Adaptive Particle Resolution in Smoothed Particle Hydrodynamics

A Thesis

submitted to

Indian Institute of Science Education and Research Pune

in partial fulfillment of the requirements for the

BS-MS Dual Degree Programme

by

Sohan Sarangi

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA

March, 2016

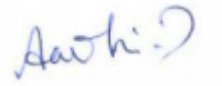
Supervisor: Aarthi Thyagarajan

This page was intentionally left blank

Certificate

This is to certify that this dissertation entitled "**Adaptive Particle Resolution in Smoothed Particle Hydrodynamics**" towards the partial fulfilment of the **BS-MS Dual Degree** programme at the **Indian Institute of Science Education and Research, Pune** represents original research carried out by **Sohan Sarangi**, Indian Institute of Science Education and research under the supervision of **Aarthi Thyagarajan**, Computational Scientist, **Shell Technology Centre, Bangalore**, during the academic year **2015-2016**.

Sohan Sarangi
Student



Aarthi Thyagarajan
Thesis Supervisor

This page was intentionally left blank

Declaration of Authorship

I hereby declare that the matter embodied in the thesis entitled " **Adaptive Particle Resolution in Smoothed Particle Hydrodynamics**" are the results of the investigations carried out by me at the **Computational Centre of Expertise, Shell Technology Centre, Bangalore**, under the supervision of **Aarthi Thyagarajan** and the same has not been submitted elsewhere for any other degree.

Sohan Sarangi

This page was intentionally left blank

Acknowledgement

I would like to thank my thesis supervisor Aarthi Thyagarajan for the continuous support during my thesis study and research and for her patience, motivation and immense knowledge about the subject. Also, I am thankful for the uncountable technical discussions we had. I have learnt a lot from her; not just about the project but also about the way of life. She has been a friend and a guide to me. I would also like to thank Bertwim van Beest for the innumerable code reviews and helping me to get my code to a proper shape. Those code reviews were useful, fun and I learnt a lot about software development from Bertwim. I would like to thank Chiranjib Sur for helping me through the initial stages of Object Oriented Programing (OOP) and being there for whenever I needed help. I am also thankful to Rutger IJzermans and Vinay R Gopala for the constant feedback during the internship and Sunil K Appanaboyina for being a one stop solution for most of the problems related to HPC.

I would also like to thank the entire CCoE team for being there when I needed them the most. My sincere thanks to Kunj Tandon, Vetrivel Rajappan and Vianney Koelman, for the exciting opportunity. All the simulations were run using the computing resources provided by Shell and it is worth to mention that thesis progress was leveraged by rapid adoption of the latest HPC technologies available in Shell.

My sincere thanks to Apratim Chatterji for agreeing to be a part of thesis advisory committee and for his support and valuable feedback.

I would like to thank my family and friends who have always been with me in my course of last 5 years. Their constant support and encouragement was the main reason behind my achievements.

This page was intentionally left blank

Abstract

In this thesis we outline the formulation, implementation and validation of Adaptive Particle Resolution (APR) in Smoothed Particle Hydrodynamics (SPH). Traditionally, SPH systems are modelled with unifrom mass distribution, through our approach we introduce a run time refinement and de-refinement algorithm for both 2D and 3D systems.

Each particle represents a mass of fluid in its local region. Particles are refined into several particles for finer sampling in regions of complex flow. In regions of smooth flow, neighboring particles are merged. This new development offers the capability of doing multi-resolution simulations in SPH framework. We have implemented various kinds of APR such as domain based, dynamic (based on the distance between solid and fluid particles) or multiphase flows in open channels and these implementations in the SPH method provide a simple yet powerful framework for applications like off-shore structures and reactor modelling In general, the current APR implementation can be applied for simulating multiscale models or systems that develop a wide spectrum of length scales. We have clearly shown the reduction in computational cost in the APR-SPH simulations without losing the accuracy.

All of our simulations were run in single machine and multiple cores (shared memory). A multi-machine approach is possible, but less efficient with the given memory architecture of the simulator. For further optimizations a review of the memory allocation strategy in the simulator is required.

This page was intentionally left blank

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Smoothed Particle Hydrodynamics (SPH) is a mesh-free Lagrangian computational fluid dynamics method for simulating fluid flows. This technique was initially developed for astrophysical problems [2]. Since the last couple of decades this field has been gaining importance in both academia and industry in the field of fluid dynamics [3]. In this method fluid is modelled as a collection of particles, each of which can be tracked and evolved using an equation of motion. Continuum properties like density and pressure are interpolated from its neighbouring particles by using a smoothing kernel. In contrast to traditional, grid-based, computational fluid dynamics methods (fig. 1.1), SPH is based purely on the solution of the equations of motion of "fluid particles", i.e. pockets of fluid that could be seen as a moving set of discrete points in space at which the continuum equations of fluid mechanics may be solved.

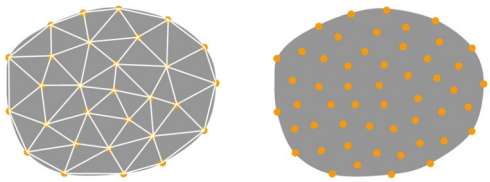


Figure 1.1: The figure on the right shows example of grid based method where the grids/meshes formed by joining the nodes where all the field variables are defined. On the right we see the SPH formalism which a mesh-free method, properties of the meshes are carried by nodes/particles.

This first chapter is devoted to introducing the reader to the topic of the thesis and problem being addressed. Following which, we lay the basic foundations of fluid mechanics and the governing SPH equations.In the next chapter we explain the methods developed for the problem and other numerical corrections which is needed. We validate and discuss various type of cases with our implantation, the cases involved cover different aspects of physical component under study.

1.1 Literature Survey

Most of the SPH simulations are done using particles of uniform size throughout. This leads to a huge compute cost when using smaller particle sizes or a decrease in accuracy of simulation with larger particle size. In order to strike a balance between reduced compute cost and better accuracy of simulations, Adaptive Particle Resolution (APR) is necessary. It is beneficial to vary particle sizes by means of particle refinement in the region where better resolution is needed and coalesce it elsewhere, at the same time assuring that conservation laws are followed and the predicted results are physically consistent.

Adaptivity is a well studied concept in classical Eulerian computational models, computationally efficient models have been developed for mesh based systems to deal with multiphase flows. Adaptivity in SPH is comparatively new concept and very few attempts have been made. Earlier techniques in mesh-free method involved re-meshing and particle insertion and removal techniques [4] [5] [6]. Recently Feldman et. al. [7] and Vacondio et. al. [8] [9] [10] recently showed a runtime particle refinement and de-refinement. It is variationally consistent and conserves momentum. A similar approach was proposed by López et. al. [11] in which daughter particles are given uniform mass distribution as opposed to non-uniform distribution in other approaches. A recent approach was proposed by Barcarolo [12] in the the region of refinement and de-refinement respectively.

1.2 Problem Statement

Adaptivity or Adaptive Particle Resolution (APR) is one of the most active areas of research in SPH Community. It is declared as one of the Grand Challenges by SPHERIC (SPH European Research Interest Community) [1]. As outlined in the abstract, the aim of this thesis is to formulate, implement and validate adaptive particle resolution in Smoothed Particle Hydrodynamics with an aim of reducing the computational cost. We employ a run-time particle refinement and de-refinement algorithm.

We implement the approach proposed by López et al. for 2D systems [11] and also formulate and implement a new approach for 3D systems as opposed to recent 3D refinement optimisations proposed by [13]. We also extend the implementation to deal with ordered, disordered, multiphase systems and heat transfer. We show a huge decrease in the compute cost of the simulation when simulations are performed in single node. With proper load balancing across multiple nodes, the speed up can further be improved in future.

[1] `https://wiki.manchester.ac.uk/spheric/index.php/SPH_Numerical_Development_Working_Group` (Accessed on 23rd March, 2016)

We validate our implementation by testing it for number of test cases. The SPH simulation results are compared with analytical results and also against experimental results for some. We try to validate every physical concept implemented , such as closed and ordered (eg. Poiseuille flow), free surface flows, disordered flow through regular porous medium, heat conduction and multiphase periodic systems. Thus we lay the ground for using adaptivity for packed bed reactor modelling with complex shapes of catalysts with multiphase flow and heat transfer. Together with other features , APR can be applied for engineering problems. Example, by using APR and extreme wave models in SPH framework, one can predict the wave effects on offshore structures at a better accuracy. Similarly, changing and moving interphases are difficult to describe with classical grid-based methods. APR together with Multiphase modelling capability in SPH can be applied for Chemical Process Engineering. With multiphase and APR, the multi-phase flow in realistic/actual porous media can be done at reduced compute cost.

This page was intentionally left blank

Chapter 2

Theory

In this chapter we focus on the equations of motion used in SPH for APR. The form of SPH we use in our approach is called weakly compressible SPH (WCSPH). For brevity we explain in detail only the implementations of WCSPH and not Incompressible SPH. In this chapter we start with the basic fluid mechanics equations and on to explain their approximations using a smoothing kernel in SPH counter part. We focus on three fundamental laws.

1. Conservation of Mass
2. Conservation of Momentum
3. Conservation of Energy

2.1 Equations of Fluid Mechanics

Most accurate description of fluid is given by Boltzmann Equation, which can be used to track the trajectory of each particle i.e. molecule but for large systems it is impractical to solve for millions and billions of particles. Navier Stokes equation is derived from the continuum of Boltzmann Equation with the assumption that, Newton's Second Law is valid for point particles if the fluid is treated as a continuum. We start with the continuous description of fluid and then on to discretise it.

The basic equations for conservation of mass, momentum and energy in fluid mechanics for an incompressible, non-relativistic Newtonian fluid is given by,

$$\frac{d\rho}{dt} = -\rho \nabla . v \tag{2.1}$$

$$\rho \frac{dv}{dt} = -\nabla p + \mu \nabla^2 v + \frac{1}{3} \nabla (\nabla . v) + f \tag{2.2}$$

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\rho v^2 + \rho\hat{U}\right) + \nabla.\left[\left(\frac{1}{2}\rho v^2 + \rho\hat{U}\right)v\right] = -\nabla.q - \nabla.(pv) - \nabla.(\tau.v) + \rho(g.v) \quad (2.3)$$

where $\frac{d}{dt}$ represents the total derivative, ρ represents density, $v$velocity, p pressure, μ dynamic viscosity, f external force, assuming Fourier series $-\nabla.q = -\nabla.(k\nabla T)$, where k denotes thermal conductivity,. τ and τ denotes viscous shear stress tensor, the external force is usually the gravity in the system.

2.2 SPH Approximations

SPH equations are defined to solve hydrodynamic equations computationally. The defining feature of SPH method is that the continuum properties are calculated based on interpolation over its neighbours. Interpolation in SPH is based on two basic approximations.

Consider an arbitrary scalar field A and the Dirac Delta Function (distribution) δ. The following identity holds,

$$A(\mathbf{x}, t) = \int_\Omega A(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')dx' \quad (2.4)$$

where Ω is the entire physical space under consideration. In SPH we replace the dirac delta function with a kernel whose approximation is a Dirac-Delta distribution. Let us denote the kernel function by $|W(\mathbf{x}-\mathbf{x}'|)$. Then Eq. 2.4 can be approximated as follows:

$$A(\mathbf{x}, t) = \int_\Omega A(\mathbf{x}', t)\delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \approx \int_\Omega A(\mathbf{x}', t)W(|\mathbf{x} - \mathbf{x}')dx' \quad (2.5)$$

In order for the approximation to be true to first order the following conditions need to fulfilled:

$$\int_\Omega (|\mathbf{x} - \mathbf{x}'|)d\mathbf{x}' = 1 \quad (2.6)$$

$$\int_\Omega W(|\mathbf{x} - \mathbf{x}'|)\mathbf{x}d\mathbf{x}' = 0 \quad (2.7)$$

The above can be seen by Taylor expanding the scalar function $A(\mathbf{x}', t)$ around $(\mathbf{x}, t)$. this shows the first order interpolation. This can further be extended to higher order kernel approximations, for minimising errors. But second order kernel approximations may lead to negative density. So generally only first order approximation is used.

$$\int_\Omega W(|\mathbf{x} - \mathbf{x}'|)\mathbf{x}^2 d\mathbf{x}' = 0 \quad (2.8)$$

Second approximation in which SPH relies on the fact that we can approximate the integral as a finite sum.

$$\int_\Omega A(\mathbf{x}', t)W|(\mathbf{x} - \mathbf{x}')|dx' \approx \sum_j A(\mathbf{x}_j, t)W(|\mathbf{x} - \mathbf{x}_j|)V_j \quad (2.9)$$

where V_j is the volume of the j_th particle. Using $m_j = \rho_i.V_j$. We get,

$$A(x_i, t) \approx \sum_j \frac{m_j}{\rho_j} A(x_j, t) W(|\mathbf{x} - \mathbf{x}_j|) \tag{2.10}$$

and gradient of scalar field in terms of kernel approximation is given by,

$$\nabla A(x_i, t) \approx \sum_j \frac{m_j}{\rho_j} A(x_j, t) \nabla W(|\mathbf{x} - \mathbf{x}_j|) \tag{2.11}$$

2.3 Kernels

A Kernel is a function which replaces the dirac delta distribution. In SPH properties continuum properties are calculated by interpolating over the neighbours defined by the kernel. A number of kernels have been defined in SPH framework, such as Wedland, Gaussian, Cubic, Quintic etc... They define a compact support i.e. upto which the neighbours would be considered. There is a basic underlying assumption that every kernel is continuous over the domain, there by defining the derivative of the kernel in a consistent manner.

$$W(q) = \begin{cases} 1 - 6q^2 + 6q^3, & 0 \leq q \leq \frac{1}{2} \\ 2(1-q)^2, & \frac{1}{2} \leq q \leq 1 \\ 0, & q > 1 \end{cases} \tag{2.12}$$

where $q := \frac{|x|}{h}$, h represents smoothing length.

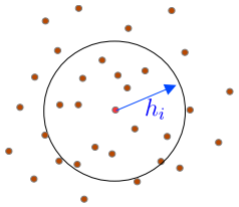


Figure 2.1: Figure shows kernel interpolation with smoothing length acting as the cut-off radius.

2.4 Application of SPH to the Governing Equations

We represent the basic equations in SPH formalism using kernel interpolation. As mentioned earlier, throughout the thesis we consider only cubic kernel. In this section we outline the basic conservation laws like mass, momentum and energy in SPH formalism.

2.4.1 Density Equation

There are two alternative forms of density calculation commonly used in SPH. One is summation density and the other is continuity density.

Summation density is based on kernel interpolation.Using the SPH interpolation formula 2.10 and replacing scalar field by ρ_i of the i^{th} particle we get,

$$\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W(|\mathbf{x}_i - \mathbf{x}_j|) = \sum m_j W_{ij} \tag{2.13}$$

Summation density is more expensive as it introduces another loop for calculating density value. In the thesis we stick to continuity density. We employ the most common formulation of density proposed by Liu:

$$\frac{d\rho_i}{dt} = \sum_{j=1}^{N} m_b (v_i - v_j).\nabla_i W(|\mathbf{x}_i - \mathbf{x}_j|, h) = \sum_{j=1}^{N} m_j v_{ij}.\nabla W_{ij} \tag{2.14}$$

In above eq. 2.14 the shorthand used are $v_{ij} = v_i - v_j$ and $W_{ij} = W(x_i - x_j, h)$. Using eq. 2.14 density equation can be solved along with the equation of motion of each particle there by resulting in a reduction of compute cost.

2.4.2 Momentum Equation

Every fluid dynamic simulation depends upon the accurate solution of Navier-Stokes Equation. The numerical discretisation of pressure gradient plays an important role. In order to get to an expression which conserves physical properties such as momentum and energy, a Lagrangian is used for derivation. A final expression for resulting equation of motion is obtained by using Euler-Lagrange equation, a variational approach.

An equation describing the symmetrised expressions was described by Adami et. al. [14]. We simplify it by using $\nu = \mu/\rho$ where ν is the kinematic viscosity and μ is the dynamics viscosity. We get,

$$\frac{d\mathbf{v}_i}{dt} = -\frac{V_i^2 + V_j^2}{m_i} \sum_j \left[\frac{\rho_j p_i + \rho_i p_j}{\rho_i + \rho_j} + \frac{2\nu_i \rho_i \rho_j}{\rho_i + \rho_j} \right] \nabla W_{ij} - \alpha \sum_j m_j hc \frac{\mathbf{v}_{ij}\mathbf{x}_{ij}}{\rho \mathbf{x}_{ij}^2 + 0.01h^2} \tag{2.15}$$

The terms on the right hand side include the pressure term, shear forces and viscosity term.

2.4.3 Pressure Equation

Pressure equation plays a crucial role in solving Navier-Stokes Equation. The pressure gradient term can be represented as,

$$\left(\frac{\nabla P}{\rho} \right)_i = \nabla \left(\frac{P}{\rho} \right)_i + \frac{P}{\rho_i^2} \nabla \rho_a \tag{2.16}$$

Using Eq. 2.10 and 2.11 we get,

$$\left(\frac{\nabla P}{\rho} \right)_i = \sum_{j=1}^{N} \left(m_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij} + m_j \frac{P_i}{\rho_i^2} \nabla_i W_{ij} \right) \tag{2.17}$$

Simplifying the above equation we get [15],

$$\left(\frac{\nabla P}{\rho}\right)_i = \sum_{j=1}^{N} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) \nabla_i W_{ij} \tag{2.18}$$

We use the above pressure equation for all the simulations in this thesis. The equation intuitively may feel like there is some inconsistency as one would have expected a difference n pressure gradient rather than sum, but as mentioned shown by Price et. al. [15] we see that it rather helps in uniform particle distribution.

2.4.4 Time Stepping Criteria

Time stepping is an important criteria for maintaining the stability of a simulation. We use an explicit time stepping scheme to solve the weakly compressible implementation of SPH, in which an equation of state is used to relate the pressure to the fluid density. Hence, the time step $\triangle$t needs to be kept below a certain maximum in order to guarantee a numerically stable simulation [16] [17] [18]. An often-used stability criterion is based on the mean particle spacing h and the relevant velocity scales in the fluid. It is important to note that while simulating a system with variable resolution it is important to choose the time step length corresponding to minimum particle size.

$$\triangle t \leq 0.125 \frac{h^2}{\nu} \tag{2.19a}$$

$$\triangle t \leq 0.25 \frac{h}{3c} \tag{2.19b}$$

$$\triangle t \leq 0.25 \min_i (h/3|\mathbf{F}_i)^{1/2} \tag{2.19c}$$

where $|\mathbf{F}_i|$ is the magnitude of force on each particle, h is smoothing length and c speed of sound.

2.4.5 Time Integration

We use the traditional Leapfrog numerical integration [19] for integrating the differential rate equations.

2.4.6 Viscosity Models

Based on the system at hand we employ two different viscosity models in our simulations. Commonly artificial viscosity model is used for free surface flows of high Reynold's number and a laminar viscosity model is used for laminar regimes.

Laminar Viscosity

Various viscosity models exists in SPH literature which implements a combination of kernel gradient approximation and finite difference approximations [15]. Based on local

compression and expansion of particle field [20],

$$(\rho^{-1}\nabla.(\mu\nabla\mathbf{v})) = -\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i \tag{2.20}$$

where Π_{ij} is defined as,

$$\Pi_{ij} = -\frac{8\mu(\mathbf{v}_{ij}.\mathbf{x}_{ij})}{\rho_i \rho_j x_{ij}^2} \tag{2.21}$$

where μ represents dynamic viscosity.

Artificial Viscosity

This is an alternative approach compared to the physical viscosity model which adds a dissipation to the simulation. This is commonly known as artificial viscosity and was developed by Monaghan [21] for a stable numerical simulations of shocks. The term consists of a linear term directly proportional to a dimensionless parameter α and a quadratic term proportional to a dimensionless parameter β and D_{ij} is distance between i and j particle.

$$f_{\text{i,artificial}} = -\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i W_{ij} \tag{2.22}$$

where Π_{ij} is,

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{c}_{ij} D_{ij} + \beta D_{ij}^2}{\bar{\rho}_{ij}}, & \mathbf{v}_{ij}.\mathbf{x}_{ij} < 0 \\ 0, & \mathbf{v}_{ij}.\mathbf{x}_{ij} \geq 0 \end{cases} \tag{2.23}$$

2.4.7 Speed of Sound

The value of sound should ideally be chosen equal to the physical speed of sound, but is often convenient to choose c (Speed of Sound) in a numerical simulation much lower than physical value. This allows for larger time steps to be used in the numerical simulation. Usually speed of sound is chosen to be 10 times the maximum velocity the system can possibly attain stability during the simulation.

2.4.8 Boundary Condition

In this section, we discuss the boundary conditions used in the SPH simulations. It plays a very important role for any type of simulations as it governs how the fluid particles interact with the boundary particle. Conceptually, boundary particles form a rigid wall whose equations of motion is zero. It poses two conditions on the fluid,

- The boundary wall is rigid and should be impermeable to fluid, thus the normal wall velocity component should be zero.
- If viscosity is taken into account, the fluid velocity at the wall should be equal to wall velocity. This is taken care by the no-slip boundary condition.

It is also important to note that the wall density shouldn't fall below solid density, else the solid would tend to attract the fluid there by resulting in fluid passing through the solid particles. We force the solid boundary density to the initial density if it falls below a certain threshold. Generally, in SPH solid walls are also simulated by particles, which balance the pressure of inner fluid particles and prevent them from penetrating through the wall. The wall boundary condition is modelled as,

Fixed fluid particles at the wall: In this method, we attach fixed fluid particles at the solid boundaries, that exert a repulsive force on the inner fluid particles through a force function [22] and was employed by [23] where the boundary particles repel away the fluid particle with a normal force similar to the Leonard-Jones formulation. The contribution of a near fluid particle i from a wall particle w is,

$$F_{iw} = \begin{cases} 0, & |\mathbf{x}_{iw}| > B_2 h \\ \left(\frac{1}{2}\frac{B_3+2}{B_2}\right)^2 \left(\frac{|\mathbf{x}_{iw}|}{h-B_2}\right)^2, & B_1 h < |\mathbf{x}_{iw}| \leq B_2 h \\ \left(\frac{B_1 h}{|\mathbf{x}_{iw}|}\right)^{B_3}, & |\mathbf{x}_{iw}| \leq B_1 h \end{cases} \tag{2.24}$$

where $B_1 = \frac{B_1 B_2}{B_3+2}, B_2 = 1$ and $B_3 = 8$. The shear force at the boundary is calculated using the viscous term As the number of particles decrease near the boundary, the accuracy of numerical simulation degrades for the particle near the wall [24]. In our approach of adaptive particle resolution, the decreasing number of particle leads to numerically incorrect results. To deal with such a scenario, we employ higher order corrections in the form of corrective smoothed particle hydrodynamics the details of which are mentioned in the next chapter.

2.4.9 Neighbour Search

In order to calculate the density of each particle one needs to determine which particles are neighbours. The most naive way of doing it is looping over all the particles in order to determine the distance from the particle of interest. It is very expensive and involves a double loop thus resulting in a complexity of $O(N^2)$. There are two ways to reduce the number of neighbours, one is to create a neighbour list (which stores particles which fall within cut-off) and update it based on the system at hand (for turbulent flows the updating step would be very small and other way for laminar flows). The other approach involves dividing the entire computational domain into grids, the length of the grid should be greater than the cut-off radius of each particle. We collect all the neighbouring grids in this case which would be 8 for 2D and 26 for 3D. The nearest neighbour search we use involves a combination of both, collecting the grids and finding the nearest neighbours in the collected grids. This method is known as spatial hashing method and is bound by $O(NlogN)$

This page was intentionally left blank

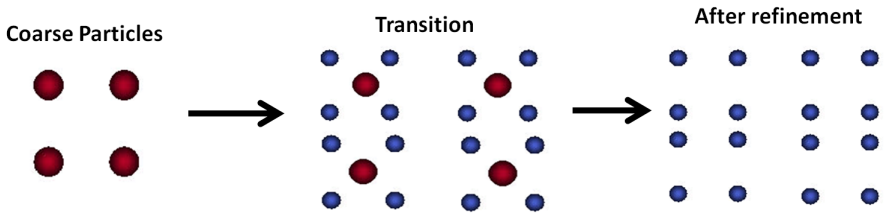
Chapter 3

Methods

In this chapter, we present methods specific to Adaptive Particle Resolution as applied to Smoothed Particle Hydrodynamics. We first make an in-depth analysis of the schema for adaptivity proposed by López et. al. [11] We extend a similar approach to 3D systems which has not been studied in literature before. During the time of this Master Thesis a 3D extension of adaptivity was published by Vacondio et. al. [13] but the approach is conceptually different from ours. Based on the above implementation we try to formulate algorithms to validate generic cases of ordered, disordered, inflow-outflow and multiphase systems. Adaptivity in SPH has mostly been studied only for highly ordered systems. The novelty in our implementation is that it is suited and robust enough to handle multiphase, and disordered flows like flow through porous medium. In this section, we also address the variable resolution inputs and other challenges faced during the implementation of APR in the current SPH framework.

3.1 Basics

SPH simulations are generally based on uniform distribution of particles of equal mass. So, a decrease in particle size and increase in size of domain leads to huge computational cost and vice versa. Implementation of a refinement and de-refinement algorithm enables us to variably increase particle resolution in regions of interest. This enables a simulation to have sparsely and densely populated regions in the same domain. Our aim is to achieve the reduction in compute cost and at the same time conserve the physical laws.



3.1.1 Particle Refinement Algorithm

A general refinement algorithm was proposed by Feldman. et. al. [7] According to which,

- Global properties like mass, kinetic energy, linear momentum and angular momentum should be conserved during the refinement process, as shown in Table3.1.
- Density errors arising due to change in local mass distribution which leads to a change in density fields should be minimised by the refinement process.
- The proposed method should be able to handle transition regimes where fine and coarse particles interact.

It was shown by Feldman et. al. [7], in order for the global properties to be conserved the refined particles should take the same velocity as that of coarse particles. This idea was further built up by Vacondio [10] and Lopez [11] . We first implement and validate the approach of constant mass distribution of among refined particles proposed by López for 2D and extend the minimisation criteria to 3D, multiphase and heat transfer problems enabling us to set a framework ready for simulating and predicting realistic cases. As shown in table 3.1 it is of utmost importance that basic physical conservation laws like mass, energy, linear and angular momentum are conserved.

Table 3.1: Conservation Properties

Global Properties	**Before Refinement**	**After Refinement**
Mass	m_a	$\sum_{a_i} m_{a_i}$
Kinetic Energy	$\frac{1}{2} m_a \mathbf{v}_a . \mathbf{v}_a$	$\frac{1}{2} \sum_{a_i} m_{a_i} \mathbf{v}_{a_i} . \mathbf{v}_{a_i}$
Linear Momentum	$m_a \mathbf{v}_a$	$\sum_{a_i} m_{a_i} \mathbf{v}_{a_i}$
Angular Momentum	$\mathbf{x}_a \times m_a \mathbf{v}_a$	$\sum_{a_i} \mathbf{x}_{a_i} \times m_{a_i} \mathbf{v}_{a_i}$

3.1.2 Refinement Variables

In order to refine particles the physical entity carried by each particle needs to be taken care of. We assign following parameters for refined particles, based on a set of refinement variables which needs to be optimised for minimising error occurred during refinement and de-refinement. In the Fig. 3.1 the following parameters are defined,

- Separation Parameter ($\epsilon \in [0, 1]$) - It is a scalar parameter which decides the distance between two daughter particles and hence the spread of daughters around the mother particle under consideration.

  $$\triangle x_d = \epsilon \triangle x \tag{3.1}$$

  where $\triangle x$ is the separation between two coarse particles and $\triangle x_d$ is the separation between two daughter particles.

- Smoothing Ratio ($\alpha \in [0, 1]$) - Instead of recalculating the smoothing length based on separation between two daughters, we rescale smoothing length of the daughter, so as to minimise refinement error. Smoothing ratio is a scalar parameter which defines the smoothing length of daughters based on that of mother.

$$h_d = \alpha h_m \tag{3.2}$$

  where h_m is the smoothing length of coarse particle and h_d is the scaled smoothing length.

- Mass Ratio ($\lambda_d \in [0, 1]$) - It is a scalar factor which is used to redefine mass of each of the refined particle.

$$m_d = \lambda_d m_m \tag{3.3}$$

  where m_m is the mass of the coarse (mother) particle and m_d is the scaled mass. In order for mass conservation to hold,

$$\sum_{d=1}^{N} \lambda_d = 1 \tag{3.4}$$

  where N is the number of refinements

- Time Stepping - Time stepping criteria for the entire simulation should be calculated based on the particle separation of the most refined particle size which is shown in the previous chapter in the section 2.4.4

Other field variables are either copied or interpolated from its neighbours depending upon their variation with change in mass.

3.1.3 Refinement Error

Particle refinement modifies the local properties of a system leading to an error in current state of the system. It is important to minimise the error in order to increase the accuracy of simulation. Many approaches are reported in the literature in order to minimise the density refinement error. In our approach we minimise the gradient of kernel rather than the kernel itself, as gradient is used in the derivation of most of the SPH equations.

Consider a scenario where particle p is refined into N daughter particles, this changes the approximation of the gradient of the function,

$$\langle \nabla f(\mathbf{x}) \rangle^* = \langle \nabla f(\mathbf{x}) \rangle - \frac{m_p}{\rho_p} f(\mathbf{x}_p) \nabla W_p(\mathbf{x}) + \sum_{d=1}^{N} \frac{m_d}{\rho_d} f(\mathbf{x}_d) \nabla W_d(\mathbf{x}) \tag{3.5}$$

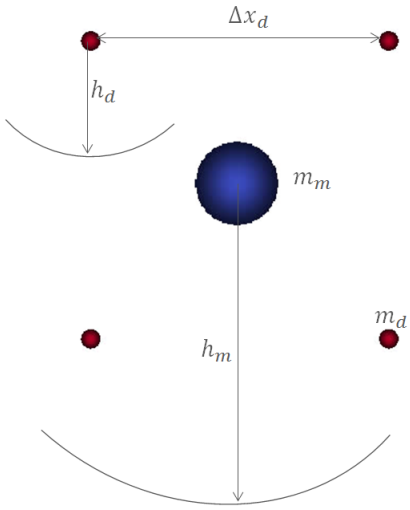


Figure 3.1: Refinement parameters (Red - Daughter (Refined Particles), Blue - Mother (Coarse Particles))

Local error produced at some arbitrary point $\mathbf{x}$ due to refinement of particle p cab be defined as,

$$e_p^2(\mathbf{x}) = \left(\left\langle \frac{\partial f(\mathbf{x})}{\partial x^\alpha} \right\rangle - \left\langle \frac{\partial f(\mathbf{x})}{\partial x^\alpha} \right\rangle^* \right)^2 \tag{3.6}$$

$$= m_p^2 \left(\frac{f(\mathbf{x}_p)}{\rho_p} \frac{\partial W_p(\mathbf{x})}{\partial x^\alpha} - \sum_{d=1}^{N} \lambda_d \frac{f(\mathbf{x}_d)}{\rho_d} \frac{\partial W_d(\mathbf{x})}{\partial x^\alpha} \right)^2 \tag{3.7}$$

where $m_d = \lambda_d m_p$ and $\sum_{d=1}^{N} \lambda_d = 1$.
Global error is given by,

$$E_p = \int_\Omega e_p(\mathbf{x}) d\mathbf{x} \tag{3.8}$$

The generalised approach presented is applies to continuity density approach,

$$e_p^\rho(\mathbf{x}) = \left\langle \frac{D\rho(\mathbf{x})}{Dt} \right\rangle - \left\langle \frac{D\rho(\mathbf{x})}{Dt} \right\rangle^* \tag{3.9}$$

$$= m_p(\mathbf{v}(\mathbf{x}) - \mathbf{v}_p)) \left(\nabla W_p(\mathbf{x}) - \sum_{d=1}^{N} \lambda_d \nabla W_d(\mathbf{x}) \right) \tag{3.10}$$

Global refinement error can be obtained using eq. 3.8. With the assumption that change in the velocity difference and mass over time is nearly constant we calculate the error by minimising,

$$E_p^{\nabla W} = \int_\Omega \left(\frac{\partial W_p(\mathbf{x})}{\partial x^\alpha} - \sum_{d=1}^{N} \lambda_d \frac{\partial W_d(\mathbf{x})}{\partial x^\alpha} \right) d\mathbf{x} \tag{3.11}$$

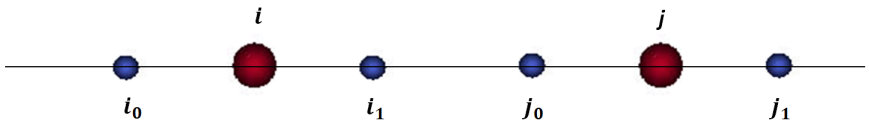


Figure 3.2: Two 1D refinement of particles. Each coarse particle is refined into two refined particles

In Eq. 3.11 we know that $E_p^{\nabla W}$ depends on α, ϵ and λ_d which need to be optimised so as to reduce the refinement error.

3.1.4 Parameter Optimisation

For 2D situations, we refine the particles in the form of square lattice to smaller particles of equal mass distribution and for 3D we refine in the form of a body centred cubic lattice with the unrefined particle at the centre (which is deleted after refinement) with equal mass distribution among refined particles. This optimisation is done specific to cubic kernel. The optimised value remains the same irrespective of the system under consideration till the kernel is unchanged.

Based on the approach by Lopez et. al. [11] we show that, In Fig. 3.2 each mother is refined into two daughters, and numerical instabilities are analysed. Based on the notation from section on refinement parameters and $h = r_a \triangle x$ where $r_a = 1.2$, the following were calculated by Lopez et. al.

Condition 1:

$$\begin{aligned} \overline{i_0 i_1} &< \frac{2}{3} h_{i_0 i_1} \\ \epsilon \triangle x &< \frac{2}{3} \alpha r_a \triangle x \\ 0 &< \frac{2}{3} \alpha r_a - \epsilon \end{aligned} \tag{3.12}$$

Condition 2:

$$\begin{aligned} \overline{i_i j_0} &< \frac{2}{3} h_{i_1 j_0} \\ \overline{ij} - \epsilon \triangle &< \frac{2}{3} \alpha r_a \triangle x \\ \frac{\overline{ij}}{\triangle x} &< \frac{2}{3} \alpha r_a + \epsilon \end{aligned} \tag{3.13}$$

Condition 3:

$$\begin{aligned} \overline{i_1 j} &< \frac{2}{3} h_{i_1 j} \\ \overline{ij} - \frac{\epsilon}{2}\triangle x &< \frac{(1+\alpha) r_a \triangle x}{3} \\ \frac{\overline{ij}}{\triangle x} &< \frac{(1+\alpha) r_a}{3} + \frac{\epsilon}{2} \end{aligned} \tag{3.14}$$

Based on the above condition and minimising Eq. 3.11 we optimise the value for α and ϵ for both 2D and 3D. For a cubic spline kernel we find that keeping the value of $\alpha = \epsilon = 0.5$, $\lambda_d = 0.25$ for 2D and $\lambda_d = 0.125$ for 3D minimises refinement error at the same time taking care of numerical instability.

3.2 Refinement Types

We have defined, formulated and implemented the refinement criteria based on three broad categories,

1. Domain Based
2. Dynamic Refinement
3. Interfacial Refinement

The APR scheme can be done in two ways: 1)Refining and coalescing based on the zones/domain and 2) Refining and coalescing based on distance from the fixed solid particles. The former approach works well for efficient modelling of flow through hollow-pipes, and refining particles only close to the solid structures in Off-shore structure simulations. The latter approach is valuable for situations like pore-scale flow where the solids are packed randomly and hence the boundary layers are also present randomly. We would detail each of the refinement types formulated by us in this thesis in the sections below and a detailed validation by comparison with analytic and experimental result is presented in the next chapter.

3.2.1 Domain Based

In this approach the domain of refinement and de-refinement is explicitly stated by the user. This approach comes in handy when we apriori know the region of refinement and de-refinement. (Fig. 3.5).Based on the geometry of problem, domain based refinement is implemented in two types of geometry,

- **Block Domain** - This type of refinement takes care of well ordered regular geometry having well defined vertical and horizontal bounding walls. This can be further classified into wall bounded domain (when the refinement domain is bounded by

wall on one side eg. 2D Poiseuille flow as shown in 3.3) and fluid bounded domain (where the refinement domain is bounded by fluid on all sides eg. flow past a cylinder).

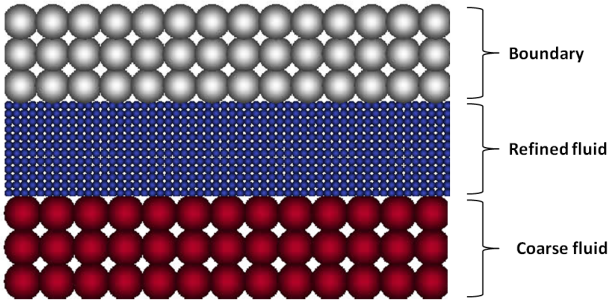


Figure 3.3: Poiseuille's flow in 2D with refinement along boudary

- **Circular Domain** - This type of refinement is used when the boundary is circular or spherical in shape. e.g. Flow in a pipe. (Fig. 3.4). Similar to the block domain case, the circular domain case can again be classified as wall bounded domain and fluid bounded domain.

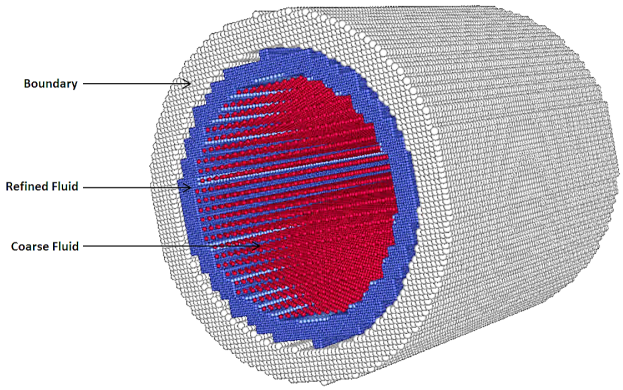


Figure 3.4: Poiseuille Flow in 3D in a pipe with refinement along the boundary

3.2.2 Dynamic Domain

In this approach the domain is defined based on pre-defined distance, which is an user input and it denotes the distance between the boundary and fluid. This enables us to identify refinement domains and de-refinement takes place outside these refinement domains. It is a versatile method and can be used in all the cases where static refinement

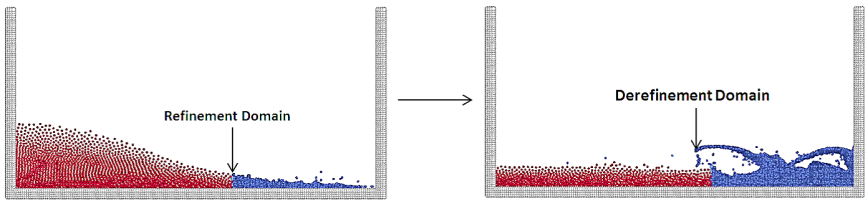


Figure 3.5: 2D Dambreak model with refinement and de-refinement regions.

is implemented with the added advantage that it can handle random particle distribution. One of the application where static refinement fails is porous medium flow, where the boundary particles are randomly distributed. Fig. 3.6 illustrates refinement and de-refinement regions for this algorithm.

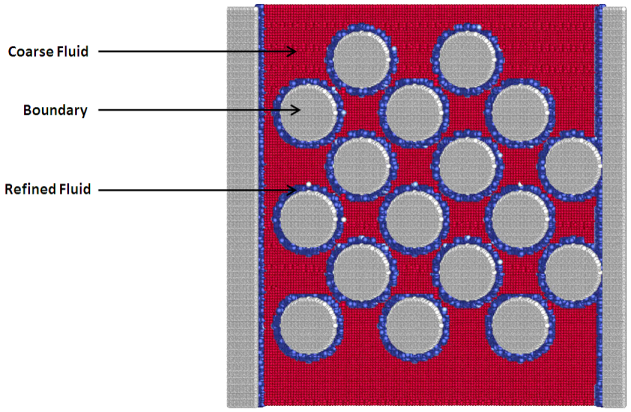


Figure 3.6: Porous medium 2D showing implementation of dynamic domain implementation

3.2.3 Interfacial

In this approach refinement takes place at the interface of two phases. This type of refinement is useful in case of complex multiphase flows like bubble in water. Currently it works only with periodic boundary conditions, further work needs to be done to make it compatible with closed system (Fig. 3.7).

3.3 De-refinement

In order to get the practical advantage of adaptivity in SPH we need to de-refine the system in regions where we don't need higher accuracy. Coalescence of two particles is based on the barycentric definition of different quantities, so as to satisfy conservation laws. To maintain stability, only two neighbouring particles can be merged at a time. If

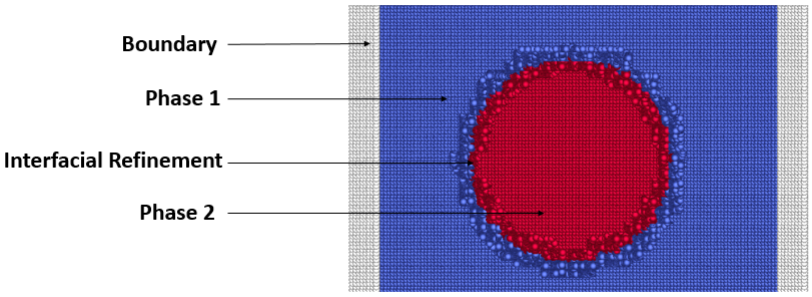


Figure 3.7: Figure shows bubble in water with interfacial refinement. Red - Bubble Phase, Blue-Water Phase and Gray - Solid Boundary

the two particles which are to be merged are named a and b respectively and the new particle formed is named m. Then,

- **Mass**
$$m_m = m_a + m_b \tag{3.15}$$
- **Position**
$$x_m = \frac{m_a x_a + m_b x_b}{m_m} \tag{3.16}$$
- **Velocity**
$$v_m = \frac{m_a v_a + m_b v_b}{m_m} \tag{3.17}$$
- **Smoothing Length**
We define the inverse of smoothing ratio (β) and calculate smoothing length of particle based on:
$$\beta = \frac{1}{\alpha} \tag{3.18}$$
$$h_m = \frac{m_m}{\text{mass of coarse}} \times \beta \times \text{smoothing length of coarse} \tag{3.19}$$

As mentioned in the previous sections, de-refinement takes place in the region outside the region of refinement for both static and dynamic cases. This ensures the considerable computational advantage yet maintaining higher accuracy. Other dynamic properties which have compounded effects are taken as average of the two particles else we stick with the barycentric rule.

During de-refinement we ensure that nearest or next nearest neighbour undergoes coalescence. This is necessary as it prevents merging two particles far away and random placement of the particle in the domain, leading to instability. We ensure that when two particles are merged it doesn't exceed the starting mass of the coarse particle.

3.4 Implementation Overview

Adaptive particle resolution was implemented in the in-house code of Shell Technology Centre, Bangalore [25]. APR acts as a plug-in for the main code and it is decoupled from the API (Application programming interface) by using the concept of class abstraction. It is generic enough to handle various types of adaptivity in a very efficient way and can also be extended further types with ease. The figure below shows a class hierarchy for the implementation, Different approaches of adaptivity mentioned in the previous

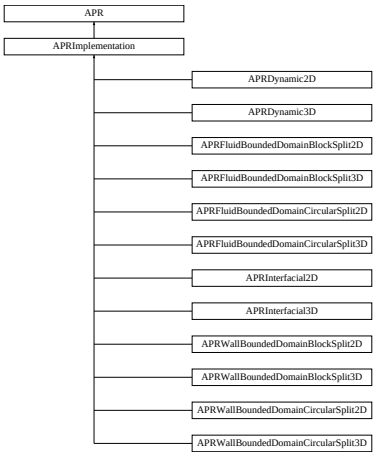


Figure 3.8: Figure shows a class hierarchy for APR implementation.

section are shown in schematics as shown in fig. 3.9

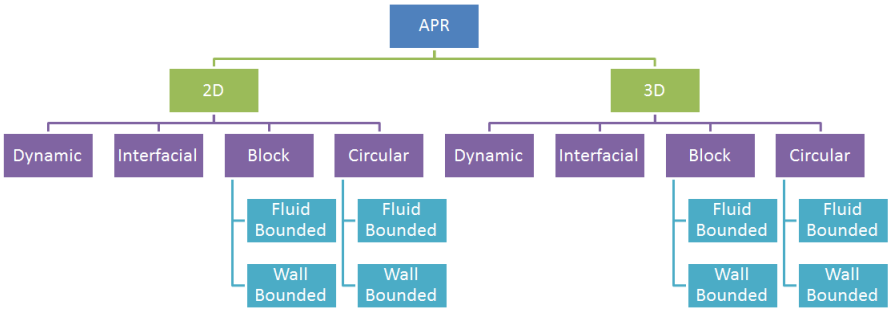


Figure 3.9: Schematics for different types of adaptivity implemented

3.5 Variable resolution

Any implementation of adaptivity in SPH framework should take care of the variable resolution encountered at the interface of refined and coarse particles. In this sec-

tion we explain how density fluctuations are minimised by implementation of corrective smoothed particle hydrodynamics (CSPH) and why the implementation was necessary. Also, we explain why the repulsive force as calculated by L-J potential had to be modified.

3.5.1 CSPH Implementation in current framework

Corrective Smoothed Particle Hydrodynamics is a higher order kernel approximation methods. This method was initially proposed by Chen et. al. for correcting the short coming of standard SPH formalism for particle deficiency at the boundaries. It is an interpolated scheme which corrects the particle properties by interpolating properties of its neighbours into it.

Let us consider a 1-D case of kernel approximation. Taylor expanding the series for $f(x)$ and integrating over the entire domain Ω we get,

$$\int_\Omega f(x)W_i(x)dx = f(x_i)\int_\Omega W_i(x)dx + f_x(x_i)\int_\Omega (x-x_i)W_i(x)dx + \frac{f_{xx}(x_i)}{2}\int_\Omega (x-x_i)^2 W_i(x)dx + ... \tag{3.20}$$

where $f_x(x_i)$ represents the first derivative and $f_{xx}(x_i)$ represents the second derivative and so on, $W_i(x) = W(x_i; h)$. Neglecting derivatives in Eq. 3.10 leads to ,

$$f(x_i) \approxeq \frac{\int_\Omega f(x)W_i(x)dx}{\int_\Omega W_i(x)dx} \tag{3.21}$$

for points par away from interface, the integral over $W_i(x)$ is equal to 1. Hence it reduces to standard kernel estimate.

Reasons for CSPH implementation

During particle refinement, at coarse and refined interface (fluid-fluid and boundary-fluid) we have an unequal size distribution of particles. As a result of the repositioning of refined particles with respect to coarse, some of the refined particles may lie outside the smoothing length of coarse as illustrated in the fig. 3.10 . As a result of which there are alternate peaks in density observed. In order to reduce this numerical error we invoke CSPH correction for density. We choose density as the property to be corrected because, all other SPH equations are calculated based on density of given step or the previous step. So a simple correction to density should result in a cascading effect leading to correction in other quantities.

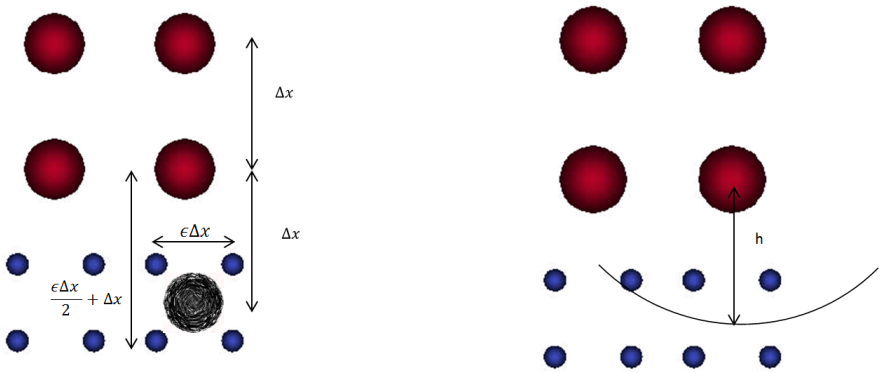


Figure 3.10: Problems arising due to variable resolution at interface. Black sketch represents the coarse particle which should have been present had there been no refinement.

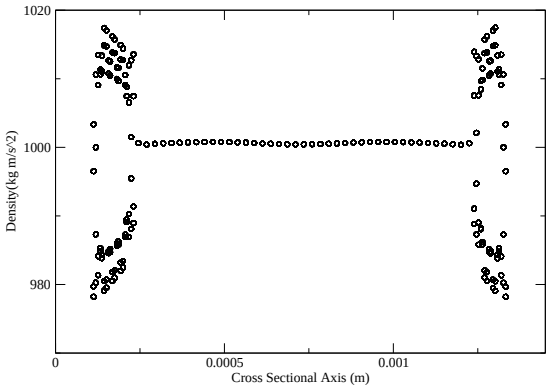


Figure 3.11: Huge density fluctuations are observed at the interface of two unequal size particles without correction.

3.5.2 Retuning L-J Parameters

Repulsive boundary force plays a very important role in adaptivity. Because of a change in the particle size and number density fluid may penetrate through the wall. To prevent this we use repulsive boundary force in the form of Lenard-Jones potential.

$$V_{LJ} = \epsilon \left[\left(\frac{\sigma}{r} \right)^{n1} - \left(\frac{\sigma}{r} \right)^{n2} \right] \tag{3.22}$$

where ϵ is the depth of potential well, σ is the finite distance at which inter particle separation is zero and r is the distance between the particles. In the fig. 3.12 we have plotted the the variation in LJ parameters for different set of parameters. We have done this exercise to find which combination of (n1,n2) say (4,2) or (12,6) [22] gives stable repulsive force which prevents particle penetration and at the same time doesn't affect the simulation dynamics.

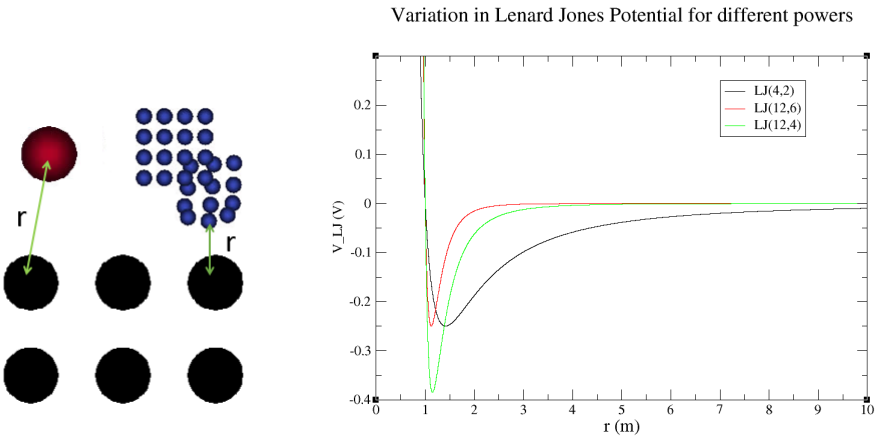


Figure 3.12: Figure on the left shows the reduction in distance between the boundary particle and refined particle as compared to that to coarse. Black- Boundary Particle, Red- Coarse particle, Blue - Refined particles. On the right, it shows variation in Lenard Jones potential for various power pairs

3.5.3 Background Pressure

An additional factor in pressure calculation is added to prevent particle clumping. Tensile instability as it is known in SPH literature happens due to build up of negative pressure. We employ the formulation developed by Monaghan [26] to deal with tensile instability.

This page was intentionally left blank

Chapter 4

Results and Discussion

In this chapter we focus on the validation of the APR implementation. We compare the results with the analytical results. We take into consideration a variety of cases to validate different implementations of APR.

Domain based wall bounded refinement is validated using Poiseuille Flow case by comparing with the analytical results. This validation result is for a closed system flow. Following which we validate free surface flow for domain based wall bounded system and for free surface flow using dam break 2D case. We validate domain based fluid bounded refinement using Cummin's dam break case [1] where we compare simulation results with the experimental results. Following which we run simulations through regular porous medium both 2D and 3D for validating dynamic refinement criterion and comparing it with analytical solution obtained from Ergun's equation [27]. We also, try to test the robustness of the implementation by trying to validate an inflow-outflow case which has certain limitations. We also show a proof of concept that adaptivity can capture temperature and heat effects at the same time conserving total energy. We also report inter-facial adaptivity in periodic systems and discuss the limitations of the approach formulated to deal with dynamics of multiphase closed system.

4.1 Poiseuille Flow 2D

We validate the methodology presented in the previous sections based on Poiseullie flow and compare it with analytical results calculated based on an weakly-compressible fluid.

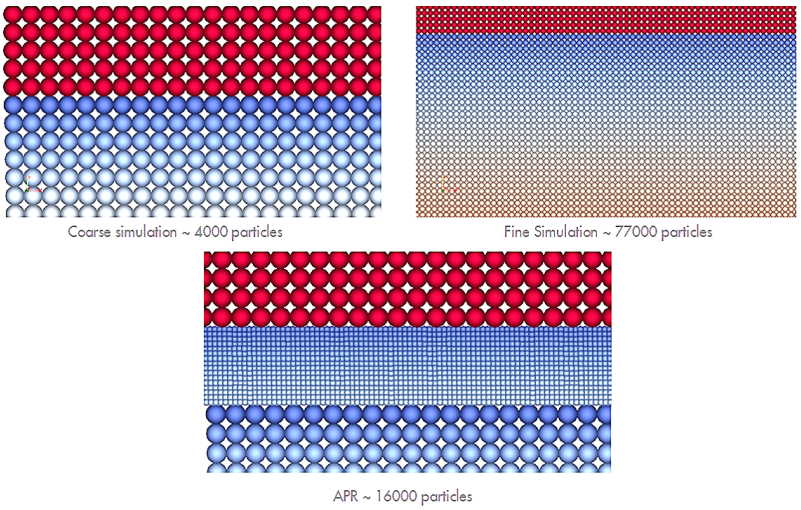


Figure 4.1: Pictographic representation of the above model (Visit: https://goo.gl/ZN9UKL for simulation or click on the image above)

4.1.1 Analytical Equation

We try to simulate a flow between two infinite plates and water flowing between the two so as to simulate Poiseuille flow in 2D. The analytic solution for which is determined by,

$$v_x(y,t) = \frac{F}{2\nu}y(y-L) + \sum_{n=0}^{\infty} \frac{4FL^2}{\nu\pi^3(2n+1)^3} sin\left(\frac{\pi y}{L}(2n+1)\right) exp\left(-\frac{(2n+1)^2\pi^2\nu}{L^2}t\right) \tag{4.1}$$

where, L is the width of the channel, $v_z(y,t)$ is the fluid velocity calculated at position y and time t, ν is the kinematic viscosity and F is the driving force applied to the flow.

4.1.2 Validation Runs

In order to validate adaptive particle resolution implementation and to show the advantage of adaptivity in compute cost we run the following simulations for the below mentioned geometry and parameters:

1. Coarse Simulation with 4500 particles
2. Fully Refined Simulation with 77000 particles
3. Adaptive particle resolution starting with coarse simulation.

We calculate the maximum velocity for each of the simulation and the results are shown in table 4.1. We also calculate the analytical value for the above mentioned geometry using eq. 4.1,

$$\text{Maximum Velocity attained at Steady State} = 1.88 \times 10^{-5}\text{m/s} \tag{4.2}$$

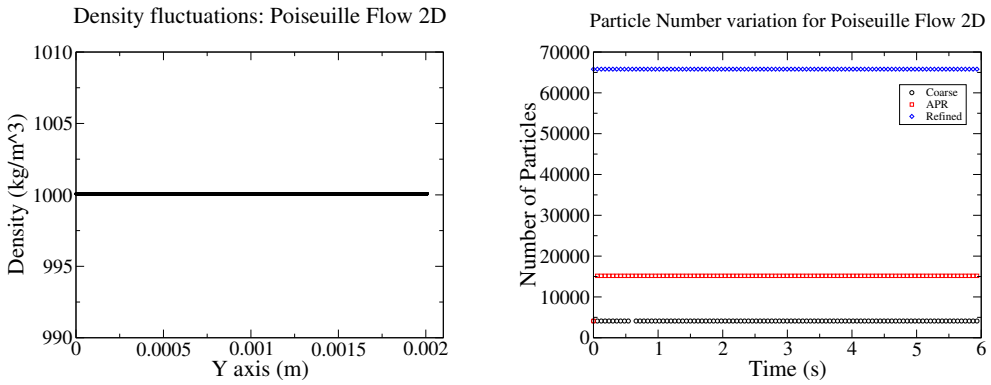


Figure 4.2: Figure on left shows the density fluctuations in Poiseuille flow 2D case, we observe that density fluctuations are less than (1-2)%, on the right, figure shows a comparison between number of particles of a coarse, fine and adaptive particle simulation.

Quantity	**APR**	**High Resolution**	**Coarse**
Avg. Time Step (in s)	0.024	0.19	0.012
Maximum velocity (m/s)	0.00001915	0.00001896	0.00001974
Error %	1.86	0.85	5
Number of Particles	16400	77000	4500
Simulation Time (s)	5.9	5.9	5.9
CPU Time (s)	22111	168739	10656
Initial Particle Separation (m)	0.0000245	0.00000613	0.0000245
No. of Splits	2	0	0
Split Particle Separation (m)	0.00000613	-	-

Table 4.1: Table shows a comparison between APR, Refined and Coarse particle size simulation results for Poiseuille Flow 2D Case

By comparing the results obtained from simulation and analytical equation we conclude that, the results from simulations with APR tend to match well with the results from a refined high resolution simulation and analytical results. We show a computational advantage of nearly a factor of 7 (700%) (fig. 4.4) there by preserving and predicting the correct maximum velocity (fig. 4.3).

As we know the crux of SPH calculation is density, we have plotted the variation of density with position for the final time step (fig. 4.2) and we observe density fluctuations

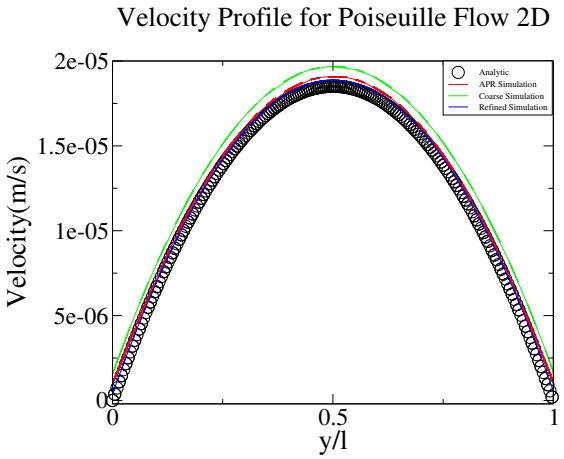


Figure 4.3: Figure shows the comparison of maximum velocity obtained in Poiseuille Flow 2D for coarse, refined, APR and analytical solution.

less than 1% which is consistent with WCSPH formulation.

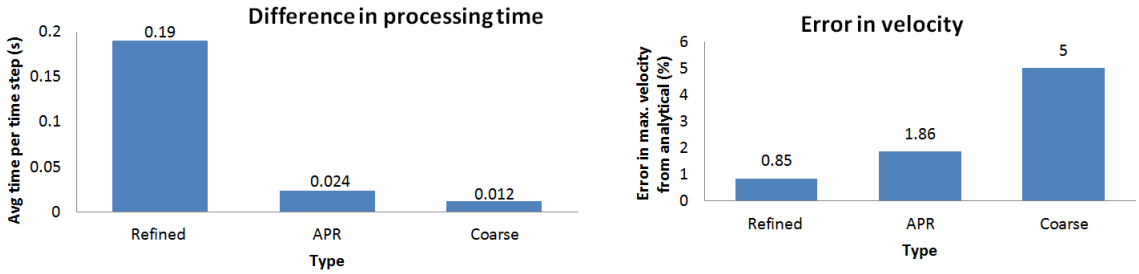


Figure 4.4: Pictographic representation of decrease in compute time and percentage error when compared with analytical results for maximum velocity

4.2 Dam Break

In order to validate domain based refinement for free surface flow (for both wall bounded and fluid bounded types), we consider the case of Dam Break.

4.2.1 Dam Break 2D

We consider a 2D Dam Break Case for wall bounded domain based refinement for free surface flows. In this model, we try to capture the wave topology when it bounces back after hitting a wall. We compare three different particle resolutions as before. While using APR we refine particles near wall and try to capture the topology with reduced

compute cost and compare it with a coarser system which doesn't capture the correct wave topology.

Results

We observe that APR simulation captures the topology of the system fairly well when compared to the refined system (as shown in fig. 4.5) and much better than the system with coarser particles. Here also we show the density fluctuations for the final time step and variation in total number of fluid particles (Fig. 4.6).

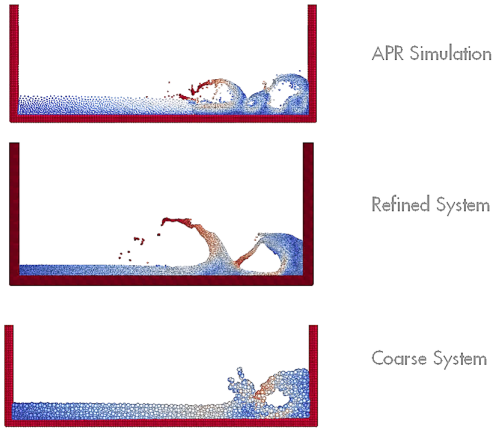


Figure 4.5: Comparison of the topology of wave for 2D Dam Break Simulation with Refined, Coarse and APR. Colours represent velocity gradient, Blue:Low, Red:High.(Visit: `https://goo.gl/NS9heS` for simulation or click on the image above)

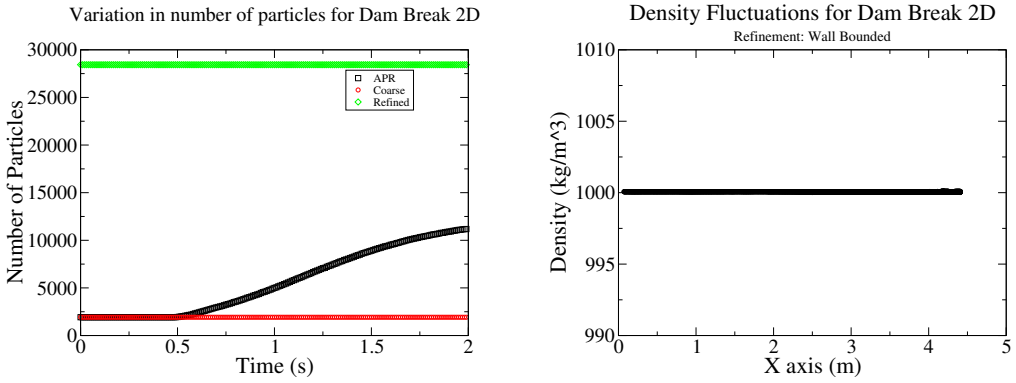


Figure 4.6: Figure on left shows varying particle number for the case of 2D dam break. We observe that the particle number increases and then decreases when the wave hits back and de-refinement takes place. On the right, it shows density fluctuations for the final save step of dam break 2D case.

4.2.2 Dam Break 3D

We consider this case to validate our implementation of refinement and de-refinement of particles in 3D. The case we consider is Cummins Dam Break [1]. In this, we measure the force of fluid on a vertical column and compare it with experimental results. We refine near the vertical column and coalesce else where (fluid bounded domain). There by we show the advantage of APR.

Experimental Results

Cummins et. al. [1] validated the force on a vertical column. They compared their SPH simulation results with experimental results using a system with equal particle size distribution and the results obtained by them are shown in fig. 4.7

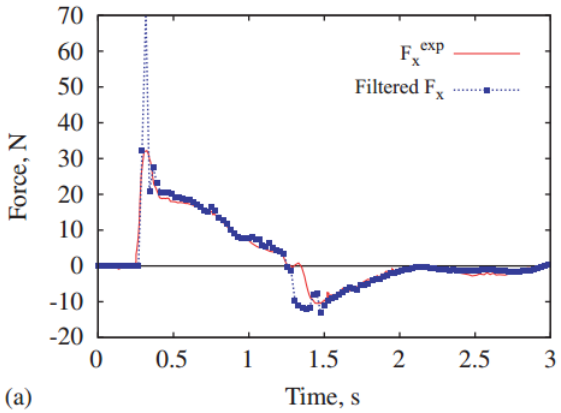


Figure 4.7: The above is a plot shown by Cummins [1] in their paper where they compare experimental results with computational results.

Validation

In our simulation, the metric of interest is the force on the vertical column during the wave surge. We extract the experimental data from fig. 4.7 for simulation validation. We compare our results obtained from refined, coarse and APR simulation with experimental data,

In fig. 4.8 we have calculated the force on the vertical column by adding up the X-component of forces on the column and excluding the force on the side walls and bottom. We see that force predicted by APR and refined seem to match well with the experimental data. The sudden surge in the initial force on the wall at about 0.5s is due the repulsive force of the boundary on fluid and is consistent with results from Cummins case. Force calculated by simulation with coarser particles is way off from experimental

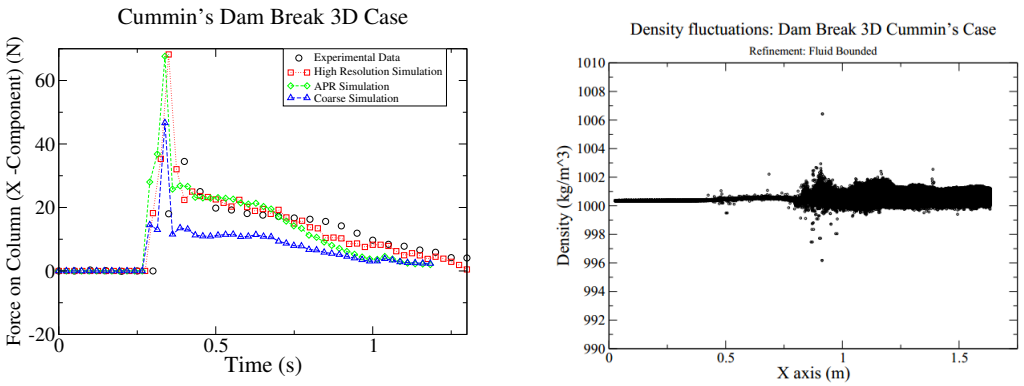


Figure 4.8: Figure on the left shows the force on the pillar in comparison with Refined, Coarse, APR and Experimental data obtained from Cummins DamBreak [1]. We observe the higher surge point due to repulsive boundary force in line with the results obtained by Cummins. On the right it shows density fluctuations in 3D Dambreak Case. We observe that density fluctuation is less than (1-2)% which is within acceptable error limit

results. This shows that 3D refinement and optimised scheme parameters developed by us in this work is capable of predicting correct experimental results. Here we compare only the initial surge and not the when the force exerted by the rebounding wave.

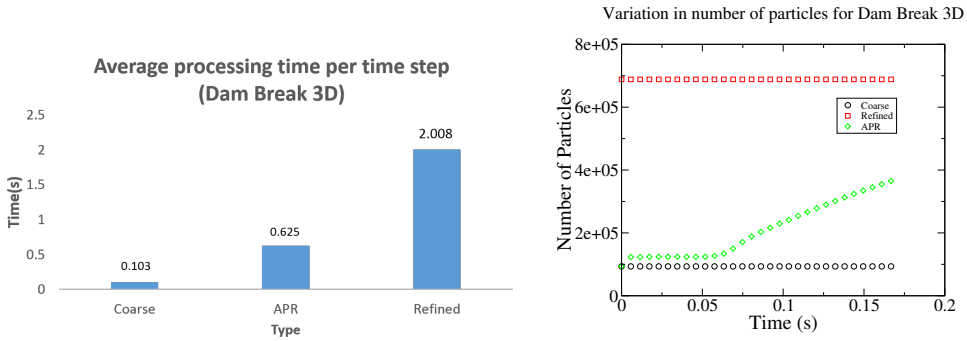


Figure 4.9: Figure on the left shows a comparison in processing time of three different resolution simulations for 3D Dam Break. On the right it shows a comparison of the number of particles for three different resolution simulation.

In fig 4.8 we observe that with the minimisation of error for 3D developed in this thesis works correctly predicts density for 3D free surface flow, fig. 4.9 shows it to be less than 2% which is consistent with WCSPH formalism.

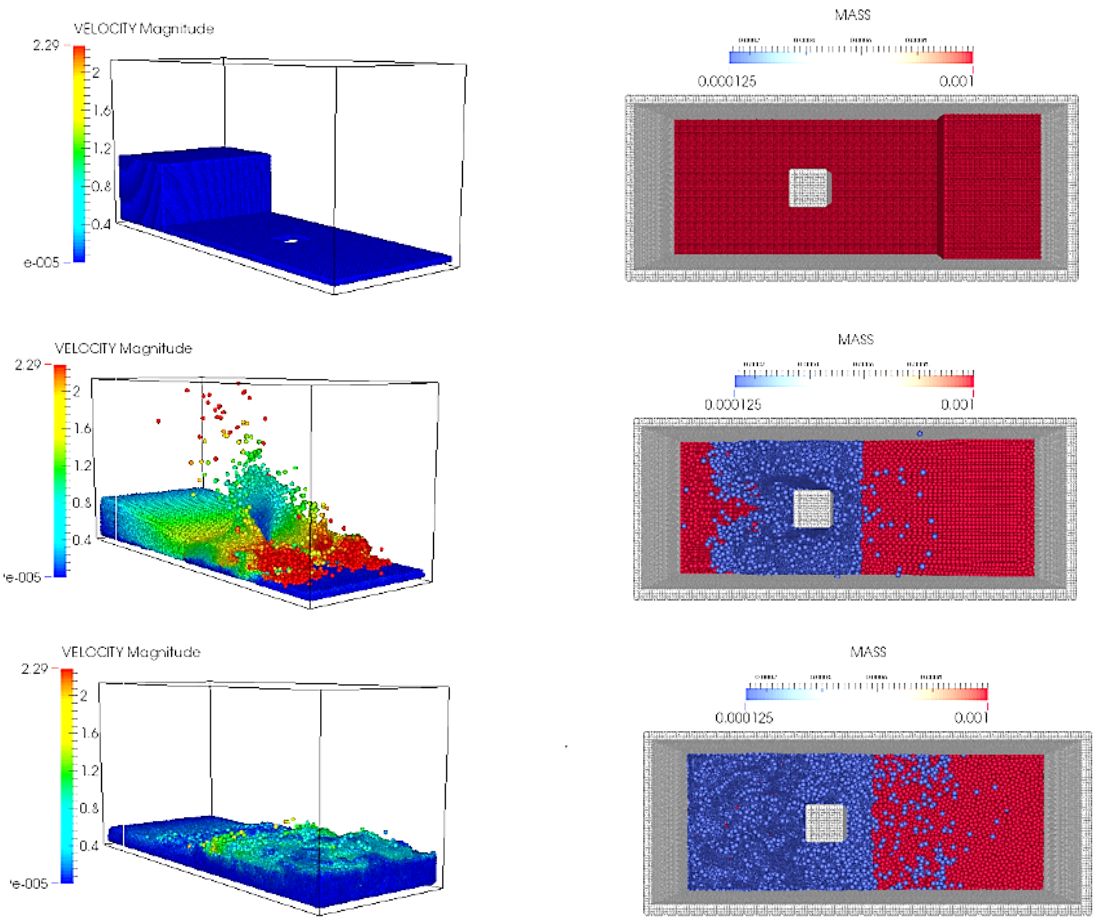


Figure 4.10: Figure shows different time snapshots of Dam Break 3D. Images on the left represent velocity gradient from a side view, the snapshots on the right represent mass gradient (Red: High, Blue: Low)(Visit: `https://goo.gl/1IX1gC` and `https://goo.gl/b8te1Y` for simulation or click on the image above)

4.3 Flow through regular porous medium

In a realistic scenario we would always come across numerous cases where the distribution of solid cannot be predicted apriori. To deal with such cases we employ dynamic refinement in which the program looks for the solid and refines the small region of fluid surrounding it. To test this concept and validate the same numerically, we consider the flow through a regular packed bed reactor. The validation of these cases also lays a foundation for simulating packed bed reactor with random catalyst pellet shapes.

Ergun's Equation: In a packed bed reactor simulation, the important metric of interest is pressure drop. This can be calculated using Ergun's equation for pressure drop [27].

$$\triangle p = \frac{150\mu L}{D_p^2}\frac{(1-\epsilon)^2}{\epsilon^3}v_s + 1.75L\frac{\rho}{D_p}\frac{(1-\epsilon)}{\epsilon^3}v_s^2 \tag{4.3}$$

In eq.4.3, $\triangle p$ represents pressure drop across the packed bed, μ is the viscosity, L length of packed bed, ϵ porosity of the system, v_s superficial velocity and D_p represents the diameter of the pellet. The first part of the equation is for Laminar regime (Blake-Kozeny Equation) and the second part is valid for Turbulent regimes(Burker-Plummer Equation). In our simulations we deal with turbulent regime so the second term is more prominent than the first.

4.3.1 Porous Medium 2D

We consider a packed 2D porous medium with periodic boundary conditions to validate dynamic refinement in 2D. We have packed 17 pellets in the system where the packing is in the form of HCP structure.

Validation

In order to validate the pressure drop in a packed bed reactor and show the advantage of APR over simulation with higher resolution of particles, we simulate the same system with different gravity values as shown in table 4.1.

Mass flow rate is calculated from which one can calculate the superficial velocity. To calculate the mass flow rate we consider a section of the system in the region where the pellets end. We calculate the number of particles which have passed a given line between two consecutive save steps taking into account that the particles which are undergoing periodic boundary condition is not taken into account. From superficial velocity one can calculate pressure drop using eq. 4.3.

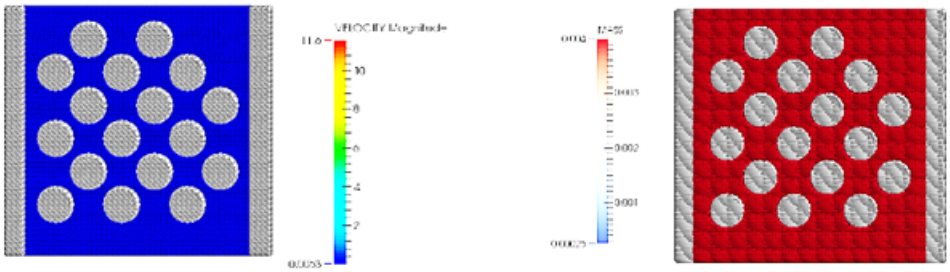
Pressure drop is calculated from simulation in a given length, by summing over the loss of force from the fluid onto the pellet. Given that, we have calculated the total force lost, knowing the geometry of the system one can find pressure drop, as pressure drop

Table 4.2: Table shows the comparison between pressure drop calculated using Ergun's equation and pressure drop calculated from simulation and corresponding deviation form Ergun's equation for different gravity values.

Type	dm (kg)	Gravity (m/s^2)	Velocity (m/s)	Ergun's Equation (Pa/m)	Simulation (Pa/m)	Error %
APR	1.0721	7.8	1.04	1099398.96	1114818.41	-1.40
APR	1.089	8.8	1.06	1135558.48	1148569.77	-1.14
APR	1.1032	9.81	1.07	1164063.61	1182135.76	-1.55
APR	1.2801	19.62	1.25	1567029.87	1519504.29	3.03
APR	1.398	29.43	1.36	1870669.19	1711472.56	8.51
APR	1.483	39.24	1.44	2104249.61	1899081.09	9.75
APR	1.543	49.05	1.50	2277817.62	2061881.58	9.47
Coarse	0.68	9.81	0.66	442629.55	1050388.62	-137.31
Coarse	0.98	19.62	0.95	918738.60	2130909.63	-131.34
Coarse	1.16	29.4	1.13	1286934.05	3059437.73	-137.73
Fine	1.132	9.81	1.10	1225593.83	1211900.80	1.11
Fine	1.253	19.62	1.22	1501420.15	1463295.48	2.53
Fine	1.364	29.46	1.33	1779049.44	1702439.27	4.30

is the force exerted by the fluid on the pellets per unit volume. From the Table 4.2 and Fig. 4.12 shows that APR simulations give much better result than that of simulations with coarser particles and are comparable The results also show that APR simulation results are comparable to simulation with system of higher resolution at much reduced compute cost. The comparison is shown in Fig. 4.13.

In Fig. 4.12 we observe that the density fluctuations in the final time step is less than 1% which is consistent with the WCSPH framework.



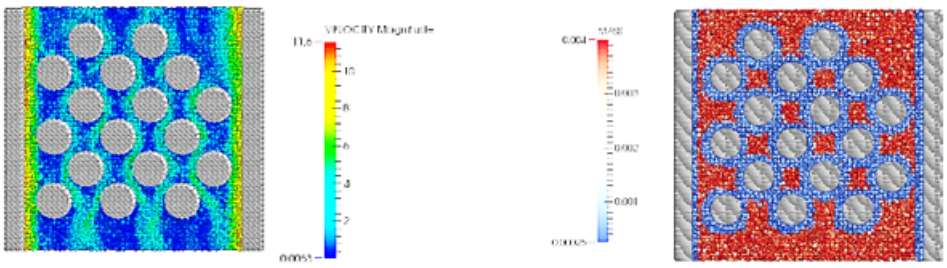


Figure 4.11: Figure shows different time snapshots of flow through a porous medium. Left: Velocity gradient with Red representing high and blue-low. Right: Mass gradient with Red representing high and blue-low (Visit: `https://goo.gl/R2ydxq` for simulation or click on the image above)

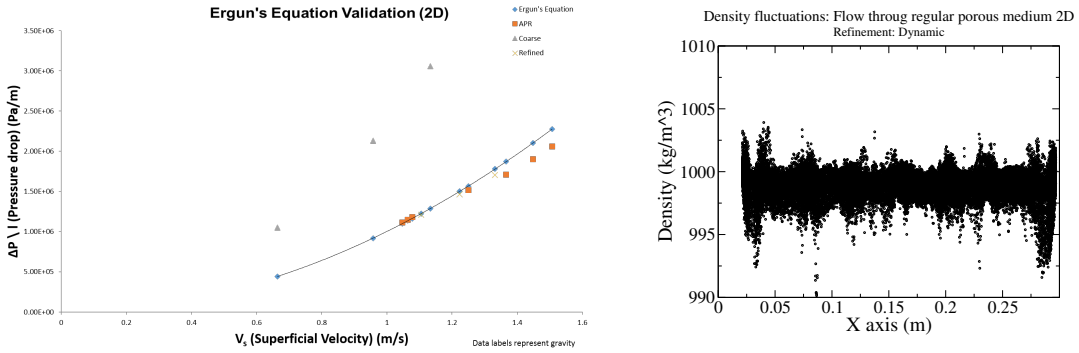


Figure 4.12: Figure on the left shows a comparison between pressure drop for refined, APR and coarse systems with that of Ergun's equation. On the right it shows the density fluctuations for flow through 2D regular porous medium. We observe that density fluctuations are less than (1-2)% which is within the permissible error limit.

4.3.2 Porous Medium 3D

We consider the case of 3D porous medium for validating dynamic refinement in 3D. This system consists of spherical pellets in a cylinder with periodic boundary conditions. Having shown the advantage of adaptivity in previous cases we consider only APR simulation and show that it matches with the analytical solution.

Validation

We follow a similar procedure as mentioned in the previous section. We compare our results with the analytical result obtained using Ergun's equations (Eq.4.3) for pressure dropped in a packed bed. As we have provided the detailed analysis of advantage of adaptivity for both 2D and 3D cases we do not do the same here as the time taken

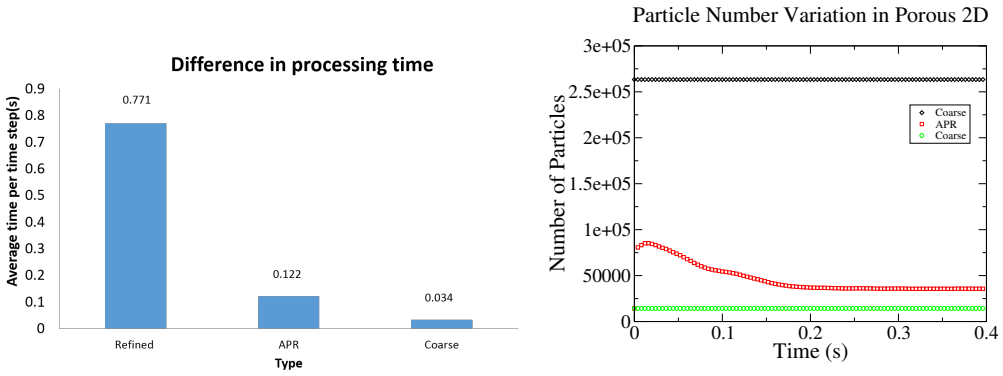


Figure 4.13: Figure on the left shows a comparison between average processing time per time step there by showing the advantage of APR. On the right it shows a comparison between total number of particles for different resolution of simulation.

based on operations per particle is 27.64 days for a single simulation. We have performed simulations for three different gravity values with APR, and their results are shown in table 4.3 and fig. 4.14. As before, we have plotted the density fluctuations for the final save step which is shown in fig. 4.14 and we observe that density fluctuation is less than 1%.

Table 4.3: Table shows comparison between results obtained from simulation using APR and Ergun's equation

dm kg	Gravity m/s^2	Velocity m/s	From Ergun's Equation Pa/m	From Simulation Pa/m	Error(in %)
1.411	5	11.691	953096.07	904420.93	5.10
1.423	10	11.790	969376.25	1011973.47	-4.39
1.621	20	13.43	1257904.13	1212119.34	3.63

4.4 Dam Break 2D with Heat

To deal with complex cases such as flow through a porous medium with heated pellets or any system involving heat transfer we extend our approach of particle refinement to systems involving heat. Each particle carries a physical entity (a passive scalar) called temperature and when a particle is refined, the same value of temperature is given to the refined particles, taking care conservation of energy. In the following example of heat conduction in 2D dam break case [28] we show a proof of concept that refinement near the boundary is able to conserve the total energy of the system. We begin with

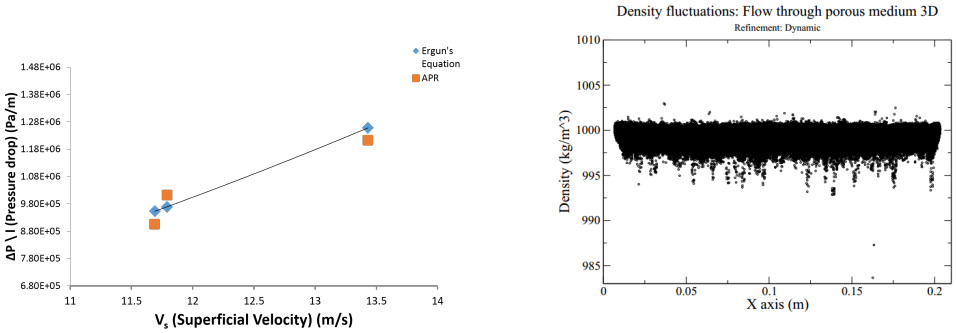


Figure 4.14: Figure on the left shows pressure drop calculations for flow through a Porous Medium 3D with Dynamic APR. On the right it shows density fluctuations for flow through 3D porous medium. We observe that density fluctuations in about (1-2)% which is within permissible error limit.

heated fluid and boundary being at relatively zero temperature.

4.4.1 Validation

We calculate the total energy of the system before the simulation and then calculate the total energy from simulation data by summing over each of energy of each particle. We use the following,

$$Q = mc\triangle T \tag{4.4}$$

where Q is the energy, m mass of each particle and $\triangle T$ change in temperature. From fig. 4.15 we observe that the systems conserves energy. We also plot density fluctuations for the final save step and observe that (fig. 4.15) density fluctuations is less than 1%.

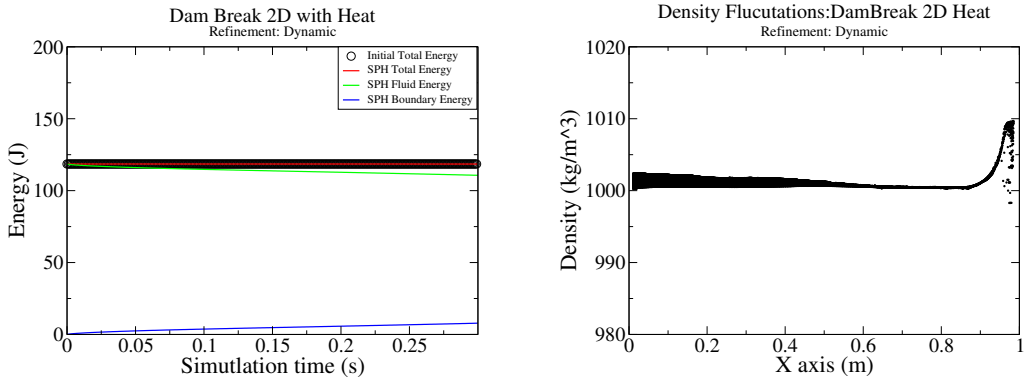


Figure 4.15: Figure on the left shows conservation of energy in 2D Dam Break Case with heat and dynamic refinement. This acts as a proof of concept that adaptivity is able to resolve the temperature effects. On the right it shows the density fluctuations for 2D DamBreak with Heat.

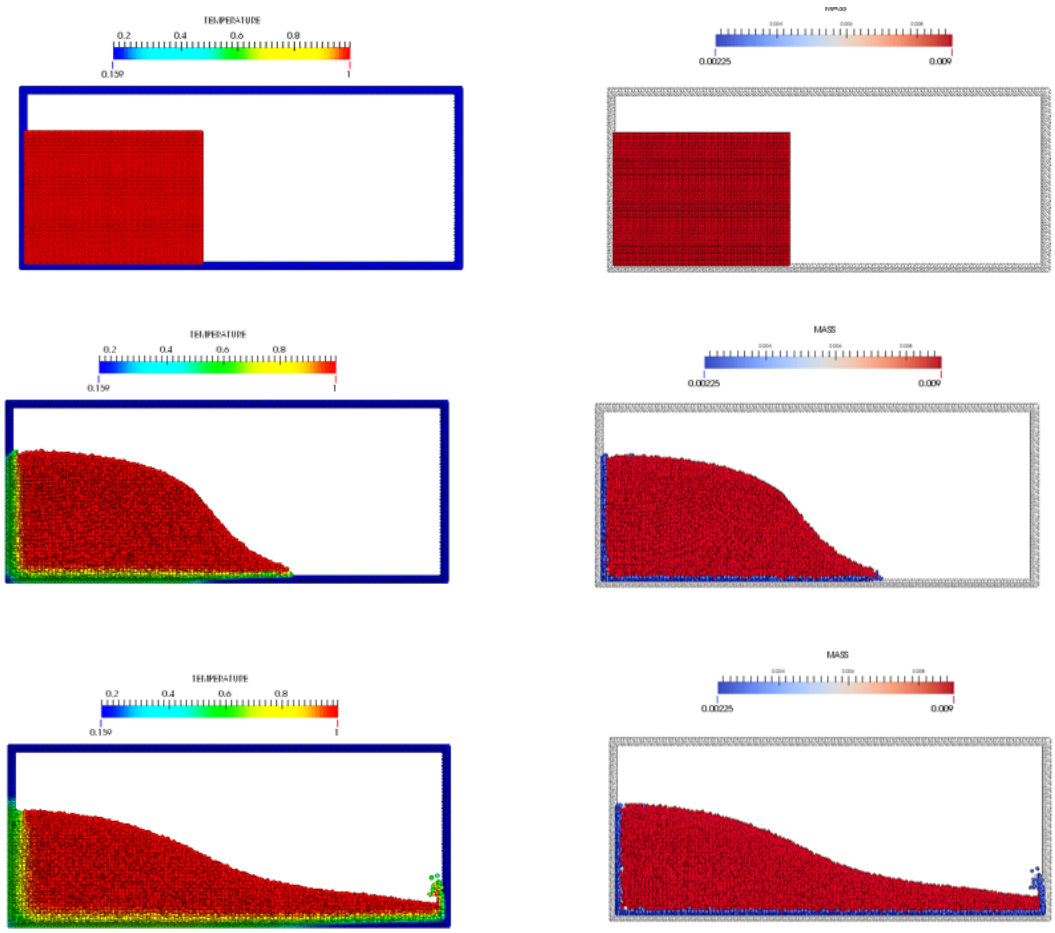


Figure 4.16: The figure above shows different time snapshots of dambreak with heat. Initially the fluid is heated and boundary is at a relatively zero temperature. The figure on the left represents the increase in boundary temperature and decrease in fluid temperature. Figure on the right shows the mass variation near the boundary (i.e.,dynamic refinement).

4.5 Heat conduction through a rod

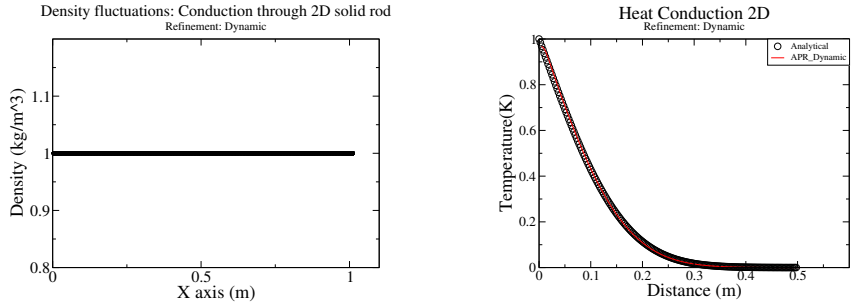
To validate heat conduction for a closed system and different types of refinement criteria we consider the heat conduction through a solid rod and compare it with the analytical results [29].

4.5.1 Conduction 2D

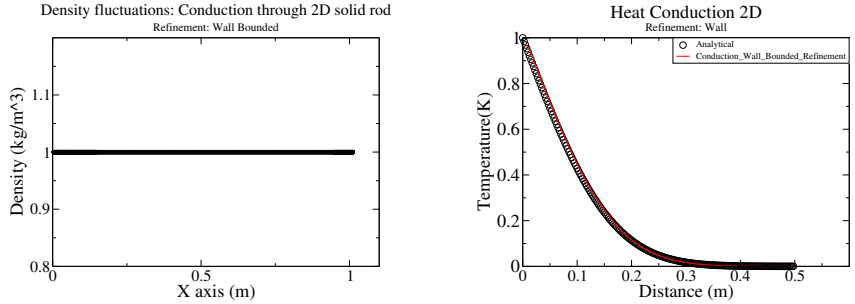
We consider three different implementation of adaptivity, wall bounded, fluid bounded and dynamic domain based refinement. We show as a proof of concept that APR results match with analytical results.

Validation

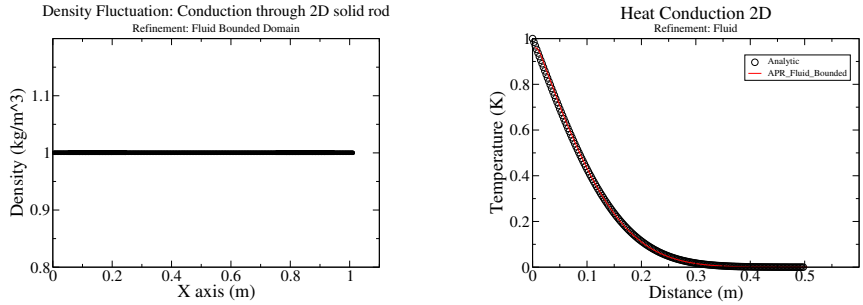
We compare the results obtained from simulation with analytical results [29] and they match very closely. we also plot the density fluctuation for each of the adaptivity cases and we observe density fluctuations to be less than 1% .



(a) The figure on left shows that density fluctuation for heat conduction with dynamic refinement, right shows comparison with analytical solution.



(b) The figure on top left shows density fluctuation for heat conduction with domain based wall bounded refinement, right shows comparison with analytical solution.



(c) The figure on top left shows density fluctuation for heat conduction with wall bounded refinement, right shows comparison with analytical solution.

Figure 4.17: The above figures show different particle resolution plots.

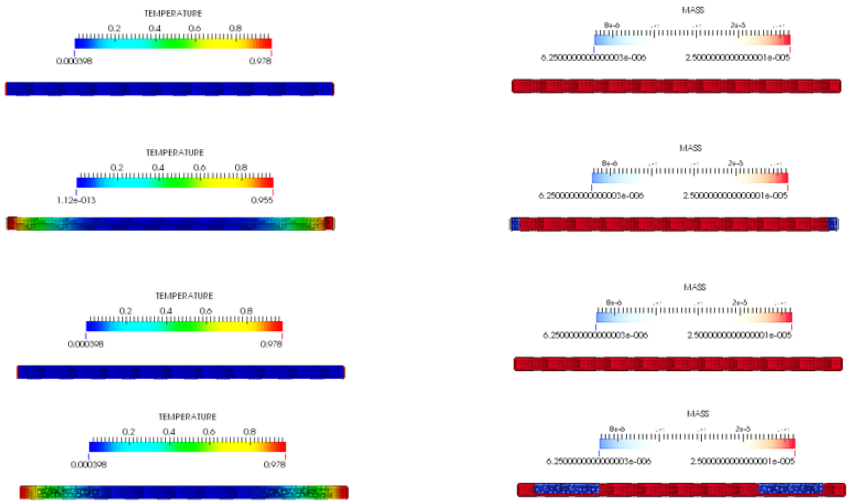


Figure 4.18: Figure shows conduction different time snapshots of heat conduction through a 2D rod. Top figure represents dynamics and wall bounded domain and bottom represents fluid bounded domain.

4.5.2 Conduction 3D

In order to validate heat conduction with 3D refinement we consider the same case as before but with a 3D rod and with fluid bounded refinement.

Validation

We compare the results obtained from simulation with analytical results [29] and they match very closely. we also plot the density fluctuation for each of the adaptivity cases and we observe density fluctuations to be less than 1% .

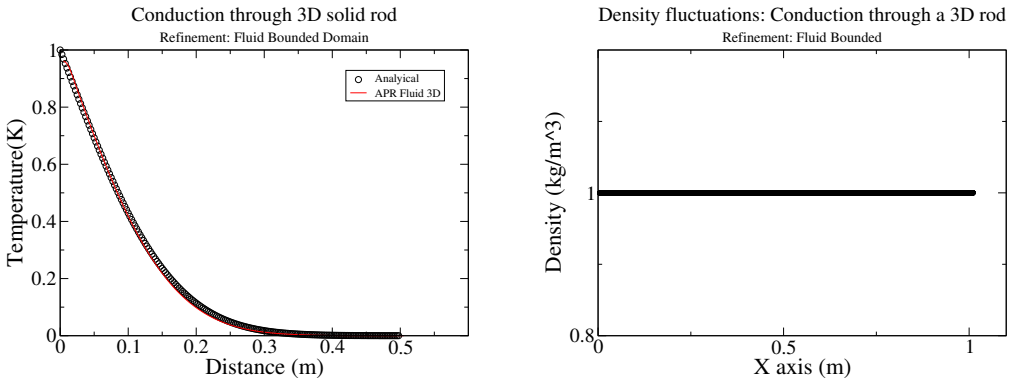


Figure 4.19: The figure on top right shows density fluctuation for heat conduction 3D with fluid bounded refinement, on the left shows comparison with analytic.

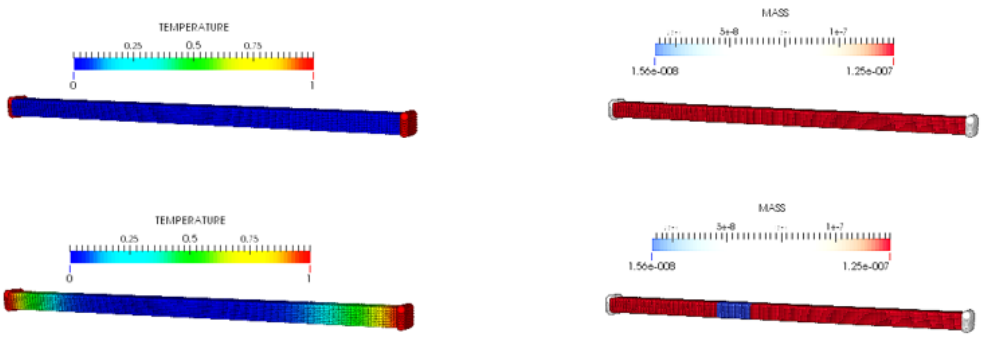


Figure 4.20: The above figure shows different time snapshots of heat conduction in a 3D rod with heated boundary. The plot on the left shows a variation in temperature variation and one on the right shows variation of mass.

4.6 Flow Past Cylinder

In order to study the applicability of adaptivity to systems with inflow-outflow boundary conditions, we employ a fluid bounded domain based refinement for the test case of flow past a cylinder.

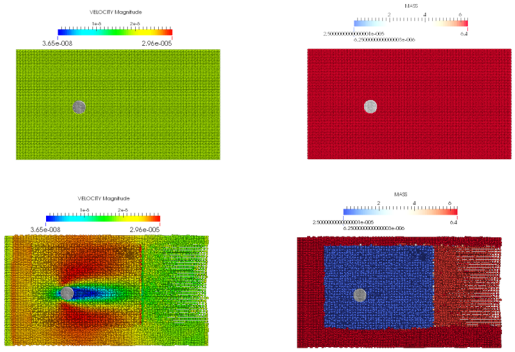


Figure 4.21: The above figure shows different time snapshots of flow past a cylinder. The figure on the left represents the velocity plot (Red: High and Blue: Low) and the figure on the right shows mass variation. We observe that qualitatively APR is able to predict the velocity profile built up. (Visit: `https://goo.gl/48zZqF` for simulation or click on the image above)

4.6.1 Validation

Inflow-Outflow boundary condition - implemented based on Lastiwka's approach [30]. The approach depends strongly on the use of Gaussian Kernel with additional compact support to account for larger number of particles for the stability of pressure waves generated in the inflow zone. Our formulation of adaptivity is based on minimisation of

kernel refinement error for cubic kernel (E.q.2.5). Because of this discrepancy, the results obtained from APR don't match well with the refined case. Due to time constraint and the scope of this thesis we didn't try to reformulate inflow-outflow boundary condition or optimise alpha and epsilon for Gaussian kernel. Force on the cylinder for three different particle resolution is shown in fig. 4.22 and density fluctuations are shown in fig 4.22.

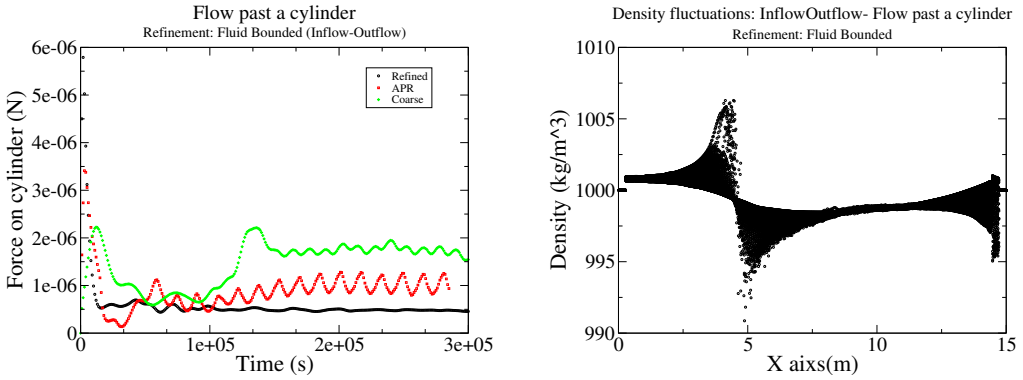


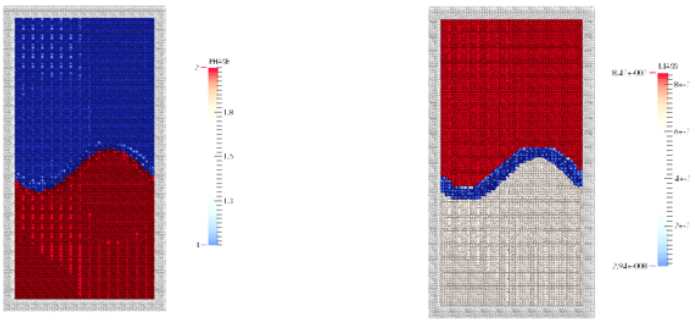
Figure 4.22: Figure on the left shows force on the cylinder from refined SPH simulation, analytical equation and coarse SPH simulation. On the right it shows, that density fluctuations for flow past a cylinder for inflow-outflow problems is within permissible error limit of (1-2)%.

4.7 Multiphase

In this section, we focus on the effects of adaptivity when dealing with systems involving multiphase flows. We study the effects of refinement on closed and open systems.

4.7.1 Closed System

We first study the dynamics of a closed system with higher density fluid on top of lower density fluid and observe the effects in presence of gravity.



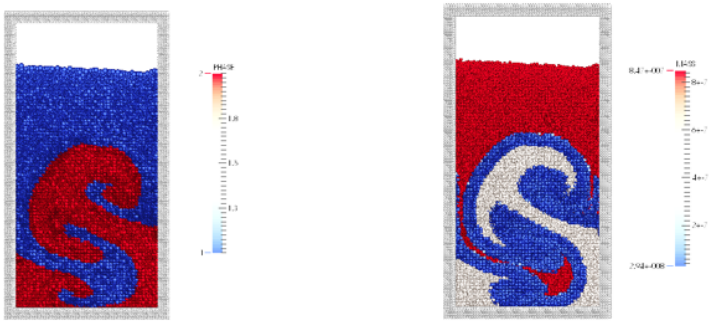


Figure 4.23: The figure above shows two different time snapshots for RT instability. The figure on the left shows phase plot and the figure on the right shows mass variation. We observe the dynamics to be incorrect with the experiments, solution to which is beyond te scope of this thesis. (Visit: `https://goo.gl/J1I8br` for simulation or click on the image above)

Results

We observe that the adaptivity fails for closed system interfacial APR. The propagation of pressure is improper through the refined regions. This requires development of new pressure equation, which is beyond the scope of this thesis. We plot the density fluctuation for the last save steps, although the fluctuations are within (1-2)% but the results don't match with the experimental results.

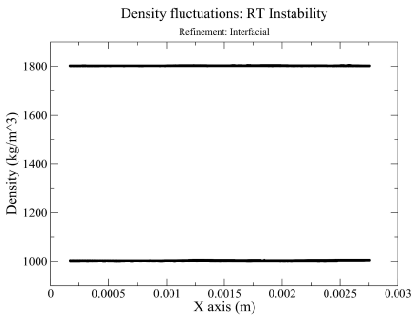


Figure 4.24: Figure shows density fluctuations for Rayleigh-Taylor instability We observe that density fluctuations is (1-2)% of respective phases which is within permissible error limit

4.7.2 Open System

In this section we try to study the effects of interfacial APR in multiphase system with periodic boundary condition.

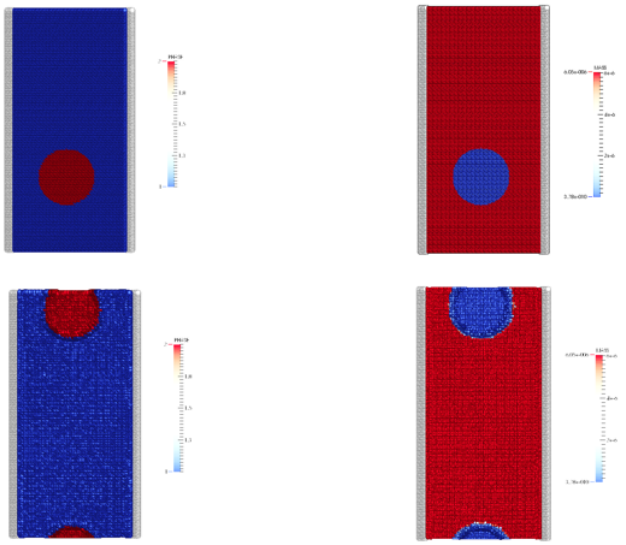


Figure 4.25: The above figure shows different time snapshots of bubble in water with periodic boundary conditions. The figure on left shows difference in phases and one on the right shows variation in mass.

Results

We observe that interfacial APR is able to retain the bubble shape and is stable with periodic boundary condition, and the density fluctuations are less (1-2)% for each phase. This proof of concept is an important step for reactor modelling which involves multi-phase flows inside randomly packed catalytic reactor.

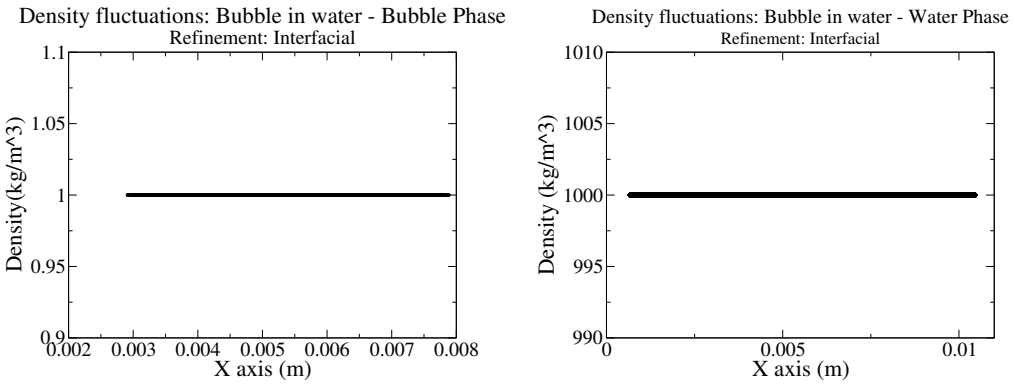


Figure 4.26: Figure on the left shows density fluctuations for Bubble(air) phase which is (1-2)% and is within permissible error limit. On the right figure shows density fluctuations for water phase which is (1-2)% and is within permissible error limit

Chapter 5

Conclusions

The new implementation of adaptive particle resolution (APR) in the SPH method offers the capability of doing multi-resolution simulations in SPH framework. Our approach offer both runtime particle refinement and de-refinement to provide dynamic adaptive resolution and our implementation takes care of both 2D and 3D. We have implemented various kinds of APR such as domain based or based on the distance between solid and fluid particles and these implementations in the SPH method provide a simple yet powerful framework for applications like off-shore structures and reactor modelling

In general, the current APR implementation can be applied for simulating multiscale models or systems that develop a wide spectrum of length scales. We have clearly shown the reduction in computational cost (simulation time) in the APR SPH simulations without losing the accuracy. All of our simulations were run in single machine and multiple cores (shared memory).

The main limitation of our implementation is in memory model, which makes it less efficient for run across multiple machines. Our recommendation for future is to extend the memory model to dynamic memory model so that advantage of APR is achieved across multiple machines (distributed memory) as well.

List of Abbreviations

APR	**A**daptive **P**article **R**esolution
SPH	**S**moothed **P**article **H**ydrodynamics
NNS	**N**earest **N**eighbor **S**earch
SPHERIC	**SPH** **E**uropean **R**esearch **I**nterest **C**ommunity
WCSPH	**W**eakly **C**ompressible **S**moothed **P**article **H**ydrodynamics
EOS	**E**quation **O**f **S**tate

Bibliography

[1] SJ Cummins, TB Silvester, and Paul W Cleary. Three-dimensional wave impact on a rigid structure using smoothed particle hydrodynamics. *International journal for numerical methods in fluids*, 68(12):1471–1496, 2012.

[2] Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.

[3] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[4] S Børve, M Omang, and J Trulsen. Regularized smoothed particle hydrodynamics: a new approach to simulating magnetohydrodynamic shocks. *The Astrophysical Journal*, 561(1):82, 2001.

[5] S Børve, M Omang, and J Trulsen. Regularized smoothed particle hydrodynamics with improved multi-resolution handling. *Journal of Computational Physics*, 208(1):345–367, 2005.

[6] Martin Lastiwka, Nathan Quinlan, and Mihai Basa. Adaptive particle distribution for smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 47(10-11):1403–1409, 2005.

[7] J. Feldman and J. Bonet. Dynamic refinement and boundary contact forces in sph with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering*, 72(3):295–324, 2007.

[8] R Vacondio, BD Rogers, and PK Stansby. Accurate particle splitting for smoothed particle hydrodynamics in shallow water with shock capturing. *International Journal for Numerical Methods in Fluids*, 69(8):1377–1410, 2012.

[9] R Vacondio, BD Rogers, PK Stansby, and P Mignosa. Shallow water sph for flooding with dynamic particle coalescing and splitting. *Advances in Water Resources*, 58:10–23, 2013.

[10] R Vacondio, BD Rogers, PK Stansby, P Mignosa, and J Feldman. Variable resolution for sph: a dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering*, 256:132–148, 2013.

[11] Yaidel Reyes López, Dirk Roose, and Carlos Recarey Morfa. Dynamic particle refinement in sph: application to free surface flow and non-cohesive soil simulations. *Computational Mechanics*, 51(5):731–741, 2013.

[12] Daniel A Barcarolo, David Le Touzé, Guillaume Oger, and F De Vuyst. Adaptive particle refinement and derefinement applied to the smoothed particle hydrodynamics method. *Journal of Computational Physics*, 273:640–657, 2014.

[13] R. Vacondio, B.D. Rogers, P.K. Stansby, and P. Mignosa. Variable resolution for sph in three dimensions: Towards optimal splitting and coalescing for dynamic adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 300:442 – 460, 2016.

[14] S Adami, XY Hu, and NA Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012.

[15] Daniel J Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, 2012.

[16] Yi Zhu, Patrick J Fox, and Joseph P Morris. A pore-scale numerical model for flow through porous media. *International journal for numerical and analytical methods in geomechanics*, 23(9):881–904, 1999.

[17] Gui-Rong Liu and Moubin B Liu. *Smoothed particle hydrodynamics: a meshfree particle method*. World Scientific, 2003.

[18] Alexandre M Tartakovsky, Paul Meakin, Timothy D Scheibe, and Rogene M Eichler West. Simulations of reactive transport and precipitation with smoothed particle hydrodynamics. *Journal of Computational Physics*, 222(2):654–672, 2007.

[19] WF Van Gunsteren and HJC Berendsen. A leap-frog algorithm for stochastic dynamics. *Molecular Simulation*, 1(3):173–185, 1988.

[20] JJ Monaghan. Smoothed particle hydrodynamic simulations of shear flow. *Monthly Notices of the Royal Astronomical Society*, 365(1):199–213, 2006.

[21] JJ Monaghan and RA Gingold. Shock simulation by the particle method sph. *Journal of computational physics*, 52(2):374–389, 1983.

[22] Joe J Monaghan. Simulating free surface flows with sph. *Journal of computational physics*, 110(2):399–406, 1994.

[23] Songdong Shao and Edmond YM Lo. Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface. *Advances in water resources*, 26(7):787–800, 2003.

[24] Kamil Szewc, Jacek Pozorski, and J-P Minier. Analysis of the incompressibility constraint in the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 92(4):343–369, 2012.

[25] K. Pan, R. H. A. IJzermans, B. D. Jones, B. W. H. A. Thyagarajan, van Beest, and J. R. Williams. Application of the sph method to solitary wave impact on an offshore platform. *Computational Particle Mechanics*, pages pp 1–12, 12 September 2015.

[26] Joseph J Monaghan. Sph without a tensile instability. *Journal of Computational Physics*, 159(2):290–311, 2000.

[27] Sabri Ergun and Ao Ao Orning. Fluid flow through randomly packed columns and fluidized beds. *Industrial & Engineering Chemistry*, 41(6):1179–1184, 1949.

[28] Li-qiang MA, Jian-zhong CHANG, Mou-bin LIU, and Han-tao LIU. Numerical simulation of dam-break flows using sph method [j]. *Hydro-Science and Engineering*, 3:011, 2010.

[29] JH Jeong, MS Jhon, JS Halow, and J Van Osdol. Smoothed particle hydrodynamics: Applications to heat conduction. *Computer Physics Communications*, 153(1):71–84, 2003.

[30] Martin Lastiwka, Mihai Basa, and Nathan J Quinlan. Permeable and non-reflecting boundary conditions in sph. *International journal for numerical methods in fluids*, 61(7):709–724, 2009.

Appendix A

Derivation of Equation of State from Lagrangian

Consider a Lagrangian for an N particle system,

$$L = \sum_{i=1}^{N} m_i \left[\frac{1}{2} \mathbf{v}_i^2 - e_i(\rho_i, s_i) \right] \tag{5.1}$$

where e_i represents internal energy per unit mass. We choose the above Lagrangian as it confers symmetric and conservative properties to the derived equations. Applying least action principle to to Lagrangian yields the following equation for a particle k,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \mathbf{v}_k} \right) - \frac{\partial L}{\partial \mathbf{x}_k} = 0 \tag{5.2}$$

with

$$\frac{\partial L}{\partial \mathbf{x}_k} = m_k \mathbf{v}_k \tag{5.3}$$

$$\frac{\partial L}{\partial \mathbf{x}_k} = - \sum_{i=1}^{N} m_i \left. \frac{\partial e_i}{\partial \rho_i} \right|_s \frac{\partial \rho_i}{\partial \mathbf{x}_k} \tag{5.4}$$

Equation 5.4 is valid for constant entropy, s. As there are no dissipative terms being considered at the point the equation conserves entropy. Using thermodynamic first principle for constant entropy.

$$\left. \frac{\partial e_i}{\partial \rho_i} \right|_s = \frac{P}{\rho^2} \tag{5.5}$$

Substituting in eq. 5.4,

$$\frac{\partial \rho_i}{\partial \mathbf{x}_k} = \left[1 - \frac{\partial h_i}{\partial \rho_i} \sum_{p=1}^{N} m_p \frac{\partial W_{i,p}(h_i)}{\partial h(x_i)} \right]^{-1} \sum_{p=1}^{N} m_p \frac{\partial W i, p(h_i)}{\partial \mathbf{x}_k} (\delta_{i,k} - \delta_{p,k}) \tag{5.6}$$

symmetrising the kernel conserves physical laws, there by allowing for variable smoothing length. We get,

$$\frac{d\mathbf{v}_k}{dt} = - \sum_{i=1}^{N} m_i \left(\frac{P_i}{\rho_i^2} + \frac{P_k}{\rho_k^2} \right) . \nabla_k W_{k,i} \tag{5.7}$$

Eq. 5.7 represents the most simplest form of SPH equation. Further forms with dissipation can be derived based on a similar approach taking different approximations into consideration.

Appendix B
SPHERIC Conference Abstract

The following abstract was submitted for the upcoming annual international conference organised by SPH European Research Interest Community (SPHERIC) and is one of the most important conference in the field of SPH. This abstract was considered to be suitable and accepted for a full publication and a oral presentation at SPHERIC Conference to be held in Munich, Germany (Technische Universitt Mnchen) from June 13-16, 2016. "The quality of the abstract was assessed using averaged ratings for 3 equally important categories. i.e. (1) novelty, (2) applicability & (3) predictive accuracy and predictive improvements over state-of-the-art. The abstract was reviewed at least by three experts " [1]. As the notification was given on 15 March, 2016, the full-paper is currently work in progress.

[1]Source: SPHERIC 2016: `https://www.events.tum.de/frontend/index.php?folder_id=268` (Accessed on 23rd March, 2016)

Modeling the multiphase flow characteristics in random packed bed reactor with complex shapes of catalyst pellets using adaptive SPH

S. Sarangi*[1], A. Thyagarajan†[2], K. Pan[3], and J.R. Williams[3]

[1]Indian Institute of Science Education and Research, Pune
[2]Shell Technology Centre, Bangalore
[3]Massachusetts Institute of Technology, United States of America

The goal of the study is to use Smoothed Particle Hydrodynamics (SPH) for better understanding of the flow properties of packed beds, and use this method for a smarter reactor design. So far, the influence of process parameters (flow of gas, liquid, pressure, temperature) on the pressure drop and heat transfer properties of the bed could be studied either experimentally or using CFD for small systems. The shape of the catalyst pellet is known to have an effect on properties like heat transfer and pressure drop. This can be qualitatively understood as one can expect the shape influencing both the catalyst packing as well as the flow patterns. SPH can be used to address flow in the catalyst bed in a systematic, quantitative approach.

SPH is advantageous for porous medium flow because it can handle complex geometries, multiphase flows and heat transfer. SPH can also be easily extended from 2D to 3D, and it can treat the flow inside pore structures at a mesoscopic level if needed. To successfully apply SPH for this area, we have 1) extended adaptive resolution algorithm in SPH proposed by Roose *et al.*. [1] to 3-D systems, 2) converted the complex geometry of the packed bed reactor into SPH particles (the geometry contains the validated bed packing for complex shapes of the catalyst pellets), 3) calculated the pressure drop characteristics for single phase and multiphase and 4) used inflow-outflow boundary condition to enable different inlet velocities for different phases. The method can be extended to non-isothermal studies.

Previous methods in Adaptive Particle Resolution (APR) are either based on remeshing, particle insertion/removal [1], or on a varying smoothing length [2]. Recently, Vacondio [2] showed three optimal methods of particle distribution to minimize mass density error. We employ a similar stencil with each particle being refined to 8 smaller particles, each of which are placed at the edge of a cube.We assign all smaller particles equal masses as opposed to the unequal mass distribution by Vacondio [2]. We minimize the error encountered in the gradient of kernel. We have also implemented simplistic de-refinement algorithm, where particles are merged in pairs. We employ a higher order correction to deal with density fluctuations and a single layer of boundary particles representing the complex geometry of the bed packing.

We have tested and validated the implementation against 2D and 3D porous medium cases involving ordered geometry with both periodic and inflow-outflow [3] boundary conditions. The results match closely with the predictions made by Ergun's equation [4]. We have simulated the flow through a random packed bed reactor packed with cylindrical pellets. Although we will show results for only cylindrical pellets, this approach works for any arbitrary pellet shape. Our work shows that adaptive SPH can be used for simulating multiphase flows inside a randomnly packed bed reactor for predicting bed characteristics.

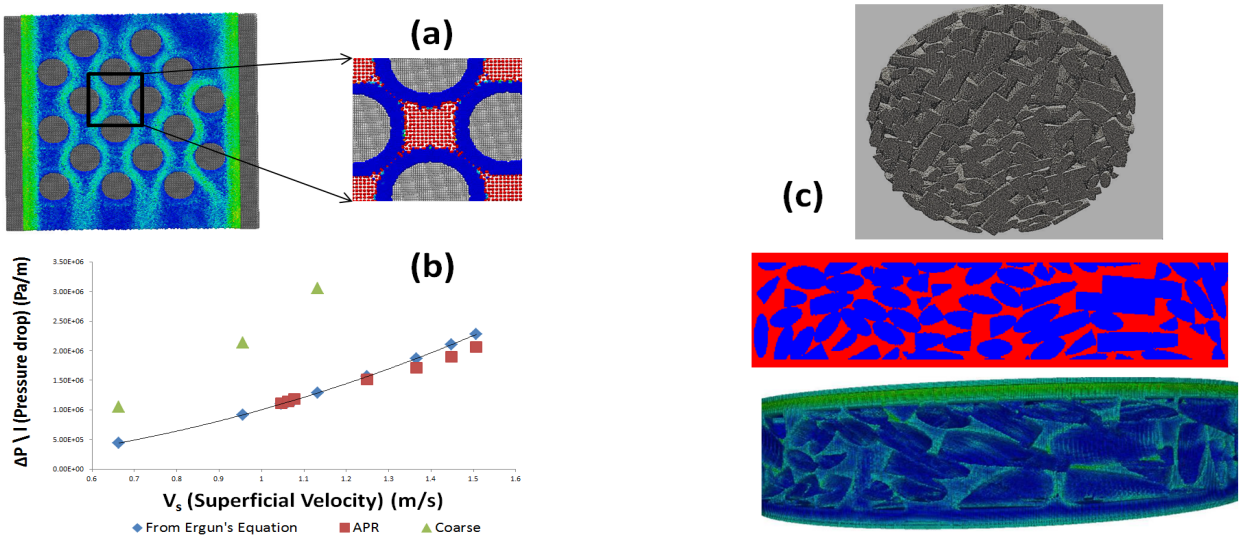


Figure 1: (a) Left: velocity profile of flow in porous medium in 2D (blue = low velocity, green = high velocity). Right: zoomed mass plot with red representing coarse particles and blue representing refined ones. (b) Comparison of SPH predicted pressure drop (with and without adaptivity) against Ergun's Equation. (c) Represents complex geometry of packing of cylindrical pellets (3D system). Top: top view of the packed bed. Middle: vertical slice showing pellets in blue colour and fluid in red. Bottom: flow dynamics inside this packed bed reactor (blue = low velocity, green = high velocity).

References

[1] Y. R. López, D. Roose, and C.R. Morfa. Dynamic particle refinement in sph: Application to free surface flow and non-cohesive soil simulations. *Comput. Mech.*, 51(5):731–741, May 2013.

[2] R. Vacondio, B.D. Rogers, P.K. Stansby, and P. Mignosa. Variable resolution for sph in three dimensions: Towards optimal splitting and coalescing for dynamic adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 300:442 – 460, 2016.

[3] M.Lastiwka and N.J. Quinlan M. Basa. Permeable and non-reflecting boundary conditions in sph. *International Journal for Numerical Methods in Fluids*, 61(7):709–724, 2009.

[4] S. Ergun. Fluid flow through packed columns. *Chem. Eng. Prog*, 48, 1952.

*sohan.sarangi@gmail.com
†a.thyagarajan@shell.com