Improvement and Application of Smoothed Particle Hydrodynamics in Elastodynamics

LISHA HE

How to cite:

HE, LISHA (2015) Improvement and Application of Smoothed Particle Hydrodynamics in Elastodynamics. Doctoral thesis, Durham University.

Durham University

Improvement and Application of Smoothed Particle Hydrodynamics in Elastodynamics

Thesis by

Lisha He

Submitted as partial consideration towards
the degree of Doctor of Philosophy

Mechanics Group
School of Engineering & Computing Sciences
Durham University
United Kingdom

November 2015

Improvement and Application of Smoothed Particle Hydrodynamics in Elastodynamics

Lisha He

Abstract

This thesis explores the mesh-free numerical method, Smooth Particle Hydrodynamics (SPH), presents improvements to the algorithm and studies its application in solid mechanics problems. The basic concept of the SPH method is introduced and the governing equations are discretised using the SPH method to simulate the elastic solid problems. Special treatments are discussed to improve the stability of the method, such as the treatment for boundary problems, artificial viscosity and tensile instability. In order to improve the stability and efficiency, (i) the classical SPH method has been combined with the Runge-Kutta Chebyshev scheme and (ii) a new time-space Adaptive Smooth Particle Hydrodynamics (ASPH) algorithm has been developed in this thesis.

The SPH method employs a purely meshless Lagrangian numerical technique for spatial discretisation of the domain and it avoids many numerical difficulties related to re-meshing in mesh-based methods such as the finite element method. The explicit Runge-Kutta Chebyshev (RKC) scheme is developed to accurately capture the dynamics in elastic materials for the SPH method in the study. Numerical results are presented for several test examples applied by the RKC-SPH method compared with other different time stepping scheme. It is found that the proposed RKC scheme offers a robust and accurate approach for solving elastodynamics using SPH techniques. The new time-space ASPH algorithm which is combining the previous ASPH method and the RKC schemes can achieve not only the adaptivity of the particle distribution during the simulation, but also the adaptivity of the number of stage in one fixed time step. Numerical results are presented for a shock wave propagation problem using the time-space ASPH method compared with the analytical solution and the results of standard SPH. It is found that using the dynamic adaptive particle refinement procedure with adequate refinement criterion, instead of adopting a fine discretisation for the whole domain, can achieve a substantial reduction in memory and computational time, and similar accuracy is achieved.

Declaration

The work in this thesis is based on research carried out in the Mechanics Group, School of Engineering and Computing Sciences, Durham University. No part of this report has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Journal

L. He and M. Seaid. Runge-Kutta-Chebyshev SPH algorithm for elastodynamics. *International Journal for Numerical Methods in Engineering* 2015 (under review).

L. He and M. Seaid. Time-space adaptive SPH method for shock propagation in solids. *Advances in Applied Mathematics and Mechanics* 2015 (submitted).

Conference

L. He, R.S. Crouch, M. Seaid and C.E. Augarde. Strong-stability preserving explicit Runge-Kutta methods for SPH elastodynamics. *Asia Pacific Congress on Computational Mechanics and International Symposium on Computational Mechanics (APCOM & ISCM)*, Singapore, 2013.

L. He, R.S. Crouch and M. Seaid. Comparative study of time-stepping schemes for SPH solution in elastodynamics. *International Conference on Computational Mechanics (CM13)*, UK, 2013.

Acknowledgments

I would like to thank my supervisor Dr Mohammed Seaid for his patience
and giving me the freedom to really explore this challenging topic,

Professor Roger Crouch who first led me to this area, imparted knowledge patiently
and unreservedly and provided me invaluable supervision and insight,

Professor Charles Augarde for his encouragement and help,

my husband Yang who has encouraged and supported me throughout the whole process,

my parents for fostering my love of all things science
and supporting me all the time,

my daughter who gave me infinite power to finish this thesis.

Thank you all.

Contents

List of Figures

List of Tables

Nomenclature

W	smoothing function
$\nabla \mathrm{W}$	spatial gradient of the smoothing function
W''	second derivative of the smoothing function
V	integral volume of the support domain
S	surface of the support domain
h	smoothing length of the support domain
κ	scaling factor and represents the spread of the smoothing function
κh	radius of the support domain
R	relative distance between two position comparing with the smoothing length
α_d	scaling factor to promise the unity feature of the smoothing function
∇	spatial gradient operator
i	index of a particle
j	index of neighbouring particle
N	number of particles within the support domain
ΔV	finite volume of a particle
m	mass of a particle
ρ	density of a particle
$\mathbf{x}_i$	position of a particle i
$\mathcal{O}$	truncation error term
$\widetilde{\mathrm{W}}$	correction form of the smoothing function
$\nabla \widetilde{\mathrm{W}}$	correction form of the spatial gradient of the smoothing function
$error_1$	L^1 norm error
$error_2$	L^2 norm error
$\frac{D}{Dt}$	substantial derivative
$\mathbf{v}$	velocity vector
∂	partial differential operator
$\mathbf{g}$	acceleration caused by external force
$\boldsymbol{\sigma}$	stress tensor
$\dot{\boldsymbol{\sigma}}$	rate of stress tensor
$\boldsymbol{\varepsilon}$	strain tensor
$\dot{\boldsymbol{\varepsilon}}$	rate of strain tensor
E	Young's modulus
G	shear modulus
K	elastic bulk modulus

ν	poisson's ratio
tr	trace of a matrix
$\dot{\boldsymbol{e}}$	deviatoric strain rate tensor
$\mathbf{I}$	Kroneckers delta tensor
$\hat{\dot{\boldsymbol{\sigma}}}$	Jaumann stress rate
$\boldsymbol{\omega}$	rotation tensor
Δp	particle spacing
β_g	influence factor of the ghost particle
$\beta_{\max}$	safety factor to avoid extremely high velocity
g	subscript for ghost particle
r	subscript for real particle
$\alpha\ \beta$	indices of the stress tensor
Π_{ij}	artificial viscosity between particle i and j
a_1	adjustable non-dimensional constant
c	wave speed
φ	factor to prevent numerical divergence
α_Π	constant in artificial viscosity
β_Π	constant in artificial viscosity
ϕ	bulk viscosity
r_s	distance between particle and stress point
$[\![(\cdot)]\!]$	integer part of $(\cdot)$
ψ	family of Wendland radial interpolation function
d	dimensionality
f_{ij}	effect of distance between two neighboring particles
n	exponent dependent of term f_{ij}
$\mathbf{R}$	repulsive force
θ	rotation angle of the stress tensor
$\bar{\boldsymbol{\sigma}}$	principal stress tensor
ϵ	constant parameter of repulsive force
$(\cdot)^0$	value of $(\cdot)$ at initial state
$\dfrac{d}{dt}$	ordinary differential respect with time
$\mathbf{U}$	vector of the SPH system
$\mathbf{F}(\mathbf{U})$	ordinary differential of vector $\mathbf{U}$
$\mathbf{D}$	elastic matrix
Δt	time step
g_n	global error
$\widetilde{\mathbf{v}}$	corrected velocity vector
$\mathcal{U}$	state of the $\mathbf{U}$ for a certain stage in one step
k	k^{th} stage in one time step
s	total number of stages in one time step
L_k	the slope of the k^{th} stage
a_{kl}	coefficient of time integration methods
b_l	coefficient of time integration methods
c_k	coefficient of time integration methods

ϱ	polar coordinate
ϑ	polar coordinate
ξ	reference velocity gradient
ζ	reference distance
(i, f)	index of the refinement particle which is split from coarse particle i
ϵ_r	spacing ratio of the refinement particle
α_r	smoothing ratio of the refinement particle
n_i	number of refinement particle for one coarse particle
e_i	local error of particle i after splitting
E_i	global error of particle i after splitting
e_i^{ρ}	local error of density after refinement
E_i^{ρ}	global error of density after refinement
$E_i^{\nabla\widetilde{\mathrm{W}}}$	kernel gradient error

Chapter 1

Introduction

1.1 Background

Computational simulation has become more and more popular in solving complicated problems in engineering applications and scientific research. Numerical simulation has provided an alternative means of scientific research, whether for the expensive, time consuming or dangerous experiments in laboratories or on site. For those problems that cannot be directly computed or observed to obtain specific and complete information, numerical methods give better performance and are much more practical than traditional experimental methods. The benefit brought by numerical simulation through computers is the capability of providing the verification for theory, explaining the experiment's results and new findings.

The processes of numerical simulation are similar for different numerical methods in practical applications. Mathematical models can be set up by feasible simplification and assumptions through observing physical phenomena. These mathematical models are usually represented in the form of governing equations with initial and boundary conditions. Governing equations can be a series of ordinary differential equations, partial differential equations or other forms of equations defined by well known physical laws.

The essence of numerical simulation is to solve the governing equations and obtain a solution to the physical problem from these equations. To achieve this, the problem domain should be first discretised into several components. Different numerical methods have different techniques of domain discretisation and discretised components, which can involve a set of grid-based elements or mesh-free components. Domain discretisations with a set of grid-based elements are the traditional problem domain discretisation techniques and widely applied in several numerical methods, i.e. finite element method (FEM) and finite difference method (FDM) etc. Numerical methods using domain discretisations with a set of mesh-free elements are called mesh-free numerical methods and will be introduced in Section 1.3. The grid-based elements usually contain a finite number of grid nodes and also make up the geometry of the problem domain. The variables at the locations of the grid nodes are evaluated, and each grid node is related to others by a certain kind of nodal connectivity. This nodal connectivity is the basis of generating a mesh for grid nodes in the problem domain. The accuracy of the numerical discretisation depends on the size and shape of the mesh cell.

The numerical discretisation process has provided a method of transferring the integral and derivative operations of the governing equations from a continuous form into a discretised form. This is highly related to the technique of domain discretisation. Liu pointed out that numerical discretisation is based on the function approximation theory [56]. Once the domain and numerical discretisations are performed, the governing equations are transferred to a set of algebraic equations or ordinary differential equations whose solution can be obtained through numerical methods currently available.

In the process of numerical simulation using computers, we need to convert the domain discretisation and numerical method into a computer program. The accuracy and computational cost are very important factors which need to be taken into consideration while coding, as well as the robustness (continuity and error checking) and operability (ease of reading, utilization and modification) of the program. Before performing numerical simulation, the code should be verified through experimental results, theoretical solutions, or other available numerical methods to ensure the accuracy of the code in solving practical engineering problems.

In the numerical simulation of a hydrodynamics problem, the governing equations are established through conservation laws which can be expressed using the field variables in the system. For example, mass, momentum and energy should be conserved during system evolution. The details of the conservation laws will be discussed in Chapter 3. By combining these three fundamental conservation laws with the characteristics of the material, the boundary and initial conditions and the behaviour characteristics of the fluid and solid systems can be determined. These physical conservation laws are usually represented by basic mathematical equations, partial differential equations in most cases.

With some exceptions, it is very difficult to compute the analytical solution for ordinary or partial differential equations. The approximation technique usually applied in solid mechanics is to discretise the integral or differential terms of the equations into a set of simple algebra summation formulations first and then adopt these summation formulations to spatially approximate these integral or differential terms. This approximation will generate a series of algebraic equations (or simple ordinary differential equations related only to time). By solving these equations, we will obtain the numerical value of the field function at the discretised point in the time and space frames (such as density, pressure, velocity). The classical solid mechanics numerical simulation includes the following:

- Governing equations.
- Boundary and initial conditions.
- Discretise the problem domain into a set of individual components.
- Appropriate numerical discretisation method to approximate the governing equations.
- Numerical method on solving the corresponding algebraic equations or ordinary differential equations.

1.2 Limitation of grid-based methods

Numerical methods can be classified as two different types through the different forms of the discretised components, with or without mesh (grid). Classical grid-based methods (i.e. FEM and FDM) have been widely employed and achieved success in different areas of computational solid mechanics. The discretisation techniques adopted are the main methods of domain and numerical discretisation. However, grid-based numerical methods must still confront unavoidable difficulties in some practical applications. In this section, the limitations of grid-based methods will be discussed in detail.

The primary work in grid-based methods is to discretise the problem domain and then generate the mesh. For the methods with Eulerian grids like FDM, it is very difficult to generate a regular mesh for irregular and complex domains. To construct the mesh for a complex geometry domain, additional complicated mathematical transformations are necessary and are sometimes even more complex than the problem itself. This disadvantage makes it difficult to simulate the problems with free surfaces, deformation boundaries or moving interfaces using this method. Eulerian grid-based methods are not sufficient when applied to problems which need to monitor the properties of the material in a fixed geometry domain, such as particulate flows [57]. Similarly, in the methods using a Lagrangian grid like FEM, the mesh generation work usually entails a large computational cost during the numerical simulation. Additionally, it is very important for Lagrangian grid-based methods to simulate large deformation problems which normally require special treatment such as mesh rezoning, which is very complicated and computationally time consuming [26]. This sometimes leads to inaccurate solutions.

When using grid-based numerical methods to simulate hydrodynamic problems such as high velocity impact, the disadvantages of these kind of numerical methods are especially evident. Shock wave propagation in high velocity impact problems behaves like similar phenomena in fluid dynamics when impacting or colliding with objects. Theoretically, motion and high-pressure state equations are the key features used to describe the material behavior property. Problems like moving material interfaces, large deformations, free surfaces and boundary deformations also exist in high velocity impact simulations. Evidently, they are very difficult to solve when applying grid-based numerical methods [57].

Grid-based methods also face limitations and challenges when the main simulated domain is a series of physical discretised particles rather than a continuum domain, for instance star interactions in astrophysics, atomic movements in an equilibrium or non-equilibrium state, dynamic behaviour of molecules in heat systems, etc. Continuum grid-based numerical methods have no capability to simulate these kinds of discrete systems, but mesh-free methods can.

1.3 Mesh-free methods

There is a new generation of numerical methods, mesh-free methods, which are recognized as a better choice than grid-based numerical methods like FDM and FEM in many applications. The main idea of the mesh-free method is to adopt a set of randomly distributed nodes (particles), which do not need to be connected to each other through the mesh to simulate the problem system. These can then solve all kinds of integral equations or PDEs including the boundary conditions. More accurate and stable numerical solutions can be obtained with this kind of method. Initially, the purpose of studying mesh-free methods is to amend the internal mesh structure of grid-based methods like FDM and FEM in order to make the methods more stable, adaptive and robust. Thus a lot of effort has been spent on problems that can not be solved by traditional FDM and FEM, free-surface boundary deformation, moving interface (for FDM), large deformation (FEM), and the generation of complex mesh and self adaptivity of the mesh (for FEM and FDM) problems [56]. Recently, many mesh-free methods have been applied in the solid, structure and fluid fields. These mesh-free methods have some common points, but the approximation of function and the implementation processes are different. The mesh-free methods can be classified into two types by the forms of the equations: mesh-free based on strong form equations will be introduced in Section 1.3.1 and mesh-free based on weak form equations will be introduced in Section 1.3.2. Section 1.3.3 will provide an overview the combination of mesh-free methods and traditional methods.

1.3.1 Strong form methods

In mesh-free methods based on strong forms, there is no requirement for the integral form when establishing the discretisation system, the advantages are simple implementation, high efficiency of computation and ‘real’ mesh-free, such as the type of mesh-free collocation methods. However these methods are not stable and have low accuracy in many cases, especially the problem with irregular distribution of nodes used for governing equations with Neumann boundary conditions, such as problems in solid mechanics with the stress boundary conditions.

Of all mesh-free methods, smoothed particle hydrodynamics (SPH) has proved to be one of the most popular. Standard SPH is a 'real' mesh-free method which was first invented to simulate astrophysical phenomena by Lucy, Gingold and Monaghan in 1977 [67, 30], and then widely applied in continuous solid and fluid mechanics. The SPH method and its different variants are among the most popular particle methods in engineering, and have been incorporated into commercial software [53, 88, 110, 111, 69, 100, 87]. The reproducing kernel particle method (RKPM) proposed by Liu and Chen is based on the study of the consistency and reproducing ability of the SPH method [63]. This approach showed better accuracy of the SPH approximation, especially on the boundary. For details of RKPM and its application, see the works of Liu *et al.*, Li and Liu [64, 50].

1.3.2 Weak form methods

Weak form methods such as the element free Galerkin method (EFGM), meshless local Petrov-Galerkin method (MLPGM), point interpolation method (PIM) and material point method (MPM) have better stability and accuracy. The problems with Neumann boundary conditions can be successfully solved by the employment of weak form equations including smooth (integral) operators. However, weak form methods do not represent the real meaning of mesh-free, because they require a background mesh for the integration during the process. Therefore, most of the weak form methods are also called meshless methods, such as EFGM, MPM and MLPGM.

Harlow proposed the particle-in-cell method (PIC) to solve the solid and fluid mechanics in 1957 [38]. Sulsky, Chen and Schreyer develop the material point method (MPM) based on the combination of PIC and FEM in solid mechanics [104]. In the MPM method, the points carry the material information including density, velocity and stress etc. and then the information is mapped to the background mesh to update for next time step. Then the MPM method is generalized using a PetrovGalerkin discretization scheme by Bardenhagen and Kober and they named this new method the generalized interpolation material point (GIMP) method [6]. Liu *et al.* invented the Points Interpolation Method (PIM). They focused on solving the singularity of the polynomial PIM method, and propose many different ways of finding a solution [56]. With the employment of the radial primary function (or combined with the polynomial), the problems of local Petrov-Galerkin weak form and Global Galerkin weak form are both solved [54, 116]. Mesh-free methods are also used to develop boundary integral methods, and as a result, to propose boundary mesh-free methods. In this kind of method, only the boundary of the domain is arranged with particles. Mukherjee and Mukherjee formulated an equation which is based on the employment of the element free Galerkin equation by using moving least square approximation [83, 84]. Gu and Liu proposed the Boundary Point Interpolation Method (BPIM) by applying the Polynomial Point Interpolation Method (PIM) and the Radial Point Interpolation Method. They obtained a set of small discretisation system equations from the property of the Dirac function of PIM shape functions [35] .

The Diffuse Element Method (DEM) is one of the weak form methods developed by Nayroles *et al.*, which introduces the moving least square approximations to the Galerkin method [85]. On the basis of the DEM, Belytschko *et al.* proposed the element free Galerkin method (EFGM), which is now one of the most widely used mesh-free methods [7]. This method can be applied to many solid mechanics problems by the employment of the background mesh through integration. Atluri and Zhu (1998) proposed the Meshless Local Petrov-Galerkin method (MLPG), which computes the integral for the local background mesh [4]. Since the MLPG method does not compute the integral for the global background mesh, it is widely used in the analysis of structure beams and plates, fluid flow problems and many other mechanical problems [3]. These two methods basically transformed the original problem into a local weak formulation and the shape functions were constructed from using the moving least-squares approximation to interpolate the solution variables.

.

1.3.3 Combination of different methods

In practical applications, one mesh-free method can be used in combination with other mesh-free methods or other traditional grid-based methods to obtain the advantages of different methods. Examples include the combination of SPH and FEM, the element free Galerkin method coupled with the boundary element method (BEM) and the meshless local petrov-Galerkin method can also be applied in combination with BEM or FEM [57]. EFGM can also be coupled with Boundary Point Interpolation Method in solid mechanics [36]. Liu and Gu have proposed the Mesh-Free Weak-Strong method (MWS) based on strong form and local weak form [55]. In this method, the strong form equation is applied on the internal points and the points on the essential boundaries. However, the local weak form (Petrov-Galerkin weak form) equation is only applied on the points near the natural boundaries. Therefore, there is no need to do the numerical integrals for all the internal nodes and the nodes on the essential boundary. The numerical integral operation is only carried out for those nodes on the natural boundary. In other words, this method only needs to arrange the local background mesh for the nodes near the natural boundary. The shape function of the MWS method is set up by local radial point interpolation and moving least square approximations. The final system matrices are sparse and band matrices, which will improve the computational efficiency. The MWS method preserves the stable solution even with the irregularly distributed nodes while it only has a very coarse mesh in the system in the whole computational process. This makes the method approach the condition fulfillment of a 'real' mesh-free method, which obtains stable and accurate results by using irregular distributed nodes for solid mechanics problems.

Most research on mesh-free methods focuses on the complex application of computational solid mechanics and computational fluid dynamics. Since the analysis in mesh-free methods is made of a number of randomly distributed nodes, rather than fixed nodes in mesh-based systems, it is useful to apply mesh-free methods to problems that traditional grid-based methods cannot solve, such as large deformation problems in solid mechanics, oscillation analysis, nonlinear foundation consolidation problems and incompressible flows, etc. The application of the shock wave propagation to impact problems and large deformation problems will be analysed in following chapters.

In this thesis, the discussion will focus on one of the earliest mesh-free methods, the SPH method, which derives from an integral representation to approximate the field function. The SPH method is very similar to the mesh-free methods based on weak form equations [56]. The difference is that the weak form operation of the SPH method is carried out in the function approximation process but in ordinary weak form methods like EFG, MLPG, RKPIM, PIM etc., the weak form operation is implemented in the generation process of a discretised system. As the integral representation is applied on a field function, the differential operators of a field function will be passed to the smoothing function (weight function). This reduces the requirement of continuous order to the approximation of a field function. The SPH method has determined that stable numerical results can be obtained by using even randomly distributed points for many large deformation problems [58]. The accuracy of the SPH method is dependent on the selection of the smoothing function to a great degree, which will be discussed in Section 2.2.

1.4 Smoothed Particle Hydrodynamics

1.4.1 Standard SPH method

In the SPH method, the problem system is represented by using a number of particles which have the individual properties of material and the behaviour of each particle follows the principle of conservation governing equations. The SPH method was developed to solve the astrophysical problem in three-dimensional open space [67, 30]. It has been widely investigated and expanded, and then applied to the solid dynamics response problem and fluid dynamics with large deformation.

As a Lagrangian mesh-free method, the SPH method has specific characteristics. Comparing with other traditional numerical methods based on meshes, it has some special advantages, the most outstanding is the self-adaptive character of the SPH method. The self-adaptive character can be obtained in the early stage of field function approximations, which is carried out on the current randomly distributed particles in each time step. Due to this special property of the SPH method, the construction of the SPH equations is not affected by the random distribution of particles. In other words, this method can be applied to solve an extremely large deformation problem. This makes the SPH method both attractive and powerful.

The mesh-free properties of the SPH method can be attributed to the following three factors: firstly, the adaptive character as mentioned above; secondly, the problem domain is represented by particles; finally, the particles are used as the computational frame of the approximations for the field variables. In the process of computation, there is no requirement on a predefined mesh for the SPH approximation to provide connection information between particles. The mesh-free property shows its usefulness especially for those problem that are hard to solve by using traditional FEM and FDM.

As mentioned previously, the SPH method was developed to simulate nonaxisymmetric 3-D phenomena in astrophysics. Because the movement of these points is similar to the movement of fluid or gas, we can use classical Newton conservation laws in this method to solve the dynamics problems.

Apart from the mesh-free and self-adaptive characteristics, another attractive characteristic is the combination of Lagrangian equations and the particle approximation. The SPH particles represent the field variable at the position of the problem domain that makes the method more powerful. The difference between the SPH method and the other mesh-less methods such as PIC, MPM and GIMP methods, lies on the fact that the SPH method does not need the background mesh to calculate spatial derivatives as in the above mentioned methods. On the other hand, the implementation of the SPH method is far simple than other mesh-less methods. It should also be mentioned that it would be difficult for other mesh-less methods using background mesh such as MPM to solve the large deformation problems. This is mainly because of the errors caused by the mesh mapping process when the material points move across several mesh grids in one step. It is necessary for these mesh-less methods to include special treatments and complex correction techniques on the mapping process to solve large deformation problems. However, the implementation process of the SPH method is much simple as the particles of the SPH method carry all the information required for computations. They can move in space and construct the computation frame for solving the partial differential equations based on classical conservation laws without relying on background meshes. This feature makes the SPH method attractive and powerful to solve the large deformation problems.

The core ideas of the SPH method are described by three words: smoothed, particles, hydrodynamics. The first word, smoothed, means to obtain the stable smoothing approximation by calculating the weighted average of the neighbouring particles. The second word, particles, represents that the property of the basic computational frame in this method is a kind of particle method. The third word, hydrodynamics, refers to this method's application to hydrodynamical problems. The combination of self-adaptive, particle and Lagrangian characteristics make the SPH method popular in the field of engineering applications.

As mentioned before, since the SPH method is very powerful in introducing the complex physical effects into the SPH equations, the SPH method is widely applied in computational mechanics. When SPH approximations are applied to construct the point-dependent shape function, they can be employed to other areas of mechanics not only classic hydrodynamics. Therefore, Kum *et al.* [46] and Posch *et al.* [90] also called the SPH method smoothed particle mechanics.

The early applications of the SPH method were mainly in the field related to fluid dynamics, such as elastic flow, quasi-incompressible flows, gravity currents, flow through porous media, heat conduction, shock simulations, heat transfer and mass flow, etc. [76]. A very important application area is the high velocity impact (HVI) problem caused by high speed particles colliding with space equipment (satellite, space shuttle, space station). In HVI problems, when the shock wave propagates through the body the physical behaviour resembles that of fluid [119, 120]. Libersky, Randles and their collaborators have contributed to the application of the SPH method to impact problems [52, 93, 92]. Another noteworthy application is the explosion phenomena caused by high-efficiency explosives. Swegle and Attaway have studied the feasibility of simulating underwater explosions with the SPH method [106]. Liu *et al.* simulated a series of explosion phenomena with the SPH method, including the explosion of high-efficiency explosives, underwater shock and the remittance of underwater shock [60, 62, 61].

The SPH method has had many improvements and advances during the application process. Numerical computation has been improved, some internal weaknesses of the SPH method are pointed out as well as the corrective technique and improvement method. Swegle *et al.* pointed out the problem of tensile instability for a material with strength [105]; Morris has solved the inconsistency problem of particles which leads to the lower accuracy to the SPH method [81]. Monaghan has introduced a symmetrised equation with better performances [78] for the method. Liu and Chen have proposed the Reproducing Kernel Particle Method (RKPM) which can bring higher accuracy in particle approximation [63]. Chen *et al.* developed a Corrective Smoothed Particle Method (CSPM), which brings better accuracy for both inside the problem domain and near boundary areas [16, 17]. Other famous corrected and revised SPH methods include: moving least square particle hydrodynamics (MLSPH) by Dilts [21, 22] and corrected integration kernel by Bonet and Kulasegaram [11]. An open SPH code, called SPHysics, was developed by Gomez-Gesteira *et al.* [32] in 2007 and improved into several versions in recent years. Rogers *et al.* adopted the open-source code

SPHysics combining with a Riemann solver-based formulation to simulate the caisson breakwater movement [97]. Currently, SPHysics code can simulate general dynamics problems very well [53, 88, 110, 111, 69, 100, 87].

Libersky and Petschek extended the SPH method to solid Mechanics [51, 52]. This method was then applied to simulate the fracture of brittle solids by Benz and Asphaug [9] and metal forming by Bonet and Kulasegaram [11]. In the application of simulating large deformation and impulsive loading, the SPH method is powerful and attractive. In recent years, some correction and improvement has been developed to restore the consistency of the method and enhance its accuracy. These modifications have generated many versions of the SPH method and corresponding equations. Randles and Libersky [94] have applied the stress point method, which was developed by Dyka and Ingel [24], to multi-dimensional space problems; this improved the tensile instability problem and zero energy mode problem.

Although the SPH method has been applied to different fields, there are still many problems which need further investigation, especially this method's numerical analysis with its mesh-free nature; the techniques that were developed from grid-based Euler or Lagrangian methods are not suitable to be directly applied in the SPH method.

SPH is a method ideally suited for dynamics with large nonlinear deformation, the SPH system is to be solved for a time interval endowed with given initial properties and boundary conditions. The information on each particle in the problem domain is updated by time integration. To simulate the time integration, the time interval is discretised into several steps. Explicit time stepping schemes can be chosen for ordinary partial equations (ODEs) in the SPH method [58]. It should be stressed that, because explicit time stepping schemes evaluate explicitly the right-hand side of the ODEs, then it has to satisfy a stability condition. This stability criterion can be based on the Courant-Friedrichs-Levy (CFL) condition. This leads to the small size of the time step in the SPH method and limits the efficiency of the method: the large number of time steps will lead to error accumulation, and the accuracy of the SPH method will also be affected.

1.4.2 Adaptive SPH method

In some real engineering applications, only part of the problem domain has dynamic behaviour (i.e. change of velocity, stress or temperature) during a certain time. For example, consider a beam in the shock wave propagation problem, which is fixed at the right end, the loading velocity is at the left end of the beam. The right part of the beam will not receive the dynamic information until the propagation of the shock wave reaches the right end. In this situation, the properties of the particles in the right part of the beam are not the feature of interest but will still need to be calculated, this is time consuming and limits the efficiency of the SPH method.

For this reason, a new version of the SPH method with a time-varying particle distribution, called the Adaptive SPH method (ASPH), is attractive. Kitsionas and Whitworth first implement a particle splitting method to the classical SPH method for the astrophysics problem in 2002 [44]. Due to the adaptivity property of the particle distribution in the ASPH method, the accuracy of the method is increased using a relatively smaller number of particles compared to the standard SPH method. Then Lopez applied the Adaptive Particle Splitting (APS) technique, which is the basic concept of Adaptive Mesh Refinement (AMR) being adopted on the particle methods, for the SPH method in fluid flow simulations [65]. Although the APS technique is not as mature as the AMR technique, it is still worth studied, improved and applied to the SPH method to obtain the new Adaptive SPH algorithm.

Omidvar *et al.* applied variable mass particle distribution to simulate 2D and 3D fluid dynamic problems [87, 88] through the SPH method and Lastiwka *et al.* presented a more general algorithm with particle insertion and removal for the SPH method based on this [48]. Feldman and Bonet developed a dynamic particle refinement algorithm for the SPH method [27]. Then Vacondio *et al.* expanded the method with splitting and coalescing techniques on fluid dynamics [108, 109, 110]. Spreng *et al.* then applied the adaptive discretization algorithm for the SPH method on solid mechanics [101]. Lopez *et al.* expanded the application of the particle refinement in SPH from fluid to non-cohesive soil model. [66].

A new refinement procedure is developed by Barcarolo *et al.* [5]. In their work, the mother particle is not removed but turned off by an operator and daughter particles are turned on when they are created. One of the key points in the particle refinement technique is the varying smoothing length, which can then be adopted as the adaptive smoothing length for this new algorithm, unlike the method proposed by Shapiro *et al.* [98] and Owen *et al.* [89], which only focuses on the adaptive kernel estimation. The adaptive SPH algorithm studied by the above researchers includes not only the adaptive kernel estimation but also the adaptive number and distribution of particles during the time integration process. Since the size of time step depends on the particle spacing, the time step will become smaller after the particle splitting. A smaller time step should be chosen for all particles after the splitting process in previous ASPH simulations [5]. In this thesis, the work will focus on improve the efficiency and stability of the standard SPH and ASPH methods.

1.5 Purpose of the thesis

There is significant interest in the SPH method and its application in solid mechanics. Comparison with traditional grid-based methods (FEM and FDM, etc.) shows the SPH method to have these benefits:

- The SPH method can perform better solution in several application areas than using some traditional grid-based methods, such as FEM and FDM. For example Jankowiak and Lodygowski compared the SPH and FEM simulations on the blast impact problem and obtain the conclusion that SPH is more suitable for this problem [42]. Bui *et al.* investigated the advantages of SPH for simulating the discontinuous soil failure problems comparing with the FEM and LEMs methods [13].

- The SPH method can be widely applied in several areas, from micro-scale to astronomical problems. It can solve both discretised and continuum systems.

- The SPH method is one of the oldest mesh-free particle methods and approaching maturity. A variety of modifications and corrections have been developed to improve the performance of the method in the engineering practical applications.

However, there are still limitations to the method during engineering practical application and science research, as mentioned in the last section. The limitations which will be solved for the SPH method in this thesis can be summarised as following:

- The explicit SPH method is limited by the CFL condition. The small size of the time step will influence the efficiency and accuracy of this method.

- Increasing the number of particles can result in higher accuracy of the method but lower efficiency, since the entire domain may not need a uniform distribution of particles to simulate the physical behaviour.

The emphasis of this thesis will be on the improvement and modification of the SPH method to solve the problems presented above. A combination of the Runge-Kutta Chebyshev time stepping scheme and the SPH method will be introduced in this thesis to increase the size of the time step and also demonstrate the accuracy of this method. A new form of the ASPH method will be developed for the application of elastodynamics to improve the efficiency of the method and save computational cost during the analysis. Some applications, which are difficult for FEM and FDM, will also be presented in detail, such as the shock wave problem and large deformation problem.

The basics of the SPH method will be introduced in Chapter 2. The method of kernel approximation and the particle discretisation will be discussed in this chapter. Different smoothing functions will be introduced for different usages. The treatment for the boundary accuracy problem will also be solved by using a correction form for the SPH approximation of the system properties and their spatial derivatives. Three different neighbouring particle searching methods are discussed.

In Chapter 3, the discussion will focus on applying the SPH approximation concept to the dynamics problem. First both Eulerian and Lagrangian approaches are introduced. The SPH method belongs to the Lagrangian numerical method. The governing equations are derived for the dynamics problems using the classical conservation laws. The stress tensor calculation is derived through Hooke's law, then the governing equations and the stress tensor for each particle are represented in the SPH approximation form. In addition, the special treatments for the SPH method to improve its performance will also be discussed in this chapter, such as boundary condition treatment, artificial viscosity and a treatment for tensile instability. The SPH system with the special treatments will be summarised at the end of Chapter 3.

One of the key improvements for the application of the SPH method is discussed in Chapter 4, the Runge-Kutta Chebyshev time stepping scheme. The importance of the time stepping scheme for the numerical methods will be presented. The classical Euler and Runge-Kutta family time integration algorithms are discussed here, including the error analysis for each scheme. Finally, the Runge-Kutta Chebyshev (RKC) time integration algorithm, which is developed by Verwer [112], will be proposed for the SPH method to improve its performance.

The application of the SPH method combined with the RKC time integration scheme will focus on the shock wave propagation problem and large deformation problems in elastodynamics, which are challenging for traditional numerical methods, such as FEM and FDM. These applications will be presented in Chapter 5.

A new algorithm of the SPH method will be proposed in Chapter 6, which is referred to as time-space adaptive smoothed particle hydrodynamics in this thesis. The smoothing length in the time-space ASPH method is varying with the adaptive particle spacing. The calculation of the SPH approximation with varying smoothing length is first raised by Monaghan [71, 73]. Omidvar *et al.* also applied variable mass particle distribution to simulate 2D and 3D fluid dynamic problems [87, 88]. However Monaghan and Omidvar *et al.* only considered the situation of disorder distribution particles caused by partly compression or tension in fluid dynamics with all initial uniform particle distribution. Then Vacondio *et al.* expanded the method with splitting and coalescing techniques on fluid dynamics [110]. In this thesis, the time-space ASPH is developed in which the distribution of the particles and the stages in one step is adjusted during the time integration. The techniques of particles splitting and merging will be introduced and then the algorithm is applied to elastodynamics. The details about this new algorithm will be shown in Chapter 6.

Chapter 7 summarises the contribution of this thesis: the combination of the RKC method to improve the stability and efficiency of the SPH method, proposing a new time-space ASPH algorithm and its applications. The potential future work of the method is then explored along with suggestions of where further modifications might be made to improve the method in engineering practical applications.

Chapter 2

Smoothed particle hydrodynamics

Introduction

The Smoothed Particle Hydrodynamics (SPH) method was first developed by Lucy, and Gingold and Monaghan [67, 30]. In this method, the continuum domain is discretised into particles which carry the field variables. These variables are calculated from the contribution of the neighbouring particles by means of a kernel function. The SPH method is a true mesh-free method based on the transformation of differential equations into integral ones which are then discretised using a distribution of moving particles. It is traditionally applied to modeling fluid flows. In recent years, there has been a growing interest in applying the SPH method to solid mechanics problems. The main feature of the SPH method is that it is a particle-based technique and does not require any underlying grid structure to represent the problem geometry. This feature makes the SPH avoid the disadvantage ideally associated with traditional mesh-based methods methods (FEM, FVM and BEM), for example, maintaining mesh integrity and quality under large deformation. The mesh-free nature of the SPH method makes it ideally suit to model processes that involve large deformations and discontinuities, fracture and fragmentation, metal forming, etc. It produces good results in many applications in both fluid and solid mechanics.

The primary concept and the essential formulations of the SPH method are discussed to understand the various ingredients of the SPH method. The SPH method is developed to simulate the hydrodynamic problems with a set of partial differential equations (PDE). Normally, it is difficult to get an exact solution of the PDEs except for some simple cases. Therefore, the basic SPH concept for the hydrodynamic problem is to first discretise the continuous problem domain defined by the PDE into particles and then approximate the solution function and its derivatives for each particle. After the function approximation, the PDE can be transformed into a set of ordinary differential equations (ODE) which are solved by each individual particle.

The SPH algorithm can be divided into two steps: (i) representing the field variables and their derivatives in an integration form by using smoothing functions, (ii) discretising the continuous problem domain into a set of particles, which carry physical properties, i.e. mass, volume and density.

In this chapter, these two main steps of the SPH method will be introduced in Section 2.1 and Section 2.3, including the error order analysis of the approximation. The boundary accuracy problem as well as the solution to the problem will be discussed in Section 2.4. Furthermore, three common neighbouring particle searching algorithms are applied for the SPH method. The SPH method combined with PDEs for the dynamics is further discussed in Chapter 3.

2.1 Kernel approximation

2.1.1 Kernel approximation of a function

In the SPH method, a kernel approximation is applied to a function and its spatial derivatives by using smoothing functions (also called weight functions). The kernel approximation can be derived by beginning with the Dirac function,

$$f(\mathbf{x}) = \int_{\mathrm{V}} f(\mathbf{x}')\delta(\mathbf{x}-\mathbf{x}')d\mathbf{x}', \tag{2.1}$$

where f is a spatially dependent function with respect to $\mathbf{x}$ and $\delta(\mathbf{x}-\mathbf{x}')$ is the Dirac Delta function. $\mathbf{x}'$ and $\mathbf{x}$ indicate points with respect to the integral volume V.

$$\delta(\mathbf{x}-\mathbf{x}') = \begin{cases} 1, & \mathbf{x}=\mathbf{x}', \\ 0, & \mathbf{x}\neq\mathbf{x}'. \end{cases} \tag{2.2}$$

Equation (2.1) indicates that a certain function could be rewritten into integral form by using the Dirac delta function, under the condition that $f(\mathbf{x})$ is continuous in the integral volume V [58].

Since the Dirac delta function has only one point in the domain, it cannot be applied to construct discrete numerical models. However, if the point is enlarged to be an area of radius κh, (where h is the smoothing length discussed later and κ is the number of smoothing lengths that defines the radius of compact support), we can assume the total integral of the points in this finite area is 1. Then the kernel approximation of a function $f(\mathbf{x})$ can be expressed by replacing the Delta function with a smoothing function $\mathrm{W}(\mathbf{x}-\mathbf{x}',h)$,

$$< f(\mathbf{x}) >= \int_{\mathrm{V}} f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}'. \tag{2.3}$$

It can be noted that $\mathrm{W}(\mathbf{x}-\mathbf{x}', h)$ is also known as the smoothing function and κh is the radius of the influence area. The Dirac function can be considered as a special case of this function. Here $< f(\mathbf{x}) >$ indicates an approximation to $f(\mathbf{x})$ and $\mathbf{x}'$ is a random point in the integral region. Since the smoothing function is not exactly a Dirac function, the integral form of the function $f(\mathbf{x})$ can only be an approximation, except for some special cases. It is worth noting that h is the smoothing length, which defines the size of the influence domain of the smoothing function and κ is the number of smoothing lengths in the influence domain (the factor κ is fixed for one particular smoothing function). The smoothing length is empirically chosen from 0.8 to 1.8 times of the initial particle spacing for different application problems [58]. El-Gammal *et al.* [26] also investigated the influence of smoothing length in 2013.

The kernel approximation is proven to have a second-order accuracy [73, 76]. This can be simply derived by applying the Taylor series to Equation (2.3) shown as (2.4). There are several special properties of smoothing functions which will be discussed in detail in Section 2.2. The smoothing function has the normalization condition ($\int \mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = 1$) and is an even function. The support domain of the smoothing function is $|\mathbf{x}'-\mathbf{x}| \leq \kappa h$ (κ is the scaling factor) and the value should be zero when $|\mathbf{x}'-\mathbf{x}| > \kappa h$. This property is named the compact support condition. Therefore, the integration domain V is the same as the support domain. The errors of kernel approximation can be identified by applying a Taylor series to the function $f(\mathbf{x}')$ around the point $\mathbf{x}' = \mathbf{x}$. This then gives us,

$$\begin{aligned} < f(\mathbf{x}) > &= \int_{\mathrm{V}} \left(f(\mathbf{x}) + f'(\mathbf{x})(\mathbf{x}'-\mathbf{x}) + O((\mathbf{x}'-\mathbf{x})^2)\right) \mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' \qquad (2.4) \\ &= f(\mathbf{x}) \int_{\mathrm{V}} \mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' + f'(\mathbf{x}) \int_{\mathrm{V}} (\mathbf{x}'-\mathbf{x})\mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' + O(h^2), \end{aligned}$$

where $O((\mathbf{x}'-\mathbf{x})^2)$ represents the residual terms after the second-order derivative. Since $\mathrm{W}(\mathbf{x}-\mathbf{x}', h)$ is an even function, the term $(\mathbf{x}'-\mathbf{x})\mathrm{W}(\mathbf{x}-\mathbf{x}', h)$ should be an odd function. Hence,

$$\int_{\mathrm{V}} (\mathbf{x}'-\mathbf{x})\mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = 0. \qquad (2.5)$$

Combining Equations (2.4), (2.5) and the property of the smoothing function, we find that the kernel approximation of a function has a second-order accuracy, i.e.

$$< f(\mathbf{x}) >= f(\mathbf{x}) + O(h^2). \qquad (2.6)$$

Normally, for a small value of h, the term $O(h^2)$ can be ignored, giving,

$$f(\mathbf{x}) = \int_{\mathrm{V}} f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}'. \qquad (2.7)$$

2.1.2 Kernel approximation of the derivatives

The kernel approximation of a function's derivative can be obtained by replacing $f(\mathbf{x})$ with $\nabla\big(f(\mathbf{x})\big)$ in equation (2.7) (where $\nabla = [\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \quad \frac{\partial}{\partial z}]$),

$$< \nabla\big(f(\mathbf{x})\big) >= \int_{\mathrm{V}} \nabla f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}', \tag{2.8}$$

where the term on the right hand side $\Big(\nabla\big(f(\mathbf{x}')\big)\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\Big)$ is found using the product rule,

$$\nabla\Big(f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\Big) = \nabla\big(f(\mathbf{x}')\big)\mathrm{W}(\mathbf{x}-\mathbf{x}', h) + f(\mathbf{x}')\nabla\big(\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\big). \tag{2.9a}$$

Then we find,

$$\nabla\big(f(\mathbf{x}')\big)\mathrm{W}(\mathbf{x}-\mathbf{x}', h) = \nabla\Big(f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\Big) - f(\mathbf{x}')\nabla\big(\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\big). \tag{2.9b}$$

Combining Equations (2.8) and (2.9), a function derivative kernel approximation can be represented as,

$$< \nabla\big(f(\mathbf{x})\big) >= \int_{\mathrm{V}} \nabla\Big(f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\Big)d\mathbf{x}' - \int_{\mathrm{V}} f(\mathbf{x}')\nabla\big(\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\big)d\mathbf{x}'. \tag{2.10}$$

The first term on the right-hand side in equation (2.10) can be rewritten into an integration on the surface S of the original domain V by using the Divergence Theorem. Then Equation (2.10) becomes,

$$< \nabla\big(f(\mathbf{x})\big) >= \oint_{\mathrm{S}} \Big(f(\mathbf{x}')\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\Big) \cdot \vec{\mathbf{n}}dS - \int_{\mathrm{V}} f(\mathbf{x}')\nabla\big(\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\big)d\mathbf{x}', \tag{2.11}$$

where $\vec{\mathbf{n}}$ is the unit vector normal to the surface S. Since the smoothing function includes the compact condition and is continuous in the domain, the value of the smoothing function on the surface of the domain is zero. Hence, the first integral term of (2.11) is zero, and

$$< \nabla\big(f(\mathbf{x})\big) >= -\int_{\mathrm{V}} f(\mathbf{x}')\nabla\big(\mathrm{W}(\mathbf{x}-\mathbf{x}', h)\big)d\mathbf{x}'. \tag{2.12}$$

Here the differential operation can be seen to be transferred from the function to the smoothing function in the kernel approximation. In other words, a function spatial gradient kernel approximation can be represented by the integration of the product of the function's value and the smoothing function spatial derivative.

Similar to determining the accuracy of the kernel approximation of a function, the accuracy of the kernel approximation of its derivative can be determined by using the Taylor series expansion on $\nabla\big(f(\mathbf{x}')\big)$ around $\mathbf{x}' = \mathbf{x}$,

$$\begin{aligned} < \nabla\big(f(\mathbf{x})\big) > &= \int_{\mathrm{V}} \Big(\nabla\big(f(\mathbf{x})\big) + \nabla\big(f(\mathbf{x})\big)'(\mathbf{x}' - \mathbf{x}) + O((\mathbf{x}' - \mathbf{x})^2)\Big) \mathrm{W}(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \qquad (2.13) \\ &= \nabla\big(f(\mathbf{x})\big) \int_{\mathrm{V}} \mathrm{W}(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + \nabla\big(f(\mathbf{x})\big)' \int_{\mathrm{V}} (\mathbf{x}' - \mathbf{x}) \mathrm{W}(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + O(h^2). \end{aligned}$$

Since $\mathrm{W}(\mathbf{x} - \mathbf{x}', h)$ is an even function, $(\mathbf{x}' - \mathbf{x})\mathrm{W}(\mathbf{x} - \mathbf{x}', h)$ will be an odd function. Hence the integration of $(\mathbf{x}' - \mathbf{x})\mathrm{W}(\mathbf{x} - \mathbf{x}', h)$ with its complete support domain will be zero. Combining with $\int_{\mathrm{V}} \mathrm{W}(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1$ it can be seen that the kernel approximation of a function's derivative has a second-order accuracy, i.e.

$$< \nabla\big(f(\mathbf{x})\big) >= \nabla\big(f(\mathbf{x})\big) + O(h^2). \qquad (2.14)$$

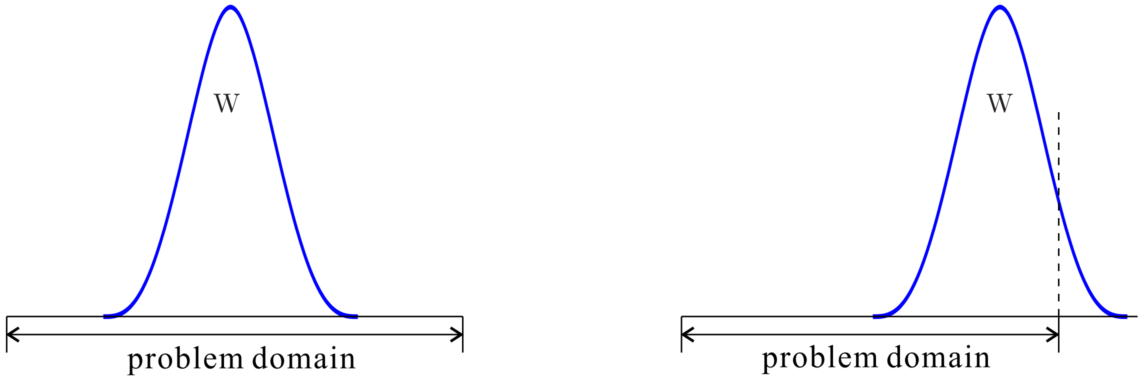
Once again the term $O(h^2)$ can be ignored, and combining (2.12) and (2.14) we find,

$$\nabla\big(f(\mathbf{x})\big) = -\int_{\mathrm{V}} f(\mathbf{x}') \nabla\big(\mathrm{W}(\mathbf{x} - \mathbf{x}', h)\big) d\mathbf{x}'. \qquad (2.15)$$

Therefore the kernel approximation of higher order derivatives can be achieved by recursively using (2.15) with regards to the first-order derivative of its next lowest order derivative. Since a function's kernel approximation and its derivatives all have second order accuracy, the SPH method is referred to as a second-order accurate method. Note that (2.6) and (2.14) are true only when the problem domain V contains the support domain of the smoothing function ($|\mathbf{x}' - \mathbf{x}| \leq \kappa h$) as shown in Figure 2.1(a). However, it is possible that the computational domain intersects the support domain of the smoothing function as shown in Figure 2.1(b), where the support domain is cut by the boundary of the problem domain. In this case, the integration of the smoothing function with respect to $\mathbf{x}$ will not be unity and Equation (2.5) will not be satisfied. The SPH kernel approximation does not necessarily have second-order accuracy in this case. This issue is called boundary accuracy and will be discussed in more detail in Section 2.4.

2.2 Smoothing functions

Effective approximation is a major issue of mesh-free methods. The SPH method adopts the weighting function (also called smoothing function) to establish the kernel approximation. The smoothing function not only represents particle interpolation for the research object but also defines how a particle influences others and gives the influencing area (support area or domain) of the particle. The choice of the smoothing function in the SPH method strongly affects the accuracy of the approximation. Many



(a) (b)

Figure 2.1: (a) Computational domain contains the support domain of the smoothing function. (b) Computational domain intersects with the support domain of the smoothing function.

researchers have studied the properties and requirements of smoothing functions to improve the performance of the SPH method. Different smoothing functions have been adopted in the SPH method in the literature [28, 29, 80, 81]. Various properties and requirements for smoothing functions have been discussed in different situations in these papers. However, all smoothing functions should satisfy some basic requirements:

1. The smoothing function should be normalised over its support area, which means that the integration of a smoothing function over the support domain is *Unity* i.e.,
$$\int_{\mathrm{V}} \mathrm{W}(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' = 1. \tag{2.16}$$
2. Compact condition: The weight function should only be supported in a reduced 'support domain',
$$\mathrm{W}(\mathbf{x}-\mathbf{x}') = 0 \quad \text{when} \quad |\mathbf{x}-\mathbf{x}'| > \kappa h, \tag{2.17}$$
where h is the smoothing length, κ is a scaling factor and represents the spread of the smoothing function. $|\mathbf{x}-\mathbf{x}'| \leq \kappa h$ indicates the dimensions of the influence area at point $\mathbf{x}$. This property transfers the global problem domain into the local support domain of the smoothing function. This property is also called *compact support.*
3. Values of the smoothing function in the support domain at any point $\mathbf{x}$ should be positive (i.e. $\mathrm{W}(\mathbf{x}-\mathbf{x}') \geq 0$). It is not a necessary requirement mathematically for convergence but is important to ensure the physical meaning for some physical phenomena. In SPH simulations of real hydrodynamics problems, negative values of smoothing functions would lead to problematic consequences, for example negative density or energy. This property is called *positivity.*

4. The value of the smoothing function at particle $x = x'$ should be at the peak point. The field value of particles in the support domain should have decreased with the distance away from the associated particle. In other words, particles near the associated particle will have more influence than those far away from the considered particle. This property is called *decay*.

5. The smoothing function should satisfy the Dirac Delta function property when the smoothing length approaches zero,

$$\lim_{h\to 0} \mathrm{W}(\mathbf{x}-\mathbf{x}', h) = \delta(\mathbf{x}-\mathbf{x}'). \tag{2.18}$$

   This delivers the real function value when the smoothing length tends to zero ($< f(\mathbf{x}) >= f(\mathbf{x})$ when $h \to 0$), and is called the *Delta function property*.

6. The smoothing function should be symmetric, which means that it has to be an even function. In other words, particles at the same distance but different positions have the same influence on the associated particle. This property is called the *symmetric property*.

7. The smoothing function should be sufficiently smooth. For the kernel approximation of a function and its derivative, a smoother weight function can bring higher accuracy to the approximation and better numerical stability. When the smoothing function is smoother, it leaves less room for the errors of a kernel approximation caused by irregularly spaced particles. This property is called *smoothness*.

Any function satisfying the requirements above can be regarded as a smoothing function. Several commonly used smoothing functions are presented below. These smoothing functions and their spatial derivatives are shown in Figure 2.2 and Figure 2.3.

The smoothing function adopted by Lucy in the original SPH paper [67] is the Bell-shaped function,

$$\mathrm{W}(R,h) = \alpha_d \begin{cases} (1+3R)(1-R)^3 & R \le 1, \\ 0 & R > 1, \end{cases} \tag{2.19}$$

where R is the relative distance between two particles at points $\mathbf{x}$ and $\mathbf{x}'$, $R = \frac{r}{h} = \frac{|\mathbf{x}-\mathbf{x}'|}{h}$. α_d equals $\frac{5}{4h}, \frac{5}{\pi h^2}$ and $\frac{105}{16\pi h^3}$ respectively in one-, two- and three-dimensional space. In Figure 2.2, the one-dimensional case is shown for $h = 1$. For two- and three-dimensional problems, $(\mathbf{x}-\mathbf{x}')$ becomes the L_2-norm of the vector difference between the two locations.

The Gaussian function has also been selected as a smoothing function by Gingold and Monaghan [30] in Figure 2.2. The authors first applied this smoothing function to simulate non-spherical stars,

$$\mathrm{W}(R,h) = \alpha_d e^{-R^2}, \tag{2.20}$$

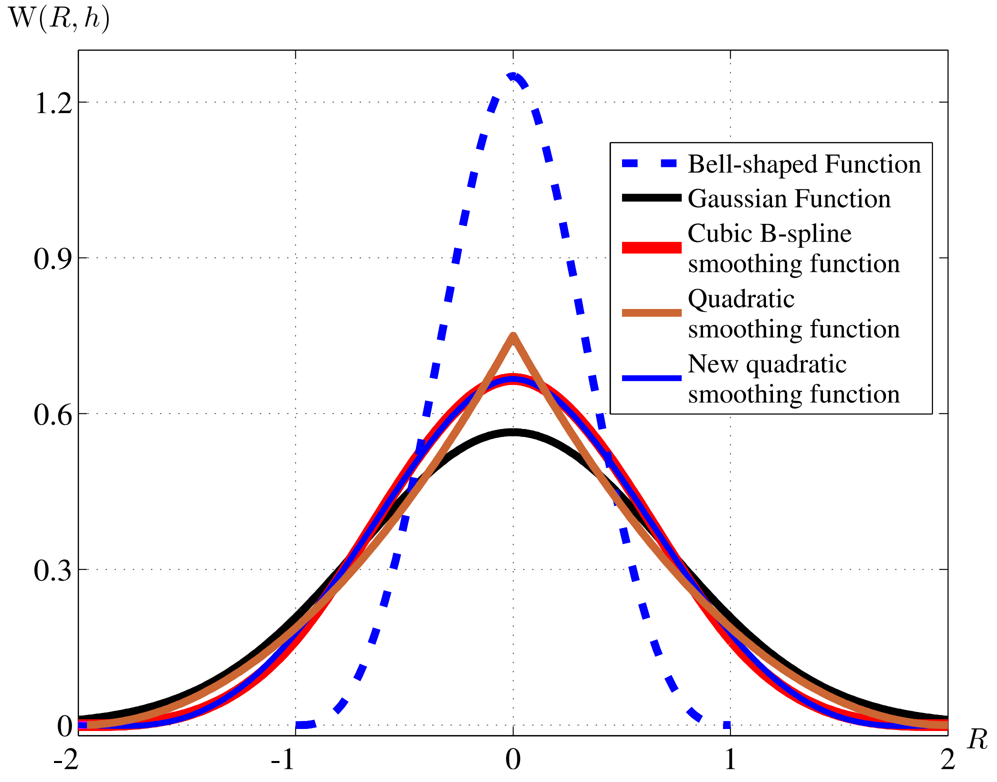


Figure 2.2: Different commonly used smoothing functions

where α_d equals $\frac{1}{\sqrt{\pi}h}$, $\frac{1}{\pi h^2}$ and $\frac{1}{\pi^{\frac{3}{2}}h^3}$ respectively in one-, two- and three-dimensional space. The Gaussian function has a sufficiently high smoothness even for high orders of derivatives. This property makes the approximation very stable and gives high accuracy on disordered distributions of particles. However, the function does not satisfy the compact support condition. The value of the function never reaches zero theoretically unless R equals infinity. However, since the function numerically approaches zero very fast, the error caused by the kernel approximation can almost be ignored. However these non-zero values can lead to higher computational cost as there are more particles inside the support domain with very small values.

One of the most popular smoothing functions is the cubic B-spline function, which was first used by Monaghan and Lattanzio [78] shown in Figure 2.2,

$$W(R,h) = \alpha_d \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3, & R \leq 1, \\ \frac{1}{6}(2-R)^3, & 1 \leq R \leq 2, \\ 0, & R > 2, \end{cases} \tag{2.21}$$

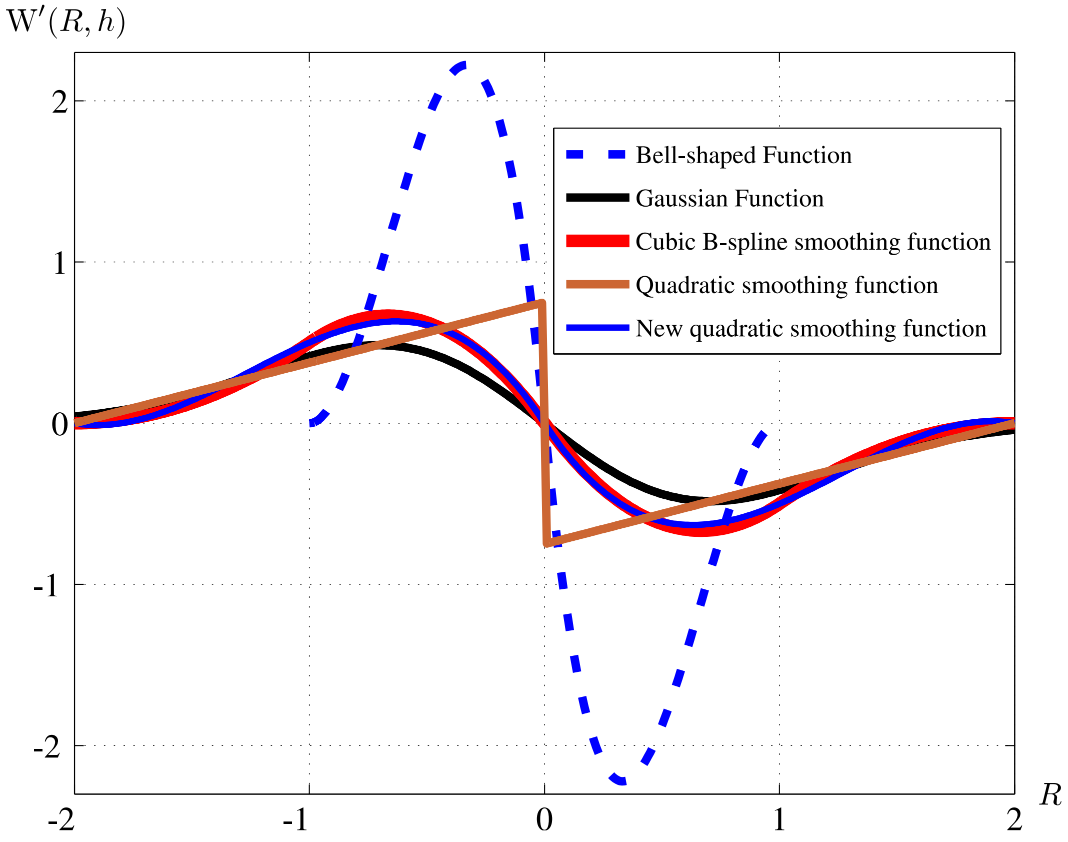


Figure 2.3: The spatial derivatives of different commonly used smoothing functions.

where α_d equals $\frac{1}{h}$, $\frac{15}{7\pi h^2}$ and $\frac{3}{2\pi h^3}$ respectively in one-, two- and three-dimensional space. This function has been most widely applied in previous SPH literature, because its curve is close to the Gaussian function but has a completely compact support. However, the second derivative of this function is a linear piecewise function, not a smooth function and can lead to a loss of stability of the SPH method compared to other smoothing functions.

Johnson *et al.* developed a quadratic smoothing function to simulate high velocity impact problems [43] in 1996.

$$\mathrm{W}(R,h) = \alpha_d \begin{cases} \frac{3}{16}R^2 - \frac{3}{4}R + \frac{3}{4}, & R \leq 2, \\ 0 & R > 2, \end{cases} \tag{2.22}$$

where α_d equals $\frac{1}{h}$, $\frac{2}{\pi h^2}$ and $\frac{5}{4\pi h^3}$ respectively in one-, two- and three-dimensional space. The derivative of the quadratic smoothing function increases as particles move closer and decreases as particles move apart. The authors believed the application of this smoothing function brought an improvement over the cubic spline function. It can solve the compressive instability problem, because of the rate of repulsive force between particles increasing as particles moving closer.

Liu *et al.* developed an algorithm to construct a smoothing function to meet different requirements [59]. A new quartic smoothing function was constructed to demonstrate the advantage and effectiveness of the algorithm, i.e.

$$\mathrm{W}(R,h)=\alpha_d\begin{cases}\frac{2}{3}-\frac{9}{8}R^2+\frac{19}{24}R^3-\frac{5}{32}R^4, & R\leq 2,\\ 0 & R>2,\end{cases}\tag{2.23}$$

where α_d equals $\frac{1}{h}$, $\frac{15}{7\pi h^2}$ and $\frac{315}{208\pi h^3}$ respectively in one-, two- and three-dimensional cases. The behaviour of this smoothing function and its spatial derivative are very close to the cubic B-spline function as shown in Figure 2.4, but is has smoother second order derivatives. Liu *et al.* [59] thought that this smoothing function could bring stable SPH simulations.

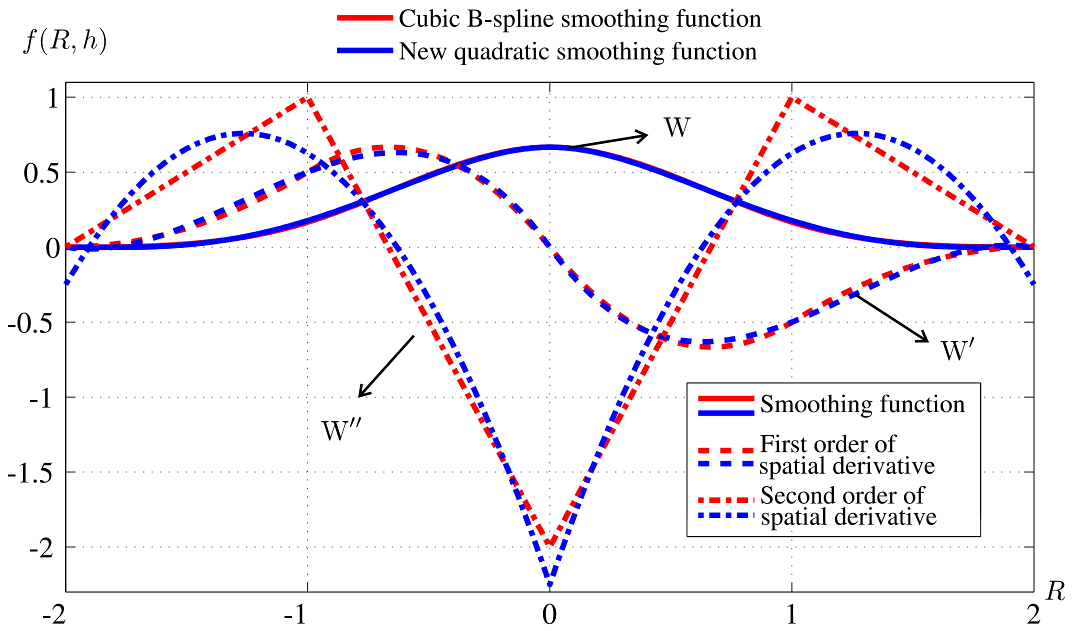
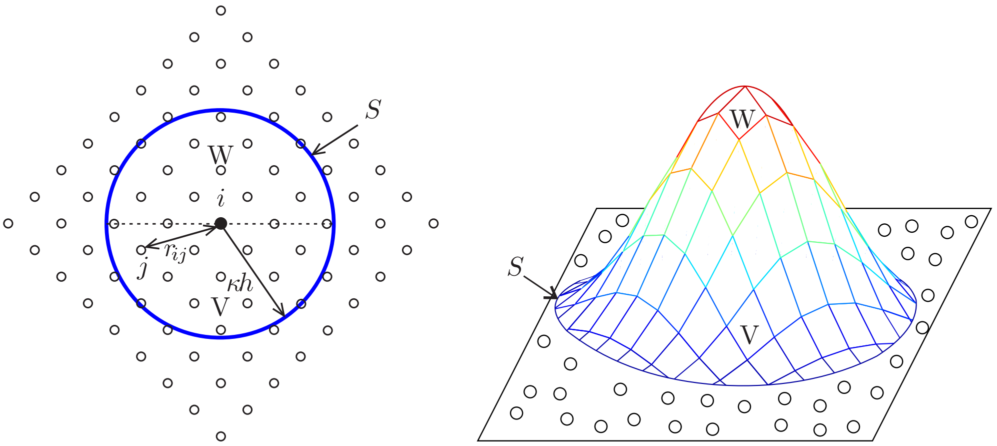


Figure 2.4: The difference between the cubic B-spline function first used by Monaghan and Lattanzio [78] and the new quartic smoothing function constructed by Liu *et al.* [59] (here $f(R,h)$ includes $\mathrm{W}(R,h)$, $\mathrm{W}'(R,h)$ and $\mathrm{W}''(R,h)$).

2.3 Particle approximation

In the SPH method, the continuum domain is discretised into a set of particles. These particles carry their individual physical properties to present the physical system. For example, particles carry volume to represent the integral finite small volume such as the mass and density. The integral form of a function and its derivative (Equations (2.7) and (2.15)) can also be transformed to the discretised form as a summation of all the particles in the support domain. This is the second key step of the SPH concept, which is called particle approximation. This step is carried out by using the smoothing functions and the properties of particles. The term $d\mathbf{x}'$ in Equation (2.7) is



(a) (b)

Figure 2.5: Particle approximation by smoothing function W in support domain V with radius κh. (a) Top view of the problem domain and the support domain of the smoothing function. (b) The surface of the smoothing function in a two-dimensional case.

determined to be an infinitesimal segment, area and volume in one-, two- and three-dimensional cases respectively. Equation (2.7) can be represented in a common form for any dimensional problem,

$$f(\mathbf{x}_i) = \int_{\mathrm{V}} f(\mathbf{x}_j)\mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)dV_j, \tag{2.24a}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ indicate the locations of particles i and j, the same as $\mathbf{x}$ and $\mathbf{x}'$ in Equation (2.7), and dV_j is a common representation of $d\mathbf{x}'$. The integral form can be approximated by a summation form by discretising the domain into a finite number of particles,

$$f_i = \sum_{j=1}^{N} f_j \mathrm{W}_{ij} \Delta V_j, \tag{2.24b}$$

where f_i, f_j and W_{ij} are abbreviations for $f(\mathbf{x}_i)$, $f(\mathbf{x}_j)$ and $\mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)$ respectively. The subscript j indicates a neighbouring particle of the considered particle i; ΔV_j indicates the volume of the particle j; N is the number of particles within the support domain V. The finite volume ΔV_j of particle j can be calculated by the mass m_j and density ρ_j of particles by relating them to physical phenomena,

$$\Delta V_j = \frac{m_j}{\rho_j}. \tag{2.25}$$

Combining Equations (2.24) and (2.25), the discretised form of the kernel approximation can be found as

$$f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \mathrm{W}_{ij}. \tag{2.26}$$

This formula states that the value of the function at the position of particle i is the summation of contributions of neighbouring particles j in the support domain.

The particle approximation of a function's derivative can be achieved by following, the same process as Equation (2.15), i.e.

$$\begin{aligned} \nabla\big(f(\mathbf{x}_i)\big) &= -\int_{\mathrm{V}} f(\mathbf{x}_j) \frac{\partial \mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)}{\partial \mathbf{x}_j} d\mathbf{x}_j \qquad (2.27) \\ &= \int_{\mathrm{V}} f(\mathbf{x}_j) \frac{\partial \mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)}{\partial \mathbf{x}_i} d\mathbf{x}_j \\ &\cong \sum_{j=1}^{N} f(\mathbf{x}_j) \frac{\partial \mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)}{\partial \mathbf{x}_i} \Delta V_j \\ &= \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) \frac{\partial \mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)}{\partial \mathbf{x}_i}, \end{aligned}$$

or

$$\nabla f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \nabla_i \mathrm{W}_{ij}, \tag{2.28}$$

where

$$\nabla_i \mathrm{W}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{R_{ij}} \frac{\partial \mathrm{W}_{ij}}{\partial R_{ij}} = \frac{\mathbf{x}_{ij}}{R_{ij}} \frac{\partial \mathrm{W}_{ij}}{\partial R_{ij}}. \tag{2.29}$$

In the second line of Equation (2.27), the minus sign can be removed because the derivative of $\mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)$ is an odd function and $\frac{\partial \mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)}{\partial \mathbf{x}_j} = -\frac{\partial \mathrm{W}(\mathbf{x}_i - \mathbf{x}_j, h)}{\partial \mathbf{x}_i}$.

Equation (2.28) once again shows that the spatial derivative operator is transformed from the variables on particle i to the smoothing function by summing the contribution of variables on all the neighbouring particles in the support domain.

It can be noted that the leading truncation error term of the SPH approximation is $\mathcal{O}(h^2)$. The finiteness of the kernel approximation support means that only a limited number of neighbouring particles play a role in all the sums of conservation equations. This is used to reduce the computational time by building a linked list between particles.

Using particle approximation in Equations (2.26) and (2.28), we can transform the continuous integral representation of a function and its derivatives into the discretised summation form through a finite number of particles. Obviously, the application of particle approximation by discretising integration into a summation form plays an essential role in making the SPH method purely mesh-free.

2.4 Boundary accuracy

2.4.1 Correction forms

As mentioned in Section 2.1, the SPH approximation will not have second order accuracy when the problem domain intersects with the support domain of a smoothing function; for example, when the support domain is cut by the boundary of the problem domain. It is clear that for a particle near a domain boundary, the support domain may lack sufficient neighbouring particles, as shown in Figure 2.6. Equation (2.5) will not be satisfied, and consequently Equation (2.14) will not be true. Therefore, particles near the boundary will affect the accuracy of the SPH method. In this section, this 'boundary accuracy' problem will be discussed and a correction method to be applied to the approximation function will be introduced to overcome this drawback [11].

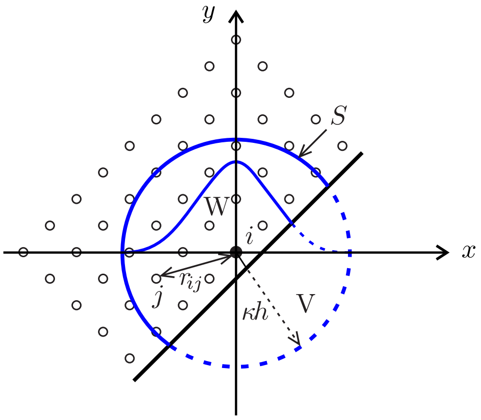


Figure 2.6: The support domain has been truncated by the boundary. There is no sufficient neighbouring particles in the support domain, when the particle is near the boundary.

The correction forms are based on the *Unity* property of the smoothing function as mentioned in Section 2.2. In other words, the sum of the volumes of neighbouring particles multiplied by the values of the smoothing function should be one, when particle i is not near the problem boundary,

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathrm{W}_{ij} = 1. \tag{2.30}$$

When a particle is near the boundary, Equation (2.30) is not met. To ensure the unity property of the smoothing function in the SPH method for all particles, we use the original representation (2.26), divide the correction form (left-hand side of equation (2.30)) and obtain a new form particle approximation in the SPH method.

$$f_i = \frac{\displaystyle\sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \mathrm{W}_{ij}}{\displaystyle\sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathrm{W}_{ij}} = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \widetilde{\mathrm{W}}_{ij}, \tag{2.31a}$$

where $\widetilde{\mathrm{W}}_{ij}$ gives the correction form of the smoothing function,

$$\widetilde{\mathrm{W}}_{ij} = \frac{\mathrm{W}_{ij}}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathrm{W}_{ij}}. \tag{2.31b}$$

Equation (2.31b) ensures that the correction form works only when particle i is near the domain boundary; when particle i is inside the problem domain we will find $\widetilde{\mathrm{W}}_{ij} = \mathrm{W}_{ij}$.

For the spatial derivative of a function, accuracy is also influenced by the boundary accuracy problem. To solve this, a correction form for the derivative is used. We know the spatial derivatives of functions $a(\mathbf{x}) = \mathbf{x}$ and $b(\mathbf{x}) = 1$ are $\nabla\big(a(\mathbf{x})\big) = \dfrac{\partial \mathbf{x}}{\partial \mathbf{x}} = 1$ and $\nabla\big(b(\mathbf{x})\big) = \dfrac{\partial 1}{\partial \mathbf{x}} = 0$ respectively. Combining these two equations with the equation (2.28) we obtain,

$$\nabla \mathbf{x}_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathbf{x}_j \nabla \mathrm{W}_{ij} = 1, \tag{2.32a}$$

and

$$\mathbf{x}_i \nabla 1 = \mathbf{x}_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} \nabla \mathrm{W}_{ij} = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathbf{x}_i \nabla \mathrm{W}_{ij} = 0. \tag{2.32b}$$

Combining these two equations we find

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i) \nabla \mathrm{W}_{ij} = 1. \tag{2.32c}$$

However, equations (2.32) are also based on the unity condition of the smoothing function. When the particle is near the boundary, the accuracy of the particle approximation (2.28) will not be second-order and neither Equation (2.16) nor Equation (2.32c) will be satisfied. We solve this problem in the same way as before, using the original representation (2.28) and divided by the spatial correction form (2.32c) to obtain a new spatial particle approximation for derivatives,

$$\nabla f_i = \frac{\sum\limits_{j=1}^{N} \frac{m_j}{\rho_j} f_j \nabla \mathrm{W}_{ij}}{\sum\limits_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i) \nabla \mathrm{W}_{ij}} = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \nabla \widetilde{\mathrm{W}}_{ij}. \tag{2.33a}$$

The difference and summation of the variables on the two particles can also be taken to obtain the spatial derivative result,

$$\nabla f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} (f_j - f_i) \nabla \widetilde{\mathrm{W}}_{ij}. \tag{2.33b}$$

$$\nabla f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} (f_j + f_i) \nabla \widetilde{\mathrm{W}}_{ij}. \tag{2.33c}$$

where

$$\nabla \widetilde{\mathrm{W}}_{ij} = \frac{\nabla \mathrm{W}_{ij}}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i) \nabla \mathrm{W}_{ij}} \tag{2.33d}$$

This is the correction representation of the spatial derivative of the smoothing function. When particle i inside the problem domain is far away from the boundary, we find $\nabla \widetilde{\mathrm{W}}_{ij} = \nabla \mathrm{W}_{ij}$. This correction form for derivatives ensures that Equation (2.33d) works only when particle i is near the domain boundary. Equation (2.33d) now contains a denominator which is different for each particle which destroys the strict conservation properties of SPH.

2.4.2 Examples

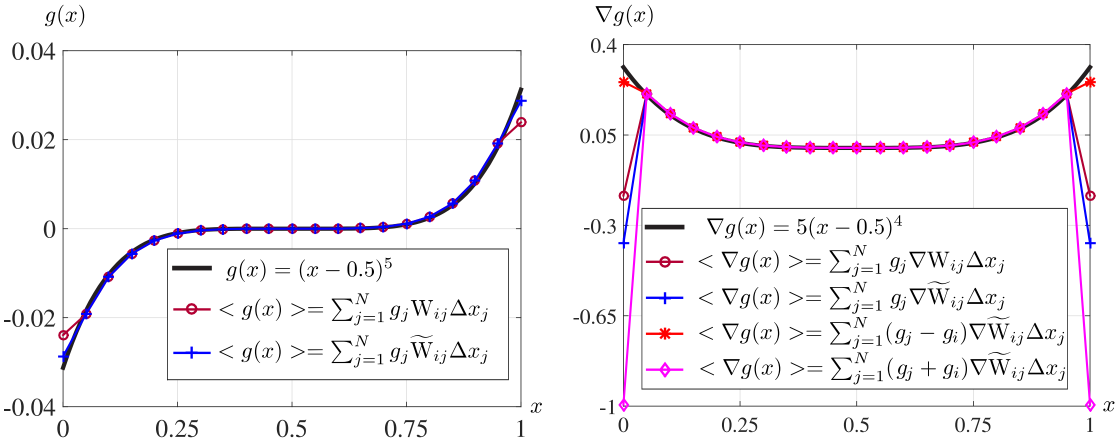
In order to present the difference between original particle approximations (2.26) and (2.28) and the correction form of particle approximations (2.31) and (2.33), a simple example is shown here to demonstrate the benefit brought by the correction forms.

For a 1-D problem with a function $g(x)$, first derivative $\nabla g(x)$ and boundary $0 \leq x \leq 1$,

$$g(x) = (x - 0.5)^5, \tag{2.34a}$$

$$\nabla g(x) = 5(x - 0.5)^4. \tag{2.34b}$$

First, we discretise the problem domain ($[0, 1]$) for the function $g(x)$ into 21 particles. Since this is a one dimensional problem, the volume of each particle should be a segment. The volume of each particle can be calculated by $\Delta V = \Delta x = 0.05$. Note that the cubic B-Spline function has been chosen as the smoothing function in this thesis with $\kappa = 2$ and the value of the α_d depends on the dimension of the problem and the smoothing length (α_d equals $\dfrac{1}{h}$, $\dfrac{15}{7\pi h^2}$ and $\dfrac{3}{2\pi h^3}$ respectively in one-, two- and three-dimensional space). Since the summation of the volumes of neighboring particles multiplied by the values of the smoothing function is one ($\sum \Delta x W_{ij} = 1$) when $h = \Delta x$ in the cubic B-Spline function, the smoothing length is chosen to be the same as the spacing of the particles ($h = \Delta x$). Thus the value of α_d is $\dfrac{1}{h} = 20$ in this example.



(a) (b)

Figure 2.7: (a) Comparison of results of real function, original approximation and corrected approximation methods; (b) Comparison of results of spacial gradient of a function.

In above figures, the curves are the analytical function and its first-order gradient. The line with circles shows the results of using traditional uncorrected methods with Equations (2.26) and (2.28). The line with plus signs gives the results of considering boundary accuracy by using Equation (2.31) and (2.33a). The line with star signs presents the results by using Equation (2.33b) to solve the boundary problem. The line with diamond signs shows the results by using Equation (2.33c) to solve the boundary problem. It is easy to see that the internal parts of the simulation results are the same in different methods as shown in Figure 2.7. The corrected SPH approximation only works on the boundary and increases the accuracy of the simulation near the boundary. The L^1 and L^2 norm errors of different versions of the SPH approximation are shown in Table 2.1, where the L^1 and L^2 norm errors are defined as,

$$error_1 = \frac{\sum |u^{SPH} - u^{exact}|}{\sum |u^{exact}|}.$$

$$error_2 = \frac{\sqrt{\sum |u^{SPH} - u^{exact}|^2}}{\sqrt{\sum |u^{exact}|^2}}.$$

Note that using differences between particles in the corrected SPH approximation can obtain higher accuracy than others in the spatial derivative approximation.

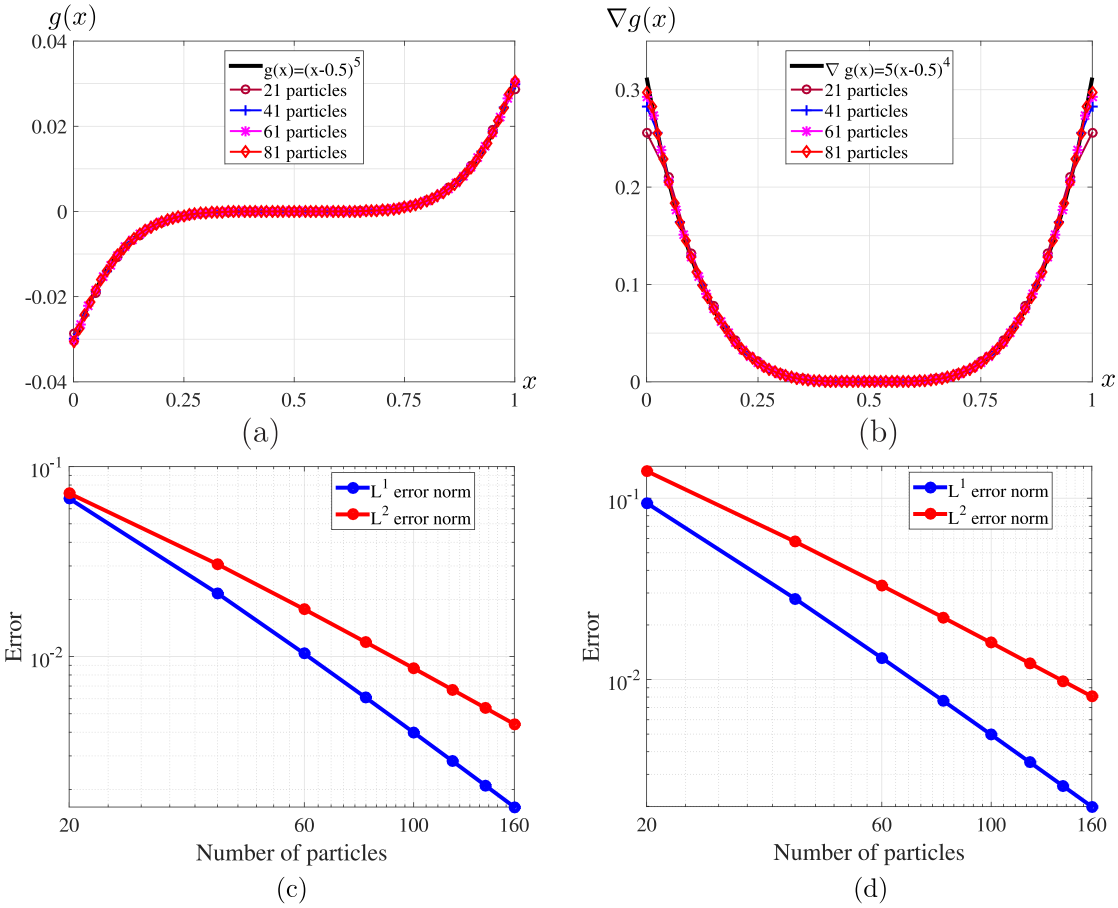
Equations	L^1 norm error	L^2 norm error
$< g(x) >= \sum g_j \Delta x_j W_{ij}$	0.1373	0.1943
$< g(x) >= \sum g_j \Delta x_j \widetilde{\mathrm{W}}_{ij}$	0.0680	0.0724
$< \nabla g(x) >= \sum g_j \Delta x_j \nabla W_{ij}$	0.6504	1.2282
$< \nabla g(x) >= \sum g_j \Delta x_j \nabla \widetilde{\mathrm{W}}_{ij}$	0.8835	1.6841
$< \nabla g(x) >= \sum (g_j - g_i) \Delta x_j \nabla \widetilde{\mathrm{W}}_{ij}$	0.0940	0.1410
$< \nabla g(x) >= \sum (g_j + g_i) \Delta x_j \nabla \widetilde{\mathrm{W}}_{ij}$	1.6730	3.2283

Table 2.1: Error rates of using the different types of SPH approximation for function $g(x) = (x - 0.5)^5$ and its spatial derivative.

Then more particles (41, 61 and 81 particles) are applied to approximate the function and investigate the convergence of the SPH approximation. The volume of each particle can be calculated by $\Delta V = \Delta x = 0.025, \quad 0.025, \quad 0.0167$, and 0.0125 respectively. The corrected forms are applied in the approximation and the differences between particles are adopted in the spatial derivative approximation to obtain higher accuracy. The approximation results and the convergence order are shown in Figure 2.8. The errors are presented in the Table 2.2. It has been found in the literature that the convergence rate of the SPH method is problem dependent and it also depends on the number of neighbouring particles [102, 18, 118]. It is easy to obtain that the convergence rate in this example is 1.9 in L^1- norm and 1.4 in L^2- norm.

	$< g(x) >$				$< \nabla g(x) >$			
Particles	21	41	61	81	21	41	61	81
L^1 norm error	0.068	0.0215	0.0103	0.0060	0.0940	0.0279	0.0131	0.0076
L^2 norm error	0.0724	0.0306	0.0177	0.0119	0.1410	0.0576	0.0329	0.0220

Table 2.2: Error rates of the SPH approximation with different numbers of particles for function $g(x) = (x - 0.5)^5$ and its spatial derivative.



(a) (b)

(c) (d)

Figure 2.8: (a) and (b) are the comparative results of approximation of the function and its derivative with different numbers of particles; (c) and (d) show the L^1 and L^2 norm error of the approximations results of the function and its derivative with different numbers of particles.

2.5 Neighbouring particle searching methods

In the SPH method, as the smoothing function has a compact support domain, the supporting domain with the diameter of κh only includes a finite number of particles, which are used in the particle approximations. These particles are called Nearest Neighbouring Particles (NNP) of the considered particle. The process of searching for the nearest particles is normally called Nearest Neighbouring Particle Searching (NNPS). The difference between mesh-based numerical methods and SPH is that the neighbouring particles of a considered particle can vary with time in the SPH method. While, in the mesh-based numerical methods, the positions of the adjacent grid-cells are fixed once the mesh is identified. The three widely used NNPS approaches in SPH applications are the all-pair search algorithm, the linked-list search algorithm, and the tree search algorithm.

2.5.1 All-pair search

The all-pair search approach is a simple and direct NNPS method (Figure 2.9). For a given particle i, the all-pair search approach is to calculate the distance R_{ij} from i to each particle ($j = 1, 2, \cdots, N$, when N is the total number of particles in the problem domain). If the distance R_{ij} is less than the radius κh of the support domain for the considered particle i, then the particle j belongs to the neighbouring particle of particle i in the support domain. Therefore particles i and j are a pair of adjacent particles, and particle i is also the neighbouring particle of particle j in the support domain. This searching process is applied to all particles, i.e. conducting a pair searching for all particles $i = 1, 2, \cdots, N$, and each search includes all particles ($j = 1, 2, \cdots, N$). The order of all-pair search method is $O(N^2)$ and the efficiency is poor[23]. It is worth noting that, NNPS is carried out in the computation of each time step; as a result, this method is very time consuming and not favourable for problems involving large number of particles. Therefore the all-pair search method is only efficient for numerical problems with small particle numbers.

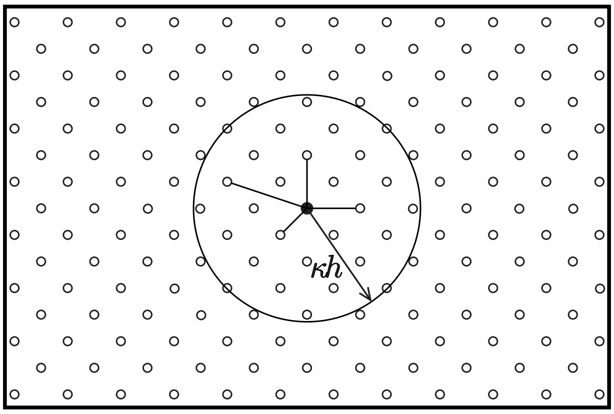


Figure 2.9: All-pair search method for searching for the neighbouring particles in a two-dimensional case. The distances between every considered particle and other particles is compared with the radius of the support domain of the considered particle to identify whether these two particles are adjacent.

2.5.2 Linked-list search algorithm

When the smooth length is constant, the linked-list search algorithm is very effective. Monaghan and Gingold (1983) mentioned that adopting cells as a bookkeeping device can significantly reduce computation time [77]. By assigning all the particles into different cells and identifying them by linked-lists, only a group of particles is searched during the NNPS process which can save substantial computational time. Monaghan

(1985) described the process of searching the nearest neighboring particle as using the linked-list algorithm [70]. Hockney and Eastwood (1988) applied this method in their discussion of short-range forces with particle simulation methods and introduced more details of this linked-list search algorithm [40]. Rhoades (1992) has applied this method to search the nearest neighbouring particles and claimed this algorithm is very effective and practical in vector computation [95]. Simpson (1995) provided more details of the linked-list search algorithm when applied to the SPH method in the three-dimensional accretion disks problem [99].

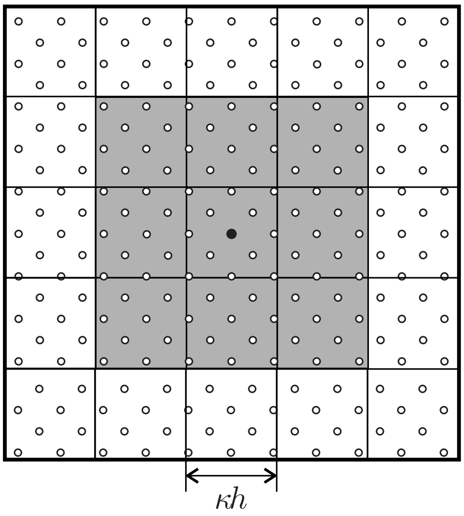


Figure 2.10: Cell linked-list algorithm for searching for the nearest neighboring particles in two-dimensional cases. The smoothing length is constant for each particle.

When implementing the linked-list algorithm, a temporary mesh is allocated on the problem domain (Figure 2.10). The size of a mesh cell is based on the radius of the support domain. If the computation scale of the support domain is κh, the size of the mesh cell shall be set to κh. Then for a considered particle i, the neighbouring particles can only be in the same mesh cell or the adjacent cells. Therefore when $\kappa = 2$, the search range in one-, two- and three-dimensional space is in 3,9,27 cells respectively. The linked-list algorithm allocates each particle in the mesh cell and connects all the particles in each cell through simple allocation rules. Dominguez *et al.* (2010) discussed four different allocation rules in [23].

Two different variants of the linked-list algorithm have been referenced in [23], cell linked list and verlet list. The main difference between these two algorithms is the size of the cells. The cell linked list has the size of cell κh and updates the neighbour list for each time step. The size of the cell is $2h + \Delta h$ in the verlet list but it updates the neighbour list after certain time steps. If the average number of particles in each cell is small enough, the complexity of the linked-list algorithm is of order $O(N)$. However, the disadvantage is when the smooth length is variable, the mesh space is not able to adapt to each particle. Thus the searching efficiency of the linked-list algorithm is low in this case.

2.5.3 Tree search algorithm

The tree search algorithm works well for problems with variable smoothing lengths. It involves creating ordered trees according to the particle positions. Once the tree structure is created, it can be used efficiently to find the nearest neighboring particles. In this thesis, an adaptive hierarchy tree search method is adopted to suit the needs of adaptive smoothing lengths. This tree method recursively splits the maximal problem domain into octants that contain particles, until a leaf on the tree has only one individual particle (Figure 2.11). After the tree structure is constructed, the search process can be performed.

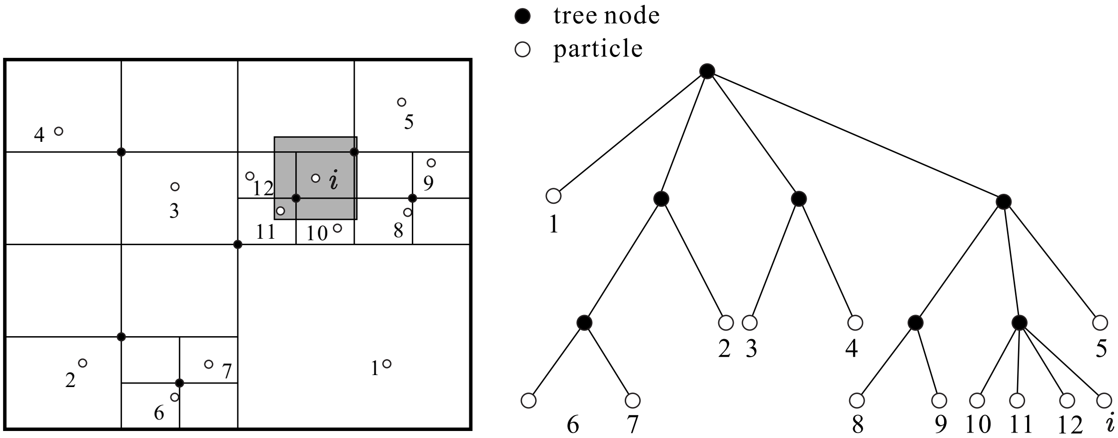


Figure 2.11: Tree structure and tree search algorithm in two-dimensional space. The tree is constructed by recursively splitting the maximal problem domain into octants that contain particles, until the leaves on the tree are individual particles. The tree search algorithm is performed by checking whether the volume of the search cube (shaded area) for a given particle overlaps with the volume represented by the current node.

For a given particle i, a cube with the side of $2\kappa h_i$ is used to enclose the particle, which is located at the center of the cube. At each level, checks are done to verify whether the volume of the search cube overlaps with the volume represented by the current node in the tree structure. If not, continuation is halted on that particular path. If yes, the procedure continues the tree descent and goes down to the next level repeatedly until the current node represents a particle. Then the algorithm checks whether the particle is within the support domain of the given particle i and records the one in the support domain as a neighbouring particle.

The complexity of this tree search algorithm is of order $O(N \log N)$ (Hernquist and Katz, 1989) [39]. Numerical tests show that the SPH method combined with tree search method is very efficient and robust especially for a large number of particles of variable smoothing lengths.

Concluding remarks

In this chapter, the basic concept of the SPH method has been presented. The first step is to represent the field variables and their derivatives in an integration form by using smoothing functions to do the kernel approximation. The second step is to discretise the continuous problem domain into a set of particles, which carry the field variables on the position of the particles. Different smoothing functions and their applications are also introduced in this chapter. One example with the standard SPH analysis result and corrected SPH result is shown to prove that accuracy is increased by the correction form near the boundary. Three different nearest neighboring particle searching methods are stated in Section 2.5. One of them, the k-d tree method, will be adopted in this thesis.

This research applies the SPH method to solid mechanics. The SPH estimation method will be adopted to approximate the governing equations to simulate solid dynamic physical behaviour. The difference between the Eulerian and Lagrangian approaches will be presented in the next chapter since the governing equations of these two approaches are different. The SPH approximation form of the governing equations will also be derived and different special treatments will be added into the equations, like artificial viscosity and the treatment for tensile instability. Then an SPH system will be obtained. All the details will be presented in Chapter 3.

Chapter 3

SPH and dynamics

Introduction

The concept and the essential equations of the SPH method have been derived in this chapter, in order to illustrate the various components of the SPH method. The SPH method is developed to simulate hydrodynamic problems using a set of governing equations i.e., mass and momentum conservation equations. The governing equations of elastic mechanics will be represented in the SPH approximation form in this chapter. Therefore, the basic SPH concept for hydrodynamic problem is to first discretise the continuous problem domain into particles and then approximate the solution function and its derivatives for each particle. Based on the function approximation, the governing equations can be transformed to a set of ordinary differential equations (ODE) which can be solved on each individual particle.

The governing equations can be derived using both Eulerian and Lagrangian approaches which will be introduced in Section 3.1. Then the SPH form of the mass and momentum conservation equations is derived in Section 3.2, together with the application of the Jaumann stress rate to large deformation problems. Unphysical oscillations are observed when handling the shock wave problem through the SPH method [58]. To improve the stability and accuracy of the method, some special treatments will be introduced into the SPH algorithm. The boundary condition will be discussed to avoid the reduced accuracy caused by solid boundary, non-slip and slip boundary problems in Section 3.3; artificial viscosity will be introduced into the momentum equation to dissipate unphysical oscillations in Section 3.4; and artificial stress will be presented to solve the tensile instability problem in Section 3.5.

3.1 Eulerian and Lagrangian approaches

The SPH approximation is applied to the governing equations to simulate dynamic problems. The SPH method is a purely Lagrangian numerical method since the governing equations are derived based on the Lagrangian approach. In this section, both Eulerian and Lagrangian approaches will be introduced since the governing equations can be derived based on either Eulerian or Lagrangian approaches.

Consider a flow field described by streamlines in Figure 3.1. There are two approaches of representing the motion of the flow; Eulerian and Lagrangian. The Eulerian approach assumes a control volume V fixed in the flow field with fluid moving through it. The control surface S binds the control volume. Two different ideas can be embedded into this approach. One assumes the control volume is infinitesimal whereas, the other assumes a reasonably large and finite volume. The equations for the Eulerian description of fluid dynamics are known as a conservation form of the governing equations, either integral or partial differential form [1]. Alternatively, the assumption of the Lagrangian description is that the control volume moves with the flow and the fluid particles inside the volume are kept the same. This approach (Lagrangian) also has two kinds of ideas, one assumes the control volume is infinitesimal whereas, the other is reasonably large and finite volume. The equations obtained from the Lagrangian approach are known by the non-conservation form of the governing equations [1]. Thus, instead of studying the whole fluid flow at once, it is easier to pay attention to the control volume itself and to describe the motion of the fluid flow. Fundamental concepts are applied to the control volume and some key equations called governing equations are obtained. These governing equations are based on the conservation of mass, momentum and energy, which are discussed in Appendix. This project adopts the particle based method, thus the infinitesimal moving fluid element model is used to do further studies.

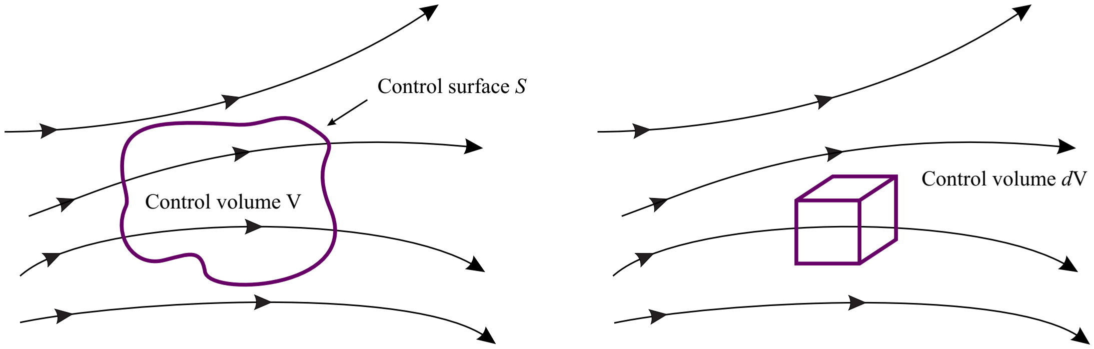
3.2 Deriving the SPH equations

In solid mechanics, conservation laws produce three important governing equations, including mass, linear momentum and energy conservation. Since the temperature change can be ignored in the problem presented here, energy conservation is not considered in this study. The rate of change of field variables can be expressed by these governing equations, such as density and velocity, which are used to build up a set of Ordinary Differential Equations (ODE) with respect to time. The governing equations in elastodynamics express the conservation of mass and momentum as follows:

$$\frac{D\rho}{Dt} = -\rho\nabla\cdot\mathbf{v}, \tag{3.1a}$$

$$\frac{D\mathbf{v}}{Dt} = \frac{1}{\rho}\nabla\boldsymbol{\sigma} + \mathbf{g}, \tag{3.1b}$$

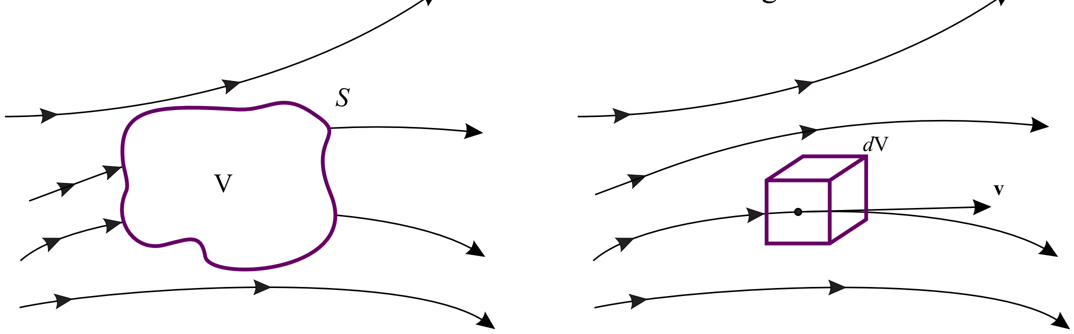
where $\nabla = \left\{\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \quad \frac{\partial}{\partial z}\right\}^T$ is the gradient operator, $\mathbf{g}$ is the acceleration caused by external force which is ignored in this thesis, ρ is the density, $\mathbf{v} = \left\{\mathrm{v}_x \quad \mathrm{v}_y \quad \mathrm{v}_z\right\}^T$



Finite control volume fixed in space with the fluid moving through it

(a)

Infinitesimal control element fixed in space with the fluid moving through it



Finite control volume moving with the fluid such that the same fluid particles are always in the same control volume

(b)

Infinitesimal fluid element moving with the fluid with the velocity **v** equal to the local flow velocity at each point

Figure 3.1: (a)Eulerian Approach. (b)Lagrangian Approach. (Anderson [1])

is the velocity, σ is the stress tensor, $\frac{D}{Dt} = \frac{\partial}{\partial t} + \nabla \cdot \mathbf{v}$ is the substantial derivative, and $\nabla \cdot \mathbf{v} = \frac{\partial \mathrm{v}_x}{\partial x} + \frac{\partial \mathrm{v}_y}{\partial y} + \frac{\partial \mathrm{v}_z}{\partial z}$. The mass (3.1a) and momentum (3.1b) conservation equations are derived in the Appendix.

Since the use of the product rule of differentiation, we can express the term $\frac{1}{\rho}\nabla\boldsymbol{\sigma}$ of equation (3.1b) in the following form,

$$\nabla\left(\frac{\boldsymbol{\sigma}}{\rho}\right) = \frac{1}{\rho}\nabla\boldsymbol{\sigma} - \frac{\boldsymbol{\sigma}}{\rho^2}\nabla\left(\frac{1}{\rho}\right), \tag{3.2a}$$

then we have,

$$\frac{1}{\rho}\nabla\boldsymbol{\sigma} = \nabla\left(\frac{\boldsymbol{\sigma}}{\rho}\right) + \frac{\boldsymbol{\sigma}}{\rho^2}\nabla\left(\frac{1}{\rho}\right). \tag{3.2b}$$

As mentioned, elastodynamic problems will be analysed here. The total strain rate can be written as,

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right). \tag{3.3}$$

The derivations of the strain and stress rate are included in the Appendix. According to Hooke's law, we can write a stress tensor into this expression,

$$\dot{\boldsymbol{\sigma}} = 2G\dot{\boldsymbol{e}} + K\left(tr(\dot{\boldsymbol{\varepsilon}})\right)\mathbf{I}, \tag{3.4}$$

where $tr(\dot{\boldsymbol{\varepsilon}})$ means the trace of the matrix which is $(\varepsilon^{xx} + \varepsilon^{yy} + \varepsilon^{zz})$ in three dimensional space; $\dot{\boldsymbol{e}} = \dot{\boldsymbol{\varepsilon}} - \frac{1}{3}tr(\dot{\boldsymbol{\varepsilon}})\mathbf{I}$ is the deviatoric strain rate tensor, $\mathbf{I}$ is the Kronecker's delta tensor; K is the elastic bulk modulus and G is the shear modulus which can be represented by Young's modulus E and Poisson's ratio ν ;

$$K = \frac{E}{3(1-2\nu)} \quad \text{and} \quad G = \frac{E}{2(1+\nu)}.$$

However, regarding a large deformation problem, the Jaumann stress rate $\dot{\hat{\boldsymbol{\sigma}}}$ will be adopted to introduce a rotation influence on the constitutive relations.

$$\dot{\hat{\boldsymbol{\sigma}}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma}\boldsymbol{\omega} - \boldsymbol{\omega}\boldsymbol{\sigma} \tag{3.5a}$$

Equation (3.4) then becomes,

$$\dot{\boldsymbol{\sigma}} = 2G\dot{\boldsymbol{e}} + K\Bigg(tr(\dot{\boldsymbol{\varepsilon}})\Bigg)\mathbf{I} - \boldsymbol{\sigma}\boldsymbol{\omega} + \boldsymbol{\omega}\boldsymbol{\sigma}, \tag{3.5b}$$

where $\boldsymbol{\omega}$ is the rotation tensor and can be represented by the gradient of velocity,

$$\boldsymbol{\omega} = \frac{1}{2}\left(\nabla\mathbf{v} - (\nabla\mathbf{v})^T\right). \tag{3.6}$$

Then we discretise the continuum domain by particles and apply the kernel approximation to the SPH system, obtaining the spatial gradient of the velocity as follows,

$$\nabla\mathbf{v}_i = \sum_{j=1}^{N}\frac{m_j}{\rho_j}\mathbf{v}_j\nabla\widetilde{\mathrm{W}}_{ij}. \tag{3.7a}$$

Since we know $\nabla(1) = \sum_{j=1}^{N}\frac{m_j}{\rho_j}\nabla\widetilde{\mathrm{W}}_{ij} = 0$, we can use the difference between neighbouring particles to increase the accuracy by adding a term $(-\mathbf{v}_i\nabla(1) = -\sum_{j=1}^{N}\frac{m_j}{\rho_j}\mathbf{v}_i\nabla\widetilde{\mathrm{W}}_{ij} = 0)$ on the right-hand side of equation (3.7a) and obtain the following equation,

$$\nabla\mathbf{v}_i = \sum_{j=1}^{N}\frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i)\nabla\widetilde{\mathrm{W}}_{ij}. \tag{3.7b}$$

As mentioned before ($\nabla\cdot\mathbf{v} = \dfrac{\partial \mathrm{v}_x}{\partial x}+\dfrac{\partial \mathrm{v}_y}{\partial y}+\dfrac{\partial \mathrm{v}_z}{\partial z}$), the equation expressing rate of change of density (3.1a) can be rewritten into the following form,

$$\frac{D\rho_i}{Dt} = -\rho_i \sum_{j=1}^{N} \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i)\cdot\nabla\widetilde{\mathrm{W}}_{ij}. \tag{3.8a}$$

Then the kernel approximation of the acceleration equation (3.1b) can be represented as,

$$\frac{D\mathbf{v}_i}{Dt} = \sum_{j+1}^{N} m_j\left(\frac{\boldsymbol{\sigma}_j}{\rho_j^2}+\frac{\boldsymbol{\sigma}_i}{\rho_i^2}\right)\nabla\widetilde{\mathrm{W}}_{ij}. \tag{3.8b}$$

The stress rate of small and large deformations (3.4) and (3.5b) can be expressed by the spatial gradient of velocity;

$$\frac{D\boldsymbol{\sigma}_i}{Dt} = 2G\dot{\boldsymbol{e}}_i + K\Big(tr(\dot{\boldsymbol{\varepsilon}}_i)\Big)\mathbf{I}, \tag{3.8c}$$

$$\frac{D\boldsymbol{\sigma}_i}{Dt} = 2G\dot{\boldsymbol{e}}_i + K\Big(tr(\dot{\boldsymbol{\varepsilon}}_i)\Big)\mathbf{I} - \boldsymbol{\sigma}_i\boldsymbol{\omega}_i + \boldsymbol{\omega}_i\boldsymbol{\sigma}_i, \tag{3.8d}$$

where $\dot{\boldsymbol{e}}_i$, $\dot{\boldsymbol{\varepsilon}}_i$ and $\boldsymbol{\omega}_i$ could all be represented by the spatial gradient of velocity. Combine Equations (3.3), (3.6) and (3.7b), then we will obtain the SPH discretisation form of the strain and spin rate,

$$\dot{\boldsymbol{\varepsilon}}_i = \frac{1}{2}\bigg(\sum_{j=1}^{N}\frac{m_j}{\rho_j}(\mathbf{v}_j-\mathbf{v}_i)\nabla\widetilde{\mathrm{W}}_{ij} + \big(\sum_{j=1}^{N}\frac{m_j}{\rho_j}(\mathbf{v}_j-\mathbf{v}_i)\nabla\widetilde{\mathrm{W}}_{ij}\big)^T\bigg), \tag{3.9a}$$

$$\dot{\boldsymbol{\omega}}_i = \frac{1}{2}\bigg(\sum_{j=1}^{N}\frac{m_j}{\rho_j}(\mathbf{v}_j-\mathbf{v}_i)\nabla\widetilde{\mathrm{W}}_{ij} - \big(\sum_{j=1}^{N}\frac{m_j}{\rho_j}(\mathbf{v}_j-\mathbf{v}_i)\nabla\widetilde{\mathrm{W}}_{ij}\big)^T\bigg). \tag{3.9b}$$

3.3 Boundary conditions

As mentioned in Section 2.4, the SPH method will generate errors when particles are near or on the boundary, see Figure 2.6. Equations (2.31) and (2.33) aim to correct the incomplete support of a truncated kernel for free-surfaces. To improve the accuracy of solid boundary, non-slip and slip boundary problems, an approach with ghost particles has been introduced and improved by many authors [74, 82, 93, 107, 12]. Herein, the simple algorithm from Bui [12] is applied to solve the non-slip and slip (symmetric) solid boundary problems.

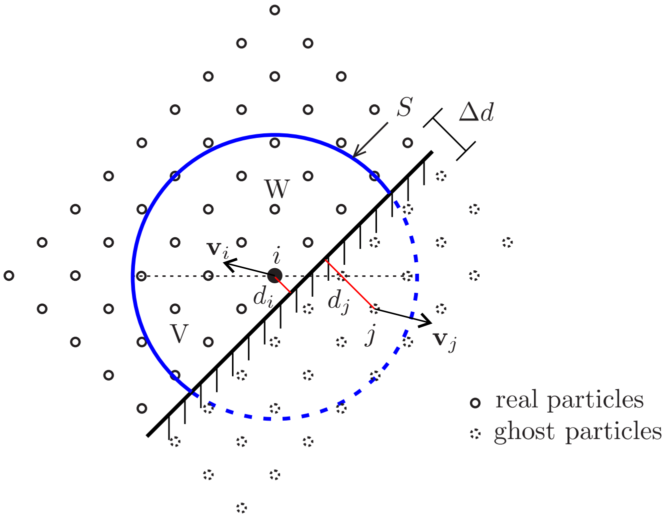


Figure 3.2: Solid boundary treatment with ghost particles.

This approach generates three layers of ghost particles outside of the solid boundary with the uniform distribution as real particles. These particles are located in parallel with the solid boundary with the distance $\Delta p/2$ between first layer and solid boundary Figure 3.2, where Δp is the initial spacing of the particles. The ghost particles will have the same density and mass as the corresponding real particles. The velocity of a ghost particle is related to the velocity of the i^{th} particle, the velocity of boundary and the distance between particles and a non-slip boundary. The artificial velocity of ghost particle j of the considered real particle i could be calculated as,

$$\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j = \beta_g(\mathbf{v}_i - \mathbf{v}_{boundary}), \tag{3.10a}$$

where $\beta_g = \min\left(\beta_{\max}, 1.0 + \dfrac{d_j}{d_i}\right)$, and $\beta_{\max}$ is the safety factor to avoid extremely high velocity while the real particle is too close to the boundary. The factor $\beta_{\max}$ is taken empirically to be 1.5. The velocity of the ghost particle can be represented in the following form;

$$\mathbf{v}_j = (1 - \beta_g)\mathbf{v}_i - \beta_g \mathbf{v}_{boundary}. \tag{3.10b}$$

The stress tensor of the ghost particle j should also be identified in order to calculate Equation (3.8b). The most general way to assign the stress tensor of ghost particle j is derived by Randly and Libersky [93]. Herein, a simpler approach is applied as introduced by [12]. In this approach, the stresses near the boundary are assumed to be uniform. Therefore, if the ghost particle j is in the support domain of the real particle i, then the stress tensor could be assigned the same value as the considered real particle i. This approach can save a lot computational cost and still perform well compared with other methods [12].

$$\boldsymbol{\sigma}_j = \boldsymbol{\sigma}_i, \quad \text{if } j \text{ is in the support domain of particle } i. \tag{3.11}$$

For the symmetric or slip boundary problem, we still allocate a set of ghost particles outside the boundary. The physical properties of these ghost particles are the same as the real particles except for the velocity and the stress. The rule of assigning velocity and stress for ghost particles is different. The velocity component normal to the boundary of the ghost particles will have the opposite direction of the corresponding real particles, in order to prevent the real particles from moving across the boundary. The velocity component parallel to the boundary of the ghost particle is set to be the same as real particles. The stress tensor of ghost particles is set according to the corresponding real particles.

$$\sigma_g^{\alpha\beta} = \begin{cases} \sigma_r^{\alpha\beta}, & \text{if } \alpha = \beta, \\ -\sigma_r^{\alpha\beta}, & \text{if } \alpha \neq \beta, \end{cases} \tag{3.12}$$

where the subscripts g and r indicate the ghost and real particles respectively, the superscripts α and β are the indices of the stress tensor. The approach of ghost particles works well in straight boundary problems but with less accuracy in curve boundary problems [12].

3.4 Artificial viscosity

In isotropic elastodynamic problems including wave propagation, Klepaczko points out that an optimisation term must be involved in the computational simulation [45], which is assumed to be a proper artificial viscosity. This is because shock wave exists in most numerical problems and frequently occurred at the early stage of the analysis with relaxed initial conditions [12], such as the solid mechanics. The thickness of the shock wave is usually much smaller than the length scale in continuum mechanics problems, and it is impractical to simulate a macroscopic problem with such a small size particle to express all the information of the shock wave [14]. The shock waves always exist at the first stage of the simulation [12]. The discontinuity in the velocity, pressure, energy and density leads to the unphysical oscillations in the simulation.

In order to simulate the hydrodynamic problem, the numerical methods should have the capability of simulating shock waves. Otherwise this model may cause unphysical oscillations around the shock regions. The system will become unstable unless we introduce some special treatments into the the governing equations. The artificial viscosity is usually applied in order to smooth out the unphysical oscillation caused by the shock wave [14]. When the conservation of mass and momentum is applied on the shock wave front, the kinetic energy should be transferred into heat energy for the simulation. The transformation of energy can be expressed as a form of viscous dissipation. To improve the stability of the numerical analysis and dampen out the unphysical oscillation, introducing a form of viscosity into the momentum equation to dissipate the oscillations is necessary.

$$\frac{D\mathbf{v}_i}{Dt} = \sum_{j=1}^{N} m_j \left(\frac{\boldsymbol{\sigma}_j}{\rho_j^2} + \frac{\boldsymbol{\sigma}_i}{\rho_i^2} - \Pi_{ij}\mathbf{I} \right) \nabla \widetilde{\mathrm{W}}_{ij}, \tag{3.13}$$

where Π_{ij} is known as artificial viscosity.

In order to obtain the dissipation term Π_{ij}, it is necessary to mention the von Neumann-Richtmyer artificial viscosity, which was developed by von Neumann and Richtmyer in 1950 [115]. The developments of artificial viscosity in recent years are based on this earliest equation,

$$\Pi_1 = \begin{cases} a_1 \Delta x^2 \rho (\nabla \cdot \mathbf{v})^2, & \nabla \cdot \mathbf{v} < 0, \\ 0, & \nabla \cdot \mathbf{v} \geq 0, \end{cases} \tag{3.14}$$

where a_1 is an adjustable non-dimensional constant. The von Neumann-Richtmyer artificial viscosity is only considered during compression. It is worth noting that the von Neumann-Richtmyer artificial viscosity is actually a quadratic function of velocity divergence.

The von Neumann-Richtmyer artificial viscosity has been improved to be a linear artificial viscosity Π_2, which can dampen the unphysical oscillations further than the quadratic artificial viscosity term.

$$\Pi_2 = \begin{cases} a_2 \Delta x c \rho \nabla \cdot \mathbf{v}, & \nabla \cdot \mathbf{v} < 0, \\ 0, & \nabla \cdot \mathbf{v} \geq 0, \end{cases} \tag{3.15}$$

where a_1 is an adjustable non-dimensional constant and c is the wave speed.

Both the von Neumann-Richtmyer artificial viscosity Π_1 and linear artificial viscosity Π_2 are very popular in the application of FDM, FVM, FEM and etc., in order to remove numerical oscillations during the simulation [57]. Dissipation terms are added into the pressure term to diffuse the sharp variations in the simulation.

In early applications, the SPH method was used to simulate the problem of low or no dissipation. Then Monaghan developed artificial viscosity for the SPH method and applied it to simulate the shock problems [73]. This artificial viscosity Π_{ij} is the most popular applied so far in SPH literature. It has the capability of improving numerical stability and preventing penetration between particles during compression. The details of the equations are as follows.

$$\Pi_{ij} = \begin{cases} \dfrac{-\alpha_\Pi c_{ij}\phi_{ij} + \beta_\Pi \phi_{ij}^2}{\rho_{ij}}, & \mathbf{v}_{ij}\cdot\mathbf{x}_{ij} < 0, \\ 0, & \mathbf{v}_{ij}\cdot\mathbf{x}_{ij} \geq 0, \end{cases} \tag{3.16a}$$

where we have,

$$\phi_{ij} = \frac{h_{ij}\mathbf{v}_{ij}\cdot\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2 + \varphi^2}, \tag{3.16b}$$

$$c_{ij} = \frac{c_i + c_j}{2}, \tag{3.16c}$$

$$\rho_{ij} = \frac{\rho_i + \rho_j}{2}, \tag{3.16d}$$

$$h_{ij} = \frac{h_i + h_j}{2}, \tag{3.16e}$$

$$\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j \qquad \mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j, \tag{3.16f}$$

where c is the wave speed of the material and can be calculated by $c = \sqrt{E/\rho}$; c_{ij}, ρ_{ij} and h_{ij} are the average wave speed, density and smoothing length between particles respectively; $\mathbf{x}_{ij}$ is the distance between particle i and j; $\mathbf{v}_{ij}$ is the difference in velocity between particles i and j; the factor $\varphi = 0.1h_{ij}$ is used to prevent numerical divergence as two particles move closer; α_Π and β_Π are constants, the term associated with α_Π is a bulk viscosity and the other term associated with β_Π, which is used to prevent particle interpenetration at a high Mach number, is similar to the von Neumann-Richtmyer artificial viscosity. The values of these two factors should be chosen depending on the practical situation. Monaghan adopted $\alpha_\Pi = 0.01$ and $\beta_\Pi = 0$ on the problem of free-surface flows [74], he also suggested $\alpha_\Pi = 1$ and $\beta_\Pi = 1$ would bring better results in most cases [73]. Chapter 5 adopts $\alpha_\Pi = 2.5$ and $\beta_\Pi = 2.5$ in the simulation since Libersky *et al.* advised these in SPH simulation of solid mechanics [52].

As Monaghan introduced a shear viscosity into the artificial viscosity for hydrodynamics problems, Hernquist and Katz developed another type of artificial viscosity depending on velocity divergence [39],

$$\Pi_{ij} = \frac{q_i}{\rho_i^2} + \frac{q_j}{\rho_j^2}, \tag{3.17a}$$

where we have,

$$q_i = \begin{cases} \alpha_\Pi h_i \rho_i c_i |\nabla\cdot\mathbf{v}_i| + \beta_\Pi h_i^2 \rho_i |\nabla\cdot\mathbf{v}|^2, & \nabla\cdot\mathbf{v} < 0, \\ 0, & \nabla\cdot\mathbf{v} \geq 0, \end{cases} \tag{3.17b}$$

This thesis adopts the artificial viscosity developed by Monaghan to improve the numerical stability and smooth the unphysical oscillation.

3.5 Tensile instability

3.5.1 Reason for the instability

Swegle *et al.* first studied the numerical instability of the SPH method [105], which is referred to as tensile instability. The instability is considered to be caused by the property of the second derivative of the smoothing function and the sign of stress in [105]. Swegle *et al.* pointed out that neither artificial viscosity nor the time integration method can solve the tensile instability problem.

A two dimensional example has been simulated to show the instability in tensile situation but stable in compression. The problem domain is a two-dimensional square plate with a uniform initial stress, either compressive or tensile. In order to avoid the rebound of the wave, particles near the boundaries are fixed.

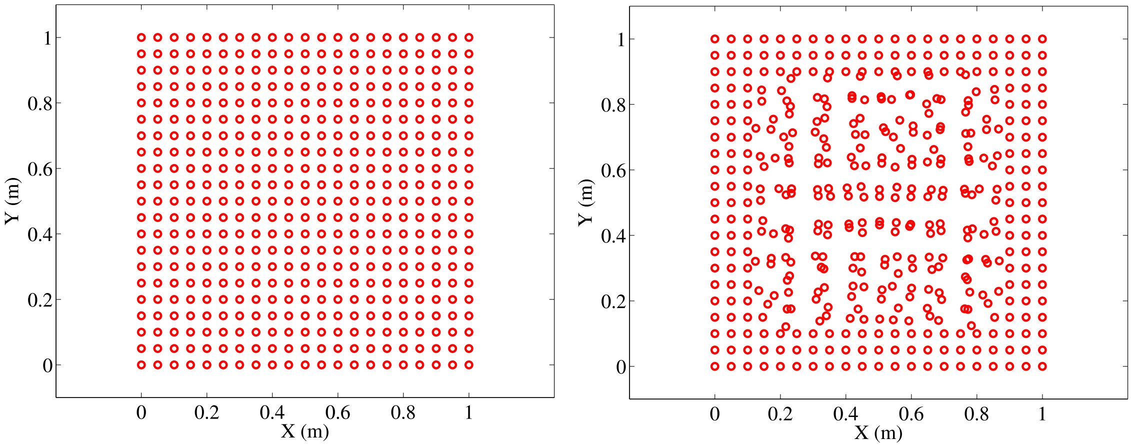
A very small perturbation velocity was applied on a particle in the center of the problem domain ($\mathrm{v} = 10^{-7}\ m/s$). The material chosen here is magnesium and the smoothing length is $h = 1.2\Delta p$, here Δp indicates the initial particle spacing. Figure 3.3 shows the particle distribution after 1 ms with the square plate under compression and tension, respectively (the stress value of both compression and tension is $\sigma = 3\ GPa$). It is easy to find that the particle distribution has no change when the stress is compressive, as shown in Figure 3.3(a). However the particles clumped together and the distribution became disordered in the results with the tension as shown in Figure 3.3(b).

Figure 3.4(a) shows the velocity of the perturbated particle with respect to time under the different stresses. The results show that the value of stress only influences time when the system becomes unstable under tension and changing the value of stress cannot solve the problem. However, the change of the velocity of the perturbated particle under compression is very small during the simulation.

Swegle *et al.* analysed the reason of tensile instability and identified the criterion of stability for the SPH simulation in one-dimensional space [105]. The criterion is based on the second derivative of the smoothing function and the stress state. The system condition which becomes unstable is expressed as follows,

$$\mathrm{W}''\sigma > 0, \tag{3.18}$$

where W'' is the second derivative of the smoothing function and σ is the stress. The SPH system becomes unstable when the product of W'' and σ is larger than zero. The second derivative of the smoothing function is shown in Figure 3.4(b). The stress is negative in compression and positive in tension. Figure 3.4(b) shows the stability condition for the cubic spline smoothing function. If the second derivative of the smoothing



(a) Compressive stress (b) Tensile stress

Figure 3.3: The particle distribution of the square plate after $t = 1$ ms under the compression and tension loading.

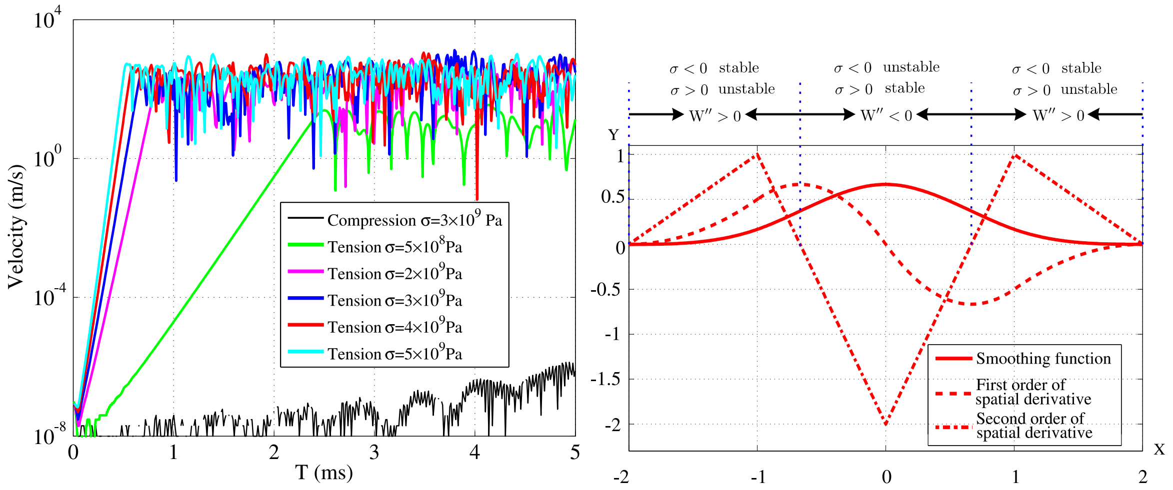
function is negative, the SPH system will become unstable under compression. If the second derivative is positive, the SPH system will become unstable under tension.

However, Robinson (2009) highlighted that this analysis did not fully explain the reason behind tensile instability [96]. He simulated the forced turbulence problem by using different smoothing functions and calculated the radial particle density function for particle pairs. Particle clumping occurred in the SPH simulation of forced turbulence problems with positive pressure by using the cubic spline kernel function. Analysis of the radial particle density function shows the relationship between the tendency of particles clumping and the spline point of the cubic spline kernel. Robinson performed the Fourier analysis of the forced turbulence simulation and obtained the conclusion that this instability is caused by the property of the kernel functions. In the research of Robinson, another smoothing function is used to avoid particle clumping in the forced turbulence problem, Wendland kernel. The Fourier transform of the Wendland kernel has also been analysed to strengthen the argument.

3.5.2 Solution of tensile instability

In order to solve the tensile instability, Dyka *et al.* developed an approach with stress points for the SPH method to overcome this numerical instability [24, 25, 94]. In this algorithm, the stress values are not calculated on the particle but on a set of virtual stress points to overcome the tensile instability. The stress points are located away from the SPH particles with a distance r_s and r_s is limited by Equation (3.19).

$$0 \leq \frac{r_s}{\Delta p} \leq 0.5. \tag{3.19}$$



(a) (b)

Figure 3.4: (a) The velocity change of the perturbation particle. (b) The second derivative of the cubic spline kernel.

When the $\frac{r_s}{\Delta p}$ equals 0.5, the algorithm is the traditional SPH method since the stress points are located on the particle, shown in Figure 3.5. The stress rate of the stress points is calculated by the velocity of the neighbouring SPH particles and the acceleration of the SPH particles is computed by the stress on the neighbouring stress points. The stress calculation in this algorithm is less accurate than the traditional SPH method, but it can solve the tensile instability problem.

Since Robinson considered particle clumping caused by the property of the cubic spline function, a new smoothing function which is one of the family of radial interpolation functions $\psi_{l,k}$ was applied to simulate the forced turbulence problem. The radial interpolation function was developed by Wendland (1995), with the compact support

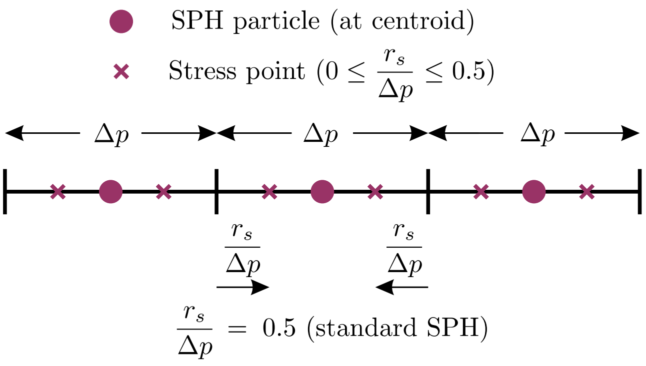


Figure 3.5: One dimensional case for the SPH method with stress points, when $\frac{r_s}{\Delta p} =$ 0.5, it becomes the standard SPH method.

and positive values in the support domain [117] ($\psi(R) = 0$ when $R > 2$).

$$A = (\psi(|\mathbf{r}_i - \mathbf{r}_j|)), \quad 1 \leq i, j \leq N. \tag{3.20}$$

The components l and k of these functions $\psi_{l,k}$ have the relationship that $l = [\![d/2]\!]+k+1$ ($[\![d/2]\!]$ means the integer part of $d/2$) and d indicates the dimensionality. Robinson chose one of these functions ($\mathrm{W}(R,h) = \psi_{3,1}$ in two dimension) as the smoothing function on simulation of the forced turbulence problem to remove the particle clumping [96] .

$$\mathrm{W}(R,h) = \frac{\beta_W}{h^d} \begin{cases} (2-R)^4(1+2R), & R \leq 2, \\ 0 & R > 2, \end{cases} \tag{3.21}$$

where α_d equal $\dfrac{7}{64\pi}$ in two- and three-dimensional space.

Monaghan introduced an artificial repulsive force into the SPH algorithm [75] to prevent neighboring particles clumping together under the tension. Gray *et al.* improved this algorithm to determine the artificial stresses (artificial repulsive force) by the signs of principal stresses [34], which is now the most popular method of addressing the tensile instability. According to the definition, this repulsive force should be increasing as the two neighboring particles move together. The momentum equation (3.13) can be replaced with,

$$\frac{D\mathbf{v}_i}{Dt} = \sum_{j=1}^{N} m_j \left(\frac{\boldsymbol{\sigma}_j}{\rho_j^2} + \frac{\boldsymbol{\sigma}_i}{\rho_i^2} - \Pi_{ij}\mathbf{I} + f_{ij}^n(\mathbf{R}_i + \mathbf{R}_j) \right) \nabla\widetilde{\mathrm{W}}_{ij}, \tag{3.22}$$

where factor n is the exponent dependent of term f_{ij} and $n = 4$ is demonstrated to be the optimum value by Monaghan when $h = 1.5\Delta p$ [75]. The repulsive force term f_{ij} is specified to represent the effect of distance between two neighboring particles $f_{ij} = \dfrac{\mathrm{W}_{ij}}{\mathrm{W}(\Delta p)}$. We know Δp is the initial spacing of the particles so that $\mathrm{W}(\Delta p)$ is a constant. This term ensures that when the distance of two neighbouring particles is smaller than Δp, the repulsive force term $(\mathbf{R}_i + \mathbf{R}_j)$ will be more effective.

According to Gray *et al.* [34], the components of artificial stress could be determined by the principal stress of the particle. Only two dimensional examples are shown herein, therefore the components of the artificial stress can be represented by standard transformation. The rotation angle θ_i can be calculated as,

$$\tan(2\theta_i) = \frac{2\sigma_i^{xy}}{\sigma_i^{xx} - \sigma_i^{yy}}. \tag{3.23}$$

The stress tensor can be transformed to the principal stress and the component can be expressed as,

$$\bar{\sigma}_i^{xx} = \cos^2\theta_i \sigma_i^{xx} + 2\sin\theta_i\cos\theta_i\sigma_i^{xy} + \sin^2\theta_i\sigma_i^{yy}, \tag{3.24a}$$

$$\bar{\sigma}_i^{yy} = \sin^2\theta_i\sigma_i^{xx} - 2\sin\theta_i\cos\theta_i\sigma_i^{xy} + \cos^2\theta_i\sigma_i^{yy}. \tag{3.24b}$$

The principal stress is applied to identify the diagonal components of the artificial stress,

$$\bar{R}_i^{xx} = \begin{cases} -\epsilon\dfrac{\bar{\sigma}_i^{xx}}{\rho_i^2}, & \bar{\sigma}_i^{xx} > 0, \\ 0, & \bar{\sigma}_i^{xx} \le 0, \end{cases} \tag{3.25}$$

where ϵ is a constant parameter with a value ranging from 0 to 1; and the minus sign indicates the cancellation of part of the stress while tension ($\sigma_i^{xx} > 0$ indicate the state of tension). We will choose the constant parameter ϵ to be 0.3 since Gray *et al.* suggested this is the best for elastic solid problem [34]. $\bar{R}_i^{yy}$ can be simply calculated by changing the subscript xx into yy of the equation (3.25). Then we need to transform the diagonal components of the artificial stress to the original coordinates,

$$R_i^{xx} = \bar{R}_i^{xx}\cos^2\theta_i + \bar{R}_i^{yy}\sin^2\theta_i, \tag{3.26a}$$

$$R_i^{yy} = \bar{R}_i^{xx}\sin^2\theta_i + \bar{R}_i^{yy}\cos^2\theta_i, \tag{3.26b}$$

$$R_i^{xy} = (\bar{R}_i^{xx} - \bar{R}_i^{yy})\sin\theta_i\cos\theta_i, \tag{3.26c}$$

where $\mathbf{R}_j$ can simply be calculated by replacing the subscript i with j in Equations (3.23), (3.24), (3.25) and (3.26). This approach is only effective when the particles clump together unphysical under the tension.

Vignjevic *et al.* proposed a total Lagrangian formalism for the SPH method to remove the tensile instability [114, 113]. In this algorithm, the initial particle positions $\mathbf{x}(t_0)$ are used to evaluate the value of the smoothing function and the derivative during the simulation. Equations (3.1a) and (3.1b) are re-written into the form,

$$\frac{D\rho}{Dt} = -\rho^0\nabla^0\cdot\mathbf{v}, \tag{3.27a}$$

$$\frac{D\mathbf{v}}{Dt} = \frac{1}{\rho^0}\nabla^0\boldsymbol{\sigma} + \mathbf{g}, \tag{3.27b}$$

where the superscript 0 indicates the initial configuration. The SPH discretisation form of Equation (3.27) can be represented as follows,

$$\frac{D\rho_i}{Dt} = -\rho_i^0\left(tr\left(\sum_{j=1}^{N}\frac{m_j}{\rho_j^0}(\mathbf{v}_j - \mathbf{v}_i)\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0\right)\right). \tag{3.28a}$$

$$\frac{D\mathbf{v}_i}{Dt} = \sum_{j+1}^{N} m_j\left(\frac{\boldsymbol{\sigma}_j}{(\rho_j^0)^2} + \frac{\boldsymbol{\sigma}_i}{(\rho_i^0)^2}\right)\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0. \tag{3.28b}$$

The strain and spin rate (3.9a) and (3.9b) can be re-written into the total Lagrangian formalism,

$$\dot{\boldsymbol{\varepsilon}}_i = \frac{1}{2}\Big(\sum_{j=1}^{N}\frac{m_j}{\rho_j^0}(\mathbf{v}_j - \mathbf{v}_i)\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0 + \big(\sum_{j=1}^{N}\frac{m_j}{\rho_j^0}(\mathbf{v}_j - \mathbf{v}_i)\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0\big)^T\Big), \tag{3.28c}$$

$$\dot{\boldsymbol{\omega}}_i = \frac{1}{2}\Big(\sum_{j=1}^{N}\frac{m_j}{\rho_j^0}(\mathbf{v}_j - \mathbf{v}_i)\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0 - \big(\sum_{j=1}^{N}\frac{m_j}{\rho_j^0}(\mathbf{v}_j - \mathbf{v}_i)\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0\big)^T\Big), \tag{3.28d}$$

where the spatial derivative of the smoothing function is normalised to maintain the zero order consistency,

$$\nabla_{\mathbf{x}_i^0}\widetilde{\mathrm{W}}_{ij}^0 = \frac{\nabla_{\mathbf{x}_i^0}\mathrm{W}(|\mathbf{x}_i^0 - \mathbf{x}_j^0|)}{\sum_{j=1}^{N}\frac{m_j}{\rho_j^0}(\mathbf{x}_j^0 - \mathbf{x}_i^0)\nabla_{\mathbf{x}_i^0}\mathrm{W}(|\mathbf{x}_i^0 - \mathbf{x}_j^0|)} \tag{3.28e}$$

Equations (3.28c) and (3.28d) can be applied in Equations (3.8c) and (3.8d) to calculate the stress rate.

According to the work of Robinson, the reason for the tensile instability is related to the property of the smoothing function. However the Wendland smoothing function applied in Robinson's work [96] cannot solve the tensile instability in the solid problem stated here. The total Lagrangian formalism of the SPH method has the capability to eliminate the particle clumping in some problems [114], since the treatment focuses on the approximation of the smoothing function. However, the total Lagrangian formalism is not suitable for large deformation problem such as the oscillational beam in Section 5.3. In this thesis, the artificial stress will be applied to solve the tensile instability.

3.6 SPH system with special treatments

As mentioned in Section 3.2, the governing equations based on the conservation laws can be represented in the SPH approximation form, then the special treatments should be added into this system. This thesis only discusses one- and two-dimensional problems. Note that the substantial derivative equations of the velocity and density can be changed into ordinary partial equations by the SPH discretisation. Combining (3.22) with (3.8), the SPH discretisation in two-dimensional space can be reformulated in a compact system of ODE of this term,

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}\left(\mathbf{U}\right), \qquad t \in [0, T], \tag{3.29}$$

where $\mathbf{U} = \begin{bmatrix} \mathrm{v}^x & \mathrm{v}^y & \sigma^{xx} & \sigma^{yy} & \sigma^{xy} & \mathrm{x}^x & \mathrm{x}^y \end{bmatrix}^T$, x^x and x^y indicate the components of the location of a particle. The right-hand side $\mathbf{F}(\mathbf{U})$ should be calculated for each particle i,

$$\mathbf{F}(\mathbf{U}_i) = \begin{bmatrix} \sum_{j=1}^{N} m_j \left(\frac{\sigma_j^{xx}}{\rho_j^2} + \frac{\sigma_i^{xx}}{\rho_i^2} - \Pi_{ij}^{xx} + f_{ij}^n (R_i^{xx} + R_j^{xx}) \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial x} + \sum_{j=1}^{N} m_j \left(\frac{\sigma_j^{xy}}{\rho_j^2} + \frac{\sigma_i^{xy}}{\rho_i^2} + f_{ij}^n (R_i^{xy} + R_j^{xy}) \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial y} \\ \sum_{j=1}^{N} m_j \left(\frac{\sigma_j^{yx}}{\rho_j^2} + \frac{\sigma_i^{yx}}{\rho_i^2} + f_{ij}^n (R_i^{yx} + R_j^{yx}) \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial x} + \sum_{j=1}^{N} m_j \left(\frac{\sigma_j^{yy}}{\rho_j^2} + \frac{\sigma_i^{yy}}{\rho_i^2} - \Pi_{ij}^{yy} + f_{ij}^n (R_i^{yy} + R_j^{yy}) \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial y} \\ \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(D_{11} \mathrm{v}_j^x - D_{11} \mathrm{v}_i^x \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial x} + \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(D_{12} \mathrm{v}_j^y - D_{12} \mathrm{v}_i^y \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial y} \\ \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(D_{21} \mathrm{v}_j^x - D_{21} \mathrm{v}_i^x \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial x} + \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(D_{22} \mathrm{v}_j^y - D_{22} \mathrm{v}_i^y \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial y} \\ \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(D_{33} \mathrm{v}_j^y - D_{33} \mathrm{v}_i^y \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial x} + \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(D_{33} \mathrm{v}_j^x - D_{33} \mathrm{v}_i^x \right) \frac{\partial \widetilde{\mathrm{W}}_{ij}}{\partial y} \\ \mathrm{v}_i^x \\ \mathrm{v}_i^y \end{bmatrix},$$

where D_{ij} are the entries of the elastic matrix $\mathbf{D}$ for plane stress *i.e.*,

$$\mathbf{D} = \frac{E}{(1 - \nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix}$$

.

The solution procedure for the system (3.29) is completed when a time integration of the semi-discrete SPH equations is selected.

3.7 Discussion

The dynamic behaviour on solid will be considered in this research. Since both solids and fluids are continuous, mass conservation and Newton's second law are true in any type of continuum mechanics problem, the governing equations of predicting their motions are similar. Then the governing equations of the SPH system in solid dynamics is also based on the continuity and momentum conservations, which are similar to fluid dynamics. However, the constitutive law is different in solid dynamics.

Here, the constitutive model of a two dimensional plane is represented in the SPH approximation form. The acceleration and stress rate is calculated through the SPH method to model the motion of the system. It is worth noting that there are two types of problem in this plane analysis: plane stress and plane strain. The problem using plane stress is supposing the geometry of the problem domain to have one dimension much smaller than the others and the stress in that direction is zero. The problem using plane strain is for the geometry to have one dimension much larger than the other two dimensions and the strain in that direction is zero. Plane stress is adopted in this research to calculate the stress rate in the SPH system for two dimensional problems. Note that the governing equations in this thesis are limited in solving the linear elastic problems and the computational domain should be continuous. Neither non-linear problem nor discontinuous system (such as fracture problems) can be modelled by these equations.

Concluding remarks

In this chapter, the Euler and Lagrangian approaches have firstly been introduced since the governing equation can be derived based on these two different approaches. The governing equations of hydrodynamics have been represented in the SPH discretisation form. Special treatment for boundary conditions is also added into the system to obtain high accuracy near the slip and non-slip boundaries. In order to eliminate the numerical instability caused by the shock wave, artificial viscosity is included to smooth out the unphysical oscillation. In Section 3.5, the reason for the tensile instability has been investigated and different solutions have been introduced to remove the tensile instability.

With the constructed SPH system, Equation (3.29) is combined with a time integration algorithm to obtain the predicted solution for the motion of the system. This stage can be handled by any implicit ordinary differential equation (ODE) solver, since they are computationally without risk by virtue of their accuracy and linear unconditional stability. This allows for larger time steps in the integration process. However, due to the large set of linear system of algebraic equations at each time step, these methods may be computationally inefficient. As an alternative, we use a series of explicit methods. It should be stressed that the explicit method has to satisfy a stability condition, because explicit time stepping schemes evaluate explicitly the right-hand side of the equation (3.29). This stability criterion can be guaranteed by the Courant-Friedrichs-Levy (CFL) condition, which will be explained in Chapter 4. Different time stepping schemes will be also introduced in Chapter 4.

Chapter 4

Runge-Kutta Chebyshev scheme

Introduction

The SPH system (3.29) is to be solved for a time interval $(0, T]$ with given initial and boundary conditions. The information on each particle in the problem domain is updated by time integration. To develop the time integration, the time interval $[0, T]$ is discretised into several time steps Δt with

$$0 = t_0 < t_1 < t_2 < \cdots < t_{n-1} < t_n = T,$$

where the time step $\Delta t = t_n - t_{n-1}$.

In explicit hydrodynamic methods, the Courant-Friedrichs-Levy condition (CFL) plays an important role in time integration of the simulation. It is a necessary condition of the stability of the analysis when solving Partial Differential Equations (PDE). The CFL condition ensures that the computational domain of dependence in a numerical simulation contains the physical domain of dependence, which means that the propagation speed of the numerical simulation should be smaller than the speed of physical propagation. In other words, the size of time step in the simulation is limited by the CFL condition. Large time steps will cause instability of the numerical simulation, whereas, small time steps can bring high accuracy for the numerical simulation but require an increased computational cost.

This chapter discusses different time stepping schemes, Euler, predictor-corrector, Symplectic schemes and Classical Runge-Kutta methods. Then a new time integration method will be introduced, Runge-Kutta Chebyshev (RKC) scheme, to improve the performance of the SPH simulation.

4.1 Euler time stepping scheme

The Euler method is the simplest time integration scheme for explicit hydrodynamic methods. As mentioned previously, the time at the n^{th} step is denoted as t_n and the computational solution of the SPH system at the n^{th} step is presented as $\mathbf{U}(t_n)$ with the abbreviation $\mathbf{U}^n$. The Euler time stepping scheme is based on the Taylor series expansion. Expanding $\mathbf{U}(t_n)$ around $t = t_{n+1}$

$$\begin{aligned}\mathbf{U}(t_{n+1}) &= \mathbf{U}(t_n) + (t_{n+1} - t_n)\mathbf{U}'(t_n) + O((t_{n+1} - t_n)^2) \\ &= \mathbf{U}(t_n) + \Delta t\mathbf{U}'(t_n) + O(\Delta t^2). \qquad (4.1)\end{aligned}$$

Then the explicit Euler method can be expressed as,

$$\mathbf{U}(t_{n+1}) = \mathbf{U}(t_n) + \Delta t\mathbf{F}\left(\mathbf{U}(t_n)\right). \qquad (4.2)$$

It is worth noticing from equation (4.2) that an error has occurred in each step as a result of ignoring the truncation term $O(\Delta t^2)$ in the Taylor series. This is called the local truncation error (LTE) of the scheme. Note that the LTE is different from the global error (g_n), which is defined as the absolute value of the difference between the analytical and numerical solutions, i.e. ,$g_n = |\mathbf{U}(t_n) - \mathbf{U}^n)|$. However the analytical solution is unknown in most situations, as a result the global error cannot be calculated. If we ignore the round-off errors, it is reasonable to calculate global errors at the n^{th} time step t_n by multiplying n and the LTE, which means the error is accumulated at each time step in physical sense. Since the steps n are proportional to $\frac{1}{\Delta t}$ (because $n = \frac{T}{\Delta t}$), then g_n should be proportional to $\frac{LTE}{\Delta t} = (\Delta t)$. This means that the Euler time stepping scheme is a first-order method, which also implies that the time integration method where $LTE = O(\Delta t^k)$ is a $(k-1)^{th}$-order method.

In the explicit Euler method, the LTE equals $O(\Delta t^2)$ which is obviously related to the time step Δt. Therefore, the accuracy of the Euler time stepping is related to both the number and size of the time step. As mentioned in Section 4, when the time stepping scheme (4.2) evaluates explicitly the right-hand side of the equation (3.29), it has to satisfy a stability condition (CFL condition),

$$c\frac{\Delta t}{\Delta p} \leq 1, \qquad (4.3)$$

where $c = \sqrt{\frac{E}{\rho}}$ is the wave speed and Δp is the initial spacing between two particles. Notice that Δt should depend linearly on initial spacing between two particles and reciprocal of the wave speed. It can be observed from the restriction (4.3) that, either by decreasing the smoothing spacing or increasing the wave speed, the considered explicit scheme needs time steps Δt small enough to maintain its stability. In other words, the time step is limited by the CFL condition. The small time step will lead to a large number of steps in the same time duration compared with other time stepping schemes. Both of these mean the accuracy of Euler method is very poor.

4.2 Predictor-corrector scheme

Monaghan raised a time stepping scheme to predict the evolution of the SPH system [72], which is widely used in combination with the SPH method [31], predictor-corrector scheme. This scheme conserves the linear and angular momentum and has second-order errors in time and space ($O(\Delta t^2)$ and $O(h^2)$).

In this algorithm for the SPH method, the velocity will be corrected by the contribution of neighbouring particles,

$$\widetilde{\mathbf{v}}_i = \mathbf{v}_i + \sum_{j=1}^{N} \frac{m_j}{\rho_i j}(\mathbf{v}_j - \mathbf{v}_i)\mathrm{W}_{ij}. \tag{4.4}$$

The procedure of the scheme is first to calculate the field valuables ($\mathcal{U}^{\frac{1}{2}}$) at the half time step. Then the values ($\widetilde{\mathcal{U}}^{\frac{1}{2}}$) will be corrected by the slope computed at the half step values ($\mathbf{F}\left(\mathcal{U}^{\frac{1}{2}}\right)$) and finally calculate the field variables at the next time step ($\mathbf{U}(t_{n+1})$),

$$\begin{aligned} \mathcal{U}^{\frac{1}{2}} &= \mathbf{U}(t_n) + \frac{1}{2}\Delta t \mathbf{F}\left(\mathbf{U}(t_n)\right), \\ \widetilde{\mathcal{U}}^{\frac{1}{2}} &= \mathbf{U}(t_n) + \frac{1}{2}\Delta t \mathbf{F}\left(\mathcal{U}^{\frac{1}{2}}\right), \\ \mathbf{U}(t_{n+1}) &= (2\widetilde{\mathcal{U}}^{\frac{1}{2}} - \mathbf{U}(t_n)). \end{aligned} \tag{4.5}$$

4.3 Runge-Kutta time stepping scheme

4.3.1 Second and fourth order Rouge-Kutta

According to Section 4.1, the Euler time stepping scheme is a first-order method. In order to improve the performance of the SPH approximation, high-order accuracy methods should be developed to increase the size of time steps and the accuracy of time integration methods.

Assuming $t_{n+\frac{1}{2}} = t_n + \frac{1}{2}\Delta t$, and then applying a Taylor expansion series to the system (3.29),

$$\mathbf{U}(t_{n+1}) = \mathbf{U}(t_{n+\frac{1}{2}}) + \frac{\Delta t}{2}\mathbf{U}'(t_{n+\frac{1}{2}}) + \frac{\Delta t^2}{8}\mathbf{U}''(t_{n+\frac{1}{2}}) + \frac{\Delta t^3}{48}\mathbf{U}'''(t_{n+\frac{1}{2}}) + O(\Delta t^4), \tag{4.6a}$$

and

$$\mathbf{U}(t_n) = \mathbf{U}(t_{n+\frac{1}{2}}) - \frac{\Delta t}{2}\mathbf{U}'(t_{n+\frac{1}{2}}) + \frac{\Delta t^2}{8}\mathbf{U}''(t_{n+\frac{1}{2}}) - \frac{\Delta t^3}{48}\mathbf{U}'''(t_{n+\frac{1}{2}}) + O(\Delta t^4). \tag{4.6b}$$

It is easy to get the expression of $\dfrac{\mathbf{U}(t_{n+1}) - \mathbf{U}(t_n)}{\Delta t}$ by taking (4.6a) subtracting (4.6b) and then divide by Δt,

$$\frac{\mathbf{U}(t_{n+1}) - \mathbf{U}(t_n)}{\Delta t} = \mathbf{U}'(t_{n+\frac{1}{2}}) + \frac{\Delta t^2}{24}\mathbf{U}'''(t_{n+\frac{1}{2}}) + O(\Delta t^4), \tag{4.7}$$

where the term $\mathbf{U}'(t_{n+\frac{1}{2}})$ on the right-hand side can be expressed as (Euler method has been applied on the spatial derivatives here),

$$\mathbf{U}'(t_{n+\frac{1}{2}}) = \frac{1}{2}(\mathbf{U}'(t_n) + \mathbf{U}'(t_{n+1})). \tag{4.8}$$

Therefore, Equation (4.7) can be reformulated into,

$$\mathbf{U}(t_{n+1}) = \mathbf{U}(t_n) + \frac{\Delta t}{2}\Big(\mathbf{U}'(t_n) + \mathbf{U}'(t_{n+1})\Big) + O(\Delta t^3). \tag{4.9}$$

Here, the term $\mathbf{U}'(t_{n+1})$ on the right-hand side of Equation (4.9) has two different forms, explicit and implicit form.

$$\mathbf{U}'(t_{n+1}) = \mathbf{F}\left(\mathbf{U}(t_{n+1})\right), \tag{4.10}$$

where the term $\mathbf{U}(t_{n+1})$ on the right-hand side is the real solution implicit form. The explicit form of the term $\mathbf{U}'(t_{n+1})$ is presented as follows,

$$\hat{\mathbf{U}}(t_{n+1}) \;=\; \mathbf{U}(t_n) + \Delta t\mathbf{U}'(t_n) = \mathbf{U}(t_n) + \Delta t\mathbf{F}\Big(\mathbf{U}(t_n)\Big), \tag{4.11a}$$

where $\hat{}$ denotes the estimation solution of the field value in t_{n+1} step. Then we have,

$$\mathbf{U}'(t_{n+1}) \;=\; \mathbf{F}\left(\hat{\mathbf{U}}(t_{n+1})\right). \tag{4.11b}$$

In order to get the explicit Runge-Kutta method expression, we combine equations (4.9) and (4.11) to obtain the explicit trapezoidal method,

$$\hat{\mathbf{U}}(t_{n+1}) = \mathbf{U}(t_n) + \Delta t\mathbf{F}\Big(\mathbf{U}(t_n)\Big), \tag{4.12a}$$

$$\mathbf{U}(t_{n+1}) = \mathbf{U}(t_n) + \frac{\Delta t}{2}\left(\mathbf{F}\Big(\mathbf{U}(t_n)\Big) + \mathbf{F}\left(\hat{\mathbf{U}}(t_{n+1})\right)\right). \tag{4.12b}$$

This set of equations explains the procedure of the explicit second order Runge-Kutta scheme.

The procedure of the classical fourth-order Runge-Kutta (RK4) method to advance the solution from the time t_n to the next time t_{n+1} can be carried out as:

$$\begin{aligned}
\mathcal{U}^{(1)} &= \mathbf{U}(t_n), \\
\mathcal{U}^{(2)} &= \mathbf{U}(t_n) + \frac{\Delta t}{2}\mathbf{F}\big(\mathcal{U}^{(1)}\big), \\
\mathcal{U}^{(3)} &= \mathbf{U}(t_n) + \frac{\Delta t}{2}\mathbf{F}\big(\mathcal{U}^{(2)}\big), \\
\mathcal{U}^{(4)} &= \mathbf{U}(t_n) + \Delta t\mathbf{F}\big(\mathcal{U}^{(3)}\big), \\
\mathbf{U}(t_{n+1}) &= \mathbf{U}(t_n) + \frac{\Delta t}{6}\Big(\mathbf{F}\big(\mathcal{U}^{(1)}\big) + 2\mathbf{F}\big(\mathcal{U}^{(2)}\big) + 2\mathbf{F}\big(\mathcal{U}^{(3)}\big) + \mathbf{F}\big(\mathcal{U}^{(4)}\big)\Big),
\end{aligned} \tag{4.13}$$

where the abbreviation $\mathcal{U}^{(k)}$ indicates the value of $\mathbf{U}$ at the k^{th} stage of time step n.

This kind of explicit time integration scheme has become popular in computational fluid dynamics, [33]. The main feature of this method lies in the fact that (4.13) is a convex combination of first-order Euler steps which exhibit strong stability properties. Therefore, the scheme (4.13) is Total Variation Diminishing (TVD) and stable under the usual CFL condition equation (4.3).

4.3.2 General formulation of Runge-Kutta methods

In numerical simulations, the family of Runge-Kutta schemes plays an important role of application in temporal discretisation for the approximation of solutions of ordinary differential equations. The general formulation of Runge-Kutta schemes is developed by Press *et al.* [91].

In an s-stage Runge-Kutta method, the ordinary differential equations (ODE) system (3.29) should be first written into the following form,

$$\mathbf{F}\Big(\mathbf{U}(t_n)\Big) = \mathbf{F}\Big(\mathbf{U}(t_n), t_n\Big). \tag{4.14}$$

We can then write the general procedure of a RK method as:

$$\mathbf{U}(t_{n+1}) = \mathbf{U}(t_n) + \Delta t \sum_{k=1}^{s} b_k L_k, \tag{4.15a}$$

where k is the k^{th} stage of the method and s is the total number of stages, L_k indicates the slope of each stage and can be represented as follows;

$$\begin{aligned}
L_1 &= \Delta t\mathbf{F}\Big(\mathbf{U}(t_n)\Big),\\
L_2 &= \Delta t\mathbf{F}\Big(\mathbf{U}(t_n)+a_{21}L_1, t_n+c_2\Delta t\Big),\\
L_3 &= \Delta t\mathbf{F}\Big(\mathbf{U}(t_n)+a_{31}L_1+a_{32}L_2, t_n+c_3\Delta t\Big), \qquad (4.15\text{b})\\
\vdots &\quad \vdots\\
L_s &= \Delta t\mathbf{F}\Big(\mathbf{U}(t_n)+a_{s1}L_1+a_{s2}L_2+\cdots a_{s,s-1}L_s, t_n+c_s\Delta t\Big)
\end{aligned}$$

To identify a particular order of the method, the number of stages s should be provided as well as the coefficients a_{kl} (for $1 \leq l < k \leq s$), b_l (for $k = 1, 2, \cdots, s$) and c_k (for $k = 2, 3, \cdots, s$). These three factors are called a Runge-Kutta matrix, weights and nodes respectively [41]. These factors are usually arranged in a Butcher tableau as shown in Table 4.1,

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \dots & b_{s-1} & b_s
\end{array}$$

Table 4.1: Butcher tableau for Runge-Kutta methods

The Runge-Kutta matrix $[a_{lk}]$ and nodes c_l should satisfy the condition,

$$c_l = \sum_{k=1}^{l-1} a_{lk} \quad \text{with} \quad l = 2, 3, \cdots, s. \qquad (4.16)$$

These conditions imply that there may be variations in certain orders of Runge-Kutta methods, for example the RK4 method. There is another version of the RK4 method called 3/8-rule [37]. The advantage of this version of the method is that the error coefficients are smaller than the popular version, however more floating point operations are required per time step. Its Butcher tableau is given in Table 4.2.

4.4 Runge-Kutta Chebyshev scheme

Difficulties often appear when the spectral radius of the Jacobian of $\mathbf{F}$, $\partial\mathbf{F}/\partial\mathbf{U}$, have large eigenvalues. This may give rise to numerical stiffness. Thus, time integration

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{3} & \frac{1}{3} & & & \\
\frac{2}{3} & -\frac{1}{3} & 1 & & \\
1 & 1 & -1 & 1 & \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}
\end{array}
$$

Table 4.2: Butcher tableau for 3/8 rule RK4 method

schemes for (3.29) depend strongly on the spectral radius $\rho\left(\partial\mathbf{F}/\partial\mathbf{U}\right)$ and node refinements. For these reasons it is preferable that these schemes have to be either implicit or explicit with large stability regions. The current work considers the Runge-Kutta Chebyshev (RKC) method studied in many papers [19, 112, 20]. The RKC method has been designed for explicit time integration of systems of parabolic equations. To solve (3.29) the RKC scheme takes the form

$$
\begin{aligned}
\mathcal{U}^{(0)} &= \mathbf{U}(t_n)\,, \\
\mathcal{U}^{(1)} &= \mathbf{U}(t_n) + \tilde{\mu}_1 \Delta t \mathbf{F}_n^{(0)} \\
\mathcal{U}^{(k)} &= \mu_k \mathcal{U}^{(k-1)} + \nu_k \mathcal{U}^{(k-2)} + (1-\mu_k-\nu_k)\mathcal{U}^{(0)} + \tilde{\mu}_j \Delta t \mathbf{F}_n^{(k-1)} + \tilde{\gamma}_j \Delta t \mathbf{F}_n^{(0)}, \qquad 2 \le k \le s\,, \\
&\;\vdots \\
\mathbf{U}(t_{n+1}) &= \mathcal{U}^{(s)}\,,
\end{aligned}
\tag{4.17}
$$

where $\mathbf{U}(t_n)$ is the solution computed at time step t_n, $\mathbf{F}_n^{(k)}$ denotes the term $\mathbf{F}\big(t_n + c_k\Delta t, \mathcal{U}^{(k)}\big)$ and $\mathcal{U}^{(k)}$ are the internal vectors for RKC stages. The coefficients in (4.17) are available in an analytical form for arbitrary $s \ge 0$ from some others' work[19, 112]. For the convenience of the reader the formulation for these coefficients is included here. Consider the Chebyshev polynomial of the first kind of degree k;

$$
T_k(z) = \cos(k\arccos z), \qquad -1 \le z \le 1\,.
$$

Then

$$
\epsilon = \frac{2}{13}, \qquad q_0 = 1 + \frac{\epsilon}{s^2}, \qquad q_1 = \frac{T_s'(q_0)}{T_s''(q_0)}\,,
$$

$$
b_k = \frac{T_k''(q_0)}{(T_k'(q_0))^2}, \qquad (2 \le k \le s), \qquad b_0 = b_2, \quad b_1 = \frac{1}{q_0}\,,
$$

and

$$
\tilde{\mu}_1 = b_1 q_1, \qquad \mu_j = 2q_0\frac{b_k}{b_{k-1}}, \qquad \nu_k = -\frac{b_k}{b_{k-2}}, \qquad \tilde{\mu}_k = 2q_1\frac{b_k}{b_{k-1}}\,,
$$

$$
\tilde{\gamma}_k = -(1 - b_{k-1}T_{k-1}(q_0))\tilde{\mu}_k, \qquad (2 \le k \le s)\,.
$$

The coefficients c_k are

$$c_k = \frac{T_s'(q_0)}{T_s''(q_0)}\frac{T_k''(q_0)}{T_k'(q_0)} \approx \frac{k^2-1}{s^2-1} \quad (2 \le k \le s), \qquad c_0 = 0, \qquad c_1 = c_2, \qquad c_s = 1\,.$$

It should be pointed out that two criteria have been taken into consideration for the calculation of the above coefficients; (i) the real stability boundary, $\beta(s)$, has to be as large as possible to obtain good stability properties for parabolic equations, and (ii) the application of the method with an arbitrary number of stages should not damage the convergence properties. That is, the accumulation of local errors does not grow without boundaries. Observe that the number of stages s in our SPH method and the conventional RKC scheme varies with Δt such that, see [112],

$$s = 1 + \left[\!\left[\sqrt{1 + \frac{c\Delta t}{0.131\Delta p}} \right]\!\right], \tag{4.18}$$

where $[\![\, x \,]\!]$ denotes the integer part of x, $c = \sqrt{\frac{E}{\rho}}$ is the wave speed and Δp is the initial particle spacing.

It is easy to verify that this method can adapt the stages in one time step itself for a particular CFL value. The number of stages in one step is large when we have a large CFL value (a large Δt). As we know, the accuracy of the traditional Runge-Kutta method reduces as the number of stages in one time step increases. This is because an increased number of stages increasingly dissipates more simulation information to obtain smoother results. However, this problem does not exist in the Runge-Kutta Chebyshev method. The accuracy is not influenced when we use a large number of stages in one large time step. This will be shown by the numerical results in Chapter 5.

Concluding remarks

The SPH method should be combined with the time integration algorithm to obtain the predicted motion of the system in the simulation. As mentioned in Section 3.7, the integration stage can be achieved by the explicit method. In explicit hydrodynamic methods, the Courant-Friedrichs-Levy condition (CFL) plays an important role in time integration of the simulation. Different time stepping schemes are introduced to do the time integration for the SPH method and they will be then compared through the CFL values and accuracies in Chapter 5.

As with other explicit numerical hydrodynamic methods, one can apply standard time stepping schemes to integrate the discretised SPH equations; i.e. the Euler, predictor-corrector and RK4 schemes, etc. Euler is the traditional time stepping scheme. It is simple but produces low accuracy. The predictor-corrector scheme is currently one of the most popular time stepping algorithms in the SPH application, since it has higher accuracy and stability than the Euler [72]. However the time step size in the Euler and predictor-corrector schemes may be very small when the smoothing length is small in some cases. In this situation, the RK4 schemes show more stability for the numerical simulation and the capability of larger time steps, which may be larger than the time step estimated by CFL condition. However, increasing the number of stages per time step, increases the computational cost.

Compared with other time stepping methods which are currently widely used, the RKC method has the advantages of flexible size and stages of each time step. Verwer *et al.* has done the stability analysis for the RKC time-stepping scheme and shown the stability region for RKC is larger than the other classical methods in the Runge-Kutta methods family [112]. The RKC also has the properties that it can adapt the number of stages for different time steps itself. However, unlike the other high order accuracy schemes such as Total Variation Diminishing shock capturing technique in [49], the RKC method only has second order accuracy. The applications will be simulated to demonstrate the advantages of this method in Chapter 5.

Chapter 5

Application to pure elastodynamic problems

Introduction

In order to build a firm foundation for the application of the Runge-Kutta-Chebyshev SPH algorithm in solid mechanics, a set of benchmark problems, which have a theoretical solution, are simulated in this chapter to investigate the advantage of this algorithm in key areas of elastodynamics. Although the SPH method is usually used to solve dynamical system, in the present study we use the steady-state SPH results to be compared to the FEM applied to a static problem. The benefit of this is that when the RKC-SPH method is applied in a traditional dynamic problem, the results at any time interval can be shown. We first solve a one-dimensional shock wave problem and compare the results with other time stepping schemes. Then the compression loading on a two-dimensional plate with a circular hole are simulated to compare with the static result of the FE simulation. After these benchmark problems, large deformation problems are analysed here. The capability of solving large deformation problems is proved by the results of using the RKC-SPH method.

In Section 5.1, a one-dimensional shock wave problem is first simulated by the RKC-SPH method. Since it is easy to obtain the analytical solution in wave propagation problems. In this simulation different time stepping schemes, such as Euler, predictor-corrector and Fourth-order Runge-Kutta (RK4) integration schemes, are applied and compared with the performance of the RKC method. The algorithm is then applied to simulate the quasi-static limit problem and compare the results with the standard FE solutions and the SPH method with predictor-corrector scheme in Section 5.2. Although the SPH is a dynamic behaviour simulating method, the result of the RKC-SPH method is still better than the the FEM applied to this static problem. In this example, the stress field of the two-dimensional finite plate with a circular hole in the middle under compression is investigated. These two examples are used to demonstrate the performance of the RKC-SPH method.

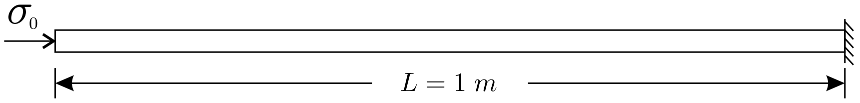


Figure 5.1: Compression loading of the one-dimensional magnesium bar.

The RKC-SPH method is shown to have the capability to solve large deformation problems, i.e. the two-dimensional oscillatory beam problem in Section 5.3. The stress fields and velocity vectors are presented to show the dynamic behaviour of the problem. The oscillation frequencies of different beams are presented to compare with the analytical frequency. In Section 5.4, the dynamic process of compression loading on a porous structure is simulated through the RKC-SPH method and the results show the stability of this algorithm.

5.1 One-dimensional wave propagation problem

5.1.1 Shock wave problem

In order to examine the performance of the different time stepping schemes for the SPH method in classical elastic problems, the problem of propagation of a shock wave in a one-dimensional elastic Magnesium bar is solved in this section.

The length of the magnesium bar is $L = 1m$ (see Figure 5.1), with material properties of $\rho = 1738\ kg/m^3$ and Young's modulus is $E = 45 \times 10^9\ Pa$. Initially, the bar is at rest with $\mathrm{v} = 0$ and $\sigma = 0$. The velocity at the right end of the bar is fixed ($\mathrm{v} = 0$) and we apply a compression stress on the left boundary $\sigma_0 = 8.8436 \times 10^6\ Pa$. This is a simple one-dimensional wave propagation problem which has a theoretical solution. Here, the velocity under the compression loading can be calculated as follows;

$$\mathrm{v}_0 = \frac{\sigma_0}{\sqrt{\rho E}} = 1. \tag{5.1}$$

The wave will propagate through the bar with the wave speed,

$$c = \sqrt{\frac{E}{\rho}}. \tag{5.2}$$

The wave propagates through the bar and bounds back at the fixed right end of the bar and the amplitude of the stress shock will be doubled when the wave arrives at the right end. The original wave and the reflective wave meet at the middle point of the bar and the stress value will be the double of the initial stress.

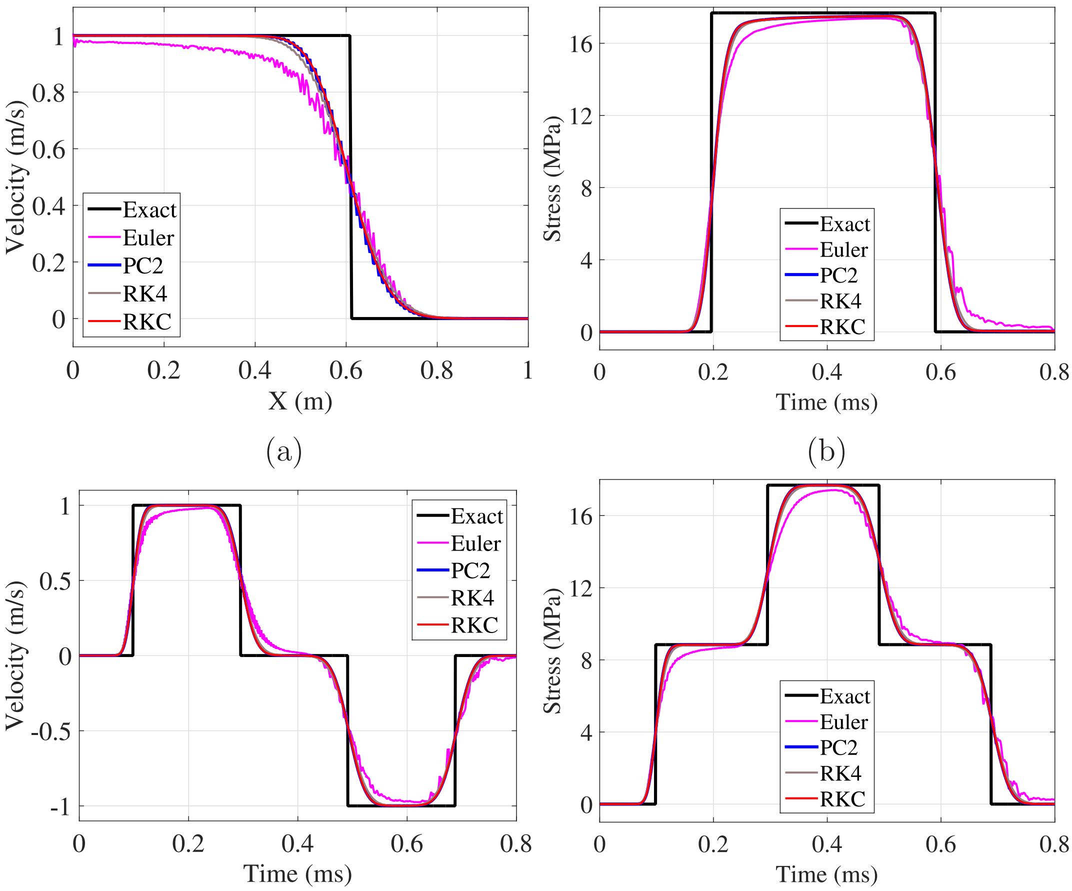
In this simulation, Euler, predictor-corrector, RK4 scheme and RKC scheme are applied in combination with the SPH method. A set of 250 particles is arranged in the analysis to simulate the dynamic behaviour of the elastic bar and we take $h = 1.5\Delta x$, here Δx is the initial spacing of the particles. We set CFL$= 0.6$ for all four time stepping schemes. Here PC2 is the abbreviation of the predictor-corrector scheme in all the figures. The number of stages in the RKC scheme is calculated by the equation (4.18) and for this text example we obtain the stage,

$$s = 1 + \left[\!\left[\sqrt{1 + \frac{c\Delta t}{0.131\Delta p}} \right]\!\right] = 3. \tag{5.3}$$

Note that the number of stages is fixed in the simulations presented in Chapter 5. Artificial viscosity (3.16) is applied for this analysis using,

$$\alpha_\Pi = 2.5, \quad \beta_\Pi = 2.5, \tag{5.4}$$

These values are suggested by Libersky *et al.* [52] in solid mechanics.

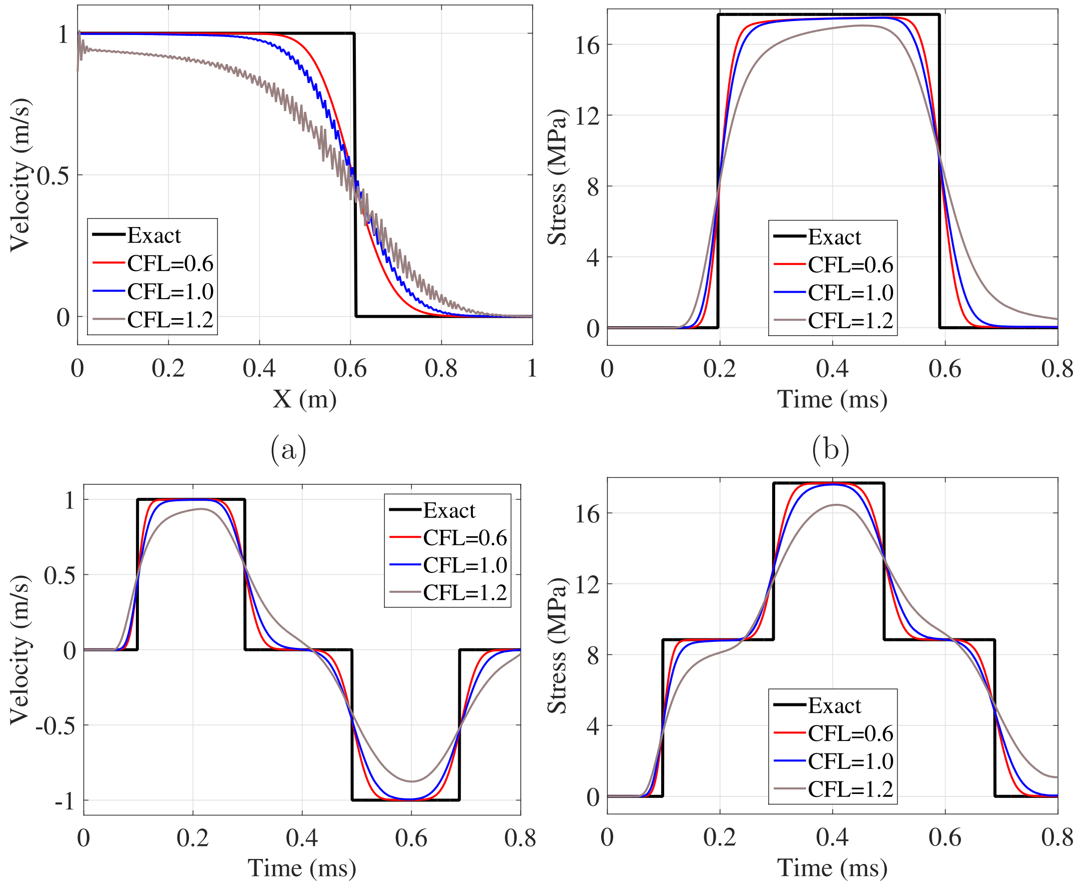


(a) (b)

(c) (d)

Figure 5.2: (a) Velocity distributions at $t = 1.2 \times 10^{-4}\ s$ along the bar with different time stepping schemes; (b) the stress at the end of the bar ($x = L$); the time evolution of velocity (c) and stress (d) at the mid point of the bar ($x = \frac{L}{2}$) using different stepping schemes.

At time $t = 1.2\times10^{-4}\ s$, the velocity distribution along the bar is displayed in Figure 5.2(a) and the stress at the end of the bar is shown in Figure 5.2(b). Figures 5.2(c) and 5.2(d) show the time evolution of velocity and stress at the mid point of the bar ($x = \frac{L}{2}$) respectively. These are obtained using the considered time stepping schemes and 250 particles. It is easy to see that there is some oscillation in the results obtained by the Euler and predictor-corrector schemes. The results of the RKC scheme only showed slight improvement compared to the RK4 scheme. However, the computational cost of the RKC is smaller than the RK4, since there are four stages in the RK4 time stepping scheme but only three stages in the RKC time integration algorithm. The simulation with the predictor-corrector scheme was found to become unstable when CFL> 0.6.

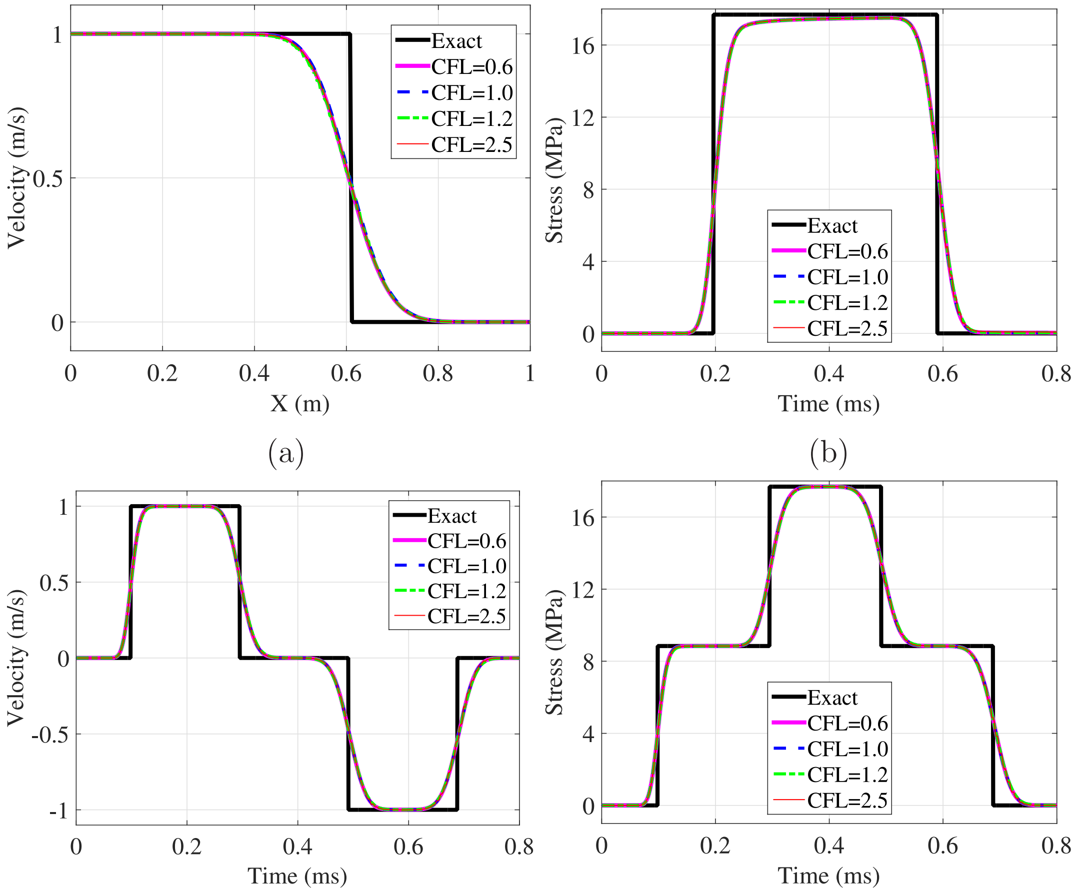


(a) (b)

(c) (d)

Figure 5.3: (a) Velocity distributions at $t = 1.2 \times 10^{-4}\ s$ along the bar with different time stepping schemes; (b) the stress at the end of the bar ($x = L$); the time evolution of velocity (c) and stress (d) at the mid point of the bar ($x = \frac{L}{2}$) using the RK4 time.

In order to further compare the performance of the RKC scheme and the RK4 scheme, this wave propagation problem is simulated using different CFL values for these two time stepping schemes. The results of the RK4 and RKC schemes are shown in Figure 5.3 and Figure 5.4.



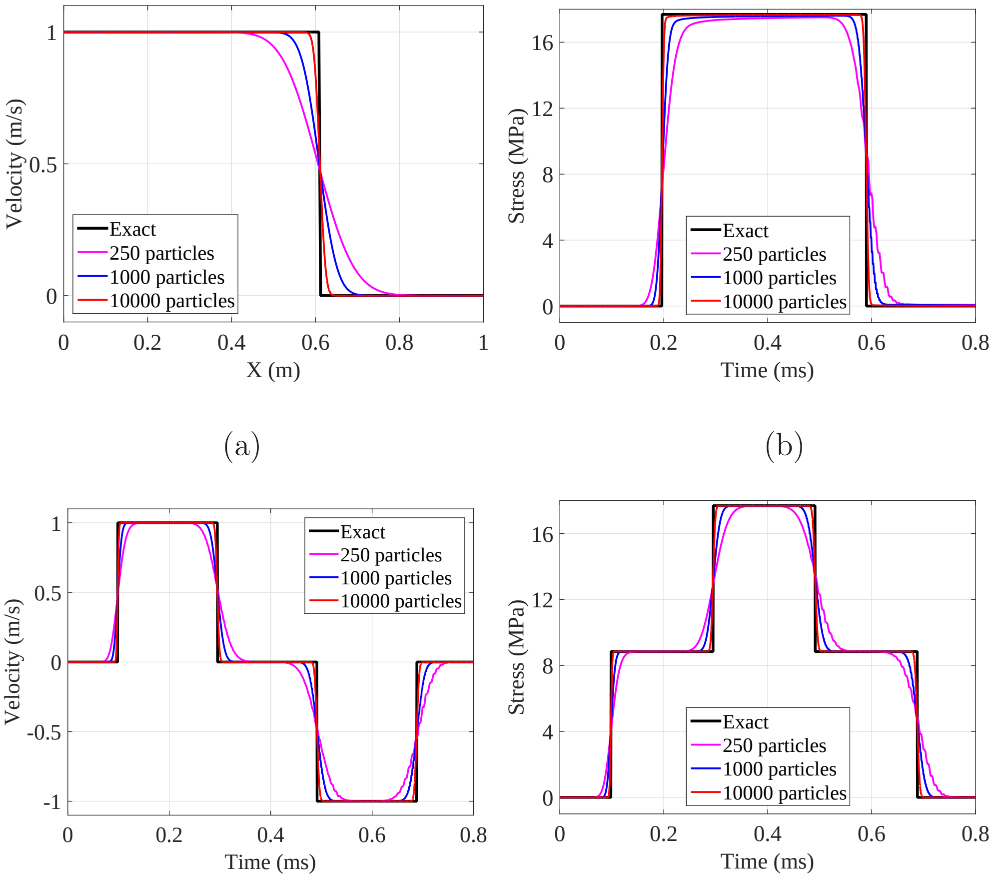
(a) (b)

(c) (d)

Figure 5.4: (a) Velocity distributions at $t = 1.2 \times 10^{-4}\ s$ along the bar with different time stepping schemes; (b) the stress at the end of the bar ($x = L$); the time evolution of velocity (c) and stress (d) at the mid point of the bar ($x = \frac{L}{2}$) using RKC method.

The results of the RK4 scheme in Figure 5.3 show that the oscillation and the dissipation increase as the values of CFL grow and the accuracy deteriorates as the CFL values increase. For the RK4 time stepping scheme, CFL$= 1.2$ is the upper-limit value, i.e. the solution becomes unstable when the CFL value is larger than 1.2.

However, it is easy to observe that the accuracy and numerical dissipation of the RKC time stepping scheme for the SPH method with different CFL are very similar in Figure 5.4. The stages of the RKC time stepping scheme are calculated for different CFL values. We obtain three stages for CFL$= 0.6$ and 1.0, four stages for CFL$=1.2$ and five stages for CFL$= 2.5$. It is worth noting that the CFL value cannot be increased without limitation when increasing the number of stages, because the RKC time stepping scheme is still an explicit scheme. A too large CFL value will lead to deterioration of the solution's accuracy. The errors for Euler, predictor-corrector, RK4 and RKC schemes of the SPH method in this shock-wave propagation simulation are shown in Table 5.1. Only the RKC time stepping scheme produces a reasonable error when CFL$= 2.5$.



(a) (b)

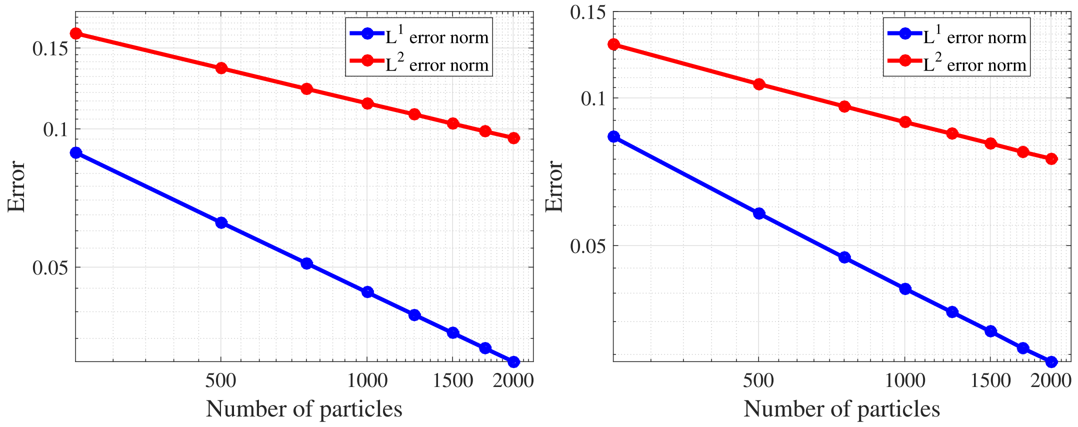
(c) (d)

Figure 5.5: (a) Velocity distributions at $t = 1.2 \times 10^{-4}$ s along the bar with different time stepping schemes; (b) the stress at the end of the bar ($x = L$); the time evolution of velocity (c) and stress (d) at the mid point of the bar ($x = \frac{L}{2}$) using RKC method with more particles.

	CFL = 0.6				CFL = 1.0			
	Euler	Predictor	RK4	RKC	Euler	Predictor	RK4	RKC
L^1 error	0.1469	0.1056	0.1005	0.0889	—	—	0.1260	0.0889
L^2 error	0.1943	0.1737	0.1710	0.1612	—	—	0.1899	0.1611
	CFL = 1.2				CFL = 2.5			
	Euler	Predictor	RK4	RKC	Euler	Predictor	RK4	RKC
L^1 error	—	—	0.2497	0.0888	—	—	—	0.0888
L^2 error	—	—	0.2632	0.1616	—	—	—	0.1614

Table 5.1: Errors of using the SPH method for solving shock-wave propagation with different time schemes at $t = 1.2 \times 10^{-4}$ s.

We then apply a large number of particles to simulate this wave propagation problem in order to obtain high accuracy results, as in Figure 5.5. The error rates are calculated to show the convergence of the RKC-SPH method in Figure 5.6. Figure 5.6(a) shows the errors of the velocity distribution at $t = 1.2 \times 10^{-4}$ s and Figure 5.6(b) shows the errors of the time evolution of the stress at the end of the bar. Here the rates of convergence in these two figures are 0.75 for L^1-error norm and 0.5 for L^2-error norm, which do not agree with the convergence rate of the SPH method in Section 2.4.2.



(a) (b)

Figure 5.6: (a) The L^1 and L^2 error norms of the velocity distribution at $t = 1.2 \times 10^{-4}$ s; (b) the L^1 and L^2 error norms of the time evolution of the stress at the end of the bar ($L = 1$ m).

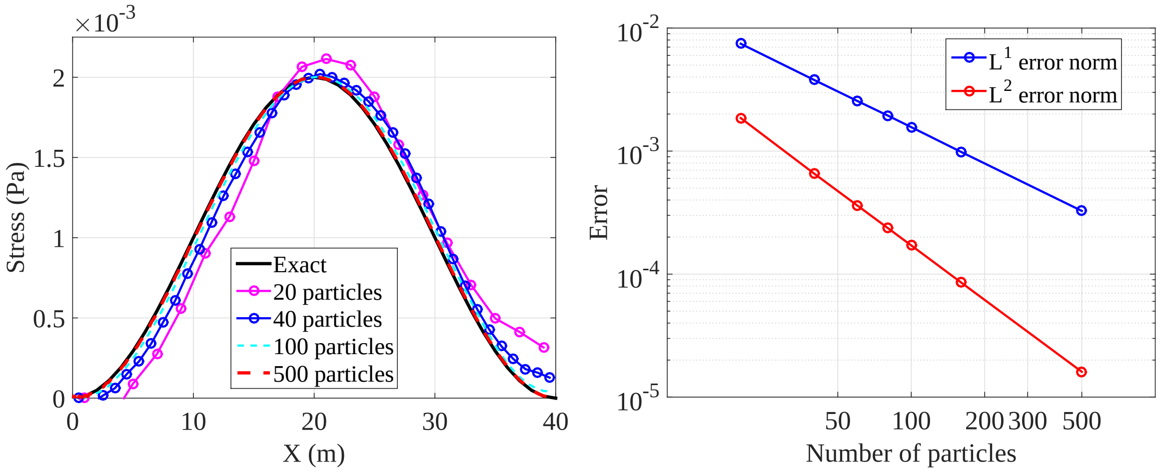
Since the shock wave problem stated in this thesis is a discontinuous solution, it is necessary to include artificial viscosity to smooth the unphysical oscillation. However, the artificial viscosity will dissipate the accuracy of the simulation. Therefore, the rates of convergence for the numerical results with artificial viscosity are reduced for the RKC-SPH method. In order to investigate the rate of convergence for the RKC-SPH method, another example with a smooth analytical solution in [49] is presented in Section 5.1.2. Rates of convergence of the RKC-SPH method obtained in this section will be compared with the theoretical order of the RKC-SPH method.

5.1.2 Convergence analysis

Similar to the problem in Section 5.1.1, a compression stress is loaded at the left end of a one-dimensional bar with unity Young's modulus E and density ρ. The bar is fixed at the right end and with the length $L = 40\ m$. In order to avoid a discontinuous exact solution, a forcing function of the compression stress in [49] is given by,

$$\sigma(t) = \zeta(\sin(\frac{\pi}{20}t - \frac{\pi}{2}) + 1), \quad t \geq 0, \tag{5.5}$$

where $\zeta = 0.001$ is the constant. There is no artificial viscosity in this analysis. Different numbers of particles (20, 40, 60, 80, 100, 160, 500 particles) are arranged to simulate this problem. Choosing CFL=1.0, the number of stages is calculated to be 3 by (4.18). Figure 5.7(a) represents the stress distribution along the bar at $t = 40\ s$ with different numbers of particles. The L^1 and L^2 error norms are shown in Figure 5.7(b). The rate of convergence of the RKC-SPH method obtained here is 1 for L^1 error norm and 1.5 for L^2 error norm without the influence of artificial viscosity.

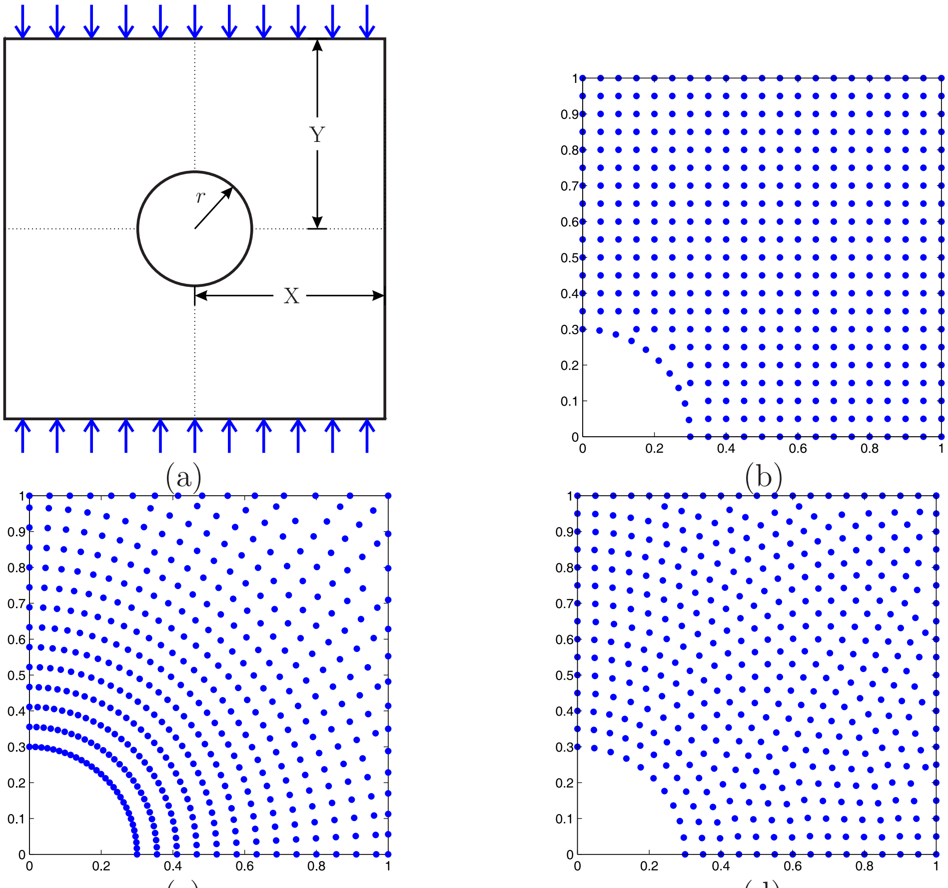


(a) (b)

Figure 5.7: (a) The stress distribution along the bar at $t = 40\ s$; (b) the L^1 and L^2 error norms of the stress distributioin at $t = 40\ s$.

5.2 Two-dimensional elastic plate

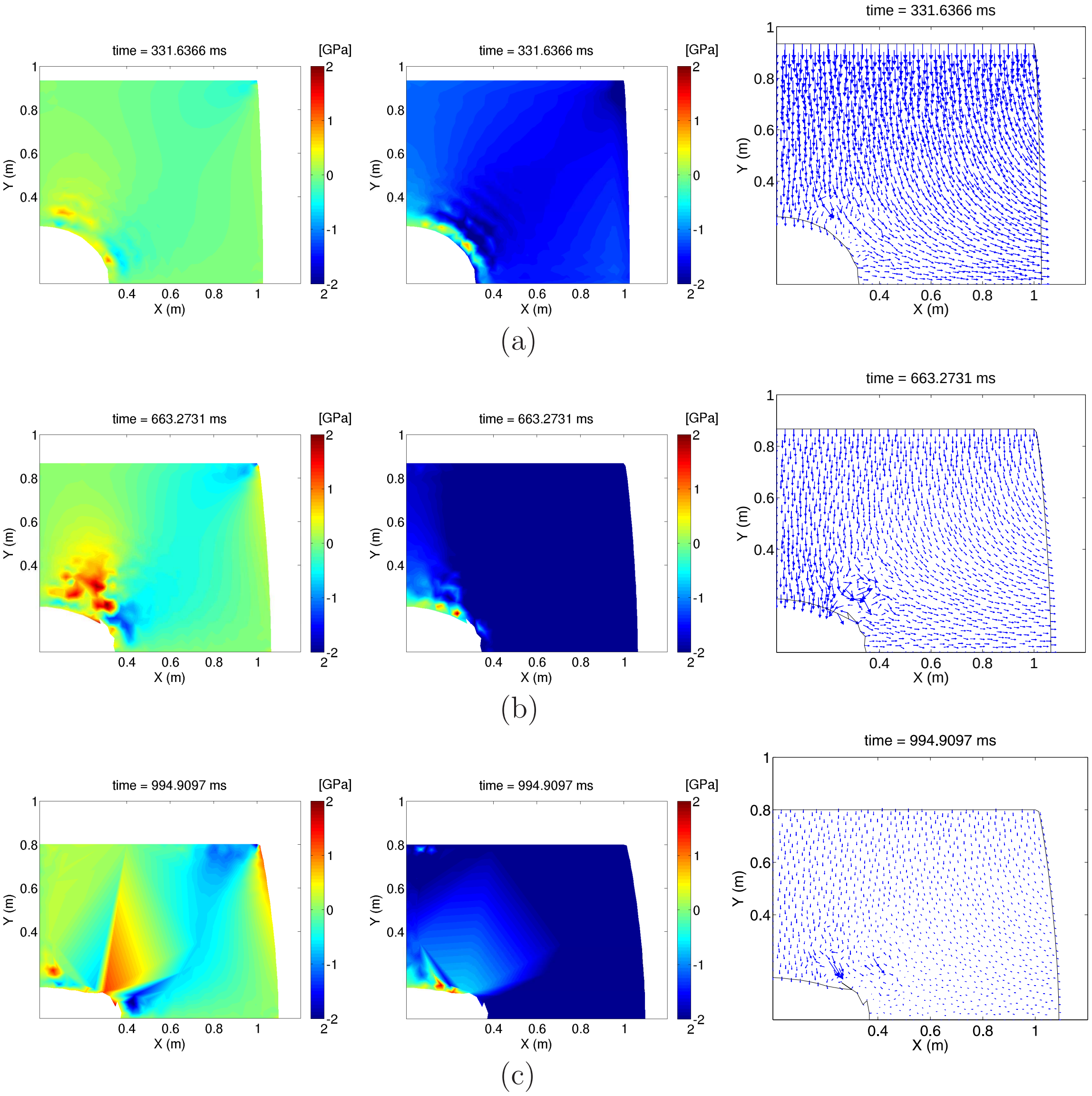
In this section a classical example in two-dimensional space is analysed using the RKC-SPH method since it has an analytical solution for the stress on the cross section. The compression loading on a two-dimensional plate with a circular void in the middle is simulated here. Since the SPH method is a particle-based method, the distribution of particles would influence the accuracy of the analysis. In this section, we first allocate three different distributions of particles for the compression loading on a plate with a circular void in the middle, in order to explore the influence of particle distribution. The plate is shown in Figure 5.8(a) and the analysis can be applied on a quarter of the plate because of the symmetric property. The plate material is still Magnesium and all the material constants are the Magnesium properties. The sizes of the plate are X=Y=2 m and $r = 0.3\ m$. The compression velocity is v = 2 m/s loaded on the top and bottom boundary and the left and right boundaries are free. The loading duration is 1 s. Take the center of the plate as the origin of coordinates, three different particle distributions for the quarter of the plate are considered, see Figure 5.8.



(a) (b) (c) (d)

Figure 5.8: (a) Whole problem domain. (b) Squared distribution. (c) Radial distribution. (d) Equally radial distribution.

These three distributions are squared distribution (Figure 5.8(b)), radial distribution (Figure 5.8(c)) and equally radial distribution (Figure 5.8(d)). The numbers of particles in the different distributions are slightly different, 897 for the squared, 922 for the radial and 916 for the equally radial. The initial particle spacing for squared and equally radial distributions are the same $\Delta p = 0.033\ m$. Since the change of σ_{xx} is very small during the whole process, the results of the σ_{xx} field are not displayed. The stresses (σ_{xy} and σ_{yy}) fields using a squared distribution at different times are shown in Figure 5.9.



(a)

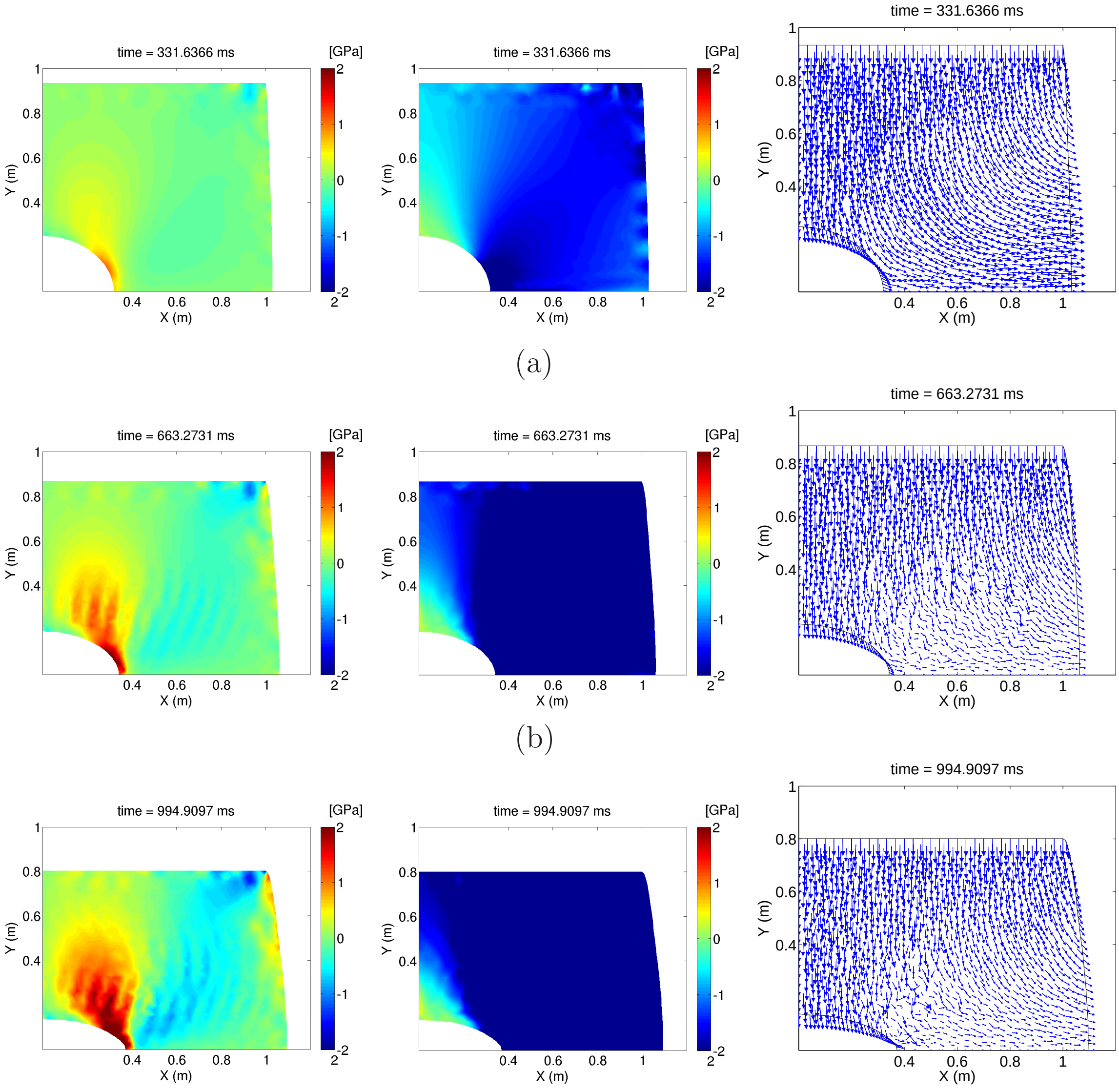
(b)

(c)

Figure 5.9: The stresses σ_{xy} (left column), σ_{yy} (middle column) and velocity $\mathbf{v}$ (right column) fields using squared distribution at different times. (a) $t_1 = 331.6336\ ms$. (b) $t_2 = 663.2731\ ms$. (c) $t_3 = 994.9097\ ms$.

It is clear from the results using square distribution, that there are some unphysical oscillations at the boundary of the circular hole and the velocity near the circular hole becomes disordered. This is because the particle distribution near the circular hole is not uniform enough to avoid the error accumulating during the simulation process.

The stresses σ_{xy} and σ_{yy} and the velocity $\mathbf{v}$ field using radial distribution at different times are shown in Figure 5.10. It is worth noting that the smoothing length of each particle here can be different because of the different initial particle spacing.



(a)

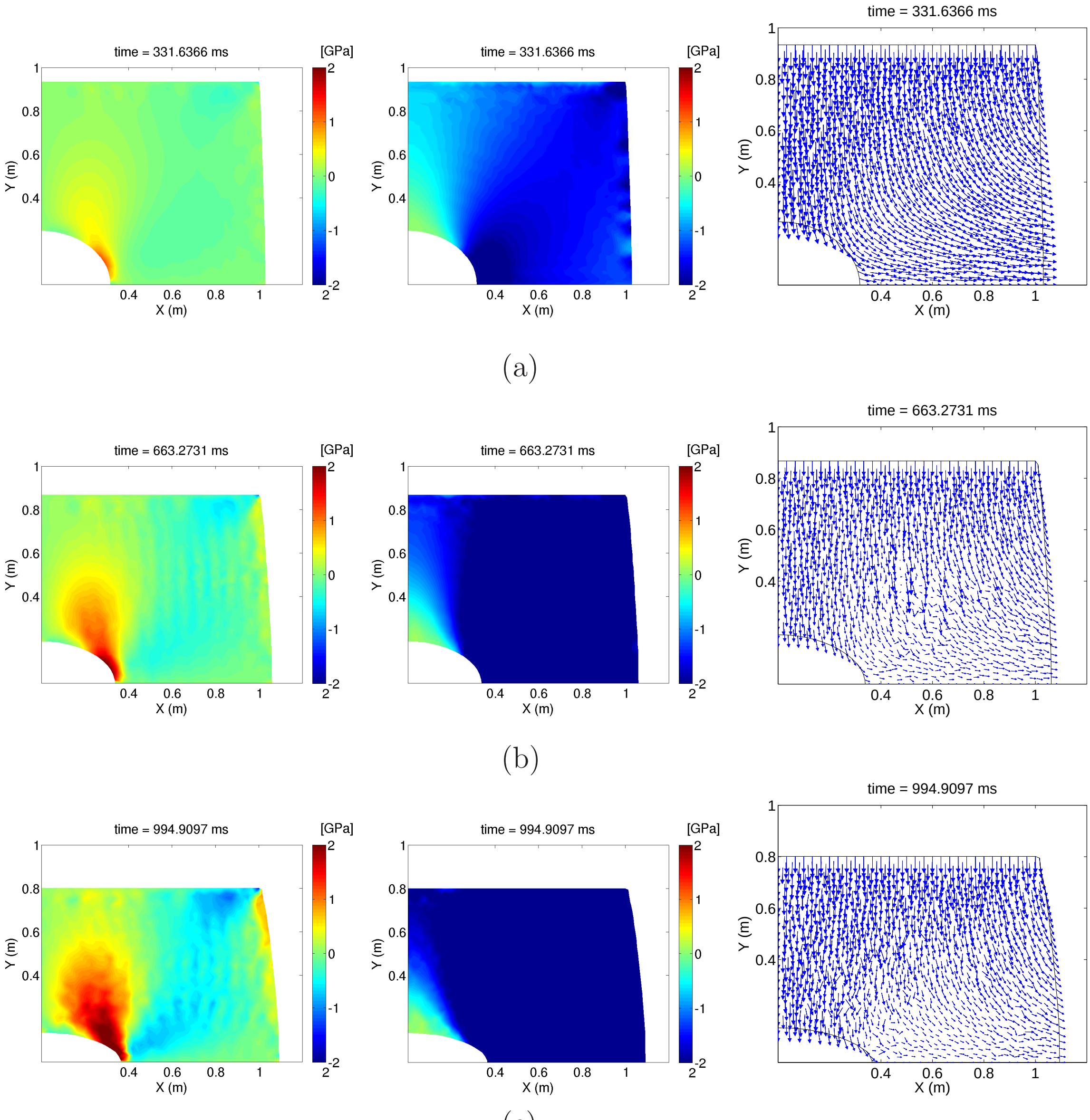
(b)

(c)

Figure 5.10: The stresses σ_{xy} (left column), σ_{yy} (middle column) and velocity $\mathbf{v}$ (right column) fields using radial distribution at different times. (a) $t_1 = 331.6336\ ms$. (b) $t_2 = 663.2731\ ms$. (c) $t_3 = 994.9097\ ms$.

In this simulation, the oscillation of the stress field occurs in the problem domain. There are still slightly disordered situations for the velocity inside the problem domain. This is because the smoothing length of each particle is different and the number of particles in the support domain is different. This will reduce the accuracy of the method if there is no special treatment (the solution of different smoothing lengths will be introduced in Chapter 6).

The stresses σ_{xy} and σ_{yy} and the velocity $\mathbf{v}$ field using equally radial distribution at different times are shown in Figure 5.11.



(a)

(b)

(c)

Figure 5.11: The stresses σ_{xy} (left column), σ_{yy} (middle column) and velocity $\mathbf{v}$ (right column) fields using equally radial distribution at different times. (a) $t_1 = 331.6336\ ms$. (b) $t_2 = 663.2731\ ms$. (c) $t_3 = 994.9097\ ms$.

Theoretically, the stress fields on an infinite plate with a circular hole under uniaxial far field compression loading σ^∞ can be predicted by (5.6), which is given by Ashby and Jones [2]. Take the polar coordinates ϱ and ϑ for the plate and the stress field can be presented as,

$$\sigma_\varrho(\varrho,\vartheta) = \frac{\sigma^\infty}{2}(1-\frac{r^2}{\varrho^2}) + \frac{\sigma^\infty}{2}(1-\frac{r^2}{\varrho^2})(1-3\frac{r^2}{\varrho^2})\cos 2\vartheta, \tag{5.6a}$$

$$\sigma_\varphi(\varrho,\vartheta) = \frac{\sigma^\infty}{2}(1+\frac{r^2}{\varrho^2}) - \frac{\sigma^\infty}{2}(1+\frac{r^4}{\varrho^4})\cos 2\vartheta, \tag{5.6b}$$

$$\sigma_{\varrho\vartheta}(\varrho,\vartheta) = \sigma_{\vartheta\varrho}(\varrho,\vartheta) = -\frac{\sigma^\infty}{2}(1-\frac{r^2}{\varrho^2})(1+3\frac{r^2}{\varrho^2})\sin 2\vartheta. \tag{5.6c}$$

Therefore, the normal stress σ_{yy} along the $y = 0$ can be calculated by σ_ϑ with $\vartheta = \frac{\pi}{2}$,

$$\sigma_{yy}(x,0) = \sigma^\infty\left(1+\frac{1}{2}\frac{r^2}{x^2}+\frac{3}{2}\frac{r^4}{x^4}\right). \tag{5.7}$$

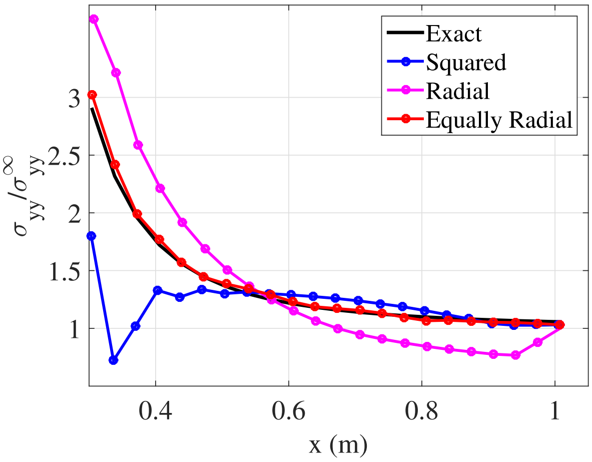


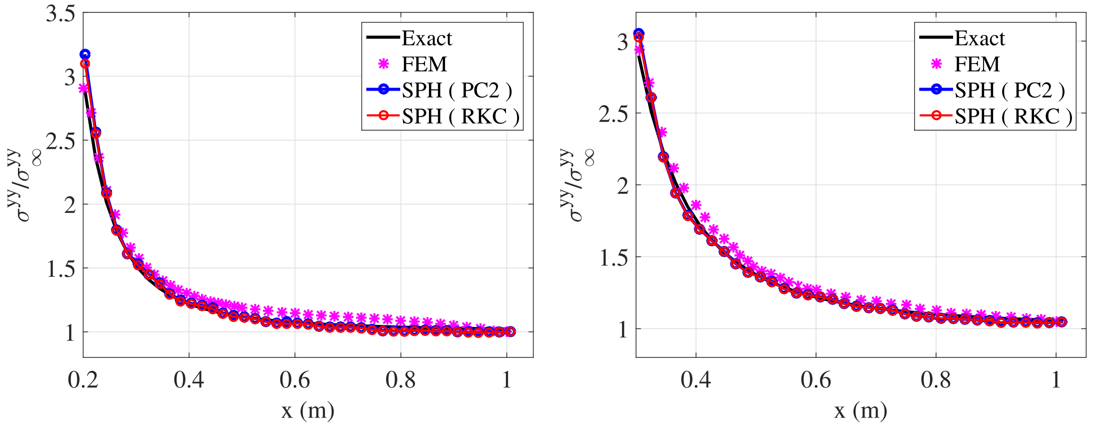
Figure 5.12: σ_{yy} along the cross section of the plane with circular hole ($y = 0$) using different particle distributions.

In this simulation, the compression stress is loaded on top and bottom of far field with $\sigma^\infty = 20\ MPa$ and the stresses on the each boundaries are set to be the analytical solution, which can be calculated by Equation (5.6). The σ_{yy} distributions along the cross section of the plane for each particle distribution are shown in Figure 5.12. Compared with the other two particle distributions, the equally radial distribution performs better than other distributions for the RKC-SPH method in this problem. This is because the initial particle spacing of the distribution is relatively more uniform than the other two distributions. This can reduce the error accumulation which leads to an unstable solution.

In order to further investigate the performance of the RKC-SPH method, equal radial distribution is used for the SPH method with both classical predictor-corrector

	$r = 0.2\ m$			$r = 0.3\ m$		
	FEM	SPH (PC2)	RKC-SPH	FEM	SPH (PC2)	RKC-SPH
Number of particles for whole domain	3222	2560	2560	3025	2459	2459
Number of particles on cross section	41	41	41	36	36	36
L^1 error norm	0.0521	0.0309	0.0304	0.0422	0.0272	0.0272
L^2 error norm	0.0537	0.0498	0.0442	0.0434	0.0367	0.0357

Table 5.2: Errors for the stress distribution σ_{yy} along the cross section of plane with circular hole ($y = 0$). Because this is a two-dimensional case and the number of particle is not linearly increasing when the particle spacing decreases, then the particle spacing is adopted to investigate the convergence rate of the method.



(a) $r = 0.2\ m$ (b) $r = 0.3\ m$

Figure 5.13: σ_{yy} along the cross section of plane with circular hole ($y = 0$) by using different methods.

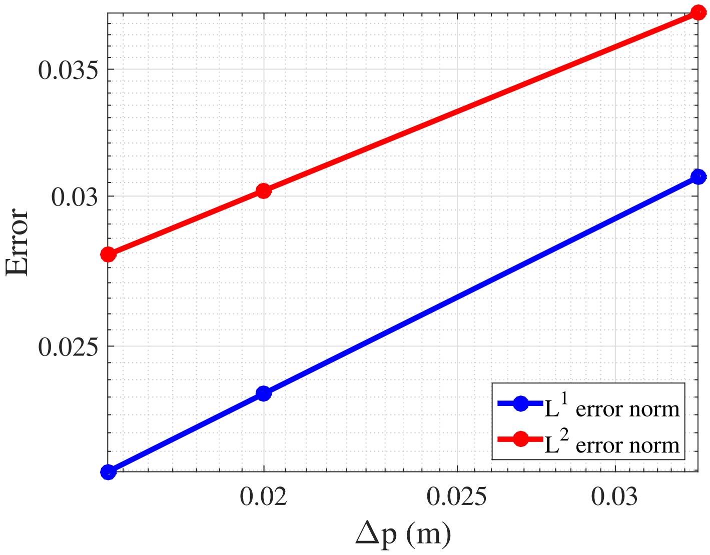


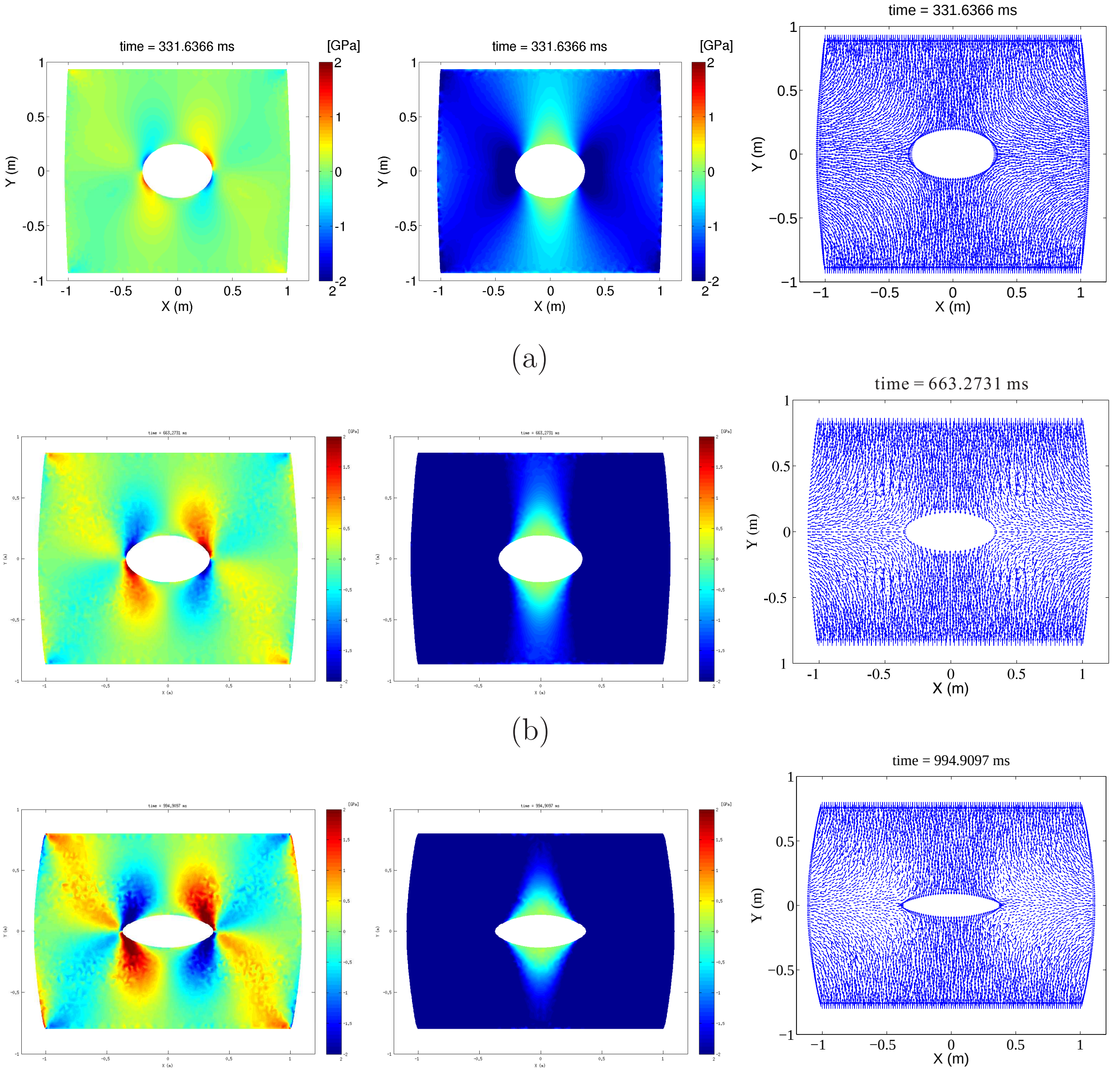
Figure 5.14: Errors for the stress σ_{yy} distribution along the cross section of plane with different number of particles.

and the RKC time stepping schemes. The CFL value in this problem is 2.5 in the RKC method with five stages and 0.6 in the Predictor scheme. The initial particle spacing is $\Delta p = 0.02$ here and Table 5.2 shows the different numbers of particles arranged for different void radius of the domain ($r = 0.2\ m$ and $r = 0.3\ m$). Then the traditional grid-based method (FEM) is also used to simulate this problem as a comparison with the RKC-SPH method. In order to be consistent with the SPH simulations, the same number of nodes (41 nodes for $r = 0.2\ m$ and 36 nodes for $r = 0.3\ m$) is arranged on the cross section, then the number of nodes for this domain is shown in Table 5.2. The FEM simulation is carried out by the ABAQUS software.

Note that the SPH method is a dynamic method and the results calculated by it are only snapshots of the whole dynamic process rather than a perfectly steady state but the FEM result is from a static simulation. In order to obtain results close to steady state, the stresses of particles over the last 10 time steps were averaged for the SPH simulations. It is also necessary to run the simulation for a long time until the shock waves have been dissipated. The comparison results of the FEM, SPH with predictor-corrector scheme and RKC-SPH method for different sizes circular holes in the plane are shown in Figure 5.13. The L^1 and L^2 error norms for different methods are shown in Table 5.2.

It is easy to see from Figure 5.13 and Table 5.2, the SPH method has the capability to achieve higher accuracy compared to the FEM. Compared to the SPH method with the predictor-corrector scheme, a larger CFL value in the RKC scheme can save computational cost and slightly higher accuracy is obtained in the RKC-SPH simulation since a larger number of time step will increase the error accumulation in the simulation.

Next, the equally radial distribution for the RKC-SPH method with more particles



(a)

(b)

(c)

Figure 5.15: The stresses σ_{xy} (left column), σ_{yy} (middle column) and velocity $\mathbf{v}$ (right column) fields using equi-radial distribution with more particles (3504) at different times. (a) $t_1 = 331.6336\ ms$. (b) $t_2 = 663.2731\ ms$. (c) $t_3 = 994.9097\ ms$.

(3504) is applied to simulate this problem ($r = 0.3\ m$). The stresses and velocity fields of the whole domain are shown in Figure 5.15. Here the results of the whole domain show the accuracy is proved to be higher than the traditional grid-based method, FEM. The L^1 and L^2 error norms for the stress σ_{yy} distribution along the cross section are presented in Figure 5.14. Note here the size of the particle spacing is the X axis since the number of particles does not linearly increase when decreasing the size of particle spacing in two-dimensional space. As mentioned in Section 5.1.1, the rate of convergence (0.5 for both L^1 and L^2 error norms) here is influenced by the artificial viscosity and accuracy for the RKC-SPH method. The real rate of convergence for the RKC-SPH method is 1.5 for L^2 and 1 for L^1 error norms as outlined in Section 5.1.2.

It is worth noting that there is another way to solve the problem caused by the non-uniform particle distributions in the SPH method. In this study, the mass of each particle is assumed to be constant in any distribution in an attempt to simplify the simulation process and focus on how to improve the SPH algorithm. If we adopt the voronoi area time the density to calculate the masses for particles, the problem caused by the nonuniform distribution near the boundary can be solved and similar results obtained (Figure 5.15).

5.3 Two-dimensional oscillatory beam

A large deformation problem which is similar to the example in [34] is modelled in this section. The large deformation problem of a thin two-dimensional beam with a fixed end is considered, see Figure 5.16. A set of perpendicular velocities v_y is loaded on the beam at the initial time. The length of the beam is L and the width of the beam is d.

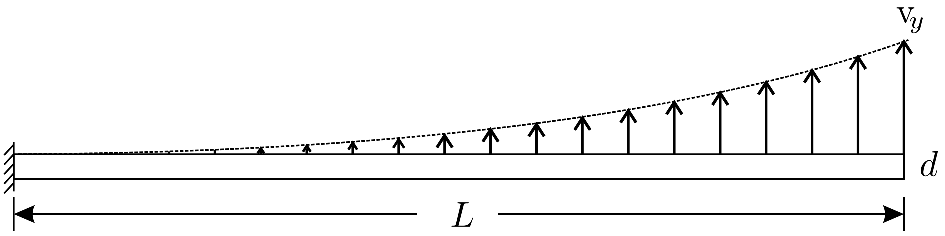


Figure 5.16: A two-dimensional beam fixed at left and free on right is pulled by a set of velocity.

The values of the velocity at each point on the beam can be expressed as follows,

$$\frac{\mathrm{v}_y}{c} = V_f \frac{M\left(\cos(\kappa x) - \cosh(\kappa x)\right) - N\left(\sin(\kappa x) - \sinh(\kappa x)\right)}{Q}, \tag{5.8a}$$

	$L = 0.5\ m$	$d = 0.1\ m$		$L = 0.5\ m$	$d = 0.05\ m$	
	$\omega_{theoretic}$	ω_{SPH}	error	$\omega_{theoretic}$	ω_{SPH}	error
$V_f = 0.05$	2066 Hz	1888 Hz	8.62%	1033 Hz	982 Hz	4.94%
$V_f = 0.02$	2066 Hz	1929 Hz	6.63%	1033 Hz	983 Hz	4.84%
$V_f = 0.01$	2066 Hz	1943 Hz	5.95%	1033 Hz	1000 Hz	3.19%
$V_f = 0.005$	2066 Hz	1943 Hz	5.95%	1033 Hz	1007 Hz	2.52%
$V_f = 0.002$	2066 Hz	1943 Hz	5.95%	1033 Hz	1007 Hz	2.52%
$V_f = 0.001$	2066 Hz	1943 Hz	5.95%	1033 Hz	1007 Hz	2.52%

Table 5.3: Oscillation frequency and period for analytical and SPH results.

where c is the wave speed of the material, V_f is a factor for the velocity calculation and the factor κ should satisfy the condition;

$$\cos(\kappa L)\cosh(\kappa L) = -1. \tag{5.8b}$$

It is easy to obtain $\kappa L = 1.875$ in this mode and other factors can be calculated as follows;

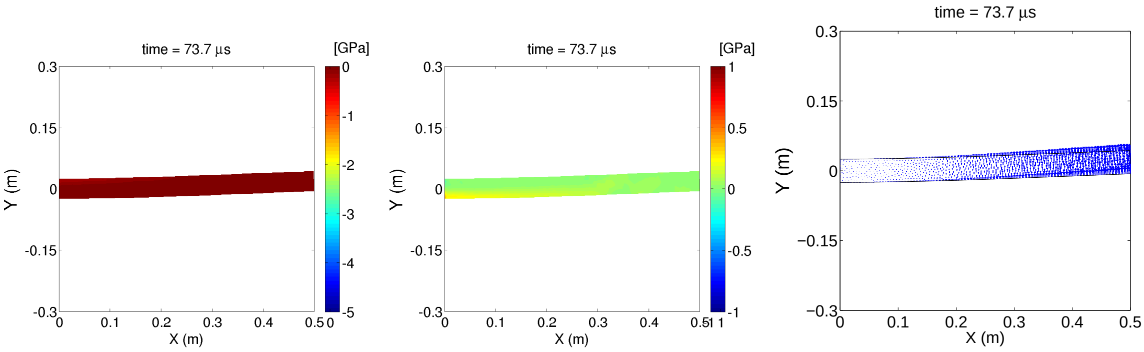
$$\begin{aligned} M &= \sin\kappa L + \sinh(\kappa L), \\ N &= \cos(\kappa L) + \cosh(\kappa L), \\ Q &= 2(\cos(\kappa L)\sinh(\kappa L) - \sin(\kappa L)\cosh(\kappa L)). \end{aligned} \tag{5.8c}$$

The equation of frequency ω calculating for a two-dimensional oscillatory beam with left end fixed and the other end free is given by Landau and Lifshitz [47];

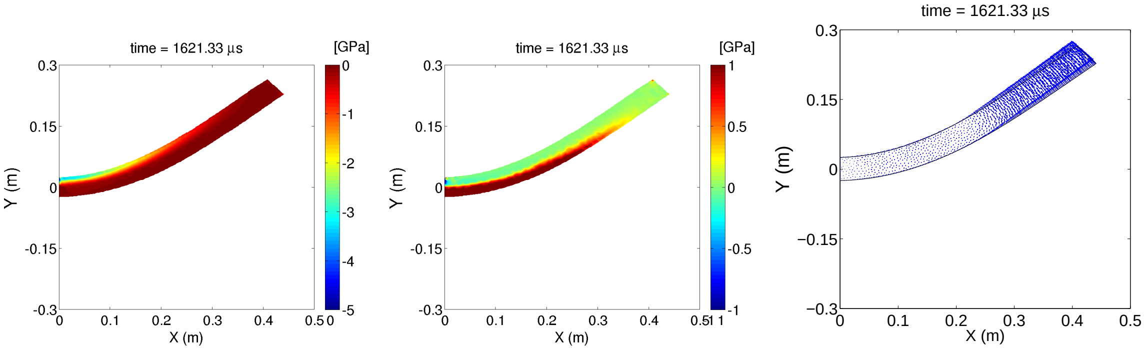
$$\omega = \sqrt{\frac{Ed^2\kappa^4}{12\rho}}, \tag{5.9}$$

We discretise the two-dimensional beam into a set of particles to simulate the oscillation process. The material of the beam here is still chosen to be Magnesium. Here we take $h = 1.5\Delta x$ (Δx is the initial spacing of the particles) and the initial particle distribution is placed on a Cartesian square grid. The factor $V_f = 0.02$. The size of this beam is $L = 0.5\ m$ and $d = 0.05\ m$. The principal stress fields σ_{xx}, σ_{yy} and the velocity $\mathbf{v}$ are shown in Figure 5.17 at different times.

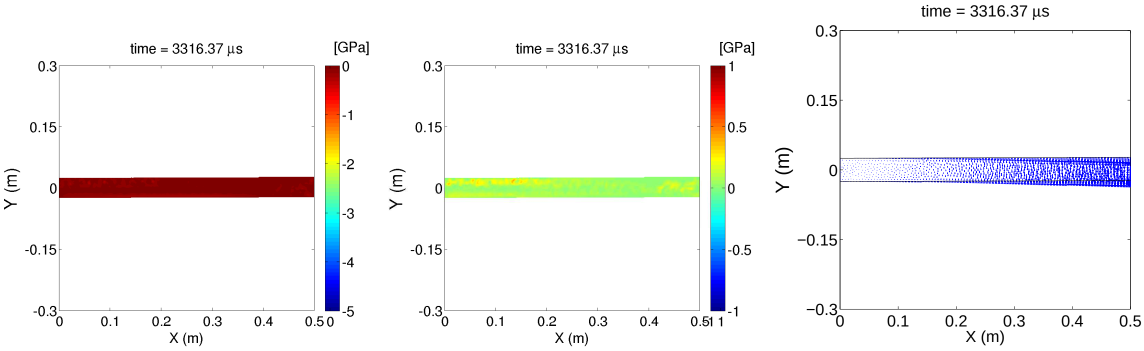
We then select different values for the factor V_f to simulate the oscillation process for the two-dimensional beams with different sizes to further determine the accuracy of this method. The oscillation frequency and period obtained by the RKC-SPH method are shown in Table 5.3 compared with the analytical frequency.



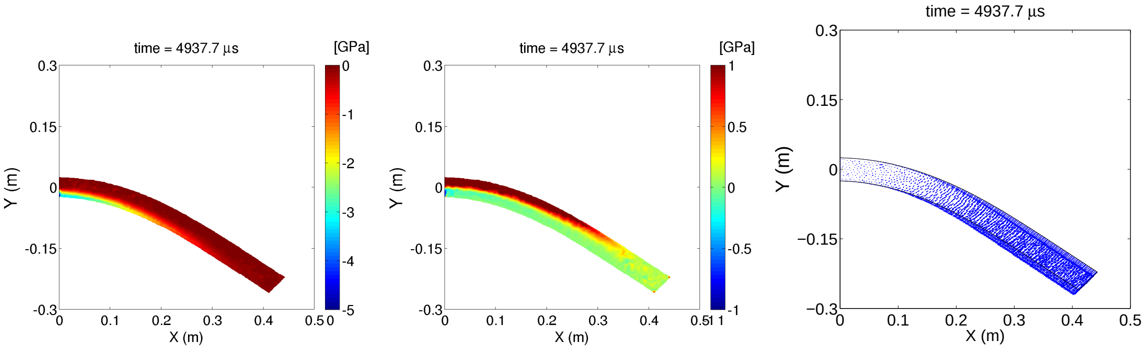
(a) $t_1 = 73.7\mu s$



(b) $t_2 = 1621.33\mu s$

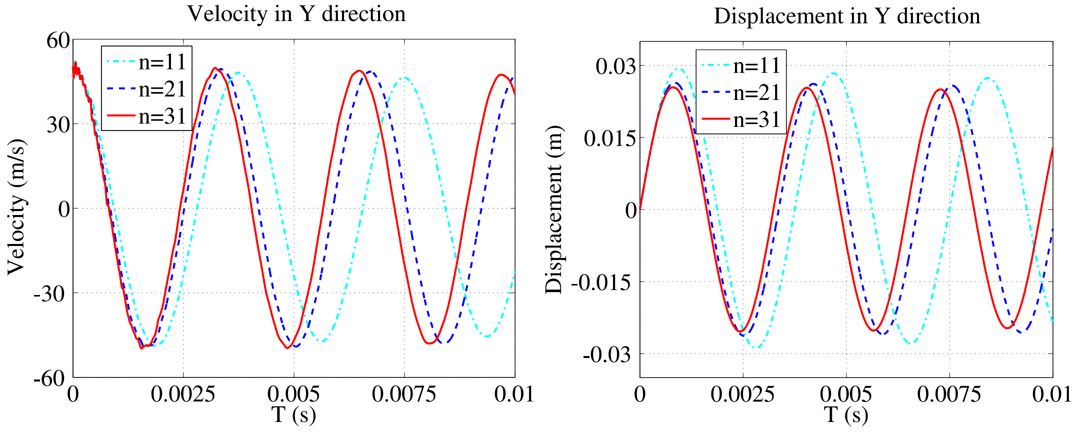


(c) $t_3 = 3316.37\mu s$



(d) $t_4 = 4937.7\mu s$

Figure 5.17: The principal stress fields σ_{xx} (left column), σ_{yy} (middle column) and velocity $\mathbf{v}$ (right column) of the oscillation beam ($L = 0.5\ m$ and $d = 0.05\ m$) using the RKC SPH method at different times t_1, t_2, t_3 and t_4.



(a) (b)

Figure 5.18: Comparative results of the velocity (a) and displacement (b) at the end of the beam with different numbers (n= 11, 21 and 31) of particles allocated on the width ($L = 0.5\ m$, $d = 0.1\ m$ and $V_f = 0.01$)

In order to determine the convergence of the RKC-SPH method, different numbers of particles are arranged to simulate a problem. The length of the beam is 0.5 m and width is 0.1 m, using three sets of particles with the initial spacing $\Delta p = 0.01\ m$, 0.005 m and 0.0033 m to simulate the oscillation process. The changes of velocity and displacement at the end of the beam are shown in Figure 5.18. The errors for the frequencies from the RKC-SPH method and analytical results are shown in Figure 5.19, here the L^1 and L^2 error norms for the frequency is the same. It is easy to see that the results of this method converge when we decrease the size of particles spacing and the rate of convergence is 1 here.

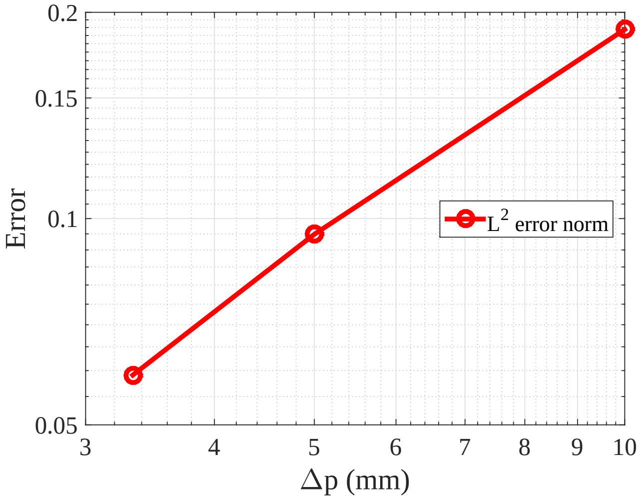
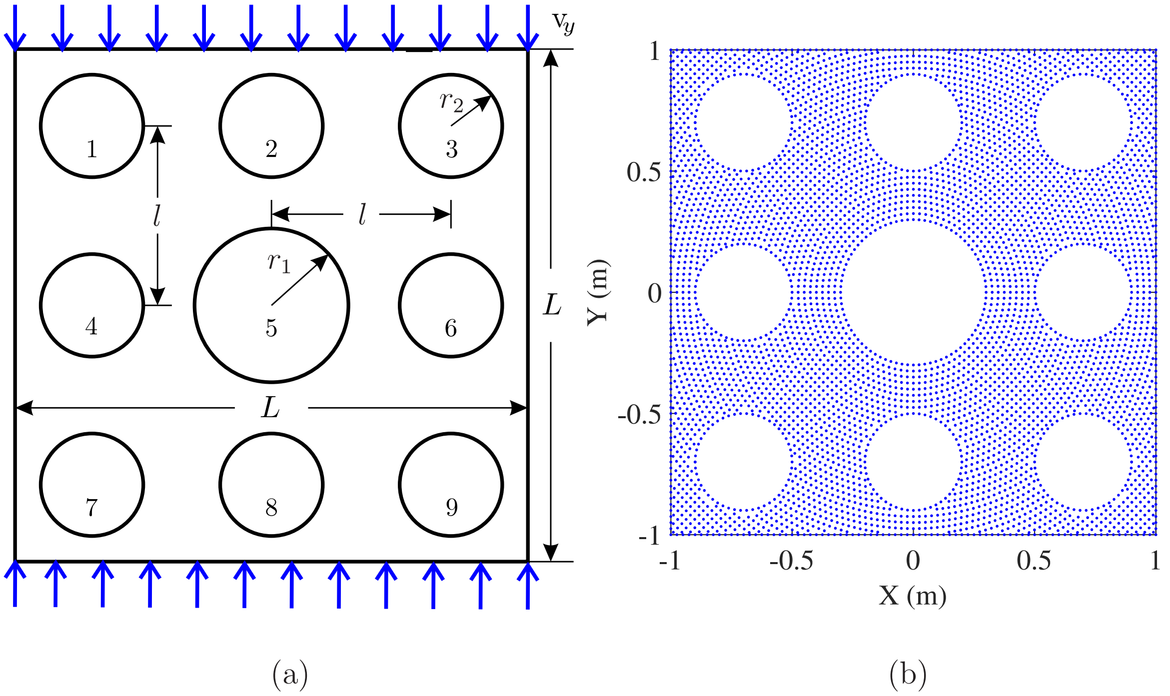


Figure 5.19: The errors for the frequencies from the RKC-SPH method results with different numbers of particles.

5.4 Elastodynamics in a porous plate under compression

The deformation problem in a porous structure is also modelled by the RKC-SPH method. The material is Magnesium and the whole plate is shown in Figure 5.20(a). The sizes of the plate are $L = 2\ m$, $r_1 = 0.3\ m$, $r_2 = 0.2\ m$ and $l = 0.7\ m$. There are nine circular holes inside the plate, the radius of middle one is r_1 and the rest of the holes have the same radius r_2.



(a) (b)

Figure 5.20: (a) The plate with nine circular holes inside. (b) The particle discretisation for the problem domain.

Note that the equi-radial distribution is only equally distributed for the problem with one void in the center. Here we select the equi-radial distribution to present the problem domain and arrange another 8 sets of particles on the boundary of the circular holes which are not in the middle as shown in Figure 5.20 (b). The loading velocity is $\mathrm{v}_y = 10\ m/s$ applied on the top and bottom surface and the duration is 0.02 s. After the loading process, the total strain of the plate will be $\varepsilon = 0.2$. 4616 particles are allocated for a quarter of the problem domain in this simulation, since the porous structure is a symmetric domain. The smoothing length is $h = 1.5\Delta p$. The factors of artificial viscosity are given by $\alpha_\Pi = 2.5$ and $\beta_\Pi = 2.5$. In order to distinguish each void, the nine voids have been numbered in Figure 5.20 (a). In this simulation, another set of particles is arranged on the circular boundaries of the voids except the middle voids 5, see Figure 5.20(b).

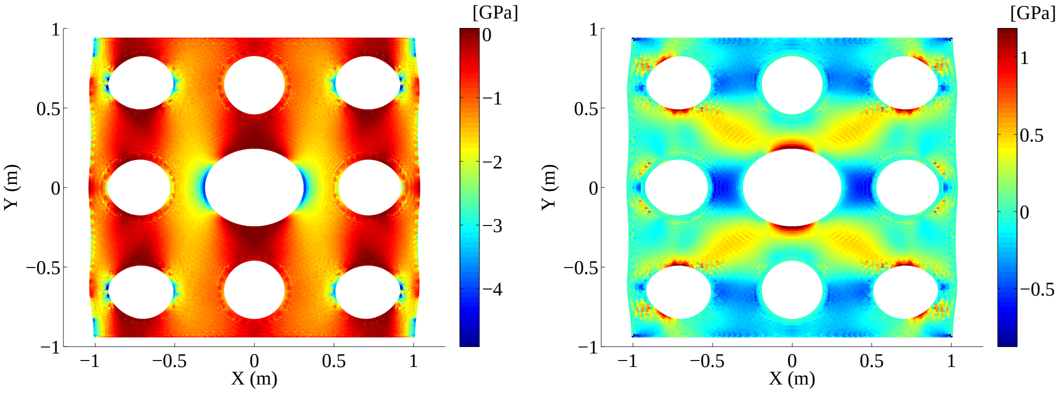
The principal stress fields of the compression plate are shown in Figure 5.21. As mentioned before, the particle is non-uniform distributed near the boundary of circular holes, this non-uniform particle distribution will cause some unphysical oscillation as shown in Figure 5.21. The stresses σ_{11} and σ_{22} always concentrate on the boundary of the four circular voids (1, 3, 7 and 9) near the corners and middle void 5. Most of the deformation occurs near these five circular holes and the strain on the two voids 4 and 6 is smaller than these five voids. The tension occurs on the upper and lower surface of the middle void 5 and there is very small strain on the voids 2 and 8. This property is very important in engineering applications, such as the design of the cellular solid. Since the total strain of the plate in Section 5.4 is $\varepsilon = 0.2$, this compression loading simulation on the porous plate is a large deformation problem. The results of this simulation show that the RKC-SPH method has the promising capability to solve the large deformation on the porous structure.

Concluding remarks

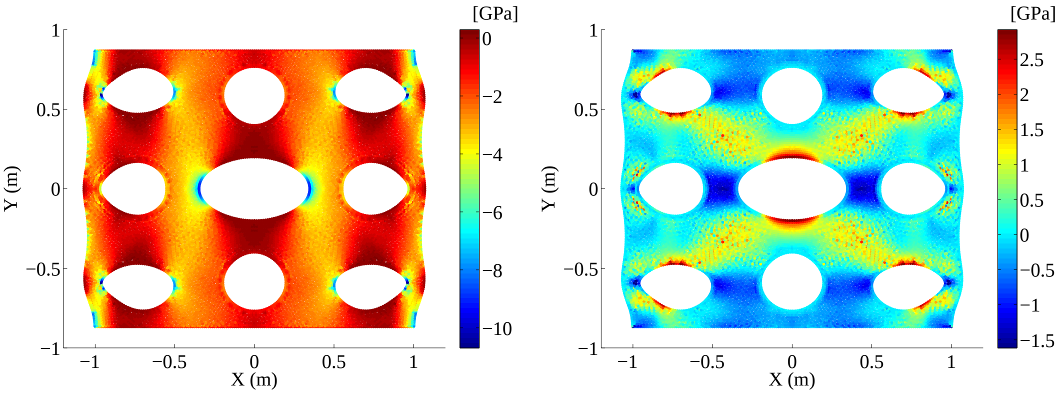
The applications of the SPH method have been analysed by using different time stepping schemes to investigate the advantages of the Runge-Kutta Chebyshev time integration algorithm in this chapter. Note that although the SPH method is a dynamic behaviour simulating technique, the performance of the method shown in the static problem still proved better than the traditional static grid-based method.

Two benchmark problems with theoretical solution have been analysed in one- and two- dimensions to show the advantages of the RKC-SPH method. A shock wave problem is first simulated by the Runge-Kutta-Chebyshev SPH method in a one-dimensional case in Section 5.1, comparing with the Euler, Predictor and RK4 time stepping schemes. The results show that the RKC-SPH method can bring higher accuracy results than applying other time stepping schemes in the SPH method and the size of time step can be larger to save the computational cost. The accuracy will not been influenced by the larger CFL value in the RKC method compared with the RK4 scheme. The influence of the particle distribution has been investigated by analysing the compression loading process on a two-dimensional elastic plate with a circular hole in the middle, as described in Section 5.2. And then the best distribution, equal radial distribution has been applied with the RKC-SPH method to compare the performance with the Predictor scheme and the FEM result. Even though the RKC-SPH method is a dynamic method, the results of quasi-static problem analysed by the RKC-SPH method show higher accuracy than the static FEM method and the SPH method with the Predictor time stepping scheme.

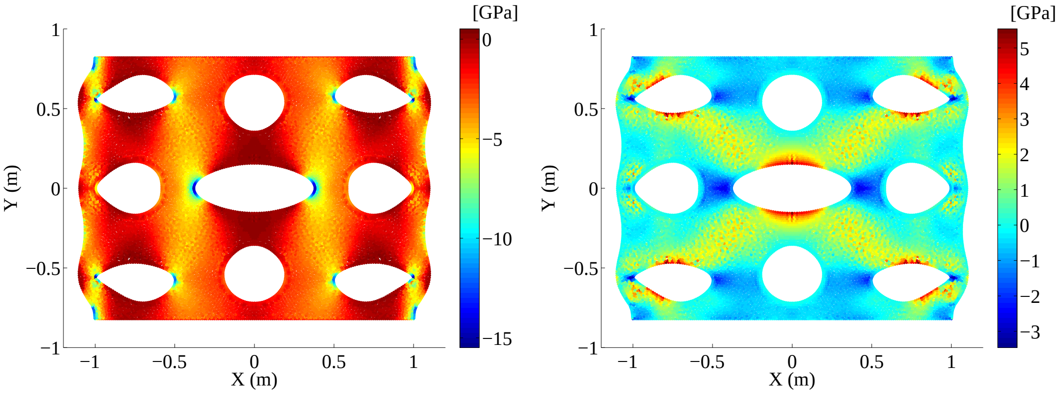
Then the RKC-SPH method is applied to simulate large deformation problems, a two-dimensional oscillatory beam in Section 5.3 and loading process on the high porosity structure in Section 5.4. The simulation results show the promising performance of the RKC-SPH on solving large deformation problems.



(a) $t_1 = 0.00667\ s$



(b) $t_2 = 0.01333\ s$



(c) $t_3 = 0.02\ s$

Figure 5.21: The stresses σ_{11} (left column) and σ_{22} (right column) fields using Runge-Kutta-Chebyshev SPH method at different times. (a) $t_1 = 0.00667\ s$. (b) $t_2 = 0.01333\ s$. (c) $t_3 = 0.02\ s$.

In this chapter, the advantages of the RKC-SPH method have been demonstrated by the test cases. Larger size of time step can be used by combining the RKC time stepping scheme with the SPH method to improve the efficiency and save computational cost. And the results show that the RKC-SPH method has the promising capability to solve large deformation problems. The adaptivity property of the RKC-SPH method can then be explored, such as the adaptive number of particles and spacing, which will be presented in Chapter 6.

Chapter 6

Time-space adaptive SPH method

Introduction

SPH is a purely Lagrangian particle method based on the kernel interpolation in which the problem domain is discretised into particles. As mentioned before, the SPH method has been applied on complex and nonlinear problems in areas from large-scale astrophysical systems to small-scale fluids and solids. Unlike the traditional grid-based methods, it is fully mesh-free and easy to simulate problems with complex physics and arbitrary geometries.

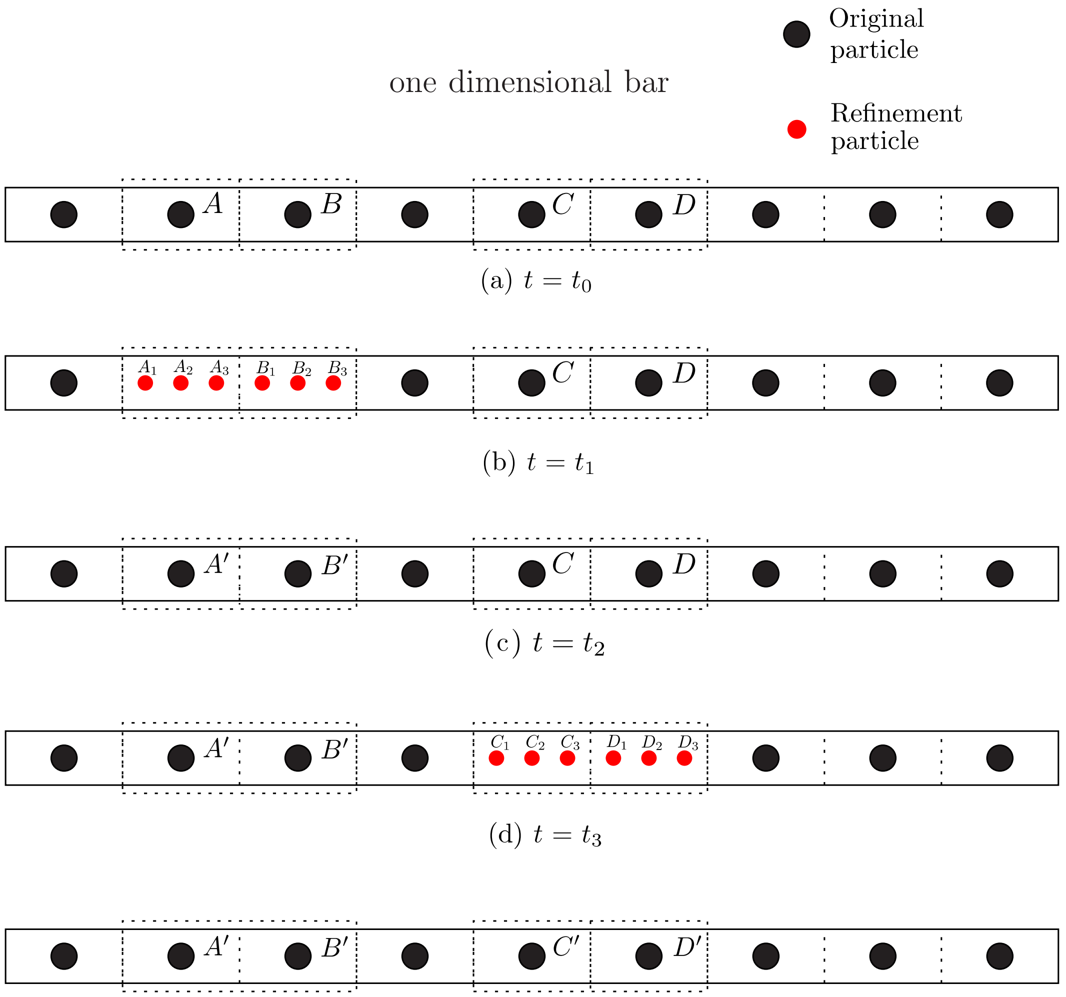
In practical engineering applications, the current status of the SPH method simulation has reached a limit by which a large scale problem domain requires arrangement of a large number of particles to get sufficient resolution, in order to obtain high accuracy. However, even arranging a large number of particles on high performance computers still cannot meet the requirement for large spatial problems in some cases and also leads to expensive computational cost.

In some real engineering applications, only part of the problem domain has dynamic behaviour (i.e. change of velocity, stress or temperature) during a certain time. For example, consider a beam in the shock wave propagation problem, which is fixed at the right end, the loading velocity is at the left end of the beam. The right part of the beam will not receive the dynamic information until the propagation of the shock wave reaches the right end. In this situation, the properties of the particles in the right part of the beam are not the feature of interest but will still need to be calculated, this is time consuming and limits the efficiency of the SPH method.

For this reason, a new version of the SPH method with, called the Adaptive SPH method, a time-varying particle distribution is attractive and studied by several researchers [5, 65, 87, 110, 88, 108, 109, 101]. First a particle splitting method has been implemented by Kitsionas and Whitworth for the astrophysics problem in 2002 [44]. Due to the adaptivity property of the particle distribution in the ASPH method, the accuracy of the method is increased using a relatively smaller number of particles compared to the standard SPH method. In order to achieve this, the basic concept of Adaptive Mesh Refinement (AMR) can be adopted on the particle methods, and is called the Adaptive Particle Splitting (APS) technique. Although the APS technique is not as mature as the AMR technique, it is still worth studied, improved and applied to the SPH method to obtain the new Adaptive SPH algorithm. Lopez applied the APS technique for the SPH method in fluid flow simulations [65]. Omidvar *et al.* applied variable mass particle distribution to simulate 2D and 3D fluid dynamic problems [87, 88] through the SPH method and Lastiwka *et al.* presented a more general algorithm with particle insertion and removal for the SPH method based on this [48]. Feldman and Bonet developed a dynamic particle refinement algorithm for the SPH method [27]. Then Vacondio *et al.* expanded the method with splitting and coalescing techniques on fluid dynamics [108, 109, 110]. Spreng *et al.* then applied the adaptive discretization algorithm for the SPH method on solid mechanics [101]. A new refinement procedure is developed by Barcarolo *et al.* [5]. In their work, the mother particle is not removed but turned off by an operator and daughter particles are turned on when they are created. One of the key points in the APS technique is the varying smoothing length, which can then be adopted as the adaptive smoothing length for this new algorithm, unlike the method proposed by Shapiro *et al.* [98] and Owen *et al.* [89], which only focuses on the adaptive kernel estimation. The adaptive SPH algorithm studied by the above researchers includes not only the adaptive kernel estimation but also the adaptive number and distribution of particles during the time integration process.

Take a one-dimensional bar as an example, the particle distribution is refined to be varied at different times. The basic concept of the new Adaptive SPH method in a 1D problem is shown in Figure 6.1. The two original particles A and B were split into three refined particles each with a smaller mass at $t = t_1$. The total mass and momentum of the system should be conserved (the sum of the mass of three refinement particles should be the same as one original particle). Then the refinement particles which are split at $t = t_1$ are merged to the coarse particles A' and B' at $t = t_2$. Another two of the original particles C and D are split into six refinement particles at $t = t_3$, see Figure 6.1. Then the refinement particles which are split at $t = t_4$ are merged to the coarse particles C' and D' at $t = t_4$.

Since the size of time step depends on the particle spacing, the time step will become smaller after the particle splitting. A smaller time step should be chosen for all particles after the splitting process in previous ASPH simulations [5]. In this thesis, the adaptivity of the method proposed here is different from the method introduced by the other researchers; here it involves not only the adaptive particle spacing but also the adaptive number of stages in one time step. One consistent time step relating to the initial particle spacing can be applied in the whole simulation and the number of stages



(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

(e) $t = t_4$

Figure 6.1: The basic concept of the new Adaptive SPH method in one dimensional space. (a) The particle splitting at $t = t_1$. (b) The particle merging and splitting at $t = t_2$.

in one time step will be adaptive when particle splitting and merging occur. This is achieved by combining the ASPH with the RKC time-stepping method, because of the stage adaptivity of the RKC scheme. This method is called the time-space Adaptive SPH method. The time-space Adaptive SPH method will be applied on elastodynamics to demonstrate the accuracy and efficiency of the method.

In this chapter, varying smoothing length will be first introduced and then developed to be the adaptive smoothing length for the adaptive SPH method. Different approaches to the adaptive kernel estimation will be discussed and one will be chosen for the ASPH method. Then the APS technique will be discussed and applied for the SPH method, including the splitting and merging of the particles. The technique of error control is applied in this method to minimise errors after particle splitting. The adaptive stages for a time step will be introduced to achieve the time adaptivity of the time-space Adaptive SPH method. Finally, the time-space ASPH method is applied to the simulation of the shock wave propagation in one-dimensional space to prove the accuracy and efficiency. Another static compressive problem in a one-dimensional case will be also analysed using the time-space ASPH method.

6.1 Adaptive kernel estimation

6.1.1 Varying smoothing length

In the SPH method, the choice of smoothing length h directly affects the accuracy of the simulation result and the computational efficiency. If the smoothing length is too small, there are not enough particles in the support domain ($r \leq \kappa h$) to represent the properties of the considered particle by the total contribution from the neighbouring particles, leading to a low accuracy of the results. Whereas if the smoothing length is too large, the computational efficiency will obviously decrease and the accuracy will also suffer as too many particles in the support domain will smooth out the local properties and information in the considered particle.

The particle approximation in the SPH method requires a sufficient but not excessive number of particles in the support domain to ensure the accuracy of the results [57]. The efficiency of the analysis will also be influenced by the number of particles. The smoothing length is empirically chosen from 0.8 to 1.8 times the initial particle spacing for different application problems [58]. For example, the number of neighbouring particles should be about 5, 21, 57 respectively in one-, two- and three- dimensional cases, when $h = 1.2\Delta p$ (Δp is the initial particle spacing) and $\kappa = 2$.

In the early application of the SPH method, the smoothing length was chosen based on the initial average density of the whole problem system. Then the individual smoothing length of each particle should be modified to depend on the local density of the considered particle for the expansion or compression problem of the fluid, this can maintain the consistency of the accuracy through the whole problem domain [71, 73]. Later many researchers recognised that the smoothing length should be adaptive in both space and time [39, 86, 103], which can be treated as the foundation of the Adaptive SPH method. Shapiro *et al.* [98] and Owen *et al.* [89] both developed an Adaptive SPH method only focusing on the adaptive kernel estimation with a fixed number of particles, which is different from the new ASPH algorithm developed in this thesis.

The main purpose of a varying smoothing length is to ensure relatively constant number of neighbouring particles inside the support domain during the simulation. There are many ways to achieve this and the common point of them is that the smoothing length can be updated according to the average local density. The simplest approach can be represented as;

$$h_i = h_i^0 \left(\frac{\rho_i^0}{\rho_i} \right)^{1/d}, \tag{6.1}$$

where h_i indicates the individual smoothing length of particle i and h_i^0 is the initial smoothing length. Similarly, ρ_i indicates the local density of particle i and ρ_i^0 is the initial local density and d refers to the number of dimensions.

Benz suggested another way to identify the smoothing length during the time integration of the analysis [8]. The equation includes the time derivative of the smoothing length;

$$\frac{dh_i}{dt} = -\frac{1}{d}\frac{h_i}{\rho_i}\frac{d\rho}{dt}. \tag{6.2}$$

This equation can be easily discretised by using the SPH approximation and added into the SPH system (3.29) to be calculated in parallel.

In this thesis, the method for evolving the varying smoothing length is different. Since only solid mechanics is considered here, the change of density can be ignored during the simulation. The only thing that influences the choice of smoothing length is the spacing of the particles. The distribution of the particles changes with time.

$$h_i(t) = a\Delta p_i(t). \qquad a \in [0.8, 1.8], \tag{6.3}$$

where $\Delta p_i(t)$ is the local spacing of particle i at time t and a is a given constant. This equation means that the smoothing length $h_i(t)$ is changing with respect to space and time. This equation can be applied in one- two- and three-dimensional space.

6.1.2 The symmetric influence between particles

In this method, every particle has an individual smoothing length which varies both in space and time, rather than a constant smoothing length for the whole system as in the original SPH method. If the smoothing lengths of two different particles i and j are different, the support domain of particle i may include the particle j but not necessarily vice versa. This means that the force exerted from particle i to particle j may be not the same as the corresponding reaction from j to i. This situation violates Newton's third law. In order to solve this problem, some treatments have been proposed to ensure the symmetric property of particle interactions. There are two main approaches for the treatment to maintain the symmetry of particle interactions, one is to correct the smoothing length and the other is to adopt the averaged value of the smoothing function for the two different particles.

There are several different ways to obtain a symmetric smoothing length for two particles with a different original h. Benz proposed a way of calculating the symmetric smoothing length, which is simply to take the arithmetic mean or the average of the smoothing lengths of two different interacting particles [9].

$$h_{ij} = \frac{h_i + h_j}{2}. \tag{6.4}$$

Other treatments can be adopted to obtain the symmetric smoothing length for two interacting particles i and j by calculating the geometric mean of h_i and h_j,

$$h_{ij} = \frac{2h_i h_j}{h_i + h_j}. \tag{6.5}$$

Sometimes the maximal value of the smoothing lengths is used,

$$h_{ij} = \max(h_i, h_j), \tag{6.6}$$

or the minimal value of the smoothing lengths,

$$h_{ij} = \min(h_i, h_j). \tag{6.7}$$

All these treatments for the smoothing lengths are to ensure symmetric values of the smoothing function between particles i and j (i.e. $\mathrm{W}_{ij} = \mathrm{W}_{ji}$). Then the value of the smoothing function can be obtained by using the symmetric smoothing length,

$$\mathrm{W}_{ij} = \mathrm{W}(R_{ij}, h_{ij}). \tag{6.8}$$

There are advantages and disadvantages of each method to ensure the symmetric smoothing length h_{ij}. Using the arithmetic mean or the maximal value of the smoothing lengths will lead to more neighbouring particles being included in the support domain and increase the computational cost. Sometimes a larger symmetric smoothing length will also smooth out the interactions between particles. On the other hand, the geometric mean or the minimal value of the smoothing length will tend to include less neighbouring particles inside the support domain and reduce the accuracy.

As mentioned before, the main purpose is to maintain the symmetry of particle interactions. The other approach is to directly calculate the average of the smoothing function values without adopting the symmetric smoothing length [39],

$$\mathrm{W}_{ij} = \frac{1}{2}\Big(\mathrm{W}(R_{ij}, h_i) + \mathrm{W}(R_{ij}, h_j)\Big). \tag{6.9}$$

These approaches that maintain the symmetry of the particle interactions are both widely used in the application of the SPH method. The arithmetic mean of the smoothing lengths (equation (6.4)) is adopted to achieve symmetric particle interactions in this thesis.

6.2 Refinement criterion

There are several ways to identify where and when to carry out the APS technique. Monaghan applied a linked-cell list containing particles and considered it as a guide to splitting particles [79]. Kitsionas and Whitworth adopted a criteria called Jeans condition to split particles [44]. The particle velocity gradient was used by Lastiwka *et al.* [48] as a guide to carry out the APS technique. And there are some other criteria to detect which particles should be split in [27, 65].

As far as we know, the criteria on where and when to carry out the APS technique depends much on the application problems and cannot be unified. The system domain will be classified into the splitting and merging sub-domains through the refinement criteria. The original particle i belonging to the splitting sub-domains will be split into several (n_i) refinement particles, which we call fine particles (i, f), while the fine particles belonging to the merging sub-domains will merge with other neighbouring fine particles into a coarse particle.

This thesis applies two different refinement criteria in the simulation addressed here.

1. The particle velocity gradient will be a prescribed criterion to assign the splitting and merging sub-domains in shock wave problem simulations. The velocity gradient of each particle will be calculated and a maximum value of the velocity

gradient will be identified. The system domain will be then be classified into different sub-domains through comparison between the reference velocity gradient ξ and the velocity gradient of particles

$$|\nabla \cdot \mathrm{v}_i| \geq \xi \qquad \text{Splitting,} \tag{6.10a}$$

$$|\nabla \cdot \mathrm{v}_{(i,f)}| < \xi, \quad (f = 1, 2, \ldots, n_i), \qquad \text{Merging.} \tag{6.10b}$$

2. We should adopt the particle splitting technique to obtain high accuracy when the particles are near the boundary. The particles will be split when the distance between the considering particles and boundary ($\mathbf{x}_{boundary}$) smaller than reference distance ζ,

$$|\mathbf{x}_i - \mathbf{x}_{boundary}| \leq \zeta \qquad \text{Splitting,} \tag{6.11a}$$

$$|\mathbf{x}_{(i,f)} - \mathbf{x}_{boundary}| > \zeta, \quad (f = 1, 2, \ldots, n_i), \qquad \text{Merging.} \tag{6.11b}$$

6.3 Particle splitting

In the SPH simulations, it is possible that part of the problem domain is simulated with slowly varying behaviour; i.e. the sub-domain with a low gradient of velocity or slow velocity. In this case, only few particles are required to ensure accuracy in these sub-domains. On the other hand, some sub-domains with sharp change of property or wave rebound behaviour, require more particles to obtain the solution with high accuracy. In this case, it is necessary to carry out the APS technique to achieve the balance between accuracy and computational cost in the SPH simulation. Note that the computational cost of the APS technique should be compared with the fully refined domain and that the accuracy is only a little lower than the fully refined domain.

Feldman and Bonet proposed an approach for general refinement [27]. Lopez *et al.* and Vacondio *et al.* also applied this approach in their work and obtained good solutions for different simulations [66, 110]. In their approach, an original mother particle is refined into four daughter particles with a symmetric pattern location in two-dimensional space. There are two refinement parameters: the separation parameter ϵ_r and smoothing ratio α_r. The separation parameter ϵ_r represents the spacing ratio of the daughter particles compared to the corresponding mother particle. The smoothing ratio α_r indicates the smoothing length ratio of daughter particles and mother particle. According to the stability analysis, the parameters $(\epsilon_r, \alpha_r) = (0.5, 0.5)$ in two-dimensional problems can avoid numerical instability [66]. This means that the spacing between the daughter particle from two different mother particles are the same as the spacing of the daughter particles from one mother particle, see Figure 6.2. Note

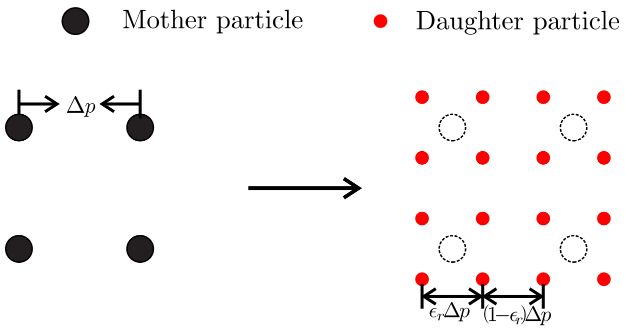


Figure 6.2: The refinement pattern stated by Lopez *et al.* [66] in 2D space, when $\epsilon_r = 0.5$.

that the circles with dashed outlines indicate the positions of the previous mother particles after particle splitting.

Here we will apply this symmetric pattern with the parameters $(\epsilon_r, \alpha_r) = (0.5, 0.5)$ to the distribution of refinement particles in one-dimensional space. In this thesis, the original particles are called coarse particles and refinement particles are named fine particles. Let i denote the original coarse particle, (i, f) represent corresponding fine particles and n_i be the number of fine particles for one coarse particle i. The properties of fine particles should be identified for the simulation,

$$\left\{ \mathbf{x}_{(i,f)}, m_{(i,f)}, \rho_{(i,f)}, h_{(i,f)}, \mathbf{v}_{(i,f)}, \boldsymbol{\sigma}_{(i,f)} \right\}. \tag{6.12}$$

First, a coarse particle is split into several fine particles with a symmetric pattern location around the considered coarse particle in a one-dimensional case, see Figure 6.3 (five fine particles in the figure).

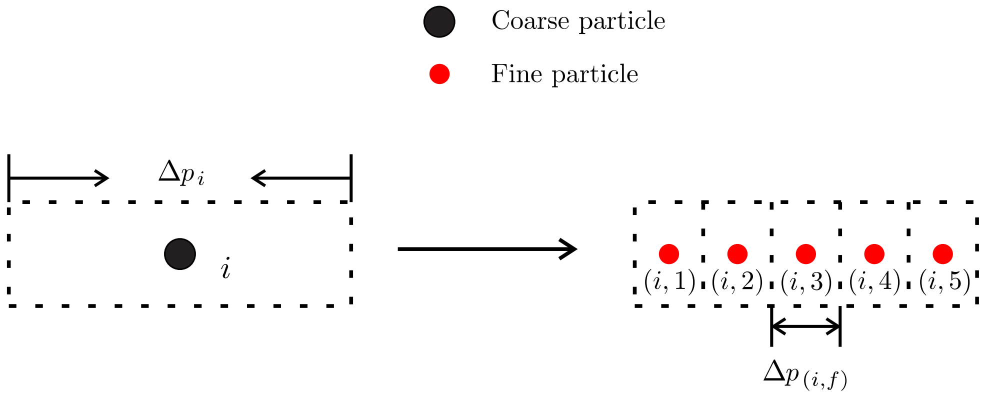


Figure 6.3: Particle splitting process in one-dimensional problems.

The spacing of fine particles is $\Delta p_{(i,f)}$, which can be represented as,

$$\Delta p_i = \sum_{f=1}^{(n_i)^{1/d}} \Delta p_{(i,f)}, \tag{6.13a}$$

where d is the dimension of the system. Normally, the fine particles will be uniformly distributed and have the same spacing with the fine particles split from other coarse particles, see Figure 6.3. Therefore the spacing can be rewritten as,

$$\Delta p_{(i,f)} = \frac{\Delta p_i}{(n_i)^{1/d}}. \tag{6.13b}$$

Then it is easy to obtain the smoothing length of the fine particles through equations (6.3) and (6.13b),

$$h_{i,f} = a\Delta p_{(i,f)}, \qquad a \in [0.8, 1.8], \tag{6.14a}$$

or

$$h_{i,f} = \frac{h_i}{(n_i)^{1/d}}. \tag{6.14b}$$

In this algorithm, the splitting procedure presented should be based on mass and momentum conservation. The total mass and momentum of the system should be conserved during the splitting process. Therefore the mass of one coarse particle should be the same as the sum of the corresponding fine particles.

$$m_i = \sum_{f=1}^{n_i} m_{(i,f)} \quad \text{or} \quad m_{(i,f)} = \frac{m_i}{n_i}. \tag{6.15}$$

Since only solid mechanics are discussed here, the density is normally uniformly distributed in solids and the spatial gradient of the density can be ignored. Then the density of the fine particles is the same as the coarse particle.

$$\rho_{(i,f)} = \rho_i. \tag{6.16}$$

For the velocity of the fine particles, Lopez *et al.* [66], Feldman and Bonet [27] applied the same velocity as the original coarse particle,

$$\mathbf{v}_{(i,f)} = \mathbf{v}_i, \quad f = 1, 2, \ldots, n_i. \tag{6.17}$$

This approach ensures that the linear momentum and kinetic energy of the system are conserved before and after the particle splitting process.

The stresses of the fine particles will be interpolated by the method in the work of Lopez *et al.* [66], Feldman and Bonet [27], which is originally from the Corrective Smoothed Particle Method (CSPM) [15],

$$\boldsymbol{\sigma}_{(i,f)} = \frac{\sum_{j=1}^{N} \frac{m_j}{\rho_j} \boldsymbol{\sigma}_j \mathrm{W}(\mathbf{x}_{(i,f)} - \mathbf{x}_j, h_j)}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathrm{W}(\mathbf{x}_{(i,f)} - \mathbf{x}_j, h_j)}, \tag{6.18}$$

6.4 Error control

6.4.1 Refinment error

The procedures of particle splitting and merging will introduce an error in the simulation system, since the local properties are modified by the particle interpolation technique. This error will influence the accuracy of the simulation and leads to the instability of the system. It is important to estimate and minimise the error in order to keep the simulation accuracy. Feldman and Bonet defined a measurement method for the density refinement error [27], which is the error of the refinement particle approximation of a function. This measure is also been improved by Vacondio *et al.* [109]. However, the momentum equation of the SPH method is based on the spatial gradient of the smoothing function and the method derived by Feldman and Bonet cannot measure the errors on the refinement particle approximation of the spatial gradient of a function. Then Lopez *et al.* proposed a similar method to measure the error of the particle approximation of the spatial gradient of field variables [65, 66].

Consider a set of particles, the particle approximation of the spatial gradient of field variables are determined by (2.33). Now the candidate particle i is refined into n_i fine particles, then we obtain,

$$< \nabla f_i >^{*} = < \nabla f_i > - \frac{m_i}{\rho_i} f_i \nabla \widetilde{\mathrm{W}}_i + \sum_{j=1}^{n_i} \frac{m_{(i,f)}}{\rho_{(i,f)}} f_{(i,d)} \nabla \widetilde{\mathrm{W}}_{(i,f)}. \tag{6.19}$$

The local error of the splitting procedure applied on particle i can be calculated by the sum of the squares of the difference between the original properties and the properties after splitting in each direction of the gradient. Combining with (6.15) we obtain,

$$e_i(\mathbf{x}) = m_i^2 \left(\frac{f_i}{\rho_i} \frac{\partial \widetilde{\mathrm{W}}_i^{\alpha}}{\partial \mathbf{x}^{\alpha}} - \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{f_{(i,d)}}{\rho_{(i,f)}} \frac{\partial \widetilde{\mathrm{W}}_{(i,f)}^{\alpha}}{\partial \mathbf{x}^{\alpha}} \right)^2, \tag{6.20}$$

where the summation convention applies when a super-script α appears twice in a term and here (6.20) will be the summation of the components of the gradients. Then the global error is the integration of (6.20) in the whole problem domain,

$$E_i = \int_\Omega e_i(\mathbf{x})d\mathbf{x}, \tag{6.21}$$

where Ω indicates the problem domain.

6.4.2 Density refinement error

The local error of density after refinement is then calculated by using the continuity density approach (3.8a) and Equation (6.19). Since the velocity of refinement particles is the same as the original coarse particle, the local error of density after refinement caused by the particle approximation of the spatial derivative can be written as,

$$\begin{aligned} e_i^\rho(\mathbf{x}) &= < \frac{D\rho(\mathbf{x})}{Dt} > - < \frac{D\rho(\mathbf{x})}{Dt} >^* \qquad (6.22) \\ &= m_i(\mathbf{v}(\mathbf{x}) - \mathbf{v}_i)\Big(\nabla\widetilde{\mathrm{W}}_i(\mathbf{x}) - \frac{1}{n_i}\sum_{f=1}^{n_i}\nabla\widetilde{\mathrm{W}}_{(i,f)}(\mathbf{x})\Big). \end{aligned}$$

Then the global error of density after refinement can be presented as,

$$E_i^\rho = \int_\Omega (e_i^\rho(\mathbf{x}))^2 d\mathbf{x}, \tag{6.23}$$

Here it is obvious that the density global error is related to the velocity, which changes with location and time. This means the error should be calculated for each time step and leads to high computational cost. The aim is to identify the variables $(\mathbf{x}_{(i,f)}, h_{(i,f)})$ in Equation (6.23) in order to minimise the error. Since the velocity does not change when carrying out particle splitting at time t, the optimisation of the density global error can be achieved by minimising the value of the following equation,

$$E_i^{\nabla\widetilde{\mathrm{W}}} = \int_\Omega \Big(\frac{\partial\widetilde{\mathrm{W}}_i^\alpha(\mathbf{x})}{\partial\mathbf{x}^\alpha} - \frac{1}{n_i}\sum_{f=1}^{n_i}\frac{\partial\widetilde{\mathrm{W}}_{(i,f)}^\alpha(\mathbf{x})}{\partial\mathbf{x}^\alpha}\Big)^2 d\mathbf{x}, \tag{6.24}$$

which is called the kernel gradient error in the work of Lopez *et al.* [66].

It is clear that the $E_i^{\nabla\widetilde{\mathrm{W}}}$ depends on the h, which changes with location and time. However, the smoothing length $h_{(i,f)}$ of the refinement particles is proportional to the smoothing length h_i of the original coarse particles in this thesis, see Equation (6.14b), which is related to the number of refinement particles for one coarse particle. Considering this, the kernel global error (6.24) can be simplified to depend on two components: initial smoothing length of the coarse particles and the number of refinement particles for one coarse particle. The target is to find the smallest value of $E_i^{\nabla\widetilde{\mathrm{W}}}$ to obtain the approximation of the refinement particle distribution for problems.

6.5 Merging of particle properties

The procedure of particle merging will be simpler than that for particle splitting. All the corresponding fine particles of a certain original coarse particle will be merged together into a coarse particle when the merging criterion is satisfied. Figure 6.4 presents the process of fine particles merging in a one-dimensional case.

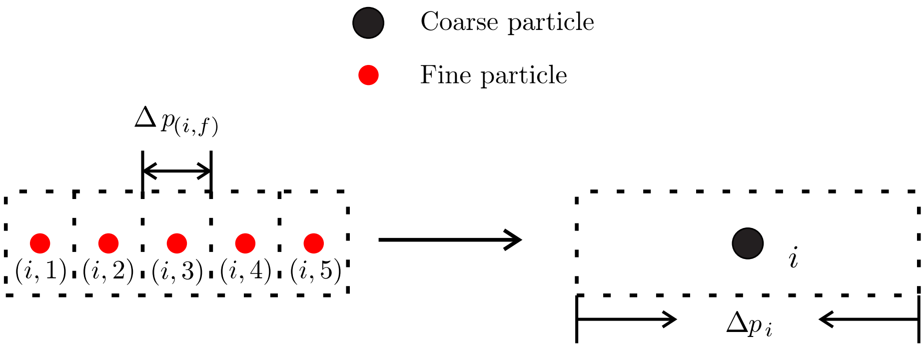


Figure 6.4: Particle merging process in one-dimensional problem.

The properties of new coarse particles should be identified for the simulation,

$$\left\{\mathbf{x}_i, m_i, \rho_i, h_i, \mathbf{v}_i, \boldsymbol{\sigma}_i\right\}.$$

Since the fine particles are normally uniformly distributed at the beginning of the splitting process, it is easy to merge the mass, location and density for the new coarse particle i using Equations (6.25), (6.26) and (6.27),

$$m_i = n_i m_{(i,f)}, \tag{6.25}$$

$$\mathbf{x}_i = \frac{\sum_{f=1}^{n_i} \mathbf{x}_{(i,f)}}{n_i}, \tag{6.26}$$

$$\rho_i = \rho_{(i,f)}. \tag{6.27}$$

The velocity of the new coarse particle should be the average velocity of the corresponding fine particles to ensure momentum conservation.

$$\mathbf{v}_i = \frac{\displaystyle\sum_{f=1}^{n_i} \mathbf{v}_{(i,f)}}{n_i}, \tag{6.28a}$$

and the total momentum after particle merging is conserved,

$$m_i \mathbf{v}_i = \sum_{f=1}^{n_i} m_{(i,f)} \mathbf{v}_{(i,f)}. \tag{6.28b}$$

The CSPM interpolation is still applied to obtain the value of the stress for the new coarse particle after merging. Note that the neighbouring particles are all from the fine particles in (6.29),

$$\boldsymbol{\sigma}_i = \frac{\displaystyle\sum_{j}^{n_i} \frac{m_{(i,f)}}{\rho_{(i,f)}} \boldsymbol{\sigma}_{(i,f)} \mathrm{W}\Big(\mathbf{x}_i - \mathbf{x}_{(i,f)}, h_{(i,f)}\Big)}{\displaystyle\sum_{f=1}^{n_i} \frac{m_{(i,f)}}{\rho_{(i,f)}} \mathrm{W}\Big(\mathbf{x}_i - \mathbf{x}_{(i,f)}, h_{(i,f)}\Big)}. \tag{6.29}$$

Vacondio *et al.* and Spreng *et al.* proposed an approach to achieve the new smoothing length without error [101, 110], which is adopted in the revised thesis. Since the density at position $\mathbf{x}_i$ should be constant before and after the particle merging process, which means the density error at $\mathbf{x}_i$ after merging is zero,

$$e(\mathbf{x}_i)^\rho = m_i \mathrm{W}(\mathbf{x}_i - \mathbf{x}_i, h_i) - \sum_{f=1}^{n_i} m_{(i,f)} \mathrm{W}\Big(\mathbf{x}_i - \mathbf{x}_{(i,f)}, h_{(i,f)}\Big). \tag{6.30}$$

Since the value of $\mathrm{W}(\mathbf{x}_i - \mathbf{x}_i, h_i)$ is equal to $\frac{3}{2} h_i$ for the cubic B-spline function in one-dimensional space, Equation (6.30) can be rewritten into,

$$e(\mathbf{x}_i)^\rho = m_i \frac{3}{2} h_i - \sum_{f=1}^{n_i} m_{(i,f)} \mathrm{W}\Big(\mathbf{x}_i - \mathbf{x}_{(i,f)}, h_{(i,f)}\Big) = 0. \tag{6.31}$$

Then the smoothing length of coarse particle i after merging can be calculated by,

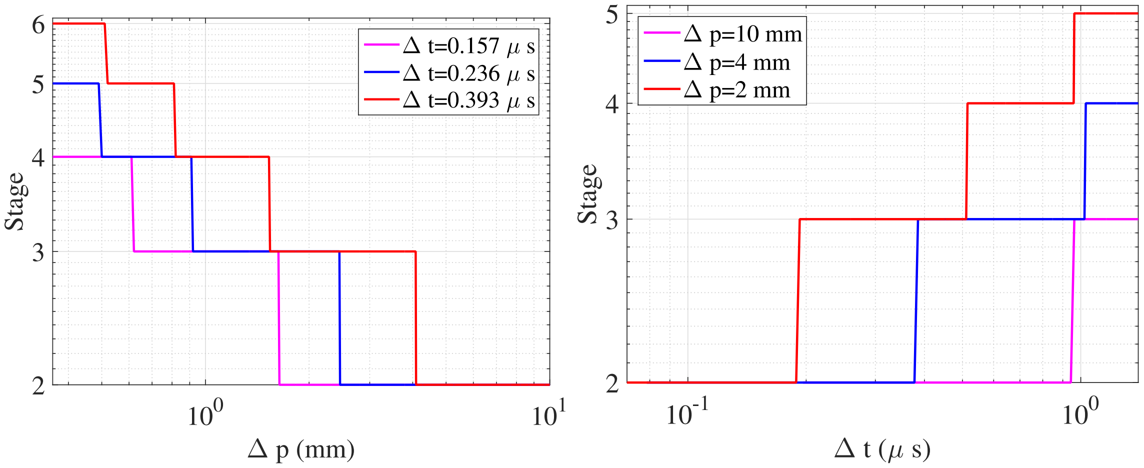
$$h_i = \frac{2 \sum_{f=1}^{n_i} \mathrm{W}\Big(\mathbf{x}_i - \mathbf{x}_{(i,f)}, h_{(i,f)}\Big)}{3 n_i}. \tag{6.32}$$

Note that the merging strategy stated here is only suitable for small deformation problems, since the technique assumes the refinement particles stay close together.

6.6 Adaptive stages for one time step

It is known that the size of a time step should satisfy the CFL condition. According to Equation (4.3), if the CFL value is fixed, then the size of the time step depends on the particle spacing. In the traditional SPH method applied to solid mechanics, the change of particle spacing is very small in the simulation. Therefore, the size of the time step will not change during the standard SPH simulation. However particle spacing is adaptive in the ASPH simulation. Then the smallest time step will be chosen to do the time integration after the particle splitting in the whole simulation in order to ensure accuracy [5].

As mentioned in Section 4.4, the stages of one time step depend on the size of Δt and the particle spacing Δp in the RKC time stepping scheme, shown in Equation (4.18). In other words, the number of stages will only depend on the particle spacing when the size of the time step is fixed. Figure 6.5(a) represents the relationship between the number of stages and the particle spacing for the problem in Section 5.1 with three fixed time steps $\Delta t = 0.157\ \mu s$, $0.236\ \mu s$ and $0.393\ \mu s$. Figure 6.5(b) shows the number of stages depending on the size of time steps with three fixed particle spacing $\Delta p = 10\ mm$, $4\ mm$ and $2\ mm$.



(a) (b)

Figure 6.5: (a) The relationship between stages and particle spacing with different fixed time steps. (b) The relationship between stages and time steps with different fixed particle spacing.

It is clear to see from Figure 6.5, that the number of stages in one fixed time step will adapt itself according to the changed particle spacing. Therefore the time adaptivity, which indicates one fixed time step in the simulation but with different stages, can be achieved by combining the RKC time stepping scheme and the ASPH method.

The time integration process of the previous ASPH method is unlike the Adaptive mesh refinement method, in which there are different time steps according to the mesh discretisation. For example, the time step in the regions with refined mesh is smaller than the coarse-mesh regions [10]. In the previous ASPH method, the smallest time step was chosen to do the time integration for all the particles after the particle splitting in order to ensure the accuracy of the ASPH method [5].

However, in this time-space ASPH method, there is only one fixed time step before or after not only the particle splitting but also merging process by combining the RKC time stepping scheme and the ASPH method. The stages at different time steps will adapt themselves according to the particle spacing after the particle splitting or merging process. The error analysis in Section 5.1 also proves that the influence of the number of stages and the CFL values on simulation accuracy is very small. For instance, the accuracy of the analysis with a small time step and fewer stages is very similar to the analysis with a large time step and more stages in the RKC-SPH method. Therefore, combined with the RKC time stepping scheme, the time-space ASPH method can bring very similar accuracy as the previous ASPH method which applies the smallest time step, but save computational cost.

6.7 Application of ASPH method

6.7.1 Shock wave propagation with the ASPH method

In order to examine the performance of the time-space ASPH method for the classical elastodynamics, here we solve the problem of shock wave propagation on a one-dimensional elastic bar in the work of Mabssout [68] through the time-space ASPH method.

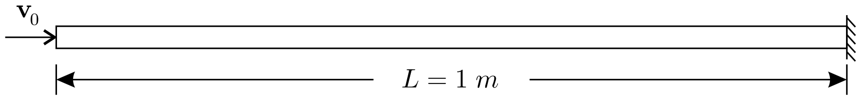


Figure 6.6: Compression loading on a one-dimensional solid bar.

The length of the considered elastic bar is $L = 1m$ as shown in Figure 6.6, with material property of $\rho = 2000\ kg/m^3$ and the Young's modulus is $E = 80 \times 10^6\ Pa$. Initially, the bar is at rest with $\mathrm{v} = 0$ and $\sigma = 0$. The velocity at the right end of the bar is fixed ($\mathrm{v} = 0$) and we apply a velocity on the left boundary as follow;

$$\mathrm{v}(t) = \begin{cases} 1\ m/s, & \text{if} \quad 1\ ms \leq t \leq 3.5\ ms, \\ 0\ m/s, & \text{otherwise.} \end{cases}$$

This is a one-dimensional wave propagation problem which has an analytical solution. Here, the value of stress under the compression loading can be calculated as follow:

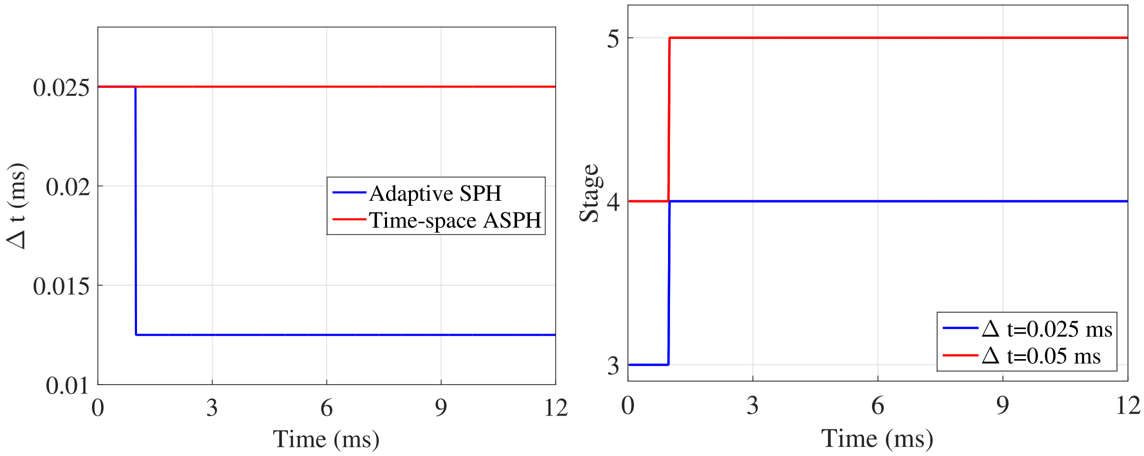
$$\sigma_0 = \mathrm{v}_0\sqrt{\rho E} = 4 \times 10^5\ Pa. \tag{6.33}$$

The stress wave will propagate through the bar with the physical wave speed, $c = \sqrt{\frac{E}{\rho}} = 200\ m/s$. The wave propagates through the bar and bounds back at the fixed right end. Then the stress σ at this point ($L = 1\ m$) will be doubled to a value of $8 \times 10^5\ Pa$ and the velocity of the shock wave after reflecting will also propagate along the bar with the opposite direction and the value is $\mathbf{v}_0 = -1\ m/s$.

First, we apply the standard SPH, ASPH and time-space ASPH method to simulate this model in order to investigate the advantages of the time-space ASPH method. 240 particles are arranged to model the behaviour by using the standard SPH method ($\Delta p_{SPH} = 4.167\ mm$ here), and we take $h = 1.5\Delta p$ here. In both the ASPH and time-space ASPH analysis, 120 original coarse particles are allocated for this one-dimensional bar, the initial spacing is $\Delta p_0 = 8.333\ mm$. One coarse particle will be split into two fine particles when the refinement criterion is satisfied. Using two fine particles in these two cases ensures that the smallest spacing in these three simulations is the same ($\Delta p_{(i,f)} = \Delta p_{SPH} = 4.167\ mm$). Here the analytical solution of the problem is discontinuous, we choose 1% of the largest analytical velocity gradient as the reference velocity gradient. The reference velocity gradient is calculated as $\xi = 1\% * 1 = 0.01$ and the reference distance as $\zeta = \Delta p$. Several simulations using the time-space ASPH method with different frequency of particle refinement have been implemented to find the value of frequency which can balance the accuracy and CPU time. Here the criteria for splitting and merging will be detected every ten time steps.

Because of the boundary condition, the state of the bar is steady before $t = 1\ ms$ and the refinement criterion will be satisfied after $t = 1\ ms$. Taking CFL$= 0.6$ for these three methods, the initial time step is $\Delta t_1 = 0.0125\ ms$ for the standard SPH simulation with 240 particles and $\Delta t_2 = 0.025\ ms$ for the ASPH and time-space ASPH simulations with 120 coarse particles. Then the change of time steps in two ASPH simulations is shown in Figure 6.7(a). The blue line in Figure 6.7(b) presents the change of stages in one time step in this time-space ASPH simulation (with $\Delta t_2 = 0.025\ ms$). The red line is the change of stages in one time step if the CFL value is 1.2 and size of time step is $\Delta t_2 = 0.05\ ms$.

The results of different methods are shown in Figure 6.8. Figure 6.8(a) and (b) present the velocity and stress distributions at $t = 5 \times 10^{-3}\ s$ along the bar respectively; Figure 6.8(c) shows the time evolution of the velocity at the mid point of the bar ($x = \frac{L}{2}$) and (d) is the stress at the right end point of the bar ($x = L$). It is easy to find that there are more particles in the sub-domain where there is a sharp change in



(a) (b)

Figure 6.7: (a) The change of time steps during the simulation. (b) The change of stages in one time step for the time-space ASPH.

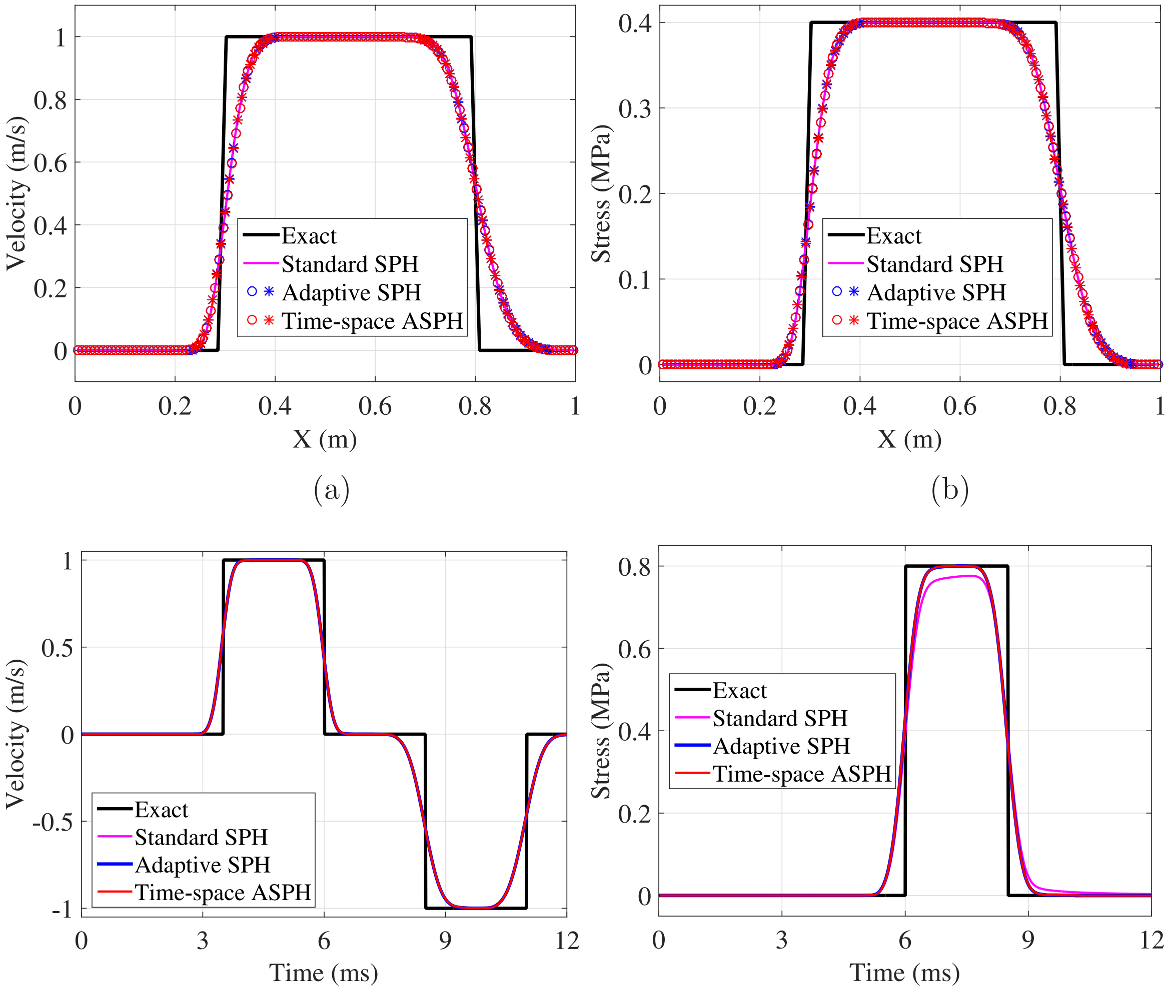
velocity and stress in the ASPH simulations, see Figure 6.8(a) and (b). The accuracies of the analysis simulated by these three methods are very similar in Table 6.1, but the CPU time of the time-space ASPH method is smaller than the other two methods, the computational cost can be reduced by the time-space ASPH method.

	Standard	Adaptive SPH	Time-space ASPH
L^1 error	0.1274	0.1287	0.1284
L^2 error	0.1958	0.1970	0.1969
Refinement error $E_i^{\nabla \widetilde{W}}$	-	1.48×10^{-4}	1.48×10^{-4}
CPU time(s)	1193.7	892.7	642.6
Number of particles	240	172	171

Table 6.1: Errors and computational time of using the standard, ASPH and time-space ASPH methods for solving shock-wave propagation at time $t = 5\ ms$.

It is worth investigating the influence of the number of extra fine particles on the analysis accuracy. We apply different combination types of the coarse particles and fine particles to simulate the shock wave propagation problem. Keeping the same smallest particle spacing $\Delta p = 4.167\ mm$ and the time step $\Delta t = 0.05\ ms$, seven different types of combination of the coarse and fine particles can be obtained as follows,

- Type 1: 120 original coarse particles, splitting into 2 fine particles each.
- Type 2: 80 original coarse particles, splitting into 3 fine particles each.



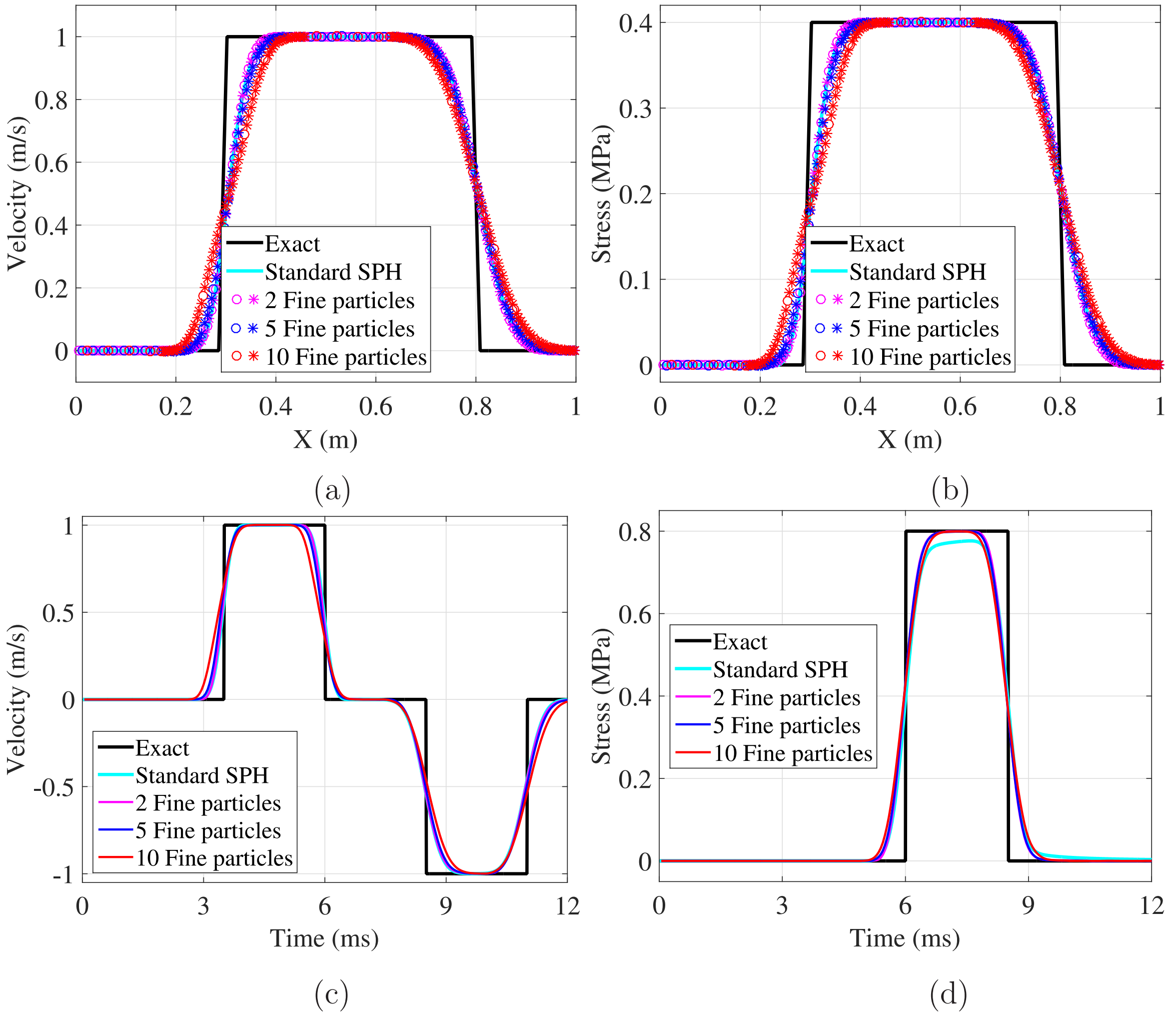
(a) (b)

(c) (d)

Figure 6.8: Velocity (a) and stress (b) distributions at $t = 5 \times 10^{-3}$ s along the bar; (c) Time evolution of the velocity at the mid point of the bar ($x = \dfrac{L}{2}$); (d) Time evolution of the stress at the right end point of the bar ($x = L$).

- Type 3: 60 original coarse particles, splitting into 4 fine particles each.
- Type 4: 48 original coarse particles, splitting into 5 fine particles each.
- Type 5: 40 original coarse particles, splitting into 6 fine particles each.
- Type 6: 30 original coarse particles, splitting into 8 fine particles each.
- Type 7: 24 original coarse particles, splitting into 10 fine particles each.

The results of the analysis are shown in Figure 6.9. The change of the stages, accuracy and CPU time of the results for each combination of coarse and fine particles are presented in Table 6.2. The CPU time is increasing when one coarse particle is split into more than four fine particles. However, the accuracy is decreasing with the



(a) (b) (c) (d)

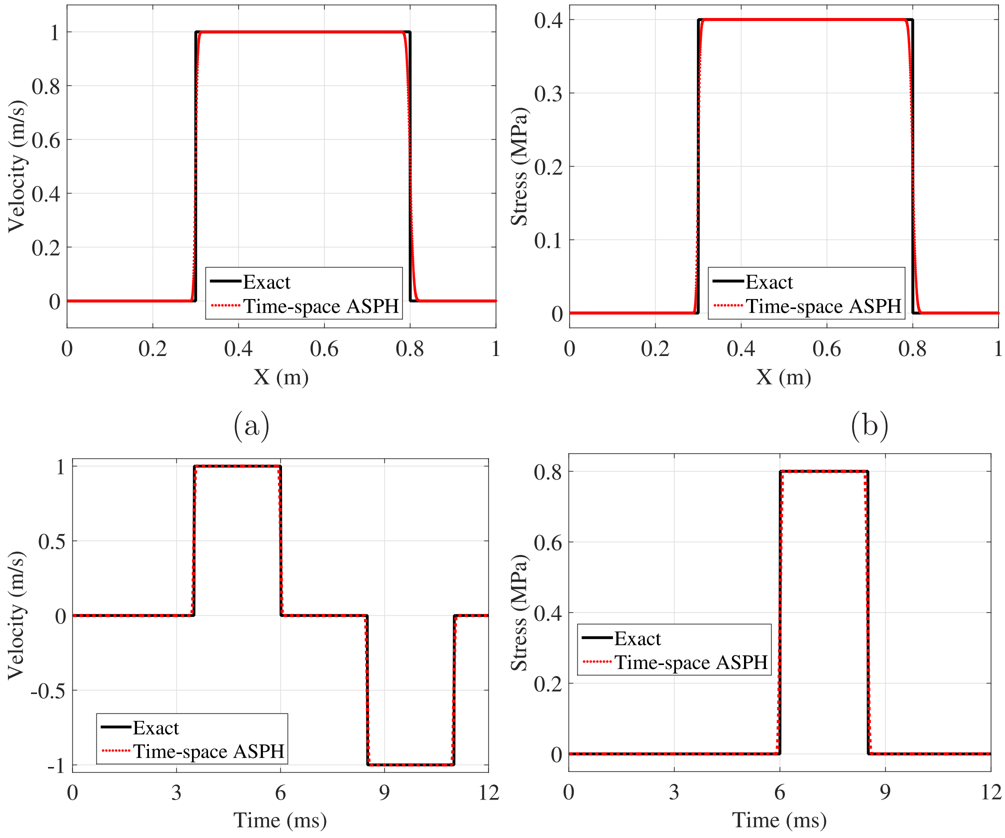
Figure 6.9: Velocity (a) and stress (b) distributions at $t = 5\ ms\ s$ along the bar with different particle combinations through the time-space Adaptive SPH methods; (c) the time evolution of velocity at the mid point of the bar ($x = \dfrac{L}{2}$); (d) the time evolution of stress at the right end point of the bar ($x = L$)

number of fine particles increasing. In other words, the number of fine particles cannot be increased without limit regarding not only accuracy but also computational cost.

Finally, we take 1000 coarse particles with one coarse particle splitting into two fine particles to simulate this shock wave propagation problem to obtain high accuracy results. Taking CFL= 1.0 for the initial particle spacing, the time step is $\Delta t = 5\ \mu s$. The number of stage is 3 calculated by (4.18) at the beginning and the number of stage will become to 5 after splitting. The simulation results are shown in Figure 6.10.

	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7
L^1 error	0.1286	0.1309	0.1361	0.1416	0.1482	0.1655	0.1865
L^2 error	0.1968	0.1987	0.2026	0.2068	0.2120	0.2256	0.2424
Refinement error $E_i^{\widetilde{\nabla \mathrm{W}}}$	0.00017	0.00059	0.0015	0.0026	0.004	0.0079	0.0133
Stages before splitting	4	3	3	3	3	2	2
Stages after splitting	5	5	5	5	5	5	5
CPU time(s)	559.7	499.4	464.6	478.6	489.4	492.0	512.2

Table 6.2: Errors and computational time of using time-space ASPH methods for solving a shock-wave propagation problem at time $t = 5\ ms$ with different types of coarse and fine particle combinations.



(a) (b)

(c) (d)

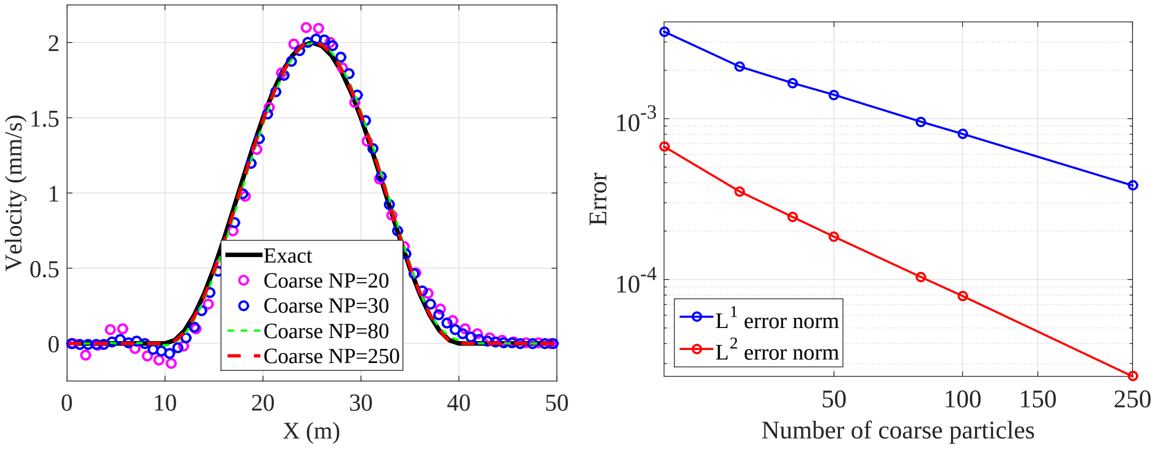
Figure 6.10: Velocity (a) and stress (b) distributions at $t = 4 \times 10^{-3}\ s$ along the bar with 50 original particles plus 101 extra refinement particles by the Adaptive SPH method; (c) time evolution of the velocity at the mid point of the bar ($x = \dfrac{L}{2}$); (d) time evolution of the stress at the right end point of the bar ($x = L$)

Convergence analysis

Similar to the RKC-SPH method, the rate of convergence for the numerical results is reduced by the artificial viscosity for the time-space ASPH method. To investigate the rate of convergence for this method, an example which is similar to the example in Section 5.1.2 is simulated by the time-space ASPH method without artificial viscosity here. A velocity is loaded at the left end of a one-dimensional bar with unity Young's modulus E and density ρ. The bar is fixed at the right end and with the length $L = 40\ m.$, the loading velocity is given by,

$$\mathrm{v}(t) = \begin{cases} \zeta(\sin(\frac{\pi}{15}t - \frac{\pi}{2}) + 1)\ m/s & \text{if} \quad t \leq 30\ s, \\ 0\ m/s, & \text{otherwise.} \end{cases}$$

where $\zeta = 0.001$. Then different numbers of coarse particles (20, 30, 40, 50, 80, 100, 250 coarse particles) are arranged to simulate this problem and one coarse particle will be split into two fine particles when the refinement criteria (ξ) is satisfied. Choosing CFL$= 1.0$ for the initial spacing of different particle arrangements, the number of stage is 3 obtained from (4.18) at the beginning and the number of stage will become to 5 after splitting. Figure 6.11(a) present the velocity distribution along the bar at $t = 40\ s$ with different numbers of particles. The L^1 and L^2 error norms are shown in Figure 6.11(b). The rate of convergence of the RKC-SPH method obtained here is 1 for L^1 error norm and 1.3 for L^2 error norm without the influence of artificial viscosity, which is slightly lower than RKC-SPH because of the refinement error.



(a) (b)

Figure 6.11: (a) The velocity distribution along the bar at $t = 40\ s$; (b) the L^1 and L^2 error norms of the stress distributioin at $t = 40\ s$.

6.7.2 Static compressive problem

In this section, a static compressive loading problem on a one-dimensional bar will be simulated to investigate the capability of the time-space ASPH method in the static problem. Take the same properties of the bar as the example in Section 6.7.1, a compression stress is loaded at the left end of the bar.

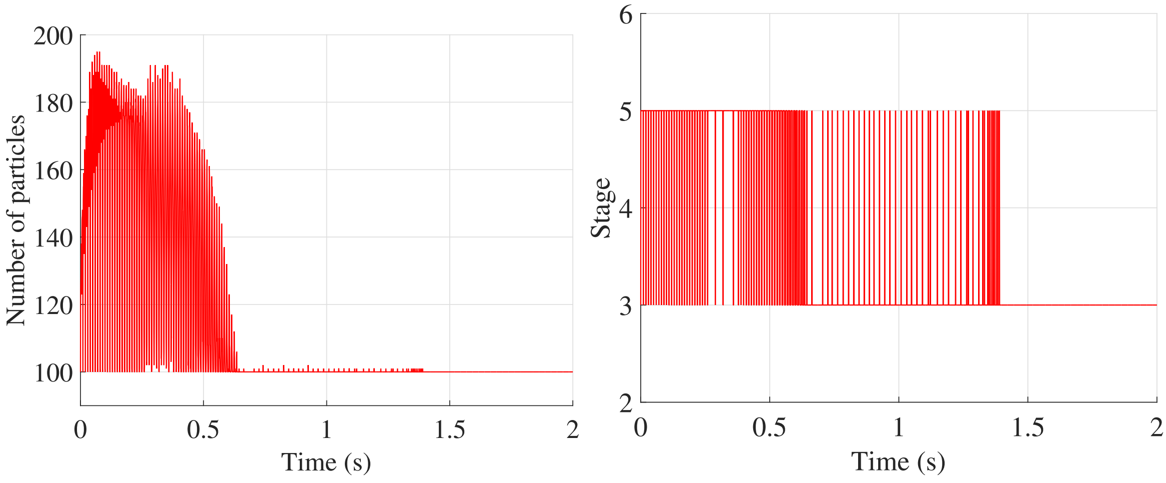
$$\sigma_0 = 100\ Pa,$$

then the strain of the bar will be $\varepsilon = \dfrac{\sigma_0}{E} = 1.25 \times 10^{-6}$ on elastic problem. The displacement of the left end of the bar will be $\Delta L = 1.25\ \mu m$. In dynamic, the velocity at left end caused by loading will be $\mathrm{v}_0 = \dfrac{\sigma_0}{\sqrt{E\rho}} = 2.5^{-4}\ m/s$. The wave propagates through the bar and bounds back at the fixed right end. Then the original wave and the reflective wave meet at the middle point of the bar and the stress value will be the double of the initial stress $\sigma_{mid} = 200$. However the bar will reach a steady stage and maintain quiescent condition in real case. Therefore the amplitude of stress at any point of the bar will be close to 100 Pa and the amplitude of velocity will be close to zero.

100 coarse particles are arranged to simulate this problem by using the time-space ASPH method. One coarse particle will be split into two fine particles when the refinement criteria is satisfied. The refinement criteria depends on the spatial derivative of the velocity as in (6.10), $\xi = 2.5 \times 10^{-6}$ and $\zeta = \Delta p$ are chosen for this problem. The CFL value is 1.0 for the initial particle spacing, so the time step is $\Delta t = 5 \times 10^{-5}\ s$ here. The time evolutions of number of particles and stages for the simulation are shown in Figure 6.12. This is because the spatial gradient of the velocity are smaller than ξ after $t = 1.4\ s$ for all the particles, all the fine particles have been merged into coarse particles. Therefore the stage remain to be three after $t = 1.4\ s$.

Figure 6.13 shows the displacement of left end simulated by the time-space ASPH methods, which is approaching to 1.25 μm. The velocity (a) and the stress (b) at the middle of the bar have been presented in Figure 6.15. The amplitudes of the stress and velocity are approaching to 100 Pa and zero after 1 s, respectively. The gradient of density error after splitting is $E_i^{\nabla \widetilde{\mathrm{W}}} = 0.00021$ and the error after merging is $E_i^{\nabla \widetilde{\mathrm{W}}} = 4.3 \times 10^{-5}$.

Since the velocity gradients of all the coarse particles are smaller than the reference velocity gradient at the steady stage, the number of particles reduces to 100 as shown in Figure 6.15(a). The theoretic solutions of the velocity and stress at the steady stage are $\mathrm{v} = 0\ m/s$ and $\sigma = 100\ Pa$ for the whole bar. The associated plot of the stress errors at the steady stage is shown in Figure 6.15(b). The error of the stress for each



(a) (b)

Figure 6.12: (a) The number of particles in the simulation. (b) The change of number of stages in the simulation.

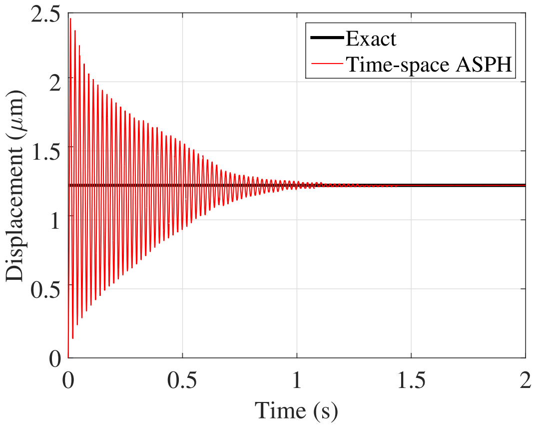
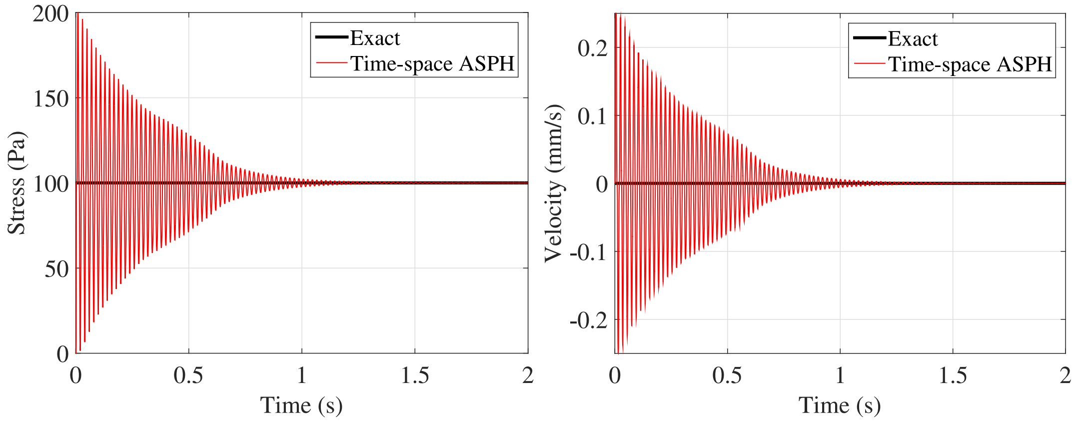


Figure 6.13: Displacement at the left end of the bar.

particle is smaller than 1%. It is clear that the quiescent condition can be maintained in this problem by using the time-space ASPH method and the time adaptivity of this method are demonstrated.

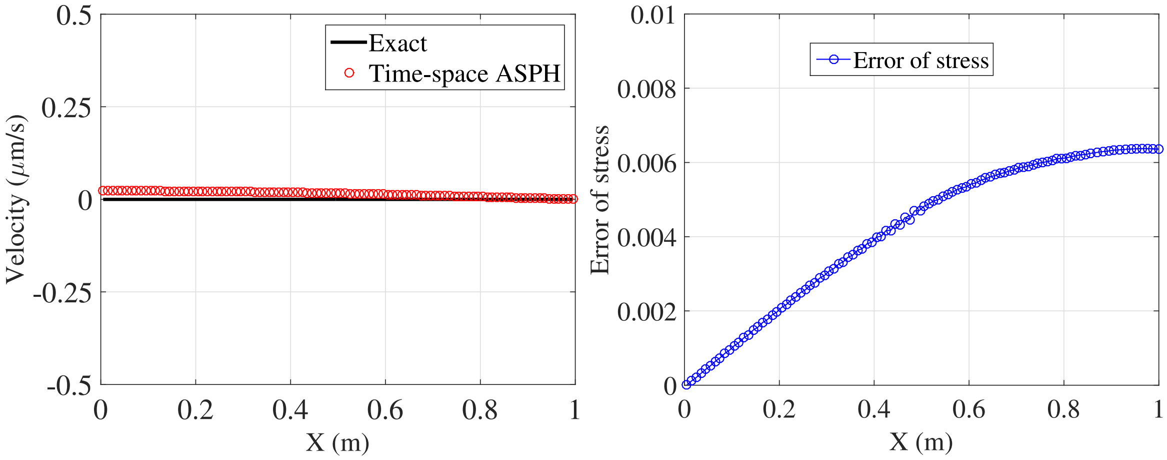
Concluding remarks

In this chapter, the ASPH method is combined with the RKC time stepping scheme, in which the distribution of particles adjusts itself and the stages in one time step is adaptive before and after splitting. This is called the time-space ASPH method. The varying smoothing length is introduced and then developed to be the adaptive smoothing length for the adaptive SPH method. Different approaches to the adaptive



(a) (b)

Figure 6.14: Velocity (a) and stress (b) at the mid point of the bar ($x = \dfrac{L}{2}$).



(1) (b)

Figure 6.15: (a) The velocity distribution along the bar at the steady stage. (b) The error of the stress along the bar at the steady stage.

kernel estimation will be discussed and the one raised by Benz [9] will be adopted in the ASPH method. The APS technique is applied for the SPH method, including the splitting and merging of the particles. The technique of interpolating and merging particle properties in [5, 27, 65, 101, 108, 109, 110] is applied in this chapter. The refinement error is measured and optimised by the method in [65, 66]. The adaptivity of the number of stages is explored for the time-space ASPH method in Section 6.6. Due to the adaptive property of the particles distribution and stages in one step, the accuracy and efficiency of the method will be greatly improved with small number of particles and fixed time step.

In order to determine the performance of the new algorithm, the time-space ASPH was applied on the simulation of the shock wave propagation in one-dimensional problems to prove the accuracy and efficiency of this method compared with the standard

SPH and ASPH methods. The refinement criterion was chosen to be the spatial gradient of the velocity for each particle. The particles near the boundary are also split to ensure the accuracy. The computational time and accuracy of the standard SPH method, ASPH and time-space ASPH method are compared in the analysis. In the time-space ASPH simulation, one fixed time step is applied in the whole simulation. The stages of one time step adjust themselves according to the smallest particle spacing. It is found that the results of the time-space ASPH simulation have similar accuracy to the standard SPH and ASPH simulation. However the large fixed time step can be applied in the time-space ASPH method to save computational cost. The accuracy of the time-space ASPH method by using different combinations between coarse particles and fine particles has been investigated using the technique in [65]. It can be concluded that the time-space ASPH method has adaptivity on both time and space. It can achieve a substantial reduction in memory and computational time, moreover nearly the same accuracy is obtained.

Chapter 7

Conclusions

7.1 General remarks

This thesis presents improvements of the Smoothing Particle Hydrodynamics method. A strong interest is focused on the application of the improved SPH method to elastodynamics. The contribution and achievement of the research will be outlined in this chapter, including the improvements and applications of the SPH method.

In Chapter 1, the importance and the key points of numerical methods were first introduced. Numerical methods can be classified into two types based on the form of problem domain discretisation: the grid-based and mesh-free methods. Compared with mesh-free methods, grid-based numerical methods are limited by the complex mesh generation in the simulation and challenged by the dis-continuum problem domain. Mesh-free numerical methods avoid the grid to make the discretisation components more flexible and powerful in many applications. The SPH method is one of the oldest mesh-free methods, uses integral representation to approximate the field function which is discretised to give the particle summation. In this research the SPH method will be improved and applied on elastodynamics.

The basic concept and the equations are derived for the SPH kernel estimation process in Chapter 2. The SPH approximation can be divided into two steps: (i) kernel approximation to represent the field variables and their derivatives in an integration form by using smoothing functions and (ii) particle approximation to discretise the continuous problem domain into a set of particles, which carry physical properties, i.e. mass, volume and density. Different smoothing functions are introduced and compared in this chapter. The approximation accuracy was influenced by the domain boundary and there are several versions of the corrected SPH method to solve this problem. A function approximation example is presented to compare the different versions of the corrected SPH method. The convergence analysis indicates that the SPH method with corrected form has second order convergence.

To fulfill the requirements of the physical model, the governing equations of the hydrodynamics are represented in the SPH approximation form in Chapter 3. Since the governing equations can be derived based on both Eulerian and Lagrangian approaches, these two approaches were introduced. The governing equations of hydrodynamics have been derived into the SPH discretisation form. In addition, the accuracy near the slip and non-slip boundaries is improved by including the ghost particles. Artificial viscosity is introduced to smooth out the unphysical oscillation caused by the shock wave in the elastodynamics. The SPH method is challenged by the tensile instability and different special treatments are introduced to solve this numerical problem in Chapter 3. The SPH system has been built by the particle approximation of the governing equations and the special treatments to improve the accuracy of the method.

The SPH system is then combined with a time integration algorithm to obtain the predicted solution for the motion of the system. This stage can be handled by the explicit ordinary differential equation (ODE) solver, in which the Courant-Friedrichs-Levy (CFL) condition plays an important role. The Runge-Kutta Chebyshev (RKC) time stepping scheme is introduce in Chapter 4 to integrate the discretised SPH equations including other time integration methods, such as the Euler, predictor-corrector and classical Runge-Kutta (RK) schemes. Compared with other time stepping methods which are currently widely used, the RKC method has the advantages of flexible size and stages of each time step. The RKC time stepping scheme is more robust than the other classical methods in the RK methods family since the stability region for RKC is larger than others [112].

As a explicit hydrodynamic method, the RKC time stepping scheme is combined with the SPH method to increase efficiency and accuracy. In order to explore the advantages of the RKC-SPH method, different time stepping schemes are applied to perform the time integration for the SPH method and they will be then compared through the CFL values and accuracies in Chapter 5. A set of benchmark problems, which have a theoretical solution, are simulated in this chapter to investigate this algorithm applied to key areas of elastodynamics. A one-dimensional shock wave problem is first simulated by the RKC-SPH method to show the performance of the RKC-SPH method, compared to the Euler, Predictor and RK4 time stepping schemes. Although the SPH method is usually used to simulate dynamical systems, in the present study we use the steady-state SPH results in comparison to the FEM applied to a static problem. The results of quasi-static problem analysed by the RKC-SPH method show higher accuracy than the static FEM method and the SPH method with the predictor-corrector time stepping scheme.

The RKC-SPH method is shown to have the capability to solve large deformation problems, i.e. the two-dimensional oscillatory beam problem in Section 5.3. The stress fields and velocity vectors are presented to show the dynamic behaviour of the problem. The oscillation frequencies of different beams are presented to compare with the analytical frequency. A large number of particles is used in the RKC-SPH method to show the convergence of the method. In Section 5.4, the dynamic process of compression loading on a porous structure is simulated through the RKC-SPH method and the results show the stability of this algorithm. These simulation results show the strong capability of the RKC-SPH in solving large deformation problems.

Since the RKC-SPH method is a particle-based method, the influence of the initial particle distribution on the simulation results has been explored in the problem of the two-dimensional elastic plate (Section 5.2). Three different particle distributions were arranged to simulate the compression process of the two-dimensional elastic plate with a circular void; square distribution, radial distribution and equally radial distribution. Comparison results of the stresses and the velocity fields among the three distributions show that the equi-radial distribution has the best performance since particles are more uniformly distributed near the boundary in this allocation plan.

However, there is another way to solve the problem caused by the non-uniform particle distributions in the SPH method. In this research, the mass of each particle is assumed to be constant in distributions because we try to simplify the simulation process and only focus on how to improve the SPH algorithm. If we adopt the area times the density to calculate the masses for particles, the problem caused by the nonuniform distribution near the boundary can be solved.

In engineering practical applications, the current status of the SPH method simulation has reached a limit is here that a large-scale problem domain requires using a prohibitively large number of particles to get the sufficient resolution, in order to ensure high accuracy. However, even using a large number of particles on high performance computers still cannot meet the requirement for large spatial problems in some cases and also leads to expensive computational cost. To address this, a new time-space ASPH has been proposed in Chapter 6. Unlike previous ASPH simulations, the smallest time step is chosen for all particles after the splitting process, one fixed time step can be applied for the time-space ASPH method. The number of stages for one time step is adaptive according to the refinement of the particles. The time-space ASPH is applied to elastodynamics to obtain similar accuracy as the standard SPH but with a small number of particles. In this new method, not only the distribution of the particles but also the number of stages are adjusted during the time integration in order to save computational cost in the analysis. A one-dimensional shock wave problem is simulated by the time-space ASPH method to demonstrate the efficiency and accuracy compared to the standard SPH and ASPH methods. Another static compressive loading problem is modelled to prove the capability of the time-space ASPH on elastodynamics.

7.2 Achievements

Considerable achievement has been made throughout this thesis to provide the improvements of the SPH method applied to elastodynamics. The contribution of the work can be summarised into two aspects:

1. *It is the first to combine the SPH method with the RKC time stepping scheme to achieve high accuracy and efficiency of the simulation.* In the explicit hydrodynamic method, the Courant-Friedrichs-Levy condition (CFL) plays an important role in the simulation of time integration. This is a necessary condition of the numerical stability when solving ODEs. The CFL condition ensures that the propagation speed of the numerical simulation should be smaller than the speed of physical propagation, which means the size of the time step in the simulation is limited by the CFL condition. Large time steps will cause instability in the numerical simulation, whereas small time step sizes can bring high accuracy for the numerical simulation but high computational cost.

   The RKC time stepping scheme is a new algorithm in the Runge-Kutta methods family. The advantage of this method is that it can adapt the number of stages itself in one time step according to the value of CFL. When we have a large CFL value (large size of Δt) the number of stages in one step is large to avoid instability. The stability region of the RKC method is larger than the other traditional Euler and Runge-Kutta family methods. Unlike the other traditional time stepping schemes, the accuracy of the RKC method is not influenced when we have large numbers of stages in one large time step. In the other traditional time stepping schemes, accuracy reduces while the size of time step increases, because larger sizes of time step dissipate more simulation information. This is demonstrated by the numerical results of the one-dimensional shock wave propagation example in Chapter 5.

   The application results of the SPH method combined with the RKC time stepping scheme in Chapter 5 show a strong stability for this time integration algorithm compared with other traditional algorithms, such as Euler, predictor-corrector and Fourth order Runge-Kutta methods. The efficiency of the RKC-SPH algorithm is also improved by the RKC time stepping scheme since it can use a small number of stages when the CFL value is the same as RK4 scheme, saving computational time.

   Note that the SPH method is a dynamic method and the results calculated by it are only snapshots of the whole dynamic process rather than a perfectly steady state. However, this RKC-SPH algorithm still shows strong capability in the steady state result compared to the FEM result when handling the quasi-static problem, even the FEM result is from a static simulation carried out by the software ABQUAS. The result of the SPH method with the predictor-corrector time stepping scheme has slightly lower accuracy than the RKC-SPH and the FEM results. In other word, combining with the RKC time stepping scheme improves the performance of the SPH method. The convergence analysis results also show that the rate of convergence for the RKC-SPH method has the 1.5 for L^2 error norm.

In conclusion, the RKC-SPH method has the advantages:

- Improve the stability of the method compared with the other traditional time stepping schemes.
- Save the computational cost during the simulation since the CFL value can be much larger than in other schemes.

2. *The new time-space Adaptive SPH method has been explored and applied on elastodynamics.* As mentioned before, the current limitation of the SPH method is the requirement of a large number of particles in order to ensure high accuracy when solving large scale problems. In this situation, a new version of the SPH method, which can adapt the particle distribution in the simulation, has been raised by some researchers [65, 87, 88, 110]. In previous ASPH simulations, the smallest time step will be chosen for all particles after the splitting process [5], the new time step Δt_{new} is regarding as the new particle spacing (Δp_{new} or $\Delta p_{(i,f)}$). In other words, the size of Δt in previous ASPH will be reduced after the particle splitting process. In this research, the adaptivity of the method proposed here is different from the method introduced by the other researchers. One consistent time step Δt regarding to the initial particle spacing ($\Delta p_{initial}$ or Δp_i) can be applied in the whole simulation and the number of stages s will be adaptive according to the process of particle splitting and merging. Therefore, this method involves not only adaptive particle spacing Δp but also an adaptive number of stages s in one fixed time step Δt. This is achieved by combining the ASPH with the RKC time-stepping method, because of the stage adaptivity of the RKC scheme. This method is called the time-space Adaptive SPH method. The adaptivity of the stages is explored in this research. Due to the adaptive property of the particles distribution and the stages s in one time step Δt, the accuracy of the method is similar to the simulation with large number of particles and the computational cost is reduced.

   In this method, the basics of adaptive mesh refinement (AMR) are adopted into the particle methods. This is called the adaptive particle splitting (APS) technique, studied by Lopez in fluid flow simulations [65]. Since the particle distribution is changed, the smoothing length is also varied for each particle. The calculation of varying smoothing length and the adaptive kernel estimation are introduced. The properties of the refinement particles are defined by the CSPM interpolation method. On the other hand, the properties of the particles after the merging process are also calculated based on the CSPM interpolation method.

The refinement criterion also plays an important role in the ASPH method, which defines when and where to interpolate refinement particles and merge refinement particles into a coarse particle. As far as we know, the criteria on where and when to carry out the APS technique depend much on the application problems and cannot be unified. The system domain will be classified into splitting and merging sub-domains through the refinement criteria during the simulation. The original particle belonging to the splitting sub-domains will be split into several refinement particles, while the fine particles belonging to the merging sub-domains will merge with other neighboring fine particles into a coarse particle. The error after carrying out the APS technique compared with the original distribution will be analysed and the error minimisation approach of the work in [65] is applied to ensure the stability of the ASPH method.

A dynamic adaptive particle refinement algorithm for SPH is presented in Chapter 6 and the ASPH method is applied on two benchmark one-dimensional problems, which have the theoretical solutions. The results of the standard SPH, ASPH and time-space ASPH are compared and the rate of convergence for this method has been proved to be 1.3 for L^2 error norm.

The time-space ASPH method is first used to model the classical one-dimensional shock wave problem. The results determine the performance of the time-space ASPH method in comparison with the standard SPH method and ASPH method, since there is a analytical solution for this shock wave problem. The refinement criterion is the spatial gradient of the velocity of each particle. The second example is the simulation of a classical static compressive problem, which also has an analytical solution. The refinement criterion is the spatial gradient of the velocity of each particle and the distance to the solid boundary.

The computational time and accuracy of the stand SPH, Adaptive SPH and time-space SPH methods are compared and it is found that it only needs a small number of particles in the time-space ASPH simulation to obtain similar accuracy results with standard SPH simulation. The large size of the time step can be chosen in the time-space SPH method and the stages adjust according to the particle distribution. Using the dynamic adaptive particles refinement procedure and adaptive stages for one time step with the adequate refinement criterion, instead of adopting a fine discretisation for the whole domain and a small size of time step in the whole simulation, can deliver a substantial reduction in memory and computational time, while nearly the same accuracy is achieved. The results of the ASPH method show its strong capability to solve elastodynamic problems.

In conclusion, the time-space ASPH method has the advantages:

- Improve the stability of the method compared with the original ASPH method since the number of stages is adaptive with the particle refinement procedure.
- Save computational cost during the simulation.

7.3 Future work

During the research process, unsolved and partially unsolved issues were encountered. These are presented here as suggestions for future work.

1. *Reconstruction of asymptotic-preserving schemes to resolve different time scales.*

   Through applying the Runge-Kutta Chebyshev time stepping scheme for the SPH method, it is found that an adequate time integration algorithm plays an important role in the SPH simulation. It is worth reconstructing asymptotic-preserving time stepping schemes to resolve different time scales for SPH simulations in future.

2. *Apply the time-space ASPH method to nonlinear elastodynamics and elastoplastic dynamic cases.*

   In this research, all the applications focus on pure elastodynamic problems. It is worth enhancing the application of the time-space ASPH method to the areas of nonlinear elastodynamics and elastoplastic dynamic cases.

3. *Combine the time-space ASPH method with the FEM method to solve large size scale problems*

   In the real engineering application, the analysed structure always has a large scale size, requiring a large number of particles to be allocated (probably $10^7 - 10^9$ particles for the whole domain). As we know, it is very time consuming for the SPH simulation with such large numbers of particles. Therefore, it is worth considering the combination of the time-space ASPH method and the FEM method to construct a more efficient numerical method.

   Take a large scale structure formed by cellular solids as an example, the initial work involves building an ASPH code to simulate the deformation of a small cube of cellular solids. The boundary conditions need careful attention offering alternative approaches to be explored to capture zero and full rebound responses. After the mechanical response has been simulated in the cube, subjected to many different boundary stresses (uniform on each face, but of different magnitude in the three dimensions), a plasticity model will be constructed to represent the macroscopic relationship between stress and strain. Then the FEM employing this plasticity model will be used to simulated the mechanical behaviour of the (large) structural component.

4. *Implementation on GPUs for 3D problems.* The simulation of complex three-dimensional system is always limited by the computational cost in CPU. The parallel feature of GPUs has the ability on modelling large scale and complex structure domain. It is more powerful for the computational capability of the graphical processing units (GPUs) to carry out the numerical simulation.

Bibliography

[1] J. D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill, 1995.

[2] M.F. Ashby and D.R. H. Jones. *Engineering Materials 2*. Elsevier, 2013.

[3] S.N. Atluri and S.P. Shen. *The meshless local Petrov-Galerkin (MLPG) method*. Tech Science Press, USA, 2002.

[4] S.N. Atluri and T. Zhu. A new meshless local petrov-galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

[5] D.A. Barcarolo, D. LeTouze, G. 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.

[6] S. G. Bardenhagen and E. M. Kober. The generalized interpolation material point method. *Computer Modeling in Engineering & Sciences*, 5(6):477–495, 2004.

[7] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[8] W. Benz. Smoothed particle hydrodynamics: a review. *NATO Workshop, Les, France*, 1989.

[9] W. Benz and E. Asphaug. Simulations of brittle solids using smoothed particle hydrodynamics. *Computer Physics Communications*, 87:253–265, 1995.

[10] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys*, 82:6484, 1998.

[11] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with application in metal forming simulations. *Int. J. Numer. Methods Eng*, 47:1189–1214, 2000.

[12] H. H. Bui, R. Fukagawa, K. Sako, and S. Ohno. Lagrangian meshfree particles method (sph) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model. *International journal for numerical and analytical methods in geomechanics*, 32:1537–1570, 2008.

[13] H. H. Bui, R. Fukagawa, K. Sako, and J. C. Wells. Slope stability analysis and discontinuous slope failure simulation by elasto-plastic smoothed particle hydrodynamics (sph). *Geotechnique*, 61(7):565–574, 2011.

[14] J. Campbell and R. Vignjevic. Artificial viscosity methods for modelling shock wave propagation. *Predictive Modeling of Dynamic Processes*, page 349365, 2009.

[15] J. K. Chen and J.E. Beraun. A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. *Computer Methods in Applied Mechanics and Engineering*, 190:225–239, 2000.

[16] J. K. Chen, J.E. Beraun, and T.C. Carney. A corrective smoothed particle method for boundary value problem in heat conduction. *Computer Methods in Applied Mechanics and Engineering*, 46:231–252, 1999.

[17] J. K. Chen, J.E. Beraun, and C.J. Jih. Completeness of corrective smoothed particle method for linear elastodynamics. *Computational Mechanics*, 24:273–285, 1999.

[18] W. Dehnen and H. Aly. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2):1068–1082, 2012.

[19] P.J. Van der Houwen. Explicit Runge-Kutta formulas with increased stability boundaries. *Numer. Math.*, 20:149–164, 1972.

[20] P.J. Van der Houwen and B.P. Sommeijer. On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values. *Z. Angew. Math. Mech.*, 60:479–485, 1980.

[21] G.A. Dilts. Moving least square particle hydrodynamics i: consistency and stability. *Int. J. Numer. Methods Eng*, 44:1115–1155, 1999.

[22] G.A. Dilts. Moving least square particle hydrodynamics ii: conservation and boundaries. *Int. J. Numer. Methods Eng*, 48:1503–1524, 2000.

[23] J.M. Dominguez, A.J.C. Crespo, M. Gomez-Gesteira, and J.C. Marongiu. Neighbour lists in smoothed particle hydrodynamics. *International Journal for Numerical Methods IN Fluids*, 67(12):2026–2042, 2011.

[24] C. T. Dyka and R. P. Ingel. An approach for tension instability in smoothed particle hydrodynamics (sph). *Computers and Structures*, 57:573–580, 1995.

[25] C. T. Dyka and P. W. Randles. Stress points for tension instability in sph. *International Journal of Numerical Methods in Engineering*, 39:2725–2741, 1996.

[26] T. El-Gammal, E.E. Khalil, H. Haridy, and E. Abo-Serie. Influence of smoothing length and virtual particles on sph accuracy. *International Journal of Materials, Mechanics and Manufacturing*, 1(2):166–170, 2013.

[27] 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.

[28] D.A. Fulk. A numerical analysis of smoothed particle hydrodynamics. *Air Force Institute of Technology*, 1994.

[29] D.A. Fulk and D.W. Quinn. An analysis of 1-d smoothed particle hydrodynamics kernels. *Journal of Computational Physics*, 126(1):165–180, 1996.

[30] R. A. Gingold and J. J. Monaghan. Smoothed particles hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.

[31] M. Gomez-Gesteira, B.D. Rogers, A.J.C. Crespo, R.A. Dalrymple, and M. Narayanaswamy. Sphysics development of a free-surface fluid solverpart 1: Theory and formulations. *Computers and Geosciences*, 48:289–299, 2012.

[32] M. Gomez-Gesteira, B.D. Rogers, R.A. Dalrymple, A.J.C. Crespo, and M. Narayanaswamy. User guide for the sphysics code v1.0. *http://www.sphysics.org*, 2007.

[33] S. Gottlieb, C.W. Shu, and E. Tadmor. Strong stability preserving high order time integration methods. *SIAM Rev.*, 43:89–112, 2001.

[34] J. P. Gray, J. J. Monaghan, and R. P. Swift. SPH elastic dynamics. *Computational Methods in Applied Mechanics and Engineering*, 190:6641–6662, 2001.

[35] Y. T. Gu and G. R. Liu. A boundary point interpolation method for stress analysis of solids. *Computational Mechanics*, 28(1):47–54, 2002.

[36] Y. T. Gu and G. R. Liu. Hybrid boundary point interpolation methods and their coupling with the element free galerkin method. *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, 27(9):905–917, 2003.

[37] E. Hairer, S.P. Norsett, and G. Wanner. *Solving ordinary differential equations I: Nonstiff problems.* Number ISBN 978-3-540-56670-0. Springer-Verlag, 1993.

[38] F.H. Harlow. The particle-in-cell method for numerical solution of problems in fluid dynamics. *Proceeding of Symposia in Applied Mathematics*, 1963.

[39] L. Hernquist and N. Katz. Treesph- a unification of sph with the hierarchical tree method. *Astrophysical Journal Supplement Series*, 70:419–446, 1989.

[40] R. W. Hockney and J. W. Eastwood. Computer simulations using particles. *Adamhilger, New York*, 1988.

[41] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations.* Number ISBN 978-0-521-55655-2. Cambridge University Press, 1996.

[42] T. Jankowiak and T. Lodygowski. Smoothed particle hydrodynamics versus finite element method for blast impact. *Predictive Modeling of Dynamic Processes*, page 349365, 2009.

[43] F.R. Johnson, R.A. Stryk, and S.R. Beissel. Sph for high velocity impact computations. *Computational Methods Applied Mechanical Engineering*, 139(14):347–373, 1996.

[44] S. Kitsionas and A. P. Whitworth. Smoothed particle hydrodynamics with particle splitting, applied to self-gravitating collapse. *Monthly Notices of the Royal Astronomical Society*, 330(1):129–136, 2002.

[45] J.R. Klepaczko and T. Lodygowski. *Advances in Constitutive Relations Applied in Computer Codes.* CISM International Centre for Mechanical Sciences, Italy, 2010.

[46] O. Kum, W.G. Hoover, and H.A. Posch. Viscous conducting flows with smooth-particle applied mechanics. *Physical Review E*, 109:67–75, 1995.

[47] L.D Landau and E.M. Lifshitz. *Theory of Elasticity.* Number Volume 7. Pergamon Press, 2nd ed. edition, 1970.

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

[49] C.H. Lee, A.J. Gil, and J. Bonet. Development of a cell centred upwind finite volume algorithm for a new conservation law formulation in structural dynamics. *Computers and Structures*, 118:13–38, 2013.

[50] S. Li and W.K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55(1):1–34, 2002.

[51] L. D. Libersky and A. G. Petschek. Smooth particle hydrodynamics with strength of materials, advances in the free lagrange method. *Lecture Notes in Physics*, 395:248–257, 1990.

[52] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi. High strain lagrangian hydrodynamics: a three dimensional sph code for dynamic material response. *Journal of Computational Physics*, 109:67–75, 1993.

[53] S.J. Lind, R. Xu, P.K. Stansby, and B.D. Rogers. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231:1499–1523, 2013.

[54] G. R. Liu and Y. T. Gu. A local radial point interpolation method (lr-pim) for free vibration analysis of 2-d solids. *Journal of Sound and Vibration*, 246(1):29–46, 2001.

[55] G. R. Liu and Y. T. Gu. A truly meshless method based on the strong-weak form. *Advances in Meshfree and X-FEM Methods*, in G.R. Liu (Ed.):259–261, 2002.

[56] G.R. Liu. *Mesh Free methods: moving beyond finite element method.* CRC Press, Boca Raton, 2002.

[57] G.R. Liu and M.B. Liu. *Smoothed particles hydrodynamics: A Meshfree Particle Method.* World Scientific Publishing, Singapore, 2004.

[58] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. archives of computational methods in engineering. *Monthly Notices of the Royal Astronomical Society*, 17(1):25–76, 2010.

[59] M.B. Liu, G.R. Liu, and K.Y. Lam. Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 155(2):263–284, 2003.

[60] M.B. Liu, G.R. Liu, Z. Zong, and K.Y. Lam. Numerical simulation of underwater explosion by sph. *In Atluti SN and Brust FW Advances in Computational Engineering and Science*, pages 1475–1480, 2000.

[61] M.B. Liu, G.R. Liu, Z. Zong, and K.Y. Lam. Computer simulation of the high explosive explosion using smoothed particle hydrodynamics methodology. *Computer and Fluids*, 32(3):305–322, 2003.

[62] M.B. Liu, G.R. Liu, Z. Zong, and K.Y. Lam. Smoothed particle hydrodynamics for numerical simulation of underwater explosions. *Computational Mechanics*, 30(2):106–118, 2003.

[63] W. K. Liu and Y. Chen. Wavelet and multiple-scale reproducing kernel methods. *International Journal for Numerical Methods in Fluids*, 21:901–931, 1995.

[64] W. K. Liu, Y. Chen, S. Jun, J.S. Chen, T. Belyschko, C. Pan, R.A. Uras, and C.T. Chang. Overview and applications of the reproducing kernel particle methods. *Archives of Computational Methods in Engineering State of the Art, Review*, 3:3–80, 1996.

[65] Y. R. Lopez and D. Roose. Particle refinement for fluid flow simulations with sph. In *Presented at Computer Methods in Mechanics, CMM-2011*, Warsaw, Poland, 2011.

[66] Y. R. Lopez, D. Roose, and C. R. Morfa. Dynamic particle refinement in sph: application to free surface flow and non-cohesive soil simulations. *Computational Mechanics*, 51(5):731–741, 2013.

[67] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82(12):1013–1024, 1977.

[68] M. Mabssout and M.I. Herreros. Runge-kutta vs taylor-sph: Two time integration schemes for sph with application to soil dynamics. *Appl. Math. Modell.*, page http://dx.doi.org/10.1016/j.apm.2012.08.014, 2012.

[69] A. Mayrhofer, B.D. Rogers, D. Violeau, and M. Ferrand. Investigation of wall bounded flows using sph and the unified semi-analytical wall boundary conditions. *Computer Physics Communications*, 184(11):2515–2527, 2013.

[70] J. J. Monaghan. Particle methods for hydrodynamics. *Computer Physics Report*, 3:71–124, 1985.

[71] J. J. Monaghan. An introduction to sph. *Computer Physics Communication*, 48:89–96, 1988.

[72] J. J. Monaghan. On the problem of penetration in particle methods. *Journal of Computational Physics*, 82:1–15, 1989.

[73] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomical and Astrophysics*, 30:543–574, 1992.

[74] J. J. Monaghan. Simulating free surface flows with sph. *Journal of Computational Physics*, 110:399–406, 1994.

[75] J. J. Monaghan. Sph without a tensile instability. *Journal of Computational Physics*, 159:290–311, 2000.

[76] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, 2005.

[77] J. J. Monaghan and Gingold R. A. Shock simulation by the particle method sph. *Journal of Computational Physics*, 52:374–389, 1983.

[78] J. J. Monaghan and J. C. Lattanzio. A refined particle method for astrophysical problems. *Astro. Astrophys*, 149:135–143, 1985.

[79] J. J. Monaghan and S. R. Varnas. The dynamics of interstellar cloud complexes. *Monthly Notices of the Royal Astronomical Society*, 231:515–534, 1988.

[80] J. P. Morris. Analysis of smoothed particle hydrodynamics with applications. *Monash University*, 1996.

[81] J. P. Morris. A study of the stability properties of smoothed particle hydrodynamics. *Publications of the Astronomical Society of Australia*, 13(1):97–102, 1996.

[82] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using sph. *Journal of Computational Physics*, 136:214–226, 1997.

[83] Y. X. Mukherjee and S. Mukherjee. Boundary node method for potential problems. *International Journal for Numerical Methods in Engineering*, 40:797–815, 1997.

[84] Y. X. Mukherjee and S. Mukherjee. On the boundary conditions in the element-free galerkin method. *Computational Mechanics*, 19:264–270, 1997.

[85] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element methods: diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.

[86] R. P. Nelson and J. C. B. Papaloizou. Variable smoothing lengths and energy conservation in smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 270:1–20, 1994.

[87] P. Omidvar, P.K. Stansby, and B.D. Rogers. Wave body interaction in 2d using smoothed particle hydrodynamics (sph) with variable particle mass. *International Journal for Numerical Methods in Fluids*, 68:686–705, 2012.

[88] P. Omidvar, P.K. Stansby, and B.D. Rogers. Sph for 3d floating bodies using variable mass particle distribution. *International Journal for Numerical Methods in Fluids*, 72:427–452, 2013.

[89] J. M. Owen, J. V. Villumsen, P. R. Shapiro, and H. Martel. Adaptive smoothed particle hydrodynamics methodology ii. *Astrophysical Journal Supplement Series*, 116:155–209, 1998.

[90] H.A. Posch, W.G. Hoover, and O. Kum. Steady-state shear flows via nonequilibrium molecular dynamics and smoothed-particle applied mechanics. *Physical Review E*, 52:1711–1719, 1995.

[91] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Number ISBN 978-0-521-88068-8. Cambridge University Press, 3rd ed. edition, 2007.

[92] P. W. Randles, T.C. Carney, L. D. Libersky, J.R. Renick, and A.G. Petschek. Calculation of oblique impact and fracture of tungsten cubes using smoothed particle hydrodynamics. *International Journal of Impact Engineering*, 17, 1995.

[93] P. W. Randles and L. D. Libersky. Smoothed particle hydrodynamics: some recent improvements and applications. *Computational Methods in Applied Mechanics and Engineering*, 139(1-4):375–408, 1996.

[94] P. W. Randles and L. D. Libersky. Normalized sph with stress points. *International Journal of Numerical Methods in Engineering*, 48:1445–1462, 2000.

[95] C. E. Rhoades. A fast algorithm for calculating particle interactions in smooth particle hydrodynamics. *Computer Physics Communications*, 70:478–482, 1992.

[96] M. Robinson. Turbulence and viscous mixing using smoothed particle hydrodynamics. *Monash University*, 2009.

[97] B.D. Rogers, R.A. Dalrymple, and P.K. Stansby. Simulation of caisson breakwater movement using 2-d sph. *Journal of Hydraulic Research*, 48:135–141, 2010.

[98] P. R. Shapiro, H. Martel, J. V. Villumsen, and J. M. Owen. Adaptive smoothed particle hydrodynamics, with application to cosmology: methodology. *Astrophysical Journal Supplement*, 103:269–330, 1996.

[99] J. C. Simpson. Numerical techniques for three-dimensional smoothed particle hydrodynamics simulations: application to accretion disks. *The Astrophysical Journal*, 448:822–831, 1995.

[100] A. Skillen, S. Lind, P.K. Stansby, and B.D. Rogers. Incompressible smoothed particle hydrodynamics (sph) with reduced temporal noise and generalised fickian smoothing applied to body?water slam and efficient wave?body interaction. *Computer Methods in Applied Mechanics and Engineering*, 265:163–173, 2013.

[101] F. Spreng, D. Schnabel, A. Mueller, and P. Eberhard. A local adaptive discretization algorithm for smoothed particle hydrodynamics. *Computational Particle Mechanics*, 1:131–145, 2014.

[102] V. Springel. Smoothed particle hydrodynamics in astrophysics. *Cosmology and Nongalactic Astrophysics*, 48:391–434, 2010.

[103] M. Steinmetz and E. Muller. On the capabilities and limits of smoothed particle hydrodynamics. *Astronomy and Astrophysics*, 268:391–410, 1993.

[104] D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.

[105] J. W. Swegle, D. L. Hicks, and S. W. Attaway. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116:123–134, 1995.

[106] J.W. Swegle and S.W. Attaway. On the feasibility of using smoothed particle hydrodynamics for underwater explosiion calculations. *Computational Mechanics*, 17:151–168, 1995.

[107] H. Takeda, S. M. Miyama, and M. Sekiya. Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Progress of Theoretical Physics*, 92(5):939, 1994.

[108] R. Vacondio, B.D. Rogers, and P.K. Stansby. Accurate particle splitting for smoothed particle hydrodynamics in shallow water with shock capturing. *Int. J. Numer. Meth. Fluids*, 69:1377–1410, 2012.

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

[110] R. Vacondio, B.D. Rogers, P.K. 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.

[111] D. Valdez-Balderas, J.M. Dominguez, B.D. Rogers, and A.J.C. Crespob. Towards accelerating smoothed particle hydrodynamics simulations for free-surface flows on multi-gpu clusters. *Journal of Parallel and Distributed Computing*, 73(11):1483–1493, 2013.

[112] J.G. Verwer, W.H. Hundsdorfer, and B.P. Sommeijer. Convergence properties of the Runge-Kutta-Chebychev method. *Numer. Math.*, 57:157–178, 1990.

[113] R. Vignjevic, J. Campbell, J. Jaric, and S. Powell. Derivation of sph equations in a moving referential coordinate system. *Comp Methods Appl Mech Eng*, 198(3032):24032411, 2009.

[114] R. Vignjevic, J.R. Reveles, and J. Campbell. Sph in a total lagrangian formalism. *CMES Comput Model Eng Sci*, 14(3):181198, 2006.

[115] J. von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamics shocks. *Journal of Applied Physics*, 21:232–247, 1950.

[116] J. G. Wang and G. R. Liu. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54:1623–1648, 2002.

[117] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389396, 1995.

[118] Q. Zhu, L. Hernquist, and Y. Li. Numerical convergence in smoothed particle hydrodynamics. *The Astrophysical Journal*, 2015.

[119] J.A. Zukas. *Impact dynamics*. John Wiley and Sons, New York, 1982.

[120] J.A. Zukas. *high velocity impact*. John Wiley and Sons, New York, 1990.

Appendix A

Appendix

A.1 Substantial Derivative

Before the derivation of governing equations, the physical meaning of substantial derivative should be introduced. For this we choose the infinitesimal moving fluid element model. and we assume the vector velocity (Figure 3.1) is,

$$\mathbf{v} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}. \tag{A.1}$$

Here bold-face notation is used to denote the vector quantities. **i**, **j** and **k** are the orthogonal directions of the Cartesian Coordinates.

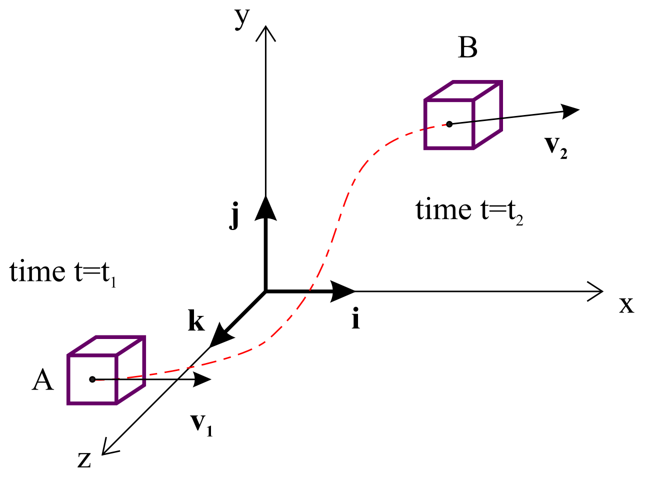


Figure A.1: Fluid element moving in the fluid flow.

The components in three dimensions are determined as,

$$u = u(x, y, z, t),$$

$$v = v(x, y, z, t),$$
$$w = w(x, y, z, t).$$

Here, we assume that the flow is unsteady and density changes with the time and space.

$$\rho = \rho(x, y, z, t)$$

Considering an infinitesimal element of fluid at time $t_1(A)$, the density of the fluid element can be presented as,

$$\rho_1 = \rho(x_1, y_1, z_1, t_1).$$

At time t_2, the same element has moved to B in Figure A.1. Therefore the density of this element at time t_2 is;

$$\rho_2 = \rho(x_2, y_2, z_2, t_2).$$

Using a Taylor Series to expand the density function $\rho = \rho(x, y, z, t)$ at point A. The Taylor Series is as follow,

$$f(x) = f(x_0) + \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

So we have,

$$\rho = \rho_1 + \left.\frac{\partial \rho}{\partial x}\right|_{x=x_1}(x - x_1) + \left.\frac{\partial \rho}{\partial y}\right|_{y=y_1}(y - y_1) + \left.\frac{\partial \rho}{\partial z}\right|_{z=z_1}(z - z_1) + \left.\frac{\partial \rho}{\partial t}\right|_{t=t_1}(t - t_1) + \ldots \quad \text{(A.2)}$$

ρ_2 can be expressed as follows

$$\rho_2 = \rho_1 + \left.\frac{\partial \rho}{\partial x}\right|_{x=x_1}(x_2 - x_1) + \left.\frac{\partial \rho}{\partial y}\right|_{y=y_1}(y_2 - y_1) + \left.\frac{\partial \rho}{\partial z}\right|_{z=z_1}(z_2 - z_1) + \left.\frac{\partial \rho}{\partial t}\right|_{t=t_1}(t_2 - t_1) + \ldots$$

Ignoring higher-order terms and dividing by $(t_2 - t_1)$, we have,

$$\frac{\rho_2 - \rho_1}{t_2 - t_1} = \left.\frac{\partial \rho}{\partial x}\right|_{x=x_1}\left(\frac{x_2 - x_1}{t_2 - t_1}\right) + \left.\frac{\partial \rho}{\partial y}\right|_{y=y_1}\left(\frac{y_2 - y_1}{t_2 - t_1}\right) + \left.\frac{\partial \rho}{\partial z}\right|_{z=z_1}\left(\frac{z_2 - z_1}{t_2 - t_1}\right) + \left.\frac{\partial \rho}{\partial t}\right|_{t=t_1}. \quad \text{(A.3)}$$

This gives the rate of change in density of an infinitesimal fluid element moving from A to B. If the step Δt between $t_1$and t_2 is very small, this term becomes

$$\lim_{t_2 \to t_1} \frac{\rho_2 - \rho_1}{t_2 - t_1} \equiv \left.\frac{D\rho}{Dt}\right|_{t=t_1}.$$

Note that, $D\rho/Dt$ is different from $\partial\rho/\partial t$. $D\rho/Dt$ is the time rate of change in density for the moving fluid element from $t_1(A)$ to $t_2(B)$. $\partial\rho/\partial t$ is the time rate of change in density for the fixed point A. The notation D/Dt is called the substantial derivative.

Because we have,

$$\lim_{t_2 \to t_1} \frac{x_2 - x_1}{t_2 - t_1} \equiv u,$$

$$\lim_{t_2 \to t_1} \frac{y_2 - y_1}{t_2 - t_1} \equiv v,$$

$$\lim_{t_2 \to t_1} \frac{z_2 - z_1}{t_2 - t_1} \equiv w.$$

When the time interval Δt is very small ($t_2 \to t_1$), the equation (A.3) will become;

$$\left.\frac{D\rho}{Dt}\right|_{t=t_1} = u\left.\frac{\partial \rho}{\partial x}\right|_{x=x_1} + v\left.\frac{\partial \rho}{\partial y}\right|_{y=y_1} + w\frac{\partial \rho}{\partial z}|_{z=z_1} + \left.\frac{\partial \rho}{\partial t}\right|_{t=t_1}.$$

This form for a particular point can be transfer into general form,

$$\frac{D\rho}{Dt} = u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z} + \frac{\partial \rho}{\partial t}. \tag{A.4}$$

Then the expression of substantial derivative can be extracted;

$$\frac{D}{Dt} = u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z} + \frac{\partial}{\partial t}. \tag{A.5}$$

Furthermore, the Nabla operator is the spatial gradient defined as follows;

$$\nabla = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}. \tag{A.6}$$

Then, combining (A.4) and (A.5), the substantial derivative can be expressed as,

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla. \tag{A.7}$$

Where D/Dt is the substantial derivative; $\partial/\partial t$ is the local derivative, which is the time rate of change at a fixed point; $\mathbf{v} \cdot \nabla$ means convective derivative, which is the time rate of change caused by the movement of the element in a field spatially different [1].

A.1.1 Divergence of the velocity

Consider a moving control volume of fluid with volume V (Figure A.2). The mass of the moving element is not changed by time and the element is made up of the same particles moving with the fluid flow. Since the element with the surface dS is moving with the velocity $\mathbf{v}$, the change of volume ΔV is due to the movement of dS during a time interval Δt.

$$\Delta V = [(\mathbf{v}\Delta t) \cdot \mathbf{n}]dS = (\mathbf{v}\Delta t) \cdot \mathbf{dS}, \tag{A.8}$$

where $\mathbf{n}$ is the unit vector to the surface at dS and the vector $\mathbf{dS}$ has the expression $\mathbf{dS} = \mathbf{n}dS$. If dS is small, the volume of the fluid element can equal $\int\int_s (\mathbf{v}\Delta t) \cdot dS$. Dividing by Δt, the result is the time rate of volume change;

$$\frac{DV}{Dt} = \frac{1}{\Delta t}\int\int_s (\mathbf{v}\Delta t) \cdot \mathbf{dS} = \int\int_s \mathbf{v} \cdot dS. \tag{A.9}$$

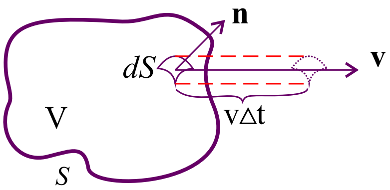


Figure A.2: Moving control volume for the physical interpretation of the divergence of velocity.

The divergence theorem is,

$$\int\int_{s} \mathbf{v} \cdot dS = \int\int\int_{V} (\nabla \cdot \mathbf{v}) dV.$$

Therefore, we can represent (A.9) as,

$$\frac{DV}{Dt} = \int\int\int_{V} (\nabla \cdot \mathbf{v}) dV. \tag{A.10}$$

Equation (A.10) is for the control volume. When the control volume is very small δV. The case becomes an infinitesimal moving fluid element case. We could rewrite (A.10) as,

$$\frac{D(\delta V)}{Dt} = \int\int\int_{\delta V} (\nabla \cdot \mathbf{v}) dV. \tag{A.11}$$

Assuming δV is small enough that $\nabla \cdot \mathbf{v}$ stays the same value throughout δV. Then we have,

$$\frac{D(\delta V)}{Dt} = (\nabla \cdot \mathbf{v}) \delta V,$$

or

$$\nabla \cdot \mathbf{v} = \frac{1}{\delta V} \frac{D(\delta V)}{Dt}. \tag{A.12}$$

A.2 Conservation of Mass

For mass conservation, we consider an infinitesimally small element moving with the fluid flow (right of Figure 3.1b). The mass of this small element is conserved, but the volume and density change during the volume moving process. We identify the mass of the small element as δm. So;

$$\delta m = \rho \delta V. \tag{A.13}$$

Because the element's mass is fixed, the substantial derivative of mass is zero.

$$\frac{D(\delta m)}{Dt} = 0$$

Combine these two equations above and do the ordinary derivation, we have

$$\frac{D(\rho\delta V)}{Dt} = \delta V\frac{D\rho}{Dt} + \rho\frac{D(\delta V)}{Dt} = 0$$

or

$$\frac{D\rho}{Dt} + \rho(\frac{1}{\delta V}\frac{D(\delta V)}{Dt}) = 0. \tag{A.14}$$

It is easy to see that the term in bracket of (A.14) is the same as divergence of the velocity ($\nabla \cdot \mathbf{v} = \frac{1}{\delta V}\frac{D(\delta V)}{Dt}$). So by combining (A.14) and (A.12),

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{v} = 0. \tag{A.15}$$

Expanding the equation above, it becomes,

$$\begin{aligned}\frac{D\rho}{Dt} &= -\rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\\ &= -\rho\left(\frac{\partial \mathrm{v}_x}{\partial x} + \frac{\partial \mathrm{v}_y}{\partial y} + \frac{\partial \mathrm{v}_z}{\partial x}\right).\end{aligned}$$

This equation can be rewritten in another form which will be used in SPH in the next section,

$$\frac{D\rho}{Dt} = -\rho\frac{\partial \mathrm{v}^\alpha}{\partial x^\alpha} = -\rho\nabla \cdot \mathbf{v}. \tag{A.16}$$

Note that the summation convention applies when a super-script appears twice in a term. That is, $\frac{\partial \mathrm{v}^\alpha}{\partial x^\alpha}$ implies $\left(\frac{\partial \mathrm{v}_x}{\partial x} + \frac{\partial \mathrm{v}_y}{\partial y} + \frac{\partial \mathrm{v}_z}{\partial z}\right)$.

A.3 Momentum equation

First we introduce the fundamental physical principle (Newton's second law),

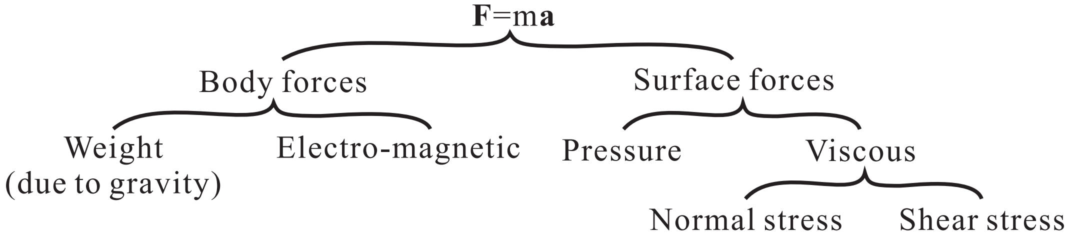
$$\mathbf{F} = m\mathbf{a}. \tag{A.17}$$

Assuming an infinitesimally small moving fluid element, applying Newton's second law here means that the sum of forces on the element is equal to the acceleration times the mass. We only consider the forces in the y direction first;

$$F_y = ma_y,$$

where the F_y and a_y are the y component of force and acceleration respectively.

There are two sources of this force: body force and surface force. The surface force, we also have two sources: pressure and viscosity. Assume the body force per unit mass



acting on the fluid element is f. So f_y is the body force in y direction. Thus the body force on fluid element in y direction is $\rho f_y(dxdydz)$.

Figure A.3 shows the force analysis in the y dimension of an infinitesimal moving fluid element. The pressures, including p and $(p + \frac{\partial p}{\partial y}dy)$ are the fluid pressure in y direction. The normal stresses, including σ_{yy} and $(\sigma_{yy} + \frac{\partial \sigma_{yy}}{\partial y}dy)$, are a suction from other elements to keep all the fluid elements staying together. The shear stresses, including σ_{xy}, σ_{zy}, $(\sigma_{xy}+\frac{\partial \sigma_{xy}}{\partial x}dx)$ and $(\sigma_{zy}+\frac{\partial \sigma_{zy}}{\partial z}dz)$, are frictional from other elements to move together. According to the force analysis Figure A.3, we obtain the surface force F_s in the y direction,

$$F_s = \bigg(-\frac{\partial p}{\partial y} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial z} \bigg) dxdydz. \tag{A.18}$$

The total force in the y direction F_y can be given by the sum of body force and surface force,

$$F_y = (-\frac{\partial p}{\partial y} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial z})dxdydz + \rho f_y(dxdydz). \tag{A.19}$$

Because $m = \rho dxdydz$, the acceleration in the y direction is a_y , a_y is equal to $\frac{Dv}{Dt}$. Then we find,

$$\rho\frac{Dv}{Dt} = \rho\frac{D\mathrm{v}_y}{Dt} = -\frac{\partial p}{\partial y} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial z} + \rho f_y.$$

Using a similar method, the equations of acceleration in x and z directions can be represented as,

$$\rho\frac{Du}{Dt} = \rho\frac{D\mathrm{v}_x}{Dt} = -\frac{\partial p}{\partial x} + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + \rho f_x,$$

$$\rho\frac{Dw}{Dt} = \rho\frac{D\mathrm{v}_z}{Dt} = -\frac{\partial p}{\partial z} + \frac{\partial \sigma_{zz}}{\partial z} + \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \rho f_z.$$

When the problem turns to solid mechanics, there is no pressure in a solid only a large stress $\boldsymbol{\sigma}$. Then we have;

$$\rho\frac{D\mathrm{v}_x}{Dt} = \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + \rho f_x, \tag{A.20a}$$

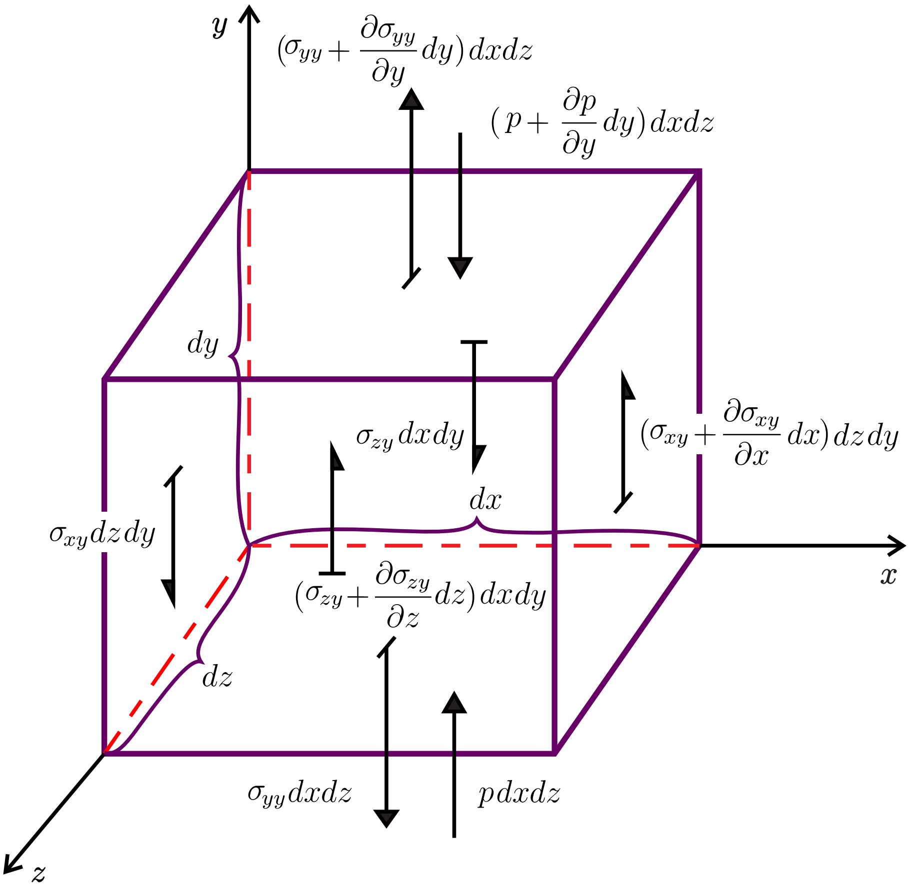


Figure A.3: Force analysis in y dimension.

$$\rho\frac{D\mathrm{v}_y}{Dt} = \frac{\partial\sigma_{yx}}{\partial x} + \frac{\partial\sigma_{yy}}{\partial y} + \frac{\partial\sigma_{yz}}{\partial z} + \rho f_y, \tag{A.20b}$$

$$\rho\frac{D\mathrm{v}_z}{Dt} = \frac{\partial\sigma_{zx}}{\partial x} + \frac{\partial\sigma_{zy}}{\partial y} + \frac{\partial\sigma_{zz}}{\partial z} + \rho f_z. \tag{A.20c}$$

The three equations (A.20a), (A.20b) and (A.20c) can be combined into one which is used in the next section.

$$\frac{D\mathrm{v}^\alpha}{Dt} = \frac{1}{\rho}\frac{\partial\sigma^{\alpha\beta}}{\partial x^\beta} + f^\alpha, \tag{A.21a}$$

or

$$\frac{D\mathbf{v}}{Dt} = \frac{1}{\rho}\nabla\boldsymbol{\sigma} + \mathbf{f}. \tag{A.21b}$$

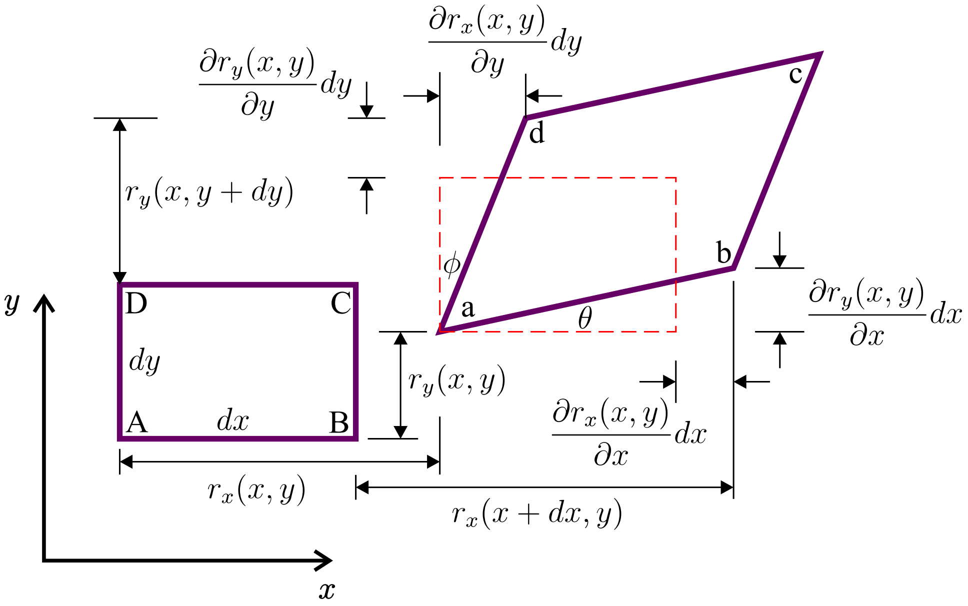


Figure A.4: The deformation analysis of a small element in two dimensions.

A.4 Stress tensor calculation

In a similar manner to stresses, strains also can be classified in two types, normal strain and shear strain. In an isotropic solid, Hooke's law states that normal strain is caused by normal stress. First, consider a two-dimensional problem Figure A.4. There is a small element with length dx and height dy. After a loading is applied, the element becomes a rhombus and moves some distance ($r(x, y)$). From the analysis in Figure A.4, we have the original size of the element,

$$\overline{AB} = dx \text{ and } \overline{CD} = dy.$$

Then after loading, the sizes become,

$$\begin{aligned}\overline{ab} &= \sqrt{\left(dx + \frac{\partial r_x(x,y)}{\partial x}dx\right)^2 + \left(\frac{\partial r_y(x,y)}{\partial x}dx\right)} \\ &= dx\sqrt{1 + 2\frac{\partial r_x(x,y)}{\partial x} + \left(\frac{\partial r_x(x,y)}{\partial x}\right)^2 + \left(\frac{\partial r_y(x,y)}{\partial x}\right)}.\end{aligned}$$

For a small displacement, the squares of the displacement gradient can be ignored, and we will obtain,

$$\overline{ab} \approx dx + \frac{\partial r_x(x,y)}{\partial x}dx.$$

The normal strain in the x-direction can be written as,

$$\varepsilon_{xx} = \frac{\overline{ab} - \overline{AB}}{\overline{AB}} = \frac{\partial r_x(x,y)}{\partial x}.$$

Similarly, transforming the equation in to three dimensions,we can get the normal strain in the y and z directions as well,

$$\varepsilon_{xx} = \frac{\partial r_x(x,y,z)}{\partial x}, \tag{A.22}$$

$$\varepsilon_{yy} = \frac{\partial r_y(x,y,z)}{\partial y}, \tag{A.23}$$

$$\varepsilon_{zz} = \frac{\partial r_z(x,y,z)}{\partial z}. \tag{A.24}$$

The definition of shear is the change of angle between $\overline{AB}$ and $\overline{AC}$. Therefore, the shear strain in Figure A.4 equals,

$$\gamma_{xy} = \theta + \phi.$$

For the analysis in Figure A.4, we find,

$$\tan\theta = \frac{\frac{\partial r_y(x,y)}{\partial x}dx}{dx + \frac{\partial r_x(x,y)}{\partial x}dx} = \frac{\frac{\partial r_y(x,y)}{\partial x}}{1 + \frac{\partial r_x(x,y)}{\partial x}},$$

$$\tan\phi = \frac{\frac{\partial r_x(x,y)}{\partial y}dy}{dy + \frac{\partial r_y(x,y)}{\partial y}dy} = \frac{\frac{\partial r_x(x,y)}{\partial y}}{1 + \frac{\partial r_y(x,y)}{\partial y}}.$$

Since the displacement gradient can be ignored, we find,

$$\frac{\partial r_x(x,y)}{\partial x} << 1; \quad \frac{\partial r_y(x,y)}{\partial y} << 1.$$

As the change in angle is very small, $\tan\theta \approx \theta$ and $\tan\phi \approx \phi$. Thus the angle values are,

$$\theta \approx \frac{\partial r_y(x,y)}{\partial x}; \quad \phi \approx \frac{\partial r_x(x,y)}{\partial y}.$$

Therefore, the shear strain can be represented as;

$$\gamma_{xy} = \theta + \phi = \frac{\partial r_y(x,y)}{\partial x} + \frac{\partial r_x(x,y)}{\partial y}.$$

Because ε_{xy} is equal to ε_{yx} and $\varepsilon_{xy} + \varepsilon_{yx} = \gamma_{xy}$, we have,

$$\varepsilon_{xy} = \varepsilon_{yx} = \frac{1}{2}\left(\frac{\partial r_y(x,y)}{\partial x} + \frac{\partial r_x(x,y)}{\partial y}\right).$$

We can apply this for all three dimensions, we can obtain,

$$\varepsilon_{xy} = \varepsilon_{yx} = \frac{1}{2}\left(\frac{\partial r_y(x,y,z)}{\partial x} + \frac{\partial r_x(x,y,z)}{\partial y}\right),$$

$$\varepsilon_{xz} = \varepsilon_{zx} = \frac{1}{2}\left(\frac{\partial r_z(x,y,z)}{\partial x} + \frac{\partial r_x(x,y,z)}{\partial z}\right),$$

$$\varepsilon_{yz} = \varepsilon_{zy} = \frac{1}{2}\left(\frac{\partial r_z(x,y,z)}{\partial y} + \frac{\partial r_y(x,y,z)}{\partial z}\right).$$

We can write the normal strain and shear strain into a short form;

$$\varepsilon^{\alpha\beta} = \frac{1}{2}\left(\frac{\partial r^{\alpha}}{\partial x^{\beta}} + \frac{\partial r^{\beta}}{\partial x^{\alpha}}\right). \tag{A.25}$$

and the time rate of strain can be presented as,

$$\dot{\varepsilon}^{\alpha\beta} = \frac{1}{2}\left(\frac{\partial \mathrm{v}^{\alpha}}{\partial x^{\beta}} + \frac{\partial \mathrm{v}^{\beta}}{\partial x^{\alpha}}\right). \tag{A.26}$$

The relationship between a strain tensor and a stress tensor can be represented by this matrix [27];

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{zx} \\ 2\varepsilon_{xy} \end{bmatrix} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix}\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix}, \tag{A.27}$$

where ν is Poisson's ratio and E is Young's Modulus. (A.27) can be represented as,

$$\varepsilon^{\alpha\beta} = \frac{1+\nu}{E}\sigma^{\alpha\beta} - \frac{\nu}{E}\delta^{\alpha\beta}\sigma^{kk}, \tag{A.28}$$

where σ^{kk} is equal to $\sigma_{xx} + \sigma_{yy} + \sigma_{zz}$ and $\delta^{\alpha\beta}$ is the Kronecker delta function,

$$\delta^{\alpha\beta} = \begin{cases} 1, & \alpha = \beta, \\ 0, & \alpha \neq \beta. \end{cases}$$

Therefore, the stress can be calculated by (A.27),

$$\sigma^{\alpha\beta} = 2\mu\varepsilon^{\alpha\beta} + \lambda\delta^{\alpha\beta}\varepsilon^{kk}, \tag{A.29}$$

where λ equals $\dfrac{2\nu}{1-2\nu}\mu$, μ equals $\dfrac{E}{2(1+\nu)}$ and ε^{kk} equals $\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}$.