέπει τον εύκολο εντοπισμό των σημείων επαφής μειώνοντας την πολυπλοκότητα του μοντέλου αλλά και τις υπολογιστικές απαιτήσεις (Donzé, Richefeu & Magnier 2009).

Οι εσωτερικές δυνάμεις επαφής διακρίνονται σε εφαπτομενικές και κάθετες δυνάμεις και εφαρμόζονται όταν οι σφαίρες βρίσκονται σε μηχανική επαφή. Δύο σφαίρες βρίσκονται σε μηχανική επαφή όταν η απόσταση των κέντρων τους είναι μικρότερη από την απόσταση δ^{ij} που ορίζεται από το άθροισμα των ακτινών τους σε μια διεύθυνση. όπως φαίνεται στην Εικόνα 7. Όταν δυο σωματίδια συγκρούονται, ένα μέρος της κινητικής τους ενέργειας εκλύεται το οποίο για παράδειγμα μπορεί να μετατρέπεται σε θερμότητα. Στις προσομοιώσεις με κοκκώδη σωματίδια, το σχήμα του σωματιδίου θεωρείται αμετάβλητο και η διαφορά θερμοκρασίας που προκύπτει συνήθως θεωρείται αμελητέα (Pöschel T., Schwager T. 2005). Το γραμμικό μοντέλο ελατηρίου των (Cundall, Strack 1979) εκτός από τις δυνάμεις επαφής μοντελοποιεί αυτή την εκλυόμενη ενέργεια με την εισαγωγή της συνεκτικής δύναμης απόσβεσης.



$$F_{\substack{inter\\particle}} = \begin{cases} F_{cn,ij}+F_{ct,ij}+F_{dn,ij}+F_{dt,ij}, & if\ \delta^{ij}>0 \\ 0 & ,\ else \end{cases}$$

$$F_{c,ij} = K\delta^{ij}$$

$$F_{d,ij} = \eta_i\, v$$

Εξίσωση 3

Εικόνα 7: Εσωτερικές δυνάμεις

Η Εξίσωση 3 παρουσιάζει τις εξισώσεις του μοντέλου όπου K είναι ο σταθερός συντελεστής ελαστικότητας ή δυσκαμψίας, η_i ο συντελεστής απόσβεσης του σωματιδίου i και $v = v_i - v_j$ η σχετική ταχύτητα μεταξύ των σωματιδίων i και j. Οι εφαπτομενικές συνιστώσες των δυνάμεων αυτών καθορίζονται από τις επιφανειακές ιδιότητες του σωματιδίου. Η τριβή μεταξύ των σωματιδίων μοντελοποιείται με τις εφαπτομενικές δυνάμεις και μπορεί να είναι στατική ή δυναμική με βάση τον ορισμό των παραμέτρων του μοντέλου. Για μεγάλες κάθετες δυνάμεις ή σχετικά μικρές ταχύτητες, η έκφραση της εφαπτομενικής δύναμης επαφής δίνεται από την Εξίσωση 4 (στατική τριβή βάσει του νόμου Coulomb).

$$\left|\mathrm{F_{ct,ij}}\right| \leq \mu_f \left|\mathrm{F_{cn,ij}}\right| \qquad (\mathrm{Nt}) \qquad \text{Εξίσωση 4}$$

Διαφορετικά όταν προκύπτει ολίσθηση (δυναμική τριβή) δίνεται από την Εξίσωση 5.

$$\left|\mathrm{F_{ct,ij}}\right| = \mu_f \left|\mathrm{F_{cn,ij}}\right| \qquad (\mathrm{Nt}) \qquad \text{Εξίσωση 5}$$

Όταν προκύπτει ολίσθηση μεταξύ των δυο σωματιδίων, υπολογίζεται μόνο η δύναμη της τριβής που λαμβάνει υπόψιν την εκλυόμενη ενέργεια και η δύναμη συνεκτικότητας αμελείται (Cundall, Strack 1979). Πολλά μοντέλα επαφής έχουν προταθεί για την μέθοδο DEM συμπεριλαμβανομένων του γραμμικού μοντέλου ελατηρίου των (Cundall, Strack 1979) και του Hertzian μοντέλου (Malone, Xu 2008). Στα πλαίσια της παρούσας διπλωματικής χρησιμοποιήθηκε και αναλύθηκε μόνο το γραμμικό μοντέλο των (Cundall, Strack 1979) και περισσότερες λεπτομέρειες μπορούν να ευρεθούν στα αντίστοιχα μέρη του κώδικα DEM που χρησιμοποιήθηκε.

Οι αλληλεπιδράσεις μεταξύ σωματιδίων και των ορίων του συστήματος χρήζουν επίσης εξαιρετικής σημασίας. Για παράδειγμα οι μηχανικές ιδιότητες του περιέχοντος δοχείου ή εξοπλισμού, η κίνηση ή μη των τοιχωμάτων του, το σχήμα, το μέγεθος και η ελαστικότητα του επιδρούν στις δυνάμεις που ασκούνται στα σωματίδια από τα όρια ή τοιχώματα του δοχείου. Οι ερευνητές (Pöschel T., Schwager T. 2005) προτείνουν την κατασκευή της γεωμετρίας από σταθερά σωματίδια για την ομαλότητα υπολογισμού των δυνάμεων. Σε πολλές εφαρμογές της μεθόδου χρησιμοποιούνται σταθερά ή κινητά τοιχώματα με διαφορετικά μοντέλα για τις δυνάμεις επαφής τοιχώματος σωματιδίου.

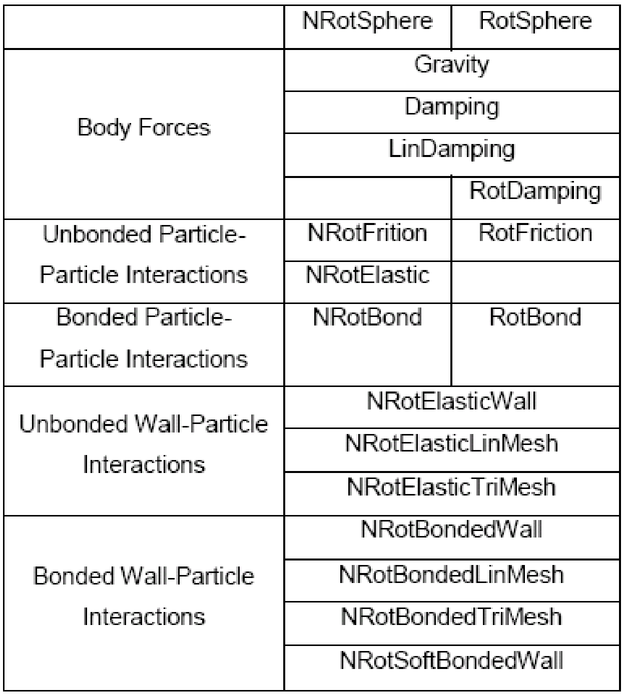
Στις τεχνολογικές και φυσικές εφαρμογές της μεθόδου, η γεωμετρία της σφαίρας δεν είναι επαρκής για να περιγράψει το σχήμα των σωματιδίων. Η τεχνική σύνδεσης σωματιδίων επιστρατεύεται για να δώσει λύση στο πρόβλημα αυτό με τη δημιουργία ομάδων σφαιριδίων συνδεδεμένων μεταξύ τους οι οποίες αναπαριστούν ένα διαφορετικό σχήμα. Η ολική μηχανική συμπεριφορά του συνδεδεμένου group σφαιρών εκτιμάται από τις συλλογικές συνεισφορές των συμμετεχόντων σωματιδίων στην κίνηση, στη μετατόπιση, στην ολίσθηση και στην περιστροφή (Donzé, Richefeu & Magnier 2009). Σύνδεσμοι σχετικής ελαστικότητας ή δυσκαμψίας μπορούν να ορισθούν και διαφορετικό μοντέλο για τις δυνάμεις επαφής μεταξύ αυτών των σωματιδίων μπορεί να χρησιμοποιηθεί.

Ο πιο σημαντικός παράγοντας στις DEM προσομοιώσεις, όπως και σε κάθε άλλη προσομοίωση μεταβατικού φαινόμενου, είναι η κατάλληλη επιλογή του χρονικού βήματος που επηρεάζει την ακρίβεια και την ευστάθεια της προσομοίωσης. Εκφράσεις μικρής τάξεως της χρονικής διακριτοποίησης, όπως η κεντρική διαφορά, χρησιμοποιούνται συνήθως. Πολλοί ερευνητές έχουν προσπαθήσει να ορίσουν ένα μέγιστο χρονικό βήμα βασισμένο στις ιδιότητες των σωματιδίων. Η τιμή αυτή εξαρτάται κυρίως από τη φυσική συχνότητα του αντίστοιχου συστήματος μάζας-ελατηρίου και για το λόγο αυτό διαφοροποιείται από εφαρμογή σε εφαρμογή. Η κρίσιμη τιμή του χρονικού βήματος εξαρτάται από τη μάζα του σωματιδίου, το συντελεστή ελαστικότητας ή δυσκαμψίας και τον ολικό αριθμό σωματιδίων που προσομοιώνονται. Το μοντέλο "ήπιας επαφής" ή μοντέλο "μετατιθέμενης δύναμης" επιτρέπει τη μοντελοποίηση πολλαπλών σωματιδίων που εφάπτονται μεταξύ τους. Σε περίπτωση όμως που οι αλληλεπιδράσεις στην κάθετη διεύθυνση επαφής των σωματιδίων απαιτούν μεγάλη δυσκαμψία και στιβαρότητα για τη μικρότερη μετατόπιση, η χρήση μικρού χρονικού βήματος είναι απαραίτητη για την ευστάθεια της λύσης. Τα μοντέλα αυτά επιτρέπουν πιο ελαστικές αλληλεπιδράσεις μεταξύ των σωματιδίων αυξάνοντας τεχνητά το χρόνο επαφής τους για την επίτευξη ευσταθών λύσεων αλλά και για μικρότερες υπολογιστικές απαιτήσεις.

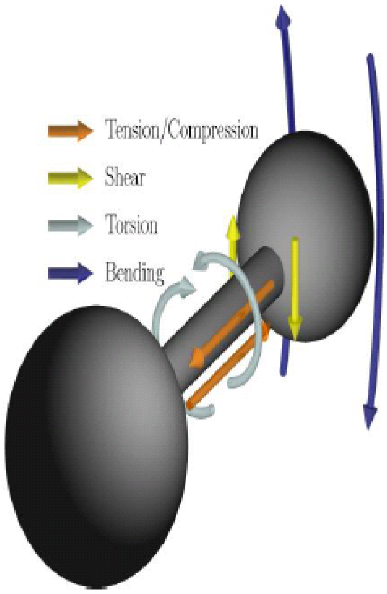
Το λογισμικό ESyS-Particle είναι ένας ανοιχτός κώδικας που εφαρμόζει τη μέθοδο DEM και έχει χρησιμοποιηθεί σε πολλές γεωφυσικές εφαρμογές. Ο κώδικας αναπτύχθηκε στις αρχές του 1990 και από τότε έχει εξελιχθεί με τη συνεισφορά πολλών ερευνητών. Η τελευταία έκδοση του παρέχεται διαδικτυακά από το πανεπιστήμιο Queensland της Αυστραλίας σε συνεργασία με το υπολογιστικό κέντρο ερευνών γεωλογικών και φυσικών συστημάτων (ESSCC). Το λογισμικό επιτρέπει στο χρήστη την περάτωση παράλληλων προσομοιώσεων με την εισαγωγή Message Passing Interface (MPI) τη διάσπαση του χώρου σε μέρη όπου ο βασικός κώδικας γραμμένος σε C++ μπορεί να επιλυθεί ξεχωριστά. Το χαρακτηριστικό αυτό επιτρέπει την περάτωση μεγάλων προσομοιώσεων σε παράλληλους υπολογιστές, clusters ή multi-core υπολογιστές σε λειτουργικά συστήματα βασισμένα σε Linux. Ο κώδικας τρέχει μετά την εκτέλεση της κατάλληλης εντολής στο Linux terminator που καλεί το βασικό κείμενο- σενάριο γραμμένο σε Python. Η επένδυση του κώδικα με τη γλώσσα προγραμματισμού Python μέσω της εφαρμογής Application Programming Interface (API) επιτρέπει τη μοντελοποίηση των αλληλεπιδράσεων της DEM μεθόδου και τον προσδιορισμό των παραμέτρων σε ένα βασικό και σύντομο κείμενο που αλληλεπιδρά με το βασικό C++ κώδικα. Η χρήση των λογισμικών POVray και VTK με την εισαγωγή συγκεκριμένων εντολών παρέχει τη δυνατότητα οπτικοποίησης των αποτελεσμάτων με την εξαγωγή εικόνων σε συγκεκριμένα χρονικά διαστήματα εικόνων των οποίων η χρήση εξυπηρετεί και σκοπούς debugging. Το λογισμικό ESyS-Particle παρέχει στους χρήστες του εκπαιδευτικό υλικό αλλά και ένα ανοιχτό forum για πληροφορίες και ερωτήσεις μέσω του launchpad.net. Περισσότερες πληροφορίες μπορούν να βρεθούν στους διαδυκτιακούς τόπους που αναφέρονται ως βιβλιογραφικές πηγές.

Ο κώδικας ESyS-Particle προσφέρει τη δυνατότητα προσομοίωσης σφαιρικών σωματιδίων ή σωματιδίων διαφορετικών σχημάτων με τη χρήση της τεχνικής σύνδεσης σφαιριδίων σε ομάδες. Στο βασικό κώδικα οι συνδεδεμένες σφαίρες μπορούν να δημιουργηθούν με τέσσερεις τρόπους: the SimpleBlock, the CubicBlock, The Hexablog and RandomPacker (Weatherley Dion, 2009b). Ένας άλλος τρόπος για τη δημιουργία συνδεδεμένων σωματιδίων, που αναπαριστούν διάφορα σχήματα, είναι με τη χρήση του εξωτερικού πακέτου Lsm.GenGeo, το οποίο χρησιμοποιεί τις βιβλιοθήκες του ESyS-Particle. Η απαιτούμενη γεωμετρία γεμίζεται τυχαία με σφαιρίδια διαφόρων ακτινών, των οποίων το εύρος καθορίζεται από το χρήστη. Το Lsm.GenGeo δημιουργεί ένα αρχείο με κατάληξη .geo το οποίο διαβάζεται από το βασικό κώδικα.

Το είδος των σφαιρών το οποίο χρησιμοποιείται στο λογισμικό διακρίνεται σε περιστρεφόμενες και μη-περιστρεφόμενες. Οι μη περιστρεφόμενες σφαίρες δεν έχουν κανένα βαθμό ελευθερίας ως προς την περιστροφή ενώ οι περιστρεφόμενες μπορούν να αλλάζουν διεύθυνση ανάλογα με την ορμή (Epydoc 3.0.1, 2009). Για τα δύο είδη σφαιρών διαφορετικά μοντέλα δυνάμεων επαφής μπορούν να εφαρμοσθούν. Για συνδεδεμένες σφαίρες ή μη οι δυνάμεις βαρύτητας και απόσβεσης παραμένουν ίδιες. Το θεωρητικό υπόβαθρο των μοντέλων μπορεί να βρεθεί στις ακόλουθες δημοσιεύσεις (Cundall, Strack 1979, Mora, Place 1994, Place, Mora 1999). Επιπλέον περισσότερες λεπτομέρειες μπορούν να βρεθούν στα αντίστοιχα τμήματα του ανοιχτού κώδικα και στο δικτυακό τόπο (Weatherley Dion, 2008b). Η Εικόνα 8 παρουσιάζει τα κυριότερα μοντέλα για τα δυο είδη σφαιρών που διατίθενται από το ESyS-Particle.



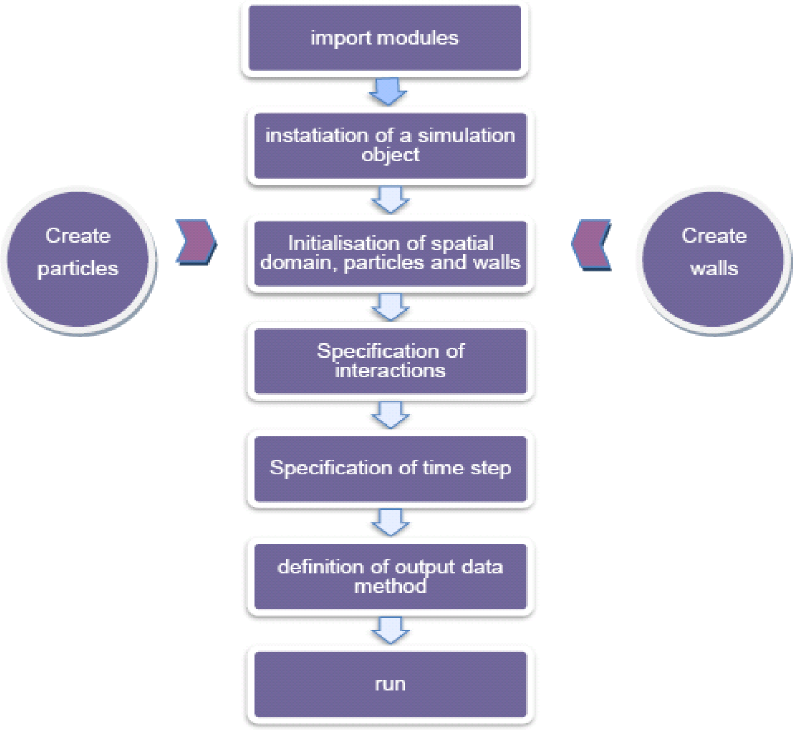
	NRotSphere	RotSphere
Body Forces	Gravity	
	Damping	
	LinDamping	
		RotDamping
Unbonded Particle-Particle Interactions	NRotFrition	RotFriction
	NRotElastic	
Bonded Particle-Particle Interactions	NRotBond	RotBond
Unbonded Wall-Particle Interactions	NRotElasticWall	
	NRotElasticLinMesh	
	NRotElasticTriMesh	
Bonded Wall-Particle Interactions	NRotBondedWall	
	NRotBondedLinMesh	
	NRotBondedTriMesh	
	NRotSoftBondedWall	



Εικόνα 8: Σύνοψη των διατιθέμενων μοντέλων από ESyS-Particle

Για τη μοντελοποίηση των ορίων του συστήματος, υπάρχουν τρία είδη τοίχων που δύναται να χρησιμοποιηθούν, οι επίπεδοι τοίχοι (πλάκες άπειρου μήκους), τοίχοι με γραμμικά διδιάστατα πλέγματα και τοίχοι με τριγωνικά τρισδιάστατα πλέγματα. Οι επίπεδοι τοίχοι μπορούν να εισαχθούν στην προσομοίωση με την ένταξη εντολών στο βασικό κείμενο του κώδικα ενώ η δημιουργία πλέγματος για τα άλλα δύο είδη απαιτεί τη χρήση κάποιου εξωτερικού σχεδιαστικού πακέτου. Το ESyS-Particle προτείνει τη χρήση του ανοιχτού σχεδιαστικού λογισμικού gmsh (http://www.geuz.org/gmsh/). Μετά τη σχεδίαση του πλέγματος στο gmsh και τη δημιουργία του αρχείου mesh, παρέχεται ένας κώδικας για τη μετατροπή του αρχείου σε lsm σύμφωνα με τις οδηγίες (Weatherley Dion, 2009d). Το αρχείο αυτό φορτώνεται στο πρόγραμμα μέσω εντολών στο βασικό κείμενο και έπειτα ορίζονται οι αλληλεπιδράσεις των σωματιδίων με τα τοιχώματα. Το λογισμικό δεν διαθέτει κάποιο μοντέλο τριβής μεταξύ σωματιδίων και τοιχώματος και για το λόγο αυτό προτείνονται δύο τεχνικές σύνδεσης σωματιδίων στα τοιχώματα.

Το βασικό κείμενο του κώδικα ξεκινά όπως άλλωστε και σε κάθε πρόγραμμα με την εισαγωγή εντολών που συνδέουν το κείμενο με τις απαραίτητες βιβλιοθήκες της γλώσσας προγραμματισμού. Κατηγορίες και υποκατηγορίες εντολών χρησιμοποιούνται για να φορτώσουν τα modules. Τα modules είναι οι απαραίτητες εντολές για την εισαγωγή των μοντέλων στο πρόγραμμα. Στη συνέχεια απαιτείται ο ορισμός ή η δημιουργία του αντικειμένου που περιγράφουν τα μοντέλα αυτά καθώς η Python είναι μια γλώσσα προγραμματισμού βασισμένη σε αντικείμενα. Έπειτα καθορίζεται ο αριθμός των επεξεργαστών και η διάσπαση του χώρου σε κάθε επεξεργαστή για παράλληλες ή μη επεξεργασίες. Ο ορισμός του είδους σφαίρας που θα χρησιμοποιηθεί, ο αλγόριθμος εύρεσης των σημείων επαφής καθώς και η συχνότητα χρήσης του προσδιορίζονται μετά την εισαγωγή του αντικειμένου προσομοίωσης. Ακολουθώντας την εγκαθίδρυση του αντικειμένου προσομοίωσης, απαιτείται η εισαγωγή των σωματιδίων , των τοιχωμάτων του συστήματος και ο προσδιορισμός των αλληλεπιδράσεων σωματιδίου- σωματιδίου και σωματιδίου-τοιχώματος. Η εξαγωγή αποτελεσμάτων είτε μέσω αρχείων δεδομένων είτε μέσω αρχείων εικόνων αλλά και το χρονικό βήμα εξαγωγής πραγματοποιείται με την εισαγωγή κατάλληλων εντολών στο βασικό κείμενο. Τέλος προσδιορίζεται το χρονικό βήμα και ο μέγιστος αριθμός χρονικών βημάτων της προσομοίωσης. Μετά την ολοκλήρωση του set up, η προσομοίωση αρχίζει με την εκτέλεση εντολής λειτουργίας του προγράμματος στο τερματικό του λειτουργικού συστήματος. Η Εικόνα 9 παρουσιάζει το βασικό διάγραμμα για το set up του προγράμματος.


import modules
instatiation of a simulation object
Create particles
Initialisation of spatial domain, particles and walls
Create walls
Specification of interactions
Specification of time step
definition of output data method
run

Εικόνα 9: Βασικό διάγραμμα set-up προσομοίωσης στο ESyS-Particle

Οι DEM προσομοιώσεις, που πραγματοποιήθηκαν για τον σκοπό αυτής της διπλωματικής, περιείχαν σφαιρικά καταλυτικά σωματίδια αλλά και συνδεδεμένα σφαιρικά σωματίδια, τα οποία αναπαριστούσαν καταλυτικούς κυλίνδρους. Για τις δύο γεωμετρίες του καταλύτη, χρησιμοποιήθηκε το είδος της μη περιστρεφόμενης σφαίρας. Η αρχική παράμετρος για τη σχεδίαση του αντιδραστήρα ορίστηκε το πηλίκο διαμέτρων το οποίο τέθηκε ίσο με N=6. O αντιδραστήρας είχε διάμετρο 60mm και ύψος 300mm ενώ η διάμετρος των σωματιδίων τέθηκε ίση με d=10mm και το ύψος των κυλινδρικών καταλυτών ίσο με 15mm.

Για την αναπαράσταση της πτώσης των σωματιδίων στον αντιδραστήρα, χρησιμοποιήθηκε μια χρονική υπορουτίνα ούτως ώστε τα σωματίδια να εισέρχονται στην προσομοίωση και αντίστοιχα στον αντιδραστήρα σε διαφορετικά χρονικά βήματα. Τα δεδομένα εισόδου εκτός από τον ορισμό των μοντέλων για τις αλληλεπιδράσεις των σωματιδίων και τους φυσικούς νόμους, απαιτούν την δημιουργία και φόρτωση σωματιδίων και τοίχων στο βασικό κείμενο του κώδικα. Για τις προσομοιώσεις με σφαιρικούς καταλύτες, η δημιουργία και εισαγωγή των σωματιδίων πραγματοποιήθηκε στο βασικό κείμενο με μία σειρά εντολών. Αντίθετα για τις προσομοιώσεις με κυλινδρικούς καταλύτες, η δημιουργία των σωματιδίων πραγματοποιήθηκε με εξωτερικά πακέτα. Το εξωτερικό πακέτο Lsm.GenGeo, που επιτρέπει τη δημιουργία διαφόρων σχημάτων με την τεχνική σύνδεσης σφαιριδίων, τρέχει ξεχωριστά από το βασικό κείμενο αποδίδοντας αρχεία κειμένου που περιέχουν απαραίτητες πληροφορίες για τη δημιουργία των σχημάτων αυτών. Παρότι το Lsm.GenGeo αποτελεί ένα πολύ χρήσιμο εργαλείο, περιέχει κάποιες τυχαίες παραμέτρους για τα χαρακτηριστικά των σφαιριδίων. Το βασικό μειονέκτημα του συνοψίζεται στον τυχαίο ορισμό του αριθμού ταυτότητας των σφαιριδίων αλλά και στην τυχαία τοποθέτηση τους μέσα στη ζητούμενη γεωμετρία. Για την αυτόματη σχεδίαση των κυλίνδρων σε ένα σχεδιαστικό πρόγραμμα απαιτείται εκτός από την ακτίνα και το ύψος του, ο προσδιορισμός του ανύσματος του και η θέση του. Για τον προσδιορισμό του ανύσματος αυτού, αρκεί ο εντοπισμός της θέσης δύο αντιδιαμετρικών σφαιριδίων σε συγκεκριμένες θέσεις. Επομένως όταν δεν υπάρχουν δυο συγκεκριμένα σωματίδια τα οποία θα παρέχουν τις απαραίτητες πληροφορίες για τα όρια της γεωμετρίας αυτής αλλά ακόμα και όταν τεχνητά δημιουργηθούν δεν έχουν συγκεκριμένους αριθμούς ταυτότητας για τον εντοπισμό τους, δεν είναι δυνατή η αυτόματη σχεδίαση των κυλίνδρων στις τρεις διαστάσεις. Το πρόβλημα αυτό αντιμετωπίστηκε με τη δημιουργία νέου κώδικα σε Python, ο οποίος απέδιδε αρχεία με την ίδια κατάληξη όπως το Lsm.GenGeo. Τα αρχεία αυτά περιείχαν τις απαραίτητες πληροφορίες (Weatherley Dion, 2009c) για τη σύνδεση των σφαιρών μέσα στη γεωμετρία του κυλίνδρου και τηρούσαν τις προϋποθέσεις για την αυτόματη σχεδίαση. Ο αριθμός των σωματιδίων n, που εισήχθη σε κάθε προσομοίωση ορίσθηκε σύμφωνα με την απαιτούμενη διαπερατότητα ε του αντιδραστήρα και δίνεται από την Εξίσωση 6.

$$\mathrm{n} = \frac{\mathrm{V_{reactor}}}{\mathrm{V_{particle}}} \cdot (1 - \varepsilon)$$

Εξίσωση 6

όπου $\mathrm{V_{reactor}}$ ο όγκος του αντιδραστήρα και $\mathrm{V_{particle}}$ ο όγκος του καταλυτικού σωματιδίου. Η διαπερατότητα ε ή ιδιότητα του πορώδους υλικού είναι ένα διαμφισβητούμενο θέμα στις

DEM προσομοιώσεις και πρέπει πάντα να ελέγχεται και να συγκρίνεται με πειραματικά δεδομένα. Η απαιτούμενη διαπερατότητα για το συγκεκριμένο αντιδραστήρα ορίσθηκε ίση με ε=0,4.

Για την εισαγωγή των κυλινδρικών τοιχωμάτων του αντιδραστήρα στις προσομοιώσεις χρησιμοποιήθηκαν τρία διαφορετικά είδη πλέγματος, τα οποία δημιουργήθηκαν με το λογισμικό gmsh σύμφωνα με τη διαδικασία που περιγράφεται στο (Weatherley Dion, 2009d). Τα τρία διαφορετικά πλέγματος κατηγοριοποιούνται σε αραιής, μέτριας και υψηλής πυκνότητας πλέγμα. Τα αρχεία mesh μετατράπηκαν σε lsm μέσω του κώδικα που παρέχεται από το λογισμικό (Weatherley Dion, 2009d).Για το πάτωμα και την οροφή του κυλινδρικού αντιδραστήρα χρησιμοποιήθηκαν επίπεδα τοιχώματα, τα οποία δημιουργήθηκαν και εισήχθησαν στο βασικό κείμενο του κώδικα. Οι γεωμετρίες σχεδιάστηκαν σε χιλιοστά.

Τα είδη αλληλεπιδράσεων που χρησιμοποιήθηκαν περιλαμβάνουν τη βαρύτητα, την ελαστική άπωση και την τριβή μεταξύ συνδεδεμένων και μη σφαιρών. Οι δυνάμεις απόσβεσης αμελήθηκαν στις προσομοιώσεις με σφαιρικούς καταλύτες λόγω ευστάθειας της λύσης. Παρότι η παράβλεψη των δυνάμεων απόσβεσης συντελεί στη μη ρεαλιστική προσομοίωση, ο σκοπός χρήσης της μεθόδου στα πλαίσια αυτής της διπλωματικής μπορεί να το επιτρέψει. Η παράβλεψη των δυνάμεων απόσβεσης οδηγεί σε διαρκείς κρούσεις μεταξύ των σωματιδίων, αλλά, καθώς αυξάνεται ο αριθμός εισαγόμενων σωματιδίων με το χρόνο, τα πρώτοεισερχόμενα σωματίδια οδηγούνται σε ηρεμία.

Η αύξηση του χρονικού βήματος, αλλά και ο συνολικός αριθμός χρονικών βημάτων έπαιξαν καθοριστικό ρόλο στην έκβαση των αποτελεσμάτων. Μόνο για τις προσομοιώσεις με σφαίρες, χρειάστηκαν 972 σφαίρες για να πληρείται το κριτήριο διαπερατότητας σύμφωνα με την Εξίσωση 6. Ο αριθμός αυτός απαιτεί μεγάλη υπολογιστική δύναμη για τον εντοπισμό των σημείων επαφής, έτσι ο χρόνος επαφής αυξήθηκε τεχνητά με την αύξηση των σωματιδίων με αποτέλεσμα τη μεγαλύτερη εισχώρηση των εφαπτόμενων σωματιδίων. Η αύξηση της τιμής ελαστικότητας ή δυσκαμψίας, ούτως ώστε να δημιουργηθεί μια πιο σκληρή επαφή σε δύσκαμπτα σωματίδια, απαιτεί μικρότερα χρονικά βήματα. Αντίθετα ο αυξανόμενος αριθμός σωματιδίων απαιτεί μεγαλύτερα χρονικά βήματα. Στις προσομοιώσεις που πραγματοποιήθηκαν, ένα ευρύ πεδίο τιμών του συντελεστή ελαστικότητας ή δυσκαμψίας και του χρονικού βήματος δοκιμάστηκαν, ώστε να βρεθεί ο βέλτιστος συνδυασμός. Κρατώντας σταθερό το χρονικό βήμα και αυξάνοντας το συντελεστή, τα σωματίδια εξαφανιζόντουσαν μερικά χρονικά βήματα μετά την εισαγωγή τους. Αντίθετα αλλάζοντας το χρονικό βήμα, τα σωματίδια δεν εισάγονταν καν στο χώρο. Στο διαδυκτιακό τόπο του κώδικα, παρουσιάζονται κάποιες συσχετίσεις για τη σχέση του συντελεστή ελαστικότητας ή δυσκαμψίας με το χρονικό βήμα.

$$\mathrm{dt} < 0.1\sqrt{\frac{\mathrm{m_{smallest}}}{\mathrm{K_{max}}}} \qquad (\mathrm{s})$$ Εξίσωση 7

όπου *dt* η χρονική αύξηση, $\mathrm{m_{smallest}}$η μάζα του μικρότερου σωματιδίου και $\mathrm{K_{max}}$ η μέγιστη τιμή του συντελεστή ελαστικότητας ή δυσκαμψίας.

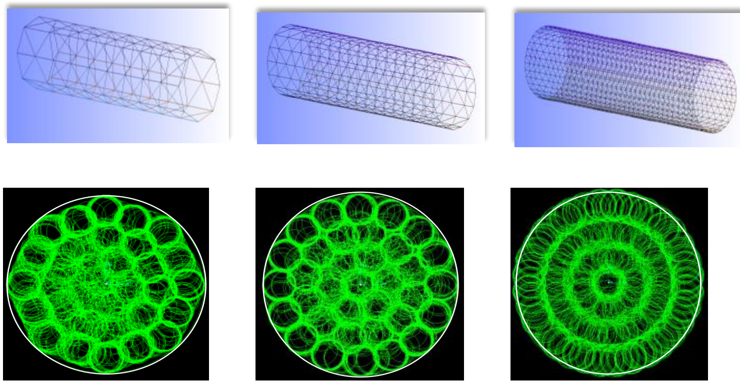
Η μαθηματική μοντελοποίηση του ESyS-Particle είναι αδιάστατη και ο ορισμός των μονάδων εξαρτάται από το χρήστη. Οι μονάδες μέτρησης που χρησιμοποιήθηκαν ήταν τα χιλιοστά του μέτρου τόσο για τις παραμέτρους των μοντέλων, όσο και για το σχεδιασμό του πλέγματος, καθώς οι πολύ μικρές τιμές ακτίνας δεν ήταν αποδεκτές για τη χρήση μεγαλύτερης μετρητικής μονάδας. Περισσότερες διευκρινήσεις για το μετρητικό σύστημα μπορούν να βρεθούν στην πηγή (Weatherley Dion, 2009e). Η διαδικασία προσδιορισμού του συντελεστή ελαστικότητας ή δυσκαμψίας, της αύξησης χρονικού βήματος και των μετρικών μονάδων παρατίθενται στο αντίστοιχο κεφάλαιο της παρούσας εργασίας. Οι μέθοδοι εξαγωγής αποτελεσμάτων που χρησιμοποιήθηκαν ήταν το πακέτο οπτικοποίησης αποτελεσμάτων POVray και το module CheckPointer.

Οι προσομοιώσεις με σφαιρικούς καταλύτες περιείχαν 972 σφαίρες ακτίνας 5 mm σύμφωνα με την Εξίσωση 6 και την προϋπόθεση της διαπερατότητας ε=0,4. Η μάζα της σφαίρας ορίσθηκε ίση με 1 και η τιμή της αρχικής ταχύτητας ίση με 1000 mm/s στην αρνητική διεύθυνση του z άξονα για να επιταχυνθεί η αργή λόγω χρονικού βήματος πτώση των σφαιρών. Ο υπολογιστικός χώρος προσδιορίσθηκε από ένα κύβο με τις ακόλουθες διαστάσεις (-30,-30,0)÷(30,30,310). Η βαρύτητα τέθηκε ίση με 9810 mm/s^2 και οι δυνάμεις απόσβεσης αμελήθηκαν για την ολική προσομοίωση. Οι αλληλεπιδράσεις μεταξύ σφαιρών και των δύο ειδών τοιχώματος ορίσθηκαν ως ελαστικέ σύμφωνα με το μοντέλο NRotElastic. Η τιμή του συντελεστή ελαστικότητας ή δυσκαμψίας για τα τοιχώματα προσδιορίσθηκε σύμφωνα με την Εξίσωση 8 που προτείνεται στο διαδυκτιακό τόπο του λογισμικού.

$$\mathrm{K_{wall}} > {}^{\mathrm{m}\cdot\mathrm{g}}/_{\mathrm{r}} \qquad (\mathrm{N/m})$$ Εξίσωση 8

Το φαινόμενο της ελαστικής άπωσης και της αντίστασης τριβής αποδόθηκε με το μοντέλο NRotFrictionPrms όπου οι τελικές τιμές του κανονικού και διατμητικού συντελεστή ελαστικότητας ή δυσκαμψίας ορίσθηκαν ίσες με 10^7 και 10^5 αντίστοιχα ενώ ο συντελεστής τριβής ίσος με 0,7. Η αύξηση του χρονικού βήματος τέθηκε ίση με 0,0001s σύμφωνα με τα κριτήρια ευστάθειας.

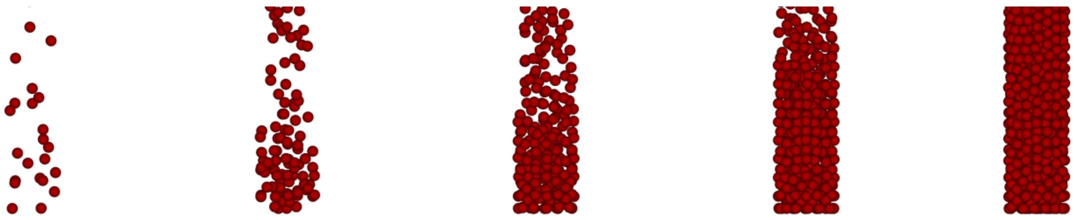
Οι σφαίρες εισήχθησαν σε τυχαίες θέσης στο επάνω μέρος του αντιδραστήρα ανά 100 χρονικά βήματα. Μετά την περάτωση 300.000 χρονικών βημάτων τα αποτελέσματα παρουσίασαν μια σταθεροποίηση που θεωρήθηκε επαρκής. Τα σωματίδια που εισήχθησαν τελευταία κοντά στην οροφή του αντιδραστήρα παρουσίαζαν μια μικρή ταλάντωση γύρω από την προβλεπόμενη θέση τους λόγω της έλλειψης απόσβεσης. Η Εικόνα 10 παρουσιάζει τα αποτελέσματα της μελέτης για την πυκνότητα του χρησιμοποιούμενου πλέγματος στις ολικές προσομοιώσεις των 972 σφαιρών.



Αραιό Μέτριο Πυκνό

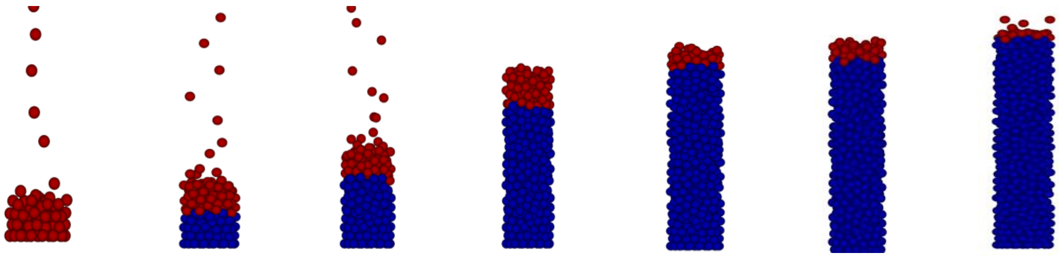
Εικόνα 10: Τα αποτελέσματα από τη σύγκριση των τριών ειδών πλεγμάτων για τις ολικές ροσομοιώσεις των 972 σφαιρών στο ESyS-Particle

Λόγω υπολογιστικού χρόνου αλλά και καλύτερης πρόβλεψης των σημείων επαφής μεταξύ τοιχώματος και σωματιδίων έγινε χρήση του μέτριου πλέγματος για τις ολικές προσομοιώσεις και του αραιού πλέγματος για τις διαδοχικές προσομοιώσεις. Στο λογισμικό υπάρχει ένα ελάττωμα (bug) για τους τοίχους με πλέγματα, το οποίο έχει αναφερθεί από πολλούς χρήστες και ονομάζεται ελάττωμα των προσκολλημένων σφαιρών. Αυτό σημαίνει ότι κάποιες σφαίρες προσκολλώνται στα τοιχώματα και παραμένουν εκεί χωρίς να ακολουθούν τους φυσικούς νόμους. Τα αποτελέσματα δείχνουν ότι το φαινόμενο αυτό ελαττώνεται όσο πιο αραιό είναι το πλέγμα του τοίχου. Η Εικόνα 11 παρουσιάζει τα αποτελέσματα της ολικής προσομοίωσης.



Εικόνα 11: Αποτελέσματα ολικής προσομοίωσης 972 καταλυτικών σφαιρών με τη μέθοδο DEM

Τα αποτελέσματα της ολικής προσομοίωσης λόγω του μεγάλου αριθμού σφαιρών παρατηρήθηκε μεγάλη εισχώρηση του ενός σωματιδίου στο άλλο. Για την αντιμετώπιση του προβλήματος αυτού, πραγματοποιήθηκε μια δεύτερη σειρά διαδοχικών προσομοιώσεων. Σε κάθε σενάριο, οι σφαίρες εισάγονταν στον υπολογιστικό χώρο ανά 100 ή 200 χρονικά βήμα και ο αριθμός τους για κάθε προσομοίωση δεν ξεπερνούσε τις 100 σφαίρες συνολικά. Τα αποτελέσματα της μιας προσομοίωσης εισέρχονταν ως δεδομένα εισόδου για την επόμενη προσομοίωση και διατηρούσαν την σταθερή υπολογισμένη θέση τους από την προηγούμενη προσομοίωση. Με την τεχνική αυτή, λιγότερες σφαίρες με πιο ακριβή αποτελέσματα για τη θέση τους και μικρότερη εισχώρηση προσομοιώθηκαν. Η μεγαλύτερη τιμή επικάλυψης που παρατηρήθηκε ήταν ίση με 0,033 χιλιοστά. Παρότι το πρόβλημα εισχώρησης της μιας σφαίρας μέσα στην άλλη ελαττώθηκε, η τιμή της διαπερατότητας του αντιδραστήρα επηρεάστηκε. Οι σφαίρες που εισερχόντουσαν ως δεδομένα εισόδου στη μια προσομοίωση συμπεριφέρονταν ως ακίνητα σωματίδια δηλαδή ως τοίχος πάνω στον οποίο προσκρούονταν τα επόμενα σωματίδια. Αυτό είχε ως αποτέλεσμα να μην αλλάζει η θέση των ακίνητων σωματιδίων (μη δυναμική συμπεριφορά) με την κρούση και επομένως παρουσιάσθηκε μια νέα διάταξη των σωματιδίων μέσα στον αντιδραστήρα. Το σύνολο των σφαιρών που κάλυψαν τις διαστάσεις του αντιδραστήρα ήταν ίσο με 775 σφαίρες αλλάζοντας έτσι την τιμή διαπερατότητας. Η Εικόνα 12 παρουσιάζει τα αποτελέσματα των διαδοχικών προσομοιώσεων.

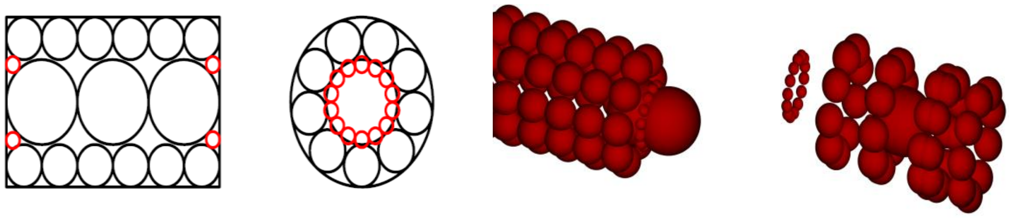


Εικόνα 12: Αποτελέσματα διαδοχικών προσομοιώσεων

Για τις διαδοχικές προσομοιώσεις χρησιμοποιήθηκαν οι ίδιες τιμές παραμέτρων με μόνη διαφορά τη χρήση του αραιού πλέγματος για τον τοίχο κυλινδρικής επιφάνειας, λόγω του προβλήματος των "προσκολλημένων σφαιρών". Επιπλέον στις διαδοχικές προσομοιώσεις, το πρόβλημα των "προσκολλημένων σφαιρών" αντιμετωπίστηκε εύκολα, καθώς οι προσκολλημένες σφαίρες στο πλέγμα αφαιρούνταν από τα δεδομένα εισόδου (ακίνητες σφαίρες) και εισέρχονταν ως ενεργές σφαίρες στην επόμενη προσομοίωση.

Για τις προσομοιώσεις με κυλινδρικούς καταλύτες, χρησιμοποιήθηκε η τεχνική σύνδεσης σωματιδίων. Οι συνδεδεμένες σφαίρες εισήχθησαν στον υπολογιστικό χώρο μέσω των αρχείων κειμένου με διαμόρφωση .geo. Η αναπαράσταση των κυλίνδρων έγινε αρχικά με τη σύνδεση δυο σφαιρών ακτίνας 5 χιλιοστών. Σύμφωνα με την Εξίσωση 6, 324 ζευγάρια σφαιρών εισήλθαν στον υπολογιστικό χώρο σε διαφορετικά χρονικά βήματα. Η μάζα των σφαιρών τέθηκε ίση με 1 και η αρχική τους ταχύτητα ίση με 1000 mm/s στην αρνητική διεύθυνση του άξονα ύψους z. Οι τιμές των παραμέτρων για τις σφαίρες που δεν ανήκαν στο ζευγάρι συνδέσμου παρέμειναν ίδιες όπως στον κώδικα για τους σφαιρικούς καταλύτες. Οι αλληλεπιδράσεις μεταξύ των συνδεδεμένων σφαιρών περιγράφηκαν από το μοντέλο NRotBondPrms όπου ο συντελεστής ελαστικότητας ή δυσκαμψίας τέθηκε ίσος με 10^7 και η απόσταση κατάργησης συνδέσμου ίση με 500. Η τιμή του συντελεστή ελαστικότητας ή δυσκαμψίας που περιέγραφε επαρκώς το φαινόμενο βρέθηκε ίση με 10^9 αλλά για την τιμή αυτή, τον αριθμό των σφαιρών και το χρονικό βήμα δεν επήλθε ευστάθεια στη λύση. Διαφορετικά σενάρια για την επίδραση των δυνάμεων απόσβεσης απέδειξαν ότι το χρονικό διάστημα μεταξύ εισαγωγής δύο σφαιρών παίζει σημαντικό ρόλο. Το τελικό κείμενο περιείχε τον όρο της απόσβεσης και οι σφαίρες εισέρχονταν στον υπολογιστικό χώρο κάθε 10.000 χρονικά βήματα.

Η καλύτερη αναπαράσταση της γεωμετρίας του κυλίνδρου εξαρτάται από τον αριθμό και επομένως και το μέγεθος των σφαιρών. Η αναπαράσταση με τρεις σφαίρες αποδείχθηκε ανεπαρκής διότι οι σφαίρες δεν συμπεριφέρονταν ως ένα στερεό σώμα, αλλά περιστρέφονταν η μια γύρω από την άλλη χωρίς να έχουν σταθερά σημεία επαφής. Για την καλύτερη αναπαράσταση των κυλίνδρων σχεδιάστηκε στο Autocad , μια διάταξη σφαιρών που περιείχε τον ελάχιστο αριθμό σφαιρών για μια επαρκή πρόβλεψη των σημείων επαφής κυλίνδρου. Η Εικόνα 13 παρουσιάζει τη διάταξη των σφαιρών και τα αποτελέσματα από την προσομοίωση αυτής της γεωμετρίας.

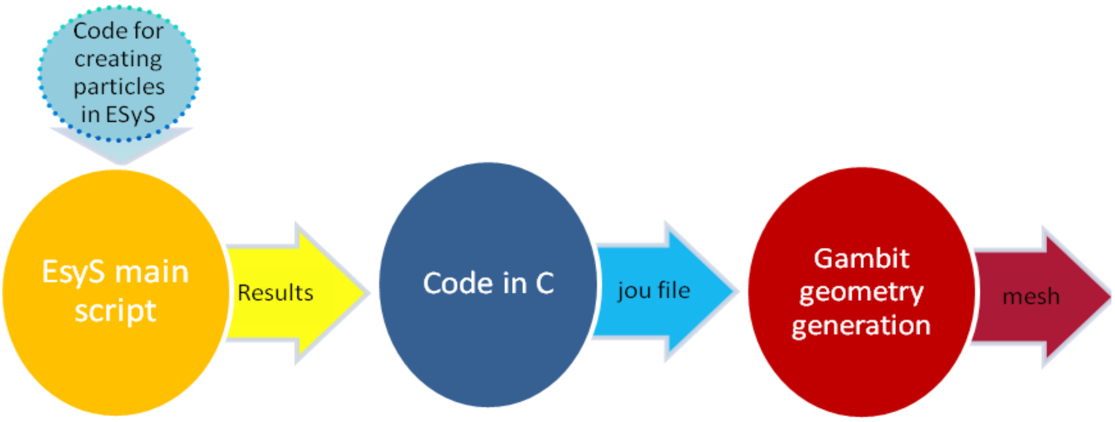


Εικόνα 13: Διάταξη σφαιρών για την αναπαράσταση κυλινδρικού καταλύτη και τα αποτελέσματα προσομοίωσης

Όπως παρουσιάζεται στην Εικόνα 13, η γεωμετρία αυτή παραμορφώνεται μετά την κρούση των συνδεδεμένων σφαιρών με το έδαφος. Η παραμόρφωση αυτή οφείλεται στο γεγονός ότι διαφορετικές δυνάμεις ασκούνται στις σφαίρες με διάφορες τιμές μαζών και ακτινών. Τα συμπεράσματα που προκύπτουν είναι ότι η τεχνική σύνδεσης σφαιρών πρέπει να εφαρμόζεται σε εφαπτόμενες σφαίρες όπου οι δυνάμεις συγκράτησης θα πρέπει να είναι αντιδιαμετρικές και περίπου ίσες. Οι διαφορετικές ακτίνες προσδίδουν διαφορετικές τιμές μαζών λόγω της σταθερής τιμής της πυκνότητας και επομένως καταλήγουν σε άνισες δυνάμεις. Διαφορετικές τιμές πυκνότητας για την απόκτηση σταθερής τιμής μάζας προδίδουν τη μοντελοποίηση του σωματιδίου από διαφορετικά κοκκώδη υλικά και δεν συνίστανται.

Η αυτόματη σχεδίαση του αντιδραστήρα και των διατεταγμένων σωματιδίων μέσα σε αυτόν πραγματοποιήθηκε μέσω της δημιουργίας ενός κώδικα που συνδέει τα αποτελέσματα της DEM μεθόδου και του σχεδιαστικού προγράμματος Gambit. Το αρχείο δεδομένων που εξάγει το λογισμικό Gambit και ονομάζεται journal file (jou file) περιέχει μια διαδοχική λίστα εντολών για τη σχεδίαση της γεωμετρίας, του πλέγματος και όλων των εργαλείων που διαθέτει. Το Gambit παρέχει τη δυνατότητα με το τρέξιμο αυτού του αρχείου την αυτόματη επανάληψη της διαδικασίας σχεδίασης, όπως αυτή έγινε σε μια προηγούμενη εφαρμογή.

Έχοντας αντιγράψει τις απαραίτητες εντολές σχεδίασης για τη δημιουργία του καταλυτικού αντιδραστήρα, διαφορετικοί κώδικες στην C γλώσσα προγραμματισμού δημιουργήθηκαν για την ένταξη των εντολών και δεδομένων στο (jou file). Με τη δημιουργία του αρχείου αυτού, η γεωμετρία που προσομοιώνεται μέσω της DEM μεθόδου μπορεί να σχεδιασθεί αυτόματα. Η Εικόνα 14 παρουσιάζει τη διαδικασία αυτόματης σχεδίασης της γεωμετρίας.


Code for creating particles in ESyS
EsyS main script
Results
Code in C
jou file
Gambit geometry generation
mesh

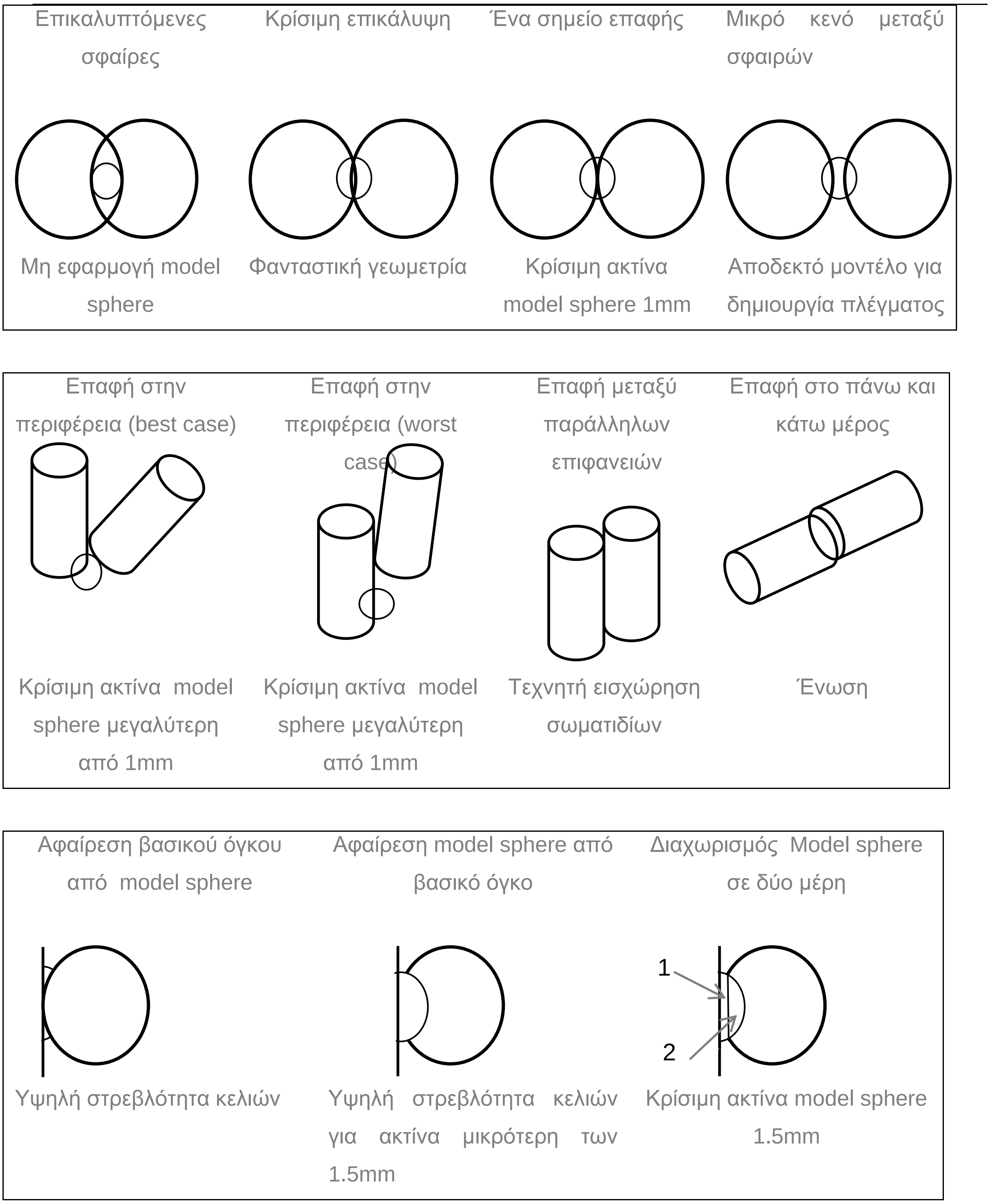
Εικόνα 14: Διαδικασία αυτόματης σχεδίασης της γεωμετρίας του καταλυτικού αντιδραστήρα

Η διαδικασία επεξεργασίας και ενσωμάτωσης των αποτελεσμάτων του ESyS-Particle μέσα στις σχεδιαστικές εντολές παρατίθεται στο Κεφάλαιο 3 της παρούσας διπλωματικής καθώς και στα αντίστοιχα παραρτήματα, όπου διατίθεται ολόκληρος ο κώδικας. Ενδεικτικά αναφέρεται ότι οι πληροφορίες από τα αρχεία δεδομένων του ESyS-Particle διαβάζονται από τον κώδικα και κατηγοριοποιούνται σε πίνακες. Στη συνέχεια ακολουθεί η μαθηματική επεξεργασία των δεδομένων για τον προσδιορισμό ανυσμάτων και θέσεων στον τρισδιάστατο χώρο. Τα αποτελέσματα από τη μαθηματική επεξεργασία διαβιβάζονται ανάμεσα στις εντολές σχεδίασης και με τη σειρά τους αυτές καταγράφονται στο ζητούμενο αρχείο κειμένου με διαμόρφωση jou.

Οι κώδικες στην γλώσσα C διαφέρουν για κυλινδρικά και σφαιρικά σωματίδια αλλά και από την στρατηγική προσομοίωσης. Για την ευελιξία του χρήστη, διατίθενται δύο σενάρια. Το Σενάριο Α περιέχει τους στερεούς όγκους των σωματιδίων ενώ το Σενάριο Β περιέχει μόνο τον εναπομείναντα όγκο του αντιδραστήρα με την αφαίρεση των όγκων των σωματιδίων. Τα διαφορετικά σενάρια επιτρέπουν διαφορετική μοντελοποίηση ανάλογα με τις απαιτήσεις του χρήστη.

Η αυτόματη σχεδίαση της γεωμετρίας δεν είναι αρκετή για τις CFD προσομοιώσεις του αντιδραστήρα, καθώς το πρόβλημα με τη διακριτή μέθοδο έγκειται στη σχεδίαση του πλέγματος. Για την αντιμετώπιση του προβλήματος των υψηλά στρεβλών κελιών πλέγματος στα σημεία επαφής, ενσωματώθηκαν στον κώδικα τεχνικές δημιουργίας πλέγματος. Το μοντέλο που χρησιμοποιήθηκε από τους πιο πολλούς ερευνητές για τη δημιουργία πλέγματος ήταν το μοντέλο “near-miss” εκτός των περιπτώσεων επικαλυπτόμενων σφαιρών ή την απουσία σημείων επαφής. Στην παρούσα διπλωματική εξετάζεται το μοντέλο “near-miss” (τεχνική 1), καθώς και η δυνατότητα εισαγωγής μικρότερων σφαιρών στα σημεία επαφής των εφαπτόμενων σωματιδίων (τεχνική 2).

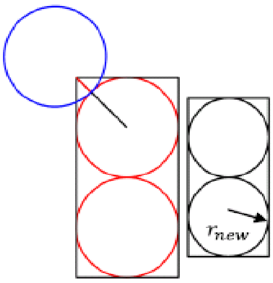
Για την ανάπτυξη της τεχνικής 2, η οποία αναφέρεται ως model sphere, πραγματοποιήθηκε μια ενδελεχής έρευνα για τον προσδιορισμό της θέσης και της ακτίνας των σφαιρών στα σημεία επαφής. Στο σχεδιαστικό πρόγραμμα Gambit, όταν ένας όγκος αφαιρείται, ενώνεται ή χωρίζεται από έναν άλλο όγκο, προκύπτουν φανταστικοί όγκοι ή επιφάνειες ή ακμές. Για την αποφυγή του προβλήματος αυτού, δόθηκε ιδιαίτερη προσοχή στη σχεδίαση των μοντέλων σφαίρες στα σημεία επαφής. Τα συμπεράσματα της έρευνας και οι απαραίτητες τεχνικές για τη δημιουργία πλέγματος με το μοντέλο σφαιρών περιλαμβάνονται στο κεφάλαιο 3 και δίνονται συνοπτικά στο σημείο αυτό από την Εικόνα 15.


Επικαλυπτόμενες σφαίρες
Κρίσιμη επικάλυψη
Ένα σημείο επαφής
Μικρό κενό μεταξύ σφαιρών
Μη εφαρμογή model sphere
Φανταστική γεωμετρία
Κρίσιμη ακτίνα model sphere 1mm
Αποδεκτό μοντέλο για δημιουργία πλέγματος
Επαφή στην περιφέρεια (best case)
Επαφή στην περιφέρεια (worst case)
Επαφή μεταξύ παράλληλων επιφανειών
Επαφή στο πάνω και κάτω μέρος
Κρίσιμη ακτίνα model sphere μεγαλύτερη από 1mm
Κρίσιμη ακτίνα model sphere μεγαλύτερη από 1mm
Τεχνητή εισχώρηση σωματιδίων
Ένωση
Αφαίρεση βασικού όγκου από model sphere
Αφαίρεση model sphere από βασικό όγκο
Διαχωρισμός Model sphere σε δύο μέρη
1
2
Υψηλή στρεβλότητα κελιών
Υψηλή στρεβλότητα κελιών για ακτίνα μικρότερη των 1.5mm
Κρίσιμη ακτίνα model sphere 1.5mm

Εικόνα15: Τεχνική Model sphere για την επαφή σφαιρών, κυλίνδρων και τοίχου

Για τις προσομοιώσεις που έγιναν με κυλινδρικά καταλυτικά σωματίδια και την τεχνική σύνδεσης δυο σφαιρών, χρειάστηκε να μειωθεί η ακτίνα των σφαιρών ούτως ώστε να ληφθεί υπόψιν η περίπτωση της εισχώρησης του ενός κυλίνδρου (ζευγαριού σφαιρών) στον άλλον, όπως φαίνεται στην Εικόνα 16. Η νέα τιμή της ακτίνας υπολογίσθηκε σύμφωνα με την

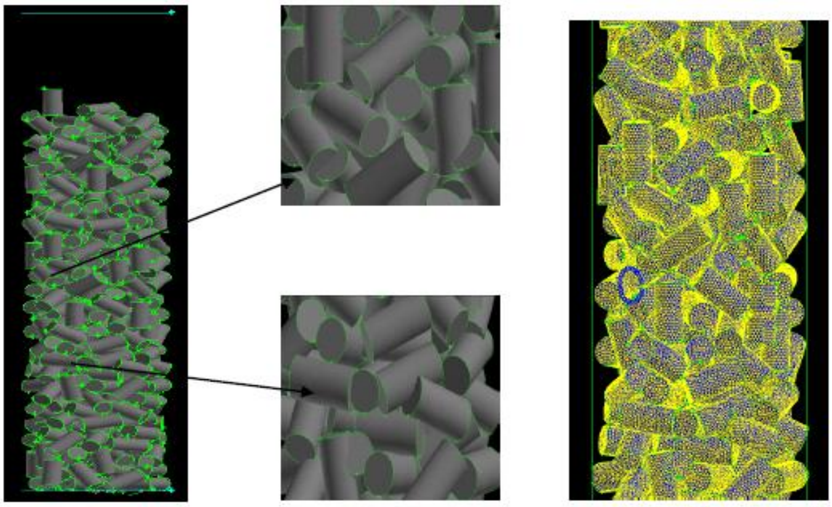
Εξίσωση 9, που δίνεται στην Εικόνα 16. Η τιμή του ύψους προσδιορίστηκε από την πρόσθεση των τεσσάρων ακτινών. Με τη μείωση αυτή των ακτινών πολλά σημεία επαφής μεταξύ των κυλίνδρων εξαλείφθηκαν. Συμπερασματικά περισσότερες σφαίρες απαιτούνται για την αναπαράσταση ενός κυλίνδρου, ώστε να επιτυγχάνεται ακριβέστερη πρόβλεψη των σημείων επαφής.



$$r_{new} = \frac{3-\sqrt{2}}{2} \cdot r \qquad (mm)$$ Εξίσωση 9

Εικόνα 16: Τροποποιήσεις στην αναπαράσταση ενός κυλίνδρου από δυο σφαίρες

Η Εικόνα 17 παρουσιάζει τα αποτελέσματα των προσομοιώσεων και την αυτόματα σχεδιασμένη γεωμετρία των κυλινδρικών σωματιδίων στο σχεδιαστικό πρόγραμμα Gambit. Τα αποτελέσματα δεν απέδωσαν τη ζητούμενη διαπερατότητα καθώς η διάταξη τους προδίδει ότι ο αντιδραστήρας χωρά περισσότερους από 324 κυλίνδρους. Αυτό επιβεβαιώνεται και από το γεγονός ότι για τον ίδιο αντιδραστήρα χρειάστηκαν 972 σφαίρες ίδιας ακτίνας και όχι 324 ζευγάρια αυτών. Το πρόβλημα έγκειται στον χαρακτηρισμό των δεσμών μεταξύ των σφαιρών. Κατά την προσομοίωση παρατηρήθηκε ότι οι αλληλεπιδράσεις για συνδεδεμένες σφαίρες δεν αφαρμόζονταν σε ένα σταθερό σημείο επαφής, αλλά επιτρεπόταν η ολίσθηση της μιας πάνω στην άλλη. Κατά αυτόν τον τρόπο οι σφαίρες έτειναν να πάρουν τη βέλτιστη θέση ανάμεσα στις άλλες διατεταγμένες θέσεις χωρίς να συμπεριφέρονται ως μια άκαμπτη μάζα κυλίνδρου, με αποτέλεσμα τον επηρεασμό της τιμής διαπερατότητας. Παρά τη μείωση των ακτινών των σφαιρών, σύμφωνα με την Εξίσωση 9 , το πρόβλημα της εισχώρησης του ενός σωματιδίου μέσα στο άλλο ήταν έντονο τόσο στις συνδεμένες σφαίρες όσο και στους κυλίνδρους μεταξύ τους. Το πρόβλημα εισχώρησης παρεμπόδισε τη δημιουργία πλέγματος καθώς κατά τη διαδικασία αφαίρεσης όγκων, μικρές επιφάνειες και ακμές δημιουργήθηκαν από την επικάλυψη του ενός σωματίδου από το άλλο, όπως τονίζεται στην Εικόνα 17.

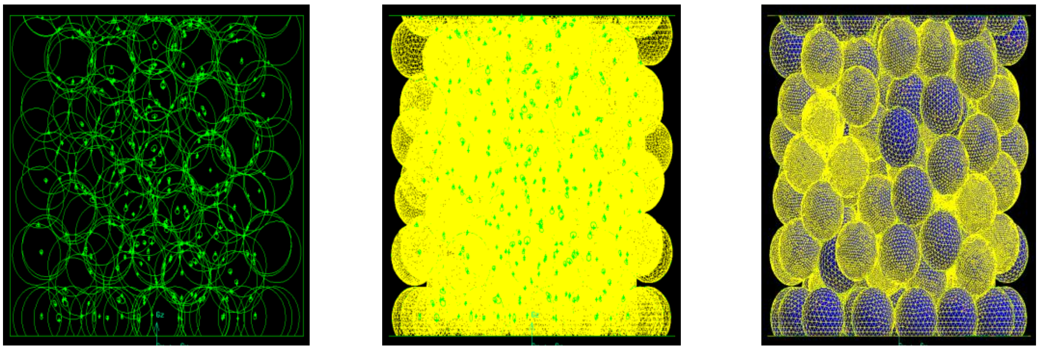


Εικόνα 17: Επικαλυπτόμενοι κύλινδροι και αποτυχία δημιουργίας πλέγματος και υπολογισμού απαιτούμενης διαπερατότητας

Μετά την ολική προσομοίωση των κυλίνδρων, η τεχνική των διαδοχικών προσομοιώσεων για ένα τμήμα του αντιδραστήρα ύψους 50 χιλιοστών χρησιμοποιήθηκε για την αντιμετώπιση του προβλήματος επικάλυψης. Η απαιτούμενη διαπερατότητα για το τμήμα αυτό δεν ήταν δυνατόν να επιτευχθεί και το πρόβλημα επικάλυψης παρέμεινε. Συμπερασματικά δεν ήταν δυνατή η δημιουργία πλέγματος για τη γεωμετρία του αντιδραστήρα με κυλινδρικά σωματίδια αναπαριστώμενα με δύο σφαίρες.

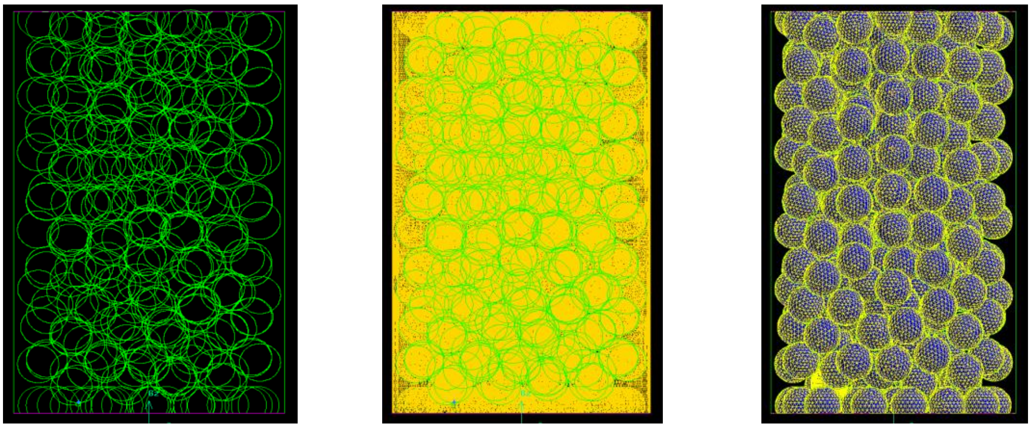
Για τις προσομοιώσεις που πραγματοποιήθηκαν με κυλινδρικά σωματίδια χρησιμοποιήθηκαν τρεις τεχνικές για τη δημιουργία πλέγματος· αυτή με επικαλυπτόμενες σφαίρες, αυτή με μειωμένη ακτίνα και αυτή με μοντέλα σφαιριδίων στα σημεία επαφής. Σε όλες τις προσομοιώσεις για την επίτευξη δημιουργίας πλέγματος, η ακτίνα του αντιδραστήρα αυξήθηκε κατά 0,5 χιλιοστά και για τις εφαπτόμενες με το κυλινδρικό τοίχωμα σφαίρες χρησιμοποιήθηκε το μοντέλο σφαιριδίων στα σημεία επαφής.

Για την περίπτωση των επικαλυπτόμενων σφαιρών (Mesh 1), δεν υπήρξε καμία αλλαγή στον κώδικα στην τιμή της ακτίνας των σφαιρών και τα αποτελέσματα της DEM μεθόδου χρησιμοποιήθηκαν αυτούσια χωρίς τροποποιήσεις. Τα αποτελέσματα αποδεικνύουν ότι δεν μπορεί να δημιουργηθεί πλέγμα ούτε για το Σενάριο Α, όπου προσομοιώνονται οι όγκοι των σωματιδίων ούτε για το Σενάριο Β όπου αμελούνται. Ο λόγος έγκειται στη δημιουργία μικρών κυκλικών πλευρών και επιφανειών, όπου η δημιουργία κόμβων καθίσταται αδύνατη. Με την αύξηση της ακτίνας και επομένως της επικάλυψης μεταξύ σφαιρών, προκύπτουν πολλά κελιά με υψηλή στρεβλότητα καθώς και φανταστικές πλευρές με τη χρήση του πρώτου Σεναρίου. Η Εικόνα 18 παρουσιάζει τα αποτελέσματα των προσομοιώσεων όπου διακρίνονται οι μικρές επιφάνειες που έχουν δημιουργηθεί με την επικάλυψη-εισχώρηση της μιας σφαίρας μέσα στην άλλη.



Εικόνα 18: Δημιουργία γεωμετρίας και πλέγματος για επικαλυπτόμενες σφαίρες

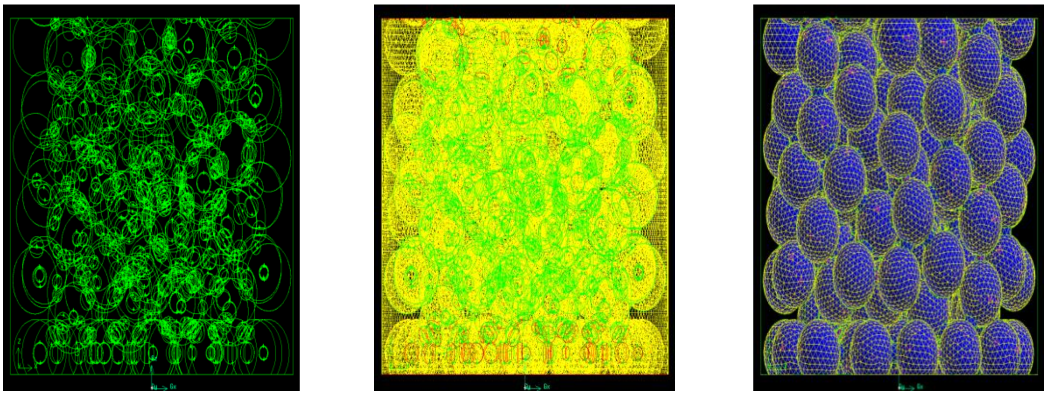
Για τις προσομοιώσεις με μειωμένη ακτίνα σφαιρών (Mesh 2), χρειάστηκε ο υπολογισμός της μέγιστης εισχώρησης που παρατηρήθηκε στα αποτελέσματα της ολικής προσομοίωσης 972 σφαιρών με τη μέθοδο DEM. Η μείωση της ακτίνας προσδιορίστηκε σύμφωνα με τη μέγιστη επικάλυψη έτσι ώστε να μην υπάρχει καμία σφαίρα που θα επικαλύπτει την άλλη και να αποφευχθεί η δημιουργία μικρών κυκλικών πλευρών και επιφανειών. Με τη μείωση αυτή πολλά σημεία επαφής χάνονται και η έρευνα αυτή μπορεί να θεωρηθεί παρόμοια με αυτή των (Jafari et al. 2008), όπου οι σφαίρες δεν είχαν καθόλου επαφή μεταξύ τους. Η ακτίνα των σφαιρών τέθηκε ίση με 4,64 χιλιοστά ενώ για τη δημιουργία του πλέγματος ο αντιδραστήρας χωρίσθηκε σε τρία μέρη των 100 χιλιοστών. Μόνο για τη δημιουργία πλέγματος στον κενό όγκο του αντιδραστήρα (Σενάριο Β), 2.611.413 κελιά δημιουργήθηκαν με μέγεθος διαστήματος ίσο με 1. Μεταξύ αυτών προέκυψαν 7 υψηλά στρεβλά κελιά στις σφαίρες με τη μέγιστη επικάλυψη , τα οποία μπορούν να εξαφανισθούν με την εισαγωγή ενός μικρού σφαιριδίου ανάμεσα τους (model sphere). Για μέγεθος διαστήματος ίσο με 2 προέκυψαν 463.422 κελιά εκ των οποίων 8 κελιά είχαν υψηλή στρεβλότητα και βρίσκονταν στην κορυφή του αντιδραστήρα. Η Εικόνα 19 παρουσιάζει τα αποτελέσματα της προσομοίωσης καταλυτικών σφαιρών με μειωμένη ακτίνα.



Εικόνα 19 : Δημιουργία γεωμετρίας και πλέγματος για σφαίρες με μειωμένη ακτίνα.

Για τις προσομοιώσεις όπου έγινε χρήση της τεχνικής εισαγωγής σφαιριδίων στα σημεία επαφής (model sphere), χρησιμοποιήθηκαν σφαιρίδια ακτίνας 1 χιλιοστού για τις εφαπτόμενες σφαίρες και 1,5 χιλιοστού για τις σφαίρες που εισέρχονταν η μία μέσα στην άλλη (Mesh 3). Ο επιπλέον εισαγόμενος στερεός όγκος με την εισαγωγή των σφαιριδίων είναι ίσος με 1,59 mm^3 για τα σφαιρίδια ακτίνας 1,5 χιλιοστού και 0.623 mm^3 για τα σφαιρίδια ακτίνας 1 χιλιοστού. Η εισαγωγή των όγκων των σφαιριδίων επηρεάζει και τη διαπερατότητα του αντιδραστήρα.

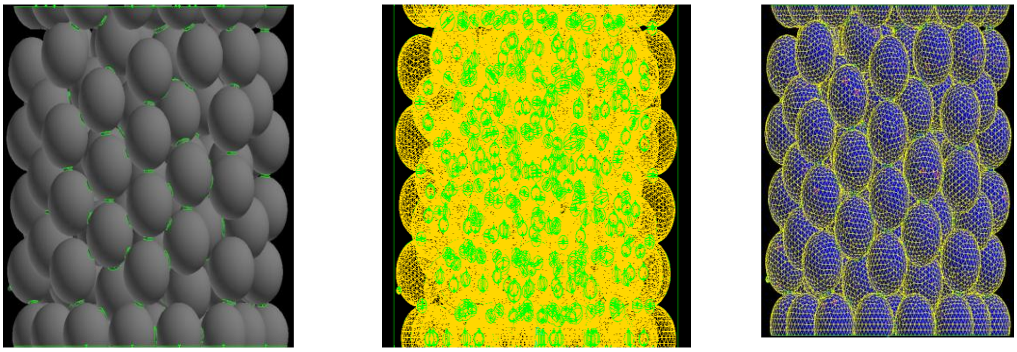
Πλέγμα για ολόκληρο τον αντιδραστήρα δεν μπόρεσε να δημιουργηθεί με τη τεχνική εισαγωγής σφαιριδίων (model sphere). Ο λόγος έγκειται στο γεγονός ότι η γεωμετρία πρέπει να χωριστεί σε μικρότερα τμήματα όπως και στις προηγούμενες τεχνικές. Ο διαχωρισμός αυτός σε όποιο ύψος του αντιδραστήρα και αν λάβει χώρα συναντά τους όγκους των σφαιριδίων και τα διαχωρίζει σε μικρότερα δημιουργώντας επιφάνειες που δεν μπορεί να δημιουργηθεί πλέγμα. Για το λόγο αυτό η τεχνική εφαρμόζεται σε ένα περιοδικό τμήμα του αντιδραστήρα όπως θα παρουσιασθεί παρακάτω (Mesh 5). Τα αποτελέσματα από την αυτόματη σχεδίαση της γεωμετρίας σφαιρών, σφαιριδίων και αντιδραστήρα παρουσιάζεται στην Εικόνα 20.



Εικόνα 20: Αποτελέσματα τεχνικής εισαγωγής σφαιριδίων σε όλο τον αντιδραστήρα

Για τις διαδοχικές προσομοιώσεις που πραγματοποιήθηκαν για την αντιμετώπιση του προβλήματος επικάλυψης, η μέγιστη εισχώρηση-επικάλυψη ήταν ίση με 0,33 χιλιοστά. Για τη σχεδίαση της γεωμετρίας και του πλέγματος, η ακτίνα των σφαιρών μειώθηκε σύμφωνα με τη μέγιστη επικάλυψη και σύμφωνα με το “near-miss” model κατά 1%. Ενδεικτικό σε αυτό το σημείο είναι ότι το μοντέλο αυτό χρησιμοποιήθηκε για προσομοιώσεις 44 σφαιρών με πηλίκο διαμέτρων ίσο με 2 ή 4 στις έρευνες των Nijemeisland and Dixon. Εδώ το μοντέλο χρησιμοποιήθηκε για 775 σφαίρες και πηλίκο διαμέτρων ίσο με 6. Η ακτίνα των σφαιρών τέθηκε ίση με r=4.9336 mm και η γεωμετρία του αντιδραστήρα χωρίστηκε σε τέσσερα μέρη στα ύψη at z=2, z=100 and z=200. Το πλέγμα περιείχε 2.894.520 κελιά εκ τω οποίων τα 68 είχαν υψηλή στρεβλότητα και βρίσκονταν στις θέσεις διαχωρισμού του αντιδραστήρα. Για τις διαδοχικές προσομοιώσεις, χρησιμοποιήθηκε και η τεχνική εισαγωγής σφαιριδίων με τα ίδια αποτελέσματα στρεβλών κελιών στις θέσεις διαχωρισμού.

Για την αντιμετώπιση των υψηλά στρεβλών κελιών στις θέσεις διαχωρισμού, πραγματοποιήθηκαν προσομοιώσεις για ένα μόνο περιοδικό τμήμα του αντιδραστήρα ύψους 50 χιλιοστών. Οι προσομοιώσεις στο ESyS-Particle περιείχαν δυο ίδια στρώματα σφαιρών στο πάνω και στο κάτω μέρος του τμήματος αυτού ούτως ώστε η γεωμετρία να είναι παρόμοια στα τμήματα εξόδου του ενός τμήματος και εισόδου του επόμενου. Σύμφωνα με τα κριτήρια διαπερατότητας για το τμήμα αυτό, 162 σφαίρες εισήχθησαν στην προσομοίωση και η μέγιστη εισχώρηση ήταν ίση με 0,215 χιλιοστά. Στον κώδικα η ακτίνα των σφαιρών τέθηκε ίση με 4,9 χιλιοστά και η τελική διαπερατότητα του τμήματος με την εισαγωγή σφαιριδίων βρέθηκε ίση με 0,48. Το πλέγμα που δημιουργήθηκε για το Σενάριο Α ήταν 952.852 κελιά ενώ για το Σενάριο Β ήταν 468.646 κελιά, εκ των οποίων κανένα δεν είχε υψηλή στρεβλότητα και το χειρότερο είχε στρεβλότητα ίση με 0,938108. Η Εικόνα 21 παρουσιάζει τα αποτελέσματα από το σχεδιασμό του περιοδικού τμήματος και τη δημιουργία πλέγματος σε αυτό. Ενδεικτικά αναφέρεται ότι η περιοδικότητα των τμημάτων που θα απαρτίζουν τον αντιδραστήρα επιτρέπει προσομοιώσεις αντιδραστήρων μεγαλύτερου ή μικρότερου ύψους με το ίδιο πηλίκο διαμέτρων χωρίς την επανάληψη των DEM προσομοιώσεων και του κώδικα σε C.



Εικόνα 21 : Αποτελέσματα προσομοιώσεων περιοδικού τμήματος

Η έρευνα που παρουσιάσθηκε στην παρούσα διπλωματική αποδεικνύει ότι η διακριτή μέθοδος για προσομοιώσεις αντιδραστήρων με σταθερά διατεταγμένα καταλυτικά σωματίδια. Οι θέσεις των σωματιδίων μπορούν να προβλεφθούν από τη μέθοδο DEM και μέσω ενός κώδικα μπορεί να δημιουργηθεί το αρχείο journal. Η γεωμετρία του αντιδραστήρα μπορεί να σχεδιασθεί αυτόματα, ενώ για τη δημιουργία ποιοτικού πλέγματος μπορεί να επιτευχθεί με το “near-miss” model για ολόκληρο τον αντιδραστήρα και με το “model sphere” για περιοδικά τμήματα.

Το πρόβλημα επικάλυψης σωματιδίων παρουσιάσθηκε σε όλες τις DEM προσομοιώσεις και μειώθηκε με τις διαδοχικές προσομοιώσεις λιγότερων σφαιρών. Επιπλέον το πρόβλημα των προσκολλημένων σφαιρών υπήρξε σε όλες τις προσομοιώσεις και συντέλεσε στο πρόβλημα επικάλυψης αλλά και στη διαφορετική διαπερατότητα του αντιδραστήρα. Ο προσδιορισμός του συντελεστή ελαστικότητας-δυσκαμψίας των σφαιρών και των τοίχων, της αύξησης του χρονικού βήματος, των μοντέλων αλληλεπιδράσεων και της πυκνότητας πλέγματος στη μέθοδο DEM παίζουν σημαντικό ρόλο στην ευστάθεια και το χρόνο της λύσης, καθώς και στην ποιότητα των αποτελεσμάτων.

Για τη δημιουργία πλέγματος στις προσομοιώσεις με σφαιρικούς καταλύτες, η ακτίνα των σφαιρών μειώθηκε σύμφωνα με τη μέγιστη εισχώρηση, ενώ οι τεχνικές “near-miss” model και “model sphere” παρουσιάζουν πιο ποιοτικό πλέγμα όσο μικρότερη είναι η εισχώρηση αυτή. Ο Πίνακας 1 παρουσιάζει συγκεντρωτικά τα αποτελέσματα των προσομοιώσεων με σφαίρες. Για τις προσομοιώσεις με κυλίνδρους δεν ήταν δυνατή η δημιουργία πλέγματος, λόγω του προβλήματος εισχώρησης. Πιο ακριβή αποτελέσματα μπορούν να αποκτηθούν με την αναπαράσταση των κυλίνδρων με περισσότερες σφαίρες στη μέθοδο DEM. Η αύξηση του αριθμού σφαιρών θα επιφέρει αύξηση του υπολογιστικού χρόνου και θα επηρεάσει την ευστάθεια της προσομοίωσης. Ακόμα θα μπορούσαν να χρησιμοποιηθεί το μοντέλο των περιστρεφόμενων σφαιρών με διαφορετικά μοντέλα αλληλεπιδράσεων στη σύνδεση σωματιδίων. Στην έρευνα που πραγματοποιήθηκε για την εισαγωγή των σφαιριδίων στα σημεία επαφής αποδείχθηκε πιο εύκολη η δημιουργία πλέγματος για κυλινδρικά σωματίδια σε σχέση με σφαιρικά· επομένως η δημιουργία πλέγματος έγκειται στην καλύτερη πρόβλεψη θέσεων της DEM μεθόδου.

Πίνακας 1: Σύνοψη αποτελεσμάτων για προσομοιώσεις με σφαιρικούς καταλύτες

Script	*Number of cells*	*Number of highly skewed elements*	*Skewness of worst Element*	*Worst element at the contact point*	*Interval Size of created mesh*	*Porosity*
Reduced Model B	2,611,413	7	0.999303	Yes	1	0.457
Reduced Model B	463,422	8	0.992768	No	2	0.457
Reduced Model A	5,798,162	7	0.999303	Yes	1	0.457
Reduced Model A	876,453	8	0.992768	No	2	0.457
Less spheres script B	2,894,520	68	0.996728	No	1	0.545
Periodic segment A	952,852	0	0.938108	Yes	1	0.48
Periodic segment B	468,646	0	0.938108	Yes	1	0.48

Συμπερασματικά , η δημιουργία πλέγματος μπορεί να γίνει με τις δυο προαναφερθείσες τεχνικές και τα αποτελέσματα των προσομοιώσεων θα πρέπει να συγκριθούν. Ειδικότερα για την τεχνική “model sphere” θα πρέπει να εξετασθεί η επιρροή των επιπλέον εισαγόμενων όγκων στο πεδίο ροής και στο θερμοδυναμικό πεδίο. Τέλος, μετά την παρουσίαση των υπαρχόντων μοντέλων για τη διακριτή μέθοδο οι προσομοιώσεις μπορούν να λάβουν χώρα με τον προσδιορισμό των οριακών συνθηκών.

Technische Universität Graz

National Technical University of Athens

CFD Simulations on fixed bed catalyst reactors

by

Efthymia Ioanna KOYTSOUMPA

Diploma thesis of studies of mechanical engineering written at the Technical University of Graz for the Institute of thermal engineering and submitted at the National Technical University of Athens.

Supervisors

Dipl.-Ing. Lorenz Griendl
Dr.-Ing. Sotiris Karellas
Univ.-Prof. Dr.-Ing. Jürgen Karl
Univ.-Prof. Dr.-Ing. Kakaras Emmanuel

Graz, May 2010

STATUTORY DECLARATION

I declare that I, Efthymia Ioanna Koytsoumpa , have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources

Graz, May 2010

Koytsoumpa Efthymia Ioanna

ABSTRACT

Title: CFD Simulations on fixed bed catalyst reactor

Author: Efthymia-Ioanna Koytsoumpa

1st keyword: CFD DEM
2nd keyword: methanation catalyst
3rd keyword: mesh

The huge energy demand, the production costs, the dependency of energy resources, the CO2 costs emissions and the global warming effect emerge the need of alternative solutions in the energy industry. The conversion of coal and biomass into Substitute Natural Gas (SNG) via the methanation process enhances the concept of local energy production fed into existing natural gas grids and thus reducing the dependency from gas resources. Computational Fluid Dynamics as a simulation tool permits the prediction, the designing and optimization of new products and processes, decreasing the need of expensive experimental equipment.

For the subject documented in this diploma thesis, the existing methods of CFD simulations of Methanation Catalysts have been investigated. An extended literature research has raised up- arisen the limitations and achievements of the two basic simulation methods, the porous media and the 3D representation of the geometric structure of the packing. For the accurate prediction of the porosity, the heat transfer, the reflection of local effects that control the efficiency and the stability of the reactor the 3D representation of the catalytic particles were chosen.

The limited literature and research have led to the very first steps of this simulation, the generation of a geometry where a mesh can be created. The Discrete Element Method was used to simulate the random packing of the particles. A code created in c programming language provides the opportunity to generate automatically the packed geometry to the pre-processing designing program Gambit.

Within the research for this diploma thesis a case study is provided in order to create a mesh able geometry. Furthermore implementations of models for the CFD simulation according to the literature research are suggested.

ACKNOWLEDGEMENT

I am particularly grateful to my supervisor Univ.-Prof. Dr.-Ing. Jürgen Karl,for giving me the opportunity to work in the Institute of thermal engineering in Technical University of Graz. I am honored with the opportunity given to me to write my diploma thesis under his supervision. I would also like to thank my supervisors Dr.-Ing. Sotiris Karellas and Univ.-Prof. Dr.-Ing. Kakaras Emmanuel for their cooperation and support on my diploma thesis.

This work was carried out under the guidance of Dipl.-Ing. Lorenz Griendl. I appreciate very much all the efforts and time he devoted helping me to reach the objectives of my work. Thanks to him and his extensive knowledge in CFD simulations, I had the chance to learn so many things and to fulfill the requirements of this thesis. I am more than grateful for his support and contribution to generation of C codes and to the crucial decisions which had to be made.

I would also like to thank Markus Gesslbauer for the installation of the software needed for the realization of the simulations as well as my working colleagues in the Institute of thermal engineering. Support for this work was also provided by the on line forum for ESyS- Particle which was enlightening for the DEM simulations performed.

Furthermore, I would like to thank my friends in Austria that supported me a lot during my stay here. Finally, I would like to mention my parents and my friends that supported and encouraged me through all of these years.

Graz, May 2010

Efthymia-Ioanna Koytsoumpa

Contents

Nomenclature

°C	degrees Celsius
C	constant
CH_4	methane
CO	carbon monoxide
CO_2	carbon dioxide
$\boldsymbol{F}_{c,ij}$	contact force between particles I and j (Nt)
$F_{cn,ij}$	contact force at the normal direction (Nt)
$F_{ct,ij}$	contact force at the tangential direction (Nt)
$\boldsymbol{F}_{d,ij}$	viscous contact damping force (Nt)
$\boldsymbol{F}_{dn,ij}$	viscous contact damping force at the normal direction (Nt)
$\boldsymbol{F}_{dt,ij}$	viscous contact damping force at the tangential direction (Nt)
$\boldsymbol{F}_i$	forces acting on the particle (Nt)
f_i	body forces acting on the fluid
$\boldsymbol{g}$	gravity acceleration $\left(\frac{m}{s^2}\right)$
H	height of the reactor (m)
H_2	hydrogen
h	specific enthalpy $\left(\frac{kJ}{kg}\right)$
h_w	wall heat transfer coefficient
K	Kelvin
K	factor of elastic stiffness
K_{max}	maximum factor of elastic stiffness
K_{wall}	factor of wall’s elastic stiffness
k	kinetic energy (kJ)
k_r	effective radial thermal conductivity
k_f	effective fluid thermal conductivity
m_i	mass of particle i (kg)
N	tube to particle ratio
Nu_w	Nusselt number at the wall
Nu	Nusselt number
n	number of particles in the reactor
Pr	Prandtl number

p	local pressure (Pa)
R	radius of the reactor(m)
Re	Reynolds
$(R_i + R_j)$	sum of radii of particles in contact
r	radius of the particle (m)
$\|r_i - r_j\|$	distance of the centres of particles in contact (m)
Sc	Scmidt number
S^h	source of enthalpy generated on volumetric basis $\left(\frac{kJ}{s\cdot m^3}\right)$
S^m	source of mass generated on volumetric basis $\left(\frac{kg}{s\cdot m^3}\right)$
S_i^u	source of momentum generated on volumetric basis $\left(\frac{kg}{s\cdot m^3}\right)$
t	time (s)
u_i	fluid's velocity $\left(\frac{m}{s}\right)$
$V_{reactor}$	volume of the reactor (m^3)
$V_{particle}$	volume of the particle (m^3)
v	relative velocity of two particles $\left(\frac{m}{s}\right)$
$\boldsymbol{v}_i$	velocity of particle I $\left(\frac{m}{s}\right)$
x_j	Cartesian coordinates (tensor notation)
y^+	dimensionless parameter defining the boundary layer and the mesh adequacy
$\rho\overline{u_i'u_j'}$	Reynolds stress terms
Δt	time increment (s)

Greek symbols

Γ_h	ratio of effective viscosity $\left(\frac{m^2}{s}\right)$
δ_{ij}	Kronecker Delta Function
δ^{ij}	displacement (m)
ε	porosity
ε	dissipation ratio
η_i	normal viscous contact damping coefficient of particle i
θ_e	angle of equiangular face or cell (°)
θ_{max}	maximum angle in face or cell (°)
θ_{min}	minimum angle in face or cell (°)

μ	fluid’s dynamic viscosity
μ_f	friction coefficient
ξ_{ij}	mutual compression of particles i and j or displacement
π	
ρ	fluid’s density $\left(\frac{kg}{s\cdot m^3}\right)$
Φ	dissipation function $\left(\frac{kJ}{s\cdot m^2}\right)$
φ	scalar variable
ω	specific dissipation ratio

1 Introduction

Methanation reactors have been used intensively the last years in the energy production field. Biomass and coal conversion to Substitute Natural Gas (SNG) via the processes of gasification and methanation is a promising technology. The limited oil resources, the wish for security of energy supply and the advantages of local energy production have forced the industry market to alternative technologies as the production of SNG.

The mathematical modelling of the steady and dynamic behaviour of fixed bed reactors has been studied extensively and many approaches for multiphase flows have been suggested. Over the past decades, Computational Fluid Dynamics (CFD) has been used for the visualised prediction of hydrodynamic and heat transfer behaviour of fixed bed reactors. For the CFD simulations of the two phase gas-solid methanation reactor, the one-continuum method and the discrete method are mostly used. The discrete method, although it is not as well documented as the one continuum method, allows the realistic representation of the flow pattern and does not make use of any porosity approximations. The quality and the size of the meshed geometry of the reactor has been a barrier for the use of the discrete method.

In the present thesis, a literature research of CFD simulations with discrete method and a comparison of the models of turbulence and heat transfer used are presented. According to the packing and the size of catalytic particles, the flow can vary from laminar to turbulent within the reactor. Small or large eddies, stagnant points or flows around the particles or near the contact points are significant characteristics of the specific geometry simulated. The local Nu number is influenced by the particle’s shape and position except from the dependency of the local flow and Re number. A good quality mesh cannot be obtained due to the highly skewed elements which occur at the contact points. The most common technique for meshing the complex geometry is the reduction of the particles’ radii according to the “near-miss” model.

Discrete Element Method (DEM) is a numerical method for computing the motion of individual particles that range in number and size by solving the Newton’s second law of motion. The method is used for the prediction of the randomly packed catalytic particles positions in the fixed bed. The parameters of DEM simulations are discussed and the simulations performed with ESyS-Particle DEM software are presented.

A C code which couples indirectly the DEM results with CFD method is used for the automatically generated geometry in the pre-processing CFD software Gambit. Except from the “near miss model”, the “model sphere” technique is presented for the mesh generation at the wall to particle contact and at the particle to particle contact. The two models can provide a mesh without highly skewed elements for both spherical and cylindrical catalytic particles when the contact points are accurately predicted from DEM simulations.

The results obtained from the ESyS-particle simulations are incorporated in the C code and the geometry with spherical and cylindrical catalytic particles is automatically generated. The mesh created and the porosity obtained from the generated packing are evaluated in the last chapter. In conclusion, the crucial parameters for the DEM simulations and the mesh generation are outlined for spherical and cylindrical particles.

.

2 Basics

The global warming effect and the pollution of the environment due to flue gas emissions and waste generation imply the urgent need of efficient and environmentally friendly technologies. The raising energy demands and the high costs of energy production due to the required limits of pollutant emissions and the limited resources have forced the industry market to turn to alternative technologies. The conversion of fossil fuels like coal and lignite as well as biomass to Substitute Natural Gas (SNG) by the gasification and methanation process is a promising technology. The gasification process with its versatility and efficiency is ideally suited to treat any wastes, byproducts and natural feed-stocks. Many researchers have concentrated on the idea of the conversion to SNG as the enhancement of the local energy production and reduction of the dependency of natural gas resources provide great potentials to the power generation. Especially the conversion of coal to SNG, due to availability of coal, adds value to coal reserves and the focus on the energy diversity allows the use of existing infrastructure of pipelines and combustion turbines.

2.1 Overview of the Conversion Processes

The basic steps for the production of SNG are the process of gasification, the process of cleaning gas and methanation. The fuel is inserted into the gasifier, where it reacts at approximately above 700°C and converts to high temperature crude gas. The gas proceeds to the removal of solids, tars and other impurities including the desulfurization process. The clean syngas, that mainly contains H_2/CO, continues to the catalytic reactor, where the methanation takes place. Figure 1 shows a scheme of the SNG production from biomass in a plant built in the Netherlands.

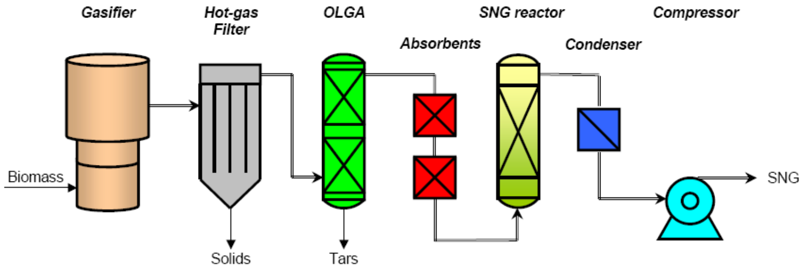


Figure 2 SNG production from biomass (Zwart R.W.R et al. 2006)

Gasification is an incomplete thermo-chemical reaction, where partial oxidation at high temperatures (over 700° C) takes place and results to the production of raw fuel syngas. The syngas is composed mainly from H_2, CO and CH_4 and its Lower Heating Value (LHV) may differ depending on the gasification process (air, oxygen, steam). Autothermal or allothermal gasification systems are used to convert the fuel to syngas, whereas higher H_2/CO ratios result from the allothermal process. Depending on the syngas treatment after the gasification, it can be used to produce SNG (methanation), electricity in combined cycle plants, hydrogen (gas shift reaction) or liquid bio-fuels (Fischer-Tropsch). After the cleaning of the syngas, the methanation process follows for the production of SNG. The purpose of the procedure is the methane enrichment of syngas and the increase of the calorific value. The clean SNG after the methanation can be fed with the optimum pressure to the natural gas grid and can be used for energy production. Figure 2 presents an overview of the steps and procedures that lead to the production of SNG after oxygen gasification.

The benefits from the production and use of SNG are not only a fuel that provides a calorific value similar to natural gas and can be fed to existing grids but also no efficiency losses compared to the conventional plants.

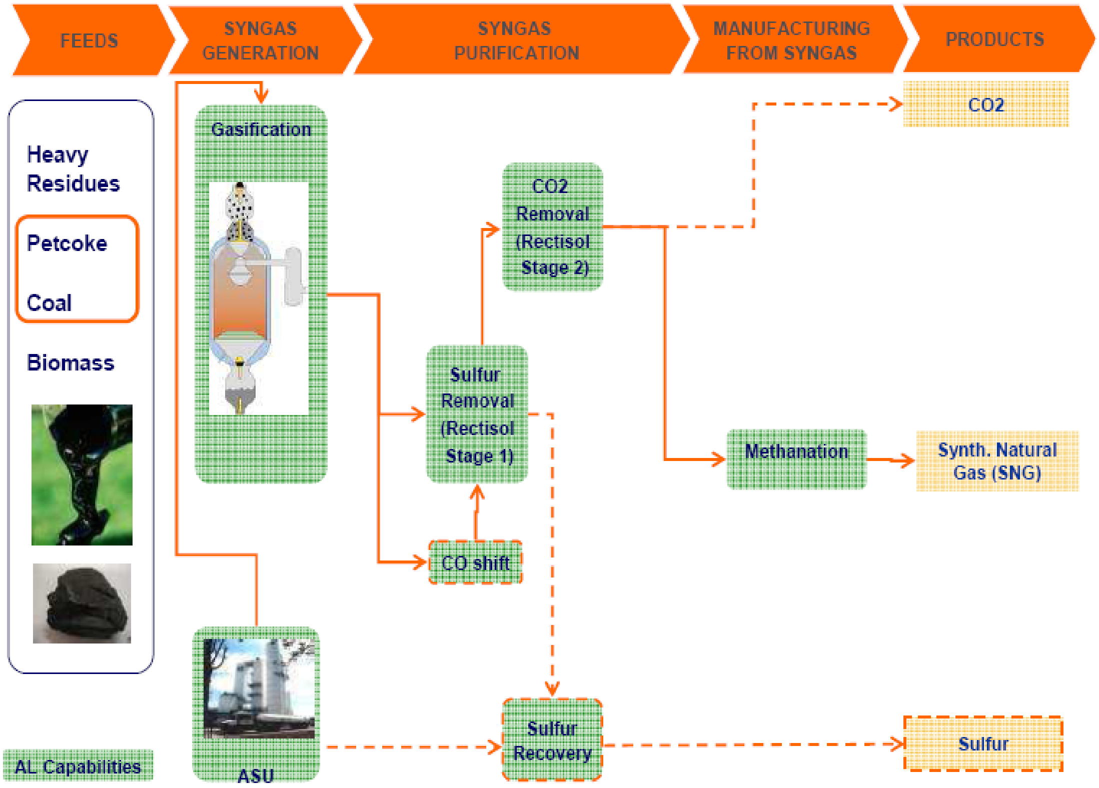


Figure 3: Overview of SNG from coal with oxygen gasification (Weiss M.et al 2008)

2.2 Methanation in catalyst reactors

Methanation is a catalytic procedure that leads to the transformation of syngas to methane. The scheme of methane synthesis consists of a multistage reaction mechanism. As a chemical reaction, methanation is highly exothermic and the temperature of the reactor, where the reaction takes place, imposes technological boundaries and requirements.

The efficiency of the methanation is based on the:

- Synthesis of the syngas (the H_2/CO ratio, the residual CO_2 content, the higher hydrocarbon content, the catalyst poisons)
- Optimal operating temperature and pressure
- Effect of steam on the catalyst activity and stability
  (Eisenlohr, Moeller & Dry 1974)

Currently there are two technological applications of methanation processes, the fixed bed methanation reactors and the fluidized beds. The fixed bed methanation reactors can work under pressure, are mostly used for SNG produced from fossil fuels, and as the heat removal plays an important role, they are usually connected in series with intermediate cooling or recycle of product gas.
Fluidised-bed reactors are known to be suitable for large-scale operations of heterogeneous catalytic reactions with high exothermicity. The heat and mass transfer is high, compared to fixed bed reactors. They offer the opportunity to easily remove, add and recycle the catalyst continuously during the process, but special attention should be paid to the attrition and entrainment of the catalyst particles.

In industrial applications the catalytic particles are very often applied in randomly packed beds. Although this packing is not the most efficient, it is used as standard packing of catalytic reactors due to its ease of use and low cost compared to fluidized bed reactors or structured fixed bed reactors.

The geometry of the catalytic particles varies depending on the application, the use, the chemical composition. For example the size of the particles may control the internal resistance to diffusion, the pressure drop in the reactor and the activity of the catalyst due to the active surface. The catalytic behavior is also influenced by the mechanical properties of the metal alloys and the heat transfer efficiency. The geometry of the reactor also plays an important role in combination with the particles' geometry as the velocities, the porosity, the temperature profile, the pressure drop are connected to the "particle to particle" and "particle to wall" correlations. A common unit representing the geometry of these correlations is the tube (reactor) to particle (catalyst) ratio :

$$N = \frac{diameter\ of\ the\ reactor\ tube}{diameter\ of\ the\ particle} = \frac{D}{d}$$ Equation 2

In summary, the fluid flows through the voids that are generated between the reactor and the particles. The fluid contains the reactant components that are transported to the particle's surface and then through catalyst pores where the chemical reaction takes place. The products of the reaction are again transferred to the bulk flow. Heat and mass dispersion take place through the reactor in radial and axial direction.

2.3 Simulating methods on fixed bed catalysts

Packed bed reactors are one of the most commonly used chemical processes for industrial processes, including high temperature reactions in shaft reactors, absorption processes for gas purification, filtration and heat recovery (Motlagh, Hashemabadi 2008, Romkes S.J.P et al. 2003). The reactor engineering characteristics have to be specified and some necessities for the design and construction of their products are required.

Experimental work, research and mathematical modelling of these reactors have been studied for many decades. The most important and controversial theme about modelling fixed bed reactors is the qualitative and quantitative description of fluid flow and heat transfer. The interactions between particle- to- particle, particle-to-fluid and particle-to-wall contribute to the long standing problem of predicting the transport phenomena in a fixed bed.

Various projects and researches have been realised but no exact correlations have been discovered and many discrepancies occur from one study to another. The modelling characteristics are based on physical principles and contain parameters from experiments. Theoretical studies also have been carried out, trying to implement models that have been determined from experiments. Especially these studies focus on small reactors or lab scale reactors before the procedure of up-scaling is conducted. The tube to particle diameter ratio is small (N <10) and few studies link local fluid flow to bed structure. For larger tube to particle ratios the method of one continuum is usually preferred due to the large amount of small particles and the complex geometry.

The basic principle of designing and scaling up a reactor was mainly based on experimental data. The increasing computational power has enabled the optimization of designing by implementing Computational Fluid Dynamics (CFD). The efficiency of process equipment can be predicted by CFD and validated with experiments.

2.3.1 Existing simulating methods

The operation, efficiency and life duration of a packed bed catalyst depend on the porosity and the tortuosity of the void space that means the position, orientation, size and shape of the particles situated inside the reactor. This complex geometry causes many problems to the simulation of two-phase simulation, not only to the basic flow pattern and simulating models but also to the generation of this specific geometry. More precisely the flow streamlines, the kinetics, the surface on which the chemical reactions take place, the mass and heat transfer models, the turbulence and boundary layers influence the results and the numerical diffusion in every position. Thus designing and simulating the full model of a catalyst reactor is a complicated task.

The mathematical modelling of the steady state and dynamic behaviour of fixed bed reactors have been studied excessively over the past decades (Wei James 1987, Dommeti, Balakotaiah & West 1999, Ioardanidis 2002).The equations of momentum, continuity, species and energy in the spatial domain and time may vary from differential equations to partial differential equations depending on assumptions and specific models used. The coupling between the scalar variables, the transport processes and the kinetics result in highly non linear equation models (Dommeti, Balakotaiah & West 1999). Several approaches for multiphase flows have already been suggested: the ‘diffusion’ model, the ‘sphere-pack’ model, the ‘percolation’ theory, the ‘porous media’ method, the ‘energy minimization’ approach. Most of these models dealt with gas–liquid concurrent down flow in 2D rectangular packed beds with a relatively large size of particles (3–6 mm) at steady state condition (Jiang et al. 2001).

Especially for the two phase gas-solid system, two principles have been established in CFD simulations, the 3D representation of the geometric structure of the packing (the discrete method and continuum) and the homogenization of all phases in one continuum based method (porous media). Table 1 shows the current state of the art of CFD simulations for fixed bed applications.

Table 1 : Principals of CFD simulations for flow through Packed Beds ((Joshi, Ranade 2003))

typical equipment	monoliths, fixed beds, packed columns, trickle-bed reactors, filters, etc.	
design issues/applications	pressure drop, mal-distribution and channeling, mixing and RTD, scale-up	
	Current Status	
knowledge of physics	lumped models with or without isotropic porosity are used reasonable results for single-phase flows; closure models suitable for multiphase flows through packed beds are not adequate	
CFD codes/design applicability	commercial CFD codes/models allow simulations of single-phase flow through porous media with isotropic or non-isotropic permeability and inertial resistance coefficients detailed modeling of a packed bed, which is very computation intensive, is being explored in recent studies capability of simulating multiphase flow through packed beds is almost nonexistent except via user routines based on empirically calibrated closure models used for designing equipment with single-phase flow through a porous medium use for designing multiphase flow through a packed bed is in the primitive stage	
	Path Forward	
limitation of current models	representing void space in packed beds: grid quality, computing resources closures for multiphase flow through packed beds: interfacial area, drag simulation of flow regimes/ transition and unsteady flows mixing and liquid dispersion: role of capillary forces/wetting	
	experiments	physical models
development needed for overcoming the limitations	measurements of porosity distribution (with different length scales) within a packed bed (mean, axially averaged, standard deviation, and so on) with quantification of the way of packing[62]	models to simulate different packing possibilities66 models to represent transient or low-*Re* turbulent flow through complex geometry with severe curvature
	pressure-drop measurements along with local turbulence characteristics[63]	
	local phase distributions and velocity measurements for gas-liquid flow through well-characterized packed bed	interphase closure models for estimating gas-solid, gas-liquid, and liquid-solid phases under different flow regimes[9] DNS or VOF simulations are needed to guide the development of closures[67]
	detailed measurements of unsteady flow regimes (quantification of key spatial and temporal scales)	quantitative representation of the role of wetting and hysteresis on contact angle
	experimental data (global as well as local measurements) on liquid dispersion, bypass, and channeling[6]	quantification of the role of gradients of porosity and capillary forces on liquid dispersion[8]

The Lattice Boltzmann method which solves the Boltzmann equations for colliding particles in Newtonian fluids is also an alternative method in early stages for CFD in fixed bed applications (Feng, Michaelides 2004, Mantle, Sederman & Gladden 2001).

2.3.2 The one-continuum method

When the number of catalyst particles is too large, the required computational power and time exceeds the limits of available computing resources. The treatment of the flow as a continuum leads to an alternative solution. For the gas-solid phase in fixed beds, the continuum is mostly represented by porous media. This means that the continuum is treated as media that incorporates porous and the fluid is flowing through this media. The temperature, the velocity, the mass and all other variables that characterized one position in the previous discussed complex geometry are now summed up and implemented as averaged values to the fluid continuum. The porous in the flow act like a resistance flow but represented as added momentum sinks. This one continuum model is the most common method used to simulate fixed bed applications in general. Empirical correlations for the radial or axial distribution of porosity, for the diffusivity of heat and mass transfer and all the relevant coefficients (conductivity and heat transfer coefficient) as well as approximations of turbulence are incorporated in this model.

The 1D plug-flow model or 1D pseudo-homogeneous model was first used where the concentration and temperature gradients were assumed only to occur in the axial direction and later it was extended with axial mixing and porosity correlations (Liu et al. 2008). This first model presents many inaccuracies and may only be used in case of negligible difference between the solid and fluid phase conditions and mild radial temperature and concentration profiles (Iordanidis 2002). The models which take into account the temperature and concentration of the fluid bulk flow and catalyst surface can be called heterogeneous models. However, when these models still account for one continuum, the effective coefficients are implemented to model the heat and mass dispersion as well as heat intra-particles resistances according to theoretical porosity distribution and many theoretical correlations are inserted.

The commercial CFD software FLUENT has the following features for modelling porous media (Fluent User’s Guide, 2003):

- The simulation can be performed in 2D or 3D but in the case of a methanation reactor a 2D approach is enough as usually the porosity is defined to be radial distributed or derived by the mean porosity and the averaged radial distribution in the longitudinal direction. As the reactor has usually the geometry of a cylinder, thus periodic boundaries do not necessitate a 3D simulation.

- The momentum sinks are defined by two coefficients, which represent the viscous and inertia loss in high velocity flows (in low velocities-laminar flows the inertia loss can be neglected). The two coefficients can be derived (either if superficial or physical velocity is used) through the known and commonly used Ergun's pressure drop semi-empirical equation or by an a priori known pressure drop or by inserting experimental velocity and pressure data.
- The fluid and solid thermal conductivity, isotropic or non-isotropic, have to be defined by the user in order to obtain the effective thermal conductivity of the medium which is proportional to the porosity distribution. In addition, the fluid and the material of the solid component should be defined.
- Source terms which represent the chemical reactions of the fluid with the solid particles for the calculation of heat transfer and are multiplied with the total volume of the cells or source terms of other scalar quantities can be included.
- The model of turbulence can be implemented either by the program’s default option, where the turbulence is treated as the solid medium had no effect (large permeability) or either by ignoring its effect on the fluid mixing and momentum and transport the turbulent quantities through the medium but not in combination (Fluent User’s Guide, 2003).

According to porous media method of CFD literature, all the above mentioned coefficients are derived from different equations according to the representation of the solid medium from cylinders or spheres. The research which has been done for the documentation of this diploma thesis was enlarged upon all the CFD simulations on fixed bed reactors including methanation and steam reforming reactors etc. Simulations in FLUENT using the continuum method can be found in (Yurong He, Thang Ngoc Cong & Yulong Ding 2006) and in (Takashi Takeuchi, Masahiko Aihara, Hitoshi Habuka). The most common approach when the method is extended in large industrial application is the negligence of axial transport. An analytical study of porous media in FLUENT with the implementation of transport model and without any empirical correlations for the Schmidt (Sc) and the Prandtl (Pr) numbers is presented in (Liu et al. 2008) and (Liu et al. 2006). Figure 3 presents the results of this work. The profiles of temperature distribution, of the effective conductivity and the turbulent thermal diffusivity are shown in the three pictures respectively.

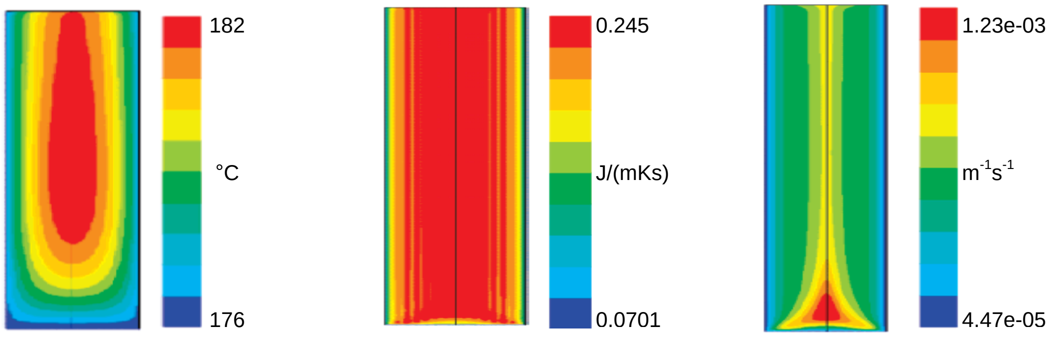


Figure 4 : Results from porous media simulation with a sophisticated transport model. The temperature profile, the effective conductivity and the turbulent thermal diffusivity along the axial direction (Liu,2008)

The interactions between viscous flow, transport and kinetics in synthesis gas and methanol production processes have been studied in (Jakobsen, Lindborg & Handeland 2002) with the 1D and 2D pseudo-homogeneous method. An extensive research on multiphase flows using the porosity approach has been presented in (Jiang et al. 2001, Jiang et al. 2002a, Jiang et al. 2002b)

An alternative simulation method, which cannot be categorized as a one continuum method but it s even far more different to the 3D representation, is the network or cell or channel modelling. The method was first introduced by Dean H. and Lapidus in 1960. The CFD results of the fluid flow in the spatial domain can be combined with a network of cells that represent the packed bed with the implementation of the porosity distribution and fluid to particle interactions. Network modelling in combination with CFD is presented in (Balhoff, Thompson 2006) for non-Newtonian flows and for multiphase flows in (Jiang, Guo & Al-Dahhan 2005). Dixon and Nijemeisland in (Dixon A. G., Nijemeisland M. 2001) refer that “the revival of the cell model approach is likely to meet the same problems as the original, as it must rely on idealized pictures of mixing in the interstices of the packing, and as it is extended to accommodate both heat and mass transfer only with difficulty”.

Another alternative method is the wave model described in (Benneker, Kronberg & Westerterp 1997)for longitudinal mass and heat dispersion in tubular reactors and studied extensively for packed bed reactors by (Iordanidis et al. 2004, Iordanidis et al. 2003). Other ways to solve the problems occurred from the complication and implementation of the physical and chemical phenomena are referred in the literature of (Dixon A. G., Nijemeisland M. 2001) but in these sources, the correlations of porosity distribution are once more used and hence approximations are inserted.

Over the years of research and after an extensive study of all the parameters of the two phase state, numerous correlations and assumptions and predictions have been made, that usually cannot be applied in general. They only match the results of the specific studies. The discrepancies that occur cannot permit qualitative and quantitative agreement with the modelling. Although the approach of one phase continuum is well documented, a sedulous research should be realised and compared to experimental data, not only to literature experimental data but also to the real simulating model-equipment.

2.3.3 The discrete method (heterogeneity and realistic representation)

The discrete method on the other hand, has not been studied extensively and is not as well documented as the homogenization method. The complex geometry of packed bed imposes a significant number of difficulties, with the main one to imply on the generation of the certain geometry, where a mesh can be created.

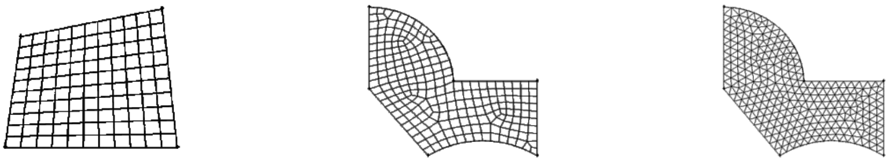
The first and basic step of generating a geometry of randomly packed particles in 3D relies on the representation of:

- the exact shape and size of the particles
- the contact points from particle to particle
- the contact points with the boundaries of the reactor (walls) or other external geometry
- the number of the particles

As a consequence, these factors influence not only the porosity but also the computational time of a simulation that will take into account all of these factors as well as the required computational power. It is possible to design a geometry with the above characteristics, although it is a complicated task. However the creation of a mesh representing thousands or millions of particles with the exact contact points and simulation with all the physical models implemented has not been achieved up to the present time with the existing computational resources as described in the next chapter.

The reason that a mesh is currently unachievable is the number of cells created for this geometry and the quality of them. The current mesh generation packages commercial or not include a 2D or 3D mesh generation. The 3D representation obviously contains a larger number of cells and requires the CFD software to consume more memory and Central Processing Unit (CPU), extending also the computational time. The discretization of the domain has an obvious impact on the discretization of the differential equations applied, on the numerical diffusion, on the convergence and stability of the simulation and as a result at the reliable or not outcome and conclusion of the simulation.

In 2D representations there are three types of mesh, the mapped for structured grids, the pave with triangles or quads for unstructured grids as shown in Figure 4. The preferable type is the map type but it cannot be applied for the representation of realistic complex geometries.



Mapped surface Quad paved surface Tri paved surface

Figure 5 :Surface mesh types

The 2D mesh is the precursor for the 3D mesh generation and for the complex geometry requested for fixed beds, only unstructured mesh can be generated, thus tetrahedrons, hexahedrons, wedges and pyramids are used to represent the cells of the mesh as shown in Figure 5.

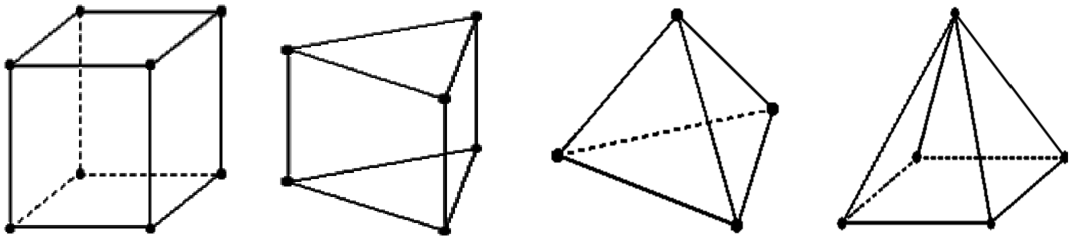


Figure 6 Types of cells (Gambit user guide)

The size of these cells and as a result the number of them control the computational time as the equations are solved for more nodes. The structured mesh consists only of quadrilateral in 2D or hexahedron cells in 3D and every volume is connected to the neighbour volumes. The unstructured mesh allows the flexibility of one volume to be connected with every other and the cells can have any shape but also introduces some problems. The computational time plays an important role although the quality of the results in CFD is even more important, so the positions and the refinement of the mesh control the simulation. The numerical diffusion is connected with the approximate solution of the partial differential equations and results in the reduction of its accuracy. The information in numerical solutions passes from one neighbour cell to the other lateral to the velocity vector, thus a small mistake in one cell can be transferred to the others. The skewness of the cells which is characterized by the acute angle that is created in tetrahedrons, hexahedrons, wedges and pyramids may lead to numerical diffusion as the solution of one equation may be solved along wrong directions (Bergeles George 2006). Especially a mesh with a large number of highly skewed elements-cells may cause problems during importation to a CFD package, the solution may never converge or become completely erroneous. In order to decrease the phenomenon of numerical diffusion, more dense meshes with an adequate aspect ratio, free of highly skewed elements are required.

The geometry of a fixed bed contains thousands or millions of small particles that touch each other in a random way, thus creating really small and acute angles at their contact point with each other as well as at the contact points with the reactor's walls. The mesh must be denser between the solid and fluid parts within the reactor, in order to capture all flow features. Especially for turbulent conditions the constricted flow areas that means the contact points between particle to particle and particle to wall has to be adequately fine and free of highly skewed cells. Additionally a sensitivity study has to be performed in order to determine the influence of the mesh on the simulation results.

The discrete method requires the representation of both phases which imposes problems at the very first steps of a simulation:

- A too refined mesh may exceed the computational resources
- A mesh with highly skewed elements may raise a question of the solution's accuracy
- A 3D representation of randomly packed particles is a complicated task due to the random positions in 3D and due to the contact points

Although several researches have been made towards this direction to face and overcome the above mentioned impediments of creating a geometry and a mesh of randomly packed bed. In section 2.3 of this thesis, the methods of generating a mesh will be presented as well as the simulation results of the 3D representation and simulation of fixed bed reactors (discrete method).

2.3.4 Choosing a method

The choice of the simulating method mostly depends on the user's requirements and on the appropriate assumptions and approximations of the model. In general a simulation of a fixed bed catalyst reactor should take into consideration the intra-particle diffusion of heat and mass, the heat and mass exchange between the catalyst pellet and bulk fluid, the convection of the fluid, the heat and mass dispersion in the fluid phase, the thermal conduction in the solid phase, the heat exchange with the confining walls (Iordanidis 2002). The dispersion effects in the reactor are not only caused by the complex flow pattern but also by the molecular diffusion, the thermal conductivity in solid and fluid phase and radiation which are connected with the heat generating source, the surface chemical reactions.

The one continuum method is well documented and is the most common used method as mentioned before. Although an extended literature can be found, it is not always possible that the results, correlations and assumptions of the other researchers will fit the specific case-application studied. On the other hand, the discrete method has computational limitations and no extended literature exists up till now.

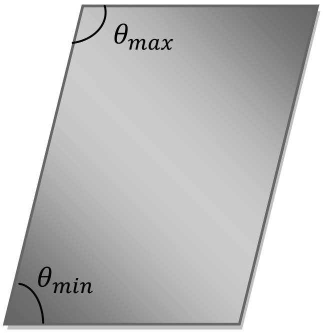
The discrete method presents several advantages compared to the one continuum method. The field of the flow represents the realistic flow streams and flow pattern and the porosity does not need any equations, assumptions and correlations to be defined as it is exactly represented by the designed geometry. The volume of the particle has the exact position and the reactions which take place in the porous particle can also be simulated. The advantages over the one continuum method lead to a more accurate method for simulating fixed bed reactors. The elaborated subject of simulating this geometry requires less simplifications and assumptions. The dominant phenomena can be described well when the simulating geometry approaches the realistic geometry.

The volume ratio and the velocity of the fluid inserted in the reactor are high leading to more intensive phenomena of dispersion. A definition of the flow if it is turbulent, laminar or transient is not appropriate as this can vary from position to position depending on the local topology. A model that does not make use of any approximations for the prediction of turbulence is thus more accurate. As high velocities occur in the industrial plants, the estimation of mass and heat transfer is more dependent of the velocity profile that is estimated with no approximations. The heat transfer coefficients are dependent of the velocity of the fluid and are correlated with the Reynolds (Re) or the Peclet number. At regions where low velocity is observed, that means around contact points, the conductivity is the dominant heat transfer mechanism, thus the lack of contact points can only raise uncertainties of the prediction of heat transfer behavior. The heat transfer in packed beds has been studied extensively, but mostly for the one continuum method and there is no general accepted method found for predicting it (Dixon A. G. 2001, Nijemeisland M. 2001, Tsotsas E., Schlünder E.U. 1990, Logtenberg, Nijemeisland & Dixon 1999, Guardo et al. 2007). For the above mentioned reasons, the work presented in this thesis is focusing on the discrete method and trying to overcome the impediments of the mesh and geometry generation.

2.4 CFD simulations with the discrete method

There are few studies using the discrete method due to the mesh generation problem. Although, researchers in the following studies have managed to create a mesh and to perform CFD simulations with spherical or cylindrical catalytic particles in a cylindrical tube.

The meshing techniques in general try to refine a coarse mesh by smoothing the mesh in the pre-processing designing package or after it is imported to the CFD package or by smoothing the surfaces of the volumes. In the pre-processor CFD designing package, Gambit the default measure of mesh quality is based on EquiAngle Skew that is defined according to the above equation:



$$max\left[\frac{\theta_{max}-\theta_e}{180-\theta_e},\frac{\theta_e-\theta_{min}}{\theta_e}\right]$$ Equation 3

where:

θ_{max} is the largest angle in the face or cell

θ_{min} is the smallest angle in face or cell

θ_e is the angle for the equiangular face or cell

$(\theta_e = 60\ for\ triangle\ or\ 90\ for\ square)$

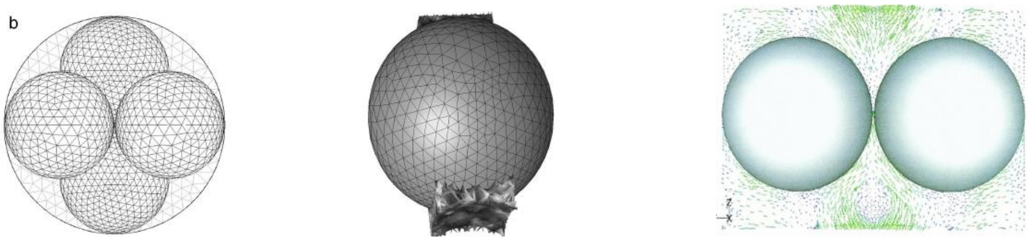
(Martens Stefan 2007)

The skewness of the cells is characterized by a factor that varies from 0 until 1. A mesh that contains cells with skewness over 0.99 cannot be imported in to the simulation package. The number of the highly skewed cells and the factor of skewness compared to the overall number play also an important role. The 2D cells on the surface of the volume should not exceed a value of skewness over 0.65 and the 3D cells in the volume should not in general exceed a value of skewness over 0.85 for hex, quad and tri cells and 0.9 for tetrahedral cells. (Martens Stefan 2007). Common techniques followed are the movement of nodes on the meshed surface, the decrease of mesh size in the high skewness region and the implementation of size functions. The size ratio should not exceed 20% (Martens Stefan 2007). In the following studies there is not an exact mention of the number and the existence or not of highly skewed cells, but the fact that the simulations performed are mesh dependent betrays that the generation of a good mesh without skewness problems. A good mesh mostly consists of smooth variations in mesh size, minimum cell skewness and varies from one geometry to the other. Other factors are also used for the characterization of the mesh quality as aspect and diagonal ratio, equivalent size e.t.c (Gambit User’s Guide).The meshing technique in the literature research is the evasion of contact points. The “near miss” model implemented by Dixon was used to account for the particles’ contact, whereas in other studies several distances between the particles were chosen in order to avoid contact points. The “near miss” model is implementing shrunk geometries of particles that means that a very small gap is created between the contact points that is proportional to the shrunk factor.

2.4.1 Simulating spherical particles

The basis of the research that has been done in the 3D representation of spherical particles starts from a series of published journals by Dixon. From 1996 until up to date, Dixon has achieved a detailed study on CFD simulations on catalyst reactors, starting from a three sphere model to predict heat transfer coefficients and continuing with two layers of four spheres without contact points (Logtenberg, Dixon 1998) and 10 spheres with contact points (Logtenberg, Nijemeisland & Dixon 1999). The mesh that included contact points was manually manipulated by including additional circle lines and spherical dead volumes around the contact points The particle to tube ratio was equal to N=2.43 and different velocity profiles were calculated for the Reynolds number (Re) in range of 42 up to 3344. For high Re numbers, eddies were formed at the contact points, whereas no eddies were formed in low Re resulting in different flow and heat transfer behavior. In addition no significant differences were observed for wall heated or cooled reactors with or without heat generation from the spheres.

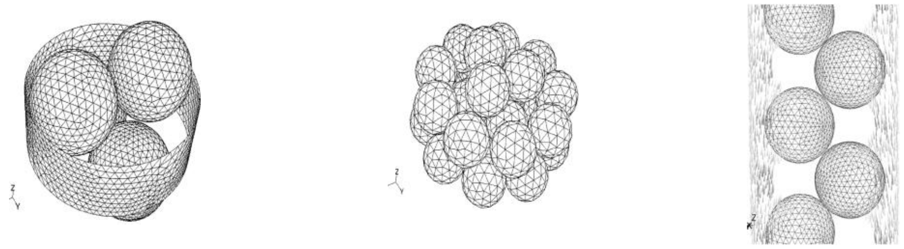
In the first simulations performed by (Nijemeisland, Dixon 2001), the contact between 44 solid particles was created with common nodes and two nodes around them for the fluid elements at the wall. Figure 6 shows the geometry and mesh generation of this study. The high skewness within this mesh was obvious and inevitable. The mesh was used for laminar flow simulations without any significant problems but when a turbulent flow simulation was attempted, the convergence of the solution was unachievable. For the necessity of high Re simulations the “near miss” model was introduced. A case study for the gap size was made as a too small gap would retain the skewness and a too large gap would influence the flow and heat transfer patterns. The optimal size was determined to the gap that is created when a sphere is shrunk to the 99% of the initial diameter because of a compromise between the quick convergence of the simulation and the velocity magnitude variation compared to the initial model. An under-prediction of temperature ranging 1.5K at Re=373 to 2K at Re=1922 was caused due to the “near miss” model introduction. The simulation case for N=2 modeled in FLUENT, the experimental set up and the results showed excellent qualitative and very good quantitative agreement.



10 layers of 4 spheres The “near miss” model Velocity profile

Figure 7:CFD Simulations with 44 spheres (Nijemeisland, Dixon 2001)

Another simulation for N=2 and N=4 was performed by the same authors in (Dixon A. G. 2001, Nijemeisland M. 2001). For N=2, 430.000 meshed volumes required 5.5 hours of CPU time for a laminar flow simulation and 11h for the turbulent flow (500 MHz DEC Alpha). For N=4 simulation, the number of spheres increased so periodic identical boundaries were used and a coarse mesh was created with which an accurate flow pattern for the laminar flow was not able to be predicted. The meshed geometry of these simulations is presented in Figure 7.

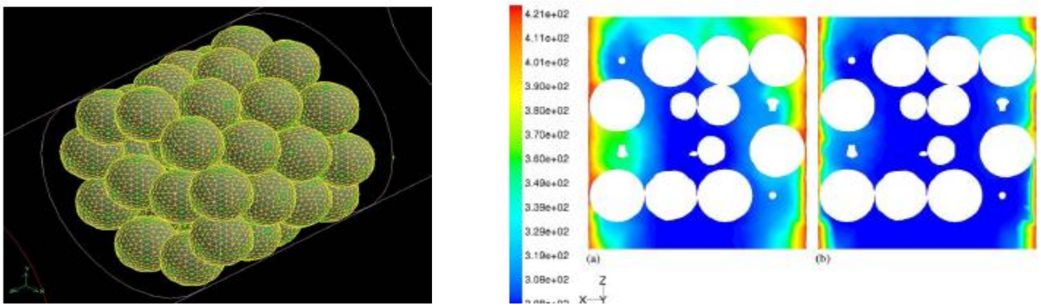


Meshed geometry for N=2 Meshed geometry for N=4 Velocity profile N=2

Figure 8:CFD Simulations with 44 spheres (Dixon A. G. 2001, Nijemeisland M. 2001, Nijemeisland, Dixon 2001)

CFD simulations and mesh generation for Composite Structured Packing (CSP) was used by (Romkes S.J.P et al. 2003) for tube to particle ratios ranging from 1 to 2. A case study of mesh density variations and simulations in CFX-5.3 was used to predict the heat transfer from particle-to-fluid of a single sphere and from wall to fluid for laminar and turbulent flow in an empty tube. For the variation of 8 to 16 spheres depending on the tube to particle ratio, the “near miss” mesh model was used and correlations of Nusselt and Reynolds number where validated with experimental results. The mesh for the laminar flow consisted of 3.4million cells. The dimensionless parameter that is used to verify the mesh adequacy for turbulent flows’ simulations is called y^+ (Romkes S.J.P et al. 2003). With the consideration that the y^+ value for the boundary layer is not influencing the heat transfer model in CFX, turbulent flow simulation with respect to discretization errors was also performed. The average error in comparison to Nusselt number correlations was less than 15% for $1 \leq N \leq 2$ except for N=1.47 where it was 31% because of the high voidage in the geometry.

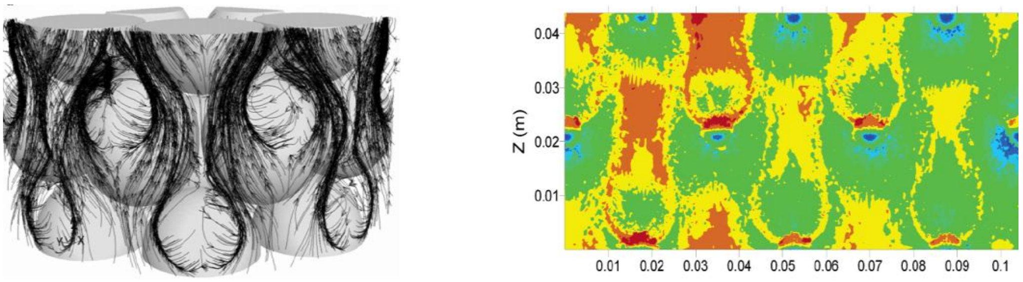
In (Guardo et al. 2005) a four layer array of 11 spheres with 9 contact points and 3.923 diameter ratio was used for CFD simulations in FLUENT 6.0 and correlations between the Nusselt and Reynolds number were obtained and compared for different turbulence models. The definition of the y^+ was crucial for the appropriate turbulent model. Discrepancies occurred in the simulation for low Re and the transition rate and the rates of convergence were much slower. Figure 8 indicates the results of the simulated geometry.



Meshed geometry Temperature profile

Figure 9: CFD Simulations with 44 spheres by (Guardo et al. 2005)

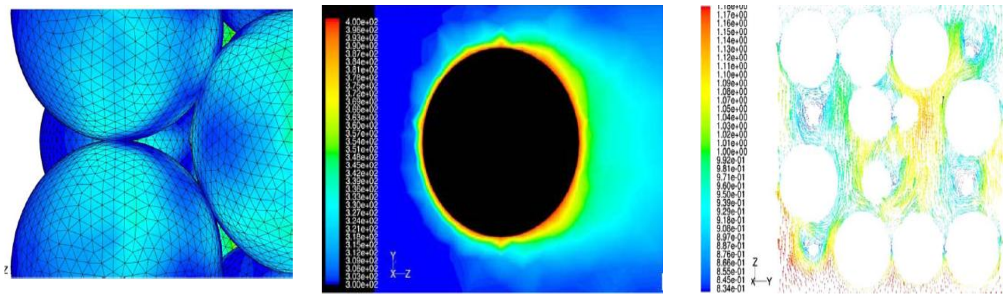
A 120° segment with two axial layers of 5 spheres and periodic boundaries was used to simulate an N=4 steam reforming reactor in FLUENT 6.1.(Dixon, Nijemeisland & Stitt 2005) The “near miss” mesh method is followed according to (Nijemeisland, Dixon 2004) and the effect of the wall conduction is studied. The results reveal a negligible difference between the wall conduction and no wall conduction cases but a reduction of the maximum temperature and an increase of the minimum temperature are observed with the wall conduction. Figure 9 indicates the flow field and temperature profile of this study.



Streamlines around the packed spheres Axial and radial temperature profile

Figure 10: CFD Simulations on steam reforming reactor by (Nijemeisland, Dixon 2004)

A validation case of one single sphere in laminar or turbulent air flow, simulations with incompressible air at low pressure for forced convection and mixed convection at high pressure with CO2 as supercritical fluid were presented in (Guardo et al. 2006). The same work was extended in (Guardo et al. 2007). The mesh was generated by overlapping spheres (0.5% of their diameter) in order to include contact points. For low Re, the results do not show a good agreement with the correlations and no mesh sensitivity was noticed. For higher Reynolds numbers (Re>10) and for a single velocity condition, the mesh influences the results of the Nusselt number but good agreement with the theoretical correlations was achieved. Also the effect of the flow rate and direction over the mass transfer is observed in the laminar flow of the simulating supercritical fluid. The meshed geometry, the temperature field of the one-sphere validation model as well as the velocity field of the 44 sphere model are presented in Figure 10.



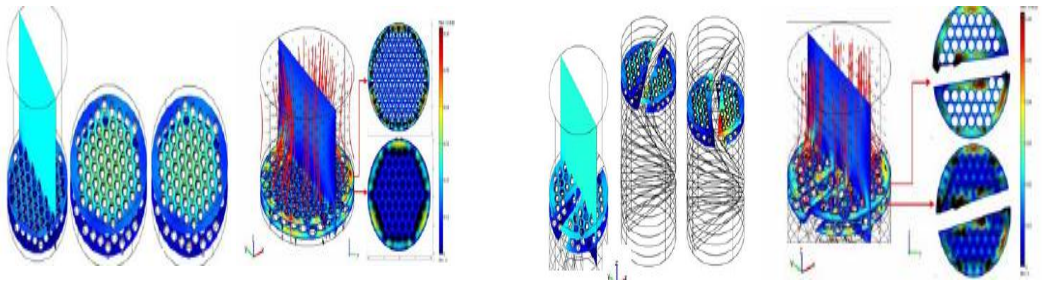
Meshed geometry

Temperature profile of one single sphere model

Velocity field

Figure 11: CFD Simulations of forced and mixed convection (Dixon A. G., Nijemeisland M. 2001, Nijemeisland, Dixon 2001)

The commercial software COMSOL MULTIPHYSICS 3.3 was used in (Phavanee N. et al. 2009) in order to simulate the 3D flow field of gas reactants CO and H2 in a tube (d=0,8cm and h=1,2 cm) with and without a mixer. 700 spheres of 650 µm diameter and without any contact points were designed and meshed (mesh size 168.450 with a static mixer and 241.471 without). Only the hydrodynamic field was simulated resulting in velocity and pressure profiles for the two cases as shown in Figure 11.



Without the mixer

With the mixer

Figure 12: Streamlines and velocity profiles in CFD Simulations of 700 spheres by (Guardo et al. 2005)

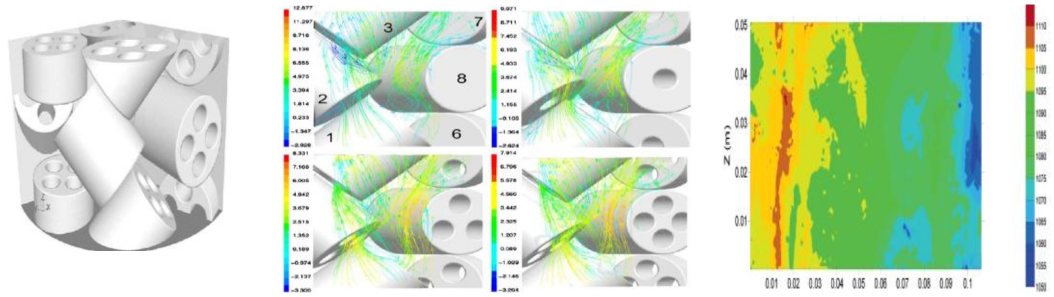
A randomly packed geometry of spheres with a variety in the number and diameter to achieve different porosity was presented in a study by (Jafari et al. 2008). The geometry was automatically generated by the journal input file for Gambit that was a result of a C++ code. More than 2×10^{6} volume cells were created to mesh the cylinder reactor of 21cm length and 6cm diameter. The transition of the flow, pressure drop, and velocity profile as well as dispersion model were investigated. A model of Navier Stokes equations was used including inertia forces but not a turbulent model according to Darcy's regimes. Nevertheless this study included theoretical correlations for the porosity and turtuosity of the geometry as the geometry was constructed using Matlab software, the 3D representation of the geometry was chosen. Figure 12 illustrates the generated geometry and velocity profile.

Automatically generated geometry Velocity profile

Figure 13: LES simulations in a randomly packed reactor (Jafari et al. 2008)

2.4.2 Simulating cylindrical particles

The cylindrical geometry is a more complicated geometry as the size, the position and orientation of the cylinders must be taken into account. In (Nijemeisland, Dixon & Hugh Stitt 2004) four cases of cylinders who had 0, 1, 3 or 4 longitude holes in a 120° periodic segment with two axial layers of particles in a steam reforming reactor were studied. The packing was created according to the Unidense Method and the same geometry was studied more detailed in (Dixon, Nijemeisland & Stitt 2005) with the implementation of wall conduction. The flow pattern and subsequently the energy equation were solved in order to treat the flow as periodic and then the outlet conditions of one segment to serve as inlet for the next one. The no heat sources or sinks case was modeled by the application of a constant heat flux to the reactor wall and for the four cases studied it was proved that the more holes in the cylindrical particles the worse is the heat transfer effectiveness of the bed. The radial profile of solid cylinders with heat sources/sinks and without presents several differences, but when the longitude holes are added in the simulation it changes dramatically. Figure 13 presents the results of the study. The simulated segment of 120°, the four different cases of longitude holes in the cylindrical particles and the temperature profile of the simulation which included the conduction mechanism are presented equivalently.

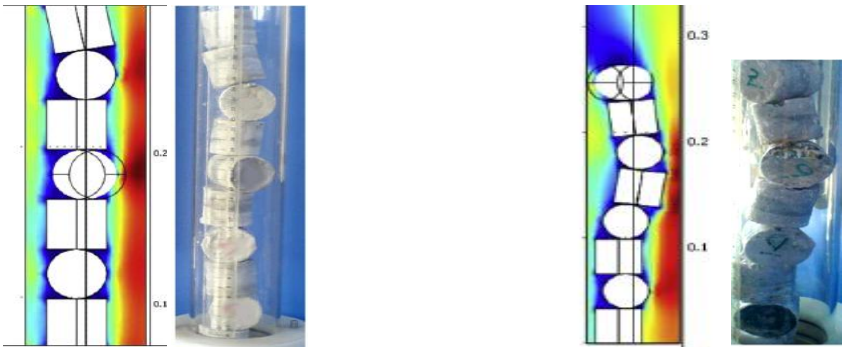


Simulated segment   Streamlines for the 4 cases   Temperature profile with conduction included

Figure 14:CFD Simulations on cylindrical particles (Nijemeisland, Dixon & Hugh Stitt 2004)

The same geometry was simulated in (Dixon et al. 2008) where in the mesh generation, except from the “near-miss” model, a prismatic mesh of layers with different growth factors was implemented for the boundary layer treatment for turbulent flows. For y^+ in range of 5÷30, the results were not trustworthy, but for higher values of y^+ this boundary layer treatment proved to be a good approach with the turbulent model used. The research did not include any endothermic reactions and concluded that the multi-holed particles present lower temperature at the walls of the steam reforming reactor.

FEMLAB 2.3 was used to simulate two different arrangements of naphthalene coated cylinders, designed as closely as possible to the experimental setup by (Motlagh, Hashemabadi 2008) as shown in the Figure 14. The hydrodynamic and heat transfer behavior of the N=2 fixed bed were investigated and validated by naphthalene sublimation mass transfer experiments. Figure 14 illustrates the contour of the heat flux in the examined reactor. The results of CFD simulations and experimental results showed good qualitative agreement.



Arrangement 1   Arrangement 2

Figure 15: 3D simulations with randomly packed cylinders (Motlagh,A.H., 2008)

2.4.3 Combining Discrete Element Method with CFD

Discrete Element Method (DEM) is a numerical method for computing the motion of individual particles that range in number and size by solving the Newton's second law of motion. The method was first introduced by (Cundall, Strack 1979) for granular media and since then various researchers have tried to extend it to fluid-particle interaction systems. The method includes two types of motion translational and rotational. Interactions between the particles as well as particle-to-wall interactions may be included to calculate the overall forces, torque and motion of an individual particle. Gravity, buoyancy, collision, inter-particle forces with the inclusion of the contact force and viscous damping contact force are some of the interaction that can be implemented in DEM simulation for various and complex geometries. The implemented interactions and the shapes of the particles from one program to the other and from one application to the other differ. According to the Wikipedia website there is a list of DEM available commercial or open source software that is presented in Table 2 (Wikipedia-DEM 2010).

Table 2 : Available DEM software (Wikipedia-DEM 2010)

Open source or non commercial DEM	*Commercial DEM software*
BALL & TRUBAL (1979–1980)	Chute Maven
LAMMPS	PFC2D and PFC3D
SDEC	EDEM (DEM Solutions Ltd.)
YADE	GROMOS 96
LMGC90	ELFEN
ESyS-Particle	MIMES
Pasimodo	PASSAGE/DEM
	UDEC aand 3DEC
	Ascalaph Molecular dynamics

Except from the above mentioned forces in macroscopic scale, the user has to implement cohesion, adhesion, liquid bridging, electrostatic attraction e.t.c or forces in the molecular level the Coulomb force, the Pauli repulsion or van der Waals force (Wikipedia-DEM 2010). Different integration methods can be employed to calculate the velocity and position of every particle for the next time step. As the computational power increases, the number of particles simulated has also increased. Millions of particles can be simulated to predict their motion in a 3D domain and define their positions at every time step. The overlap between two particles or the large displacement, on the contrary are influenced by the elastic stiffness factor. The numerical stability is also very sensitive to the elastic stiffness factor thus careful consideration should be taken for the optimization of these parameters in a DEM simulation (Xu, Yu 1997).

The combination between CFD and DEM has recently been developed for various applications. There are two existing ways of combining these methods. The first method is a direct coupling of DEM and CFD simulation. DEM tracks the position of the particle and applies particle to particle and fluid to particle forces in a computational cell at every time step. The results are inserted in CFD where the fluid drag forces from the gas flow are obtained and incorporation of the resulting forces in DEM at every time step gives the final results as shown in Figure 15.

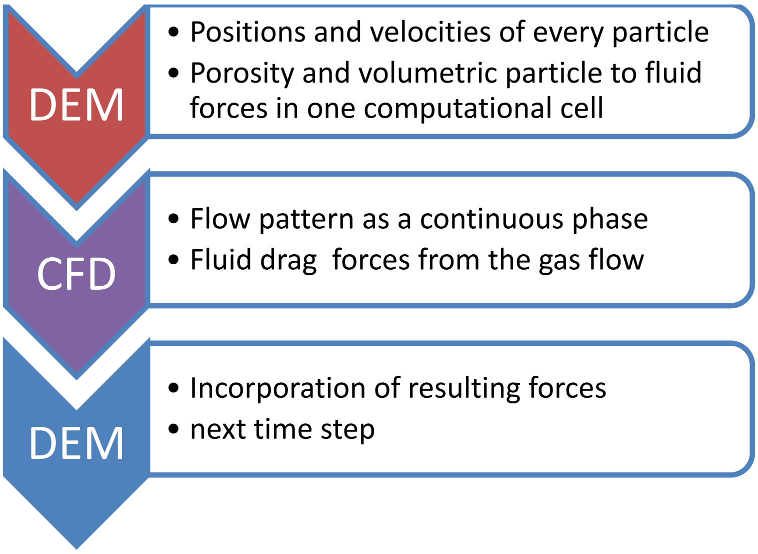
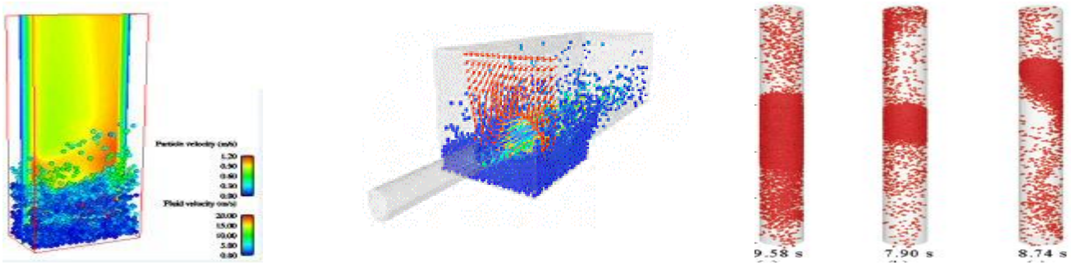

DEM
• Positions and velocities of every particle
• Porosity and volumetric particle to fluid forces in one computational cell
CFD
• Flow pattern as a continuous phase
• Fluid drag forces from the gas flow
DEM
• Incorporation of resulting forces
• next time step

Figure 16: Coupling DEM and CFD. Method 1

The location of the particle in the CFD code for the calculation of fluid drag forces indicates the need of 3D hybrid meshes, an example of which can be found in (Kuang, Yu & Zou 2008). The parallelization of the codes is achieved by Open Multi-Processing (OPenMP). OPenMP is an application programming interface (API) that consists of a set of compiler directives, library routines and environment variables and supports parallel applications in computers consisting of a shared memory multiprocessors or multi-core processors. This particular method is widely accepted for applications that require the constant movement and displacement of the particles as in pneumatic conveying (Kuang, Yu & Zou 2009) or in fluidized bed applications (Xu, Yu 1997, Di Renzo, Di Maio 2007), in entrainment and deposition as well as in separation and filtration processes as shown in Figure 16. As the method is on each early stage of development, the incorporation of the effects of turbulence is still an open research topic.



Fluidized bed simulation (http://www.dem-solutions.com/videos.php)

Entrainment of sand particles by a jet of water (http://www.efluid.com.cn/topic/detail.aspx?id=4229)

Pneumatic conveying (Kuang, Yu & Zou 2009)

Figure 17: Direct coupling Of CFD-DEM simulations

The second method provides also a combination between DEM and CFD but every simulation is performed separately. The DEM simulation starts with the calculation of the forces applied individually on every particle including only particle-to-particle and particle-to-wall interactions without any fluid interaction. The results of the simulation and the final position of the particles are implemented in the pre-processor designing program for the geometry and mesh generation. A 3D representation of the geometry as close to reality as possible is achieved this way and with the prerequisite of a good quality mesh, the CFD simulations may start with a well predicted flow pattern and exact porosity.

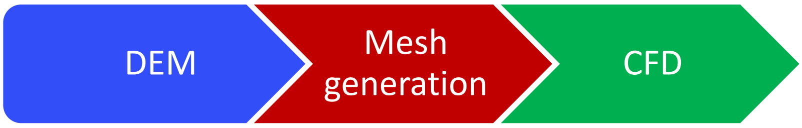
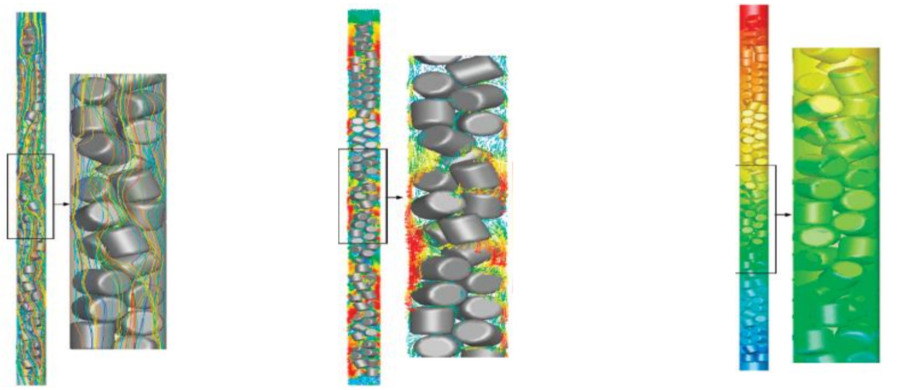


Figure 18: Coupling DEM and CFD. Method 2

This method is more applicable for fixed bed simulations as the particles are considered to have a fixed position inside the reactor. In (Bai et al. 2009) a study of flow field and pressure drop in a fixed bed reactor with the second method coupling of DEM and CFD is presented. The results of the simulation with this method used show good agreement with measurements taken. The pathlines of the flow, the velocity vectors and the pressure profile are shown in Figure 18.



Flow pathlines Velocity vectors Pressure profile

Figure 19 : Direct coupling Of CFD-DEM simulations (Bai et al. 2009)

PFC3D was used for the DEM simulations. Two experimental set-ups were constructed to validate the results, a laboratory scale set up consisting of structured or random packing of spheres or cylinders for the modeling of the whole reactor and a plant scale setup of hundreds of randomly packed spheres for tube to particle rations below 4 (N<4). For the cylindrical particles in the laboratory experimental setup, an assembly of 1000 spheres was used to represent the geometry of one cylinder as PFC3D can simulate only spheres. The geometry is shown in Figure 19. The mesh generation technique is summarized in the adoption of the “near-miss” mesh model with the a shrinkage factor of 0.5% for spheres and 1% for cylinders and for the industrial application, a segment of the reactor was simulated due to computational resources limitation.

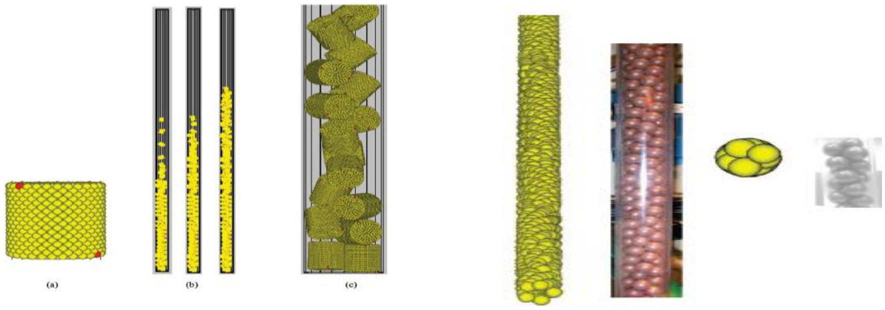


Figure 20 :Randomly packed geometries of cylinders (consisting of spheres) and spheres (Bai et al. 2009)

As the porosity of the tube influences the pressure drop, a study of the deviation between the predicted packing from DEM and the actual packing was made. It was concluded that the porosity deviation occurs from the DEM simulation, the use and the choice of one simulated segment and the particle shrinkage and a correction factor of the porosity deviation was introduced that resulted in 10% error between the CFD and plant-scale experimental results. The inconsistency and unreliability of the empirical correlations for low tube to particle ratios was also observed. Figure 20 shows the discrepancies between the Ergun's equation (grey line) with the CFD results (blue and azure line) and the experimental results (red line) as well as the pressure drop along the reactor of randomly packed spheres.

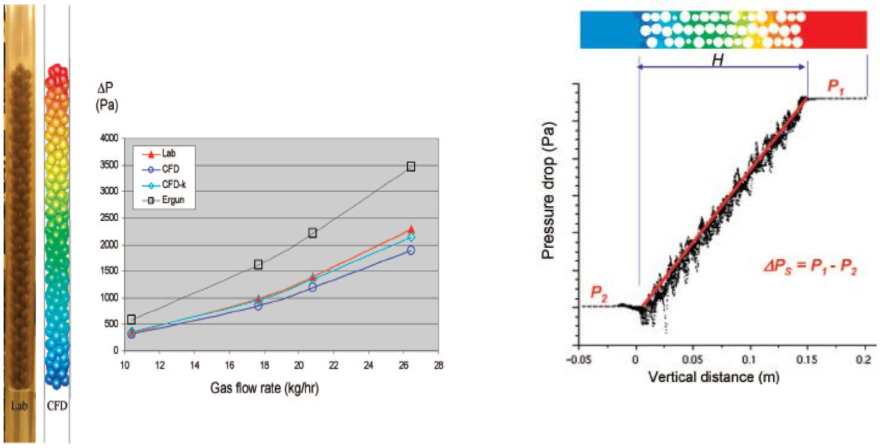


Figure 21 :Pressure drop results from DEM-CFD simulations of randomly packed spheres(Bai et al. 2009)

The 'near miss' model is an approximation of the 3D representation of the reactor's geometry. The effects of this small gap inserted between the spheres in order to avoid the contact points are still under investigation as presented in a case study for two spheres in a pebble bed reactor core (Lee et al. 2007). Nevertheless the potential of a simulation with the discrete method and the realistic geometry remains still a challenge.

2.5 Models used

While engineering modeling technology offers great potential, the appropriate modeling approach that characterizes qualitatively and quantitatively the simulated technology should be taken into careful consideration for the reliability of CFD simulations. A summary of the existing models and the models used in the above mentioned literature will be presented in this section.

2.5.1 Basic Equations

Modeling technological systems based on CFD means that pertinent physic and chemical laws have been embedded in to mathematical expressions which are solved numerically. The basic mathematical equations that characterize one system are summarized according to the physical laws to the solution or to an approach solution of the three basic equations: the conservation of mass, momentum and energy. Sub-models are coupled with these three fundamentals to predict and characterize precisely a system.

In Fluid Dynamics, the conservation of mass is described by the continuity equation:

$$\frac{\partial\rho}{\partial t}+\frac{\partial(\rho u_i)}{\partial x_i}=S^m \qquad \left(\frac{kg}{s\cdot m^3}\right)$$ Equation 4

where ρ is the fluid density, u_i is the velocity components in each of the x_i directions and S^m the source of mass generation on a volumetric basis. The terms on the left of the continuity equation represent the rate of change of mass per unit volume with time and with the motion of the fluid. The conservation of momentum, for Newtonian fluids, is described in the three x_i directions and the three equivalent velocities by the following equation:

$$\frac{\partial(\rho u_i)}{\partial t}+\frac{\partial\left(\rho u_i u_j\right)}{\partial x_j}=$$

$$-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\times\mu\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\delta_{ij}\frac{2}{3}\left(\frac{\partial u_k}{\partial x_k}\right)\right]+f_i+S_i^u \qquad \left(\frac{kg}{s\cdot m^3}\right)$$ Equation 5

where p is the local pressure , μ is the factor of fluid's dynamic viscosity and δ_{ij} the Kronecker Delta function, f_i the body forces acting on the fluid and S_i^u the sources of momentum generated on volumetric bases. The terms on the left of the momentum equation express the rate of change of momentum per unit volume with time and with the motion of the fluid. The terms combined with the dynamic viscosity μ on the right represent the shear stress caused by the fluid's motion.

After the definition of the Navier-Stokes equations, the fifth equation required to be solved is the equation of energy. The energy equation incorporating the first and second thermodynamic laws in terms of enthalpy can be written in the following form:

$$\frac{\partial(\rho h)}{\partial t}+\frac{\partial(\rho u_i h)}{\partial x_i}=\left(\frac{\partial}{\partial x_j}\right)\left(\Gamma_h\frac{\partial h}{\partial x_j}\right)+\frac{\partial p}{\partial t}+\frac{\partial u_i p}{\partial x_i}+\Phi+S^h \qquad \left(\frac{kJ}{s\cdot m^3}\right)$$ Equation 6

where h is the specific enthalpy, Γ_h is the ratio of effective viscosity and the Prandtl number, Φ is the dissipation function and S^h is the source of enthalpy generated on a volumetric source basis. The term on the left expresses the rate of enthalpy change per unit volume with time and motion equivalently. The term $\left(\frac{\partial}{\partial x_j}\right)\left(\Gamma_h\frac{\partial h}{\partial x_j}\right)$ represents the molecular diffusion of enthalpy according to Fourier's or Fick's law of heat conduction. The pressure term expresses the reversible work done on the fluid and Φ

expresses the irreversible work or dissipation of energy resulting from the shear stresses. The shear stresses represent a heat source that raises the internal energy of the fluid and the dissipation function is given by the following expression:

$$\Phi = \mu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \delta_{ij}\frac{2}{3}\left(\frac{\partial u_k}{\partial x_k}\right)\right]\frac{\partial u_i}{\partial x_j} \qquad \left(\frac{kJ}{s \cdot m^2}\right)$$ Equation 7

(Eaton et al. 1999, Bergeles George 2006, Tsaggaris Socrates 2005). The Navier-Stokes equations represent all the flow fields that exist in nature, flow fields that remain stable or change over time, flow fields of any geometry and fluids of every pressure or temperature under the prerequisite that the fluid acts as a continuum (Bergeles George 2006). As the mathematical problem of Navier-Stokes equations, with the incorporation of all the boundary conditions, does not have one unique analytical solution (Ladyzhenskaya 1975), numerical simulations are used. Since the Direct Numerical Simulation (DNS) and the vortex dynamics require extensive computational power and resources for the simulation of fundamental flow structures with the assumptions of periodic conditions and small Re numbers, the approach of finite volumes is followed for practical mechanical applications. The solution procedures for the finite volume methods, are categorized into those for compressible and for incompressible fluid. The difference between them is the coupling between the pressure and the density.

2.5.2 The model of turbulence

The problem of turbulence modeling in CFD has no final solution up to date as it is still an active field of research. The time dependent Navier- Stokes can be solved directly for low Re numbers (laminar flow) whereas in flows with great inertial effects, small eddies appear in the flow field and the terms of Navier Stokes equations change in time. For the simplification of Navier Stokes equations, the time scale of the physical phenomena is reconsidered and the expressions of velocities, pressures, stresses and heat transfer in terms of a mean value and a fluctuating component is considered to be adequate. Considering the fluctuations to be of small scale and high frequency, the governing equations become time-averaged, ensemble-averaged (Fluent’s User Guide 2003). These expressions, when applied to Navier-Stokes equations, result in the Reynolds Averaged Navier Stokes (RANS) and contain significant simplifications due to the larger time scale (Bergeles George 2006). The following equation presents the RANS equations with the ensemble-averaged or time-averaged expression of values:

$$\frac{\partial(\rho u_i)}{\partial t}+\frac{\partial(\rho u_i u_j)}{\partial x_j}=-\frac{\partial p}{\partial x_i}$$

$$+\frac{\partial}{\partial x_j}\times\mu\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\delta_{ij}\frac{2}{3}\left(\frac{\partial u_k}{\partial x_k}\right)\right]+f_i+S_i^u-\frac{\partial\left(\rho\overline{u_\iota{}'u_j{}'}\right)}{\partial x_j}\qquad\left(\frac{kg}{s\cdot m^3}\right)$$

Equation 8

For compressible flows or variable-density flows the Favre-Averaging technique is used to account for the effects of density fluctuations due to turbulence. If is φ any conversed scalar variable, then $\tilde{\varphi}$ is the new Favre-Averaged value, according to equation 8:

$$\tilde{\varphi}=\frac{\overline{\rho\varphi}}{\bar{\rho}}$$

Equation 9

Independently of the way of averaging, the RANS equations result in steady state forms with the term $\rho\overline{u_\iota{}'u_j{}'}$ known as the Reynolds stress terms. The different models of turbulence focus on the solution of these terms by the Reynolds stress model (RSM) (direct solution) or modeling of them with zero, one and two equations (Rodi Wolfgang 1993). While the RSM model solves an additional partial differential equation for every Reynolds stress term resulting in more computational intensive simulation, the other models use the Boussinesq hypothesis (Boussinesq 1877) which relates the Reynolds stresses to the mean velocity gradients.

Zero equation models use an algebraic model to determine the eddy viscosity. One of the most common is the Prandtl mixing length model (Rodi Wolfgang 1993). The one equation model inserts the term of kinetic energy (k) adding one partial differential equation for the transport of kinetic energy. A review of one model equation can be found in (Launder 1972). Spalart-Allmaras is a one equation model that has been shown to give good results for boundary layers subjected to adverse pressure gradients and used for coarse meshes (Launder 1972). The two equation models insert one more partial differential equation for the dissipation ratio (ε) or the specific dissipation ratio (ω). The k- ε models can be categorized to standard k- ε, renormalization group k- ε (RNG) and realizable k- ε that are modifications of the standard k- ε model described in (Launder 1972). The k- ω models can be categorized to standard k- ω and the shear stress transport k- ω (SST) model. The theory of the models as well as the comparison between them and suggested implementations to CFD simulations can be found in (Fluent's User Guide 2003). Table 3 by (Guardo et al. 2005) presents the basic models of turbulence used.

Table 3 Turbulence models (Guardo et al. 2005)

Model:	One equation models	Two equation models			
	Spalart-Allmaras	Standard k- ε	RNG k- ε	Realizable k- ε	Standard k- ω
Reference:	Spalart and Allmaras (1992)	Launder and Spalding (1974)	Choudhury et al. (1993)	Shih et al. (1995)	Wilcox (1998)
Features:	Uses a differential partial equation for the turbulent velocity scale. The turbulent quantity modeled is the effective viscosity.	Uses one differential equation for the turbulent velocity scale and another for the turbulent length scale. The variables modeled are the turbulent kinetic energy, *k,* and the rate of dissipation of turbulent kinetic energy, ε.	This model is derived from the instantaneous Navier–Stokes equations, using the "renormalization group "(RNG) methods. This results in a model with constants different from those in the standard *k*– ε model, and additional terms in the transport equations.	The term "realizable" means that the model satisfies certain mathematical constraints on the normal stresses, consistent with the physics of turbulent flows. It adopts a new eddy-viscosity formula and a new model equation for dissipation *(*ε *)*	This model is based on model transport equations for the turbulence kinetic energy *(k)* and the specific dissipation rate *(*ε *)*, which can also be thought of as the ratio of ε to *k*. Production terms have been added to the model equations.

The choice of a turbulent model depends on the specific geometry and application. In (Logtenberg, Dixon 1998) the choice of a turbulent model for the fixed bed applications is investigated in simulations performed in Ansys Flotran. The non touching spheres used and the void fraction of the reactor indicated that a case study for laminar or turbulent flow, should be carried out. The Reynolds number, at which the turbulent eddies begin to be formed in a fixed bed reactor, is not defined and may vary from one geometry to the other. Both laminar and turbulent flow are suggested to be used and compared in terms of velocity and temperature profiles as well as correlations with Nusselt numbers and radial effective thermal conductivity ratio. The results of this study show no difference between the laminar model and the standard k- ε turbulent model for Re in range of 58-580. The same authors in (Logtenberg, Nijemeisland & Dixon 1999), for a fixed bed consisting of 10 spheres with particle-to-particle and particle-to-wall contact points, used a laminar model and the standard k-ε turbulent model. It was found, that for Re below 182, no eddies were formed, but for larger numbers eddies were formed at the contact points between the wall and the particles which changed the trend of Nusselt number and increased the heat transfer. In (Dixon A. G. 2001, Nijemeisland, Dixon 2001), the "near-miss" model was introduced for 44 spheres and the hydrodynamic and heat transfer behavior for turbulent and laminar flows were studied. Important conclusions about the flow near the "contact" points were made. Small vortices are formed at the downstream in the wake of the spheres' "contact" points and the flow in the small void spaces around the “contact” points is very slow. This indicates that there are almost stagnant fillets around the “contact” points through which conduction is the dominant heat transport mechanism. The radial flow was well predicted at those simulations. This relies on the fact that the mechanism of heat convection is also well predicted resulting in an increase of the overall heat transfer. The simulations were performed for different Re numbers and validated by experimental results. The comparison between the two different kinds of flows for Re=1922 and Re=373 shows little difference. Additionally, the comparison between experimental data and simulations shows lower temperature profiles that are more intensive for higher Re numbers. The turbulent model used was the standard k- ε for the basic simulations, while the k- ε RNG and the RSM model were used for the comparison of the results, where no significant differences were observed. In (Dixon et al. 2008) the k- ε RNG was used for an unstructured tetrahedral mesh with a boundary layer prism mesh near the tube wall that proved to be a good approach for fixed bed simulations.

In (Romkes S.J.P et al. 2003) a case study for a one sphere model was performed for the comparison of different turbulent models and the prediction of mesh adequacy. For the turbulent flow ($127<Re<127x10^5$) where the laminar sub layer and buffer layer were described by standard wall functions, the standard k- ε, the k- ε RNG and the RSM model were used. The k- ε RNG showed better results compared to standard k- ε because of the limited performance of standard k- ε on curved boundary layers. As RSM and k- ε RNG showed similar results, k- ε RNG was chosen for the simulations due to less computational time needed. For the simulations of 8-16 spherical particles, both the standard k- ε and RNG were used to determine correlations between the Nusselt and Re number. Special attention was given to the mesh adequacy according to y^+ factor for turbulent flows for the near wall treatment. The y^+ criterion was only met for higher Re. For lower Re, y^+ values were even smaller than one but CFX code appeared to be quite robust outside the recommended values of y^+ (typical values $30< y^+<300$, recommended $10< y^+<1000$ (Romkes S.J.P et al. 2003)) .

Guardo in (Guardo et al. 2005) presents an analytical study of the influence of the turbulent models used. The mesh adequacy was studied according to y^+. The two layer modeling scheme and the wall functions for the near wall treatment when a k- ε model is used, were found not to be adequate enough. For these reasons, the Spalart-Allmaras turbulent model was used, which incorporates wall functions and damping functions. More information about the wall treatment with wall functions and two layer modeling scheme as well as for the requirements when the Spalart-Allmaras model is used can be found in (Fluent User’s Guide, 2003). All models present similar velocity profiles except at the wall boundaries. Analytically the standard k- ε model predicts larger heat transfer rates through the wall due to stagnation points which was also referred in (Logtenberg, Nijemeisland & Dixon 1999). The k- ε realizable model over-predicted the dissipation resulting in under-estimated heat transfer. The additional diffusion terms of standard k- ω affected the heat transfer parameters. The RNG k- ε showed an under-estimation of the turbulent viscosity. Spalart-Allmaras presented better results for the estimation of pressure and heat transfer parameters. Figure 21 presents the comparison of turbulence models performed in this study.

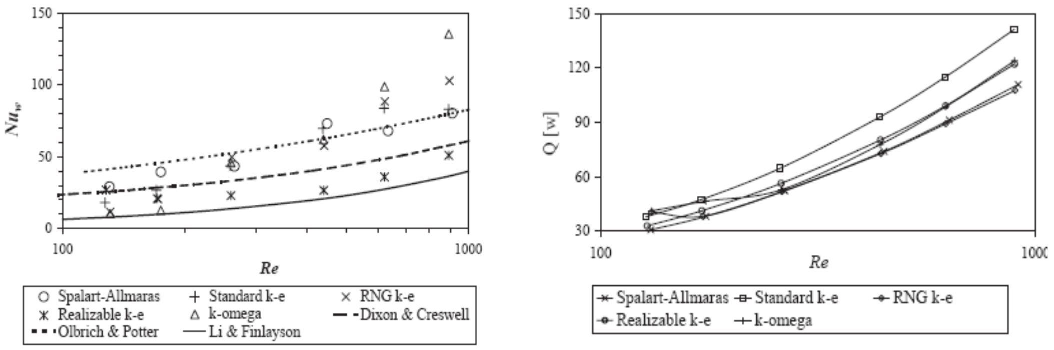


Figure 22: Results from Guardo’s comparison of turbulence models according to the Nusselt number and heat transfer rate through the wall equivalently (Guardo et al. 2005)

Special consideration should be taken for Re<300 as RANS models fail to predict the transition from laminar to turbulent flow and many discrepancies between simulation results, experimental data and empirical correlations occur (Guardo et al. 2005,Guardo et al. 2006). The Spalart-Allmaras turbulent model was used to the following studies of the author (Guardo et al. 2005, Guardo et al. 2006, Guardo et al. 2007).

Dixon and Nijemeisland in (Dixon, Nijemeisland & Stitt 2005, Nijemeisland, Dixon 2004) used the RNG k- ε model with the two layer modeling scheme in order to compute the kinetic energy in the wall-adjacent cells. The solution of the flow was first determined and subsequently the energy solution in periodic segments. The two solutions were decoupled as in the high industrial flow rates, the gas temperature was not expected to influence the flow. In (Phavanee N. et al. 2009) the laminar model was used for the steam reforming simulations.

The Large Eddy Simulation model (LES) is another approach of solving Navier-Stokes equations. The Navier-Stokes equations can also be manipulated to filter the small scales. In (Jafari et al. 2008) LES was proved to be a useful tool for the prediction of turbulent flow regimes in porous beds. According to the authors the dispersion in the fluid flow plays an important role and can be categorized in three principal mechanisms: the molecular diffusion across the streamlines, the diffusion caused from stochastic velocity fluctuations due to the random geometry of the particles and the non- mechanical diffusion due to dead-end pores. Laminar flow and turbulent flow modelled by LES and RSM for comparison were investigated in this study of non overlapping randomly packed spheres. The dispersion was modelled by the volume averaged concentration of the solute in Fick's law with a constant effective diffusion coefficient accounting for the three mechanisms. The Re number was estimated according to permeability, porosity and interstitial fluid velocity magnitude. The transition criteria according to the author have not been defined yet and they vary according to permeability. The modelling of a flow regime with wide range of time and length scales requires a mathematically rigorous methodology. The comparison between RSM and LES showed that LES (with Smagorinsky submodel) showed better agreement with the results of other researchers.

In (Motlagh, Hashemabadi 2008) the standard k- ε model was used in FEMLAB with a different mesh cases in order to achieve mesh adequacy. The comparison was based on the results of pressure drop for different Re numbers, empirical correlations and experimental data of the exact simulated geometry. The average error between the CFD results and the experimental results was proved to be 15.1% for the pressure drop.

In (Bai et al. 2009) the standard k- ε model, the realizable k- ε model, the RNG k- ε model, the Spalart-Allmaras model, the standard and SST k- ω model and the RSM model were compared. The variation of pressure drop was found to be less than 4% for the various k- ε and k- ω models with Spalart-Allmaras model to predict the lowest pressure drop. The RSM model predicted the highest pressure drop and it was the closest to the measurements. RNG k- ε model had an under-prediction of pressure drop less than 3% compared to RSM. Due to the features of RNG k- ε model for flows with strong stream line curvatures and for less CPU time, it was considered the appropriate model for the simulation with the implementation of wall functions.

2.5.3 Models for heat transfer

Heat transfer is in general categorised in conduction, convection, and radiation. According to the application specified heat transfer methods can be taken into consideration in the energy equation. Heat transfer is directly connected to the turbulence modelling as the flow field is determined by the modelling of Navier-Stokes equations. In addition the heat transfer coefficients are connected with Re number correlations. The diffusion term, the viscous heating, the radiation models, the energy sources, the boundary conditions as well as heat transfer coefficients have to be defined according to the application for a CFD simulation.

In (Logtenberg, Dixon 1998), Logtenberg and Dixon describe the heat transfer resistance in a packed bed by the effective radial thermal conductivity k_r and the near wall resistance with the wall transfer coefficient h_w. The effective radial conductivity accounts for all heat transfer mechanisms. Most existing models contain correlations for the k_r/k_f and Nu_w dimensionless terms with the Re number and the tube to particle ratio, N. At this study, the two dimensional pseudo-homogeneous axially-dispersed plug-flow model (ADPF) and by neglecting the axial conduction, the plug-flow model was used. Due to the fact that at low Re the axial conduction cannot be omitted, the ADPF showed better behaviour but still under-predicted the k_r/k_f and Nu_w values. The inaccuracies in this work were considered to exist due to no contact points and the work was extended in (Logtenberg, Nijemeisland & Dixon 1999) with the implementation of contact points between the particles. In this work the ADPF model was used for a range of Re numbers and k_f was given a constant value. The comparison between the empirical model and the CFD results showed many inaccuracies of the Nu_w prediction for Re ranging from 182 to 800.

In (Nijemeisland, Dixon 2001) where the “near-miss” model was introduced, no reference to the k_r/k_f and Nu_w values was made. However, the effects of the lack of particle-to-wall contact points and the gap approximations were studied and compared with experimental data. For 1% reduce of the sphere’s diameter, a temperature difference in range of 1.5 to 2 K was observed compared to the touching spheres model. For the steam reforming simulations, the wall temperature was set to 383 K and the air inlet at 298 K. Simulations that contained a corrected thermal conductivity coefficient accounting for radiation effects were also performed. No significant difference was observed in low Re, but for Re=1922, higher temperatures of 2.5 K were observed. Corrections to the wall temperatures were also implemented to take into account the heat transfer of the missing particle-to-wall contact points.

In (Dixon, Nijemeisland & Stitt 2005, Nijemeisland, Dixon 2004, Nijemeisland, Dixon & Hugh Stitt 2004) the k- ε RNG was used and the energy equation was solved based on the energy balance of k- ε models. A methodology of relating the wall heat transfer to the near-wall flow features with a quantitative and conceptual analysis could not be found even at a wide range of measurements (Dixon, Nijemeisland & Stitt 2005). These studies focused on steam reforming catalysis and simulated the energetic reaction effects through user defined volumetric heat sources or sinks (Nijemeisland, Dixon & Hugh Stitt 2004). The reaction effects were limited on the surface of the particle at 5% of their diameter. In addition constant partial pressures were implemented at positions of interest. The results obtained by the implementation of heat sources gave significantly different temperature profiles and wall temperatures. In (Dixon, Nijemeisland & Stitt 2005) the introduction of wall conduction was made for steam reforming fixed beds. The results revealed that wall conduction has little effect on the average wall temperatures and virtually none on the radial temperature profile. The high and low picks of the wall temperatures are mitigated but local variations still persist.

In (Romkes S.J.P et al. 2003), the variations of Nu according to the different Re numbers are investigated by comparing the values obtained with CFD and experimental results as well as empirical correlations. All solid surfaces were defined to be adiabatic with one or more “active” solid surfaces to have higher temperatures than the fluid’s inlet temperature. The correlations obtained between Re and Nu numbers for different N ratios showed an average error of 15% compared to the experimental results.

A comparison between simulation results and empirical correlations of Nu number and the k_r/k_f was also made by Guardo for different turbulent models. The influence of the turbulence model in CFD fixed bed simulations was proven to be really significant (Guardo et al. 2005). The variations of kinetic energy and the fluctuations of the velocity in the near wall regions reflect on the heat transfer profile. A constant temperature was applied on the particles’ surface and the heat transfer resistance coefficient resides on the fluid side. For low Re, the Nu prediction does not show a good agreement with the theoretical correlations and yields approximately the same value of the axial thermal dispersion coefficient (Guardo et al. 2006). This confirms, that for low Re numbers the particle-to-fluid heat transfer contributes less to the overall heat transfer mechanism of the reactor. In (Guardo et al. 2007) free and forced convection in laminar low for supercritical fluids were also studied and CFD simulations proved to be a reliable tool for modeling free or\and forced convection.

In (Jafari et al. 2008), LES was used to model the turbulence and heat transfer was not studied. This indicates the complicated task of including the energy balance equation in mesh with millions of cells and in sophisticated and computationally expensive models as LES. In (Motlagh, Hashemabadi 2008) the hydrodynamics and heat transfer mechanisms were investigated and compared with empirical correlations and experimental data for the approximately same geometry. The k- ε model was used and different meshes were tested in order to obtain a heat transfer rate that showed to approach the y+ criteria. A constant temperature difference of 10° C, compared to the fluid temperature, was applied on the solid surface of the particles in order to model the heat transfer from the particle to the fluid. A good qualitative agreement was obtained for the local Nu numbers between CFD and experimental results. The two arrangements of cylinders presented an average error of 27.1% and 10% with the CFD results to under-predict the Nu number. The study of the influence of cylinders' position on Nu numbers showed, that the local Nu reaches a maximum near the horizontal cylinders due to the local fluid turbulence around the horizontal cylinder. An increase of the local Nu is also observed in narrow flows between the particle and the bed wall. In (Bai et al. 2009) where the DEM method was used to define the random packing, the heat transfer was not included in the simulations.

2.5.4 Conclusions and Remarks:

The transition regime in fixed beds cannot be accurately predicted. The flow can vary from laminar to turbulent within the reactor according to the geometry. This is happening due to the geometry of the particles, the size and shape of which can define the local type of flow. Small or large eddies, stagnant points or flows around the particles or near the contact points are significant characteristics of the specific geometry simulated. Stagnant flows and laminar flows enhance the mechanism of heat conduction. On the other hand turbulent flows enhance the dispersion mechanism and according to the model of turbulence used, different heat transfer results can be observed. The fluctuations of velocities near the wall region play an important role to the heat transfer mechanisms. Moreover, the local Nu number is influenced by the particle's shape and position except from the dependency of the local flow and Re number. No generalization or exact methodology can be applied for fixed bed geometries of numerous sizes, shapes and ratios (N). According to the mesh size, to the wall treatment, to the chosen turbulence model and the mechanism of heat transfer, different considerations can be made for every application. After the literature research some more appropriate models of turbulence and heat transfer were suggested but a case study should also be performed. The RNG k- ε model was proven to predict more accurate than the other k- ε and k- ω models and similar with the more computationally expensive RSM model. According to the size of the created mesh and the computational requirements, RSM and LES can also be used for the comparison of the results. The Spalart- Allmaras model was also suggested by Guardo while other studies presented many discrepancies after its use. Mesh adequacy and wall treatment plays an important role for the reliability of the results. The comparison with theoretical correlations should be taken into careful consideration according to the models used, the range of Re numbers and the N ratio. After the definition of the modeling regime for fixed bed reactors and the problems which occur, a general conclusion is that, performing CFD simulations for fixed beds requires a robust modeling from the geometry and mesh until the choice of the models used. In order to eliminate the problems that occur during the designing of the geometry and to focus on a creation of an adequate mesh, a way of an automatically generated geometry will be presented in the current thesis. The fact of changing some parameters in the DEM code and generate automatically a geometry of the users' requirements becomes a useful tool for the simulation of every geometry of a fixed bed reactor. As for every CFD simulation but especially for the fixed bed applications with so wide range of geometries and contrary to the CFD experience of the discrete method, a validation of the results with experimental data should be made. Due to the complex geometry and phenomena in fixed bed reactors, it is extremely difficult to measure the fluid flow inside the bed by conventional means and without disturbing the packing arrangement (Nijemeisland, Dixon 2001). Non invasive experimental methods have been used to obtain the local flow patterns as laser Doppler velocimetry (LDV), marker bubbles and magnetic resonance imaging (MRI). Another method for the determination of mass transfer characteristics and heat transfer coefficients by heat and mass transfer analogy is the naphthalene sublimation method. (Motlagh, Hashemabadi 2008, Romkes S.J.P et al. 2003)

3 Geometry Generation

After the literature research and the advantages and disadvantages of the continuum method and the discrete method presented in Chapter 2, the following chapter will present the geometry generation for the 3D representation of the catalyst reactor with DEM.

3.1 DEM basics

As described in the second chapter, the Discrete Element Method is an explicit numerical scheme for the modelling of individual particles in discrete/particulate systems. The motion and position of individual particles or assemblies of them can be tracked in every time step according to the second law of Newton. The interactions between the particles, which define the forces applied on them and via Newton’s law track their motion, are of high importance. Especially the inter particle interactions vary from one DEM software to the other. When the method was first introduced by (Cundall, Strack 1979) the inter-particle contact forces were calculated based on springs, dashpots and friction sliders. In this model the interaction forces are computed when elements slightly interpenetrate each other and this force-displacement formulation is referred as “smooth contact” method or “force-displacement” method. This interpenetration or overlap, although it is unrealistic, is implemented in order to account for the relative deformation of the surface layers of the elements (Donzé, Richefeu & Magnier 2009). Alternative methods referred as “non-smooth” contact can be categorized in event-driven methods and contact dynamics. The even-driven method is not suitable for large simulations of many contacting particles, contrary to contact dynamics. More information about these methods can be found in (Moreau 1999) and (Luding S. et al. 1996). Dense isotropic arrangements of non-overlapping spheres in tetrahedral meshes which vary in size have also recently been developed for the generation of composite media.

3.1.1 Motion of particles

The particle’s translational motion can be described for individual particles based on Newton’s second law by the following expression:

$$m_i \frac{d\boldsymbol{v}_i}{dt} = \sum_{j=1, j\neq i}^{k_i} \left(\boldsymbol{F}_{c,ij} + \boldsymbol{F}_{d,ij}\right) + m_i \boldsymbol{g} + \boldsymbol{F}_i \qquad (Nt) \qquad \text{Equation 10}$$

Where m_i and $\boldsymbol{v}_i$ are the mass and the velocity of the i particle equivalently, $\boldsymbol{F}_{c,ij}$ the contact force between the particles i and j, $\boldsymbol{F}_{d,ij}$ the viscous contact damping force and $m_i\boldsymbol{g}$ the gravity force. $\boldsymbol{F}_i$ are any other forces that can act on the particle including electromagnetic forces or fluid particle interactions and will be neglected in the present work. The gravitational force acts on the mass center of the particle, whilst the inter particle forces $\boldsymbol{F}_c$ and $\boldsymbol{F}_d$ act at the contact point between the particles. In most DEM packages and also algorithms for granular media, spheres are used for the calculations as the simplest granular particle. As spherical geometry requires only definition of the radius and the contact points can easily be detected, it allows less complicated models and less computational requirements (Donzé, Richefeu & Magnier 2009).

3.1.2 Modeling Inter-particle forces:

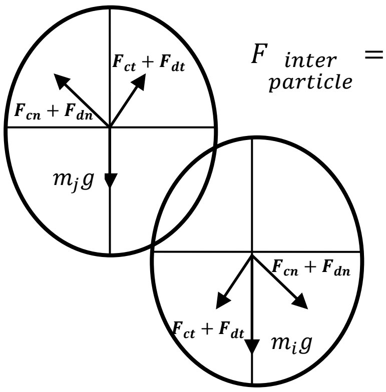
The modeling of the contact between spheres has been extensively studied in DEM and can be in general categorized in linear models, non linear models and non linear hysteric models (Xu, Yu 1997). According to (Pöschel T., Schwager T. 2005), two spheres which are in mechanical contact can be defined by the following equation:

$$\xi_{ij} \equiv (R_i + R_j) - |r_i - r_j| > 0 \qquad (mm)$$

Equation 11

if the sum of their radii is larger than the distance of their center. ξ_{ij} is called the mutual compression of particles i and j or displacement (Pöschel T., Schwager T. 2005).

The force between contacting particles can be described by the normal and tangential components as shown in Figure 22.



$$F_{\substack{inter\\particle}} = \begin{cases} F_{cn,ij} + F_{ct,ij} + F_{dn,ij} + F_{dt,ij}, & if\ \delta^{ij} > 0 \\ 0 & , \ else \end{cases}$$

Equation 12

Figure 23: Inter-particle forces in tangential and normal direction

When particles collide, a part of their kinetic energy is dissipated, for example transformed into heat. In granular particles simulation the shape of the particles is assumed to be conversed while the temperature change is neglected (Pöschel T., Schwager T. 2005). The linear-spring model of Cundall and Strack (Cundall, Strack 1979) is widely used due to its simplicity. In general form the two kinds of forces can be written as

$F_{c,ij} = K\delta^{ij}$ and $F_{d,ij} = \eta_i\, v$ (Nt) Equation 13

where K is the constant elastic coefficient, δ^{ij} is the displacement in the direction of the acting force, η_i is the normal viscous contact damping coefficient of particle i and v the relative velocity $v = v_i - v_j$ of the two particle's in contact. The normal contact forces, which occur at the center of the displacement along the line joining the centers of the two spheres, are modeled according to the spring or elastic constant normal coefficient and the normal displacement. The tangential forces or shear forces are mainly determined by the surface properties of the particle and are modeled in an intuitive way. As the normal forces are modeled by the compression or displacement factor, the modeled point of contact is exactly the point of contact for hard spheres and an approximation for deformable spheres (Pöschel T., Schwager T. 2005). Friction between particles can be simulated by the tangential forces and it can be either static or dynamic according to the defined parameters. The tangential contact force is analogous to the increment of tangential displacement, defined from the previous time step, and the tangential elastic constant coefficient, which is initialized at the time of the first contact until the particle's surface separate again. The tangential elastic coefficient has to be defined by comparison of experimental results with the simulation results (Pöschel T., Schwager T. 2005). For large normal contact forces or small relative velocities, the expression of tangential contact force is given by the following expression (static friction according to Coulomb's law),

$|F_{ct,ij}| \leq \mu_f |F_{cn,ij}|$ (Nt) Equation 14

Otherwise, sliding occurs (dynamic friction) and is given by the expression

$|F_{ct,ij}| = \mu_f |F_{cn,ij}|$ (Nt) Equation 15

If the contact forces are modeled only by the linear model then no energy dissipation will be calculated and the contact will be perfectly elastic. In reality, as mentioned before, a part of the kinetic energy during the collision is dissipated that leads to plastic deformation or heat or sound energy. This mechanism can be simulated by adding a viscous contact damping force. The viscous damping forces for particle i are proportional to the relative velocity of the two particle's in contact by the definition of the viscous damping coefficient η_i as described in equation 12. If sliding occurs then only friction damping is considered and the viscous damping is vanished (Cundall, Strack 1979). Many contact models have been in DEM including the linear model and the non-linear Hertzian model (Malone, Xu 2008) and many other more sophisticated models (Xu, Yu 1997). In the present work the linear model was described because of its implementation on the DEM code used. More details on the contact model used will be given in the relevant section of the code.

3.1.3 Modeling Boundaries

The interaction between the particles and system boundaries are of great importance for DEM simulations. For example, the material properties of the container, the motion of the container (vibrations or applications of axial compression), the shape, the size and the stiffness of the boundaries play an important role for the simulation. The wall roughness and the interaction with the particle's motion have to be modeled. In (Pöschel T., Schwager T. 2005) the representation of the wall boundaries by building up the geometry from particles is suggested. The geometrical shape of the surface of the wall also influences the interactions. A rigid wall imposes forces on the particle that normally would include elastic repulsion, friction e.t.c. The modeling of these interactions depends again on the model used in DEM. Many DEM applications implement interaction with stationary or movable walls with different models used and special consideration should be taken for the definition of the parameters. Periodic boundaries are also commonly used in simulations in order to reduce the computational time.

3.1.4 Modeling Bonded Particles

In many applications of DEM, the geometry of a sphere is not adequate, because there are cases in real applications where either the cemented materials are more of interest than granular materials or different shapes need to be simulated. The technique of bonding particles and defining different interactions for those bonds is often implemented to account for complicated geometries. The overall mechanical behavior of the bonded group of particles can be estimated through the collective contributions of the participating particles in motion, displacement, sliding and rotation (Donzé, Richefeu & Magnier 2009). Bonds of finite stiffness can be defined and another contact model should be implemented for the bonded particles' contacts.

A bonded-particle granular assembly and a cemented material should exhibit similar deformation and damage-formation processes under increasing load as the bonds are broken progressively and both systems gradually evolve toward a granular state (Potyondy, Cundall 2004).

3.1.5 Time step

The appropriate choice of time step in DEM simulations is of great importance for the accuracy and stability of the simulation as well as for the CPU requirements and computational time. In (Malone, Xu 2008), a study for the determination of a suitable time step for DEM is presented. Lower order schemes of time discretization such as central difference are often used in DEM applications. Many researchers have tried to define a critical value for the maximum time step based on the properties of the particles. This value is often correlated with the natural frequency of the equivalent mass-spring system and due to this fact it varies from one application to the other. The critical value of the time step is commonly obtained by the following equation (Malone, Xu 2008):

$$\Delta t \leq \mathrm{C}\sqrt{{m}/{K}} \qquad (s) \qquad \text{Equation 16}$$

where m is the particle's mass, K is the particle's stiffness and C is constant that can vary according to the simulation. From equation 15, the high importance of stiffness factor on the simulation is observed. The "smooth contact" method or "force-displacement" method allows the modeling of multiple particle contacts but if the normal interactions between the particles are needed to be stiff for the minimum displacement then a very small time step is required. The number of simulated particles also influences the time step as more contacts are inserted in the model. Many "smooth contact" methods allow softer interactions between the particles for less computational requirements and stable solutions by artificially increasing the duration of contact (Hoomans et al. 1996).

3.2 ESyS –Particle Software

ESyS-Particle is an Open Source software which implements DEM for the numerical modeling of granular particles. ESyS-Particle has been utilised to simulate earthquake nucleation, comminution in shear cells, silo flow, rock fragmentation, and fault gouge evolution. The origin of the software can be traced back to the SDEC software developed by Prof. Frederic Donze in the early 1990s at the IGP, Paris, France. A fore-runner code to ESyS-Particle, the Lattice Solid Model (LSM) was released as LSMEarth software. ESyS-Particle has been developed in-house within the Earth Systems Science Computational Centre (ESSCC) at the University of Queensland, Australia since 1994. After the funding of Australian Computational Earth Systems Simulator (ACcESS) Major National Research Facility and various implementations by many contributors, the latest version of ESyS-Particle v2.0 via Launchpad, was released in August 2009. More information about the development of the software can be found in (Weatherley Dion, 2008a) and (Weatherley Dion, 2009a).

3.2.1 ESyS features

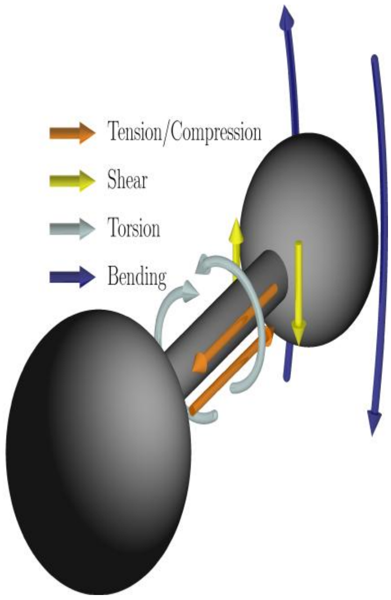
ESyS particle allows the user to perform parallel simulations via implementing the Message Passing Interface (MPI) to divide the spatial domain in parts where the C++ basic code is solved. This feature offers the opportunity of large simulations on parallel supercomputers, clusters or multi-core personal computers running a Linux-based operating systems. The code runs after an execution command in Linux terminator which calls the basic script written in Python. The Python wrapper Application Programming Interface (API) allows the modeling of DEM interactions and definitions of parameters in one basic and short script which interacts with the basic C++ code. The interface provides flexibility and simplicity in the designing of the simulation, in a short and organized way, avoiding the direct contact of the user with the huge basic code. In addition, Python is a useful programming language, object oriented and can “boost” libraries from other existing programming languages. A scriptable interactive visualization of the simulation by the use of POVray and VTK visualization tools for rendering simulation data and debugging purposes is offered to the user (http://www.povray.org and http://www.vtk.org). Except from the basic features and modeling of DEM, ESyS particle software provides its users with as a step-by-step tutorial on the basic principles and usage of the ESyS-Particle software and via launchpad.net a open forum for supporting the software’s development. ESyS documentation can be found in (Weatherley Dion, 2008b). As various information about the software is provided by launchpad, we will refer to the number of the posted question or faq as reference.

ESyS offers the possibility of implementing and modeling different kinds of the geometry based on spheres. The basic code offers the opportunity of implementing spheres or assemblies of bonded spheres. The bonded spheres in the basic code can be generated by four methods, the SimpleBlock, the CubicBlock, The Hexablog and RandomPacker (Weatherley Dion, 2009b). Each of these methods introduces different kind of packing for the block of bonded particles and the bonds between them can be formed by code fragment. Another way of generating block of particles for arbitrary shapes is by using the Lsm.GenGeo library of ESyS. This library fills the required geometry with spheres in a user-defined range and bonds them together. By running the Lsm.GenGeo script, a text file with the format .geo is created and the user has only to write some lines of code in the basic python script in order to insert it. The scriptable setup of the geometry inserted in the simulation is proved to be another useful feature of ESyS. In this point, the limitation of the porosity of this blocks or assemblies of particles should be pointed out.

The type of spherical particles existing in the software is rotational and non-rotational spheres. The non-rotational spheres do not have any rotational degrees of freedom while the rotational spheres can change orientation according to applied moments (Epydoc 3.0.1, 2009). For the two types of spheres and for bonded or not particles, different interactions can be applied in the simulation according to non-rotational and rotational dynamics. Gravity and damping forces can be implemented in the code for the two kinds of spheres, bonded or unbonded. Moreover, linear elastic repulsion as well as frictional interactions can be applied to both types. The bonded interactions for non-rotational spheres account only for the compression/tension or normal forces while the rotational spheres have a more sophisticated model which implements additional shear, torsion and bending forces. Interactions with the walls can also be implemented depending on the type of wall boundary used. The theoretical basis of these interactions lays on the following publications (Cundall, Strack 1979, Mora, Place 1994, Place, Mora 1999). More information about interactions can be found in the documentation (Weatherley Dion, 2008b), in the above mentioned sources and in the source code. Figure 23 presents the interactions available in ESyS-particle software.

<table>
<tr><th></th><th>NRotSphere</th><th>RotSphere</th></tr>
<tr><td rowspan="4">Body Forces</td><td colspan="2">Gravity</td></tr>
<tr><td colspan="2">Damping</td></tr>
<tr><td colspan="2">LinDamping</td></tr>
<tr><td></td><td>RotDamping</td></tr>
<tr><td rowspan="2">Unbonded Particle-Particle Interactions</td><td>NRotFrition</td><td>RotFriction</td></tr>
<tr><td>NRotElastic</td><td></td></tr>
<tr><td>Bonded Particle-Particle Interactions</td><td>NRotBond</td><td>RotBond</td></tr>
<tr><td rowspan="3">Unbonded Wall-Particle Interactions</td><td colspan="2">NRotElasticWall</td></tr>
<tr><td colspan="2">NRotElasticLinMesh</td></tr>
<tr><td colspan="2">NRotElasticTriMesh</td></tr>
<tr><td rowspan="4">Bonded Wall-Particle Interactions</td><td colspan="2">NRotBondedWall</td></tr>
<tr><td colspan="2">NRotBondedLinMesh</td></tr>
<tr><td colspan="2">NRotBondedTriMesh</td></tr>
<tr><td colspan="2">NRotSoftBondedWall</td></tr>
</table>

Interactions for the two types of spheres



Bonded Interactions for rotational spheres

Figure 24: Overview of the available interaction In ESyS particle (Abe Steffen, 2009)

For the boundary modeling of the simulation, three types of walls can be implemented, the planar walls, the linear 2D meshed walls and triangular meshed walls for 3D simulations. The planar walls can be directly implemented in the basic script code through ESyS while for the generation of a mesh on the wall surfaces external CAD packages are suggested. For the creation of a meshed wall, ESyS suggest the use of gmsh, an Open Source mesh generation software (http://www.geuz.org/gmsh/) and (Weatherley Dion, 2009d). The file which contains the generated mesh is converted to an .lsm format and loaded on the basic script by a code fragment. The interactions with the walls vary according to the type of the wall inserted. As there is no interaction for frictional effects when the particle is in contact with the wall, in the ESyS-particle tutorial there are two ways of including these interactions to the simulation. The first is to build a wall from particles also suggested (Pöschel T., Schwager T. 2005) or to bond some particles on the wall during the simulation.

The basic script starts as every program by importing the necessary commands that link the script with the libraries of the language. Classes and subclasses needed for the simulation are loaded by these modules. Python is an object based language, thus it requires the definition-creation of the simulation object. The number of MPI processes (serial or multi-processing simulations) and the division of the domain amongst the worker processes is required at this step. Definition of the type of sphere used and the algorithm for the dictation of contact points and frequency of updating the contact lists are also specified in the simulation object. Following the instantiation of the simulation object, the particles, the spatial domain and the wall boundaries have to be initialized and inserted on the basic script. The body forces and the interactions between particle-to-particle and particle-to-wall are defined in the next step according to the kind of simulation. The time step increment and the maximum number of time steps are specified as well as the method for the output data during the simulation. After the completion of the set up, the script is ready to run from the terminal of the operating system.

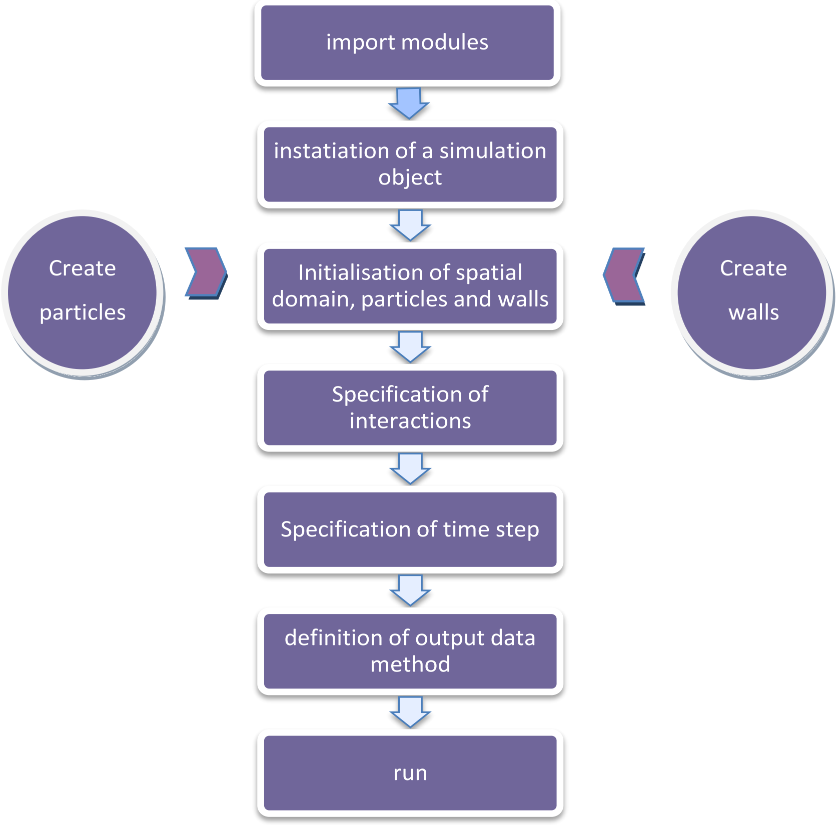

import modules
instatiation of a simulation object
Create particles
Initialisation of spatial domain, particles and walls
Create walls
Specification of interactions
Specification of time step
definition of output data method
run

Figure 25: ESyS-Particle simulation set up scheme

The methods from exporting data in ESyS particle can be summarized in text format files or in pictures via the visualization packages POVray or VTK. The kind of data that the user wants to export can be selected. A group of modules called FieldSavers provides a mechanism for the selectively output data of the simulation which can be post-processed easily.

3.3 Simulations in ESyS

The simulations performed with ESyS-Particle within the work of this thesis, contain simulations with spheres representing the catalytic particles of fixed bed reactor and bonded spheres representing cylindrical catalysts. The type of spheres used was non rotational sphere for both kinds of simulations. The initial parameter of the reactor was a particle to tube ratio of N=6 , where the dimensions of the reactor are a diameter of 60mm and height of 300mm and the catalyst's diameter for sphere or cylinder is d=10mm, and the height for cylindrical particles is 15mm.

For the representation of the falling particles in the tube, a time subroutine was used in order to insert the particles at different time steps into the tube. The script code for every case and simulation is described in the following sub-chapters. The input data for the simulation, except from the modeling of the particles interactions and physical laws, require the implementation of walls and the creation and load to the simulation of the particles. For the simulations with spheres, the creation and initialization of the inserted particles was done in the basic script of ESyS simulation. However, for the bonded spheres, representing cylinders, the creation of the spheres was done with external packages. Lsm.GenGeo is an external code-library of ESyS that allows sphere's packing in arbitrary shapes. Lsm.GenGeo runs separately from the main script, rendering text output files with the format .geo. These files contain all the basic information that ESyS requires to insert particles in the simulation. Although Lsm.GenGeo is a very helpful tool, it contains a lot of random parameters with the basic drawback to be the random definition of the spheres' identification number (id). The sphere's id in a packed geometrical shape plays an important role for the later algorithm used in order to create the geometry in Gambit automatically. Two basic spheres in the packing define the vector of the cylindrical particle and if these spheres do not have a certain id and relative position, then it is not possible to define the boundaries and the position of this packing-cylinder. Hence, for the creation of bonded spheres representing cylindrical particles, the format of the .geo file (Weatherley Dion, 2009c) was used but it was created from a pre-processor self-made code in Python. The number of particles $n,$ inserted was defined proportionally to the fixed bed's porosity $\varepsilon,$ according to equation 16. The porosity of the packing in DEM is a controversial theme and comparisons with experiments and with the expected porosity is necessary.

$$n = \frac{V_{reactor}}{V_{particle}} \cdot (1 - \varepsilon) \qquad (number\ of\ particles) \quad \text{Equation 17}$$

where $V_{reactor}$ is the volume of the reactor, $V_{particle}$ is the volume of the simulated particles and $\varepsilon,$ is the porosity of the reactor. For the simulations performed the value of the porosity was determined to be equal to 0.4.

For the implementation of the reactor's geometry in the DEM simulation, three different kinds of triangle meshes were created in gmsh meshing software according to the procedure described in (Weatherley Dion, 2009d). The three different meshes represented the cylindrical surface of the reactor and varied from a coarse mesh to more refined. A case study was also made for the choice of mesh to be created. The file from gmsh with the format .mesh was converted via a python script provided in (Weatherley Dion, 2009d) to the file with format .lsm and was imported to the basic simulation script. A floor plane wall was inserted directly in the basic code as well as a top wall to prevent the colliding particles to leave the spatial simulated domain of the reactor. The geometry was designed in millimeters units. More information about the created files and the code used can be found in Appendix.

The types of interactions used in the simulations performed, include gravity, non rotational elastic repulsion and friction for the non bonded spheres and non rotational elastic bonds for the tension or compression between the bonded packed spheres. The code used in ESyS for these interactions is included in Appendix I. The damping forces were not included in the final scripts. The reason for this, was that certain problems occurred after the insertion of the damping parameter and the solution did not remain stable. The simulation without damping is not realistic but for the purpose of the use of DEM in this study, it can be excluded. The particles without damping collide continuously but as the time increases more particles are inserted in the simulation and packed so they force the first one inserted to calm and stabilize.

The time step increment and range played an important role for the simulations performed. Only for the simulation of catalyst spheres, 972 spheres were inserted in the simulation to follow the porosity terms. This amount of spheres requires more effort for the detection of the contact points between them, thus the contact time is artificially increased and many particles seem to overlap. A higher stiffness value for a more hard approach of the contact requires smaller time steps. During the simulations performed, variations of elastic stiffness values and time steps in a wide range were implemented to deduce the optimal case. By keeping the time step increment constant and increasing the elastic stiffness, particles seemed to disappear from the domain while by the opposite way, the particles were not even inserted to the simulation. Suggestions about the interaction of elastic stiffness with the time increment for ESyS simulations can be found in (Weatherley Dion, 2009e) by the contributor Dr. Weatherley Dion where the suggested correlation is given by the following equation

$$dt < 0.1\sqrt{\frac{m_{smallest}}{K_{max}}} \qquad (s) \qquad \text{Equation 18}$$

where dt the time increment , m the mass of the smallest particle and K the largest elastic stiffness used in the model. Other suggestions by users can be found in (Peinado Martin Diego, 2009).

The mathematical implementation of ESyS-Particle is non dimensional and the definition of the units depends on the user's requirements. The modeling units in the simulations were mm as the mesh was designed in mm and very small values of particles' radius were not acceptable when the basic set up was performed. Suggestions about the units can be found in (Weatherley Dion, 2009e).

After the experience gained with ESyS simulations for the initial values of the various elastic stiffness factors, the increment of time step and units, some basic considerations can be made. The units should be determined according to the simulation requirements and mathematical modeling of the application. After the determination of the units, step by step simple simulations should be performed starting with one sphere. From the one sphere simulation, the initial value of the wall's elastic stiffness can be defined according to the suggested equation in (Weatherley Dion, 2009e) and (Weatherley Dion, 2009b):

$$K_{wall} > {}^{m \cdot g}/_{r} \qquad (N/m) \qquad \text{Equation 19}$$

From this initial value, the time step can be corrected according to equation 18. A simulation with only two spheres will then define the initial value of the interactions for unbonded spheres and the time step be corrected again if necessary. Furthermore, a simulation with bonded spheres can be performed in order to define the elastic stiffness of the bonds according to their overlap factor. The results showed, that the initial values of elastic stiffness for bonded and unbonded particles are approximately of the same scale. The time increment can be corrected again, if necessary, and frictional effects can be implemented afterwards. These suggestions are made for the specific type of modeling interactions used in the present thesis and account for the initial values for the set up. These values can change completely, if more particles are inserted and all the models are used. It is also necessary to point out the lack of the damping factor in the suggested procedure. The initial values for the set up performed in this work contained the damping forces but due to stability reasons, they were omitted in the final script.

The methods used for the output data of the simulations were the POVray visualization package and a module called CheckPointer, which belongs to the FieldSavers modules. For every simulation, different time steps were used for rendering images and zoom factors were modified in the runnable script of POVsnaps provided in (Weatherley Dion, 2009b). VTK is recommended for large simulations. The CheckPointer module saves certain data in specified time steps in a text format. Two kinds of text files are generated in the specified time step. The text file with the format timestep_1.txt contains information about the current position, the radius, the id, the mass, the velocity, the acting force of the particle in the current time step as well as values of these parameters for the previous time step, the meshed walls properties and the bonded particles. The user can specify the desired time steps at which the data will be exported in these txt files. The series of the data exported plays an important role for the post processing codes created for the work presented in this thesis.

3.3.1 Simulations with Spheres in ESyS

For the simulations with spheres, the radius of the sphere was defined to be equal to 5mm in order to retain the tube to particle ratio N=6. The number of spheres that would enter the tube was given by the equation 17 for the porosity $\varepsilon = 0{,}4$

$$n = \frac{V_{reactor}}{V_{particle}} \cdot (1-\varepsilon) = \frac{\pi R^2 H}{{}^4\!/_3\, \pi r^3} \cdot (1-\varepsilon) = 972\, spheres$$

where R is the reactor's radius $R = 30mm$, H the reactor's height $H = 300mm$, r the particle's radius $r = 5mm$.

All the particles had the same radius and the search algorithm for the contact points was defined according to that. The spatial domain was defined to be a cube with the following dimensions (-30,-30,0)÷(30,30,310). The particle's were created and initialized in the basic script of ESyS simulation. The particles' mass was defined to be equal to 1 and they were given an initial velocity of 1000 mm/s in the negative z direction of the reactor in order to accelerate the fall of the particle's as the small time step would give a very slow fall. The walls consisted of one meshed wall for the cylindrical surface and two plane walls for the top and bottom of the reactor. The gravity was set to 9810 mm/s^2 and no damping effects were included in the full simulation. The interactions with the wall were defined to be NRotElastic for plane or meshed walls and the value of elastic stiffness was obtained according to the above described procedure and to equation 18. In the simulation with spheres, no bonded particles were needed. Elastic repulsion and frictional resistance between touching spheres were modeled by the NRotFrictionPrms where the final values of the normal and shear elastic stiffness was 10^7 and 10^5 equivalently and the friction coefficient was 0.7. These values of elastic stiffness in the wall to particle and particle to particle interactions allow the filling of the tube with all the 972 particles inserted for a time step increment of 0.0001 s. Lower values lead to the loss of some particles while higher values require a smaller increment.

The particles were inserted in random positions on the top of the tube. Every 100 time steps a new particle was inserted and after 300,000 time steps the results showed an approximate quasi static solution. As the simulation reached the end the particles first inserted seemed to have an almost stable position while the last inserted on the top of the tube were still moving due to the lack of damping. The duration of the simulations varied according to the time step increment, the values of the elastic stiffness, the insertion or not of the damping forces and the refinement of the meshed wall. Figure 25 presents the results of the mesh case study for the simulations with spheres. In simulations with coarse mesh, the particles are packed in way that the only contact with the wall is observed in the quaternary points of the cylindrical surface.

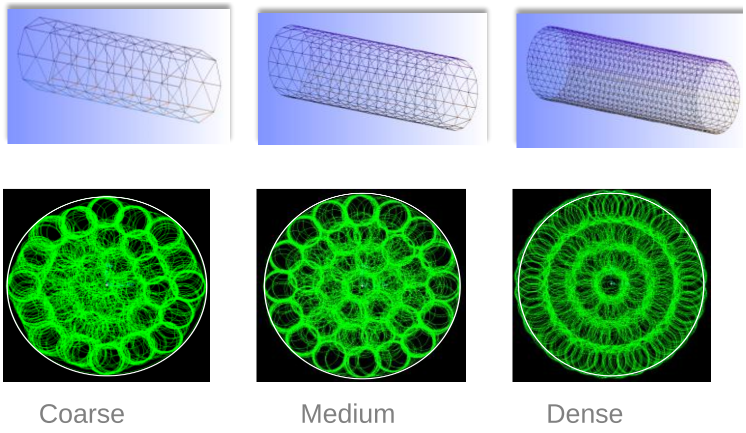


Figure 26: The three different kinds of meshes used in ESyS and the results after the simulations with spheres

The dense mesh provides a very good contact with the walls but the spheres are extended even for a small portion out of the reactor's wall boundaries. The elastic stiffness of the wall also influenced the number of the particles that exceeded the wall boundaries and for the three meshes it was kept stable. The duration of the simulation plays a very important role to the choice of the mesh. The simulations were performed in a Workstation with an Intel processor of 3.0GHz and 2.5GB memory. For the coarse mesh the simulations lasted for 24 minutes, for the medium mesh lasted for 40 min and for the dense mesh lasted for 120 minutes. After the conclusions of the mesh case study, the medium mesh was chosen and considered to be adequate for the simulation requirements. In ESyS-Particle, there is a bug for meshed walls, which reported from many users and observed on the simulations performed. This bug is the cause of hanging spheres, that means that some particles are attached on the walls and retain a stable position at the point where they touch the wall. The script used for the simulations with spheres is provided in Apendix . Figure 26 presents the results obtained from the simulation.

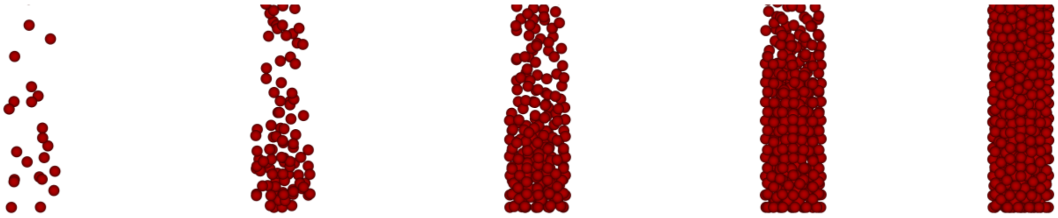


Figure 27 : The full DEM simulation with 972 particles

After the results obtained with 972 particles, a second script with spheres was written and run in ESyS-Particle. The purpose of the second simulations was to overcome the overlap between the particles and to give more exact results. For these reasons several simulations were performed where the particles inserted in the reactor in groups. Every scenario used the coarse mesh and the damping forces. The particles of every group were inserted in different time steps ranging from 100 to 200. After 200.000 time steps, the particles had a stable position. The results of the particles positions where inserted as input data to the next simulation where they were set as non dynamic particles. That means that they behaved like a stable wall where the next group of particles was crashing on. As less particles were inserted in every simulation the algorithm with the same time increment and elastic stiffness predicted better the particles positions. A smaller overlap was obtained with this method, the maximum value of which was 0.033. Although, with this method, the porosity ratio changed as 775 particles were finally inserted. This fact is justified by the non dynamic behavior of the lower layers of particles as their position was held stable when another particle crashed on them. This script will be referenced as less spheres. Figure 27 presents the results of the full simulation in different time steps. The images were rendered with POVray.

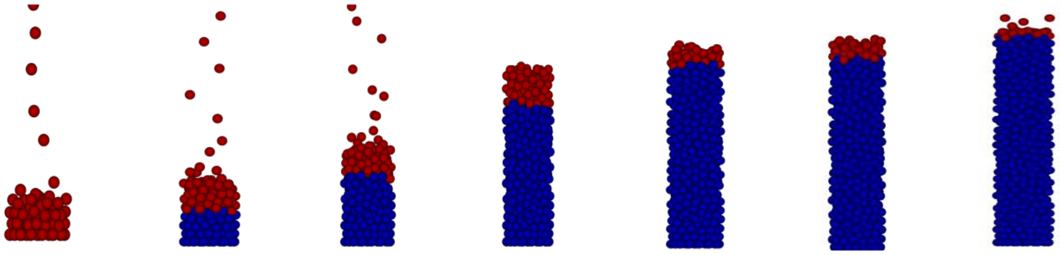


Figure 28: The “less spheres” simulations

In both simulations performed with ESyS-Particle problems occurred with hanging spheres on the meshed wall. This bug can also contribute to the overlaps observed on the results of all simulations performed. For the script of less spheres, the hanging spheres were detected easily and the simulations run repetitively until there was no sphere attached on the wall. When hanging spheres occurred, they were omitted from the input file of the next simulation and fall again in the reactor. Moreover, the coarse mesh was chosen for the simulations as it was observed that less spheres where attached to the meshed wall surface than to the one with denser mesh.

3.3.2 Simulations with Cylinders in ESyS

- Two spheres representing one cylinder

As mentioned before for the bonded particles, the spheres were inserted in the spatial domain in files with the format .geo. These files were generated by a code created in python language and contained the necessary data for the ESyS-Particle code simulation. The code for the two particles created particles in random positions at the top of the reactor and is presented in the Appendix. As the spheres had a radius of 5mm, the cylinder that they represented had a radius of 5mm and height of 20mm. The number of the particles inserted was calculated according to equation 16 for the porosity $\varepsilon = 0{,}4$:

$$n = \frac{V_{reactor}}{V_{particle}} \cdot (1-\varepsilon) = \frac{\pi R^2 H}{\pi r^2 h} \cdot (1-\varepsilon) = 324\, pairs\ of\ spheres$$

where R is the reactor's radius $R = 30mm$, H the reactor's height $H = 300mm$, r the particle's radius $r = 5mm$ and h the height of the particle $h = 20mm$.

All the particles had the same radius and the spatial domain was defined to be a cube with the following dimensions (-30,-30,0)÷(30,30,310). The particle's information was stored in the files created by the python code and a read file command in the main script inserted them in the simulation at different time steps. The particles' mass was defined to be equal to 1 and they were given an initial velocity of 1000 mm/s in the negative z direction of the reactor as described for the spherical particles before. The walls, the wall interactions, the gravity and the interactions between unbonded particles was defined exactly as the in the script for spheres. The bonded interactions were modeled by the NrotBondPrms where the elastic stiffness of the bonds was equal to 10^7 and the break distance equal to 500. An increment of 0.00001s for 300,000 time steps was determined to be adequate for the simulation. The best value for the bonded elastic stiffness was proven to be equal to 10^9 but as more particles were inserted in the simulation as well as the interactions with unbonded particles, this value decreased resulting in an increased overlap between the particles.

Different scripts were run for the bonded particles in order to take into consideration the effects of damping forces. The effect of damping as mentioned in the basics of DEM is omitted when sliding occurs. The number of the time steps passed after the next pair of spheres is inserted in the simulation plays a very important role. If the particles are inserted at every 100 time steps, they collide intensively with other particles and they crash quickly on the floor when damping is omitted. In this case, if the damping forces are included, they decrease the velocity of the particles. If the particles are inserted every 5,000 time steps (the overall number of time steps increases also) then the inclusion of the damping forces simulates the air resistance as the particle falls on particles that have reached already an approximate composure. In the final simulations the damping forces were included and the particles were inserted every 10,000 time steps.

The number of time steps passed after the next pair of spheres is inserted in the simulation influences also the packing of the spheres. When the particles have time to reach composure, then the packing is denser compared to when they are colliding intensively. The porosity obtained from the simulations is different from the one calculated due to the bonds of the two spheres. In general, it is preferable to represent the cylinder with more spheres.

- Three spheres representing one cylinder

A code for three particles representing one cylinder was made. The simulations showed that the three bonded spheres do not have stable contact points, hence the particles kept in contact but their centers did not belong in one vector-line. The applied bonded interaction forces are only in the normal direction for bonded particles and the bonded spheres could rotate during the movement.

- More spheres representing one cylinder

If more spheres are used for the representation of one cylinder, the contact points between the cylindrical particles can be predicted more accurately as more spheres are used for the representation of the boundaries of the cylindrical surfaces. Lsm GenGeo is used for filling cylindrical and other shapes with spheres but as mentioned before it can be used for due to the random particles' id. An effort to generate by hand a cylinder filled with spheres was made using AutoCAD 2006 software as shown in Figure 28. The cylindrical particle had the exact dimensions requested (diameter of 10mm and height of 15mm) and the spheres were packed in it. A minimum number of spheres and sizes was tried to be achieved with this packing in order to avoid additional needed computational time, power and to retain stability of the simulation. Three large particles with certain ids define the vector of the cylinder that would be used in the C code. The simulations performed with this geometry showed that this geometry is deformed when the spheres crash on the floor. This fact relies on the different forces applied on the spheres which have not the same shape and mass. Additionally, the bonds seem to break because neighboring spheres are not touching each other, thus the bonds created for shapes with packed spheres should only contain touching particles. Moreover, the mass of the particles is different as it is calculated with a default density and proportionally to the radii of the particles resulting in different forces applied on the particles. If the different densities are set in order to obtain the same mass of the particles then a cylinder comprised of granules of different materials is modeled.

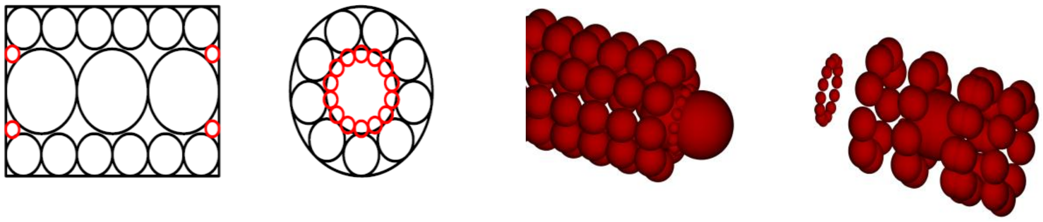


Figure 29: More spheres representing on cylinder .The geometry and the results of the simulation

3.4 Codes for creating the journal file

Gambit exports three kinds of data files in order to store the session operations performed for the geometry and mesh generation: The binary file with the format .dbs, the text file with the format .trn and the text file with the format .jou. The text file with the jou format, named as journal file, contains a sequential list of geometry, mesh, boundary zones and tools commands that have been executed during the session. Gambit offers the opportunity with the journal file to run the same script and commands as performed during the session.

Having copied the necessary commands that Gambit uses to generate the geometry of a fixed bed catalyst, different codes in C language were generated in order to create the journal file and store in it the session operations. The code is used for the connection between DEM results and the designing software Gambit. By obtaining the DEM final data and running the codes in C, the journal file is automatically created and by running the journal file the geometry is automatically created. The procedure is represented in Figure 29.

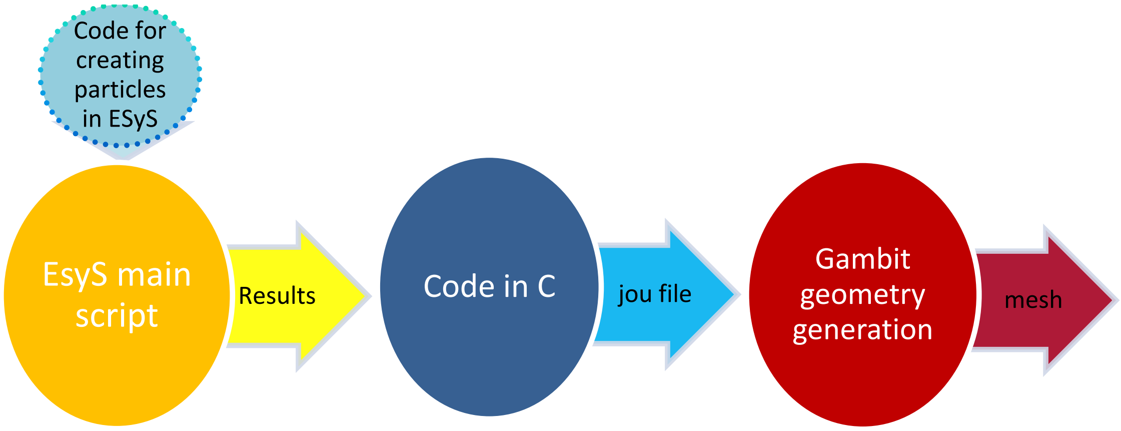

Code for creating particles in ESyS
EsyS main script
Results
Code in C
jou file
Gambit geometry generation
mesh

Figure 30: Outline of the automatically generated geometry of a fixed bed catalyst

Different codes were generated according to the simulation of spheres or cylinders and for different scenarios and meshing techniques. At this point the basic outline of the generated codes will be presented.

The data text file obtained from ESyS particle contained the number of spheres at the first line and the following information about the particles in the in exact serial order presented here for 22 columns.

Current	Radius	Id	tag	mass	Initial	Previous	Velocity	Net acting
Position					position	position		Force
x y z					x y z	x y z	x y z	x y z

The file created by the Checkpointer module, contained also information about the walls inserted in the simulation as well as the bonded particles that were neglected in the post processing procedure. The data were not stored in rows according to the time inserted in the simulation, to the bonds created or to their id. In the case of the bonded particles, it is important to obtain the data of the pair or group of bonded spheres and to know exactly which of them were bonded.

The code created in C language reads line by line the text output file from ESyS-Particle. The first line when read, stores the number of spheres simulated to the variable *spheres* and the information table is stored line by line to one array. Afterwards the array is sorted according to the id of the particles by a sort loop. Moreover the journal file is created and opened for writing the necessary commands. The journal file was a standard format for the header and then the commands for the necessary session operations follow. These commands were copied from Gambit while performing by hand the geometry generation. The commands and their specific form can be found in the Appendix where the codes are presented or either at the output file of the codes, the journal file, and hence no further reference will be made for this matter. As mentioned before different techniques were used to create the geometry. The basic outline of the code is presented in Figure 30.

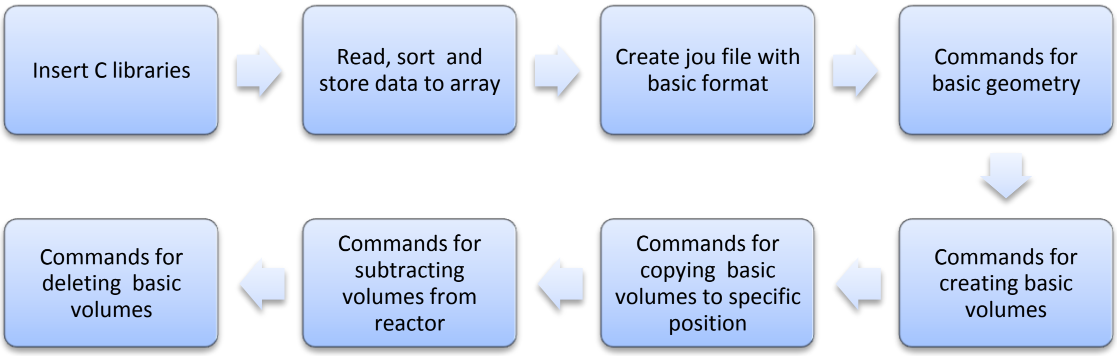

Insert C libraries
Read, sort and store data to array
Create jou file with basic format
Commands for basic geometry
Commands for creating basic volumes
Commands for copying basic volumes to specific position
Commands for subtracting volumes from reactor
Commands for deleting basic volumes

Figure 31: Basic outline of the created code in C for the automatically generated geometry

The two strategies followed for the set up of a simulation for a fixed bed can be summarized to those who model the volume of the particle (scenario A) and to those who neglect the volume and model only the surface (scenario B). Either if the “near-miss” model is used or any other solution technique the decision of whether the whole particles volume will be simulated or not is crucial for the geometry generation and mesh creation. Different parameters and models can be applied in the CFD simulations if the particles' meshed volumes with solid material properties are included or only the particles' surface on which the fluid is flowing.

The contact treatment plays also an important role for the mesh generation. The particle to particle and particle to wall contact, if included, needs many considerations and implementations in the code and in Gambit. Figure 31 presents the highly skewed elements that occur with the particle to particle contact for one small part of the reactor.

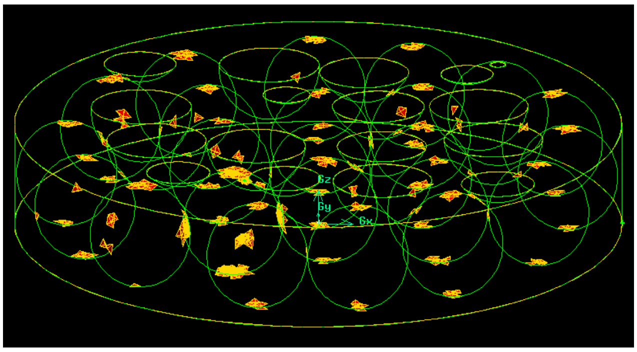


Figure 32: Highly skewed elements due to the particle to particle contact

3.4.1 Techniques for particle to particle contact

From the final output file of ESyS-Particle, some spheres inevitably overlap while some others may have one contact point or no contact point at all. In order to avoid the highly skewed cells except from the "near-miss" model, an alternative technique is suggested in this thesis. The approximation of contact points with smaller volumes than the basic particles. These smaller volumes can be spherical as the shape of sphere presents a smooth surface that prevents the creation of sharp edges.

For the application of this alternative strategy, a case study in Gambit has proven that it can be feasible. A solid volume accounting for one contact point could present completely different flow profiles and influence a lot the pressure drop, hence careful and extensive examination of the effects caused in the flow regime should be made. For the geometry generation of spheres an analytical case study is provided. The results from the DEM simulation reveal that 2372 spheres have a smaller distance than the distance of their radius. This overlap can be omitted by reducing the spheres radius proportional to the maximum overlap but this action would result in no contact points or one or two contact points in the overall reactor’s volume. This approximation is the same as used in the “near miss model”, hence it will be called “near-miss” model (technique1). The factor of shrinkage is based on the user’s requirements if it is desirable or not to have overlaps and also on the mesh and computational requirements. Different cases should also be examined and the effect of the generated mesh in the flow regime. The representation of overlaps, of the one contact point and the replacement of a small gap between two spheres by one sphere is the second technique that will be used in present thesis (technique 2).

Some basic conclusions for the technique 2 will be presented at this point. When a volume is subtracted, united or split from another volume in Gambit dublicate volumes, faces or edges may occur. A Cleanup tool is used in Gambit to identify the identical entities that are located at the exact same position by a tolerance criterion. The options provided are the virtual connect and the delete of these entities. Since the presence of virtual entities is not desirable, special treatment should be made to the creation of the geometry. After the case study for two spheres that overlapped , had one exact point and a relatively small gap between them, the results showed that the sphere volume should be inserted only for spheres that had distance equal or larger than the distance defined by the sum of their radii. Only for this case the geometry did not present virtual edges and could be meshable. The definition of the sphere’s radius plays also an important role. There is a critical radius from which virtual edges occur in the geometry if the sphere is subtracted from the basic particles or from the reactor’s volume. For the catalytic particles simulated with radius equal to 5mm, the critical radius of the model sphere that met these criteria was equal to 1mm for the minimum distance applied (the one contact point). To avoid the generation of high skewed elements this model sphere could be applied to the small gap created between two spheres. As the small gap increases, the permitted limits of the model sphere’s radius decrease. For example for a gap of 0.1mm the critical radius is 0.8mm. The range of the small gap that will be represented by one sphere should also be examined according to the same criteria. A smaller radius of the model sphere is desirable as less solid volume in the flow path would have less influence in the flow profile. Figure 32 presents the result of the case study for the application of the model sphere.

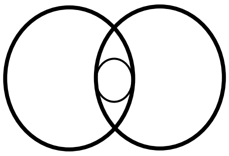
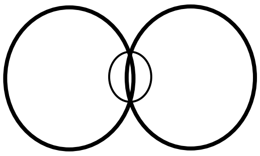
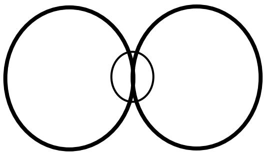
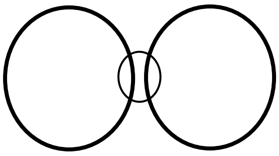
Overlap larger than the model sphere's radius	Overlap within the boundaries of model sphere	One contact point	Small gap between two basic particles
			
No meaning of model sphere	Virtual edges occur	Critical radius for model sphere 1mm	No virtual faces and meshable geometry
Method: Subtract basic particles without model sphere	Method: Subtract basic particles without model sphere	Method: Scenario A Scenario B	Method: Scenario A Scenario B

Figure 33: Results from the case study for the spheres technique 2

For the contact between cylindrical particles, a case study which is provided in the Appendix has proven that the model sphere's technique can obliterated the highly skewed elements at the contact points. As shown in Figure 32, different positions between the cylinders were studied. When the cylinders have one contact point in their cylindrical surface, the model sphere has a critical radius over 1mm. This condition applies for the best case of 45° degrees angle as well as for the worst case of 5° independently of the height that the contact point lays on. The model sphere should be subtracted from the two cylinders and retained. This is crucial for the mesh generation as if the opposite is chosen the resulting model sphere volume is too small for meshing and virtual entities occur. For cylinders, which have contact in the longitudinal direction, an artificial overlap can be created in order to avoid the highly skewed elements. The particles in this position can be either united or subtracted before subtracting from the reactor. In the case where two particles have common circular faces, no meshing problem occurs. In any other case than those presented at this point, the particles in touch should be united before the final subtract from the reactor.

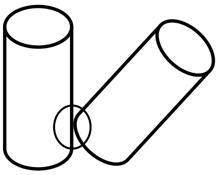
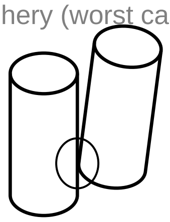
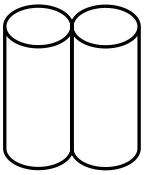
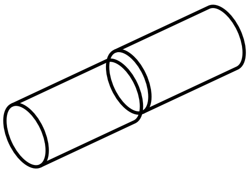
Contact in the periphery (best case)	Contact in the periphery (worst case)	Contact between parallel surfaces	Contact at the top or bottom
			
Critical radius for model sphere over 1mm	Critical radius for model sphere over 1mm	Artificial overlap	Unite and subtract
Method: Scenario A Scenario B	Method: Scenario A Scenario B	Method: Scenario A Scenario B	Method: Scenario A Scenario B

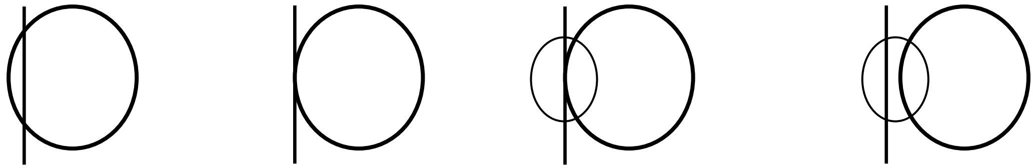
Figure 34: Results from the case study for cylinders technique 2

The model sphere technique was not applied for the cylinders as the geometry obtained from the DEM simulations contained particles which overlapped with each other. Although the case study showed that a mesh can be created with no highly skewed elements.

3.4.2 Techniques for particle to wall contact

The wall treatment for avoiding virtual faces and edges was also based on model spherical particles. A case study for the contact with the wall was made as shown in Figure 34. The DEM results also presented spheres that were outside the wall boundaries as well as spheres that contact the wall or have small gaps. With the code that generates the jou file, the user is able to change the wall boundaries according to the wall treatment. Precisely the wall’s radius can vary, in small ranges, according to the spheres included in the wall boundaries. Particles that have one contact point with the wall present many highly skewed elements. In order to overcome the problem, model spheres were used to simulate the contact. The critical radius of the model spheres was found to equal to 1.5mm in order to avoid virtual entities and to generate a mesh free of highly skewed elements. The critical distance-gap between the basic particle and the wall for which the model spheres are needed, was found to be equal to 0.11mm. For larger gaps no highly elements occurred and there was no need for model spheres.

Particle outside the wall boundaries	One contact point	One contact point Model sphere	Small gap between two basic particles



Change reactor's radius	Highly skewed elements	Critical radius for model sphere 1.5mm	Critical distance d=0.11mm
Method: Scenario A Scenario B	Method: Scenario A Scenario B	Method: Scenario A Scenario B	Method: Scenario A Scenario B

Figure 35: Results from the case for wall treatment

Independently of the scenario A or B, the model sphere had to subtracted in a certain way in order to meet the requirements of good mesh quality as shown in the Figure 35. If the basic sphere was subtracted from the volume of model sphere, then highly skewed elements still occurred. The procedure followed is summarized in the following steps:

- Find the basic spheres that have a gap between the wall and their radius in range of 0 until 0.11mm
- Create a model sphere, the centre of which will be located on the half of this distance.
- Split (with the wall's face) the model sphere in two parts and delete the one that is out of the wall boundaries.
- Subtract the half model sphere from the basic particle
- Create a face from the generated edge and split, with this face, the half sphere model in two parts as shown in the last picture of Figure 35.

The last step allows different modelling of these two volumes. Volume 2 as shown in the Figure 35 will be defined as solid and have all the properties like the basic particle while Volume 1 can be defined as solid or fluid according to the user's requirements. If the last step is omitted then the wall sphere should be defined all as solid according to the second picture in Figure 35. The procedure is included in the code created and the model spheres for the wall contact are automatically inserted. Special attention should be given to the number of volumes, faces and edges created in every case that the code is applied in order not to obtain any errors in Gambit.

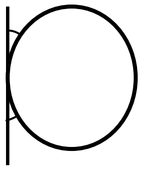
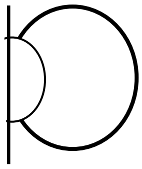
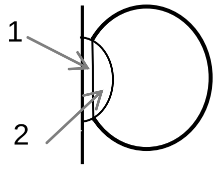
Subtract basic from the model sphere	Subtract model sphere from the basic sphere	Split the Model sphere in two parts
		
Highly skewed elements	Highly skewed elements	Critical radius for model sphere 1.5mm
Method:	Method:	Method:
------	Scenario A	Scenario A
------	Scenario B	Scenario B

Figure 36: Model sphere for walls

4 Results

The meshes and codes used to generate the geometry with catalytic spheres and cylinders and the results from every scenario and method technique will be presented here. The exact codes can be found in the Appendix as only the conclusions from the simulations will be analyzed.

4.1 Geometry generation with catalytic spheres

The results obtained from the DEM simulations for spheres can be found in Appendix. Three different techniques with both the two scenarios were used for the geometry generation of the fixed bed reactor with spherical catalysts.

4.1.1 Mesh1: Overlapping spheres

The results as obtained from the DEM simulation contained spheres that overlapped in their contact points. The geometry generated with the code Mesh1Overlapping and the jou file created, presents the exact results without any modification in the spheres radii. The contact with the wall is simulated by the insertion of one model sphere of 1.5 mm radius. With the reactor’s radius equal to 30mm as designed in the gsmh software, the particle spheres were located out of the wall boundaries, so the reactor’s radius was modified to 30.5 influencing also the N ratio. If the simulation runs with some of the spheres to exceed the reactor’s boundaries small circular surfaces occur in the reactor’s geometry resulting on unrealistic geometry in one point and creating mesh problems on the other. For this reason the reactor’s radius was slightly increased to obtain all the particles within the wall boundaries. The division of the reactor’s volume in smaller parts is necessary for meshing the volume and for the periodic boundaries for the full simulation of the reactor. The geometry constructed reveals that many highly skewed elements occur at the location where the division occurs. Another problem with the overlapping spheres is that some of the spheres have a very small overlap, which leads to the creation of small circular surfaces on the surface of one of the particles in contact. The small circular surfaces and edges cannot be meshed as the number of nodes created on the small edge is too small, thus a surface mesh and following a volume mesh cannot be created also. For larger edges, the mesh can be created but many highly skewed elements occur. In both the two scenarios used, the spheres in contact have to be subtracted from each other (this is not necessary in scenario B) and then from the reactor's volume. This fact reveals that independently of simulating the particles volumes or not, small edges will occur and a mesh cannot be obtained for such geometry. Moreover these edges create virtual faces for the scenario A where the volumes are retained. A mesh with these considerations is not able to be created for the scenario A or B. If the user would accept the overlap as realistic then the technique of model spheres could be used only for the scenario B. In this way larger edges and surfaces would occur while the simulation would impose an additional small artificial solid volume between the particles. Figure 36 presents the results from the geometry and the mesh generation in Gambit for the meshing technique of overlapping spheres.

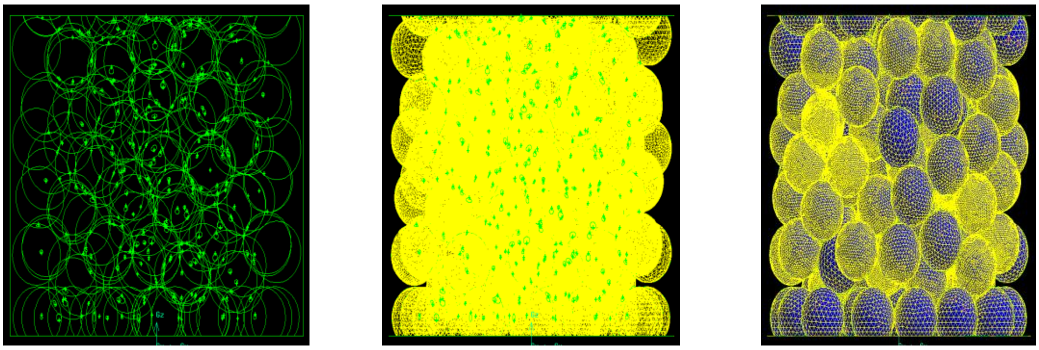


Figure 37 : Geometry and mesh generated for the overlapping spheres technique

4.1.2 Mesh2: “Radius Reduced” model

In this case study, the radius of the sphere was reduced according to the maximum overlap occurred. The spheres did not have contact anymore. This study can be similar to the “near-miss” model used with the difference that the spheres did not have the same constant gap between them because the same factor of shrinkage was applied to every sphere but their overlap was different in every contact case. This study can be also similar to (Jafari et al. 2008) where the spheres did not have contact at all.

The radius of the basic particle was modified to be equal to 4.64mm. For the scenario A, no virtual entities occurred and the geometry was able to be meshed. The geometry of the reactor was divided in parts of 100mm. Only for the void volume 2,611,413 cells were created with an interval size of 1. Among them, there were 7 elements that exceeded the skewness of 0.97 EquiAngle. These highly skewed elements occurred on the surfaces where a small gap was introduced as the factor of shrinkage was defined according to the maximum overlap of contacting spheres. If the sphere model is introduced to this pair of contacting spheres or the radius is slightly more reduced, then no highly skewed elements occur. If an interval size of 2 is used for the mesh generation then the reactor’s void meshed volume is composed of 463,422 cells among which 8 elements have skewness over 0.97 and occur at top circular wall surface of the reactor. If the sphere model is introduced at these wall boundaries then no highly skewed elements occur. Figure 37 presents the results from the geometry and the mesh generation in Gambit for the meshing technique of spheres with reduced radii.

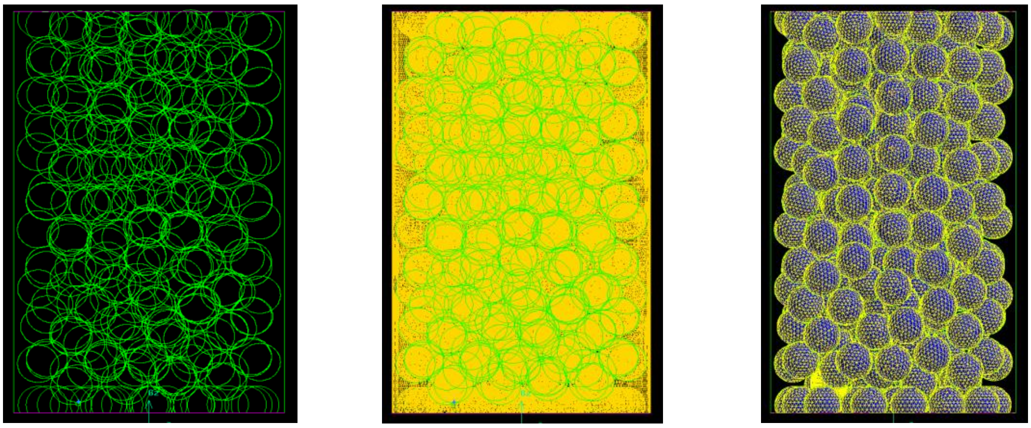


Figure 38 : Geometry and mesh generated for the radius reduced technique

4.1.3 Mesh3: Spheres model

Since the results from the DEM simulation cannot be modified to give only one contact point with every sphere, the overlap in the geometry generation is inevitable if the radius of the particles is not reduced. In Chapter 3, Figure 22 the critical radius of the model sphere implemented was defined to be equal to 1mm and no models spheres would be used to “cover” the overlap. In order to avoid the mesh problems created by the overlap of the spheres mentioned for Mesh1, a model sphere with critical radius 1.5mm was created.

For the spheres that did not overlap and had a gap of 0.1mm between them, a model sphere with 1mm radius was inserted between them. This sphere will be defined as solid and will result in the artificial addition of a solid volume of 0.623 mm^3 in the worst case of 0.1mm gap. In the case of overlap, the model sphere of 1.5mm will insert a solid volume of 1.59 mm^3 between the overlapping spheres, which will be continuously decreasing as the overlap increases. The overlapping distance defined the range where the second model sphere was implemented. The very small edges that prevented the mesh generation were found to be for a 0.2mm overlap. As a result the second model sphere was implemented for distance between two spheres ranging from one point contact until -0.2mm. The artificial volume inserted may add more solid volume in the reactor but the fact that less solid volume is simulated due to the overlap should be taken into account. For the particles that had larger overlap, the technique of subtracting the one volume from the other and then subtracting from the reactor was used for the scenario A. For the scenario B that no volumes are retained the technique was not necessary and the two particles were directly subtracted from the reactor's volume.

The results obtained with the model sphere technique present good quality between the particle to particle and particle to wall mesh but only for the internal side of one reactor's part. As no mesh can be created with Gambit for the whole reactor at once, the geometry has to be divided in parts. In addition, this division is necessary for the partial simulations of the reactor in FLUENT. If the number of cells in the reactor is exceeding the computational power and time required then parts of the reactor can be simulated and the results from one simulation can be used as input for the next one in sequence. The division in model spheres mesh was achieved with a surface that split the reactor in two parts. At every position where the reactor was split, the geometry resulted in two volumes that had the same surface. As the model spheres were inserted in 3D at the contact points of the basic particles, the surface which splits the volume, split also the model and basic spheres. This fact resulted in small edges and complex local geometry at the division position and many highly skewed elements occurred in this area. As the mesh cannot be created for the whole volume and for the splitting technique of a plane surface, the model sphere mesh technique fails. The results from the geometry and the mesh generation in Gambit for the meshing technique of spheres with reduced radius are presented in Figure 38. In the first picture the model spheres for simulating the particle's contact are shown, while in the third the additional volume inserted can be observed.

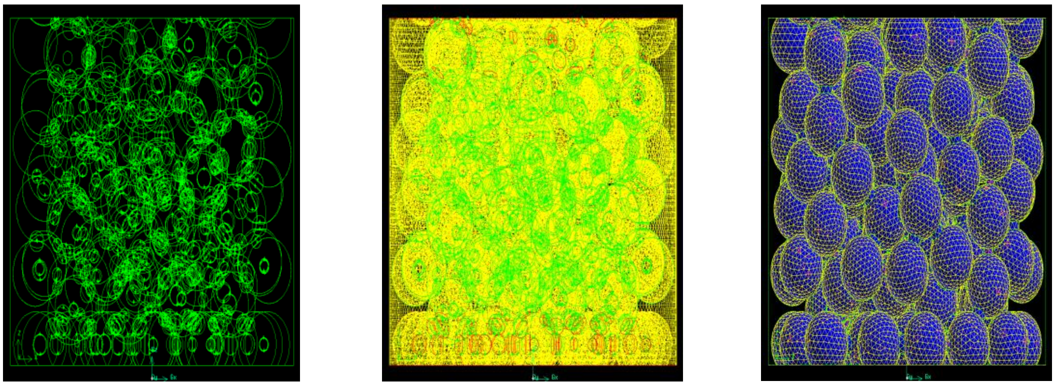


Figure 39 : Geometry and mesh generated for the model spheres technique

4.1.4 Mesh4: The less spheres script

For the less spheres script, the maximum overlap reached the 0.033mm. The technique of overlapping spheres was not used as this small overlap would create small edges that could not be meshed as concluded with the first script of the full simulation. The “near miss” can now be applied as the radius of the spheres can be reduced to 0.5% or 1% as applied in the studies of Nijemeisland and Dixon. The “near miss” model was used for spheres or cylinders with specified position with a maximum number of 44 particles with periodic segments and with a tube to particle ratio equal to 2 or 4.In this work we have 775 spheres of which some overlap. The radius of the particle was reduced according to the maximum overlap plus the 1% decrease of the near miss model, r=4.9336 mm. The reactor was divided in four parts at z=2, z=100 and z=200. The mesh created contained 2,894,520 cells among which 68 were had a skewness of over 0.97 and the worst element had a skewness of 0.996728 and was located at the position where the reactor was divided. The model sphere technique was also used and high skewed elements occurred again on the position were the division took place. As the coarse wall mesh was used in the DEM simulations, no spheres were found to exceed the wall boundaries and no contact with the wall was observed.

4.1.5 Mesh5: The periodic segment

After the simulations performed and the geometries created, it was concluded that the highly skewed elements occur on the position where the division of the reactor takes place. This fact leads to the idea of simulating only one part of the reactor that will contain similar surfaces on the top and bottom in order to perform simulations with periodic boundaries.

A new script was created in ESyS-Particle named the periodic segment where the same mesh and interactions were used. Two layers of spheres were inserted on the top and bottom of the reactor. These layers contained the exact same number of spheres and their positions with the only difference relying on the height. The first layer was created at z=0mm and the centres of the spheres were at z=5mm and the second at z=50mm and the centres of spheres at z=55mm. For the 50mm segment of the reactor 162 particles was calculated that should be inserted according to the equation 17 for the porosity $\varepsilon = 0{,}4$. As 46 spheres represented the two layers 116 spheres were let to fall from z=45mm. In order to achieve this porosity and the particles to fit in the segment, the drag forces were included and the particles were inserted every 5,000 time steps. The overlap was inevitable and had a maximum value of 0.215mm. In the C code the radius of the spheres was reduced to 4.9mm and the model sphere technique was implemented. The geometry was created without any virtual faces or edges. The segment was split with to plane surfaces at z=3.5mm and at z=53.5mm. The final void volume of the reactor was 72562.832906 mm^3 and the final porosity was increased to 0.48. The cause of this fact lays on the additional inserted volumes of model spheres, although the volume of the basic particles was decreased. As less overlap is obtained from the DEM simulations, the less volume is inserted with model spheres and no need for the reduction of the particles' radius occurs. T-grid mesh with 1 size was used to mesh the geometry. For the Scenario A where the volumes of the particles are meshed, the mesh contains 952,852 cells with the worst one to have skewness of 0.938108 and only this element was found to have skewness over 0.93. For the Scenario B, the mesh created had 468,646 elements with the same quality and 0.938108 skewness of the worst element. Figure 39 shows the geometry and mesh generation of the periodic segment with the model sphere technique.

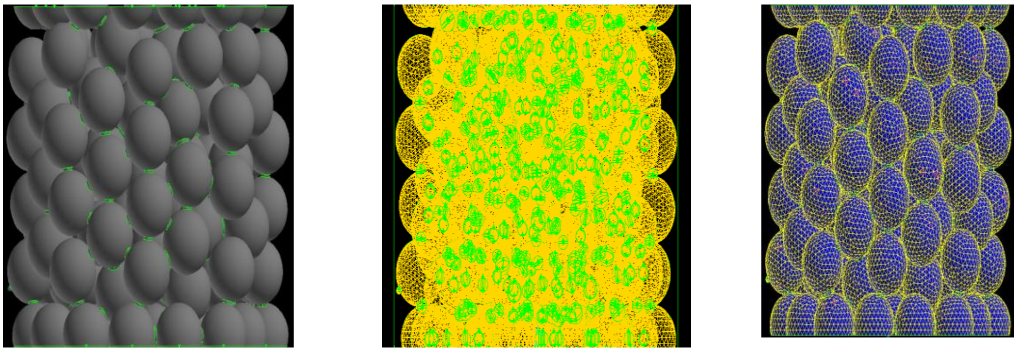


Figure 40 : Meshed geometry generated for the 50mm segment with no highly skewed elements

4.2 Geometry generation with catalytic cylinders

For the DEM simulations for cylinders, two spheres were used to represent one cylinder. With this modeling, the contact points are not predicted acurretely. Figure 40 shows the problems occuring with this modelling. In order to take into account the case of touching spheres as shown in Figure 40, the raduis and the height of the particle was reduced. The new raduis was given by the equation 19 according to the half of the distance of the touching spheres in the cylinders' boundaries. The height was calculated as before by multiplying four times the sphere's raduis.

$$r_{new} = \frac{3-\sqrt{2}}{2} \cdot r \qquad (mm) \qquad \text{Equation 20}$$

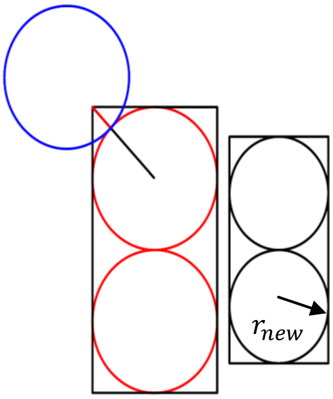


Figure 41: Representing one cylinder with two spheres and modifications of the geometry

By decreasing the raduis of the spheres, many other contact points on the periphery and on the circular surfaces between cylinders are lost, thus more spheres should be used for the representation of one cylinder.

4.2.1 Mesh1: Ovelapping cylinders

The simulations with two spheres representing one cylinder did not gave the requested porosity as expected. Less spheres were used to represent 324 cylindrical particles than in the simulation with spherical particles of the same porosity and raduis. This is also justified by the fact that the bond between the two spheres is not applied on one stable contact point, hence allow the one of the bonded spheres to roll around the other.

The problem of overlapping was also observed in the simulations. The overlap occurred both in bonded spheres as well as in touching spheres. Although the raduis of the simulated spheres was reduced, the overlap between spheres was larger than this reduction. Figure 41 shows the generated geometry and the overlap problems. This generated geometry was not able to be meshed as the cylinders, which overlapped when they were subtracted from the reactor, formed small edges and surfaces that couls not be meshed.

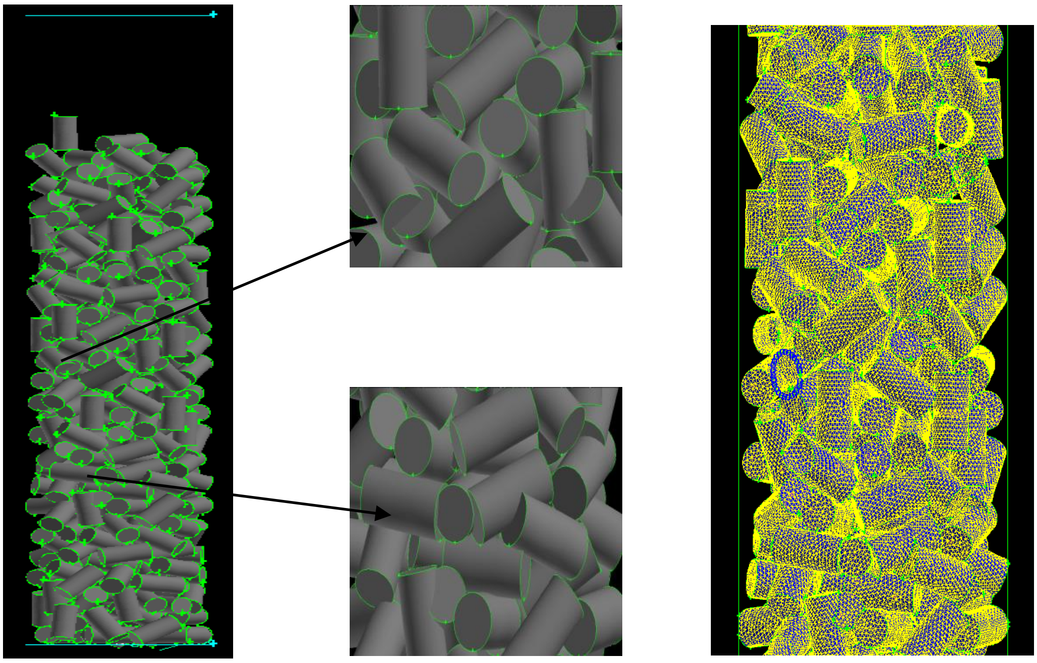


Figure 42: Overlapping cylinders and failure of the mesh generation and of the requested porosity

4.2.2 Mesh2: Periodic segment

After the full simulation of all the particles and the failure of meshing the geometry with overlapping spheres, simulations for a 50mm segment were performed. The requested porosity for the segment was not possible to be obtained even with sequential simulation as performed with spheres. The reason is again the non stable point of applied bonds, so the particles are not acting as solid bonded spheres representing one cylinder but as touching spheres falling in the tube and keeping contact. Even when the radius was reduced more than the equation 19, the overlap between the spheres persisted. In conclusion no mesh generation can be obtained with cylinders that overlap.

4.3 Conclusions and Remarks

The documentation of the present thesis proves that the discrete method can be used for CFD simulations of fixed bed reactors. The geometry of the reactor can automatically be designed in the CFD pre-processor software, Gambit. The positions of the particles can be predicted by the DEM method and the results can be transformed to the journal file of Gambit. Once the journal file is generated, the geometry is designed automatically. For the mesh generation, two techniques have proven to produce a relatively good mesh quality. The “near-miss” model technique firstly applied by Dixon and Nijemeisland and the “sphere model” technique with periodic boundaries can provide a good quality mesh for the CFD simulations with the discrete method.

With the DEM simulations, the motion of the catalytic particles was predicted from the moment they are inserted in the reactor until they reach their final position. The parameters of the time step increment, the elastic stiffness of the wall and the particles interactions as well as the time at which the particles are inserted in the simulation play a major role for the stability of the simulation and the porosity obtained. The problem of overlapping spheres occurred in all the simulations as well as the problem of hanging spheres on the wall. The minimum overlap was obtained with sequential simulations where the simulations were running until the spheres reached an approximate composure and inserted as input data of non dynamic spheres in the next simulation. The porosity of the reactor changed with this method and careful consideration should be taken when used. Additionally, the problem of overlap was not diminished, as the spheres overlapped within a distance of 0.033mm and a decrease of 0.0165mm had to be made on their radius. The problem of hanging spheres on the meshed wall as well as the problem of spheres exceeding the wall boundaries was more intense as the mesh of the cylindrical wall surface became denser. The mesh density influenced also the time of the simulation. The wall with medium mesh was found to be a good compromise for the wall contact and computational time and it was chosen for the full simulation of the reactor. In the C code the problem of spheres exceeding the wall boundaries was treated by increasing the radius of the reactor in 30.5mm and inserting model spheres for the wall contact. During the simulations for the periodic segment script, the bug of hanging spheres was found to be more intense and the coarse mesh was finally chosen. With the use of the coarse mesh no spheres were found to exceed the wall boundaries. The bug problem was overcome by exporting the hanging spheres from the obtained data and drop them again in the next simulation. For the DEM simulations in ESyS –Particle we can conclude that three parameters influence the results obtained and the quality of them:

1. The overlap problem which is related to time increment, number of particles simulated and elastic stiffness of wall to particle and particle to particle interactions.
2. The meshed wall inserted in the simulation that influences the computational time and is related also to the bug occurring in ESyS-Particle
3. The number of the particles inserted in every simulation and the time of insertion which influences the packing and the final porosity.

For the geometry and the mesh generation of spherical particles, the technique of the “near miss” model and the “sphere” model were proven to give a mesh free of highly skewed elements. As in all the DEM simulations, the problem of overlapping spheres occurred, the radius of the particles had to be reduced. Although both the two techniques provided a good quality mesh and can be applied for the simulation of contact points between the spheres. The techniques are more reliable as smaller is the overlap between the spheres. The model sphere technique can also be applied for small overlapping distances but it cannot be applied at the positions where the geometry is divided. Table 4 presents the outlook of the mesh obtained for the simulations with spheres.

For the geometry and the mesh generation of cylindrical particles, no mesh was able to be obtained because of the overlap of the cylinders. This was caused because the bonded interactions applied on the two bonded spheres, which represented one cylinder, did not have one stable point. The forces were applied only in the normal direction and bonded spheres could rotate during the movement. This fact influenced also the porosity obtained as the bonded spheres did not act as a solid cylindrical particle. For the same reasons, the representation of three or four spheres failed also. The geometry of cylindrical particles can be meshed only when accurate DEM results are obtained, thus it is crucial that the cylinders are represented from a large number of spheres. The packing of spheres should be carefully chosen. It is preferable that a small range of spheres radii is used, for the homogeneity of the particles’ mass and the applied forces. This fact will increase the number of spheres used as a more accurate representation requires smaller spheres and hence more spheres. A simulation with more spheres will increase the computational time and will also influence the stability of the simulation. In (Bai et al. 2009), an assembly of 1000 spheres was used to represent the geometry of one cylinder in PFC3D DEM software. This number of spheres for only one cylinder is the same as the simulations performed for spherical catalytic particles. The type of rotational sphere could be also used for the bonded spheres simulations with the condition that the additional interactions define stable bonds between the spheres. In the case study for particle to particle contact, which is provided in the Appendix, the mesh generation with the "model sphere" technique was more easily implemented on the cylindrical particles than on the spherical particles and different mesh sizes were used.

Table 4: Outlook of the mesh generation for catalytic spherical particles in a fixed bed reactor

Script	*Number of cells*	*Number of highly skewed elements*	*Skewness of worst Element*	*Worst element at the contact point*	*Interval Size of created mesh*	*Porosity*
Reduced Model B	2,611,413	7	0.999303	Yes	1	0.457
Reduced Model B	463,422	8	0.992768	No	2	0.457
Reduced Model A	5,798,162	7	0.999303	Yes	1	0.457
Reduced Model A	876,453	8	0.992768	No	2	0.457
Less spheres script B	2,894,520	68	0.996728	No	1	0.545
Periodic segment A	952,852	0	0.938108	Yes	1	0.48
Periodic	468,646	0	0.938108	Yes	1	0.48

segment B

In conclusion, the mesh generated with the two techniques has to be compared. Especially for the “model sphere” technique, the influence of the additional volume inserted cannot be predicted unless CFD simulations are performed and compared to the “near-miss” model technique as well as to a mesh that simulates the exact contact points only for laminar flows.   As mentioned in Chapter 2, the flow field and the turbulence around the catalytic particles play a major role for the prediction of heat transfer within the reactor. Thus the effects of the additional spherical surface at the contact points should be studied carefully. In Chapter 2, the turbulence and heat transfer models used in the existing literature for the discrete method were presented. The boundary conditions should be defined according to the application and CFD simulations for a fixed bed reactor with spherical catalytic particles can be performed.

Abbreviations

SNG	Subtitude Natural Gas
LHV	Lower Heating Value
CFD	Computational Fluid Dynamics
CPU	Computing Processing Unit
3D	three dimensions
2D	two dimensions
CSP	Composite Structured Packing
DEM	Discrete Element Method
API	Application Programming Interface
DNS	Direct Numerical Solution
RANS	Reynolds Averaged Navier- Stokes
RSM	Reynolds Stress Model
RNG k-e	Renormalization group k-e
LES	Large Eddy Simulations
SST k-w	Shear Stress Transport k-w
APDF	Axially Dispersed Plug Flow Model
LSM	Lattice Solid Model
ESSCC	Earth Systems Science Computational Center
ACcESS	Australian Computational Earth System Simulator
MPI	Message Passing Interface
OpenMP	Open Multi-Processing
txt	text file
jou	journal file
NRotElastic	Non Rotational Elastic Interactions
NRotFrictionPrms	Non Rotational Frictional Parameters

List of figures

List of tables

Bibliography

FLUENT 6.3 User's Guide 2003, , © Fluent Inc. 2003-01-28.

GAMBIT1.3 User's Guide 2000, , © Fluent Inc. 2000-04-24.

Abe Steffen 2009**,** Earth System Science Computational Center. *Documentation and Presentations.* [Online] University of Quennsland Australia, 9 10 2009. [Cited: 7 1 2010.] https://twiki.esscc.uq.edu.au/bin/view/ESSCC/DocumentationAndPresentations#Documentation_and_Presentations.

Bai, H., Theuerkauf Jörg, A., G. & M., W. 2009, *A Coupled DEM and CFD Simulation of Flow Field and Pressure Drop in Fixed Bed Reactor with Randomly Packed Catalyst Particles*, American Chemical Society, Washington, DC, ETATS-UNIS.

Balhoff, M.T. & Thompson, K.E. 2006, "A macroscopic model for shear-thinning flow in packed beds based on network modeling", *Chemical Engineering Science,* vol. 61, no. 2, pp. 698-719.

Benneker, A.H., Kronberg, A.E. & Westerterp, K.R. 1997, "Longitudinal Mass and Heat Dispersion in Tubular Reactors", *Industrial & Engineering Chemistry Research,* vol. 36, no. 6, pp. 2031-2040.

Bergeles George (ed) 2006, *Numerical Fluid Dynamics*, 4th edn, Symeon Publication, Athens.

Boussinesq, J. 1877, "Théorie de l'Écoulement Tourbillant", *Acad.Sci.Inst.Fr.,* vol. 23, pp. 46-50.

Cundall, P.A. & Strack, O. 1979, "A discrete numerical model for granular assemblies", *Geotechnique,* vol. 29, no. 1, pp. 47-65.

Di Renzo, A. & Di Maio, F.P. 2007, "Homogeneous and bubbling fluidization regimes in DEM–CFD simulations: Hydrodynamic stability of gas and liquid fluidized beds", *Chemical Engineering Science,* vol. 62, no. 1-2, pp. 116-130.

Dixon A. G. & Nijemeisland M. 2001, "CFD as a Design Tool for Fixed-Bed Reactors", *Industrial & Engineering Chemistry Research,* vol. 40, no. 23, pp. 5246-5254.

Dixon, A.G., Ertan Taskin, M., Nijemeisland, M. & Stitt, E.H. 2008, "Wall-to-particle heat transfer in steam reformer tubes: CFD comparison of catalyst particles", *Chemical Engineering Science,* vol. 63, no. 8, pp. 2219-2224.

Dixon, A.G., Nijemeisland, M. & Stitt, E.H. 2005, "CFD Study of Heat Transfer near and at the Wall of a Fixed Bed Reactor Tube: Effect of Wall Conduction", *Industrial & Engineering Chemistry Research,* vol. 44, no. 16, pp. 6342-6353.

Dommeti, S.M.S., Balakotaiah, V. & West, D.H. 1999, "Analytical Criteria for Validity of Pseudohomogeneous Models of Packed-Bed Catalytic Reactors", *Industrial & Engineering Chemistry Research,* vol. 38, no. 3, pp. 767-777.

Donzé, F.V., Richefeu, V. & Magnier, S.A. 2009, "Advances in discrete element method applied to soil, rock and concrete mechanics", *State of the art of geotechnical engineering, Electronic Journal of Geotechnical Engineering,* vol. 8, pp. 1-44.

Eaton, A.M., Smoot, L.D., Hill, S.C. & Eatough, C.N. 1999, "Components, formulations, solutions, evaluation, and application of comprehensive combustion models", *Progress in Energy and Combustion Science,* vol. 25, no. 4, pp. 387-436.

Eisenlohr, K., Moeller, F. & Dry, M. 1974, "Influence of certain reaction parameters on methanation of coal gas to SNG", *Am.Chem.Soc., Div.Fuel Chem., Prepr,* [Online], vol. 19, no. 3. Available from: http://www.anl.gov/PCS/acsfuel/preprint%20archive/Files/Volumes/Vol19-3.pdf.

Epydoc 3.0.1 2009, Module Overview. *ESyS Software Suite: Documentation. [Online] 3 9 2009.* [Cited: 7 1 2010.] http://esys.esscc.uq.edu.au/esys-particle_python_doc/current/pythonapi/html/esys.lsm.doc.Overview-module.html.

Feng, Z. & Michaelides, E.E. 2004, "The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems", *Journal of Computational Physics,* vol. 195, no. 2, pp. 602-628.

Guardo, A., Coussirat, M., Larrayoz, M.A., Recasens, F. & Egusquiza, E. 2005, "Influence of the turbulence model in CFD modeling of wall-to-fluid heat transfer in packed beds", *Chemical Engineering Science,* vol. 60, no. 6, pp. 1733-1742.

Guardo, A., Coussirat, M., Recasens, F., Larrayoz, M.A. & Escaler, X. 2007, "CFD studies on particle-to-fluid mass and heat transfer in packed beds: Free convection effects in supercritical fluids", *Chemical Engineering Science,* vol. 62, no. 18-20, pp. 5503-5511.

Guardo, A., Coussirat, M., Recasens, F., Larrayoz, M.A. & Escaler, X. 2006, "CFD study on particle-to-fluid heat transfer in fixed bed reactors: Convective heat transfer at low and high pressure", *Chemical Engineering Science,* vol. 61, no. 13, pp. 4341-4353.

Hoomans, B.P.B., Kuipers, J.A.M., Briels, W.J. & van Swaaij, W.P.M. 1996, "Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidised bed: A hard-sphere approach", *Chemical Engineering Science,* vol. 51, no. 1, pp. 99-118.

Iordanidis, A.A. 2002, *Mathematical modeling of catalytic fixed bed reactors*, Twente University Press.

Iordanidis, A.A., van Sint Annaland, M., Kronberg, A.E. & Kuipers, J.A.M. 2004, "A numerical method for the solution of the wave model and convection dominated diffusion type models for catalytic packed bed reactors", *Computers & Chemical Engineering,* vol. 28, no. 11, pp. 2337-2349.

Iordanidis, A.A., van Sint Annaland, M., Kronberg, A.E. & Kuipers, J.A.M. 2003, "A critical comparison between the wave model and the standard dispersion model", *Chemical Engineering Science,* vol. 58, no. 13, pp. 2785-2795.

Jafari, A., Zamankhan, P., Mousavi, S.M. & Pietarinen, K. 2008, "Modeling and CFD simulation of flow behavior and dispersivity through randomly packed bed reactors", *Chemical Engineering Journal,* vol. 144, no. 3, pp. 476-482.

Jakobsen, H.A., Lindborg, H. & Handeland, V. 2002, "A numerical study of the interactions between viscous flow, transport and kinetics in fixed bed reactors", *Computers & Chemical Engineering,* vol. 26, no. 3, pp. 333-357.

Jiang, Y., Khadilkar, M.R., Al-Dahhan, M.H. & Dudukovic, M.P. 2002a, "CFD of multiphase flow in packed-bed reactors: I. *k*-Fluid modeling issues", *AICHE Journal,* vol. 48, no. 4, pp. 701-715.

Jiang, Y., Khadilkar, M.R., Al-Dahhan, M.H. & Dudukovic, M.P. 2002b, "CFD of multiphase flow in packed-bed reactors: II. Results and applications", *AICHE Journal,* vol. 48, no. 4, pp. 716-730.

Jiang, Y., Khadilkar, M.R., Al-Dahhan, M.H. & Dudukovic, M.P. 2001, "CFD modeling of multiphase flow distribution in catalytic packed bed reactors: scale down issues", *Catalysis Today,* vol. 66, no. 2-4, pp. 209-218.

Jiang, Y., Guo, J. & Al-Dahhan, M. 2005, "Multiphase Flow Packed-Bed Reactor Modeling: Combining CFD and Cell Network Model", *Industrial & Engineering Chemistry Research,* vol. 44, no. 14, pp. 4940-4948.

Joshi, J.B. & Ranade, V.V. 2003, "Computational Fluid Dynamics for Designing Process Equipment: Expectations, Current Status, and Path Forward", *Industrial & Engineering Chemistry Research,* vol. 42, no. 6, pp. 1115-1128.

Kuang, S.B., Yu, A.B. & Zou, Z.S. 2009, "Computational Study of Flow Regimes in Vertical Pneumatic Conveying", *Industrial & Engineering Chemistry Research,* vol. 48, no. 14, pp. 6846-6858.

Kuang, S.B., Yu, A.B. & Zou, Z.S. 2008, "A new point-locating algorithm under three-dimensional hybrid meshes", *International Journal of Multiphase Flow,* vol. 34, no. 11, pp. 1023-1030.

Ladyzhenskaya, O.A. 1975, "Mathematical Analysis of Navier-Stokes Equations for Incompressible Liquids", *Annual Review of Fluid Mechanics,* vol. 7, no. 1, pp. 249-272.

Launder, B.E.(. 1972, *Lectures in mathematical models of turbulence [by] B. E. Launder and D. B. Spalding,* Academic Press, London, New York,.

Lee, J., Park, G., Kim, K. & Lee, W. 2007, "Numerical treatment of pebble contact in the flow and heat transfer analysis of a pebble bed reactor core", *Nuclear Engineering and Design,* vol. 237, no. 22, pp. 2183-2196.

Liu G. B., Yu K. T., Yuan X. G. & Liu C. J. 2006, "New model for turbulent mass transfer and its application to the simulations of a pilot-scale randomly packed column for CO_2-NaOH chemical absorption", *Industrial & engineering chemistry research,* vol. 45, no. 9, pp. 3220--3229.

Liu, G.B., Yu, K.T., Yuan, X.G. & Liu, C.J. 2008, "A Computational Transport Model for Wall-Cooled Catalytic Reactor", *Industrial & Engineering Chemistry Research,* vol. 47, no. 8, pp. 2656-2665.

Logtenberg, S.A., Nijemeisland, M. & Dixon, A.G. 1999, "Computational fluid dynamics simulations of fluid flow and heat transfer at the wall–particle contact points in a fixed-bed reactor", *Chemical Engineering Science,* vol. 54, no. 13-14, pp. 2433-2439.

Logtenberg, S.A. & Dixon, A.G. 1998, "Computational fluid dynamics studies of fixed bed heat transfer", *Chemical Engineering and Processing,* vol. 37, no. 1, pp. 7-21.

Luding S., Clement E., Rajchenbach J. & Duran J. 1996, "Simulations of Pattern Formation in Vibrated Granular Media", [Online], . Available from: http://iopscience.iop.org/0295-5075/36/4/247.

Malone, K.F. & Xu, B.H. 2008, "Determination of contact parameters for discrete element method simulations of granular systems", *Particuology,* vol. 6, no. 6, pp. 521-528.

Mantle, M.D., Sederman, A.J. & Gladden, L.F. 2001, "Single- and two-phase flow in fixed-bed reactors: MRI flow visualisation and lattice-Boltzmann simulations", *Chemical Engineering Science,* vol. 56, no. 2, pp. 523-529.

Martens Stefan 2007, "Blockvorlesung Strömungsmechanik und Stoffaustausch-Vernetzung in Gambit", .

Mora, P. & Place, D. 1994, "Simulation of the frictional stick-slip instability", *Pure and Applied Geophysics,* vol. 143, no. 1, pp. 61-87.

Moreau, J.J. 1999, "Numerical aspects of the sweeping process", *Computer Methods in Applied Mechanics and Engineering,* vol. 177, no. 3-4, pp. 329-349.

Motlagh, A.H.A. & Hashemabadi, S.H. 2008, "3D CFD simulation and experimental validation of particle-to-fluid heat transfer in a randomly packed bed of cylindrical particles", *International Communications in Heat and Mass Transfer,* vol. 35, no. 9, pp. 1183-1189.

Nijemeisland, M. & Dixon, A.G. 2004, "CFD study of fluid flow and wall heat transfer in a fixed bed of spheres", *AICHE Journal,* vol. 50, no. 5, pp. 906-921.

Nijemeisland, M. & Dixon, A.G. 2001, "Comparison of CFD simulations to experiment for convective heat transfer in a gas–solid fixed bed", *Chemical Engineering Journal,* vol. 82, no. 1-3, pp. 231-246.

Nijemeisland, M., Dixon, A.G. & Hugh Stitt, E. 2004, "Catalyst design by CFD for heat transfer and reaction in steam reforming", *Chemical Engineering Science,* vol. 59, no. 22-23, pp. 5185-5191.

Peinado Martín Diego 2009, Launchpad ESyS-Particle: HPC Discrete Element Modelling Software. *Sharing experiences with numerical stability q84420.* [Online] 1 10 2009. [Cited: 7 1 2010.] https://answers.launchpad.net/esys-particle/+question/84420.

Phavanee N., Karn P.-S., Sabaithip T, Rungrote K. & Prayut J. 2009, "Simulation of the Flow in a Packed-Bed with and without a Static Mixer by Using CFD Technique", vol. 53, pp. 875-880.

Place, D. & Mora, P. 1999, "The Lattice Solid Model to Simulate the Physics of Rocks and Earthquakes: Incorporation of Friction", *Journal of Computational Physics,* vol. 150, no. 2, pp. 332-372.

Pöschel T. & Schwager T. 2005, *Computational Granular Dynamics,* Springer Verlag.

Potyondy, D.O. & Cundall, P.A. 2004, "A bonded-particle model for rock", *International Journal of Rock Mechanics and Mining Sciences,* vol. 41, no. 8, pp. 1329-1364.

Rodi Wolfgang 1993, *Turbulence Models and Their Application in Hydraulics.* 3rd edn, A.A. Balkema, Rotterdam, Netherlands.

Romkes S.J.P, Dautzenberg F.M., C.M. van den Bleek & Calis H.P.A 2003, "CFD modelling and experimental validation of particle-to-fluid mass and heat transfer in a packed bed at very low channel to particle diameter ratio", vol. 96, pp. 3--13.

Takashi Takeuchi, Masahiko Aihara, Hitoshi Habuka "CFD-simulation of membrane reactor for methane steam reforming", .

Tsaggaris Socrates 2005, *Fluid Dynamics,* Symeon Publications, Athens.

Tsotsas E. & Schlünder E.U. 1990, "Heat transfer in fluid flow:Remarks on the meaning and the calculation of a heat transfer coefficient at the wall", vol. 45, no. 4, pp. 819-837.

Weatherley Dion 2008a, Launchpad ESyS-Particle. *ESyS-Particle: HPC Discrete Element Modelling Software.* [Online] 25 5 2008. [Cited: 7 1 2010.] https://launchpad.net/esys-particle.

Weatherley Dion 2008b, Launchpad ESyS-Particle: HPC Discrete Element Modelling Software. *Tutorials and Documentation.* [Online] 24 11 2008. [Cited: 7 1 2010.] https://answers.launchpad.net/esys-particle/+faq/260.

Wearthley Dion 2009a, Earth Systems Science Computational Center. *Release Notes for ESyS-Particle.* [Online] University of Qeensland Australia, 9 9 2009. [Cited: 7 1 2010.] https://twiki.esscc.uq.edu.au/bin/view/ESSCC/ESySReleaseNotes.

Wearthley Dion 2009b. Launchpad ESyS-Particle: HPC Discrete Element Modelling Software. *ESyS-Particle v2.0 Users Guide.* [Online] 7 8 2009. [Cited: 7 1 2010.] http://launchpadlibrarian.net/36688476/ESyS-Particle_Tutorial-v2.0.pdf.

Wearthley Dion 2009c, Launchpad ESyS-Particle: HPC Discrete Element Modelling Software. *How can I import a particle geometry from file? Faq 877.* [Online] 28 12 2009. [Cited: 7 1 2010.] https://answers.launchpad.net/esys-particle/+faq/877.

Wearthley Dion 2009d. Launchpad ESyS-Particle: HPC Discrete Element Modelling Software. *How can I create a triangle mesh file for ESyS-Particle simulations? Faq683.* [Online] 9 9 2009. [Cited: 7 1 2010.] https://answers.launchpad.net/esys-particle/+faq/683.

Wearthley Dion 2009e, Launchpad ESyS-Particle: HPC Discrete Element Modelling Software. *units system definition q76674.* [Online] 11 7 2009. [Cited: 7 1 2010.] https://answers.launchpad.net/esys-particle/+question/76674.

Wei James 1987, *Advantages in chemical engineering,* Copyright © 1987 Academic Press ,Inc, Florida USA

Weiss Max-Michael, Walter Stefan, Berger Ulrich, Lurgi GmbH 2008, "Lurgi's Methanation Technology for Production of SNG from Coal",[Online], . Available from: http://www.gasification.org/Docs/Conferences/2008/12BERGER.pdf

Wikipedia-DEM 2010. Wikipedia. *DEM Discrete Element Method.* [Online],. Wikipedia, Free Encyclopedia, 25 2 2010. [Cited: 26 2 2010.] http://en.wikipedia.org/wiki/Discrete_element_method

Xu, B.H. & Yu, A.B. 1997, "Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics", *Chemical Engineering Science,* vol. 52, no. 16, pp. 2785-2809.

Yurong He, Thang Ngoc Cong & Yulong Ding 2006, "Gas-solid Two-phase Mixtures Flowing Upward through a Confined Packed Bed", vol. 23, pp. 279-288.

Zwart R.W.R., Boerrigter H., Deurwaarder,E.P., van der Meijden C.M. &. van Paasen S.V.B 2006, "Production of Synthetic Natural Gas (SNG) from Biomass: Development and operation of an integrated bio-SNG system", [Online], . Available from: http://www.ecn.nl/docs/library/report/2006/e06018.pdf.

Appendix

Appendix I : Simulations with spheres

1. Full simulation :Spheres.py

```
#full simulation with 972 particles
#
#
#import the appropriate ESyS-Particle modules:
from esys.lsm import *
from esys.lsm import LsmMpi
from esys.lsm.util import *
from esys.lsm.util import Vec3,BoundingBox
from esys.lsm.geometry import *
from POVsnaps import POVsnaps
import random

#instantiate a simulation object
#and initialise the neighbour search algorithm:
sim = LsmMpi(numWorkerProcesses=1, mpiDimList=[1,1,1])
sim.initNeighbourSearch(
   particleType="NRotSphere",
   gridSpacing=2.5*5,
   verletDist=0.2*5
 )

sim.setSpatialDomain(
  BoundingBox(Vec3(-30,-30,0), Vec3(30,30,310))
 )

#read the output from previous simulation
#sim.readGeometry("previous_output.geo")

#set particles non dynamic
#sim.setParticleNonDynamic(tag=0)

particle0=NRotSphere(id=0, posn=Vec3(10,10,280), radius=5.0, mass=1)
particle0.setLinearVelocity(Vec3(0,0,-1000))
sim.createParticle(particle0)

sim.createInteractionGroup (
                 NRotFrictionPrms (
                   name = "friction",
                   normalK = 10000000.0,
                   dynamicMu=0.5,
                   shearK=100000.0
                        )
 )


#initialise gravity in the domain:
sim.createInteractionGroup(
   GravityPrms(name="earth-gravity", acceleration=Vec3(0,0,-9810))
 )

#separate wall: floor
sim.createWall(
   name="floor",
   posn=Vec3(0,0,0),
   normal=Vec3(0,0,1)
 )

sim.createInteractionGroup(
   NRotElasticWallPrms(
      name = "elasticWall",
      wallName = "floor",
      normalK = 10000000
   )
 )

#separate wall: top
sim.createWall(
   name="top",
   posn=Vec3(0,0,305),
   normal=Vec3(0,0,-1)
)

sim.createInteractionGroup(
   NRotElasticWallPrms(
      name = "elasticWall2",
      wallName = "top",
      normalK = 10000000
   )
)

#read the TriMesh wallfrom the mesh file:
sim.readMesh(
   fileName = "meduim.lsm",
   meshName = " meduim "
   )

sim.createInteractionGroup (
      NRotElasticTriMeshPrms (
         name = "WallInteraction",
         meshName = " meduim ",
         normalK = 10000000
      )
   )
#add local viscosity to simulate air resistance:
#sim.createInteractionGroup(
 #   LinDampingPrms(
 #       name="linDamping",
 #       viscosity=10.0,
  #      maxIterations=100
 #   )
#)

#set the number of timesteps and timestep increment:
sim.setNumTimeSteps(300000)
sim.setTimeStepSize(0.0001)

#add a POVsnaps Runnable:
povcam = POVsnaps(sim=sim, interval=1000)
povcam.configure(lookAt=Vec3(0,2000,80), camPosn=Vec3(0,80000,-110))
sim.addPostTimeStepRunnable(povcam)

#add a CheckPointer to save simulation data at regular intervals:
sim.createCheckPointer (
   CheckPointPrms (
     fileNamePrefix = "spheres",
     beginTimeStep = 0,
     endTimeStep = 300000,
     timeStepIncr =1000
  )
)
i=1
N=sim.getNumTimeSteps()
for n in range (N):
 sim.runTimeStep()
 h = sim.getTimeStep()
 if (i<=971):
  if (h%100==0):
        x1 = random.uniform(-20,20)
        y1 = random.uniform(-20,20)
        z1 = random.uniform(280,290)
        particle=NRotSphere(id=i,  posn=Vec3(x1,y1,z1),  radius=5.0,
mass=1)
        particle.setLinearVelocity(Vec3(0,0,-1000))
        sim.createParticle(particle)
        i=i+1

sim.exit()
```

2. Less spheres script :Spheres.py

The same script was run for the less spheres model. The simulations were run sequentially. In every simulation, one hundred particles were inserted and the output data from the one simulation were the input data for the next one where the particles were set to be non dynamic. The same script as spheres.py was run for 100,000 time steps but the coarse mesh was used. After the insertion of 600 particles, 50 particles were inserted in the simulation and the velocity was set to -200 m/s.

This method predicted the packing of only 775 spheres. The porosity of the reactor increased but the overlap between the spheres was decreased. The less number of simulated spheres, allowed a better prediction of the spheres' position. The hanging spheres were detected in the output files and removed from the input data for the next simulation.

Appendix II : Simulations with bonded spheres

1. The Python code for .geo files

```
#!/usr/bin/python

import sys
import random
import math
#import pdb

#pdb.set_trace()

start = "LSMGeometry 1.2\n\
BoundingBox -30 -30 0 30 30 305\n\
PeriodicBoundaries 0 0 0\n\
Dimension 3D\n\
BeginParticles\n\
Simple\n\
2\n"
ss = str(start)
middle = "EndParticles\n\
BeginConnect\n\
1\n"
sm = str(middle)
end = "EndConnect"
se = str(end)

for f in range (1,325,1):
  ID1 = 2*f
  ID2 = ID1+1
  x1 = random.uniform(-20,20)
  y1 = random.uniform(-20,20)
  z1 = random.uniform(270,290)
  d = math.pi/2
  theta = random.uniform(0.01,d)
  x2 = 10*math.sin(theta)+x1
  y2 = y1
  z2 = 10*math.sin(theta)+z1
  value1 =str(x1)+' '+str(y1)+' '+str(z1)+' '+str(5)+' '+str(ID1)+' 
'+str(f)
  value2 =str(x2)+' '+str(y2)+' '+str(z2)+' '+str(5)+' '+str(ID2)+' 
'+str(f)
  value3 =str(ID1)+' '+str(ID2)+' '+str(1)
  k = open("two_spheres_"+str(f)+".geo","w")
  k.write(ss)
  k.write(value1+'\n')
  k.write(value2+'\n')
  k.write(sm)
  k.write(value3+'\n')
  k.write(se)
  k.close
```

2. The bonded spheres script

```
#simulation with 324 cylinders
#
#
#import the appropriate ESyS-Particle modules:
from esys.lsm import *
from esys.lsm import LsmMpi
from esys.lsm.util import *
from esys.lsm.util import Vec3,BoundingBox
from esys.lsm.geometry import *
from POVsnaps import POVsnaps
import random


#instantiate a simulation object
#and initialise the neighbour search algorithm:
sim = LsmMpi(numWorkerProcesses=1, mpiDimList=[1,1,1])
sim.initNeighbourSearch(
   particleType="NRotSphere",
   gridSpacing=2.5*5,
   verletDist=0.2*5
)

#create a cylindrical particle of spheres
sim.readGeometry("two_spheres.geo")

sim.setParticleDensity(
  tag=0,
  mask=-1,
  Density = 0.001909859317
 )
sim.setTaggedParticleVelocity(
   tag=0,
   Velocity = Vec3(0.0,0.0,-1000.0)
 )
sim.createInteractionGroup(
                 NRotBondPrms(
                   name = "sphereBonds",
                   normalK = 10000000.0,
                   breakDistance = 500.0,
                   tag=1
                  )
           )


sim.createInteractionGroup (
                 NRotFrictionPrms (
                   name = "friction",
                   normalK = 10000000.0,
                   dynamicMu=0.5,
                   shearK=100000.0
                        )
              )
sim.createExclusion (
                     interactionName1 = "sphereBonds",
                     interactionName2 = "friction"
                      )

#initialise gravity in the domain:
```

```
sim.createInteractionGroup(
   GravityPrms(name="earth-gravity", acceleration=Vec3(0,0,-9810))
 )

#separate wall: floor
sim.createWall(
   name="floor",
   posn=Vec3(0,0,0),
   normal=Vec3(0,0,1)
 )

sim.createInteractionGroup(
   NRotElasticWallPrms(
      name = "elasticWall",
      wallName = "floor",
      normalK = 10000000
   )
 )
#separate wall: top
sim.createWall(
   name="top",
   posn=Vec3(0,0,305),
   normal=Vec3(0,0,-1)
 )

sim.createInteractionGroup(
   NRotElasticWallPrms(
      name = "elasticWall2",
      wallName = "top",
      normalK = 10000000
   )
 )

#read the TriMesh wallfrom the mesh file:
sim.readMesh(
   fileName = "coarse.lsm",
   meshName = "coarse"
   )

sim.createInteractionGroup (
      NRotElasticTriMeshPrms (
         name = "WallInteraction",
         meshName = "coarse",
         normalK = 10000000
      )
   )
#add local viscosity to simulate air resistance:
sim.createInteractionGroup(
    LinDampingPrms(
        name="linDamping",
        viscosity=8.0,
        maxIterations=100
    )
)

#set the number of timesteps and timestep increment:
sim.setNumTimeSteps(4000000)
sim.setTimeStepSize(0.00001)

#add a POVsnaps Runnable:
povcam = POVsnaps(sim=sim, interval=1000)
```

```
povcam.configure(lookAt=Vec3(0,2000,80), camPosn=Vec3(0,80000,-110))
sim.addPostTimeStepRunnable(povcam)

#add a CheckPointer to save simulation data at regular intervals:
sim.createCheckPointer (
   CheckPointPrms (
     fileNamePrefix = "ela",
     beginTimeStep = 0,
     endTimeStep = 4000000,
     timeStepIncr =1000
  )
)
i=1
N=sim.getNumTimeSteps()
for n in range (N):
 sim.runTimeStep()
 h = sim.getTimeStep()
 if (i<=323):
  if (h%10000==0):
        sim.readGeometry("two_spheres_"+str(i)+".geo")

        sim.setParticleDensity(
              tag= i,
              mask=-1,
              Density = 0.001909859317
                )
        sim.createInteractionGroup(
                 NRotBondPrms(
                   name = "sphere"+str(i)+"Bonds",
                   normalK = 10000000.0,
                   breakDistance = 500.0,
                   tag=1
                  )
           )
        sim.setTaggedParticleVelocity(
               tag= i,
               Velocity = Vec3(0.0,0.0,-1000.0)
              )

        sim.createExclusion (
                     interactionName1 = "sphere"+str(i)+"Bonds",
                     interactionName2 = "friction"
              )

        i=i+1



sim.exit()
```

Appendix III : C Codes for simulations with spheres

1. **Mesh 1:Overlapping spheres**

Scenario A

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
char *date;
double Reactor_r=30;                    //the radius of the reactor in the xy plane in mm
int Reactor_h=300;                  //the height of the reactor in the z direction in mm
char *inname="spheres.txt";    //name of ESyS output file
char *outname="output.txt";          //name of output sorted data
char *jouname="Mesh1OverlappingA.jou";         //name of the jou file

//special attention should be given to the dimension of the array if the spheres exceed
 //the number of 5000

int i,j,k,endi,vol,vol_wall,vol_p,volumes,wx;
int spheres;
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0,Ig;
double id2=0,IIg;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char colon=(char)58;
char ssl=(char)92;
char *arc_angles;
char *arc_plane;
char *arc_angles_s;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy;
char *move;
double r,rh;
double dist,metro,wall,wall_x,wall_y;


int main()
```

```
{
//create txt and open it
  FILE *input;
  FILE *output;

  input = fopen(inname,"r+");
  output = fopen(outname,"w");

  if (input == NULL)
  {
      printf("Problem opening file %s for reading.\n", inname);
   }



//read the particles,the first line
  fscanf(input,"%d",&spheres);
                 printf(" i read the number of spheres %d.\n",spheres);

//starting the loop for reading
    for (i=1;i<=spheres;i++)
    {
        for (j=1;j<=22;j++)
        {
             fscanf(input,"%lf",&u[i][j]);
        }
      }
  printf("________\n");
//close the file
  fclose(input);
//the sort loop
  endi=spheres-1;
  printf("the sort loop starts\n");
  for (i=1;i<=endi;i++)
  {


        for (j=i+1;j<=spheres;j++)
        {
          id1=u[i][5];
          id2=u[j][5];
          if (id2<id1)
          {
            val_i1=u[i][1];
            val_i2=u[i][2];
            val_i3=u[i][3];
            val_i4=u[i][4];
          val_j1=u[j][1];
          val_j2=u[j][2];
          val_j3=u[j][3];
          val_j4=u[j][4];



          u[i][1]=val_j1;
          u[i][2]=val_j2;
          u[i][3]=val_j3;
          u[i][4]=val_j4;
          u[i][5]=id2;

          u[j][1]=val_i1;
          u[j][2]=val_i2;
          u[j][3]=val_i3;
          u[j][4]=val_i4;
          u[j][5]=id1;
          }
        }
    }
//print the sort loop
//  for (i=1;i<=spheres;i++)
//  {
//     printf("%f %f %f %lf %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//     }


//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
     fprintf(output,"%f %f %f %f %0.f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }

//start generating the jou

//fist comes the header

//standard format of header

  FILE *jou;
  jou = fopen(jouname,"w");

    header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
    date="File opened for write Fri Feb 26 18:33:17 2010.";
    char *identif="creating";            //not necessary
  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
  fprintf(jou,"%c %s\n",slash,date);
//uni tolerance in Gambit
fprintf(jou,"default load %cC%c%c%cGAMBIT2.ini%c\n",quote,colon,ssl,ssl,quote);

//stadard format of the first point vertex.1
  fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);

//the reactor VOLUME 1
    arc_angles="startangle 0 endangle 360 center";
    arc_plane="xyplane arc";

  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
  fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
  fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);

//the basic particle VOLUME 2
  r=5;
  arc_angles_s="startangle 0 endangle 180 center";



  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",r,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.4%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the contact point VOLUME 3
  rh=1;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.6%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

 copy="volume cmove";
 move="volume move";
 vol=3;
 //create  the basic particles

  for (i=1;i<=spheres;i=i+1)  //the increment of i infuences the number of spheres one //cylinder two
spheres i=i+2, one cylinder 3spheres i=i+3 etc.
    {
       vol=vol+1;
```

```
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];

        fprintf(jou,"%s %cvolume.2%c offset %f %f %f\n",copy,quote,quote,xA,yA,zA);
    }

 fprintf(jou,"%c %s\n",slash,header);
 fprintf(jou,"%c Identifier %cbasic%c\n",slash,quote,quote);
 fprintf(jou,"%c %s\n",slash,date);
 fprintf(jou,"save name %cC%c%c%cMesh1Overlapping.SceA.dbs%c\n",quote,colon,ssl,ssl,quote);

   //wall contact
k=3;
vol_wall=vol;
 for (i=1;i<=spheres;i=i+1)
   {
        k=k+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];

        metro=pow((pow(xA,2)+pow(yA,2)),0.5);
        f=atan(fabs(yA/xA));
         wall=Reactor_r-r-metro;

        if (wall<=0.5)
            {
                vol_wall=vol_wall+1;
                printf("wol_wall %i in the if state\n",vol_wall);

                if (xA>0&&yA>0)
                {
                 wall_x=xA+(0.5*wall+r)*cos(f);
                 wall_y=yA+(0.5*wall+r)*sin(f);
                   }

                if (xA<0&&yA>0)
                {
                 wall_x=xA-(0.5*wall+r)*cos(f);
                 wall_y=yA+(0.5*wall+r)*sin(f);
                   }

                if (xA<0&&yA<0)
                {
                 wall_x=xA-(0.5*wall+r)*cos(f);
                 wall_y=yA-(0.5*wall+r)*sin(f);
                  }
                if (xA>0&&yA<0)
                {
                 wall_x=xA+(0.5*wall+r)*cos(f);
                 wall_y=yA-(0.5*wall+r)*sin(f);
                }
                if (xA==0)
                 {
                   if (yA>0)
                    {
                        wall_x=xA;
                        wall_y=yA+r+wall/2;
                       }
                   if (yA<0)
                     {
                         wall_x=xA;
                         wall_y=yA-r-wall/2;
                      }
                  }
                if (yA==0)
                 {
                   if (xA>0)
                    {
                       wall_x=xA+r+wall/2;
                       wall_y=yA;
                     }
                   if (xA<0)
                    {
                         wall_x=xA-r-wall/2;
                         wall_y=yA;
                    }
                  }

            fprintf(jou,"%s %cvolume.3%c offset %f %f %f\n",copy,quote,quote,wall_x,wall_y,zA);
            fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
keeptool\n",quote,quote,quote,vol_wall,quote);
            fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c
keeptool\n",quote,k,quote,quote,vol_wall,quote);
             }
    }
//subtract the basic particles  from the reactor and between them
vol=3;
  for (i=1;i<=spheres;i=i+1)
```

```
   {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        id1=u[i][5];

        for  (j=i+1;j<=spheres;j=j+1)
          {
            k=3+j;
            Dx=xA-u[j][1];
            Dy=yA-u[j][2];
            Dz=zA-u[j][3];
            id2=u[j][5];

            dist=pow((pow(Dx,2)+pow(Dy,2)+pow(Dz,2)),0.5);

           if (dist<=10.0)
            {
              fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool\n",quote,vol,quote,quote,k,quote);
            }
          }
          fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c keeptool\n",quote,quote,quote,vol,quote);
   }

  fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
  fprintf(jou,"volume delete %cvolume.3%c lowertopology\n",quote,quote);
  fprintf(jou,"save\n");
   return 0;
}

/*---------------------------------------------------------------------------------------------------------*/
```

Scenario B

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
char *date;
double Reactor_r=31;                    //the radius of the reactor in the xy plane in mm
int Reactor_h=300;                    //the height of the reactor in the z direction in mm
char *inname="spheres.txt";    //name of ESyS output file
char *outname="output.txt";          //name of output sorted data
char *jouname="Mesh1Overlapping.jou";         //name of the jou file

//special attention should be given to the dimension of the array if the spheres exceed the number of 5000
int i,j,k,endi,vol,vol_wall,vol_p,volumes,wx;
int spheres;
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0,Ig;
double id2=0,IIg;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char colon=(char)58;
char ssl=(char)92;
char *arc_angles;
char *arc_plane;
char *arc_angles_s;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy;
char *move;
double r,rh;
double dist,metro,wall,wall_x,wall_y;

int main()
{
//create txt and open it
  FILE *input;
  FILE *output;

  input = fopen(inname,"r+");
```

```
 output = fopen(outname,"w");
 if (input == NULL)
 {
   printf("Problem opening file %s for reading.\n", inname);
  }
//read the particles,the first line
 fscanf(input,"%d",&spheres);
 printf("at last i read it,number of spheres %d.\n",spheres);

//starting the loop for reading
  for (i=1;i<=spheres;i++)
  {
    for (j=1;j<=22;j++)
    {
       fscanf(input,"%lf",&u[i][j]);
    }
   }
 printf("________\n");
//close the file
 fclose(input);
//the sort loop
 endi=spheres-1;
 printf("the sort loop starts\n");
 for (i=1;i<=endi;i++)
 {
    for (j=i+1;j<=spheres;j++)
    {
     id1=u[i][5];
     id2=u[j][5];
     if (id2<id1)
     {
      val_i1=u[i][1];
      val_i2=u[i][2];
      val_i3=u[i][3];
      val_i4=u[i][4];

      val_j1=u[j][1];
      val_j2=u[j][2];
      val_j3=u[j][3];
      val_j4=u[j][4];

      u[i][1]=val_j1;
      u[i][2]=val_j2;
      u[i][3]=val_j3;
      u[i][4]=val_j4;
```

```
            u[i][5]=id2;

            u[j][1]=val_i1;
            u[j][2]=val_i2;
            u[j][3]=val_i3;
            u[j][4]=val_i4;
            u[j][5]=id1;
            }
        }
    }
//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
     fprintf(output,"%f %f %f %f %0.f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }
//start generating the jou
//fist comes the header

//standard format of header

  FILE *jou;
  jou = fopen(jouname,"w");
    header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
    date="File opened for write Fri Feb 26 18:33:17 2010.";
    char *identif="creating";
  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
  fprintf(jou,"%c %s\n",slash,date);
//uni tolerance in Gambit
fprintf(jou,"default load %cC%c%c%cGAMBIT2.ini%c\n",quote,colon,ssl,ssl,quote);

//stadard format of the first point vertex.1
  fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);

//the reactor VOLUME 1
    arc_angles="startangle 0 endangle 360 center";
    arc_plane="xyplane arc";
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
  fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
  fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);

//the basic particle VOLUME 2
  r=5;
  arc_angles_s="startangle 0 endangle 180 center";
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",r,arc_angles_s,quote,quote,arc_plane);
```

```
 fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
 fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
 fprintf(jou,"volume create revolve %cface.4%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the contact point VOLUME 3
 rh=1;
 fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
 fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
 fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
 fprintf(jou,"volume create revolve %cface.6%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

 copy="volume cmove";
 move="volume move";
 vol=3;

   //wall contact
 for (i=1;i<=spheres;i=i+1)
   {
        k=k+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        metro=pow((pow(xA,2)+pow(yA,2)),0.5);
        f=atan(fabs(yA/xA));
         wall=Reactor_r-r-metro;

        if (wall<=0.5)
            {
               vol_wall=vol_wall+1;
                if (xA>0&&yA>0)
                {
                 wall_x=xA+(0.5*wall+r)*cos(f);
                 wall_y=yA+(0.5*wall+r)*sin(f);
                  }
                if (xA<0&&yA>0)
                {
                 wall_x=xA-(0.5*wall+r)*cos(f);
                 wall_y=yA+(0.5*wall+r)*sin(f);
                   }
                if (xA<0&&yA<0)
                {
                 wall_x=xA-(0.5*wall+r)*cos(f);
                 wall_y=yA-(0.5*wall+r)*sin(f);
                   }
                if (xA>0&&yA<0)
```

```
                 {
                  wall_x=xA+(0.5*wall+r)*cos(f);
                  wall_y=yA-(0.5*wall+r)*sin(f);
                    }
                 if (xA==0)
                  {
                    if (yA>0)
                     {
                        wall_x=xA;
                        wall_y=yA+r+wall/2;
                       }
                    if (yA<0)
                      {
                         wall_x=xA;
                         wall_y=yA-r-wall/2;
                      }
                   }
                 if (yA==0)
                  {
                    if (xA>0)
                     {
                       wall_x=xA+r+wall/2;
                       wall_y=yA;
                      }
                    if (xA<0)
                     {
                         wall_x=xA-r-wall/2;
                         wall_y=yA;
                     }
                   }
        fprintf(jou,"%s %cvolume.3%c offset %f %f %f\n",copy,quote,quote,wall_x,wall_y,zA);
        fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.4%c \n",quote,quote,quote,quote);
             }
    }

    //create and subtract the basic particles
  for (i=1;i<=spheres;i=i+1)
//the increment of i infuences the number of spheres one cylynder two spheres i=i+2, one cylinder 3spheres i=i+3 etc.
    {
         vol=vol+1;
         xA=u[i][1];
         yA=u[i][2];
         zA=u[i][3];
         fprintf(jou,"%s %cvolume.2%c offset %f %f %f\n",copy,quote,quote,xA,yA,zA);
```

```
     fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.4%c \n",quote,quote,quote,quote);
   }

  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %cbasic%c\n",slash,quote,quote);
  fprintf(jou,"%c %s\n",slash,date);
  fprintf(jou,"save name %cC%c%c%cMesh1Overlapping.SceB.dbs%c\n",quote,colon,ssl,ssl,quote);
   //
  fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
  fprintf(jou,"volume delete %cvolume.3%c lowertopology\n",quote,quote);
  fprintf(jou,"save\n");

   return 0;
}

/*-------------------------------------------------------------------------------------------------------*/
```

2. Mesh 2:Raduis reduced model

It s the same as Mesh1 for the scenarios A and B and the only change is made at the line 171 where the value of the basic particle’s radius changes.

3. Mesh 3:Spheres model

Scenario A

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
char *date;
double Reactor_r=30.3;              //the radius of the reactor in the xy plane in mm
int Reactor_h=305;                 //the height of the reactor in the z direction in mm
char *inname="spheres.txt";    //name of ESyS output file
char *outname="output.txt";         //name of output sorted data
char *jouname="Mesh3_A.jou";        //name of the jou file

//special attention should be given to the dimension of the array if the spheres exceed the number of 5000

int i,j,k,endi,vol,vol_wall,vol_p,volumes,wx,face,edge,out;
int spheres;
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0,Ig;
double id2=0,IIg;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char colon=(char)58;
char ssl=(char)92;
char *arc_angles;
char *arc_plane;
char *arc_angles_s;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy;
char *move;
double r,rh,tolr,rd;
double dist,metro,wall,wall_x,wall_y;

int main()
{
//create txt and open it
  FILE *input;
  FILE *output;

  input = fopen(inname,"r+");
  output = fopen(outname,"w");

  if (input == NULL)
  {
      printf("Problem opening file %s for reading.\n", inname);
   }

//read the particles,the first line
  fscanf(input,"%d",&spheres);
  printf("at last i read it,number of spheres %d.\n",spheres);

//starting the loop for reading
    for (i=1;i<=spheres;i++)
    {
      for (j=1;j<=22;j++)
      {
           fscanf(input,"%lf",&u[i][j]);
      }
     }
//printing the array
//  printf("the array is\n");
//  for (i=1;i<=spheres;i++)
//    {
//        printf("%lf %lf %lf %lf %lf\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//    }
  printf("________\n");
//close the file
  fclose(input);
//the sort loop
  endi=spheres-1;
  printf("the sort loop starts\n");
  for (i=1;i<=endi;i++)
  {
      for (j=i+1;j<=spheres;j++)
      {
        id1=u[i][5];
        id2=u[j][5];
        if (id2<id1)
        {
          val_i1=u[i][1];
          val_i2=u[i][2];
          val_i3=u[i][3];
          val_i4=u[i][4];

          val_j1=u[j][1];
          val_j2=u[j][2];
          val_j3=u[j][3];
          val_j4=u[j][4];

          u[i][1]=val_j1;
          u[i][2]=val_j2;
          u[i][3]=val_j3;
          u[i][4]=val_j4;
          u[i][5]=id2;

          u[j][1]=val_i1;
          u[j][2]=val_i2;
          u[j][3]=val_i3;
          u[j][4]=val_i4;
```

```
            u[j][5]=id1;
            }
        }
    }
//print the sort loop
//  for (i=1;i<=spheres;i++)
//  {
//     printf("%f %f %f %lf %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//     }

//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
     fprintf(output,"%f %f %f %f %0.f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }

//start generating the jou
//fist comes the header
//standard format of header

  FILE *jou;
  jou = fopen(jouname,"w");

    header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
    date="File opened for write Fri Feb 26 18:33:17 2010.";
    char *identif="creating";             //not necessary
  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
  fprintf(jou,"%c %s\n",slash,date);
//uni tolerance in Gambit
fprintf(jou,"default load %cC%c%c%cGAMBIT2.ini%c\n",quote,colon,ssl,ssl,quote);

//stadard format of the first point vertex.1
  fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);
  copy="volume cmove";
  move="volume move";

//the reactor VOLUME 1
    arc_angles="startangle 0 endangle 360 center";
    arc_plane="xyplane arc";

  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
  fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
  fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);
```

```
//the basic particle VOLUME 2
  r=4.74;
  arc_angles_s="startangle 0 endangle 180 center";
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",r,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.4%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the wall contact point VOLUME 3
  rh=1.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.6%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the volume for tolerance VOLUME 4
  tolr=0.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",tolr,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.8%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the contact point VOLUME 5
  rd=1.0;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rd,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.10%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

 vol=5;
 //create the basic particles
  for (i=1;i<=spheres;i=i+1)  //the increment of i infuences the number of spheres one cylynder two spheres
i=i+2, one cylinder 3spheres i=i+3 etc.
    {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];

        fprintf(jou,"%s %cvolume.2%c offset %f %f %f\n",copy,quote,quote,xA,yA,zA);

     }

  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %cbasic%c\n",slash,quote,quote);
  fprintf(jou,"%c %s\n",slash,date);
  fprintf(jou,"save name %cC%c%c%cMesh3.SceA.dbs%c\n",quote,colon,ssl,ssl,quote);

//tolerance solve
   zA=u[124][3]-r-0.4;
  fprintf(jou,"%s %cvolume.4%c offset %f %f %f\n",move,quote,quote,u[124][1],u[124][2],zA);
  fprintf(jou,"volume subtract %cvolume.129%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);
    //wall contact
 k=5;
 vol_wall=vol-1;
 face=990-9;
//every vol has one surface but the reactor has 2 more in addition four faces missing 4+6+8+10 cause
//they became vol
 //plus one created from subtract
 edge=6-9; //2 the cylinder and 5 the volumes
 out=0;
  for (i=1;i<=spheres;i=i+1)
    {
         k=k+1;
         xA=u[i][1];
         yA=u[i][2];
         zA=u[i][3];

         metro=pow((pow(xA,2)+pow(yA,2)),0.5);
         f=atan(fabs(yA/xA));
          wall=Reactor_r-r-metro;
         printf("metro %10.6f\n angle f in rad %10.6f\n wall distance %10.6f\n",metro,f,wall);
         if (wall<=0.11&&wall>=0)
             {
                 vol_wall=vol_wall+2;
                 vol_p=vol_wall+1;
                 face=face+9;
                 edge=edge+9;
                 printf("wol_wall %i in the if state\n",vol_wall);

                 if (xA>0&&yA>0)
                 {
                  wall_x=xA+(0.5*wall+r)*cos(f);
                  wall_y=yA+(0.5*wall+r)*sin(f);
                 }

                 if (xA<0&&yA>0)
                 {
                  wall_x=xA-(0.5*wall+r)*cos(f);
```

```
        wall_y=yA+(0.5*wall+r)*sin(f);
       }

       if (xA<0&&yA<0)
       {
        wall_x=xA-(0.5*wall+r)*cos(f);
        wall_y=yA-(0.5*wall+r)*sin(f);
        }

       if (xA>0&&yA<0)
       {
        wall_x=xA+(0.5*wall+r)*cos(f);
        wall_y=yA-(0.5*wall+r)*sin(f);
       }

       if (xA==0)
        {
          if (yA>0)
           {
               wall_x=xA;
               wall_y=yA+r+wall/2;
              }
          if (yA<0)
            {
                wall_x=xA;
                wall_y=yA-r-wall/2;
             }
         }
       if (yA==0)
        {
          if (xA>0)
           {
              wall_x=xA+r+wall/2;
              wall_y=yA;
            }
          if (xA<0)
           {
                wall_x=xA-r-wall/2;
                wall_y=yA;
           }
         }

  printf("xA %10.6f\n yA%10.6f\n",xA,yA);
  printf("wall x %10.6f\n  wall y %10.6f\n",wall_x,wall_y);
 fprintf(jou,"%s %cvolume.3%c offset %f %f %f\n",copy,quote,quote,wall_x,wall_y,zA);
```

```
         fprintf(jou,"volume split %cvolume.%i%c faces %cface.2%c
connected\n",quote,vol_wall,quote,quote,quote);
         fprintf(jou,"volume delete %cvolume.%i%c lowertopology\n",quote,vol_p,quote,quote,quote);
         fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c
keeptool\n",quote,k,quote,quote,vol_wall,quote);
         fprintf(jou,"face create wireframe %cedge.%i%c real\n",quote,edge,quote);
         fprintf(jou,"volume split %cvolume.%i%c faces %cface.%i%c
connected\n",quote,vol_wall,quote,quote,face,quote);
         fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c %cvolume.%i%c
keeptool\n",quote,quote,quote,vol_wall,quote,quote,vol_p,quote);
          }

       if (wall<0)
          {
             out=out+1;
          }

   }

    printf("number of volumes outside boundaries %i\n",out);
    fprintf(jou,"save\n");

pi=4*atan(1);
  volumes=vol_p;
  k=5;
  wx=0;
  for (i=1;i<=spheres;i=i+1)
    {
        k=k+1;
       xA=u[i][1];
       yA=u[i][2];
       zA=u[i][3];

       for  (j=i+1;j<=spheres;j=j+1)
         {
           Dx=xA-u[j][1];
           Dy=yA-u[j][2];
           Dz=zA-u[j][3];

           Ilg=u[j][5]+6 ;////because we have 5 basic volumes before the spheres (with id 0,1,2) is ctreated
and id 0 is volume 1
           dist=pow((pow(Dx,2)+pow(Dy,2)+pow(Dz,2)),0.5);
           xB=xA-0.5*Dx;
           yB=yA-0.5*Dy;
           zB=zA-0.5*Dz;
        if (dist<=10.1)
        {
          if (dist>=10)
           {
            volumes=volumes+1;
            fprintf(jou,"%s %cvolume.5%c multiple 1 offset %f %f %f\n",copy,quote,quote,xB,yB,zB);
            fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool
\n",quote,k,quote,quote,volumes,quote);
            fprintf(jou,"volume subtract %cvolume.%0.f%c volumes %cvolume.%i%c keeptool
\n",quote,IIg,quote,quote,volumes,quote);
            fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c keeptool
\n",quote,quote,quote,volumes,quote);
           }
          if (dist<10&&dist>=9.8)
           {
            volumes=volumes+1;
            fprintf(jou,"%s %cvolume.3%c multiple 1 offset %f %f %f\n",copy,quote,quote,xB,yB,zB);
            fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool
\n",quote,k,quote,quote,volumes,quote);
            fprintf(jou,"volume subtract %cvolume.%0.f%c volumes %cvolume.%i%c keeptool
\n",quote,IIg,quote,quote,volumes,quote);
            fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c keeptool
\n",quote,quote,quote,volumes,quote);
           }
          }
      }
  }

 printf(" final spheres %i\n",volumes);

 fprintf(jou,"save\n");
k=5;
 for (i=1;i<=spheres;i=i+1)
  {
      k=k+1;
     xA=u[i][1];
     yA=u[i][2];
     zA=u[i][3];

     for  (j=i+1;j<=spheres;j=j+1)
      {
        Dx=xA-u[j][1];
        Dy=yA-u[j][2];
        Dz=zA-u[j][3];
```

```
            IIg=u[j][5]+6 ;////because we have 5 basic volumes before the spheres (with id 0,1,2) is ctreated
and id 0 is volume 1

            dist=pow((pow(Dx,2)+pow(Dy,2)+pow(Dz,2)),0.5);

            xB=xA-0.5*Dx;
            yB=yA-0.5*Dy;
            zB=zA-0.5*Dz;

            if (dist<=10.1)
            {
                if (dist<9.8&&dist>5)
                {
                    fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%0.f%c
keeptool\n",quote,k,quote,quote,IIg,quote);
                 }
            }

          }
      fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
keeptool\n",quote,quote,quote,k,quote);
    }
    fprintf(jou,"save\n");
  fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);
  fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
  fprintf(jou,"volume delete %cvolume.3%c lowertopology\n",quote,quote);
  fprintf(jou,"volume delete %cvolume.5%c lowertopology\n",quote,quote);
  fprintf(jou,"save\n");

    return 0;
}

/*---------------------------------------------------------------------------------------------------------*/
```

Scenario B

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
char *date;
double Reactor_r=30.5;                    //the radius of the reactor in the xy plane in mm
int Reactor_h=305;                   //the height of the reactor in the z direction in mm
char *inname="spheres.txt";    //name of ESyS output file
char *outname="output.txt";          //name of output sorted data
char *jouname="B_ModelSpheres.jou";         //name of the jou file

//special attention should be given to the dimension of the array if the spheres exceed
//the number of 5000

int i,j,k,endi,vol,vol_wall,vol_p,volumes,wx,face,edge,out;
int spheres;
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0,Ig;
double id2=0,IIg;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char colon=(char)58;
char ssl=(char)92;
char *arc_angles;
char *arc_plane;
char *arc_angles_s;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy;
char *move;
double r,rh,tolr,rd;
double dist,metro,wall,wall_x,wall_y;


int main()
{
//create txt and open it
```

```
  FILE *input;
  FILE *output;

  input = fopen(inname,"r+");
  output = fopen(outname,"w");

  if (input == NULL)
  {
      printf("Problem opening file %s for reading.\n", inname);
   }

//read the particles,the first line
  fscanf(input,"%d",&spheres);
  printf("at last i read it,number of spheres %d.\n",spheres);

//starting the loop for reading
    for (i=1;i<=spheres;i++)
    {
        for (j=1;j<=22;j++)
        {
             fscanf(input,"%lf",&u[i][j]);
        }
      }
//printing the array
//  printf("the array is\n");
//  for (i=1;i<=spheres;i++)
//    {
//       printf("%lf %lf %lf %lf\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//    }
  printf("________\n");
//close the file
  fclose(input);
//the sort loop
  endi=spheres-1;
  printf("the sort loop starts\n");
  for (i=1;i<=endi;i++)
  {
        for (j=i+1;j<=spheres;j++)
        {
          id1=u[i][5];
          id2=u[j][5];
          if (id2<id1)
          {
            val_i1=u[i][1];
            val_i2=u[i][2];
            val_i3=u[i][3];
            val_i4=u[i][4];

            val_j1=u[j][1];
            val_j2=u[j][2];
            val_j3=u[j][3];
            val_j4=u[j][4];

            u[i][1]=val_j1;
            u[i][2]=val_j2;
            u[i][3]=val_j3;
            u[i][4]=val_j4;
            u[i][5]=id2;

            u[j][1]=val_i1;
            u[j][2]=val_i2;
            u[j][3]=val_i3;
            u[j][4]=val_i4;
            u[j][5]=id1;
            }
        }
    }
//print the sort loop
//  for (i=1;i<=spheres;i++)
//  {
//     printf("%f %f %f %lf %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//     }

//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
    fprintf(output,"%f %f %f %f %0.f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }

//start generating the jou
//fist comes the header
//standard format of header

  FILE *jou;
  jou = fopen(jouname,"w");

    header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
    date="File opened for write Fri Feb 26 18:33:17 2010.";
    char *identif="creating";             //not necessary
  fprintf(jou,"%c %s\n",slash,header);
```

```
  fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
  fprintf(jou,"%c %s\n",slash,date);
//uni tolerance in Gambit
  fprintf(jou,"default load %cC%c%c%cGAMBIT2.ini%c\n",quote,colon,ssl,ssl,quote);

//stadard format of the first point vertex.1
  fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);
  copy="volume cmove";
  move="volume move";

//the reactor VOLUME 1
    arc_angles="startangle 0 endangle 360 center";
    arc_plane="xyplane arc";

  fprintf(jou,"edge create radius %f %s %cvertex.1%c
%s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
  fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
  fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);

//the basic particle VOLUME 2
  r=5;
  arc_angles_s="startangle 0 endangle 180 center";
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",r,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.4%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the wall contact point VOLUME 3
  rh=1.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.6%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the volume for tolerance VOLUME 4
  tolr=0.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",tolr,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.8%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the contact point VOLUME 5
  rd=1.0;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rd,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.10%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

 vol=5;
 //create the basic particles

  for (i=1;i<=spheres;i=i+1)  //the increment of i infuences the number of spheres one cylynder two
spheres i=i+2, one cylinder 3spheres i=i+3 etc.
    {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        fprintf(jou,"%s %cvolume.2%c offset %f %f %f\n",copy,quote,quote,xA,yA,zA);
     }

  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %cbasic%c\n",slash,quote,quote);
  fprintf(jou,"%c %s\n",slash,date);
  fprintf(jou,"save name %cC%c%c%cB-Mesh3B.dbs%c\n",quote,colon,ssl,ssl,quote);

//tolerance solve
   zA=u[124][3]-r-0.4;
  fprintf(jou,"%s %cvolume.4%c offset %f %f %f\n",move,quote,quote,u[124][1],u[124][2],zA);
  fprintf(jou,"volume subtract %cvolume.129%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);
    //wall contact
 k=5;
 vol_wall=vol-1;
 face=990-9; //every vol has one surface but the reactor has 2 more in addition four faces missing
4+6+8+10 cause they became vol
 //plus one created from subtract
 edge=6-9; //2 the cylinder and 5 the volumes
 out=0;
  for (i=1;i<=spheres;i=i+1)
    {
        k=k+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];

        metro=pow((pow(xA,2)+pow(yA,2)),0.5);
        f=atan(fabs(yA/xA));
         wall=Reactor_r-r-metro;
        printf("metro %10.6f\n angle f in rad %10.6f\n wall distance %10.6f\n",metro,f,wall);
```

```
if (wall<=0.11&&wall>=0)
    {
        vol_wall=vol_wall+2;
        vol_p=vol_wall+1;
        face=face+9;
        edge=edge+9;
        printf("wol_wall %i in the if state\n",vol_wall);

        if (xA>0&&yA>0)
        {
         wall_x=xA+(0.5*wall+r)*cos(f);
         wall_y=yA+(0.5*wall+r)*sin(f);
        }

        if (xA<0&&yA>0)
        {
         wall_x=xA-(0.5*wall+r)*cos(f);
         wall_y=yA+(0.5*wall+r)*sin(f);
        }

        if (xA<0&&yA<0)
        {
         wall_x=xA-(0.5*wall+r)*cos(f);
         wall_y=yA-(0.5*wall+r)*sin(f);
         }

        if (xA>0&&yA<0)
        {
         wall_x=xA+(0.5*wall+r)*cos(f);
         wall_y=yA-(0.5*wall+r)*sin(f);
        }

        if (xA==0)
         {
           if (yA>0)
            {
                wall_x=xA;
                wall_y=yA+r+wall/2;
               }
           if (yA<0)
             {
                 wall_x=xA;
                 wall_y=yA-r-wall/2;
              }
           }
          if (yA==0)
          {
           if (xA>0)
            {
              wall_x=xA+r+wall/2;
              wall_y=yA;
            }
           if (xA<0)
            {
               wall_x=xA-r-wall/2;
               wall_y=yA;
            }
           }

       printf("xA %10.6f\n yA%10.6f\n",xA,yA);
       printf("wall x %10.6f\n  wall y %10.6f\n",wall_x,wall_y);

       fprintf(jou,"%s %cvolume.3%c offset %f %f %f\n",copy,quote,quote,wall_x,wall_y,zA);
       fprintf(jou,"volume split %cvolume.%i%c faces %cface.2%c
connected\n",quote,vol_wall,quote,quote,quote);
       fprintf(jou,"volume delete %cvolume.%i%c lowertopology\n",quote,vol_p,quote,quote,quote);
       fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c
keeptool\n",quote,k,quote,quote,vol_wall,quote);
       fprintf(jou,"face create wireframe %cedge.%i%c real\n",quote,edge,quote);
       fprintf(jou,"volume split %cvolume.%i%c faces %cface.%i%c
connected\n",quote,vol_wall,quote,quote,face,quote);
       fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c %cvolume.%i%c
keeptool\n",quote,quote,quote,vol_wall,quote,quote,vol_p,quote);
       }

     if (wall<0)
       {
          out=out+1;
       }
  }
   printf("number of volumes outside boundaries %i\n",out);
   fprintf(jou,"save\n");

pi=4*atan(1);
 volumes=vol_p+1;
 k=5;
 wx=0;

 for (i=1;i<=spheres;i=i+1)
```

```
   {
        k=k+1;
       xA=u[i][1];
       yA=u[i][2];
       zA=u[i][3];

       for  (j=i+1;j<=spheres;j=j+1)
         {
           Dx=xA-u[j][1];
           Dy=yA-u[j][2];
           Dz=zA-u[j][3];

           Ig=u[j][5];
           IIg=Ig+6 ;////because we have 5 basic volumes before the spheres (with id 0,1,2) is ctreated
//and id 0 is volume 1

           dist=pow((pow(Dx,2)+pow(Dy,2)+pow(Dz,2)),0.5);

           xB=xA-0.5*Dx;
           yB=yA-0.5*Dy;
           zB=zA-0.5*Dz;

           if (dist<=10.1)
           {
              if (dist>=10)
               {
                wx=wx+1;
                fprintf(jou,"%s %cvolume.5%c multiple 1 offset %f %f %f\n",copy,quote,quote,xB,yB,zB);
                fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool
\n",quote,k,quote,quote,volumes,quote);
                fprintf(jou,"volume subtract %cvolume.%0.f%c volumes %cvolume.%i%c keeptool
\n",quote,IIg,quote,quote,volumes,quote);
                fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
\n",quote,quote,quote,volumes,quote);
               }
              if (dist<10&&dist>=9.9)
               {
                fprintf(jou,"%s %cvolume.3%c multiple 1 offset %f %f %f\n",copy,quote,quote,xB,yB,zB);
                fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool
\n",quote,k,quote,quote,volumes,quote);
                fprintf(jou,"volume subtract %cvolume.%0.f%c volumes %cvolume.%i%c keeptool
\n",quote,IIg,quote,quote,volumes,quote);
                fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
\n",quote,quote,quote,volumes,quote);
               }
```

```
                }
            }
        }

      printf(" final spheres %i\n",wx);

      fprintf(jou,"save\n");
      vol=5;
     for (i=1;i<=spheres;i=i+1)
        {
            vol=vol+1;
            xA=u[i][1];
            yA=u[i][2];
            zA=u[i][3];
            id1=u[i][5];

              fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
    \n",quote,quote,quote,vol,quote);
        }
        fprintf(jou,"save\n");
      fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.4%c
    keeptool\n",quote,quote,quote,quote);
      fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
      fprintf(jou,"volume delete %cvolume.3%c lowertopology\n",quote,quote);
      fprintf(jou,"volume delete %cvolume.5%c lowertopology\n",quote,quote);
      fprintf(jou,"save\n");

        return 0;
    }
/*-----------------------------------------------------------------------------------------------------------*/
```

4. The less spheres script

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
```

```
char *date;
double Reactor_r=30;                    //the radius of the reactor in the xy plane in mm
int Reactor_h=303;                    //the height of the reactor in the z direction in mm
char *inname="less_spsheres.txt";    //name of ESyS output file
char *outname="output.txt";          //name of output sorted data
char *jouname="NearMiss.jou";         //name of the jou file

//special attention should be given to the dimension of the array if the spheres exceed the number of
//5000

int i,j,k,endi,vol,vol_wall,vol_p,volumes,wx,face,edge,out;
int spheres;
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0,Ig;
double id2=0,IIg;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char colon=(char)58;
char ssl=(char)92;
char *arc_angles;
char *arc_plane;
char *arc_angles_s;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy;
char *move;
double r,rh,tolr,rd;
double dist,metro,wall,wall_x,wall_y;


int main()
{
//create txt and open it
  FILE *input;
  FILE *output;
  input = fopen(inname,"r+");
  output = fopen(outname,"w");

  if (input == NULL)
  {
      printf("Problem opening file %s for reading.\n", inname);
   }

//read the particles,the first line
  fscanf(input,"%d",&spheres);
  printf("at last i read it,number of spheres %d.\n",spheres);

//starting the loop for reading
    for (i=1;i<=spheres;i++)
    {
        for (j=1;j<=22;j++)
        {
             fscanf(input,"%lf",&u[i][j]);
        }
      }
//printing the array
//  printf("the array is\n");
//  for (i=1;i<=spheres;i++)
//    {
//        printf("%lf %lf %lf %lf %lf\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//    }
  printf("________\n");
//close the file
  fclose(input);
//the sort loop
  endi=spheres-1;
  printf("the sort loop starts\n");
  for (i=1;i<=endi;i++)
  {

        for (j=i+1;j<=spheres;j++)
        {
          id1=u[i][5];
          id2=u[j][5];
          if (id2<id1)
          {
            val_i1=u[i][1];
            val_i2=u[i][2];
            val_i3=u[i][3];
            val_i4=u[i][4];
            val_j1=u[j][1];
            val_j2=u[j][2];
            val_j3=u[j][3];
            val_j4=u[j][4];

            u[i][1]=val_j1;
            u[i][2]=val_j2;
            u[i][3]=val_j3;
            u[i][4]=val_j4;
            u[i][5]=id2;

            u[j][1]=val_i1;
            u[j][2]=val_i2;
            u[j][3]=val_i3;
            u[j][4]=val_i4;
            u[j][5]=id1;
            }
        }
    }
//print the sort loop
//  for (i=1;i<=spheres;i++)
//  {
//     printf("%f %f %f %lf %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//     }

//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
    fprintf(output,"%f %f %f %f %0.f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }

//start generating the jou
//fist comes the header
//standard format of header

  FILE *jou;
  jou = fopen(jouname,"w");

    header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
    date="File opened for write Fri Feb 26 18:33:17 2010.";
    char *identif="creating";           //not necessary
  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
  fprintf(jou,"%c %s\n",slash,date);
```

```
//uni tolerance in Gambit
fprintf(jou,"default load %cC%c%c%cGAMBIT2.ini%c\n",quote,colon,ssl,ssl,quote);

//stadard format of the first point vertex.1
  fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);
  copy="volume cmove";
  move="volume move";

//the reactor VOLUME 1
    arc_angles="startangle 0 endangle 360 center";
    arc_plane="xyplane arc";

  fprintf(jou,"edge create radius %f %s %cvertex.1%c
%s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
  fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
  fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);

//the basic particle VOLUME 2
  r=4.9336;
  arc_angles_s="startangle 0 endangle 180 center";

  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",r,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.4%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the wall contact point VOLUME 3
  rh=1.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.6%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the volume for tolerance VOLUME 4
  tolr=0.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",tolr,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.8%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the contact point VOLUME 5
  rd=1.0;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rd,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.10%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

 vol=5;
 //create the basic particles

  for (i=1;i<=spheres;i=i+1)  //the increment of i infuences the number of spheres one cylynder two
spheres i=i+2, one cylinder 3spheres i=i+3 etc.
    {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        fprintf(jou,"%s %cvolume.2%c offset %f %f %f\n",copy,quote,quote,xA,yA,zA);
     }

  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %cbasic%c\n",slash,quote,quote);
  fprintf(jou,"%c %s\n",slash,date);
  fprintf(jou,"save name %cC%c%c%cNearMiss2.dbs%c\n",quote,colon,ssl,ssl,quote);

//tolerance solve
  fprintf(jou,"%s %cvolume.4%c offset -18.467433 7.285868 298.837618\n",move,quote,quote);
  fprintf(jou,"volume subtract %cvolume.779%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);
    //wall contact
 k=5;
 vol_wall=vol-1;
 fprintf(jou,"save\n");
 vol=5;
 for (i=1;i<=spheres-2;i=i+1)
    {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        id1=u[i][5];
       fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
\n",quote,quote,quote,vol,quote);
    }
    fprintf(jou,"save\n");
  fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);
  fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.779%c
keeptool\n",quote,quote,quote,quote);
  fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.780%c \n",quote,quote,quote,quote);
```

```
   fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
   fprintf(jou,"volume delete %cvolume.3%c lowertopology\n",quote,quote);
   fprintf(jou,"volume delete %cvolume.5%c lowertopology\n",quote,quote);
   fprintf(jou,"save\n");

     return 0;
 }
/*---------------------------------------------------------------------------------------------------------*/
```

5. The periodic segment

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
char *date;
double Reactor_r=30;                //the radius of the reactor in the xy plane in mm
int Reactor_h=303;                  //the height of the reactor in the z direction in mm
char *inname="periosic_segment.txt";    //name of ESyS output file
char *outname="output50.txt";           //name of output sorted data
char *jouname="50.jou";         //name of the jou file

//special attention should be given to the dimension of the array if the spheres exceed the number of
//5000

int i,j,k,endi,vol,vol_wall,vol_p,volumes,wx,face,edge,out;
int spheres;
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0,Ig;
double id2=0,IIg;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char colon=(char)58;
char ssl=(char)92;
char *arc_angles;
char *arc_plane;
char *arc_angles_s;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy;
char *move;
double r,rh,tolr,rd;
double dist,metro,wall,wall_x,wall_y;


int main()
{
//create txt and open it
  FILE *input;
  FILE *output;

  input = fopen(inname,"r+");
  output = fopen(outname,"w");

  if (input == NULL)
  {
      printf("Problem opening file %s for reading.\n", inname);
   }

//read the particles,the first line
  fscanf(input,"%d",&spheres);
  printf("at last i read it,number of spheres %d.\n",spheres);

//starting the loop for reading
    for (i=1;i<=spheres;i++)
    {
        for (j=1;j<=22;j++)
        {
             fscanf(input,"%lf",&u[i][j]);
        }
      }

//printing the array
//  printf("the array is\n");

//  for (i=1;i<=spheres;i++)
//    {
```

```
//        printf("%lf %lf %lf %lf %lf\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//    }
  printf("________\n");
//close the file
  fclose(input);
//the sort loop
  endi=spheres-1;
  printf("the sort loop starts\n");
  for (i=1;i<=endi;i++)
  {
        for (j=i+1;j<=spheres;j++)
        {
          id1=u[i][5];
          id2=u[j][5];
          if (id2<id1)
          {
            val_i1=u[i][1];
            val_i2=u[i][2];
            val_i3=u[i][3];
            val_i4=u[i][4];

            val_j1=u[j][1];
            val_j2=u[j][2];
            val_j3=u[j][3];
            val_j4=u[j][4];

            u[i][1]=val_j1;
            u[i][2]=val_j2;
            u[i][3]=val_j3;
            u[i][4]=val_j4;
            u[i][5]=id2;

            u[j][1]=val_i1;
            u[j][2]=val_i2;
            u[j][3]=val_i3;
            u[j][4]=val_i4;
            u[j][5]=id1;
            }
        }
    }
//print the sort loop
//  for (i=1;i<=spheres;i++)
//  {
//     printf("%f %f %f %lf %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
//     }
```

```
//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
     fprintf(output,"%f %f %f %f %0.f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }

//start generating the jou
//fist comes the header
//standard format of header

  FILE *jou;
  jou = fopen(jouname,"w");

    header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
    date="File opened for write Fri Feb 26 18:33:17 2010.";
    char *identif="creating";             //not necessary
  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
  fprintf(jou,"%c %s\n",slash,date);
//uni tolerance in Gambit
fprintf(jou,"default load %cC%c%c%cGAMBIT2.ini%c\n",quote,colon,ssl,ssl,quote);

//stadard format of the first point vertex.1
  fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);
  copy="volume cmove";
  move="volume move";

//the reactor VOLUME 1
    arc_angles="startangle 0 endangle 360 center";
    arc_plane="xyplane arc";

  fprintf(jou,"edge create radius %f %s %cvertex.1%c
%s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
  fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
  fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);

//the basic particle VOLUME 2
  r=4.95;
  arc_angles_s="startangle 0 endangle 180 center";
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",r,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.4%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);
```

```
//the helpful sphere for the wall contact point VOLUME 3
  rh=1.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.6%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the volume for tolerance VOLUME 4
  tolr=0.5;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",tolr,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.8%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

//the helpful sphere for the contact point VOLUME 5
  rd=1.0;
  fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rd,arc_angles_s,quote,quote,arc_plane);
  fprintf(jou,"edge create straight %cvertex.5%c %cvertex.4%c\n",quote,quote,quote,quote);
  fprintf(jou,"face create wireframe %cedge.3%c %cedge.4%c real\n",quote,quote,quote,quote);
  fprintf(jou,"volume create revolve %cface.10%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);

 vol=5;
 //create the basic particles

  for (i=1;i<=spheres;i=i+1)  //the increment of i infuences the number of spheres one cylynder two
spheres i=i+2, one cylinder 3spheres i=i+3 etc.
    {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        fprintf(jou,"%s %cvolume.2%c offset %f %f %f\n",copy,quote,quote,xA,yA,zA);
     }

  fprintf(jou,"%c %s\n",slash,header);
  fprintf(jou,"%c Identifier %cbasic%c\n",slash,quote,quote);
  fprintf(jou,"%c %s\n",slash,date);
  fprintf(jou,"save name %cC%c%c%cAllo50.dbs%c\n",quote,colon,ssl,ssl,quote);

//tolerance solve

  fprintf(jou,"%s %cvolume.4%c offset -29 0 12\n",move,quote,quote);
  fprintf(jou,"volume subtract %cvolume.66%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);
    //wall contact
 k=5;
 vol_wall=vol-1;
 face=793-9; //every vol has one surface but the reactor has 2 more in addition four faces missing
//4+6+8+10 cause they became vol
 //plus one created from subtract
 edge=6-9; //2 the cylinder and 5 the volumes
 out=0;

  for (i=1;i<=spheres;i=i+1)
    {
        k=k+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];

        metro=pow((pow(xA,2)+pow(yA,2)),0.5);
        f=atan(fabs(yA/xA));
         wall=Reactor_r-r-metro;
      //  printf("metro %10.6f\n angle f in rad %10.6f\n wall distance %10.6f\n",metro,f,wall);

        if (wall<=0.11&&wall>=0)
            {
                vol_wall=vol_wall+2;
                vol_p=vol_wall+1;
                face=face+9;
                edge=edge+9;
                printf("wol_wall %i in the if state\n",vol_wall);

                if (xA>0&&yA>0)
                {
                 wall_x=xA+(0.5*wall+r)*cos(f);
                 wall_y=yA+(0.5*wall+r)*sin(f);
                }

                if (xA<0&&yA>0)
                {
                 wall_x=xA-(0.5*wall+r)*cos(f);
                 wall_y=yA+(0.5*wall+r)*sin(f);
                }

                if (xA<0&&yA<0)
                {
                 wall_x=xA-(0.5*wall+r)*cos(f);
                 wall_y=yA-(0.5*wall+r)*sin(f);
                 }
                if (xA>0&&yA<0)
                {
                 wall_x=xA+(0.5*wall+r)*cos(f);
                 wall_y=yA-(0.5*wall+r)*sin(f);
                }

                if (xA==0)
                 {
                   if (yA>0)
                    {
                        wall_x=xA;
                        wall_y=yA+r+wall/2;
                       }
                   if (yA<0)
                     {
                         wall_x=xA;
                         wall_y=yA-r-wall/2;
                      }
                  }
                if (yA==0)
                 {
                   if (xA>0)
                    {
                       wall_x=xA+r+wall/2;
                       wall_y=yA;
                     }
                   if (xA<0)
                    {
                         wall_x=xA-r-wall/2;
                         wall_y=yA;
                    }
                  }

          //  printf("xA %10.6f\n yA%10.6f\n",xA,yA);
          //  printf("wall x %10.6f\n  wall y %10.6f\n",wall_x,wall_y);

           fprintf(jou,"%s %cvolume.3%c offset %f %f %f\n",copy,quote,quote,wall_x,wall_y,zA);
           fprintf(jou,"volume split %cvolume.%i%c faces %cface.2%c
connected\n",quote,vol_wall,quote,quote,quote);
           fprintf(jou,"volume delete %cvolume.%i%c lowertopology\n",quote,vol_p,quote,quote,quote);
           fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c
keeptool\n",quote,k,quote,quote,vol_wall,quote);
           fprintf(jou,"face create wireframe %cedge.%i%c real\n",quote,edge,quote);
```

```
            fprintf(jou,"volume split %cvolume.%i%c faces %cface.%i%c
connected\n",quote,vol_wall,quote,quote,face,quote);
            fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c %cvolume.%i%c
keeptool\n",quote,quote,quote,vol_wall,quote,quote,vol_p,quote);
            }

         if (wall<0)
            {
                out=out+1;
            }

    }
     printf("number of volumes outside boundaries %i\n",out);
     printf("volumes after wall %i\n",vol_wall);
     fprintf(jou,"save\n");

pi=4*atan(1);
  volumes=vol_wall+1;
  k=5;
  wx=0;
out=0;

  for (i=1;i<=spheres;i=i+1)
    {

          k=k+1;
         xA=u[i][1];
         yA=u[i][2];
         zA=u[i][3];
         f=u[i][5];

         for  (j=i+1;j<=spheres;j=j+1)
           {
              Dx=xA-u[j][1];
              Dy=yA-u[j][2];
              Dz=zA-u[j][3];

              Ig=u[j][5];
              IIg=j+5 ;////because we have 5 basic volumes before the spheres (with id 0,1,2) is ctreated and
id 0 is volume 6
             // printf("Ig is %0.f while IIg is %0.f\n",Ig+6,IIg);
              dist=pow((pow(Dx,2)+pow(Dy,2)+pow(Dz,2)),0.5);

              xB=xA-0.5*Dx;
              yB=yA-0.5*Dy;
              zB=zA-0.5*Dz;

              if (dist<=10.01)
              {
                   volumes=volumes+1;
                  if (dist>=9.8)
                   {
                    fprintf(jou,"%s %cvolume.5%c multiple 1 offset %f %f %f\n",copy,quote,quote,xB,yB,zB);
                    fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool
\n",quote,k,quote,quote,volumes,quote);
                    fprintf(jou,"volume subtract %cvolume.%0.f%c volumes %cvolume.%i%c keeptool
\n",quote,llg,quote,quote,volumes,quote);
                    fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
keeptool\n",quote,quote,quote,volumes,quote);
                   }
                  if (dist<=9.8&&dist>=5)
                   {
                    wx=wx+1;
                    fprintf(jou,"%s %cvolume.3%c multiple 1 offset %f %f %f\n",copy,quote,quote,xB,yB,zB);
                    fprintf(jou,"volume subtract %cvolume.%i%c volumes %cvolume.%i%c keeptool
\n",quote,k,quote,quote,volumes,quote);
                    fprintf(jou,"volume subtract %cvolume.%0.f%c volumes %cvolume.%i%c keeptool
\n",quote,llg,quote,quote,volumes,quote);
                    fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
keeptool\n",quote,quote,quote,volumes,quote);
                   }
                  if (dist<=9.8)
                  {
                      out=out+1;
                      printf(" main particle\n %i",k);
                      printf(" overlaps with %0.f\n",llg);
                  }
               }
           }
    }

 printf(" spheres less than 9.967,number  %i\n",volumes);
 printf(" overlap implemented sheres %i\n",wx);
 fprintf(jou,"save\n");
 vol=5;
 for (i=1;i<=spheres;i=i+1)
    {
        vol=vol+1;
        xA=u[i][1];
        yA=u[i][2];
        zA=u[i][3];
        id1=u[i][5];
        fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.%i%c
keeptool\n",quote,quote,quote,vol,quote);
    }

    fprintf(jou,"save\n");
  fprintf(jou,"volume subtract %cvolume.1%c volumes %cvolume.4%c
keeptool\n",quote,quote,quote,quote);

  fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
  fprintf(jou,"volume delete %cvolume.3%c lowertopology\n",quote,quote);
  fprintf(jou,"volume delete %cvolume.5%c lowertopology\n",quote,quote);
  fprintf(jou,"save\n");

    return 0;
}
/*-----------------------------------------------------------------------------------------------------------*/
```

Appendix IV : C Codes for simulations with cylinders

Geometry generation with cylindrical particles

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>



/*------------------------- INPUT VARIABLES----------------------------*/
double u[5000][5];
char *date;
int Reactor_r=30;                  //the radius of the reactor in the xy plane in mm
int Reactor_h=300;                  //the height of the reactor in the z direction in mm
char *inname="cylinders.txt";    //name of ESyS output file
char *outname="output.txt";          //name of output sorted data
char *jouname=" cylinders.jou";        //name of the jou file



//special attention should be given to the dimension of the array if the spheres exceed the number of
//5000

int i,j,k,endi,vol;
int spheres;
```

```
double val_i1=0;
double val_i2=0;
double val_i3=0;
double val_i4=0;
double val_j1=0;
double val_j2=0;
double val_j3=0;
double val_j4=0;
double id1=0;
double id2=0;
double xA,yA,zA;
double xB,yB,zB;
char *header;
char quote=(char)34;
char slash=(char)47;
char *arc_angles;
char *arc_plane;
double Dx,Dy,Dz,xy,f,f1,pi,w,f2;
char *copy,*move;
char *arc_angles_s;
double new_r;
double r,rh;
double cylinder_h=20;                  //the height of cylindrical partical in the z direction in mm

int main()
{
//create txt and open it
  FILE *input;
  FILE *output;

  input = fopen(inname,"r+");
  output = fopen(outname,"w");

  if (input == NULL)
  {
     printf("Problem opening file %s for reading.\n", inname);
   }



//read the particles,the first line
  fscanf(input,"%d",&spheres);
  printf("at last i read it,number of spheres %d.\n",spheres);

//starting the loop for reading
    for (i=1;i<=spheres;i++)
  {
     for (j=1;j<=22;j++)
     {
         fscanf(input,"%lf",&u[i][j]);



     }
   }
//printing the array
 // printf("the array is\n");
 // for (i=1;i<=spheres;i++)
 //   {
  //      printf("%lf %lf %lf %lf %lf\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
   // }
  printf("________\n");
//close the file
  fclose(input);
//the sort loop
  endi=spheres-1;
  printf("the sort loop starts\n");
  for (i=1;i<=endi;i++)
  {

     for (j=i+1;j<=spheres;j++)
     {
       id1=u[i][5];
       id2=u[j][5];
       if (id2<id1)
       {
         val_i1=u[i][1];
         val_i2=u[i][2];
         val_i3=u[i][3];
         val_i4=u[i][4];

         val_j1=u[j][1];
         val_j2=u[j][2];
         val_j3=u[j][3];
         val_j4=u[j][4];




         u[i][1]=val_j1;
         u[i][2]=val_j2;
         u[i][3]=val_j3;
         u[i][4]=val_j4;
```

```
            u[i][5]=id2;

            u[j][1]=val_i1;
            u[j][2]=val_i2;
            u[j][3]=val_i3;
            u[j][4]=val_i4;
            u[j][5]=id1;
            }
        }
    }
//print the sort loop
  for (i=1;i<=spheres;i++)
  {
     printf("%f %f %f %lf %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }
//write to the file the final data
  for (i=1;i<=spheres;i++)
  {
     fprintf(output,"%f %f %f %f %f\n",u[i][1],u[i][2],u[i][3],u[i][4],u[i][5]);
  }

  fclose(output);
//start generating the jou

//fist comes the header

//standard format of header

FILE *jou;
jou = fopen(jouname,"w");

  header="Journal File for GAMBIT 2.4.6, Database 2.4.4, ntx86 SP2007051421";
  date="File opened for write Fri Feb 26 18:33:17 2010.";
  char *identif="creating";            //not necessary
fprintf(jou,"%c %s\n",slash,header);
fprintf(jou,"%c Identifier %c%s%c\n",slash,quote,identif,quote);
fprintf(jou,"%c %s\n",slash,date);

//stadard format of the first point vertex.1
fprintf(jou,"vertex create %cvertex.1%c coordinates 0 0 0\n",quote,quote);



//the reactor
  arc_angles="startangle 0 endangle 360 center";
  arc_plane="xyplane arc";
```

```
fprintf(jou,"edge create radius %d %s %cvertex.1%c
%s\n",Reactor_r,arc_angles,quote,quote,arc_plane);
fprintf(jou,"face create wireframe %cedge.1%c real\n",quote,quote);
fprintf(jou,"volume create translate %cface.1%c vector 0 0 %d\n",quote,quote,Reactor_h);

//the basic particle
r=5;
new_r=(3-pow(2,0.5))*r*0.5;
cylinder_h=4*new_r;                //the height of cylindrical partical in the z direction in mm
 printf("%f %f %f\n",r,new_r,cylinder_h);
fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",new_r,arc_angles,quote,quote,arc_plane);
fprintf(jou,"face create wireframe %cedge.3%c real\n",quote,quote);
fprintf(jou,"volume create translate %cface.4%c vector 0 0 %f\n",quote,quote,cylinder_h);
fprintf(jou,"volume move %cvolume.2%c offset 0 0 -%f\n",quote,quote,new_r);
//the helpful sphere for the contact point VOLUME 3
rh=1.1;
 arc_angles_s="startangle 0 endangle 180 center";
fprintf(jou,"edge create radius %f %s %cvertex.1%c %s\n",rh,arc_angles_s,quote,quote,arc_plane);
fprintf(jou,"edge create straight %cvertex.7%c %cvertex.6%c\n",quote,quote,quote,quote);
fprintf(jou,"face create wireframe %cedge.6%c %cedge.5%c real\n",quote,quote,quote,quote);
fprintf(jou,"volume create revolve %cface.7%c dangle 360 vector 1 0 0 origin 0 0 0\n",quote,quote);


//start the loop of math,move,rotate e.t.c for the particles
vol=3;
for (i=1;i<=spheres;i=i+2)
 {
   vol=vol+1;
   j=i+1;
     xA=u[i][1];
     yA=u[i][2];
     zA=u[i][3];

     xB=u[j][1];
     yB=u[j][2];
     zB=u[j][3];

     Dx=xB-xA;
     Dy=yB-yA;
     Dz=zB-zA;

     xy=pow((pow(Dx,2)+pow(Dy,2)),0.5);
     f=atan(fabs(Dy/Dx));
     w=atan(fabs(Dz/xy));
```

```
    pi=4*atan(1);

  //  printf("%f %f %f rads %lf\n",Dx,Dy,xy,f);
    copy="volume cmove";
    move="volume move";

    if (Dz>0)
     {
       f2=90-(w*180/pi);
       fprintf(jou,"%s %cvolume.2%c multiple 1 dangle %f vector 0 1 0 origin 0 0
0\n",copy,quote,quote,f2);
     }
    if (Dz<0)
     {
       f2=90+(w*180/pi);
       fprintf(jou,"%s %cvolume.2%c multiple 1 dangle %f vector 0 1 0 origin 0 0
0\n",copy,quote,quote,f2);
     }

    // printf("%f %f\n",f2,pi);

    if (Dx>0&&Dy>0)
     {
       f1=f*180/pi;
      fprintf(jou,"%s %cvolume.%d%c dangle %f vector 0 0 1 origin 0 0 0\n",move,quote,vol,quote,f1);
      }
    if (Dx<0&&Dy>0)
     {
        f1=(f*180/pi)-180;
       fprintf(jou,"%s %cvolume.%d%c dangle %f vector 0 0 1 origin 0 0 0\n",move,quote,vol,quote,f1);
      }
    if (Dx<0&&Dy<0)
     {
        f1=(f*180/pi)+180;
       fprintf(jou,"%s %cvolume.%d%c dangle %f vector 0 0 1 origin 0 0 0\n",move,quote,vol,quote,f1);
      }
    if (Dx>0&&Dy<0)
     {
        f1=360-(f*180/pi);
       fprintf(jou,"%s %cvolume.%d%c dangle %f vector 0 0 1 origin 0 0 0\n",move,quote,vol,quote,f1);
      }

   // printf("%f %f\n",f1,pi);

    if (vol>spheres-2)
```

```
        {
            break;
        }

fprintf(jou,"volume delete %cvolume.2%c lowertopology\n",quote,quote);
    return 0;
}
/*---------------------------------------------------------------------------------------------------------*/
```

Appendix V : Mesh generation for cylinders (model spheres case study)

1. Contact at the periphery of two cylinders

(Tube 60x50cm containing 5 cylinders that were not in contact and two in contact .dimension of cylinders 10x15 cm)

In the following cases, worst and best case of touching cylinders at the periphery of their surface, the mesh generated without model spheres and with model spheres will be presented.

1.1 Without model spheres

1.1.1 The worst case:

Two cylinders were examined. One contact point in the periphery of the first vertical cylinder while the second was turned 5 degrees in the zx plane

Subtract:

- No subtract one another because only one contact point
- No virtual faces or edges while subtract from the tube

Mesh particles:

-Cooper mesh for the cylinders was created without a problem for any size

-T- Grid mesh could not be created for any size but with the procedure explained afterwards it worked .Sometimes it is also possible that Gambit doesn’t t work properly.

Mesh the tube:

No mesh could be created with the meshed particles by Cooper.

Error: TG mesh domain failed with error code 1
Error: Tetrahedral meshing has failed for volume 61. This is usually caused by problems in the face meshes .Check the skewness of your face meshes and make sure the face meshes sizes are not too large in areas of small gaps.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	918.333	8	0.999998
2	1	244.058	3	0.999899
3	1	160.92	2	0.999971
4	1	152.443	4	0.999707
1	2	237.759	8	0.999998
2	2	130.545	3	0.999899
3	2	52.531	2	0.999971
4	2	41.35	3	0.999707
1	3	169.983	9	0.999998
2	3	58.701	3	0.999899
3	3	34.335	2	0.999971
4	3	25.607	3	0.999707
1	4	147.329	12	0.999998
2	4	34.916	3	0.999899
3	4	20.119	2	0.999971
4	4	17.423	3	0.999707

1.1.2 The best case:

Two cylinders were examined. One contact point in the periphery of the first vertical cylinder while the second was turned 45 degrees in the zx plane. The contact point and the first vertical cylinder were at the same position as before. One of the other cylinders with no contact points had to be moved.

Subtract:
- No subtract one another because only one contact point
- No virtual faces or edges while subtract from the tube

Mesh particles:

-Cooper mesh for the cylinders was created without a problem for any size

-T- grid mesh could be created with a specific procedure:

* Trying to unlink and delete mesh volumes, faces, edges the cylinders and then mesh the face that had the problem by meshing the edge and after the face did not work out. The procedure that was finally appropriate is by meshing the face of the circle that can be meshed then the periphery and then the face of the circle that has the problem. After meshing the volume you can obtain a mesh with a scheme of t-grid

Mesh the tube:

-No mesh could be created with the meshed particles by Cooper and T-grid

- T-grid and T-grid meshed

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	918.013	3	0.999998
2	1	244.062	1	0.999899
3	1	161.414	2	0.999971
4	1	151.929	2	0.999707
1	2	238.451	3	0.999998
2	2	130.717	1	0.999899
3	2	52.218	2	0.999971
4	2	41.06	1	0.999707
1	3	170.621	4	0.999998
2	3	58.672	1	0.999899
3	3	34.199	2	0.999971
4	3	25.482	1	0.999707
1	4	147.359	7	0.999998
2	4	34.737	1	0.999899
3	4	20.103	2	0.999971
4	4	17.475	1	0.999707

1.2 With model spheres

When spheres are created no cooper scheme can be used.

1.2.1 The worst case :

Two cylinders were examined. One contact point in the periphery of the first vertical cylinder while the second was turned 5 degrees in the zx plane

- For sphere 0,1.:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges (Gambit2.uni.)

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be created the specific procedure

-Mesh volume of the sphere 20

Mesh the tube:

-Tgrid 1-1 did not work

Error: TG mesh domain failed with error code 1

Error: Tetrahedral meshing has failed for volume 61.This is usually caused by problems in the face meshes .Check the skewness of your face meshes and make sure the face meshes sizes are not too large in areas of small gaps.

-Tgrid 1-2 did not work

Because the volume in angle with the contact point has 4 skewed elements the worst of which is 0,97528 so the tube can not be meshed with error.

Error: Meshing volume 61 aborted due to poor mesh quality

Error. The user may proceed by first meshing the faces with the poor mesh quality or attempt to mesh the volume with the existing face meshes

As the mesh size of the particles increases so does the number of highly skewed elements, no attempts will be made for further analysis for the sphere 0,1.

Way 2. Subtract cylinder from the sphere and retain the cylinder

Virtual edges

- For sphere 0,2.:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges while subtract from the tube (Gambit2.uni)

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be created the specific procedure

-Mesh volume of the sphere 24

-Mesh of volume in angle contains 17 h.el. for tgid.1 for larger size no h.s.el.

Mesh the tube:

-Tgrid 1-1 did not work

-Tgrid 2-1

As the mesh size of the particles increases so does the number of highly skewed elements, no attempts will be made for further analysis for the sphere 0,1.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	X	X	X
2	1	246.545	27	0.999267
3	1	162.827	29	0.99965
1	2	X	X	X
2	2	133.141	29	0.999267
3	2	53.247	29	0.99965
1	3	X	X	X
2	3	60.461	27	0.999267
3	3	35262	29	0.99965
1	4	X	X	X
2	4	36841	27	0.999267
3	4	21091	29	0.99965

Way 2. Subtract cylinder from the sphere and retain the cylinder

Virtual edges were created.

- For sphere 0,3.:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be created the specific procedure

-Mesh volume of the sphere 24

Mesh of volume in angle contains could not be meshed at all with interval size of 1 but after 2 it could

Mesh the tube:

-Tgrid 1-1 did not work

-Tgrid 2-1

As the mesh size of the particles increases so does the number of highly skewed elements, no attempts will be made for further analysis for the sphere 0,1.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	X	X	X
2	1	245.988	17	0.998747

Way 2. Subtract cylinder from the sphere and retain the cylinder

Virtual edges

- For sphere 0,8:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be created only for 1 .Problems occurred from 2

-T- grid mesh could be created with the above procedure. Tri pave surface scheme was used for the problematic surfaces, which were successfully meshed after many repetitions.

-Mesh volume of the sphere 26

Mesh the tube:

-Meshing the tube ended in many high skewed elements that is really undesirable in this case study. The tube will contain hundreds of particles and more than double contact points, so meshes with zero high elements will be preferred.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	866.233	1	0.992396
2	1	245.646	3	0.995245
3	1	161.969	3	0.995677
4	1	150.939	4	0.99649
1	2	237737	1	0.992396
2	2	131.737	3	0.995245
3	2	52.945	3	0.995677
4	2	38.205	3	0.99649
1	3	168.382	2	0.992396
2	3	59.011	3	0.995245
3	3	34.825	3	0.995677
4	3	21.871	3	0.99649
1	4	146.605	5	0.992396
2	4	35.451	3	0.995245
3	4	20.561	3	0.995677
4	4	16.889	3	0.99649

Way 2. Subtract cylinder from the sphere and retain the cylinder No virtual edges. But the sphere cannot be meshed.

- For sphere 1,0:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T-grid mesh could be easily created

-Mesh volume of the sphere 39

Mesh the tube:

-T-grid from 3-1 works .In order to mesh the volume in angle the above procedure was followed and size functions were ignored.

-The tube is meshed with zero high elements only when particles have a mesh size of 3.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	914.163	2	0.997814
2	1	245.168	3	0.993799
3	1	160.964	0	0.944889
4	1	150.641	5	0.995081
1	2	236.263	2	0.997814
2	2	131.484	3	0.993799
3	2	52.593	0	0.937893
4	2	38.232	4	0.995081
1	3	166.547	3	0.997814
2	3	58890	3	0.993799
3	3	34.761	0	0.947047
4	3	21929	4	0.995081
1	4	144.776	6	0.997814
2	4	35.362	3	0.993799
3	4	20.194	0	0.95674
4	4	17064	4	0.995081

Way 2. Subtract cylinder from the sphere and retain the cylinder.No virtual faces or edges but the volume is too small to be meshed even with size of 1

- For sphere 1,1:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T-grid mesh could be easily created

-Mesh volume of the sphere 30

Mesh the tube:

-T-grid works.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	916.537	1	0.989839
2	1	236.259	0	0.943596
3	1	161.543	0*	0.944889
4	1	151.983	8	0.993282
1	2	234.573	1	0.989839
2	2	131465	0	0.943596
3	2	52.815	0*	0.939851
4	2	39.638	5	0.993282
1	3	164.918	2	0.989839
2	3	59.348	0*	0.947047
3	3	34.646	0*	0.947047
4	3	23.806	5	0.993282
1	4	143.231	4	0.989839
2	4	35.414	0*	0.95674
3	4	20.311	0*	0.95674
4	4	17.757	5	0.993282

* It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

Way 2. Subtract cylinder from the sphere and retain the cylinder. No virtual faces or edges but the volume is too small to be meshed even with size of 1

- For sphere 1,2:

Way 1. Subtract sphere from the cylinder and retain the sphere
Subtract 1:
- No virtual faces or edges while subtract sphere from the cylinder
Subtract 2:
- No virtual faces or edges (Gambit2.uni).

Mesh particles:
-Cooper mesh for the cylinders is not available with spheres.
-T- grid mesh could be easily created
-Mesh volume of the sphere 30

Mesh the tube:
-Tgrid works

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	916.537	1	0.989839
2	1	236.259	0	0.943596
3	1	161.543	0*	0.944889
4	1	151.983	8	0.993282
1	2	234.573	1	0.989839
2	2	131465	0	0.943596
3	2	52.815	0*	0.939851
4	2	39.638	5	0.993282
1	3	164.918	2	0.989839
2	3	59.348	0*	0.947047
3	3	34.646	0*	0.947047
4	3	23.806	5	0.993282
1	4	143.231	4	0.989839
2	4	35.414	0*	0.95674
3	4	20.311	0*	0.95674
4	4	17.757	5	0.993282

* It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

Way 2. Subtract cylinder from the sphere and retain the cylinder. No virtual faces or edges but the volume is too small to be meshed even with size of 1

- For sphere 1,3:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges while subtract (Gambit2.uni.)

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be easily created

Mesh the tube:

-Tgrid works .

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element	skweness at contact point
1	1	650.34	0	0.960138	0.960138
2	1	244.62	0	0.932018	0.932018
3	1	161.315	0*	0.944889	0.932018
4	1	151.594	4	0.992191	0.992191
1	2	228.842	5*	0.990647	0.960138
2	2	130.545	0	0.932018	0.932018
3	2	52232	0	0.936787	0.936787
4	2	40.144	3	0.992191	0.992191
1	3	160.468	15*	0.995401	0.960138
2	3	58.406	0*	0.947047	0.932018
3	3	34.495	0*	0.947047	0.932018
4	3	24.128	3	0.992191	0.992191
1	4	137.717	24*	0.999218	0.960138
2	4	34.721	0*	0.95674	0.932018
3	4	20.233	0*	0.95674	0.932018
4	4	17.492	3	0.992191	0.992191

* It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

Way 2. Subtract cylinder from the sphere and retain the cylinder. No virtual faces or edges but the volume is too small to be meshed even with size of 1

- For sphere 1,5:

Way 1. Subtract sphere from the cylinder and retain the sphere

Subtract 1:

- No virtual faces or edges while subtract sphere from the cylinder

Subtract 2:

- No virtual faces or edges (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be created with the above procedure and map tri pave on the problematic surfaces

-Mesh volume of the sphere 30

Mesh the tube:

-The worst element on case 1-3 and 1-4 the worst element appears near the outlet surface and the closed cylinder.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	656.931	0	0.957184
2	1	245.325	2	0.987631
3	1	161.371	2	0.989881
4	1	151.999	4	0.990292
1	2	233.692	0	0.957184
2	2	130.978	2	0.987631
3	2	56.655	2	0.989881
4	2	40.771	3	0.990292
1	3	164.433	1*	0.971269
2	3	58.874	2	0.987631
3	3	34.462	2	0.989881
4	3	24.847	3	0.990292
1	4	142.531	4	0.982793
2	4	35.425	2	0.987631
3	4	20.4	2	0.989881
4	4	17.589	3	0.990292

Way 2. Subtract cylinder from the sphere and retain the cylinder. No virtual faces or edges but the volume is too small to be meshed even with size of 1

CONCLUSIONS:

When two cylinders have their contact point in their periphery then the contact point can be replaced by a sphere. the radius of the sphere depends on the mesh size that will be chosen . The minimum radius for which we have no highly skewed elements is:

Tgrid 1-1: sphere 1,2

Tgrid 2-1: sphere 1,1

Tgrid 3-1: sphere 1

Tgrid 4-1: sphere over 1,5

Tgrid 1-2: sphere 1,2

Tgrid 2-2: sphere 1,1

Tgrid 3-2: sphere 1

Tgrid 4-2: sphere over 1,5

Tgrid 1-3: sphere 1,2

Tgrid 2-3: sphere 1,1

Tgrid 3-3: sphere 1

Tgrid 4-3: sphere over 1,5

Tgrid 1-4: sphere 1,2

Tgrid 2-4: sphere 1,1

Tgrid 3-4: sphere 1

Tgrid 4-4 :sphere over 1,5

<u>Criteria:</u>

- The smallest sphere possible.(because we may have convergence problems that may be provoqued by the continuity equation and also more realistic model with the less volume instead of contact point.

- The size of grid that will give independent solutions .Here we should bare in mind that we will simulate with a laminar flow (not so fine grid)

- We have to decide by try and error maybe in Fluent which is the best resolution more accurate surface mesh on the particles , equality or more accurate fluid mesh .

***No confusion to be caused the high elements that occur due to the geometry that was chosen and not due to the contact points replaced by spheres have one * that means that the skewness in the contact point is under 0,97 and sometimes much more low.

1.2.2 The best case :

The comparison with the best case will be done for the cases 1-1,1-1,2:

- For sphere 1.0:

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	914.477	0	0.855421
2	1	X	X	X
3	1	163.588	0*	0.944889
4	1	X	X	X
1	2	236.307	0	0.861591
2	2	X	X	X
3	2	54.893	0	0.942219
4	2	X	X	X
1	3	167.012	1*	0.971269
2	3	X	X	X
3	3	34.588	0*	0.947047
4	3	X	X	X
1	4	144.868	4*	0.982793
2	4	X	X	X
3	4	20216	0*	0.95674
4	4	X	X	X

The tube could not be meshed when the mesh size of the cylinders is 2 or 4. The error that Gambit shows is :

Error: TG mesh domain failed with error code 1
Error: Tetrahedral meshing has failed for volume 61.This is usually caused by problems in the face meshes .Check the skewness of your face meshes and make sure the face meshes sizes are not too large in areas of small gaps.
The face meshes were checked and the worst 2´D element found to be 0,77.

- For sphere 1.1:

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	651.442	0*	0.905233
2	1	245.739	0	0.884926
3	1	163.592	0*	0.944889
4	1	152.442	1*	0.976807
1	2	229.395	5*	0.990647
2	2	130.897	0	0.884926
3	2	54.806	0	0.900849
4	2	39.727	0*	0.909503
1	3	159.302	15*	0.995401
2	3	59.113	0*	0.947047
3	3	34.417	0*	0.947047
4	3	23.824	0*	0.947047
1	4	138.564	24*	0.999218
2	4	35.639	0*	0.95674
3	4	20.22	0*	0.95674
4	4	17.647	0*	0.95674

- For sphere 1,2:

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	916.559	0*	0.791742
2	1	244.573	0	0.900276
3	1	163.646	0*	0.944889
4	1	151.363	1*	0.976807
1	2	238.396	0*	0.861591
2	2	130.992	0	0.900276
3	2	54.919	0	0.885012
4	2	39.934	0	0.942185
1	3	168.591	1*	0.971269
2	3	59.224	0*	0.947047
3	3	54.919	0*	0.947047
4	3	23.959	0*	0.947047
1	4	147.052	4*	0.982793
2	4	35.168	0*	0.95674
3	4	20.182	0*	0.95674
4	4	17.644	0*	0.95674

No problem occurs with the skewness of the best case and with the exception of meshing problems of the sphere 1.0. We conclude that the worst case will determine the choice of the spheres. For the contact points at the periphery, model spheres will be used to simulate contact points.

1. Contact at the circular surfaces of two cylinders

2.1 Position1: 100% touch between the two circular surfaces

- For sphere 1,0:

Subtract :

- No virtual faces or edges while subtract sphere from the cylinder

- No virtual faces or edges (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be easily created

-Mesh volume of the sphere 30

Mesh the tube:

-T-grid works for all the cases

For T-Grid 1-1 one highly skewed element appears on the sphere

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	918.92	1	0.989839
2	1	235.895	0*	0.943603
3	1	161.486	0*	0.944889
4	1	151.831	8	0.993282
1	2	262.752	1	0.989839
2	2	131.113	0*	0.943603
3	2	52.63	0	0.939851
4	2	39.826	7	0.993282
1	3	197.924	2	0.989839
2	3	62.248	0*	0.947047
3	3	34.521	0*	0.947047
4	3	23.664	5	0.993282
1	4	177.982	5	0.989839
2	4	40.517	0*	0.95674
3	4	20.955	0*	0.95674
4	4	17.759	5	0.993282

The two cylinders added, influenced the mesh of the tube, thus the mesh was created only for the sizes 2-1, 3-1, 2-2, 3-2, 2-3, 3-3, 2-4, 3-4.

* It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

2.2 Position2: 50% touch between the two circular surfaces

- For sphere 1,1:

Subtract:

- No virtual faces or edges while subtract sphere from the cylinder

- Virtual edges occurred while subtract from the tube even if Gambit2.uni was used for lower tolerance.

Solution: Unite volumes (and not retain). After uniting the two volumes no virtual faces or edges occurred.

Mesh particles:
-Cooper mesh for the cylinders is not available with spheres.
-T- grid mesh could be easily created
-Mesh volume of the sphere 30

Mesh the tube:
-T-grid works. For T-Grid 1-1 one highly skewed element appears on the sphere.

			Number	
Particles	Tube		of h.s	
mesh size	mesh size	Number of cells	elements	Worst element
1	1	873.163	1	0.989839
2	1	230.573	0	0.943603
3	1	158.112	0*	0.944889
4	1	149.439	8	0.993282
1	2	287.825	1	0.989839
2	2	131.139	0	0.943603
3	2	47.11	0	0.939851
4	2	36.864	7	0.993282
1	3	227.585	2	0.989839
2	3	52.505	0*	0.947047
3	3	31.557	0*	0.947047
4	3	20.222	6	0.993282
1	4	210.311	4	0.989839
2	4	45.533	0*	0.95674
3	4	21.766	0*	0.95674
4	4	14.791	5	0.993282

The two cylinders added, influenced the mesh of the tube, thus the mesh was created only for the sizes 2-1, 3-1, 2-2, 3-2, 2-3, 3-3, 2-4, 3-4.

* It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

2.3 Position3: touch between the two circular surfaces

In this position, the one cylinder was moved more than 5mm.

- For sphere 1,1:

Subtract:

- No virtual faces or edges while subtract sphere from the cylinder
- Virtual edges occurred while subtract from the tube even if Gambit2.uni was used for lower tolerance.

Solution: Unite volumes (and not retain). After uniting the two volumes no virtual faces or edges occurred.

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be easily created

-Mesh volume of the sphere 30

Mesh the tube:

-T-grid works .In order to mesh the volume in angle the above procedure was followed.

For T-Grid 1-1 one highly skewed element appears on the sphere.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	876.981	1	0.989839
2	1	225.259	0	0.943596
3	1	157.386	0*	0.967393
4	1	147.559	10	0.993282
1	2	311.882	1	0.989839
2	2	127.071	0	0.943596
3	2	47.354	0	0.939851
4	2	36.961	5	0.993282
1	3	254.85	2	0.989839
2	3	67.817	0*	0.947047
3	3	31.914	0*	0.947047
4	3	20.507	5	0.993282
1	4	238.495	23	0.989839
2	4	49.405	0*	0.95674
3	4	22.468	0*	0.95674
4	4	15.074	5	0.993282

The two cylinders added, influenced the mesh of the tube, thus the mesh was created only for the sizes 2-1, 3-1, 2-2, 3-2, 2-3, 3-3, 2-4, 3-4.

* It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

2.4 Position4: touch between the one vertical and one horizontal cylinder

- For sphere 1,1:

Subtract :

- No virtual faces or edges while subtract sphere from the cylinder
- No virtual edges occurred (Gambit2.uni).

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be easily created

-Mesh volume of the sphere 30

Mesh the tube:

-35 highly skewed elements occurred when the geometry was meshed with t-grid 1-1 and the 34 of them belonged to the touching surfaces of the vertical and the horizontal cylinder

- The surface should be decreased. The geometry will not be very realistic but a mesh can be created. A compromise between the real flow and the conductivity is hard to be achieved. A sphere between the two surfaces or the transport of one of the two in order to get in the other –subtract 1 and 2 can be the two ways of facing the problem.

<u>Solution 1:</u> The transport of one of the cylinders is chosen here

Particles	Tube		Number of h.s	
mesh size	mesh size	Number of cells	elements	Worst element
1	1	881.739	1	0.989839
2	1	227.499	0+	0.946725
3	1	158.392	0*	0.967393
4	1	147.887	10	0.993282
1	2	333.842	1+2	0.989839
2	2	128.264	0*	0.943603
3	2	47.958	0	0.934471
4	2	37.517	6	0.993282
1	3	279.725	1+5	0.989839
2	3	70.618	0*	0.947047
3	3	X	X	X
4	3	X	X	X
1	4	264.386	1+7+9	0.989839
2	4	53.361	0*	0.95674
3	4	X	X	X
4	4	X	X	X

The two cylinders added, influenced the mesh of the tube, thus the mesh was created only for the sizes 2-1, 3-1, 2-2, 3-2, 2-3, 2-4.

*It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.
+ It means that the worst point lays in the surface of the two volumes in contact.

Solution 2

One of the cylinders is moved 0,5mm and then a sphere of radius 1,1mm is created representing the contact between the cylinders.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	885.394	1	0.989839
2	1	226.486	0	0.943596
3	1	157.088	0*	0.967393
4	1	147.261	12	0.993282
1	2	341.54	1	0.989839
2	2	127.829	0	0.943596
3	2	47.894	0+	0.961042
4	2	37.485	6	0.993282
1	3	288.777	2	0.989839
2	3	71.774	0*	0.947047
3	3	32.561	0+	0.961042
4	3	21.112	5	0.993282
1	4	274.138	20	0.989839
2	4	53.922	0*	0.95674
3	4	23.482	0+	0.961042
4	4	15.7	5	0.993282

The two cylinders added, influenced the mesh of the tube, thus the mesh was created only for the sizes 2-1, 3-1, 2-2, 3-2, 2-3, 3-3, 2-4, 3-4.

With the second solution we managed to create a mesh for 3-3 and 3-4 also but we still have to decide which is a more realistic solution. We should also mention that adding this sphere (solution 2) the conductivity coefficient can be modified for these surfaces as they have a larger contact area. Additionally the case where the vertical cylinder ‘s surface is not completely in contact with that of the horizontal is included to the results of position 4.The results show that it s a slightly worse position.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	884.975	1	0.989839
2	1	224.688	0	0.943596
3	1	154.99	1+	0.970455
4	1	145.882	21	0.993282
1	2	340.402	1	0.989839
2	2	128.338	0	0.943596
3	2	47.494	1+	0.970455
4	2	37.034	5	0.993282
1	3	288.743	5	0.989839
2	3	71.541	0*	0.947047
3	3	32.616	1+	0.970455
4	3	21.013	5	0.993282
1	4	272.915	132*	0.997158
2	4	53.922	4*	0.981093
3	4	23.52	3(1+)	0.961042
4	4	15.848	5	0.993282

2.5 Position5: cylinders in parallel

- For sphere 1,1 :

Subtract :

- No virtual faces or edges while subtract sphere from the cylinder

- No virtual edges occurred while subtract from the tube even if Gambit2.uni was used for lower tolerance .

Mesh particles:

-Cooper mesh for the cylinders is not available with spheres.

-T- grid mesh could be easily created

-Mesh volume of the sphere 30

Mesh the tube:

Leaving the geometry as designed with contact points and subtracting the cylinders from the tube. We conclude that only 2-2,2-3,3-3,2-4,3-4 t-grid works as shown in the table.

Particles mesh size	Tube mesh size	Number of cells	Number of h.s elements	Worst element
1	1	918.8621	1	0.989839
2	1	231.713	18+	0.997252
3	1	162.629	31+	0.998612
4	1	152.623	63+	0.997776
1	2	360.908	7	0.995505
2	2	128.656	0*	0.943603
3	2	48.854	1+	0.984146
4	2	38.742	6	0.993282
1	3	305.46	10	0.989839
2	3	67.896	0*	0.947047
3	3	32.842	0*	0.96087
4	3	21.489	5	0.993282
1	4	294.253	40	0.989839
2	4	56.195	0+	0.965078
3	4	23.93	0*	0.96087
4	4	15.911	6	0.993282

The two cylinders added influenced the mesh of the tube so that mesh worked only for the sizes 2-2, 2-3,3-3, 2-4,3-4 .

*It means that the worst point lays not in the contact point but on the volume closest to the outlet surface.

+ It means that the worst point lays in the surface of the two volumes in contact.

The influence of the parallel clylinders can be decreased either by uniting them before subtracting from the tube, either by moving them and subtract each other before subtracting from the tube or by adding another geometry between them.

Choosing the most realistic solution of the above mentioned we obtained these results that are certainly better.

Solution: Uniting the two parallel vertical cylinders before subtracting from the tube

			Number of	
Particles	Tube	Number of	h.s	Worst
mesh size	mesh size	cells	elements	element
1	1	937.284	1	0.989839
2	1	226.679	0*	0.943596
3	1	155.415	0*	0.967393
4	1	145.81	13	0.993282
1	2	366.412	1	0.989839
2	2	128.724	0*	0.943596
3	2	48.197	0*	0.961042
4	2	37.711	6	0.993282
1	3	315.948	1	0.989839
2	3	74.475	0*	0.947047
3	3	32.977	0*	0.961042
4	3	21.441	5	0.993282
1	4	301.952	19	0.989839
2	4	57.576	0*	0.95674
3	4	24.3	0*	0.961042
4	4	16.121	5	0.993282

The two cylinders added influenced the mesh of the tube so that mesh worked only for the sizes 2-1, 3-1, 2-2, 3-2, 2-3, 3-3, 2-4, 4-4.

Conclusions:

- The sphere used had 1,1mm radius .This was thought to be the minimum radius that no mesh problems occur according to the periphery case study. If a larger sphere was used, better results would have been obtained and certainly fewer highly skewed elements. But as the size of sphere increases, the realistic representation of the packing fails, so the purpose is to study the worst possible case .

- At the last case study, 17 cylinders are designed and meshed. The number of elements does not change dramatically even if more cylinders are inserted.

- In the case study position 4, many highly skewed elements occurred at the contact with the wall. Size functions and individual study should be made in order to decide the appropriate model.

Summary:

- Contact points can be replaced by spheres when they are in the periphery of one cylinder.

- Cylinders that touch each other at the circular surfaces should be united before extracting from the tube.

- The most difficult position is the one of the vertical and horizontal cylinders. One of the two solutions proposed should be chosen

- The crucial decision is about the size of the sphere and also the particles' and tube's mesh size.