UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

Smoothed Particle Hydrodynamics for Nonlinear Solid Mechanics

Ernesto Bram Ismail

20th September 2009

BISRU

cerecam
Centre for Research in Computational and Applied Mechanics

Abstract

Smooth Particle Hydrodynamics (SPH) is one of the simplest meshless methods currently in use. The method has seen significant development and has been the germination point for many other meshless methods [1,2]. The development of new meshless methods regularly uses standard SPH as a starting point, while trying to improve on issues related to consistency and stability. Despite these perceived flaws it is favoured by many researchers because of its simple structure and the ease with which it can be implemented.

The research presented here centres around an explicit, time dependent implementation of SPH. SPH is used to approximate nonlinear elasticity in one and two dimensions. The SPH implementation is based on total Lagrangian formalism [3] to achieve stability. The implementation of the numerical scheme is such that time dependent boundary conditions and simple constitutive non-linearities are implemented without any matrix inversion. The scheme relies wholly on matrix multiplication, allowing for simple conversion to a multi-processor environment.

The algorithm used forms the basis of the work presented, with issues of accuracy and implementation discussed in detail. Validation of the technique is via comparison with data available in literature, as well as via comparison with one dimensional wave propagation theory. Various features of the method are demonstrated with appropriate test configurations. Special attention is given to problems experiencing time dependent loading conditions.

"So long and thanks for all the fish ..."

Douglas Adams

Declaration

I hereby grant the University of Cape Town free licence to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever of the above dissertation. I am presenting this dissertation in partial fulfilment of the requirements for my degree.

I know the meaning of plagiarism and declare that all the work in this document, save for that which is properly acknowledged, is my own.

Signed this 20$^{\text{th}}$ day of September 2009

Ernesto Bram Ismail

Acknowledgements

I wish to thank the following people for their contribution to this thesis.

Professor B.D. Reddy, for his boundless knowledge and patience. May I one day have a corner office as nice as yours. Thanks for the last three years.

Professor G.N. Nurick, for the calm he instils in times of chaos. Thanks for watching out for me.

Mr T.J. Cloete, for the insight our conversations have brought me. One day you will have a drawer filled with my ideas, as I do now with yours.

The members of BISRU and CERECAM, what's grad-school with out the grads?

The awesome foursome: Samantha, Yaseen, Karl, and myself. "To the future ..."

My parents, for just being.

My friends, for making me who I am. "I'll see you in the Laboratory"

Tracy Booysen, for giving me a home. "Brains"

Hugo Krynauw, for giving me the mountains. "At the chains bru"

Melissa Loudon, for teaching me that life does not stop for your work. I owe you ...

Annie Brookstone, for being the one who dealt with me through the worst, and has given me the best. Words cannot tell the full story.

Table of Contents

List of Tables

List of Figures

Nomenclature

Upper Case Latin Symbols

$\boldsymbol{A}$	area in reference configuration, including orientation
A	area in reference configuration
$\boldsymbol{B}$	left Cauchy-Green tensor
$\mathcal{B}$	boundary particles
B	constant
$\boldsymbol{\mathcal{C}}$	Lagrangian elasticity tensor
$\boldsymbol{C}$	right Cauchy-Green tensor
C	constant
$C_\mathfrak{r}$	reflection coefficient
$C_\mathfrak{t}$	transmission coefficient
$\mathbf{D}$	modified SPH matrix for Dirichlet BCs
$\boldsymbol{E}$	Green-Lagrange strain
$\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3$	unit direction vectors in reference configuration
E	Young's modulus
$\boldsymbol{F}$	deformation gradient
$\mathcal{G}$	ghost particles
G	scaling factor based on dimension
$\boldsymbol{I}$	identity tensor
$\mathcal{I}$	internal particles
J	Jacobian
$\mathbf{M}$	modified SPH matrix for Neumann BCs with gradient in x_2 direction
$\hat{\boldsymbol{N}}$	vector normal to surface in reference configuration
$\mathbf{N}$	modified SPH matrix for Neumann BCs with gradient in x_1 direction
N	total number of particles within cover of a particular particle
$\boldsymbol{P}$	first Piola-Kirchhoff stress
$\mathcal{R}$	residual
$\boldsymbol{R}$	rotational portion of $\boldsymbol{F}$

$\boldsymbol{S}$	second Piola-Kirchhoff stress
$\boldsymbol{T}$	Cauchy stress
$\boldsymbol{U}$	stretch portion of $\boldsymbol{F}$
$\mathbf{W}$	SPH matrix
$\mathbf{X}$, $\mathbf{Y}$	SPH derivative matrices
$\boldsymbol{X}$	reference configuration

Lower Case Latin Symbols

$\boldsymbol{a}$	area in current configuration, including orientation
a	area in reference configuration
$\boldsymbol{b}$	body forces
b	boundary particle at which we wish to enforce the boundary condition
c	general wave speed
c_b	bar wave speed
c_L	longitudinal wave speed
d	scalar increase to the value of a function
$\boldsymbol{e}$	Euler-Almansi strain
$\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$	unit direction vectors in current configuration
$\mathbf{f}$	function value at every particle
$\mathfrak{f}$	altered SPH approximation of f
h	size of cover (smoothing length)
$\hat{\boldsymbol{n}}$	unit normal
p	surface traction
r	radial distance
$\bar{\boldsymbol{t}}$	value at Neumann boundary
$\boldsymbol{t}$	traction forces
t	time
$\dot{\boldsymbol{u}}$	velocity
$\overline{\boldsymbol{u}}$	value at Dirichlet boundary
$\mathbf{u}$	vector of displacement for each particle
$\boldsymbol{u}$	current displacement
$\boldsymbol{v}$	velocity
$\boldsymbol{x}$	positional vector
$\boldsymbol{x}'$	positional vector of general point

Functions

$\delta(\boldsymbol{x} - \boldsymbol{x}')$	Dirac delta
$W(\boldsymbol{x} - \boldsymbol{x}', h)$	cover function

Greek Symbols

$\boldsymbol{\varepsilon}$	infinitesimal strain
$\boldsymbol{\Psi}$	strain energy function
ϵ	arbitrarily small scalar
η	corrective factor to enforce scalar boundary condition
$\boldsymbol{\eta}$	corrective factor to enforce vector boundary condition
Γ	surface of domain
Γ_D	Dirichlet boundary
Γ_N	Neumann boundary
λ	dimension of domain
$\lambda,\ \mu$	Lamé constants
$\lambda_1, \lambda_2, \lambda_3$	principal stretches
ν	Poisson's ratio
Ω	domain
ϕ	motion
ρ	density
σ	one-dimensional Cauchy stress

Notation

$<>$	kernel approximation operator
$\square^T$	transpose
$\square^{-1}$	inverse
ΔV_q	volume of particle's cover
$\frac{\mathrm{D}\psi}{\mathrm{D}t}$	material time derivative

Subscripts

$\square_{\mathfrak{i},\mathfrak{r},\mathfrak{t}}$	induced, reflected, and transmitted wave
$\square_{A,B,C}$	reference configuration tensorial indices
$\square_{i,j,k,l}$	current configuration tensorial indices
$\square_p$	index denoting the particle at which a function is approximated
$\square_q$	index denoting a particle within the domain

Acronyms

BISRU	Blast Impact and Survivability Research Unit
FEM	finite element method
HPB	Hopkinson pressure bar
PDE	Partial Differential Equation

SPH	Smoothed Particle Hydrodynamics

Chapter 1

Introduction

Meshless methods have enjoyed significant exposure as an alternative to established computational methods such as finite elements. They are generally more flexible, and able to address issues in established methods. Significant progress has been made in the field, and yet it remains relatively opaque, with no dedicated proprietary codes available. A potential user must delve into the method, generally in great detail.

Smoothed Particle Hydrodynamics (SPH) was one of the first of such methods, and is lauded as being one of the simplest to use. Many papers have been written discussing fine details of computations performed under SPH, skipping the supposedly "trivial" process of generating computer code to perform the actual computations.

Works such as [4] discuss fine details such as convergence, accuracy of interpolation, and invariance to geometry discretisation (equivalent to meshing in FEM). While this detail is invaluable, and the basic equations are discussed in every paper, these works generally omit all detail on actual implementation, skipping from the most basic of equations describing the method to final results.

A computational mechanist, looking to use the method, must undergo a period of code development and verification for many months before even basic results can be reproduced.

This thesis aims to build a bridge between the basic formulation which is well known and relatively easy to understand, and the solutions published in the scientific literature. As such, the work cannot be regarded as novel, and yet it is hoped that it will be valuable to researchers wishing to enter the vast world of meshless methods.

This document has been developed in a manner that will lead the reader from history of SPH, to the basics of the method through some of the pitfalls that are universally acknowledged, to a relatively basic working code. Little emphasis is placed on using

every possible technique to ensure the results are the most accurate possible; instead, the focus is on the development of a logical method to implement the most basic method possible. That said, certain methods beyond the most basic have been used. Where this is the case the motivation for this inclusion is presented.

Meshless methods have been applied to a vast range of problems, each posing its own unique demands on the method. The aim of this thesis is to focus on problems in elastodynamics.

1.1 Meshless methods in computational mechanics

The modelling of dynamic material behaviour has been the domain of partial differential equations and continuum mechanics for many years. Nearly all complex problems are currently solved computationally, through the use of some numerical approach. For standard elasticity problems, as well as a host of other, more complex problems, the finite element method (FEM) is the current tool of choice. FEM relies on a fixed discretisation of the material, with a domain being broken up into elements having nodes at least at their vertices. As such, it is unsuited to the modelling of fluids or of solids that undergo large deformations, as the discretised 'mesh' quickly becomes distorted, resulting in poor field approximations [5]. Dynamic re-meshing of highly distorted elements is currently implemented by FEM users to circumvent these problems.

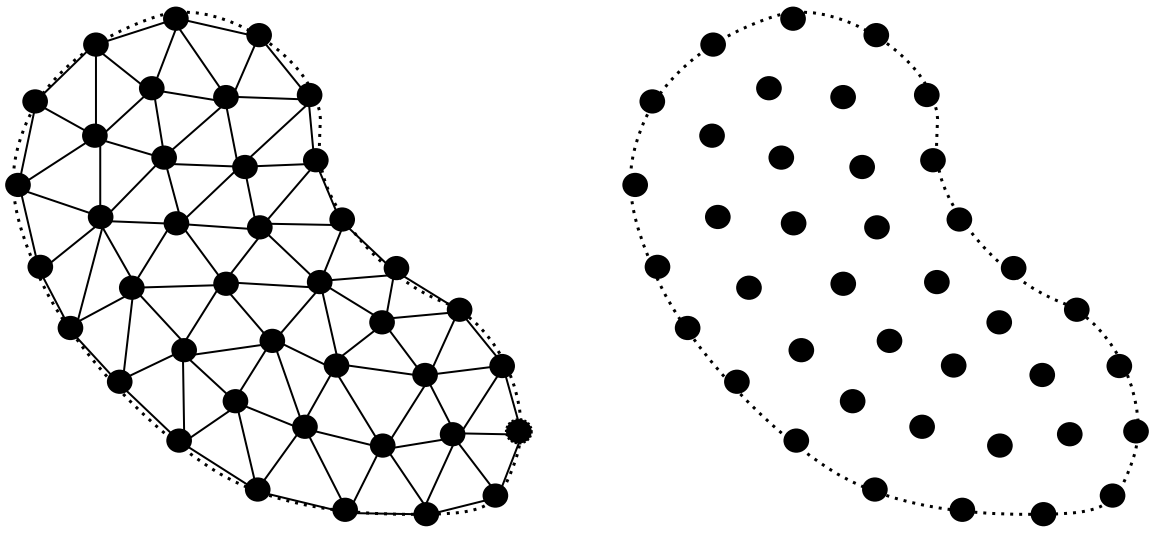


Figure 1.1: FEM and SPH approximations of a domain

Meshless methods aim to avoid the poor approximations associated with mesh distortion by removing the interconnection of nodes. A domain is discretised into nodes, but no strict connections are enforced. The integrity of the domain is maintained through the application of some interrelation of the nodes. This interrelation rule is generally a function of the distance between nodes.

The accuracy of the approximation is not determined by how nodes move relative to each other. Thus no difficulties are experienced due to mesh distortion, although there is a computational cost associated with the interrelation rule. Any change in the position of nodes will affect other properties of the domain, with nodes having a changing influence on other nodes, depending on position.

1.2 Smoothed Particle Hydrodynamics

The Smoothed Particle Hydrodynamics method is a fairly simple example of a meshless method [1]. It was first developed in the late 1970s to model astrophysical problems in three-dimensional space [6]. Due to the success of SPH in this field, it was extended to applications in computational mechanics by 1990 [cited in [7]]. It has since been utilised successfully to model various mechanics problems ranging from simple elasticity to complex fracture [8].

SPH has a remarkably simple formulation once fully derived, and one might expect an implementation to be direct. This is indeed the case for the most simple of cases, but problems are encountered early on. The first of these has to do with the imposition of *boundary conditions*. As SPH was developed for unbounded domains (as in astrophysics), the boundary term that arises in the formulation is treated as unimportant (as shall be discussed in Section 2.8 on page 30). Some special treatment of boundary conditions must be employed for bounded domains. This is in contrast to the finite element method where boundary conditions are naturally enforced. The second, and more vexing issue is that of the so-called *tensile instability* which can cause unbounded growth of the solution. This issue shaped the direction of the project and is covered in detail in Section 4.7 on page 84.

1.3 An in-house SPH solid mechanics code

In an attempt to develop expertise in the field of meshless methods it was decided to look at SPH in detail. The aim was to document the creation of an SPH code, fleshing out the details of implementation often ignored in the literature.

SPH has been used in blast and blast mitigation work [9, 10] and in impact (high strain rate) scenarios [11–13], and thus links closely with the work of the Blast Impact and Survivability Research Unit (BISRU) at the University of Cape Town.

1.4 Aims and structure of the thesis

This thesis is intended to serve as a guide to a simple implementation of SPH. As such, it is structured in a manner that closely follows an obvious developmental path. The general method of SPH is discussed before any detailed continuum mechanics. This is done in a totally abstract manner and could potentially be applied to any partial differential equation (PDE). This allows the reader to understand the basic assumptions and constraints imposed by the method.

A simple implementation of SPH for solid mechanics (most simply linear elasticity) was desired, but issues relating to the method required some deviation from this. Where the implementation, and consequences thereof, did require some deviation from linear elasticity these are discussed before some modification is motivated for and made.

The governing equations for finite strain elasticity are presented, followed by their SPH approximation. When the implementation requires a deviation from the case in question any additional theory required is presented before looking at the new implementation.

An overview of the final code is then given. The procedure is discussed in simple pseudo-code, with references to the actual code. Where applicable, issues of computational implementation are discussed.

The code is validated against fundamental theory. Here, several test cases are used to verify that the issues discussed previously are suitably dealt with. The validation tests are based on wave propagation as this links closely to the time-dependent solution method used.

Finally, a set of conclusions and recommendations are given. These discuss this project, and reflect on areas that require further attention

Structure

The structure of the rest of the thesis is as follows.

A mathematical description of the SPH approximation and a discussion of the influence of the fundamental variables of the system is presented in Chapter 2. The Governing equations for elasticity are presented in Chapter 3. An SPH approximation to the theory

presented in Chapter 3 is then presented in Chapter 4. This chapter additionally describes the validation procedure followed to verify the SPH implementation, and results from this testing. The implementation is then extended to nonlinear elastodynamics in Chapter 5. Again, the implementation is verified, and the results discussed. Finally, the results of the thesis are presented and recommendations are drawn from these.

Chapter 2

The SPH Method

SPH is formulated using an *integral representation* for *field function* approximations. In SPH the integral representation is known as the *kernel approximation*. This allows integrals to be approximated through the use of a weighting or smoothing function; this function allows the approximate value of the function at a point to be expressed in terms of the function's value at points surrounding the one in question.

This approach can be applied to any PDE. Despite this, PDEs with a spatial domain are most readily dealt with, as a particle discretisation is fairly natural.

Described simply, SPH discretises a domain into a collection of points, more generally referred to as particles, that contain all the properties of the domain. These particles have no connection with each other, and interact as a function of their distance apart. Each particle has an effect on any surrounding particles that fall into their sphere of influence. This sphere of influence is determined by the *cover of kernel function* which is defined such that the effects of nearby particles are greater than those of particles further away. This discretisation is shown in Figure 2.1.

The manner in which the influence of particles on each other diminishes is not especially important, as the approximation is intended to converge to the exact solution as the cover size is reduced. This is discussed fully in Section 2.1. Thus, as long as a domain is discretised in a "fine enough" manner, a reasonable approximation is made.

2.1 The kernel approximation

To introduce the kernel approximation formally it is best to consider the *weighting function* (also called the kernel), which determines the influence of the surrounding area on

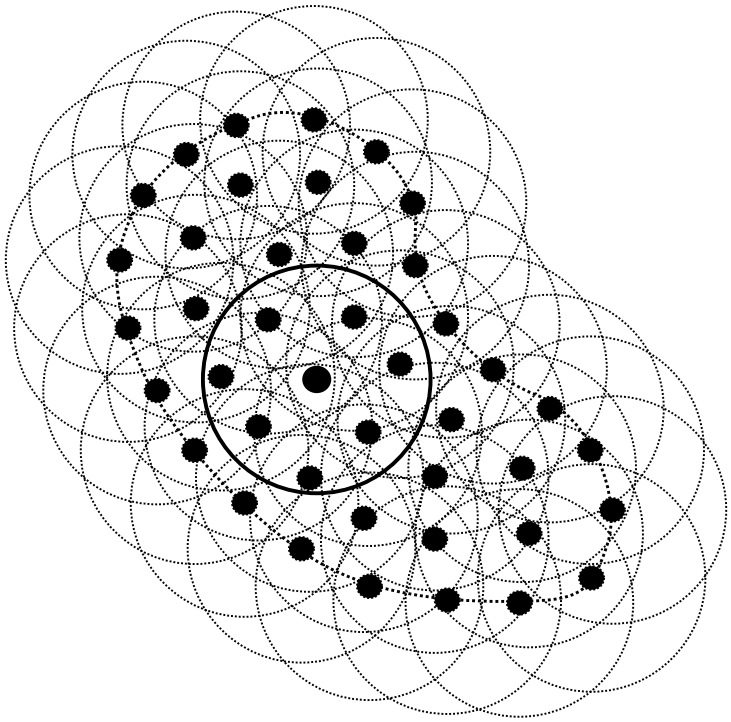


Figure 2.1: Influence of particles on each other

a particular point. The point of interest in the domain shall be denoted by $\boldsymbol{x}$. General points in the domain are represented by $\boldsymbol{x}'$.

Consider a function f, whose value of f at $\boldsymbol{x}$ can be expressed in the form

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

where

$$\delta(\boldsymbol{x} - \boldsymbol{x}') = \begin{cases} \infty & \boldsymbol{x} = \boldsymbol{x}' \\ 0 & \boldsymbol{x} \neq \boldsymbol{x}' \end{cases} \qquad \text{and} \qquad \int_{\Omega} \delta(\boldsymbol{x} - \boldsymbol{x}')\, \mathrm{d}\boldsymbol{x}' = 1 \tag{2.2}$$

is the Dirac delta.

The Dirac delta can be approximated by some function W, that satisfies the normalisation constraint, but which is defined over some finite distance h, referred to as the cover of the function. This allows an approximation of equation (2.1) to be constructed. Such a function is shown in Figure 2.2.

W is called the cover function and is defined to have the properties

$$W = W(\boldsymbol{x} - \boldsymbol{x}', h) \qquad \text{and} \qquad \int W\, \mathrm{d}\boldsymbol{x}' = 1. \tag{2.3}$$

The replacement of the Dirac delta with a cover function leads to an approximation of the function called the *kernel approximation*. This is denoted by angular brackets,<>. Thus

$$f(\boldsymbol{x}) \simeq < f > (\boldsymbol{x}) = \int_{\Omega} f(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}'. \tag{2.4}$$

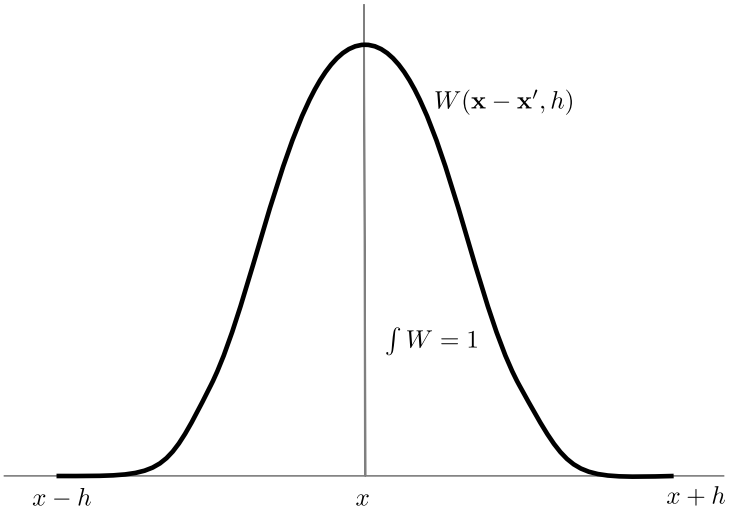


Figure 2.2: Approximation of Dirac delta

2.2 Cover functions

Because the kernel approximation is key to SPH it is worth looking at the selection of cover functions in slightly more detail. First we examine the properties for which cover functions are selected. We then present several cover functions that meet these requirements.

2.2.1 Properties of cover functions

The cover functions discussed in Section 2.1 are typically selected to meet a number of conditions. These conditions are not strict, in that one or more may ignored, but doing so may result in a change in the overall behaviour of the method. To make the approximation of f as accurate as possible it is important that W converges, in some sense, to the Dirac delta as the cover size is reduced.

The general conditions are that W must [8, page 39]:

- satisfy the delta function property. This ensures that as cover size (h) is reduced the SPH approximation increases in accuracy. This can be written in the form

$$\lim_{h\to 0} W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)=\delta\left(\boldsymbol{x}-\boldsymbol{x}'\right)\ . \tag{2.5}$$

- satisfy the unity condition. This ensures that the interpolation performed by using the cover function is conservative. This property is based on the property of the Dirac delta that:

$$\int_{\Omega}\delta\left(\boldsymbol{x}-\boldsymbol{x}'\right)\,\mathrm{d}\boldsymbol{x}'=1\ . \tag{2.6}$$

  This condition may be expressed as

$$\int_{\Omega}W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)\,\mathrm{d}\boldsymbol{x}'=1\ . \tag{2.7}$$

- have compact support. This ensures that particles are not affected by other particles that lie outside of a finite sphere of influence. This is written in the form

$$W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)=0 \text{ when } \left|\boldsymbol{x}-\boldsymbol{x}'\right|>h\ . \tag{2.8}$$

Using these assumptions we can make some generalisations of the kernel approximation. If we take the Taylor expansion of f about $\boldsymbol{x}$ (assuming f is smooth enough) we get

$$\begin{aligned}
<f>(\boldsymbol{x}) &= \int_{\Omega} f\left(\boldsymbol{x}'\right)W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)\,\mathrm{d}\boldsymbol{x}' \\
&= \int_{\Omega}\left[f\left(\boldsymbol{x}\right)+\nabla f\left(\boldsymbol{x}\right)\cdot\left(\boldsymbol{x}-\boldsymbol{x}'\right)+O\left(\left|\boldsymbol{x}-\boldsymbol{x}'\right|^{2}\right)\right]W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)\,\mathrm{d}\boldsymbol{x}' \\
&= f\left(\boldsymbol{x}\right)\int_{\Omega}W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)\,\mathrm{d}\boldsymbol{x}' \\
&\quad + f'\left(\boldsymbol{x}\right)\int_{\Omega}\left(\boldsymbol{x}-\boldsymbol{x}'\right)W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)\,\mathrm{d}\boldsymbol{x}'+\mathcal{R}\left(h^{2}\right)
\end{aligned} \tag{2.9}$$

where $\mathcal{R}$ is the residual.

If W is chosen to be radially symmetric (even in a one-dimensional context), as it usually is, we find that

$$\int_{\Omega}\left(\boldsymbol{x}-\boldsymbol{x}'\right)W\left(\boldsymbol{x}-\boldsymbol{x}',h\right)\,\mathrm{d}\boldsymbol{x}'=0\ . \tag{2.10}$$

This reduces equation (2.9) to

$$<f>(\boldsymbol{x})=f\left(\boldsymbol{x}\right)+\mathcal{R}\left(h^{2}\right)\ . \tag{2.11}$$

It is thus generally said that SPH is second-order accurate, provided that W is radially symmetric, and that the normalisation condition is satisfied. The latter condition is generally violated near the boundary even if the condition is satisfied within the domain. This is discussed in greater detail in Section 2.8 on page 30.

While the conditions listed above are there to allow for the derivation of the SPH approximation to a function, an additional condition is usually added to allow SPH to be extended to include gradients of f. This additional condition is that W must be differentiable within the cover area. This is discussed in full in Section 2.4 on page 17.

2.2.2 Some cover functions

There are endless possibilities for cover functions, with the most common functions being defined as radially symmetric. Radially defined cover functions result in an isotropic SPH interpolation field.

The most common cover functions are presented here. Each function is defined in terms of the radial distance r, which is defined by

$$r = \frac{|\boldsymbol{x} - \boldsymbol{x}'|}{h} \tag{2.12}$$

where h is once again the cover size.

Each function is shown in its one-dimensional form, together with its first two derivatives. Note that directional derivatives are not important in this one-dimensional case, but these become important in higher dimensions.

Lucy cover function

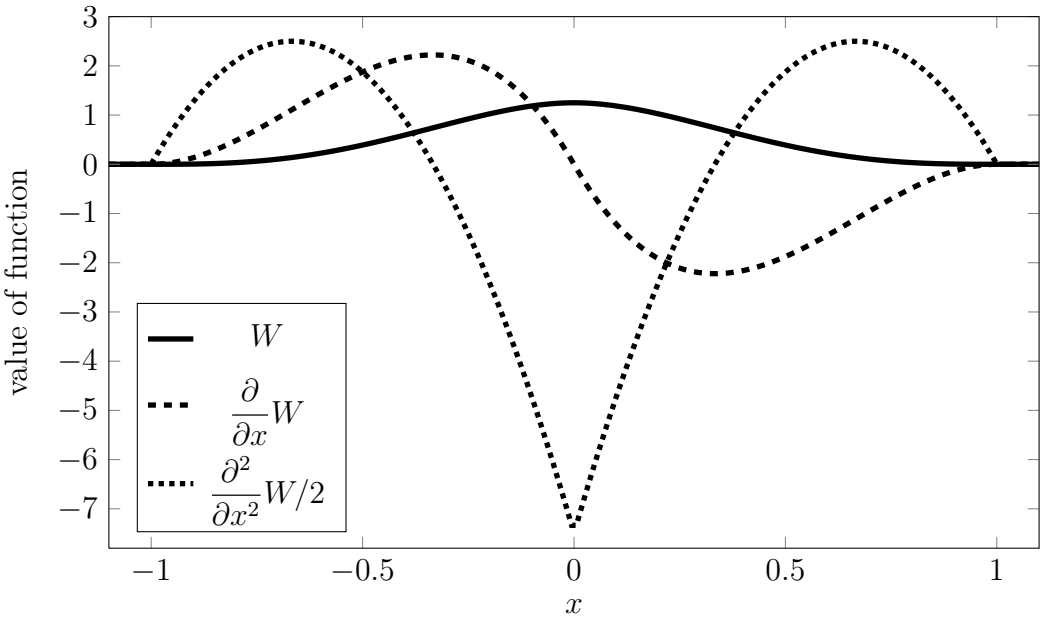


Figure 2.3: Lucy cover function

The function proposed by Lucy in 1977 [6], is defined by

$$W(r,h) = \begin{cases} \frac{G}{h^\lambda}\left[(1+3r)(1-r)^3\right] & \text{for } 0 \leq r < 1 \ , \\ 0 & \text{for } r \geq 1 \ , \end{cases} \tag{2.13}$$

where λ is the dimension of the domain. G is a normalisation factor based on λ, and is given by

$$G = \begin{cases} \frac{5}{4} & \text{for } \lambda = 1 \ , \\ \frac{5}{\pi} & \text{for } \lambda = 2 \ , \\ \frac{105}{16\pi} & \text{for } \lambda = 3 \ . \end{cases}$$

The function, as shown in Figure 2.3, meets the requirements of Section 2.2.1, but fell out of favour as some authors proposed that smoothing functions should resemble the Gaussian distribution as closely as possible [14].

The cubic B-spline

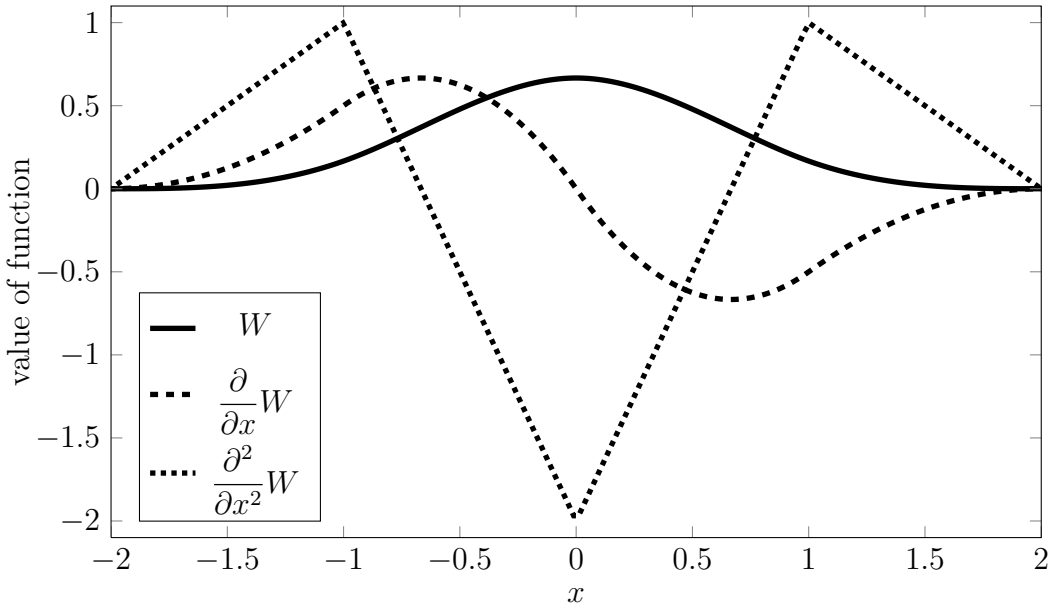


Figure 2.4: The cubic B-spline

The cubic B-spline was introduced by Monaghan [14] as a function that approximates the Gaussian distribution better than Lucy's function. It is defined by

$$W(r,h) = \begin{cases} \frac{G}{h^\lambda}\left[1 - \frac{3}{2}r^2 + \frac{3}{4}r^3\right] & \text{for } 0 \leq r < 1 \ , \\ \frac{G}{4h^\lambda}[2-r]^3 & \text{for } 1 \leq r < 2 \ , \\ 0 & \text{for } r \geq 2 \ , \end{cases} \tag{2.14}$$

where

$$G = \begin{cases} \frac{2}{3} & \text{for } \lambda = 1 \ , \\ \frac{10}{7\pi} & \text{for } \lambda = 2 \ , \\ \frac{1}{\pi} & \text{for } \lambda = 3 \ . \end{cases}$$

Monaghan suggested that the interrelation between a particle's properties and those of its neighbours would best be described by a Gaussian distribution.

The cubic B-spline, as shown in Figure 2.4, has a limitation in that it is only twice continuously differentiable. The lack of smoothness on the second derivative has been linked to instability [4].

The Gaussian or normal distribution

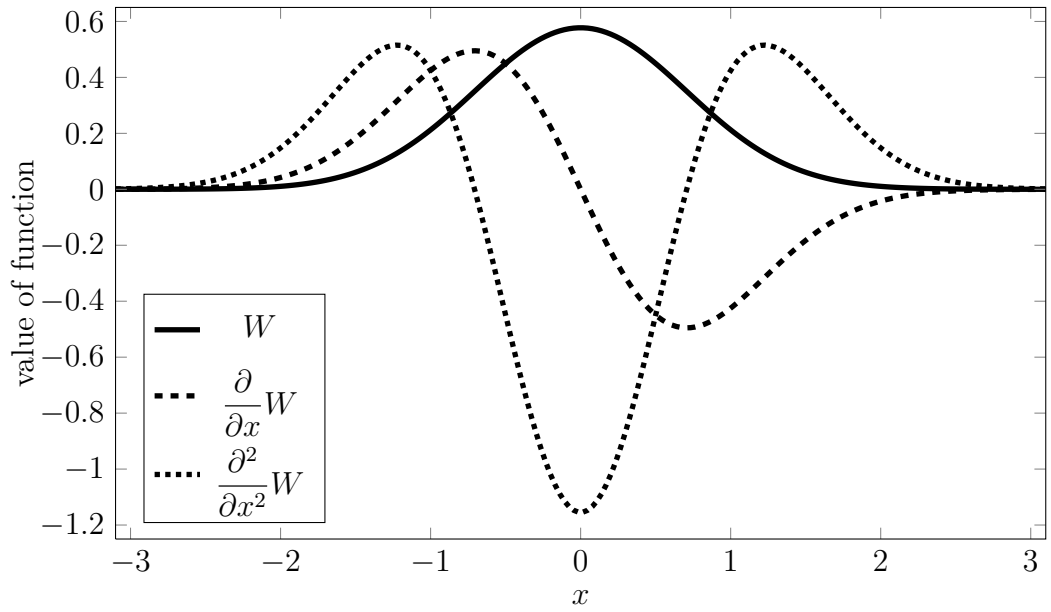


Figure 2.5: The Gaussian distribution

The Gaussian, or normal, distribution occurs in many large statistical data sets, and Monaghan [14] supposed that it was the most likely function to connect a particle to its neighbours. The distribution is given by

$$W(r,h) = \frac{G}{h^{\lambda}} e^{-r^2} \tag{2.15}$$

where

$$G = \begin{cases} \dfrac{1}{\sqrt{3}} & \text{for } \lambda = 1 \ , \\ \dfrac{1}{\pi} & \text{for } \lambda = 2 \ , \\ \dfrac{1}{\sqrt{\pi^3}} & \text{for } \lambda = 3 \ . \end{cases}$$

The Gaussian distribution does not have compact support since

$$W(r,h) \neq 0 \qquad \text{for all } r \ . \tag{2.16}$$

Despite not having compact support the Gaussian, shown in Figure 2.5, has been used for SPH, with the acknowledgement that the value of the function beyond some critical length is very small [8].

Mollifier cover function

In addition to the established cover functions above, the mollifier cover function is one that does not feature in the literature pertaining to SPH.

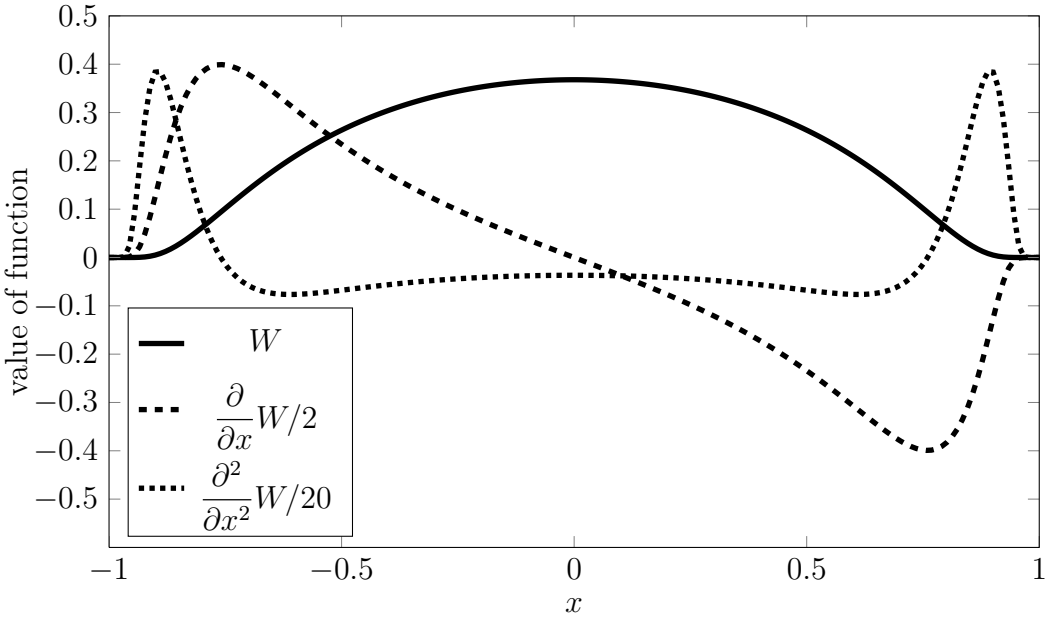


Figure 2.6: A mollifier cover function

The classical mollifier function, shown in Figure 2.6, is defined by

$$W(r,h) = \begin{cases} G\, e^{-1/(1-r^2)} & \text{for } 0 \leq r < 1 \ , \\ 0 & \text{for } r \geq 1 \ , \end{cases} \tag{2.17}$$

where

$$G = \frac{1}{\int W \, \mathrm{d}r} \ .$$

Mollifier functions are infinitely differentiable and have compact support. The function presented here is that discussed by Reddy [15], albeit for a different purpose. This function meets the criteria for a cover function but differ significantly from a normal distribution.

This function does not have a closed form integral, and thus the normalisation condition must be enforced using a numerical integration. This is acceptable for SPH approximations where h does not change, and the integral can be computed once. If, however, h is variable, a large computational load is associated with the use of such a cover function.

Modified mollifier cover function

In an effort to define an infinitely differentiable cover function, with compact support, and a profile as similar to the Gaussian distribution as possible a new mollifier cover function was devised. As only the shape of the Gaussian is of importance, this new function is compared to a scaled Gaussian function in Figure 2.7. Here the Gaussian has been scaled to a maximum value of $1/e$, and the radius modified such that

$$W_{Gaussian}\left(r,h\right) = e^{-(r/3)^2-1} \ . \tag{2.18}$$

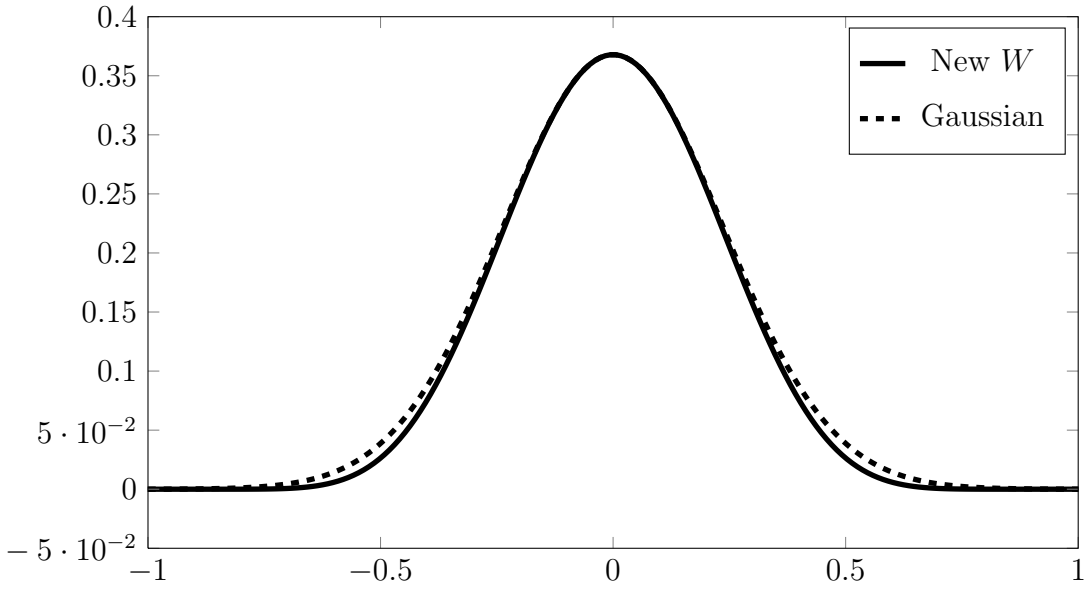


Figure 2.7: A modified mollifier cover function compared to a Gaussian cover function

This function was decided to be sufficiently close to the Gaussian cover function, and was thus used. The function can be defined as

$$W\left(r,h\right) = \begin{cases} G\, e^{-1/(1-r^2)}(1-r^2)^8 & \text{for } 0 \leq r < 1 \ , \\ 0 & \text{for } r \geq 1 \ , \end{cases} \tag{2.19}$$

where

$$G = 1 \,/ \int_{-1}^{-1} e^{-1/(1-r^2)}(1-r^2)^8 \, \mathrm{d}r \ .$$

Once again, this cover does not have a closed form integral, and must be normalised with a numerical one. Despite this, it was decided that the function met all the other requirements, and was a suitable cover function for simulations where the cover size is not varied. Figure 2.8 shows the function together with its derivatives (scaled for illustration).

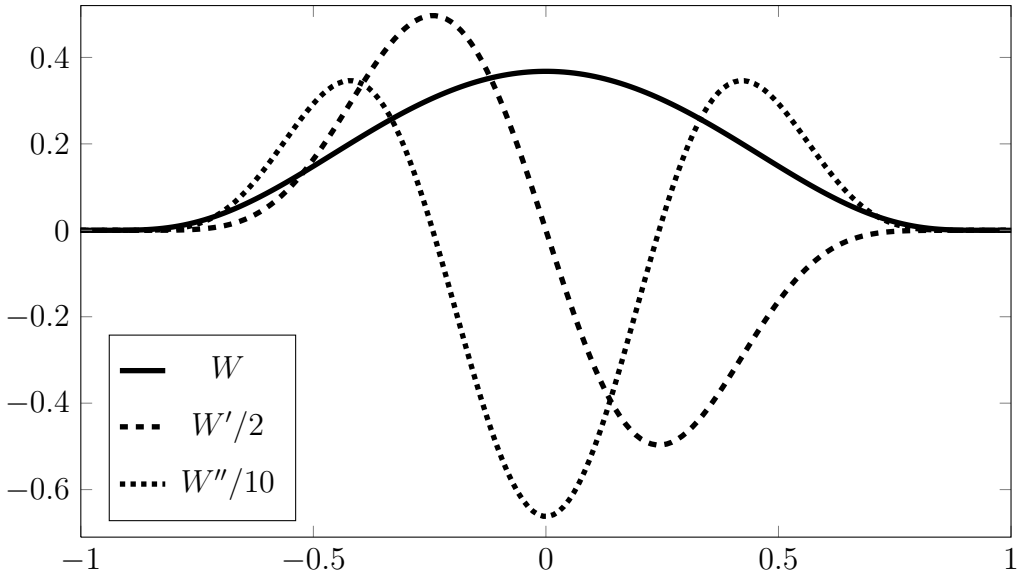


Figure 2.8: A modified mollifier cover function

2.3 Particle approximation and interpolation

The second key step in formulating SPH is the particle approximation.

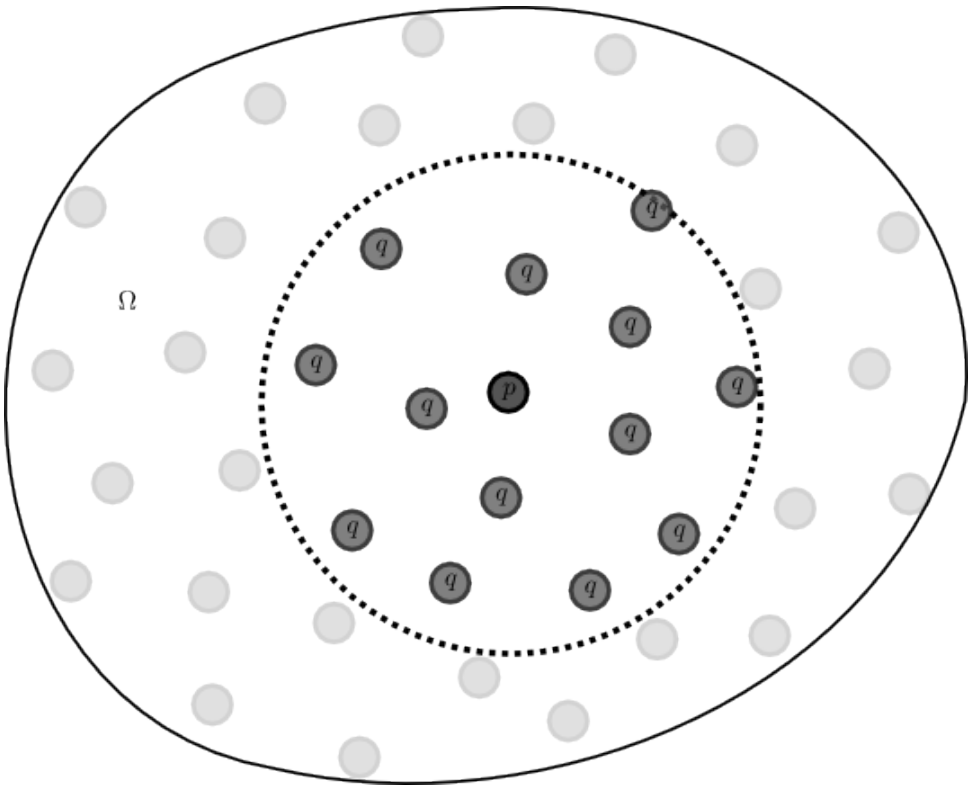


Figure 2.9: Particles q within the cover function of particle p

The domain is discretised into a collection of particles, as in Figure 2.9, and $< f >$ is

approximated according to

$$< f > (\boldsymbol{x}) = \int_{\Omega} f(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}' \tag{2.4}$$

$$\simeq \sum_{q=1}^{N} f(\boldsymbol{x}_q) W(\boldsymbol{x} - \boldsymbol{x}_q, h) \Delta V_q \tag{2.20}$$

where N is the total number of particles that fall within the support of the smoothing function of a particular particle and ΔV_q is the volume (or area in two dimensions) covered by that particle's cover function.

Equation (2.20) is the basis of SPH, and when applied to a function over some domain allows the value of the function at a particular point to be determined by the properties of particles that fall within that particle's sphere of influence.

2.4 Approximation of gradients

One of the strengths of SPH is the manner in which gradients are dealt with. In general, finding the SPH approximation of the gradient of a function is no more difficult than approximating the function itself.

To determine how gradients are dealt with all that is required is to replace f by its derivative in (2.20). This gives

$$< \frac{\partial f}{\partial x_i} > (\boldsymbol{x}) = \int_{\Omega} \left. \frac{\partial f}{\partial x_i} \right|_{\boldsymbol{x}'} W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}' \; . \tag{2.21}$$

Invoking the divergence theorem,

$$< \frac{\partial f}{\partial x_i} > (\boldsymbol{x}) = \int_{\Gamma} f(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \, \hat{n}_i \, \mathrm{d}s \ldots - \int_{\Omega} f(\boldsymbol{x}') \underbrace{\frac{\partial}{\partial x_i'} W(\boldsymbol{x} - \boldsymbol{x}', h)}_{\nabla' W(\boldsymbol{x} - \boldsymbol{x}', h)} \mathrm{d}\boldsymbol{x}' \tag{2.22}$$

where Γ is the boundary of the domain Ω and $\hat{\boldsymbol{n}}$ is the outward unit normal to this surface. For convenience, the differential operator acting on W is denoted ∇'.

Because we have selected a cover function that has compact support the first term in equation (2.22) vanishes. If the cover function overlaps the edge of the problem domain this is not the case and some special treatment must be employed. This is not of concern

however, as special treatment is required at the boundary in any case due to lack of consistency. These issues are discussed in detail in Section 2.8.

Provided we recognise the need for special treatment at the boundary we can now write:

$$< \nabla f > (\boldsymbol{x}) = -\int_{\Omega} f(\boldsymbol{x}') \nabla W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}' \simeq -\sum_{q=1}^{N} f(\boldsymbol{x}_q) \nabla' W(\boldsymbol{x} - \boldsymbol{x}_q, h) \Delta V_q \, . \tag{2.23}$$

By doing this we have shifted the calculation of the gradient of a function to the gradient of the known cover function.

Partial and higher derivatives in space can be similarly dealt with by shifting the derivative to the cover function.

2.5 General statement of SPH

Equations (2.20) and (2.23) give a discrete approximation of a function at any point in the domain, regardless of whether there is a particle at that point or not. We can associate this discrete approximation with a particle p as

$$< f > (\boldsymbol{x}_p) = \sum_{q=1}^{N} f(\boldsymbol{x}_q) W(\boldsymbol{x}_p - \boldsymbol{x}_q, h) \Delta V_q \tag{2.24}$$

and

$$< \nabla f > (\boldsymbol{x}_p) = -\sum_{q=1}^{N} f(\boldsymbol{x}_q) \nabla_q W(\boldsymbol{x}_p - \boldsymbol{x}_q, h) \Delta V_q \, . \tag{2.25}$$

Here a new notation $\nabla_q W(\boldsymbol{x}_p - \boldsymbol{x}_q, h)$ is introduced to denote which particle the gradient is taken with respect to: that is

$$\nabla_q W(\boldsymbol{x}_p - \boldsymbol{x}_q, h) = \frac{\partial}{\partial \boldsymbol{x}} W(\boldsymbol{x}_p - \boldsymbol{x}, \, h) \Big|_{\boldsymbol{x}=\boldsymbol{x}_p} \, .$$

SPH can be applied to combinations of functions to show that the kernel approximation is a linear operator.

Consider the function $g(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x})$:

$$\begin{aligned} < g > (\boldsymbol{x}_p) &= \sum_{q=1}^{N} g\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= \sum_{q=1}^{N} \left[f_1(\boldsymbol{x}_q) + f_2(\boldsymbol{x}_q)\right] W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= \sum_{q=1}^{N} \left[f_1(\boldsymbol{x}_q) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q + f_2(\boldsymbol{x}_q) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q\right] \\ &= \sum_{q=1}^{N} f_1(\boldsymbol{x}_q) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q + \sum_{q=1}^{N} f_2(\boldsymbol{x}_q) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= < f_1 > (\boldsymbol{x}_p) + < f_2 > (\boldsymbol{x}_p) \end{aligned} \tag{2.26}$$

Next consider the function $g(\boldsymbol{x}) = cf_1(\boldsymbol{x})$:

$$\begin{aligned} < g > (\boldsymbol{x}_p) &= \sum_{q=1}^{N} g\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= \sum_{q=1}^{N} \left[cf_1(\boldsymbol{x}_q)\right] W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= c \sum_{q=1}^{N} \left[f_1(\boldsymbol{x}_q) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q\right] \\ &= c < f_1 > (\boldsymbol{x}_p) \ . \end{aligned} \tag{2.27}$$

Equations (2.26) and (2.27) show that the kernel approximation is a linear operator.

Additionally the assumption

$$< f_1 \cdot f_2 > \simeq < f_1 > \cdot < f_2 > \tag{2.28}$$

is often used [16]. This can be seen not to be exact by considering the case where g is defined such that $g(\boldsymbol{x}) = f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})$:

$$\begin{aligned} < g > (\boldsymbol{x}_p) &= \sum_{q=1}^{N} g\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= \sum_{q=1}^{N} \left[f_1(\boldsymbol{x}_q) \cdot f_2(\boldsymbol{x}_q)\right] W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ &= < f_1 > (\boldsymbol{x}_p) \ \cdot f_2(\boldsymbol{x}_p) \qquad \text{or} \qquad f_1(\boldsymbol{x}) \cdot < f_2 > (\boldsymbol{x}) \end{aligned} \tag{2.29}$$

2.6 SPH matrices

The computation of the summation within the SPH approximation at any particle can be very expensive if implemented in a looping manner. It is possible to cast the SPH approximation as a simple matrix operation.

In order to do so we need to define a vector $\mathbf{f}$ such that

$$\mathbf{f} = \left\{ \begin{array}{c} f_1 \\ \vdots \\ f_N \end{array} \right\} \tag{2.30}$$

where $f_p \, (p = 1, \ldots, N)$ represents the value of f at $\boldsymbol{x}_p$.

We then define a matrix $\mathbf{W}$ such that

$$\mathbf{W} = \left[\left\{ \begin{array}{c} W_{11} \\ \vdots \\ W_{N1} \end{array} \right\} \cdots \left\{ \begin{array}{c} W_{1N} \\ \vdots \\ W_{NN} \end{array} \right\} \right] \bullet \times \left\{ \begin{array}{c} \Delta V_1 \\ \vdots \\ \Delta V_N \end{array} \right\} \tag{2.31}$$

where W_{pq} is the value of the cover function at p relative to q. Similarly ΔV_q is the volume contained by the cover function of particle q. In (2.31), $\mathbf{W}$ is formed by multiplying each column vector by the vector $\{\Delta V_1 \ldots \Delta V_N\}^T$.

If we multiply $\mathbf{W}$ and $\mathbf{f}$ we find that

$$\mathbf{Wf} = \left\{ \begin{array}{c} \sum_{q=1}^{N} f\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_1 - \boldsymbol{x}_q, h\right) \Delta V_q \\ \vdots \\ \sum_{q=1}^{N} f\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_N - \boldsymbol{x}_q, h\right) \Delta V_q \end{array} \right\} = \left\{ \begin{array}{c} < f_1 > \\ \vdots \\ < f_N > \end{array} \right\} \tag{2.32}$$

which returns the vector of SPH approximations to f.

We can similarly define matrices $\mathbf{X}$ and $\mathbf{Y}$ by

$$\mathbf{X} = \left[\left\{ \begin{array}{c} \dfrac{\partial W_{11}}{\partial x_1} \\ \vdots \\ \dfrac{\partial W_{N1}}{\partial x_1} \end{array} \right\} \cdots \left\{ \begin{array}{c} \dfrac{\partial W_{1N}}{\partial x_1} \\ \vdots \\ \dfrac{\partial W_{NN}}{\partial x_1} \end{array} \right\} \right] \bullet \times \left\{ \begin{array}{c} \Delta V_1 \\ \vdots \\ \Delta V_N \end{array} \right\} \tag{2.33}$$

$$\mathbf{Y} = \left[\left\{ \begin{array}{c} \dfrac{\partial W_{11}}{\partial x_2} \\ \vdots \\ \dfrac{\partial W_{N1}}{\partial x_2} \end{array} \right\} \cdots \left\{ \begin{array}{c} \dfrac{\partial W_{1N}}{\partial x_2} \\ \vdots \\ \dfrac{\partial W_{NN}}{\partial x_2} \end{array} \right\} \right] \bullet \times \left\{ \begin{array}{c} \Delta V_1 \\ \vdots \\ \Delta V_N \end{array} \right\} \tag{2.34}$$

which define the SPH directional derivatives with respect to x_1 and x_2.

2.7 SPH for interpolation

The methodology given in the previous section deals with the SPH approximation of known values, or of the computation of an approximate gradient. In order to understand the limitations of the method when used to solve PDEs, one must first understand the fundamental behaviour of the method.

The approximation of a known function is called interpolation. Equation (2.24) is effectively an interpolation formula. While the ability to do direct interpolation may not appear to be particularly impressive, the ease with which the gradient of this interpolation might be computed is noteworthy.

Two parameters control how well the method performs the interpolation. These are the total number of particles included within the domain, and the number of particles included within the cover of each particle. As with all numerical approximations one expects the approximation to improve with a finer discretisation (more particles). The number of particles within the cover is a slightly more complex issue. In general, too few particles results in a poor approximation of the gradient, while too many particles results in "over-smoothing", with details being lost in the primary approximation, and thus the subsequent computed gradients. This error can be expressed [17] as

$$Error \sim h^2, \left(\frac{\delta x}{h}\right)^2 . \tag{2.35}$$

The performance of the method in this regard is best evaluated relative to a simple function, with a known gradient. Zhang and Batra [7] present the SPH approximation of the function

$$f(x) = (x - 0.5)^4$$

over the range $(0, 1)$. These authors present results where the interpolation field is continued up until the last particle. Without some special treatment at these boundary and near-boundary particles, the approximation is significantly worse than expected. This is due to a reduction in the total number of particles within the cover of these particles. In general, however, most particles are well away from any boundary and the behaviour of interpolation away from the boundary can be determined by approximating the function to greater limits, and only looking at the range of interest.

For demonstration of the under-integration behaviour, a single interpolation result is given in Figure 2.10. The notation "ps" in the legend of the following figures denotes the particle spacing, *i.e.* the distance between two adjacent particles. In this figure, the interpolation near the boundary is *deficient* because the SPH approximation relies on sufficient particles to be present within the cover of a particle. If too few particles are present but the contribution of the included particles is normalised as normal, the approximation can have an error of up to 50% at the boundary. This "drooping" of the approximation of the function has a large effect on the subsequent approximations of derivatives.

This effect, and the imposition of boundary conditions when solving PDEs, are discussed in detail in Section 2.8. To address the problem occurring near the boundary, the remaining results are obtained by extending the function beyond the end points $x = (0, 1)$. This methodology is referred to as the *ghost particle* approach.

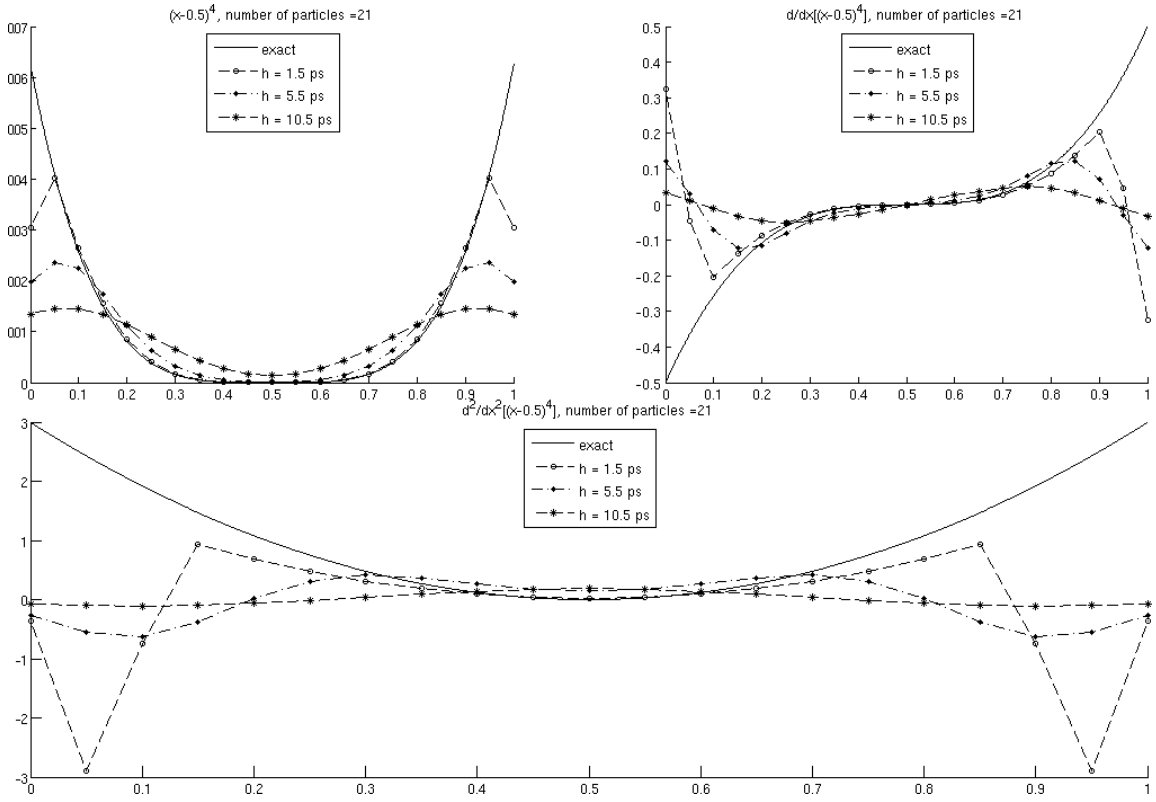


Figure 2.10: Interpolation of $(x - 0.5)^4$ - no extra particles

Where a large number of particles (as in Figure 2.11) is included in the interpolation most cover sizes interpolate the function and its first and second derivatives well. The very small cover size of 1.5 times the particle spacing does not approximate the gradients well, and this suggests that this cover size is not suitable for slowly changing functions. Here the second gradient is found by finding the gradient of the already computed first gradient. One could find the second gradient directly from SPH, but as this is not possible

in the dynamic simulations, this has not been presented here.

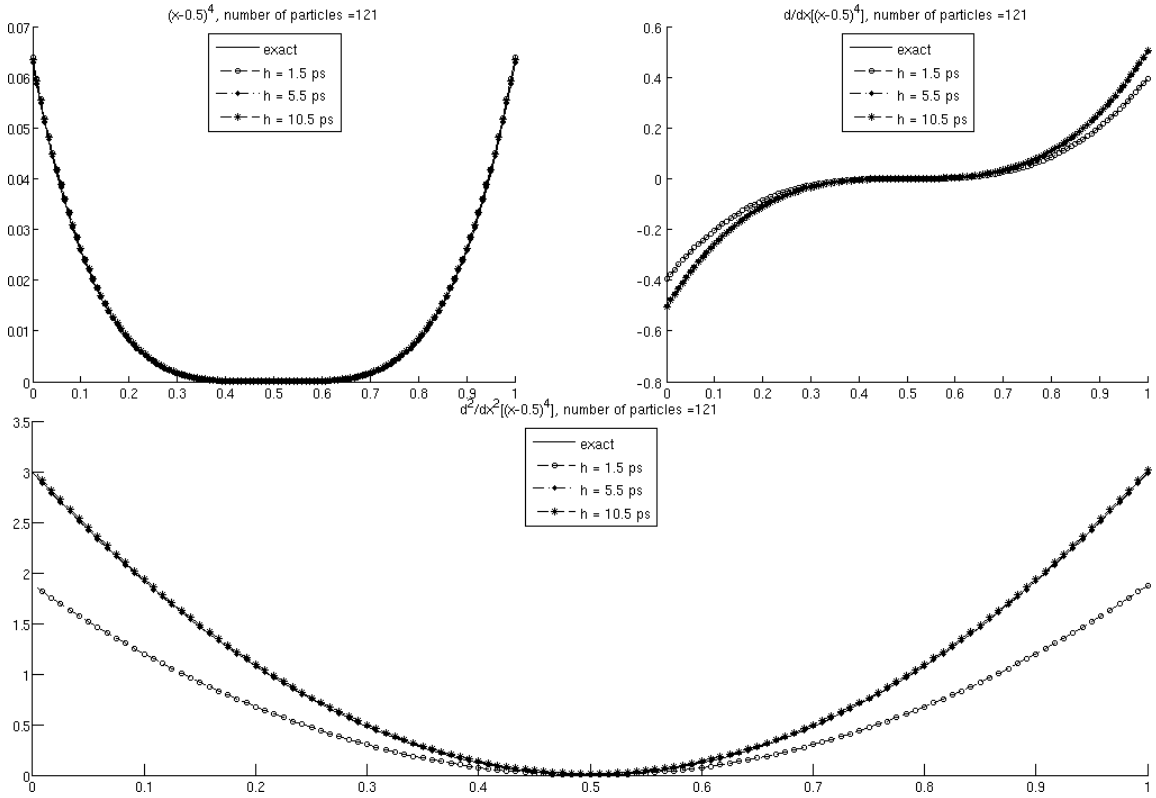


Figure 2.11: Interpolation of $(x - 0.5)^4$ - fine discretisation

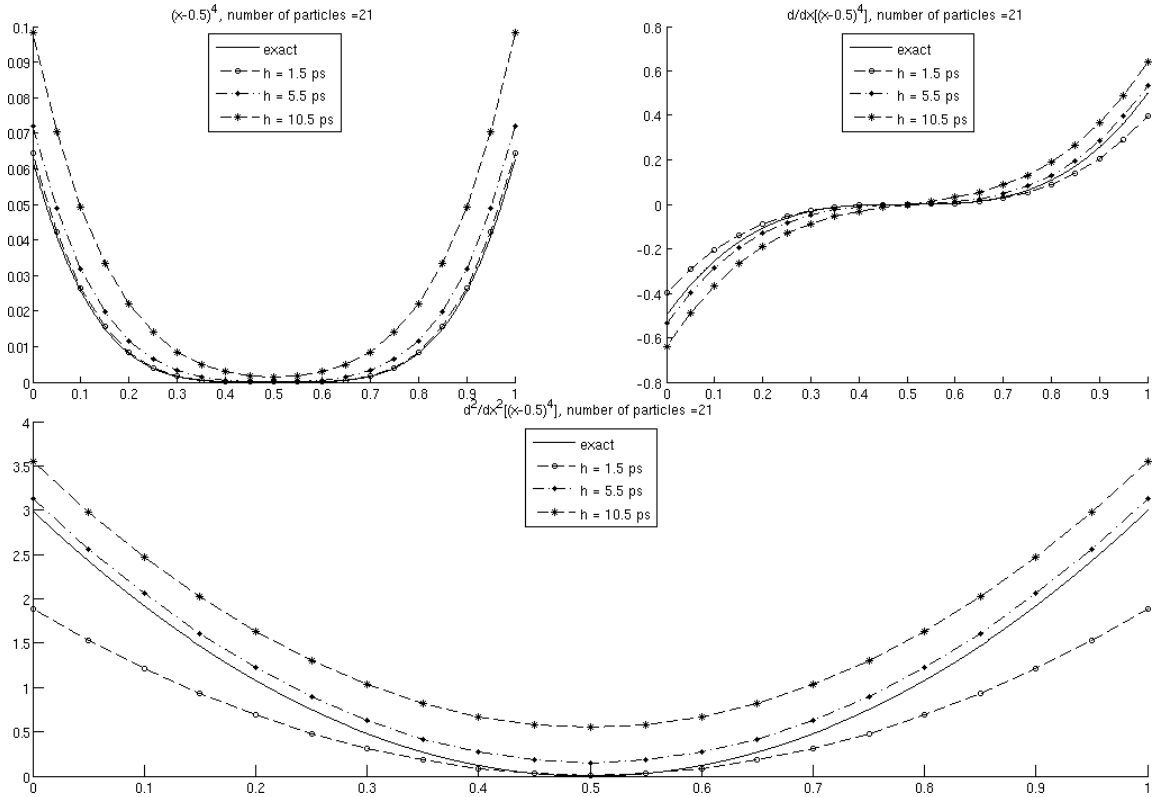


Figure 2.12: Interpolation of $(x - 0.5)^4$ - coarse discretisation

If the domain is discretised with very few particles, as in Figure 2.12, substantial errors are seen in the interpolation of the function, regardless of the cover sizes used. This suggests that it is essential to choose a discretisation that will provide approximations of a desired accuracy.

2.7.1 Interpolation of $\sin x$

The function $f(x) = \sin x$ is a good one to examine in detail, since f and its derivatives are periodic and all have the same amplitude. Here, any accumulated error is easily identified. Figures 2.13 and 2.14 are defined from $x = (0, 3\pi)$ to show the periodic behaviour.

Once again, a fine discretisation (as in Figure 2.13) results in a good interpolation approximation. The gradients, however, see an amplitude reduction. This behaviour is typical of SPH, and an increased number of particles reduces the error. The converse is seen in Figure 2.14, where a reduction in particle density (*i.e.* fewer particles per unit length) sees significant errors introduced into the interpolation. Once again, a cover size of 1.5 times the particle spacing performs poorly, albeit better than the other particle spacing.

Figures 2.15 and 2.16 show the approximation of the function and its gradients at a finer resolution, keeping the total number of particles in the domain the same, while reducing the interpolation range from $x = (0, 3\pi)$ to $x = (0, \pi)$. It is interesting to note in Figure 2.16 that small cover sizes approximate the function well, but the derivative thereof poorly. This phenomenon is due to a large reduction in the information available for the computation of gradients (with a cover size of 1.5 times the particle spacing only two points are used to evaluate the gradient).

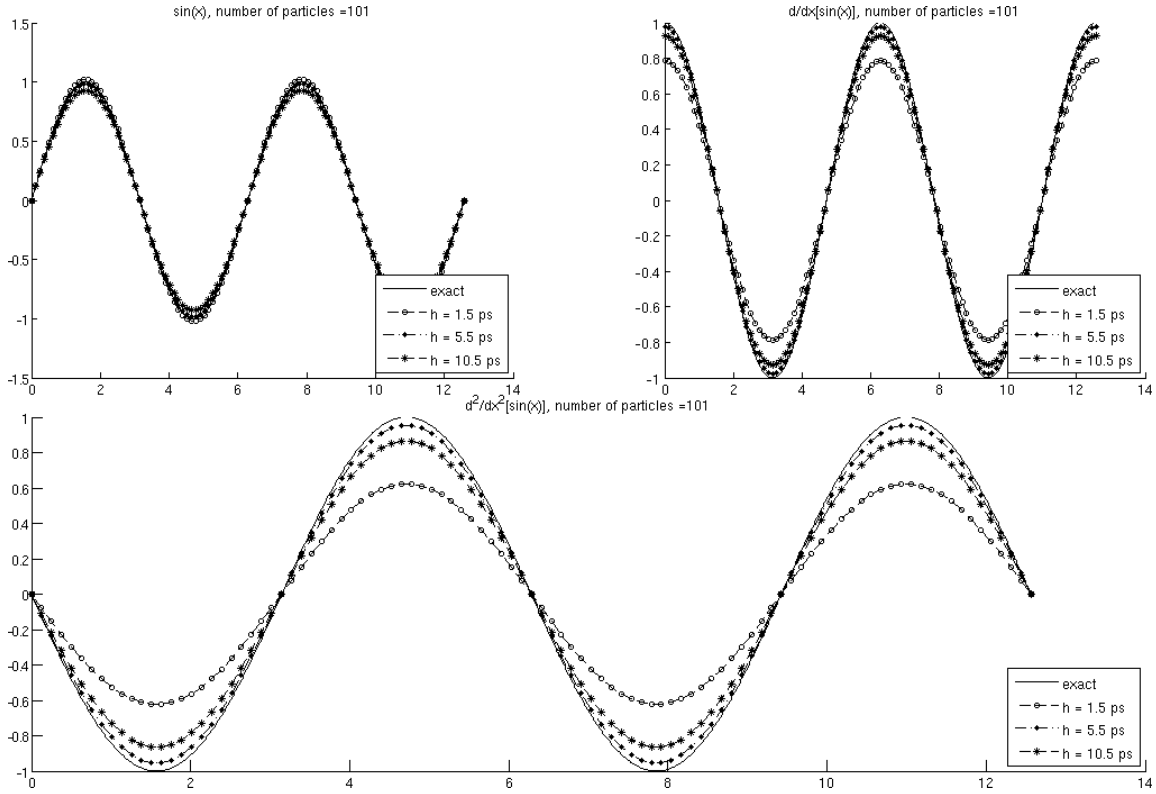


Figure 2.13: Interpolation of $\sin x$ - fine discretisation

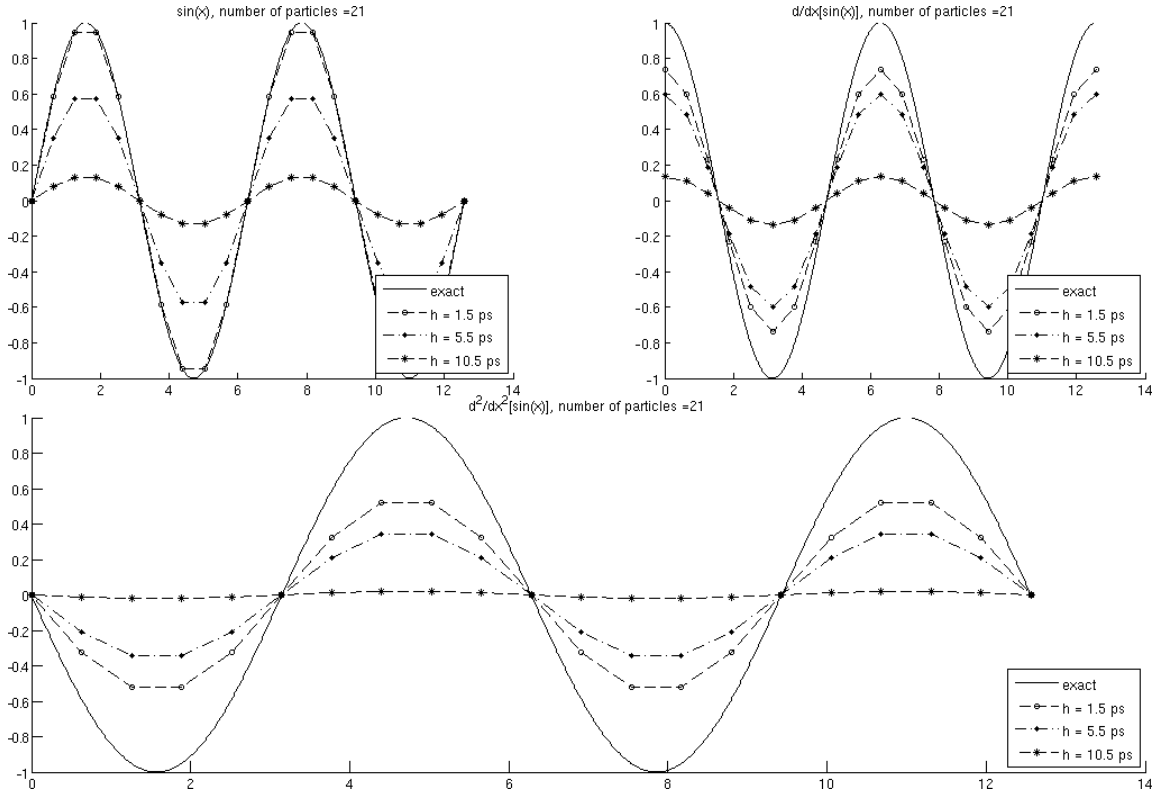


Figure 2.14: Interpolation of $\sin x$ - coarse discretisation

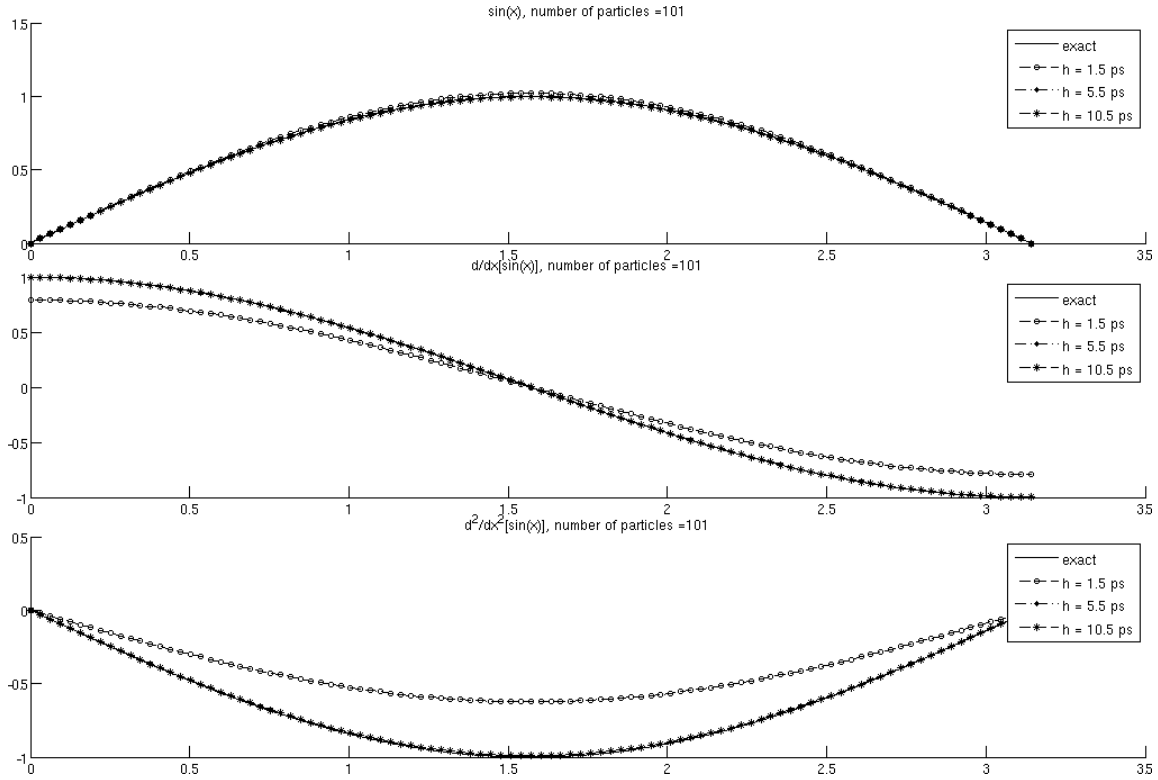


Figure 2.15: Interpolation of $\sin x$ - very fine discretisation

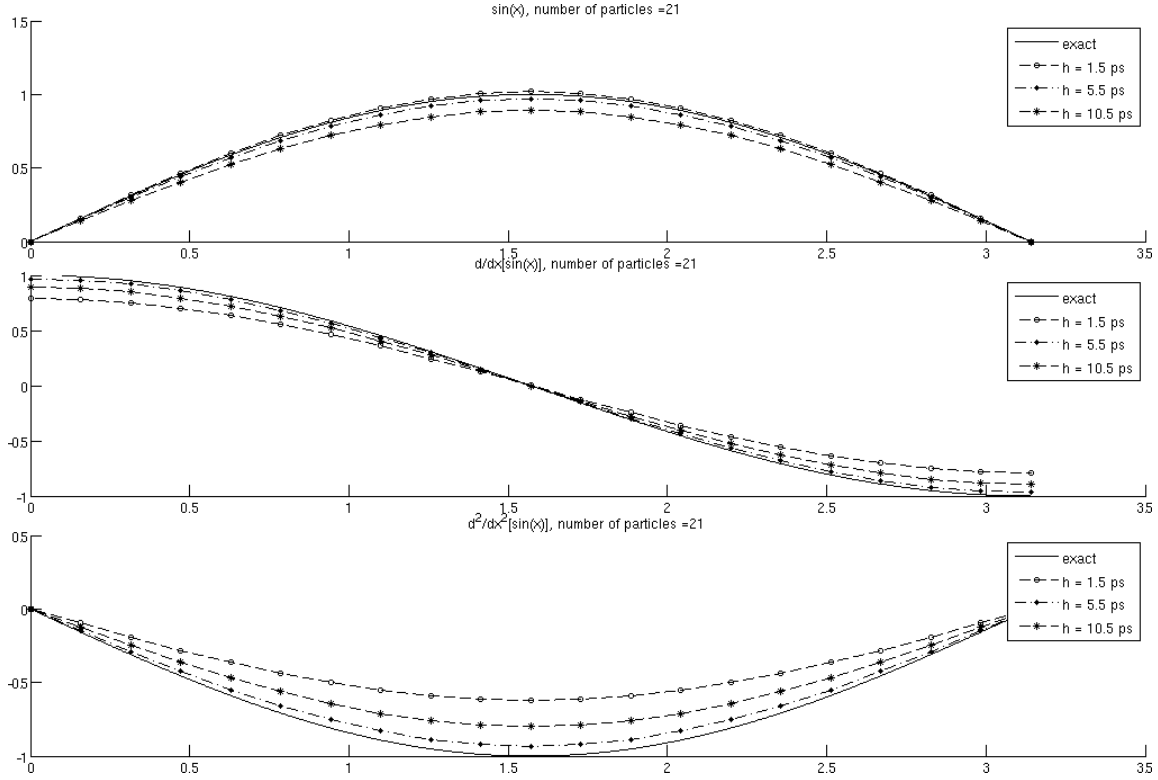


Figure 2.16: Interpolation of $\sin x$ - intermediate discretisation

2.7.2 Approximation of rapid changes

In the examples presented in the previous sections the function has varied over many times the particle spacing. In dynamic simulation this sort of function is not the norm. Functions can change dramatically in space of a few particles, and square waves are common loading conditions.

A series of tests is presented to assess the ability of SPH to deal with rapidly changing functions. Here a square wave is approximated with a linear falling edge of a known gradient. The gradient of the entire field is known, with two discontinuities where the falling edge meets the rest of the function.

For a low gradient (Figure 2.17), all of the cover sizes perform well, with a cover size of 1.5 times the particle spacing under-predicting the gradient as before. As the gradient is reduced (Figure 2.18), however, all of the cover sizes start to behave in a similar manner. Interestingly, the cover size of 1.5 times the particle spacing rapidly starts out-performing other cover sizes (Figure 2.19).

When the gradient is very high (Figure 2.20), a cover size of 1.5 times the particle spacing approximates the gradient very well, while the others are significantly different. Chen *et al* [18] use this cover size in their simulations, and it is postulated that this is done to suit the square wave loading functions imposed in the work.

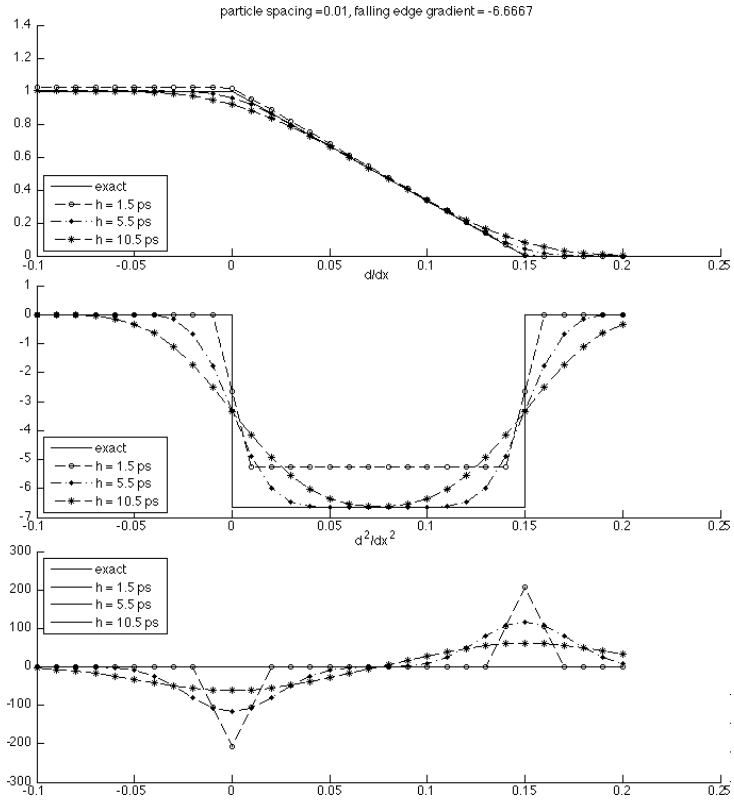


Figure 2.17: Approximation of a falling edge, very low gradients

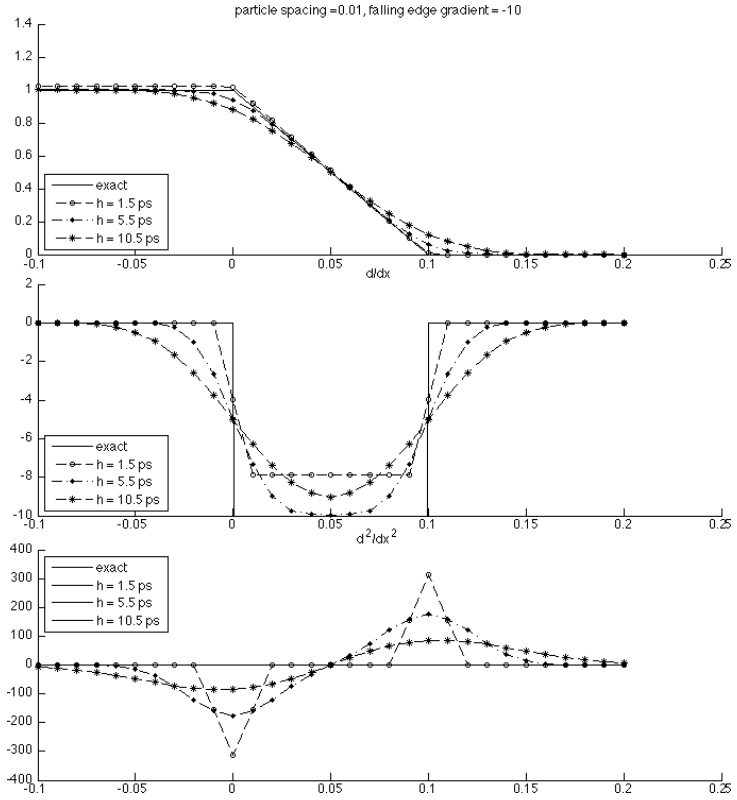


Figure 2.18: Approximation of a falling edge, low gradients

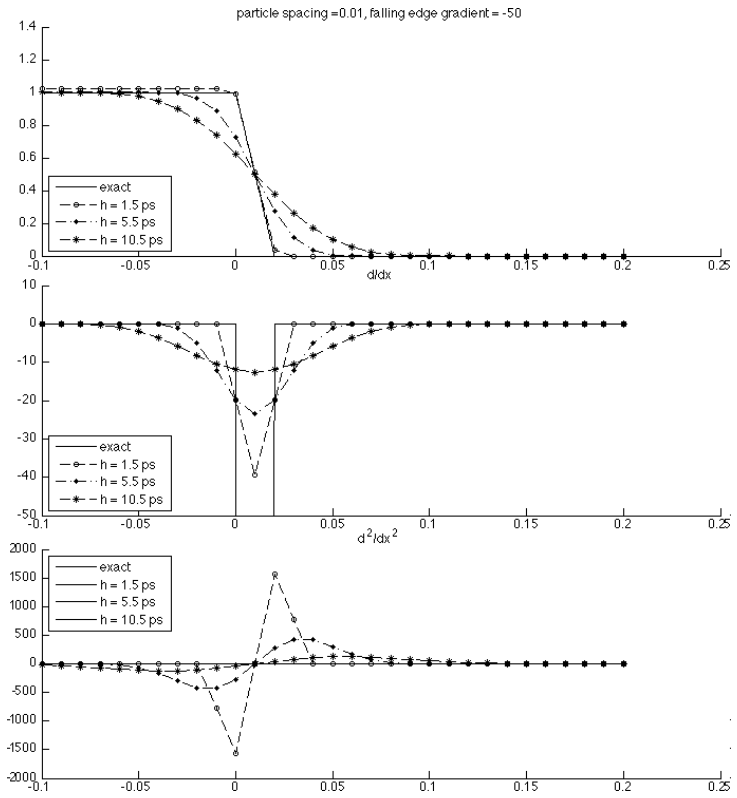


Figure 2.19: Approximation of a falling edge, high gradients

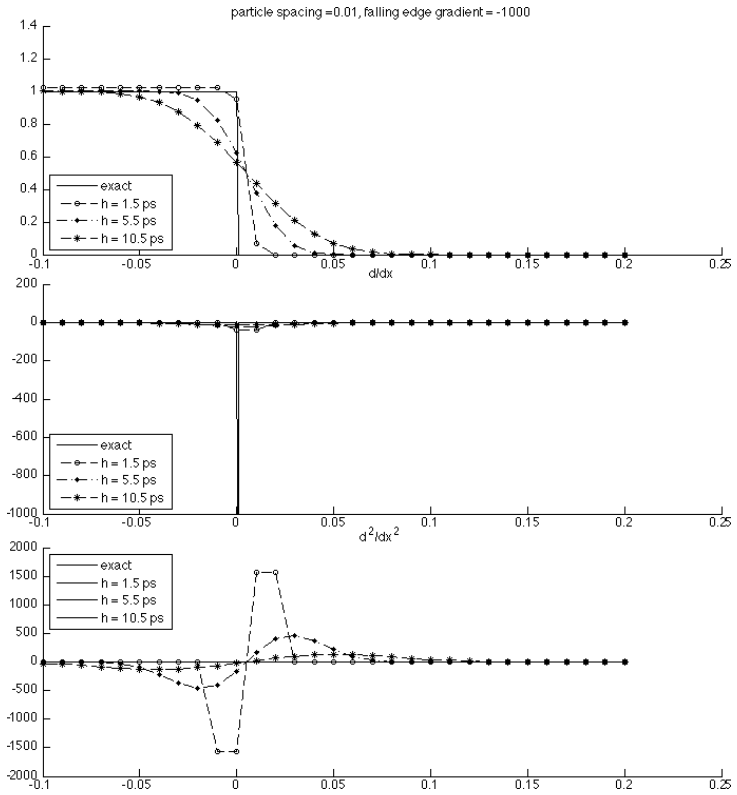


Figure 2.20: Approximation of a falling edge, discontinuous function

2.7.3 Choice of smoothing length

As has been demonstrated, the choice of smoothing length (relative to particle spacing) is key to the evaluation of a solution with SPH. If an incorrect choice is made, significant errors are introduced.

For a square wave, a good choice is a cover size of 1.5 times the particle spacing. This choice would be poor however, if some longer duration changes were to be approximated.

A large cover size removes the detail in an approximation, smoothing all edges. This is fine for slowly varying functions, but is unsuitable even for intermediate ones.

A cover size of 5.5 times the particle spacing appears to be a good compromise. Detail is not lost, and most approximations are fairly good. If a function is to change very rapidly two solutions are possible. Either the total number of particles must be increased, effectively reducing the function gradient in terms of particle spacing, or the function must be smoothed.

A certain systemic smoothing is appreciable in Figures 2.17 - 2.20, but this smoothing may not be enough to keep approximations close to the actual value. In this case, some other smoothing must be applied to the function in question.

2.8 Treatment of domain boundaries

As has been discussed in earlier parts of this chapter, boundary regions require special attention. Two different phenomena must be dealt with: this first is the overlap of particle cover functions with the edge of the domain. The second is the application of boundary conditions. A generic domain with boundary conditions is shown in Figure 2.21.

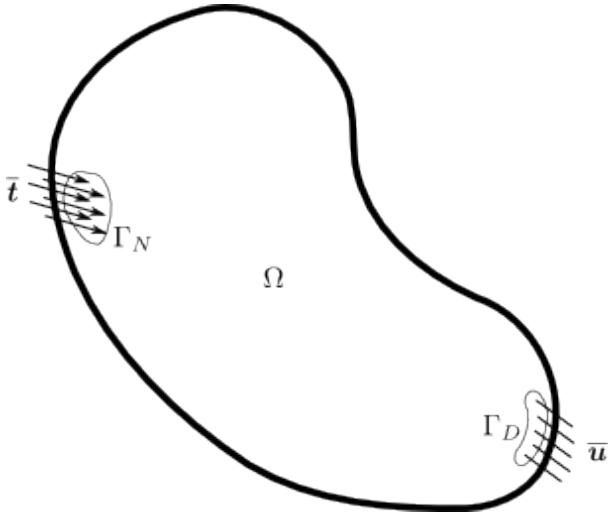


Figure 2.21: Boundary conditions

2.8.1 Boundary particles

A particle at (or near) the boundary of the domain may have a cover function that extends beyond the domain, as in Figure 2.22. In this case the normalisation condition of the kernel approximation is not met.

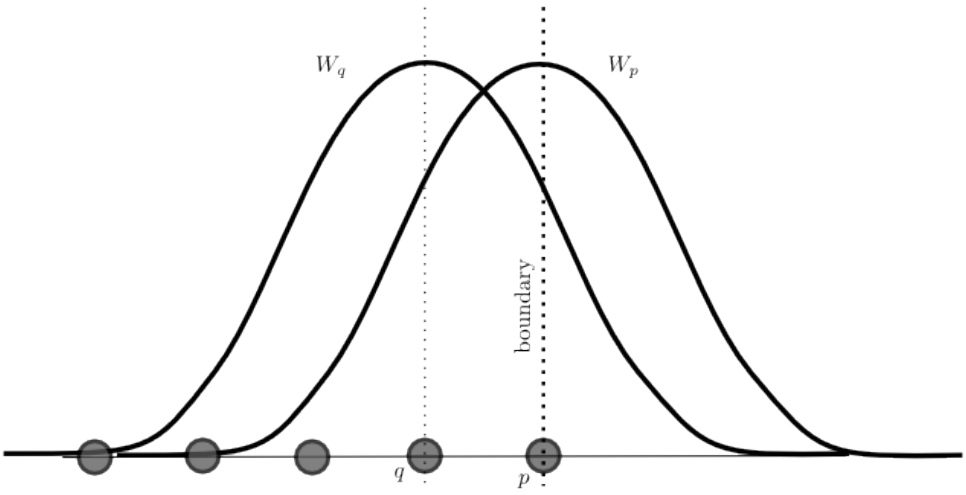


Figure 2.22: Normalisation violation of cover function at domain boundary

Three options exist to rectify the issue. The first is apparently simple, although in practice can be particularly difficult. This is to modify the cover function of the afflicted particles such that it does not extend beyond the boundary, as in Figure 2.23. While this appears simple, it is not as cover functions would have to be modified on a case-by-case basis. Further, this modification may be all but impossible on complex boundaries.

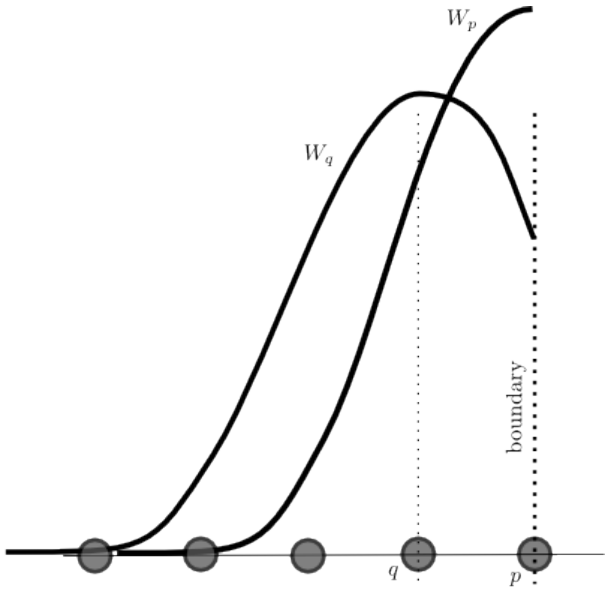


Figure 2.23: Modifying cover functions for normalisation

The second option is to include some sort of numerical rectification for the "under-integration". This involves calculating how much of the cover extends beyond the boundary and building in a numerical factor that adjusts the formulae appropriately. This can be quite computationally expensive, but has been employed in certain circumstances. The implementation of this technique indicated when a boundary is "surrounded" by domain, for instance the bar with a hole in it discussed in Section 5.4.4. The details of the implementation are discussed as the technique is required.

The third, and most simple option is to add particles beyond the boundary of the domain to ensure that those within the domain meet the normalisation condition, as depicted in Figure 2.24. This approach is called the *ghost particle* approach, and is the one used most commonly in the code developed.

A detailed study into the implementation of boundary values was published recently [19]. This paper presents several variations within those outlined above. This was not, however, available until after this project was near completion, and is mentioned simply for completeness.

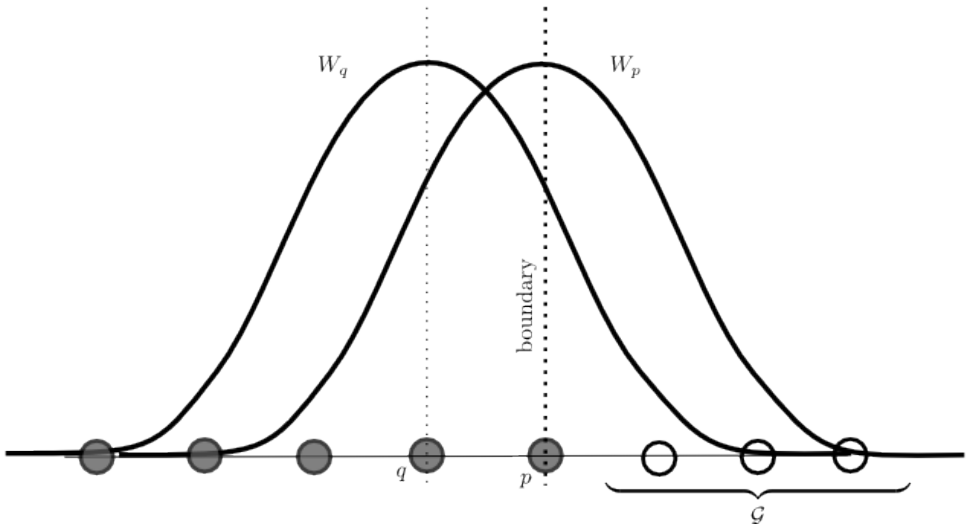


Figure 2.24: Addition of ghost particles ($\mathcal{G}$) for normalised cover functions

2.8.2 Boundary conditions

Two basic types of boundary conditions exist. The first is specification of the value of a property at the domain boundary. This is called a *Dirichlet* boundary condition and is fairly readily dealt with in SPH. The second is the specification of the normal derivative ($\partial u/\partial n = \nabla u \cdot \boldsymbol{n}$) of a function at the boundary. This is called a *Neumann* boundary condition, and is not as easily dealt with in SPH using existing methods.

2.8.2.1 Dirichlet boundary conditions

In order for the boundary condition to be satisfied all particles on the boundary must have the required value after solving the SPH equations. This is approached in two ways in the SPH literature.

The first is only possible when the value along a boundary is the same value along the entire boundary edge. Here the region of ghost particles can be set to be at the value required [8]. While the actual value of the boundary particle may not be exactly that required it will be reasonably close, because a large portion of the value is determined by the ghost particles.

The second is only possible if the SPH equations are only solved in an implicit or non-time-dependent manner. When this is done it is possible to use a Lagrange multiplier to the system of linear equations to enforce the boundary condition [8].

2.8.2.2 Neumann boundary conditions

Neumann boundary conditions can cause some difficulties in SPH because they are based on gradients of functions. A Neumann boundary condition can be described as one where

$$\nabla \boldsymbol{u} \cdot \hat{\boldsymbol{n}} \qquad \text{is set on} \qquad \Gamma_N \tag{2.36}$$

Neumann boundary conditions are seldom implemented in SPH simulations and where they are necessary they are often approximated through a carefully imposed Dirichlet condition [20].

Where a Neumann boundary condition exists one must be careful to not also attempt to impose a Dirichlet condition, as the problem then becomes over-constrained. If one attempts to set the Neumann boundary value by setting the values of ghost particles such that the gradient is correct one is effectively *also* setting a Dirichlet boundary condition. This results in direct use of ghost particles being incorrect.

The most direct approach is to modify the problem such that the Neumann boundary condition is treated as a Dirichlet one, by making the function of interest the gradient function (write gradient of $g(\boldsymbol{x})$ as $f(\boldsymbol{x}) = \nabla g(\boldsymbol{x})$). This is only possible where all boundary conditions are Neumann conditions. A problem consisting of only Neumann conditions is subject to rigid body motion which is not permissible.

Another option is to use ghost particles that do not have a set value, but are rather functions of the boundary particles they interact with. Applying this modified ghost particle approach sets the boundary gradient relative to the value of the boundary value prior to the application of SPH. Because the value of the boundary particle may have changed, the process should be iterative.

2.8.3 A corrective treatment of boundaries

A methodology for the treatment of both Dirichlet and Neumann boundary conditions was developed from basic SPH theory. This methodology assumes that any violation of the normalisation condition of the shape function at the boundary has been dealt with via one of the methodologies described in Section 2.8.1.

Consider the situation is Figure 2.25. Particles that lie on the boundary are marked $\mathcal{B}$. Particles which lie within the domain (internal particles) are marked $\mathcal{I}$. Additionally the set of ghost particles is denoted $\mathcal{G}$.

A change in the primary variable of a particle q will affect the SPH approximation of that variable at a particle p, provided q falls within the cover of p. This property can

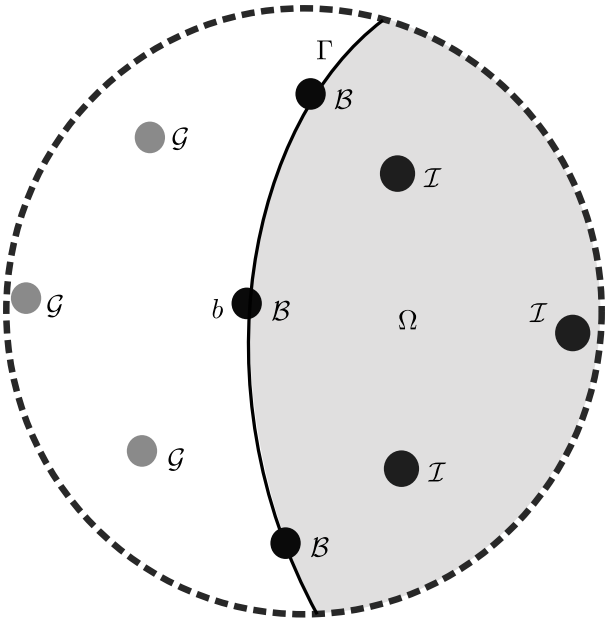


Figure 2.25: Identification of different types of particles

be exploited to force an SPH approximation to a particular value. Consider the case $< f > (\boldsymbol{x}) =< g\boldsymbol{x} + d >$ where d is some shift in $g(\boldsymbol{x})$ that is uniform throughout the domain. The value at b, the boundary particle at which we wish to enforce the boundary condition, can be chosen such that any value of $< f >$ can be enforced.

If we now restrict that change to be applied only to particles within the cover of a particular boundary particle b, and not to the rest of the domain or the particle itself we can write

$$< \mathfrak{f} > (\boldsymbol{x}_b) =< f(\boldsymbol{x}) + \Delta f > \tag{2.37}$$

where Δf is the change in $f(\boldsymbol{x})$. The symbol $\mathfrak{f}$ is used to denote the function because it is some alteration of the standard SPH approximation of f.

The function is fully defined by

$$< \mathfrak{f} > (\boldsymbol{x}_b) = \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} (f(\boldsymbol{x}_q) + \Delta f)\, W\left(\boldsymbol{x}_b - \boldsymbol{x}_q, h\right) \Delta V_q \tag{2.38}$$

where

$$\Delta f = \begin{cases} C & \text{if} \quad 0 < \|\boldsymbol{x}_b - \boldsymbol{x}_q\| < h \\ 0 & \text{otherwise} \end{cases}$$

and C is a constant determined to enforce the required change in the SPH approximation of f. Note that Δf is the same for all functions within the cover of b but does not affect the boundary particle itself. This property, that will be exploited later to enforce

a Neumann boundary condition, can be applied without changing the position of the boundary.

The problem with this approach is that a discontinuity in the variable field may occur at the limit of the cover of the boundary particles. To remove the danger of creating an artificial discontinuity the change required can be moderated by the smoothing function. This would allow a small change far from the boundary particle, and a more significant change closer to the particle. This can be done by modifying equation (2.38) such that

$$< \mathfrak{f} > (\boldsymbol{x}_b) = \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} (f(\boldsymbol{x}_q) + \Delta f)\, W(\boldsymbol{x}_b - \boldsymbol{x}_q, h)\, \Delta V_q \tag{2.39}$$

where

$$\Delta f = \begin{cases} \eta\, W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) & \text{if} \quad 0 < \|\boldsymbol{x}_b - \boldsymbol{x}_q\| < h \\ 0 & \text{otherwise} \end{cases}$$

where η is a corrective factor determined to enforce the boundary condition. If f is a vector, the corrective factor is written as $\boldsymbol{\eta}$.

Equation (2.39) can be rearranged such that η can be determined from $\mathfrak{f}$ and the SPH approximation prior to the application of the "correction". This can be expressed as

$$\begin{aligned} < \mathfrak{f} > (\boldsymbol{x}_b) &= < f > (\boldsymbol{x}_b) + \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \Delta f\; W(\boldsymbol{x}_b - \boldsymbol{x}_q, h)\, \Delta V_q \\ &= < f > (\boldsymbol{x}_b) + \eta \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left[W(\boldsymbol{x}_b - \boldsymbol{x}_q, h)\right]^2 \Delta V_q \end{aligned} \tag{2.40}$$

so that

$$\eta = \frac{< \mathfrak{f} > (\boldsymbol{x}_b) - < f > (\boldsymbol{x}_b)}{\sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left[W(\boldsymbol{x}_b - \boldsymbol{x}_q, h)\right]^2 \Delta V_q} \,. \tag{2.41}$$

If we use equation (2.41) to enforce a Dirichlet boundary condition such that

$$\overline{f}\,(\boldsymbol{x}_b) =< \mathfrak{f}\,(\boldsymbol{x}_b) >$$

where $\overline{f}$ is the boundary value we can write

$$\eta = \frac{\overline{f}\,(\boldsymbol{x}_b) - < f > (\boldsymbol{x}_b)}{\sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left[W(\boldsymbol{x}_b - \boldsymbol{x}_q, h)\right]^2 \Delta V_q} \,. \tag{2.42}$$

Care must be taken where f is a gradient function, as is demonstrated by setting $f = \nabla g$. Then

$$< \mathfrak{f} > (\boldsymbol{x}_{\mathcal{B}}) = < \nabla g > (\boldsymbol{x}_b) + \eta \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) \nabla W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) \Delta V_q \quad (2.43)$$

so that

$$\eta = \frac{< \mathfrak{f} > (\boldsymbol{x}_b) - < \nabla g > (\boldsymbol{x}_b)}{\sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) \nabla W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) \Delta V_q} \quad . \quad (2.44)$$

We can use this process to enforce a Neumann boundary condition. Suppose we wish to set

$$\frac{\partial g}{\partial \hat{n}} = \nabla g \cdot \hat{\boldsymbol{n}} = \overline{h}$$

at the boundary particle b. We can write equation (2.44) as

$$\eta = \frac{\overline{h}(\boldsymbol{x}_b) - < \nabla g \cdot \hat{\boldsymbol{n}} > (\boldsymbol{x}_b)}{\sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) \nabla W(\boldsymbol{x}_b - \boldsymbol{x}_q, h) \Delta V_q} \quad . \quad (2.45)$$

The process of finding η and applying it can be computationally expensive, especially when many boundary conditions exist. One has to compute SPH approximations twice for every boundary particle: once with the corrective term included, and once without. Despite this, the ability of this technique to reproduce boundary conditions exceeds that seen during testing of other methods.

2.9 Boundary conditions in higher dimensions

The treatment of boundary conditions in Section 2.8.3 does not always work in higher dimensions because it fails to account for the effect of ghost particles on more than one boundary particle. An alternative methodology, which is equivalent to the first method presented in Section 2.8.3 in one-dimension, is presented here. Note that while the example is one dimensional, the methodology works in higher dimensions as well.

Consider the SPH region shown in Figure 2.26 consisting of four particles, two of which lie on the boundary. We wish to find the SPH approximation of a function f subject to boundary conditions at the two boundary particles. The conventional ghost particle

methodology sets the ghost particles associated with each boundary particle to the value desired. The SPH approximation at the boundary particle will be closer to the desired boundary value had the ghost particles not been included, but may not be exact. Our task is to find what values the ghost particles must be set to such that the Dirichlet boundary conditions are applied exactly.

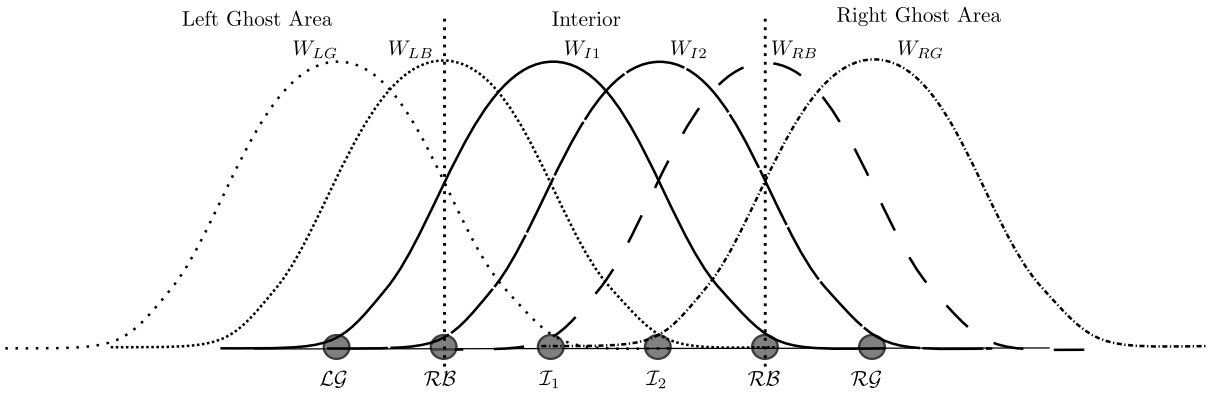


Figure 2.26: SPH discretisation - interaction of particles with ghost particles

The SPH approximation can be described by

$$\begin{Bmatrix} < f >_{GL} \\ < f >_{BL} \\ < f >_{I1} \\ < f >_{I2} \\ < f >_{BR} \\ < f >_{GR} \end{Bmatrix} = \begin{bmatrix} W_{GL-GL} & W_{GL-BL} & W_{GL-I1} & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & W_{BL-I1} & W_{BL-I2} & 0 & 0 \\ W_{I1-GL} & W_{I1-BL} & W_{I1-I1} & W_{I1-I2} & W_{I1-BR} & 0 \\ 0 & W_{I2-BL} & W_{I2-I1} & W_{I2-I2} & W_{I2-BR} & W_{I2-GR} \\ 0 & 0 & W_{BR-I1} & W_{BR-I2} & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & W_{GR-I2} & W_{GR-BR} & W_{GR-GR} \end{bmatrix} \begin{Bmatrix} f_{GL} \\ f_{BL} \\ f_{I1} \\ f_{I2} \\ f_{BR} \\ f_{GR} \end{Bmatrix}$$

where the subscripts denote the particles to which each entry relates. We allow the value of the boundary and ghost particles to be unknown prior to the approximation so we can find these values such that their value is exactly the boundary value after the approximation is made. This then allows the approximation to be made using the original internal particle values, and these newly computed boundary and ghost particle values. That is,

$$\begin{Bmatrix} \overline{f}_L \\ \overline{f}_L \\ < f >_{I1} \\ < f >_{I2} \\ \overline{f}_R \\ \overline{f}_R \end{Bmatrix} = \begin{bmatrix} W_{GL-GL} & W_{GL-BL} & W_{GL-I1} & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & W_{BL-I1} & W_{BL-I2} & 0 & 0 \\ W_{I1-GL} & W_{I1-BL} & W_{I1-I1} & W_{I1-I2} & W_{I1-BR} & 0 \\ 0 & W_{I2-BL} & W_{I2-I1} & W_{I2-I2} & W_{I2-BR} & W_{I2-GR} \\ 0 & 0 & W_{BR-I1} & W_{BR-I2} & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & W_{GR-I2} & W_{GR-BR} & W_{GR-GR} \end{bmatrix} \begin{Bmatrix} ?_{GL} \\ ?_{BL} \\ f_{I1} \\ f_{I2} \\ ?_{BR} \\ ?_{GR} \end{Bmatrix}$$

where the right hand side can be separated into knowns and unknowns as

$$\begin{Bmatrix} \overline{f}_L \\ \overline{f}_L \\ < f >_{I1} \\ < f >_{I2} \\ \overline{f}_R \\ \overline{f}_R \end{Bmatrix} = \begin{bmatrix} 0 & 0 & W_{GL-I1} & 0 & 0 & 0 \\ 0 & 0 & W_{BL-I1} & W_{BL-I2} & 0 & 0 \\ 0 & 0 & W_{I1-I1} & W_{I1-I2} & 0 & 0 \\ 0 & 0 & W_{I2-I1} & W_{I2-I2} & 0 & 0 \\ 0 & 0 & W_{BR-I1} & W_{BR-I2} & 0 & 0 \\ 0 & 0 & 0 & W_{GR-I2} & 0 & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ f_{I1} \\ f_{I2} \\ 0 \\ 0 \end{Bmatrix} + \dots$$

$$\begin{bmatrix} W_{GL-GL} & W_{GL-BL} & 0 & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & 0 & 0 & 0 & 0 \\ W_{I1-GL} & W_{I1-BL} & 0 & 0 & W_{I1-BR} & 0 \\ 0 & W_{I2-BL} & 0 & 0 & W_{I2-BR} & W_{I2-GR} \\ 0 & 0 & 0 & 0 & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & 0 & W_{GR-BR} & W_{GR-GR} \end{bmatrix} \begin{Bmatrix} ?_{GL} \\ ?_{BL} \\ 0 \\ 0 \\ ?_{BR} \\ ?_{GR} \end{Bmatrix} .$$

The term with the unknown boundary and ghost particle values can then be isolated as

$$\begin{Bmatrix} \overline{f}_L \\ \overline{f}_L \\ < f >_{I1} \\ < f >_{I2} \\ \overline{f}_R \\ \overline{f}_R \end{Bmatrix} - \begin{bmatrix} 0 & 0 & W_{GL-I1} & 0 & 0 & 0 \\ 0 & 0 & W_{BL-I1} & W_{BL-I2} & 0 & 0 \\ 0 & 0 & W_{I1-I1} & W_{I1-I2} & 0 & 0 \\ 0 & 0 & W_{I2-I1} & W_{I2-I2} & 0 & 0 \\ 0 & 0 & W_{BR-I1} & W_{BR-I2} & 0 & 0 \\ 0 & 0 & 0 & W_{GR-I2} & 0 & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ f_{I1} \\ f_{I2} \\ 0 \\ 0 \end{Bmatrix} =$$

$$\begin{bmatrix} W_{GL-GL} & W_{GL-BL} & 0 & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & 0 & 0 & 0 & 0 \\ W_{I1-GL} & W_{I1-BL} & 0 & 0 & W_{I1-BR} & 0 \\ 0 & W_{I2-BL} & 0 & 0 & W_{I2-BR} & W_{I2-GR} \\ 0 & 0 & 0 & 0 & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & 0 & W_{GR-BR} & W_{GR-GR} \end{bmatrix} \begin{Bmatrix} ?_{GL} \\ ?_{BL} \\ 0 \\ 0 \\ ?_{BR} \\ ?_{GR} \end{Bmatrix} .$$

This can be simplified to

$$\begin{Bmatrix} \overline{f}_L \\ \overline{f}_L \\ < f >_{I1} - W_{I1-I1} f_{I1} - W_{I1-I2} f_{I2} \\ < f >_{I2} - W_{I2-I1} f_{I1} - W_{I2-I2} f_{I2} \\ \overline{f}_R \\ \overline{f}_R \end{Bmatrix} =$$

$$\begin{bmatrix} W_{GL-GL} & W_{GL-BL} & 0 & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & 0 & 0 & 0 & 0 \\ W_{I1-GL} & W_{I1-BL} & 0 & 0 & W_{I1-BR} & 0 \\ 0 & W_{I2-BL} & 0 & 0 & W_{I2-BR} & W_{I2-GR} \\ 0 & 0 & 0 & 0 & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & 0 & W_{GR-BR} & W_{GR-GR} \end{bmatrix} \begin{Bmatrix} ?_{GL} \\ ?_{BL} \\ 0 \\ 0 \\ ?_{BR} \\ ?_{GR} \end{Bmatrix} .$$

This can then be modified using trivial equations such that the internal particle values

are retained in the approximation. This gives

$$\begin{Bmatrix} \overline{f}_L \\ \overline{f}_L \\ f_{I1} \\ f_{I2} \\ \overline{f}_R \\ \overline{f}_R \end{Bmatrix} = \begin{bmatrix} W_{GL-GL} & W_{GL-BL} & 0 & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & 0 & W_{GR-BR} & W_{GR-GR} \end{bmatrix} \begin{Bmatrix} ?_{GL} \\ ?_{BL} \\ f_{I1} \\ f_{I2} \\ ?_{BR} \\ ?_{GR} \end{Bmatrix} .$$

We can now solve for the column on the right giving the values required to ensure the boundary conditions are met in the approximation as

$$\begin{Bmatrix} ?_{GL} \\ ?_{BL} \\ f_{I1} \\ f_{I2} \\ ?_{BR} \\ ?_{GR} \end{Bmatrix} = \begin{bmatrix} W_{GL-GL} & W_{GL-BL} & 0 & 0 & 0 & 0 \\ W_{BL-GL} & W_{BL-BL} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{BR-BR} & W_{BR-GR} \\ 0 & 0 & 0 & 0 & W_{GR-BR} & W_{GR-GR} \end{bmatrix}^{-1} \begin{Bmatrix} \overline{f}_L \\ \overline{f}_L \\ f_{I1} \\ f_{I2} \\ \overline{f}_R \\ \overline{f}_R \end{Bmatrix} .$$

This approach can be generalised for any number of internal, boundary, and ghost particles as

$$\begin{Bmatrix} f_{\mathcal{I}} \\ f_{\mathcal{B}-\text{new}} \\ f_{\mathcal{G}-\text{new}} \end{Bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & W_{\mathcal{BB}} & W_{\mathcal{BG}} \\ 0 & W_{\mathcal{GB}} & W_{\mathcal{BB}} \end{bmatrix}^{-1}}_{\text{modified SPH matrix}} \begin{Bmatrix} f_{\mathcal{I}} \\ \overline{f}_{\mathcal{B}} \\ \overline{f}_{\mathcal{G}} \end{Bmatrix} . \tag{2.46}$$

The computation of the inverse of this modified "SPH matrix" can be expensive, but can be computed once for a simulation, and used repeatedly to compute the value of the ghost particles required to enforce the Dirichlet boundary condition once the SPH approximation is made by calculating

$$\begin{Bmatrix} < f >_{\mathcal{I}} \\ < f >_{\mathcal{B}} \\ < f >_{\mathcal{G}} \end{Bmatrix} = \underbrace{\begin{bmatrix} W_{\mathcal{II}} & W_{\mathcal{IB}} & W_{\mathcal{IG}} \\ W_{\mathcal{BI}} & W_{\mathcal{BB}} & W_{\mathcal{BG}} \\ W_{\mathcal{GI}} & W_{\mathcal{GB}} & W_{\mathcal{BB}} \end{bmatrix}}_{\text{SPH matrix}} \begin{Bmatrix} f_{\mathcal{I}} \\ f_{\mathcal{B}-\text{new}} \\ f_{\mathcal{G}-\text{new}} \end{Bmatrix} . \tag{2.47}$$

It is important to note that this technique would work to set the Dirichlet boundary conditions even if the ghost particles are omitted from the problem. The ghost particles are retained, however, as they are essential to ensure the accuracy of the approximation at near-boundary internal particles.

The disadvantage of this approach is that the technique does not work directly for gradients. The value of the first derivative of any cover function is zero over the particle itself. This leads to any modified "SPH gradient matrix" being not positive-definite, and thus not having a unique inverse. The gradient can only be set approximately, by ensuring that a secondary SPH approximation of the computed gradients satisfies the boundary condition.

This technique is the one pursued for all higher dimension SPH approximations in Chapter 5, and has, in general, given good results.

Chapter 3

Governing equations for elasticity

SPH has thus far been developed as a method of approximating functions. These approximations are generally used to obtain numerical solutions to partial differential equations (PDEs).

This study of *continuum mechanics* allows a mathematical description of macroscopic phenomena. When applied to solid bodies several things are required. The first is a description of a body in terms of time and space. This is the study of *kinematics*. The second is a description of forces acting within the body itself, which is described by *stress*. The third is a description of how properties of the body relate (in this case stress and strain), and is generally referred to as the *constitutive law*. The fourth describes the balance of momentum, and results in the *equation of motion*. This equation of motion forms the PDE to which a numerical solution is sought via SPH.

This chapter will look at each of these elements in turn. For each, the theory of *finite deformations* is examined first. This theory is very general, and makes a minimum number of assumptions. To simplify the mechanics for an initial implementation, an assumption of *infinitesimal deformations* is then made, and the theory linearised.

3.1 Kinematics

The description of a body in space, and its motion through time is known as kinematics. This study does not treat the cause of the motion, and focuses on the motion itself. The theory is developed for a continuous body.

3.1.1 Finite deformation

A brief overview of finite deformation is given. Many derivations are not complete, and only the theory applicable is developed.

Figure 3.1 shows the general motion of a deformable body.

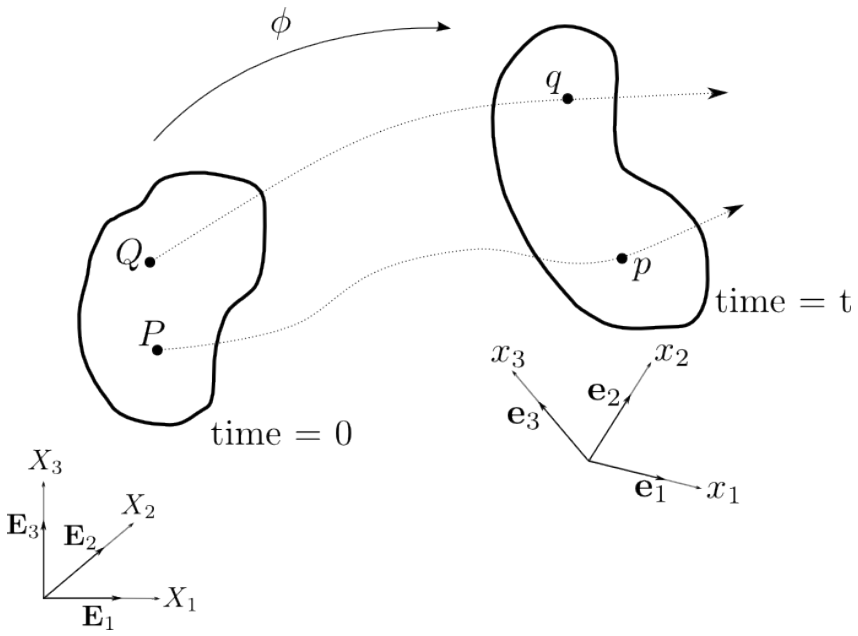


Figure 3.1: General motion of a deformable body

The original position of the body is described as the *reference* configuration. A general particle at this "snapshot" is denoted $\boldsymbol{X}$. At some later time, which we shall term the *current configuration*, all positions can be labelled $\boldsymbol{x}$. To allow us to separate the two configurations all indices relating to the current configuration are lower-case Roman letters (i.j,k), and those relating to the reference configuration are labelled in capital Roman letters (A,B,C). The unit direction vectors in the current configuration are denoted $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ and those in the reference configuration $\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3$.

The motion resulting in the current position is called ϕ and can be described by

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t) \ . \tag{3.1}$$

The current displacement is the difference in position between the current and reference configurations.

$$\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X} \ . \tag{3.2}$$

Much of the remaining continuum mechanics requires the use of the divergence and

gradient operators. These operators may be with respect to either the reference or the current configuration. To allow us to use these operators without constantly resorting to index notation the definitions described in Table 3.1 are used.

Table 3.1: Divergence and Gradient operators

w.r.t. $\boldsymbol{X}$	w.r.t. $\boldsymbol{x}$
$\text{Grad } \psi = \dfrac{\partial \psi}{\partial X_A}\boldsymbol{e}_A$	$\nabla \psi = \dfrac{\partial \psi}{\partial x_i}\boldsymbol{e}_i$
$\text{Grad } \boldsymbol{u} = \dfrac{\partial u_i}{\partial X_A}\boldsymbol{e}_i \otimes \boldsymbol{e}_A$	$\nabla \boldsymbol{u} = \dfrac{\partial u_i}{\partial x_j}\boldsymbol{e}_i \otimes \boldsymbol{e}_j$
$\text{Div } \boldsymbol{T} = \dfrac{\partial T_{iA}}{\partial X_A}\boldsymbol{e}_A$	$\text{div } \boldsymbol{T} = \dfrac{\partial T_{ij}}{\partial x_j}\boldsymbol{e}_i$
$\text{Div } \boldsymbol{u} = \dfrac{\partial u_A}{\partial X_A}$	$\text{div } \boldsymbol{u} = \dfrac{\partial u_i}{\partial x_i}$

We can now define $\boldsymbol{F}$, the *deformation gradient*, which is used to relate quantities in the current configuration to the reference configuration. It is defined by

$$\boldsymbol{F} = \text{Grad } \phi \qquad \text{or} \qquad F_{iA} = \frac{\partial \phi_i}{\partial X_A} \ . \tag{3.3}$$

Using equation (3.2) we can write equation (3.3) as

$$\boldsymbol{F} = \boldsymbol{I} + \text{Grad } \boldsymbol{u} \ . \tag{3.4}$$

The determinant of $\boldsymbol{F}$ is J, the *Jacobian*:

$$J = \det \boldsymbol{F} \ . \tag{3.5}$$

J can be related to the ratio of current to original (reference) volume at a particular point. Thus $J > 0$ and $\boldsymbol{F}$ is invertible.

We now define two new tensors which are used extensively in nonlinear mechanics. These are the *right Cauchy-Green tensor*, $\boldsymbol{C}$, and the *left Cauchy-Green tensor*, $\boldsymbol{B}$. These are

defined by

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} \ , \tag{3.6}$$

$$\boldsymbol{B} = \boldsymbol{F}\boldsymbol{F}^T \ . \tag{3.7}$$

If we look at these quantities in component form:, that is,

$$C_{AB} = F_{iA}F_{iB} \ , \tag{3.8}$$

$$B_{ij} = F_{iA}F_{jA} \ , \tag{3.9}$$

we see that $\boldsymbol{C}$ is a reference configuration quantity, while $\boldsymbol{B}$ is a current configuration quantity. It can be shown that both $\boldsymbol{B}$, and $\boldsymbol{C}$ are positive-definite and symmetric.

$\boldsymbol{F}$ can be decomposed into a rotational portion $\boldsymbol{R}$ where $\boldsymbol{R}$ is an orthogonal rotation tensor (resulting in $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$), and a positive-definite symmetric stretch portion $\boldsymbol{U}$, they can be related by

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U} \ . \tag{3.10}$$

The right Cauchy-Green tensor can be shown to be independent of rotation:

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{F}^T\boldsymbol{F} && (3.6)\\ &= (\boldsymbol{R}\boldsymbol{U})^T(\boldsymbol{R}\boldsymbol{U}) \\ &= \boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U} \\ &= \boldsymbol{U}^T\boldsymbol{U} \\ &= \boldsymbol{U}\boldsymbol{U} \ . && (3.11) \end{aligned}$$

With the Cauchy-Green tensors we can now introduce two of the most common nonlinear strain measures. These are the *Green-Lagrange strain*, $\boldsymbol{E}$ and the *Euler-Almansi strain*, $\boldsymbol{e}$. $\boldsymbol{E}$ is a reference configuration measure of strain, while $\boldsymbol{e}$ is a current configuration measure of strain. They are defined by

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right) \ , \tag{3.12}$$

$$\boldsymbol{e} = \frac{1}{2}\left(\boldsymbol{I} - \boldsymbol{B}^{-1}\right) \ . \tag{3.13}$$

3.1.2 Infinitesimal deformation

Finite deformation can be used to describe any motion of a body. This motion can be very difficult to analyse. If one introduces an assumption that deformations are "small", the theory described above can be greatly simplified. This assumption is that of *infinitesimal deformation*. While there are still differences between the reference and current configurations this difference is small enough that it need not be made for the mathematical functions described in Table 3.1.

Consider two points in a body that undergo some displacement: point P undergoes a deformation:

$$\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u}\left(\boldsymbol{X}, t\right) \tag{3.14}$$

and point Q adjacent to it (at position $\boldsymbol{X} + \mathrm{d}\boldsymbol{X}$ is moved to a point $\boldsymbol{x} + \mathrm{d}\boldsymbol{x}$ described by

$$\boldsymbol{x} + \mathrm{d}\boldsymbol{x} = \boldsymbol{X} + \mathrm{d}\boldsymbol{X} + \boldsymbol{u}\left(\boldsymbol{X} + \mathrm{d}\boldsymbol{X}, t\right) \ . \tag{3.15}$$

Subtracting equation (3.14) from equation (3.15) we get

$$\mathrm{d}\boldsymbol{x} = \mathrm{d}\boldsymbol{X} + \boldsymbol{u}\left(\boldsymbol{X} + \mathrm{d}\boldsymbol{X}, t\right) - \boldsymbol{u}\left(\boldsymbol{X}, t\right) \ . \tag{3.16}$$

Using the definition of the gradient of a vector function from table 3.1 we can write:

$$\mathrm{d}\boldsymbol{x} = \mathrm{d}\boldsymbol{X} + \left(\nabla \boldsymbol{u}\right) \mathrm{d}\boldsymbol{X} \ . \tag{3.17}$$

If we consider the right Cauchy-Green tensor in terms of the definition of $\boldsymbol{F}$ described in equation (3.4), then

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{F}^T \boldsymbol{F} && (3.6) \\ &= \left(\boldsymbol{I} + \nabla \boldsymbol{u}\right)^T \left(\boldsymbol{I} + \nabla \boldsymbol{u}\right) \\ &= \boldsymbol{I} + \nabla \boldsymbol{u} + \left(\nabla \boldsymbol{u}\right)^T + \left(\nabla \boldsymbol{u}\right)^T \nabla \boldsymbol{u} \ . && (3.18) \end{aligned}$$

The term $\left(\nabla \boldsymbol{u}\right)^T \nabla \boldsymbol{u}$ can be considered to be negligibly small. We can then write

$$\boldsymbol{C} \approx \boldsymbol{I} + \nabla \boldsymbol{u} + \left(\nabla \boldsymbol{u}\right)^T \ . \tag{3.19}$$

If these definitions are used in the definition of the Green-Lagrange strain, then

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right) \tag{3.12}$$

$$= \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right) + (\nabla \boldsymbol{u})^T \nabla \boldsymbol{u} \tag{3.20}$$

The infinitesimal strain $\boldsymbol{\varepsilon}$ can be defined using approximation (3.19), that is

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right) \ . \tag{3.21}$$

3.1.3 Time derivatives

The rate at which some property changes over time can be described by some time derivative. Care must be taken as to which configuration the derivative is performed in, as the various configurations are functions of time themselves.

Because the reference configuration is taken to be at $t = 0$ we define the time derivative with regards to some function $\psi(\boldsymbol{X}, t)$ as

$$\frac{\partial \psi}{\partial t} = \left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{X}} \ . \tag{3.22}$$

To differentiate a time derivative based in the current configuration from that in the reference configuration we introduce the *material time derivative* defined as

$$\left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{X}} = \left.\frac{\partial}{\partial t}\psi\left(\boldsymbol{x}, t\right)\right|_{\boldsymbol{X}} = \frac{\mathrm{D}\psi}{\mathrm{D}t} \ . \tag{3.23}$$

Using the definition of a derivative we get

$$\frac{\mathrm{D}\psi}{\mathrm{D}t} = \lim_{\Delta t \to 0} \frac{\psi\left(\boldsymbol{X}, t + \Delta t\right) - \psi\left(\boldsymbol{X}, t\right)}{\Delta t}$$

$$= \left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{x}} + \frac{\partial \psi}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t} \qquad \text{(using the chain rule)}$$

$$= \left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{x}} + \nabla \psi \underbrace{\frac{\partial \boldsymbol{x}}{\partial t}}_{\boldsymbol{v}} \ . \tag{3.24}$$

The term marked with $\boldsymbol{v}$ is the velocity.

For a vector quantity $\boldsymbol{w}$, the material derivative is given by

$$\begin{aligned}\frac{\mathrm{D}\boldsymbol{w}}{\mathrm{D}t} &= \left.\frac{\partial \boldsymbol{w}}{\partial t}\right|_{\boldsymbol{x}} + \left(v_i \frac{\partial}{\partial x_i}\right) w_j \\ &= \left.\frac{\partial \boldsymbol{w}}{\partial t}\right|_{\boldsymbol{x}} + (\boldsymbol{v}\cdot\nabla)\,\boldsymbol{w}\ . \end{aligned} \tag{3.25}$$

When $\boldsymbol{w}$ is set to $\boldsymbol{v}$ the expression above gives the acceleration.

3.2 Stress

A body cannot be fully described in terms of kinematics alone. The body may have forces that act upon it, these forces affect the motion of the body, and thus the kinematics thereof. The forces acting on the body can be divided into two categories. The first is that of body forces ($\boldsymbol{b}$), these forces act on the *entire* body such as gravity. The second are forces that act on some surface of the body. These forces are denoted $\boldsymbol{t}$, and are called surface traction forces.

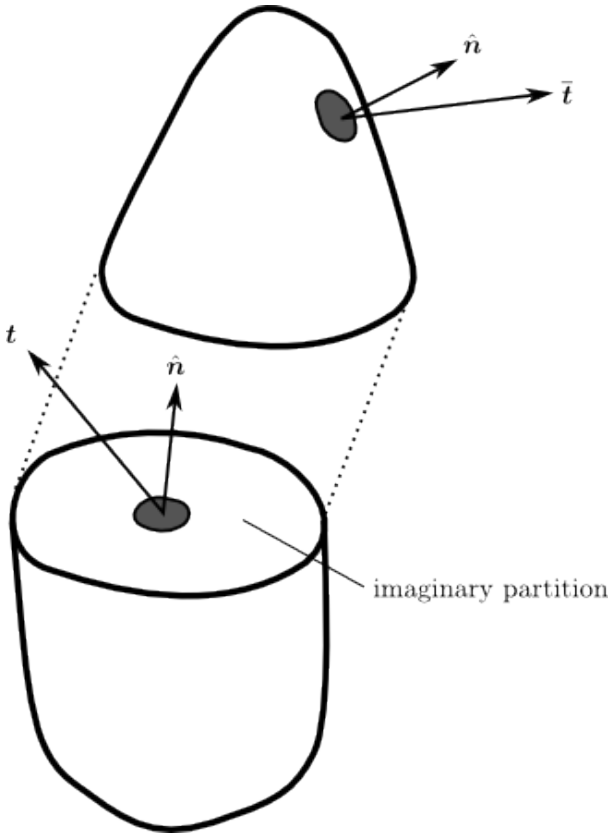


Figure 3.2: Traction on a surface

This is defined by Cauchy's Law, where $\boldsymbol{t}$ is given by

$$\boldsymbol{t} = \boldsymbol{T}\hat{\boldsymbol{n}} \quad \text{or} \tag{3.26}$$

$$t_i = T_{ij} n_j \tag{3.27}$$

where $\boldsymbol{T}$ is the *Cauchy stress tensor* and $\hat{\boldsymbol{n}}$ is a unit vector normal to the surface. These vectors are shown in Figure 3.2.

3.2.1 Stress and finite deformation

Because the surface upon which the traction force acts moves as a function of time, it is important to take care that an appropriate description of the traction force and the surface is used.

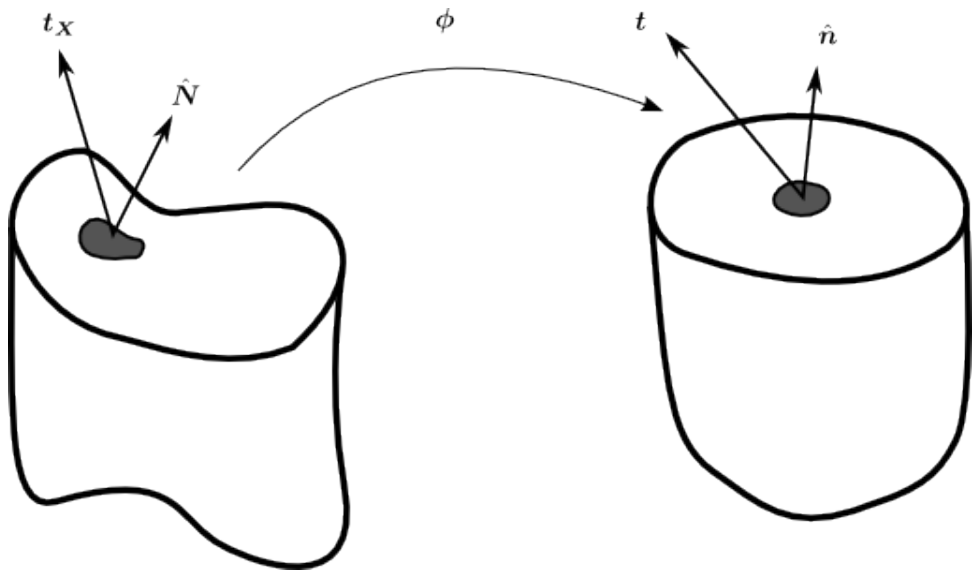


Figure 3.3: Traction in finite deformation

This vector $\hat{\boldsymbol{n}}$ can itself be represented in the reference configuration as $\hat{\boldsymbol{N}}$. Both vectors are shown in Figure 3.3. An area element can be described in both the reference and current configurations:

$$\mathrm{d}\boldsymbol{A} = \hat{\boldsymbol{N}}\,\mathrm{d}A \qquad \text{in the reference configuration} \tag{3.28}$$

becomes

$$\mathrm{d}\boldsymbol{a} = \hat{\boldsymbol{n}}\,\mathrm{d}a \qquad \text{in the current configuration} \tag{3.29}$$

These are related by *Nanson's formula*, defined as

$$\vec{\boldsymbol{n}}\,\mathrm{d}a = J\boldsymbol{F}^{-T}\hat{\boldsymbol{N}}\,\mathrm{d}A\ . \tag{3.30}$$

Two other stress measures are generally used in nonlinear mechanics. They are the first and second *Piola-Kirchhoff stresses*, $\boldsymbol{P}$ and $\boldsymbol{S}$ respectively.

$\boldsymbol{P}$ is defined as

$$\boldsymbol{P}\hat{\boldsymbol{N}}\,\mathrm{d}A = \boldsymbol{T}\hat{\boldsymbol{n}}\,\mathrm{d}a\ . \tag{3.31}$$

Using Nanson's formula $\boldsymbol{P}$ can be related to $\boldsymbol{T}$ by

$$\boldsymbol{P} = J\boldsymbol{T}\boldsymbol{F}^{-T} \tag{3.32}$$
$$\boldsymbol{T} = J^{-1}\boldsymbol{P}\boldsymbol{F}^{T} \tag{3.33}$$
$$T_{ij} = J^{-1}P_{iA}F_{jA}\ . \tag{3.34}$$

The first Piola-Kirchhoff stress can be seen to be defined relative to both the current *and* reference configurations.

$\boldsymbol{S}$ is defined by

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{T}\boldsymbol{F}^{-T} \tag{3.35}$$
$$\boldsymbol{T} = J^{-1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^{T} \tag{3.36}$$
$$T_{ij} = J^{-1}F_{iA}S_{AB}F_{jB}\ . \tag{3.37}$$

The second Piola-Kirchhoff stress is defined in terms of the reference configuration *only*.

Note that $\boldsymbol{P}$ can be directly related to $\boldsymbol{S}$ via the deformation gradient by comparing equations (3.32) and (3.35):

$$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S} \tag{3.38}$$

3.2.2 Stress and infinitesimal deformation

Traditionally stress is described in the reference configuration for small strains. Thus Cauchy stress $\boldsymbol{T}$ is most commonly used to describe small strain problems. It is important

to note that under the small strain approximation

$$\begin{aligned} \boldsymbol{F} &\approx \boldsymbol{I} \ , \\ J = \det\ \boldsymbol{F} &\approx 1 \ . \end{aligned} \tag{3.39}$$

Then using the definition of $\boldsymbol{P}$,

$$\boldsymbol{P} = J\boldsymbol{T}\boldsymbol{F}^{-T} \tag{3.32}$$

$$\approx \boldsymbol{T}\boldsymbol{I}$$

$$= \boldsymbol{T} \tag{3.40}$$

and using the definition of $\boldsymbol{S}$,

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{T}\boldsymbol{F}^{-T} \tag{3.35}$$

$$\approx \boldsymbol{I}\boldsymbol{T}\boldsymbol{F}$$

$$= \boldsymbol{T} \ . \tag{3.41}$$

3.3 Constitutive laws

In the previous section stress was motivated as a method for describing forces that vary through the body. The distribution of strain within the body is function of the material properties of the body. There are a great many material models to choose from, and typically one seeks a model that captures the key phenomenological behaviour with a minimum of complexity.

If a material's constitutive law is a function of the current state of deformation only (path and time independent) the material is termed *elastic*. Any deformation will result in energy being stored in the body. If we assume that some stored strain energy function, $\boldsymbol{\Psi}$, could be determined from material tests this function would be in terms of the deformation gradient. This can be written: as

$$\boldsymbol{\Psi} = \boldsymbol{\Psi}\left(\boldsymbol{F}(\boldsymbol{X}, t), \boldsymbol{X}\right) \ . \tag{3.42}$$

It can be shown [21, page 118] that the first Piola-Kirchhoff stress can be related to the strain energy function by:

$$\boldsymbol{P} = \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}} \ . \tag{3.43}$$

A body that undergoes a rigid body motion will not store energy, and thus Ψ is not related to any rotational part of $\boldsymbol{F}$. If we invoke the fact that $\boldsymbol{C}$ is free of rotation we can note that:

$$\boldsymbol{\Psi}(\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{X}) = \boldsymbol{\Psi}(\boldsymbol{C}(\boldsymbol{X}), \boldsymbol{X}) \ . \tag{3.44}$$

Noting that $\boldsymbol{S} = \boldsymbol{F}\boldsymbol{P}$,

$$\boldsymbol{S} = 2\frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{C}} \ . \tag{3.45}$$

Invoking the definition of the Green-Lagrange strain

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right) \ , \tag{3.12}$$

we find that

$$\boldsymbol{S} = \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{E}} \ . \tag{3.46}$$

The relationship in equation (3.45) is generally nonlinear. We can linearise the relationship by looking at the directional derivative of $\boldsymbol{S}$. That is,

$$\begin{aligned} \mathrm{D}\boldsymbol{S}(\boldsymbol{u}) &= \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \boldsymbol{S}(\boldsymbol{E}(\boldsymbol{\phi} + \epsilon\boldsymbol{u})) \\ &= \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}} \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \boldsymbol{E}(\boldsymbol{\phi} + \epsilon\boldsymbol{u}) \qquad \text{(using the chain rule)} \\ &= \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}} \, \mathrm{D}\boldsymbol{E}(\boldsymbol{u}) \ . \end{aligned} \tag{3.47}$$

Stress ($\boldsymbol{S}$) and strain ($\boldsymbol{E}$) are both second order tensors. They can now be related to each other via a fourth order tensor. This tensor, $\boldsymbol{\mathcal{C}}$, is called the *Lagrangian elasticity tensor* and allows equation (3.47) to be stated as

$$\mathrm{D}\boldsymbol{S}(\boldsymbol{u}) = \boldsymbol{\mathcal{C}} : \mathrm{D}\boldsymbol{E}(\boldsymbol{u}) \qquad \text{where} \qquad \boldsymbol{\mathcal{C}} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}} \tag{3.48}$$

It can then be written:

$$\boldsymbol{\mathcal{C}} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}}$$

$$= 2\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{C}} \qquad \text{using equation (3.12)}$$

$$= 4\frac{\partial^2 \boldsymbol{\Psi}}{\partial \boldsymbol{C} \partial \boldsymbol{C}} \qquad \text{using equation (3.45) .} \tag{3.49}$$

3.3.1 St. Venant-Kirchhoff material model

The St.Venant-Kirchhoff material model has the strain energy function

$$\boldsymbol{\Psi}(\boldsymbol{E}) = \frac{1}{2}\lambda \left(tr\boldsymbol{E}\right)^2 + \mu \boldsymbol{E} \colon \boldsymbol{E} \tag{3.50}$$

where λ and μ are the Lamé constants.

The second Piola-Kirchhoff stress can be found from $\boldsymbol{\Psi}$ using equation (3.46), to be

$$\boldsymbol{S} = \lambda (tr\boldsymbol{E})\boldsymbol{I} + 2\mu \boldsymbol{E} \tag{3.51}$$

or, using equation (3.38), to be

$$\boldsymbol{P} = \lambda (tr\boldsymbol{E})\boldsymbol{F} + 2\mu \boldsymbol{F}\boldsymbol{E} \ . \tag{3.52}$$

Equation (3.51) can be directly related to Hooke's law for small strains. This comparison follows.

3.3.2 Linear elastic material model - Hooke's law

Linear elasticity is the simplest of all constitutive models. The theory is well developed for small strain problems where a change in strain can be linearly described. Using the earlier notation of $\boldsymbol{\mathcal{C}}$ to represent the elasticity tensor this can be described as

$$\boldsymbol{T} = \boldsymbol{\mathcal{C}}\boldsymbol{\varepsilon} \ . \tag{3.53}$$

Note that although by the earlier definition $\boldsymbol{\mathcal{C}}$ is the *Lagrangian* elasticity tensor, here there is no need for that specification, as there are no differences between $\boldsymbol{T}$, $\boldsymbol{P}$ and $\boldsymbol{S}$ as described in section 3.2.2 on page 49.

The linear relationship between stress and strain is commonly stated in term of the Lamé constants, λ and μ as

$$\boldsymbol{T} = \lambda(tr\,\boldsymbol{\varepsilon})\boldsymbol{I} + 2\mu\,\boldsymbol{\varepsilon}\ . \tag{3.54}$$

This is identical to equation (3.51) if this equation is stated under the small strain assumptions.

3.3.3 Isotropic hyperelasticity - Neo-Hookean material model

To develop meaningful finite strain material model without excessive complexity the assumption of material isotropy is made. This assumes that the body's behaviour is independent of the orientation of the constituent material. We denote this by writing

$$\boldsymbol{\Psi}(\boldsymbol{C}(\boldsymbol{X}), \boldsymbol{X}) = \boldsymbol{\Psi}(I_{\boldsymbol{C}}, II_{\boldsymbol{C}}, III_{\boldsymbol{C}}, \boldsymbol{X}) \tag{3.55}$$

where $I_{\boldsymbol{C}}$,$II_{\boldsymbol{C}}$ and $III_{\boldsymbol{C}}$ are the invariants of $\boldsymbol{C}$. These are defined as

$$I_{\boldsymbol{C}} = \text{tr}\ \boldsymbol{C} = \boldsymbol{C} : \boldsymbol{I}\ , \tag{3.56}$$

$$II_{\boldsymbol{C}} = \text{tr}\ \boldsymbol{C}\boldsymbol{C} = \boldsymbol{C} : \boldsymbol{C}\ , \tag{3.57}$$

$$III_{\boldsymbol{C}} = \det\ \boldsymbol{C} = J^2\ . \tag{3.58}$$

Applying the chain rule to equation (3.45)

$$\boldsymbol{S} = 2\frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{C}} \tag{3.45}$$

$$= 2\frac{\partial \boldsymbol{\Psi}}{\partial I_{\boldsymbol{C}}}\frac{\partial I_{\boldsymbol{C}}}{\partial \boldsymbol{C}} + 2\frac{\partial \boldsymbol{\Psi}}{\partial II_{\boldsymbol{C}}}\frac{\partial II_{\boldsymbol{C}}}{\partial \boldsymbol{C}} + 2\frac{\partial \boldsymbol{\Psi}}{\partial III_{\boldsymbol{C}}}\frac{\partial III_{\boldsymbol{C}}}{\partial \boldsymbol{C}}\ . \tag{3.59}$$

The derivatives of the invariants are [22]

$$\frac{\partial I_{\boldsymbol{C}}}{\partial \boldsymbol{C}} = \boldsymbol{I}, \quad \frac{\partial II_{\boldsymbol{C}}}{\partial \boldsymbol{C}} = 2\boldsymbol{C}, \quad \frac{\partial III_{\boldsymbol{C}}}{\partial \boldsymbol{C}} = J^2\boldsymbol{C}^{-1}\ . \tag{3.60}$$

Equation (3.59) can now be written as

$$\boldsymbol{S} = 2\frac{\partial \boldsymbol{\Psi}}{\partial I_{\boldsymbol{C}}}\boldsymbol{I} + 4\frac{\partial \boldsymbol{\Psi}}{\partial II_{\boldsymbol{C}}}\boldsymbol{C} + 2J^2\frac{\partial \boldsymbol{\Psi}}{\partial III_{\boldsymbol{C}}}\boldsymbol{C}^{-1}\ . \tag{3.61}$$

The compressible Neo-Hookean model is an adaptation of a material model used to model

incompressible rubber-like materials [23]. The adaptation allows small volumetric changes in a material during loading which was not possible in the original material model [24]. The strain energy function for the material is given by

$$\boldsymbol{\Psi} = \frac{\mu}{2}(I_{\boldsymbol{C}} - 3) - \mu\, lnJ + \frac{\lambda}{2}(\ln J)^2 \ . \tag{3.62}$$

Applying equation (3.61) to equation (3.62),

$$\boldsymbol{S} = \mu(\boldsymbol{I} - \boldsymbol{C}^{-1}) + \lambda(\ln J)\boldsymbol{C}^{-1} \ . \tag{3.63}$$

The implementation presented in Chapter 5 utilises this material model and it is recast in terms of $\boldsymbol{P}$. The constitutive Neo-Hookean law can be written in terms of $\boldsymbol{P}$ as

$$\boldsymbol{P} = \mu\left(\boldsymbol{F} - \boldsymbol{F}\boldsymbol{C}^{-1}\right) + \lambda\,(\ln J)\,\boldsymbol{F}\boldsymbol{C}^{-1} \ . \tag{3.64}$$

By pre-multiplying through (3.63) by $\boldsymbol{C}$ we can cast this statement into one that is similar to Hooke's law (equation (3.54))

$$\boldsymbol{C}\boldsymbol{S} = \boldsymbol{C}\mu(\boldsymbol{I} - \boldsymbol{C}^{-1}) + \lambda(lnJ)\boldsymbol{C}\boldsymbol{C}^{-1} \tag{3.65}$$

$$\boldsymbol{F}^T\boldsymbol{F}\boldsymbol{S} = \lambda(\ln J)\boldsymbol{I} + \mu(\boldsymbol{C} - \boldsymbol{I}) \tag{3.66}$$

$$\boldsymbol{F}^T\boldsymbol{P} = \lambda(\ln J)\boldsymbol{I} + +\mu\boldsymbol{E} \ . \tag{3.67}$$

This material model has proved to be useful in the finite strain domain [25]. In small strain cases it can be shown that a Neo-Hookean material reduces to a linear elastic one.

3.4 The equation of motion

An equation of motion links the motion of a body to the forces acting on that body. Once again this link can be described in either the reference or current configuration.

Consider a balance of linear momentum on a body. That is, the rate of change of momentum of any part of the body must equal the total force acting on it:

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\int_V \boldsymbol{v}\rho\,\mathrm{d}V}_{\text{part 1}} = \underbrace{\int_V \rho\,\boldsymbol{b}\,\mathrm{d}V}_{\text{part 2}} + \underbrace{\int_{\partial V} \boldsymbol{t}\,\mathrm{d}a}_{\text{part 3}} \tag{3.68}$$

where ρ is the density of the material.

How each of the three terms above are dealt with determines in which configuration the equation of motion is stated in.

3.4.1 Equation of motion in the reference configuration

For a description in the reference configuration we require the integrals to be in terms of volumes in the reference configuration.

The third part of this equation can be cast in terms of $\boldsymbol{P}$ using Nanson's formula (3.30):

$$\begin{aligned}\int_{\partial V} \boldsymbol{t}\,\mathrm{d}a &= \int_{\partial V} \boldsymbol{T}\vec{\boldsymbol{n}}\,\mathrm{d}a \\ &= \int_{\partial V_0} \boldsymbol{T}\left(J\boldsymbol{F}^{-T}\vec{\boldsymbol{N}}\right)\,\mathrm{d}A \\ &= \int_{\partial V_0} \boldsymbol{P}\vec{\boldsymbol{N}}\,\mathrm{d}A \\ &= \int_{V_0} \mathrm{Div}\quad \boldsymbol{P}\,\mathrm{d}V_0\ . \end{aligned} \tag{3.69}$$

The second part can similarly be cast in terms of the reference configuration as

$$\begin{aligned}\int_V \rho\boldsymbol{b}\,\mathrm{d}V &= \int_{V_0} \rho\,\boldsymbol{b}J\,\mathrm{d}V_0 \\ &= \int_{V_0} \rho_0\boldsymbol{b}\,\mathrm{d}V_0\ . \end{aligned} \tag{3.70}$$

Finally, the first part can be written

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \boldsymbol{v}\rho\,\mathrm{d}V = \int_{V_0} \rho_0\frac{\partial \boldsymbol{v}}{\partial t}\,\mathrm{d}V_0 \qquad \text{changing to the reference configuration .} \tag{3.71}$$

These integrals over the reference volume can be collected to give

$$\int_{V_0}\left[\rho_0\frac{\partial \boldsymbol{v}}{\partial t} - \mathrm{Div}\quad \boldsymbol{P} - \rho_0\boldsymbol{b}\right]\,\mathrm{d}V_0 = 0\ . \tag{3.72}$$

Since V and V_0 are arbitrary, the integrand must be zero: this gives

$$\rho_0 \frac{\partial \boldsymbol{v}}{\partial t} - \text{Div} \quad \boldsymbol{P} - \rho_0 \boldsymbol{b} = 0 \ . \tag{3.73}$$

3.4.2 Equation of motion in the current configuration

Returning to equation (3.68), we can transform the second term on the right-hand side into one over the entire volume:

$$\begin{aligned} \int_{\partial V} \boldsymbol{t} \, \mathrm{d}a &= \int_{\partial V} \boldsymbol{T}\vec{\boldsymbol{n}} \, \mathrm{d}a \\ &= \int_V \text{div} \ \boldsymbol{T} \, \mathrm{d}V \ . \end{aligned} \tag{3.74}$$

Additionally, it can be shown that

$$\frac{\mathrm{D}}{\mathrm{D}t} \int_V \boldsymbol{v} \rho \, \mathrm{d}V = \int_{V_0} \dot{\boldsymbol{v}} \rho \, \mathrm{d}V \ . \tag{3.75}$$

Equation (3.68) can now be written as

$$\frac{\mathrm{D}}{\mathrm{D}t} \int_V \boldsymbol{v} \rho \, \mathrm{d}V = \int_V \rho \, \boldsymbol{b} \, \mathrm{d}V + \int_{\partial V} \boldsymbol{T}\vec{\boldsymbol{n}} \, \mathrm{d}a \tag{3.76}$$

$$\int_V \rho \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} \, \mathrm{d}V = \int_V \rho \, \boldsymbol{b} \, \mathrm{d}V + \int_V \text{div} \ \boldsymbol{T} \, \mathrm{d}V \ . \tag{3.77}$$

Collecting the integrals over the current volume, we get

$$\int_V \left[\rho \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} - div \ \boldsymbol{T} - \rho \, \boldsymbol{b} \right] \mathrm{d}V = 0 \ . \tag{3.78}$$

Again, since V is arbitrary, we get

$$\rho \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} - div \ \boldsymbol{T} - \rho \, \boldsymbol{b} = 0 \ . \tag{3.79}$$

Chapter 4

SPH in linear elasticity

Due to its uncomplicated nature, a linear elastic code is first implemented. Linear elasticity is also relatively simple to solve for uncomplicated geometries. Thus, an SPH implementation could be directly validated with closed form solutions.

The literature records that time-dependent problems are readily solved with SPH [26], thus the problem to be modelled was extended to time-dependent linear elasticity, or elastodynamics. An SPH approximation of the theory governing linear elasticity discussed in Chapter 3 is presented. This corresponding algorithm is developed for use in an SPH implementation.

4.1 Linear elastodynamics

SPH approximations have been presented for functions of position. When looking at a time-dependent problem SPH cannot be used to solve an ordinary differential equation in time. Thus some other method is used to perform the time integration. This then reduces the remaining equations to be solved to functions of position.

To facilitate the approximation of the theory a summary is provided in Table 4.1.

The equation of motion (point 3 in Table 4.1), can only be solved if it is presented as an initial-boundary value problem. These additional conditions are described in Table 4.2.

The constraints added onto the equation of motion are defined over some region. In the case of the initial values this region is the entire domain at time $t = 0$. The boundary where displacement is set is called the Dirichlet boundary (Γ_D). The boundary where the stress is set is called the Neumann boundary (Γ_N). The value to which a boundary value is set is denoted by an overbar ($\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{t}}$). These constraints on the domain are shown in Figure 4.1.

Table 4.1: Summary of linear elastodynamics

1	$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right) \quad (3.21)$
2	$\boldsymbol{T} = \lambda(\text{tr } \boldsymbol{\varepsilon})\boldsymbol{I} + 2\mu\,\boldsymbol{\varepsilon} \quad (3.54)$
3	$\rho\frac{\partial^2 \boldsymbol{u}}{\partial t^2} - \text{div } \boldsymbol{T} - \rho\mathbf{b} = 0 \quad (3.79)$

Table 4.2: Additional constraints on the equation of motion

Initial values	$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$ on Ω $\dot{\boldsymbol{u}}(\boldsymbol{x}, 0) = \dot{\boldsymbol{u}}_0(\boldsymbol{x})$ on Ω
Boundary conditions	$\boldsymbol{u} = \overline{\boldsymbol{u}}$ on Γ_D $\boldsymbol{T}\hat{\boldsymbol{n}} = \overline{\boldsymbol{t}}$ on Γ_N

4.2 Time integration

Explicit time integration schemes assume equilibrium at the current time step, this allows for a direct computation of the displacement at the next time step. Implicit time integration schemes assume equilibrium at the next time step. In order to determine the displacement at the next time a system of equations must be solved. Where only a few degrees of freedom are present solving such a system is not overly computationally expensive, but for problems with many degrees of freedom (such as those generated by meshless methods), the computational cost is very high. A time free problem also required the solution of a system of equations. SPH is thus suited for solution via explicit methods.

Explicit time integration schemes are conditionally stable, with stability depending on some maximum time step. This maximum time step is based on some length parameter, in this case, the size of the cover function (h). This time step is generally "small", but in applications where a small time step is required anyway this is not an issue.

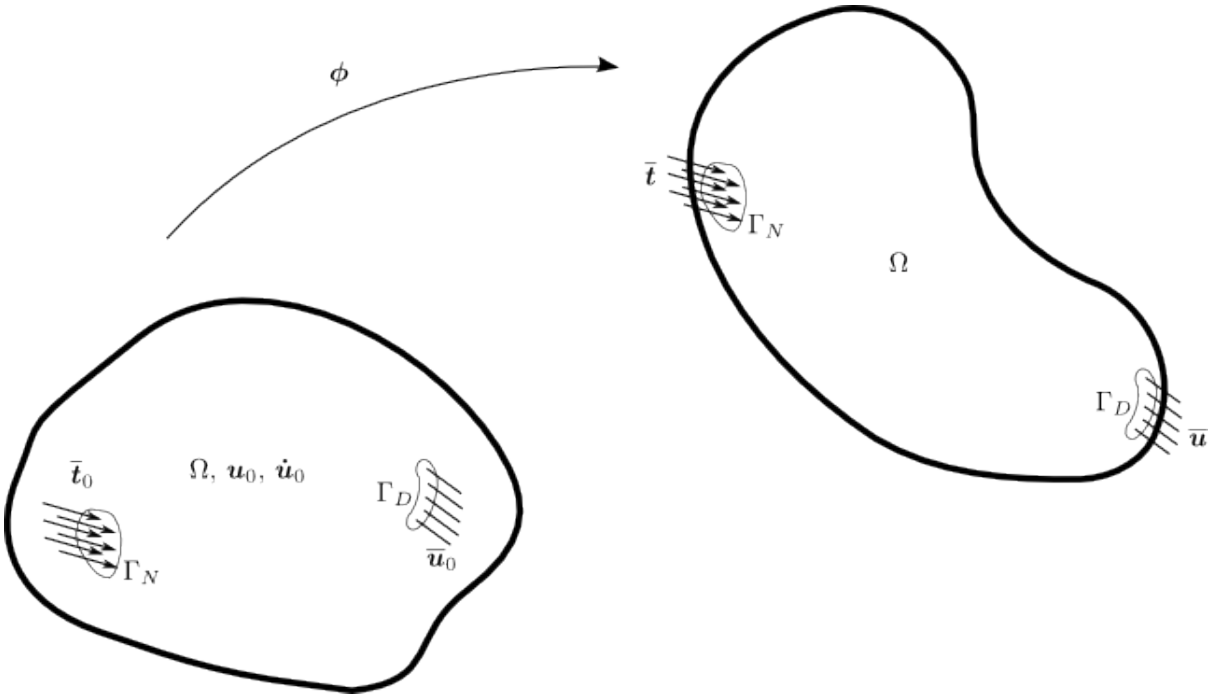


Figure 4.1: Additional constraints on Ω

Elastodynamics is such an application, where high wave velocities necessitate small time steps.

The literature documents a "Leapfrog" or *Verlet* approach to explicit time integration [7,18]. But, as these are simply a central difference, with "half stepping", a simple central difference approach was followed. The central difference scheme used is developed below.

We begin by taking a Taylor expansion of the position at time $n+1$ and at time $n-1$ about time n:

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta t \frac{\partial \boldsymbol{u}_n}{\partial t} + \left(\frac{\Delta t^2}{2}\right) \frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} + \text{higher order terms} \tag{4.1}$$

$$\boldsymbol{u}_{n-1} = \boldsymbol{u}_n - \Delta t \frac{\partial \boldsymbol{u}_n}{\partial t} + \left(\frac{\Delta t^2}{2}\right) \frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} + \text{higher order terms} \tag{4.2}$$

Adding equation (4.2) to equation (4.1) we get:

$$\frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} = \frac{1}{\Delta t^2} \left(\boldsymbol{u}_{n+1} - 2\boldsymbol{u}_n + \boldsymbol{u}_{n-1}\right) \tag{4.3}$$

Making the displacement at the next time step the subject of the formula, we obtain

$$\boldsymbol{u}_{n+1} = \Delta t^2 \frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} + 2\boldsymbol{u}_n - \boldsymbol{u}_{n-1} \ . \tag{4.4}$$

Equation (4.3) is the evolution equation used to find the displacement for the next time step. This equations requires knowledge of $\frac{\partial^2 \boldsymbol{u}_n}{\partial t^2}$ at the current time step. This value shall be determined using SPH approximations.

4.3 SPH approximation of linear elastic mechanics

The term $\frac{\partial^2 \boldsymbol{u}_n}{\partial t^2}$ in (4.4) can be found by rearranging equation (3.79) to give

$$\frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} = \mathbf{b}_n + \frac{\text{div } \boldsymbol{T}_n}{\rho} \ . \tag{4.5}$$

It is assumed that all body forces ($\mathbf{b}$) are known for all time steps. It is assumed that any change to density (ρ) will be negligible based on the infinitesimal strain assumption. $\boldsymbol{T}_n$is defined by (3.54) and is dependent on $\boldsymbol{\varepsilon}$, which is described by equation (3.21) which in turn is dependent on $\nabla \boldsymbol{u}_n$

We apply the SPH approximation (2.20) to each equation, determining an approximation of each value required at each particle.

We begin with the SPH approximation of $\nabla \boldsymbol{u}_n$ at a particle q. Using the approximation for a gradient

$$< \nabla f > (\boldsymbol{x}) = -\sum_{q=1}^{N} f(\boldsymbol{x}_q) \nabla W(\boldsymbol{x} - \boldsymbol{x}_q, h) \Delta V_q \ , \tag{2.23}$$

we obtain

$$< \nabla \boldsymbol{u}_n >_p = -\sum_{q=1}^{N} (\boldsymbol{u}_n)_q \nabla W(\boldsymbol{x}_p - \boldsymbol{x}_q, h) \Delta V_q \ . \tag{4.6}$$

This approximation can in turn be used to find $\boldsymbol{\varepsilon}$ from the expression

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \right) \ , \tag{3.21}$$

$$(\boldsymbol{\varepsilon}_n)_p \approx \frac{1}{2} \left(< \nabla \boldsymbol{u}_n >_p + (< \nabla \boldsymbol{u}_n >_p)^T \right) \ . \tag{4.7}$$

From equation (3.54) the stress can be directly computed from this value of $\boldsymbol{\varepsilon}$. That is,

from

$$\boldsymbol{T} = \lambda(\text{tr } \boldsymbol{\varepsilon})\boldsymbol{I} + 2\mu\,\boldsymbol{\varepsilon} \tag{3.54}$$

we get

$$(\boldsymbol{T}_n)_p \approx \lambda(\text{tr } (\boldsymbol{\varepsilon}_n)_p\boldsymbol{I} + 2\mu\,(\boldsymbol{\varepsilon}_n)_p \ . \tag{4.8}$$

An approximation for $div\,\boldsymbol{T}_n$ with respect to the value of $\boldsymbol{T}_n$ at the particles is required. This is also done with an SPH approximation. The notation of div $f = \nabla \cdot f$ is used since we are working in Cartesian coordinates. from the expression

$$\text{div } \boldsymbol{T} = \nabla \cdot \boldsymbol{T} \tag{4.9}$$

we get

$$< \nabla \cdot \boldsymbol{T}_n >_p = -\sum_{q=1}^{N} (\boldsymbol{T}_n)_q \nabla \cdot W\,(\boldsymbol{x}_p - \boldsymbol{x}_q, h)\,\Delta V_q \ . \tag{4.10}$$

This allows us to find the acceleration of each particle q using equation (4.5). That is,

$$\frac{\partial^2 (\boldsymbol{u}_n)_q}{\partial t^2} = (\mathbf{b}_n)_q + \frac{(\text{div } \boldsymbol{T}_n)_q}{\rho_q} \ . \tag{4.11}$$

Finally, the displacement of each particle p is found using equation (4.4) which gives

$$(\boldsymbol{u}_{n+1})_p = \Delta t^2 \frac{\partial^2 (\boldsymbol{u}_n)_p}{\partial t^2} + 2(\boldsymbol{u}_n)_p - (\boldsymbol{u}_{n-1})_p \ . \tag{4.12}$$

4.4 SPH implementation

We give a full description of a solution technique, and all that remains is to arrange the results from Section 4.3 into an algorithm.

4.4.1 Implementation of initial values and boundary conditions

Initial loading is assumed to be gradual (relative to the length of a time step). This allows for any system to be started at rest, in equilibrium. By imposing these starting conditions it is possible to begin a problem by assuming zero displacement and stress fields. Any change to the system is then applied through a time-dependent boundary

condition. Boundary conditions are imposed using the technique developed in Section 2.8.3.

4.5 One-dimensional linear elastodynamic algorithm

The algorithm included below, (Algorithm 4.1), shows the implementation in a pseudo-code format. The implementation itself was done in MATLAB, and because each test required slight modifications to the code, the code itself is not presented. It is available on the included compact disc.

The algorithm below is for the one-dimensional case only, where the tensorial computations are reduced to a single dimension. The matrices $\mathbf{W}$ and $\mathbf{X}$ are the "SPH matrices", and define the SPH approximation of every particle with respect to every other particle, as per Section 2.6. Any vector of values multiplied by one of these matrices finds the SPH approximation or the SPH approximation of the gradient of these values in a vectorised manner.

All values such as $\mathbf{u}$ and $\mathbf{T}$ should be read as the vector of these values for *every* particle unless indicated with a subscript denoting which particular particle is in question.

It is assumed that the left hand boundary is a Neumann boundary, and the right hand boundary is a Dirichlet one to allow for the inclusion of both types of boundary condition in the algorithm. The values D and N represent the denominators in equations (2.42) and (2.44) respectively and are used in the application of the Dirichlet and Neumann boundary conditions. Defining these values before the time stepping loop saves considerable computational time.

The algorithm implemented uses the modified mollifier function presented in Section 2.2.2. It should be noted that the calculation of stress in the algorithm is done using Young's modulus (E), as defined in equation (4.22) on page 67.

Algorithm 4.1: SPH linear elasticity

. **Data**: Initial geometry, $\overline{\mathbf{u}}(t)$, $\overline{\mathbf{t}}(t)$, Δt

set ,$\mathbf{u}_0$, $\mathbf{u}_1$,$\dot{\mathbf{u}}_0$, $\dot{\mathbf{u}}_1$,$\mathbf{T}_0$and $\mathbf{T}_1$ for all p ; // set initial values

set ΔV for all p

Compute $\mathbf{W}$ and $\mathbf{X}$ // calculate SPH matrices

$$\mathsf{D} = \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left[W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right)\right]^2 \Delta V_q$$

$$\mathsf{N} = \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \nabla W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \Delta V_q$$

while $n < n\_total$ **do** // for each time step

  $\eta_{\text{left}} = \left(\overline{\mathbf{t}}(t) - \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left(\boldsymbol{u}(\boldsymbol{x}_q)\right) \nabla W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \Delta V_q\right) / \mathsf{N}$ // Γ_N

  $\eta_{\text{right}} = \left(\overline{\mathbf{u}}(t) - \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left(\boldsymbol{u}(\boldsymbol{x}_q)\right) W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \Delta V_q\right) / \mathsf{D}$ // Γ_D

  $\mathbf{u} = \mathbf{u} + \eta_{\text{right}} W\left(\boldsymbol{x}_{\Gamma_N} - \boldsymbol{x}_q, h\right) + \eta_{\text{left}} W\left(\boldsymbol{x}_{\Gamma_D} - \boldsymbol{x}_q, h\right)$ // set $\mathcal{G}$

  $< \dfrac{\partial \mathbf{u}}{\partial x} >= \mathbf{X}\mathbf{u}$ // calc gradient

  $\varepsilon =< \dfrac{\partial \boldsymbol{u}}{\partial x} >$

  $\mathbf{T}_{11} = E\varepsilon$

  $< \dfrac{\partial \mathbf{T}_{11}}{\partial x} >= \mathbf{X}\mathbf{T}_{11}$ // calc gradient

  $\mathbf{a} =< \dfrac{\partial \mathbf{T}_{11}}{\partial x} > /\rho$

  $\mathbf{u}_{n+1} = \Delta t^2 \boldsymbol{a} + 2\mathbf{u} - \mathbf{u}_{n+-1}$

  $n = n + 1$; // increment time step

end

return $\mathbf{u}$, $\mathbf{T}$ *for all* n

4.6 Testing of SPH implementation

Any approximation of a PDE will, by definition, vary from the exact solution. This variation needs to be quantified, and analysed. Where possible this testing will include a comparison of an approximation of a simple problem where a known solution to the PDE exists.

Stress and displacement are the key variables of elastodynamics, where the stress-displacement relationship varies with time, in a wave-like manner. In higher dimensions these waves can interact with boundaries and other waves in a highly complex manner. This can make validation of the approximation difficult.

In one-dimensional analyses however, the wave behaviour is tractable, and a closed form solution exists. This solution can be used as a measure of the SPH approximation. The theory of elastic wave propagation in one dimension is presented here, followed by an analysis of an SPH approximation of the system described.

4.6.1 The wave nature of elastodynamics

One-dimensional displacement

The elastodynamics problem is reduced to a one-dimensional scalar problem if there is only one non-zero component of displacement which is a function of one spatial coordinate and time. This direction is chosen to be the x_1 direction. This displacement can either be in the longitudinal or transverse directions which can be described respectively as

$$\boldsymbol{u} = u_1(x_1, t)\,\boldsymbol{e}_1 \tag{4.13}$$

and

$$\boldsymbol{u} = u_2(x_1, t)\,\boldsymbol{e}_2\ . \tag{4.14}$$

The longitudinal waves will be dealt with in detail for the validation of the SPH implementation.

Equation (3.54) can be written in index form as

$$T_{ij} = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij}\ . \tag{4.15}$$

The direct stress T_{11} in the x_1 direction will be shown to be dependent only on the direct strain ε_{11} for either one-dimensional configuration.

One-dimensional strain

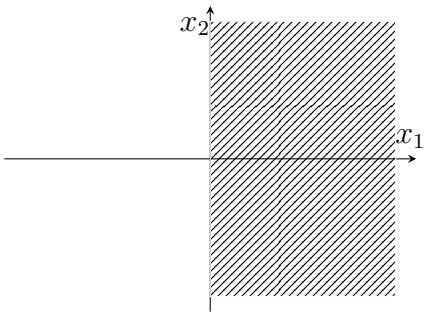


Figure 4.2: A half-space in two dimensions

A one-dimensional strain state is created when a uniform displacement in the longitudinal direction occurs in a half-space (as shown in Figure 4.2). This results in a single non-vanishing component of strain ε_{11}. While stresses acting perpendicular to the x_1 axis exist, we are only interested in the stress acting on a surface normal to the x_1 axis in the direction of the axis. For equation (4.13) we can write

$$\begin{aligned} T_{11} &= \lambda\, \varepsilon_{kk}\delta_{11} + 2\mu\varepsilon_{11} \\ &= (\lambda + 2\mu)\varepsilon_{11} \ . \end{aligned} \tag{4.16}$$

The one-dimensional strain state can also be created in a bar where no change in cross-sectional area is permitted. This configuration is shown in figure 4.3. Wide plates undergoing uni-axial loading can be treated using this approximation.

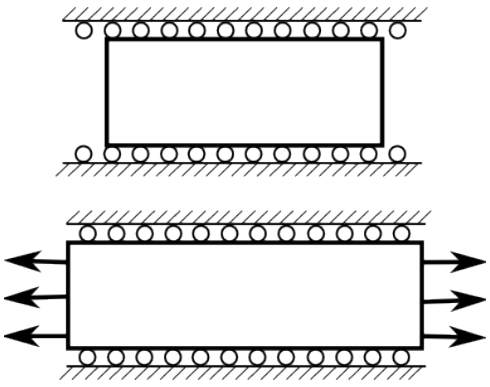


Figure 4.3: Infinitesimal portion of bar undergoing one-dimensional strain due to tension

One-dimensional stress

If a change in cross-sectional area is allowed and the bar is relatively thin, a state of one-dimensional stress is created. No shear strains are created but transverse strains

Figure 4.4: Infinitesimal portion of bar undergoing one-dimensional stress due to tension

exist due to the change in the cross-section of the bar. The only non-zero stress is that acting on a surface normal to the x_1 axis, in the direction of the x_1 axis. To find the relationship between stress and strain in this case we first need to make ε_{ij} the subject of equation (4.15). This is written as

$$\varepsilon_{ij} = \frac{1}{2\mu}\left(T_{ij} - \frac{\lambda}{3\lambda + 2\mu} T_{kk}\delta_{ij}\right) \ . \tag{4.17}$$

Applying the definition that only T_{11} is non-zero we can write

$$\begin{aligned}
\varepsilon_{11} &= \frac{1}{2\mu}\left(T_{11} - \frac{\lambda}{3\lambda + 2\mu} T_{11}\delta_{ij}\right) \\
&= \frac{\lambda + \mu}{\mu(3\lambda + 2\mu)} T_{11} \ , \qquad (4.18) \\
\varepsilon_{22} = \varepsilon_{33} &= \frac{\lambda}{2\mu(3\lambda + 2\mu)} T_{11} \\
&= -\underbrace{\left[\frac{\lambda}{2(\lambda + \mu)}\right]}_{\nu} \varepsilon_{11} \ , \\
\varepsilon_{12} = \varepsilon_{13} = \varepsilon_{23} &= 0 \ . \qquad (4.19)
\end{aligned}$$

The ratio of strains in the axis upon which displacement is dependent and those normal to it is known as the *Poisson's ratio* which is defined as

$$\nu = \frac{\lambda}{2(\lambda + \mu)} \ . \tag{4.20}$$

Using these definitions of strain we can now write equation (4.15) as

$$\begin{aligned} T_{11} &= \lambda\varepsilon_{11}\delta_{11} + \varepsilon_{22}\delta_{11} + \varepsilon_{33}\delta_{11} + 2\mu\varepsilon_{11} \\ &= \lambda\varepsilon_{11}\left[1 - \frac{\lambda}{\lambda+\mu}\right] + 2\mu\varepsilon_{11} \\ &= \underbrace{\left[\frac{\mu(3\lambda+2\mu)}{\lambda+\mu}\right]}_{E}\varepsilon_{11}. \end{aligned} \tag{4.21}$$

The relationship between stress and strain in a system undergoing one-dimensional stress is known as the *Young's modulus* and is defined as

$$E = \frac{\mu\,(3\lambda+2\mu)}{\lambda+\mu}. \tag{4.22}$$

An infinitesimal portion of the bar is shown under deformation in Figure 4.4. This approximation is used to model thin bars and rods undergoing uniaxial displacement.

General statement of the stress-strain relation in one dimension

The relationship between stress and strain for both one-dimensional stress and strain is linear. For convenience this is written as

$$T_{11} = \mathcal{C}\varepsilon_{11} \tag{4.23}$$

where $\mathcal{C}$ is the one-dimensional form of the elasticity tensor.

The one-dimensional stress assumption is regularly used in Hopkinson pressure bar (HPB) experimental work [27]. The experimental technique is used to determine material properties at high strain rates. While the technique itself is not important here, it was decided that should the SPH approximation be experimentally validated, the validation would utilise the HPB technique. Thus one-dimensional stress wave propagation was chosen to validate the SPH approximation of elastodynamics.

One-dimensional elastic wave propagation

A state of stress or strain *propagates* along a body with time. This propagation is said to be *wave-like*, and the states are referred to as either stress or strain waves. This behaviour

can be demonstrated in a semi-infinite one-dimensional body. To simplify the analysis of the wave propagation a linear elastic, homogeneous, isotropic material model is used. The development of the theory follows that in Achenbach [28].

The equation of motion, equation (3.79), can be simplified to

$$\frac{\partial T_{11}}{\partial x_1} = \rho \frac{\partial^2 u}{\partial t^2} \tag{4.24}$$

under the condition of one-dimensional longitudinal motion (described in (i) in equation (4.15)) and the assumption that body forces are negligible. Because only one component of $\boldsymbol{x}$ is used the subscript is dropped to simplify the equations which follow.

Substituting equation (4.23) into equation (4.24) we get

$$\frac{\partial^2 u}{\partial x^2} = \frac{\rho}{\mathcal{C}} \frac{\partial^2 u}{\partial t^2}. \tag{4.25}$$

For simplicity the initial conditions are chosen such that

$$\left.\begin{aligned} u &= 0 \\ \frac{\partial u}{\partial t} &= 0 \end{aligned}\right\} \qquad \text{for} \qquad t = 0 \tag{4.26}$$

Note that these initial conditions also match those assumed in Section 4.4.

A general solution to equation (4.25) [28], which is a wave equation, is

$$u(x,t) = f\left(t - \frac{x}{c}\right) + g\left(t + \frac{x}{c}\right). \tag{4.27}$$

where c is defined as

$$c^2 = \frac{\mathcal{C}}{\rho}\,. \tag{4.28}$$

Enforcing the initial conditions we get

$$f\left(\frac{-x}{c}\right) + g\left(\frac{x}{c}\right) = 0 \tag{4.29}$$

and

$$\frac{\partial}{\partial t} f\left(\frac{-x}{c}\right) + \frac{\partial}{\partial t} g\left(\frac{x}{c}\right) = 0. \tag{4.30}$$

The solution to these equations is

$$f\left(\frac{-x}{c}\right) = -g\left(\frac{x}{c}\right) = C \tag{4.31}$$

where C is a constant.

Noting that

$$t + \frac{x}{c} > 0 \qquad \text{for} \qquad t \geqq 0 \quad \text{and} \quad x > 0 \tag{4.32}$$

equation (4.27) can be simplified to state

$$u(x,t) = \begin{cases} f\left(t - \dfrac{x}{c}\right) - C & \text{for} \quad t > \dfrac{x}{c}\ , \\ \\ 0 & \text{for} \quad t \leq \dfrac{x}{c}\ . \end{cases} \tag{4.33}$$

The implication of this is that a point within the domain at $x = \overline{x}$ will not displace until time $t = \overline{t} = \dfrac{\overline{x}}{c}$. This defines a wave like motion where the disturbance moves at a speed c, the wave speed.

Applying the boundary condition at $x = 0$ we get

$$-\frac{\mathcal{C}}{c}\frac{\partial}{\partial t} f(t) = -p(t) \tag{4.34}$$

where $p(t)$ is the surface traction on the end of the bar at $x = 0$.

Integrating we get

$$f\left(t - \frac{x}{c}\right) = \frac{c}{\mathcal{C}} \int_0^{t-x/c} p(s)\, \mathrm{d}s + B \tag{4.35}$$

where B is a constant.

Setting $t = 0$ and substituting into equation (4.31) we find that $B = A$. If we note that $p = 0$ when $t \leqq 0$ we can write

$$u(x,t) = \frac{c}{\mathcal{C}} \int_0^{t-x/c} p(s)\, \mathrm{d}s. \tag{4.36}$$

Substituting this into equation (4.24) we get the expected result

$$T_{11} = -p\left(t - \frac{x}{c}\right)\ . \tag{4.37}$$

For one-dimensional strain $\mathcal{C} = \lambda + 2\mu$ and the associated wave speed is called the *longitudinal wave speed*, and is defined as

$$c_L^2 = \frac{\lambda + 2\mu}{\rho} \ . \tag{4.38}$$

For a one-dimensional stress state $\mathcal{C} = E$. Here the wave speed is referred to as the *bar wave speed*, and is defined as

$$c_b^2 = \frac{E}{\rho} \ . \tag{4.39}$$

4.6.2 Reflection and transmission of elastic waves

If the elastic body is not semi-infinite, but rather has some interface with another body, the energy of the elastic waves described in Section 4.6.1 does not disappear. Some of the energy is transmitted into the second body, while the remainder remains in the first body as a wave travelling away from the interface. By describing how the transmitted and reflective waves are related to the original induced wave we can describe bodies of finite length which interact with other bodies uni-axially.

To distinguish the three waves we use the subscript $\mathfrak{i}$ to represent the induced wave, the subscript $\mathfrak{r}$ for the reflected wave, and the subscript $\mathfrak{t}$ for the transmitted wave.

Assuming the wave has the form described in equation (4.37) we can write

$$(T_{11})_{\mathfrak{i}} = f\left(t - \frac{x}{c^A}\right) \ . \tag{4.40}$$

Naming the first body, body A, and the second body, body B we can describe the properties of the bodies with superscripts A and B. The interface is at $x = a$ as shown in Figure 4.5.

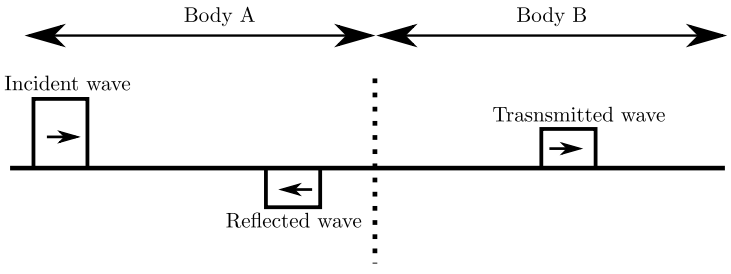


Figure 4.5: Incident, reflected and transmitted elastic waves at a material interface

If we assume no separation at the interface we can write the reflected and transmitted waves as

$$(T_{11})_{\mathfrak{r}} = g\left(t - \frac{a}{c^A} + \frac{x-a}{c^A}\right) , \tag{4.41}$$

$$(T_{11})_{\mathfrak{t}} = h\left(t - \frac{a}{c^A} - \frac{x-a}{c^B}\right) . \tag{4.42}$$

These functions g and h can be related to the incident wave f by enforcing the continuity of stress and velocity implied by the assumption that no separation occurs.

We can now write

$$(T_{11})_{\mathfrak{r}} = C_{\mathfrak{r}}\, f\left(t - \frac{x}{c^A} + \frac{x-a}{c^A}\right) , \tag{4.43}$$

$$(T_{11})_{\mathfrak{t}} = C_{\mathfrak{t}}\, f\left(t - \frac{x}{c^A} - \frac{x-a}{c^B}\right) , \tag{4.44}$$

where $C_{\mathfrak{r}}$ and $C_{\mathfrak{t}}$ are the reflection and transmission coefficients and are defined as

$$C_{\mathfrak{r}} = \frac{\rho^B c^B/\rho^A c^A - 1}{\rho^B c^B/\rho^A c^A + 1} \tag{4.45}$$

and

$$C_{\mathfrak{t}} = \frac{2\rho^B c^B/\rho^A c^A}{\rho^B c^B/\rho^A c^A + 1}. \tag{4.46}$$

If the properties of body B are related to those of body A such that $C_{\mathfrak{r}} = 0$ and $C_{\mathfrak{t}} = 1$ the wave is fully transmitted. The bodies are said to be *impedance matched*. Note that if the interface were an imaginary one in the middle of a body the two sides of the body are perfectly impedance matched and the wave propagates as expected.

If body B is selected such that it has zero stiffness (is a vacuum) $C_{\mathfrak{r}} = -1$ and $C_{\mathfrak{t}} = 0$. In this case the wave is fully reflected, with a sign inversion. This results in a tensile wave being returned as a compressive wave and vice versa.

If body B has an infinite stiffness (the interface does not move at all) $C_{\mathfrak{r}} = 1$ and $C_{\mathfrak{t}} = 0$. In this case the wave is totally reflected and the sign preserved.

4.6.3 Results for compressive behaviour

The validation of the SPH implementation of elastodynamics must be separated into four distinct areas. These relate to the basic SPH approximation to the solution, the time integration scheme, the implementation of boundaries and the overall behaviour of the system being modelled. Because each of these areas is interconnected, one is forced to examine each assuming the configuration of the others is correct. This assumption is validated once each area is proven to be valid.

The basic validation will be done using a one-dimensional "bar", where no transverse motion is permitted. Such a problem is one of one-dimensional strain, as described in Section 4.6.1. For such a problem, the wave speed is known and can be used to ensure the system is behaving correctly. Once this has been established, configuration variables relating to each of the areas listed will be modified to ascertain that each area is being correctly implemented.

This process will allow a set of general configuration settings to be generated. These settings can then be used for any further investigations.

4.6.3.1 The validation problem

The validation problem consists of a bar where only axial motion is permitted, and the bar behaves in a linear elastic manner. This allows one to model the problem using one-dimensional linear elastic mechanics.

The bar has material parameters

$$\rho = 8000$$
$$E = 200\ GPa$$

which, according to equation (4.38) results in a wave speed of

$$c_L = 5801\ m.s^{-1}\ .$$

The bar has length 0.15 m long, requiring 30 μs for a wave to travel from one side of the bar to the other.

The bar will be loaded by applying a strain pulse to the left end of the bar equivalent to a stress of

$$\boldsymbol{T}_{11} = -100\ kPa\ .$$

The right end of the bar is set to be rigidly fixed, which would result in a total reflection of any wave if the simulation were permitted to run for long enough.

To facilitate the validation, four points along the bar are identified where stress history will be compared. These points are at each end of the bar as well as at points at 0.05 m and 0.1 m along the length of the bar, as shown in Figure 4.6.

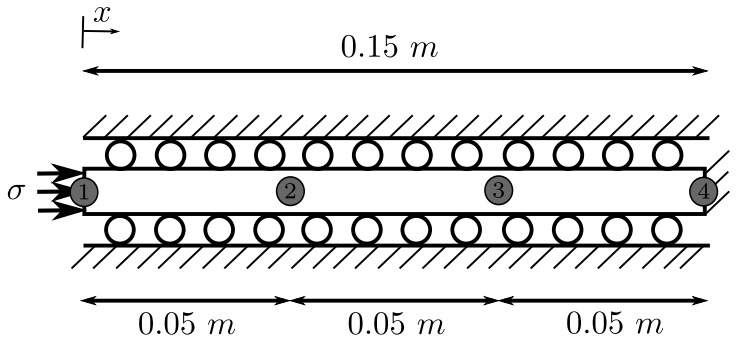


Figure 4.6: Validation problem

4.6.3.2 The effect of wave shape

In Section 2.7 the ability of SPH to approximate square waves was dealt with. It was shown that a cover size of 5.5 times the particle spacing is a good compromise. Where the waveform has steeper gradients some numerical error is to be expected. While a cover size of 1.5 times the particle spacing would handle square waves well, any other loading would cause substantial errors.

To demonstrate the effect of the wave shape on the system a smoothing has been applied to a square pulse. This smoothing is achieved by taking the SPH approximation of the square wave with various smoothing lengths. Here the particle discretisation, cover size, and time integration scheme are held constant while the loading condition is varied. The loading condition is simply an SPH interpolation of the loading pulse, where the smoothing length is as indicated.

Figure 4.7 shows the wave propagating as expected, with minimal numerical noise. Note that there is no difference between the intended load and that calculated at point 1 on the bar. Here a Neumann boundary condition has been applied exactly.

The disadvantage of such a treatment is the large level of smoothing used in the loading pulse. The wave applied is significantly different from a square wave. To reduce this approximation the amount of smoothing used on the loading function is now reduced in Figure 4.8.

Figure 4.8 is far less smoothed than Figure 4.7, approximating a square wave well. A small amount of numerical noise is present. Numerical noise of this sort is common in

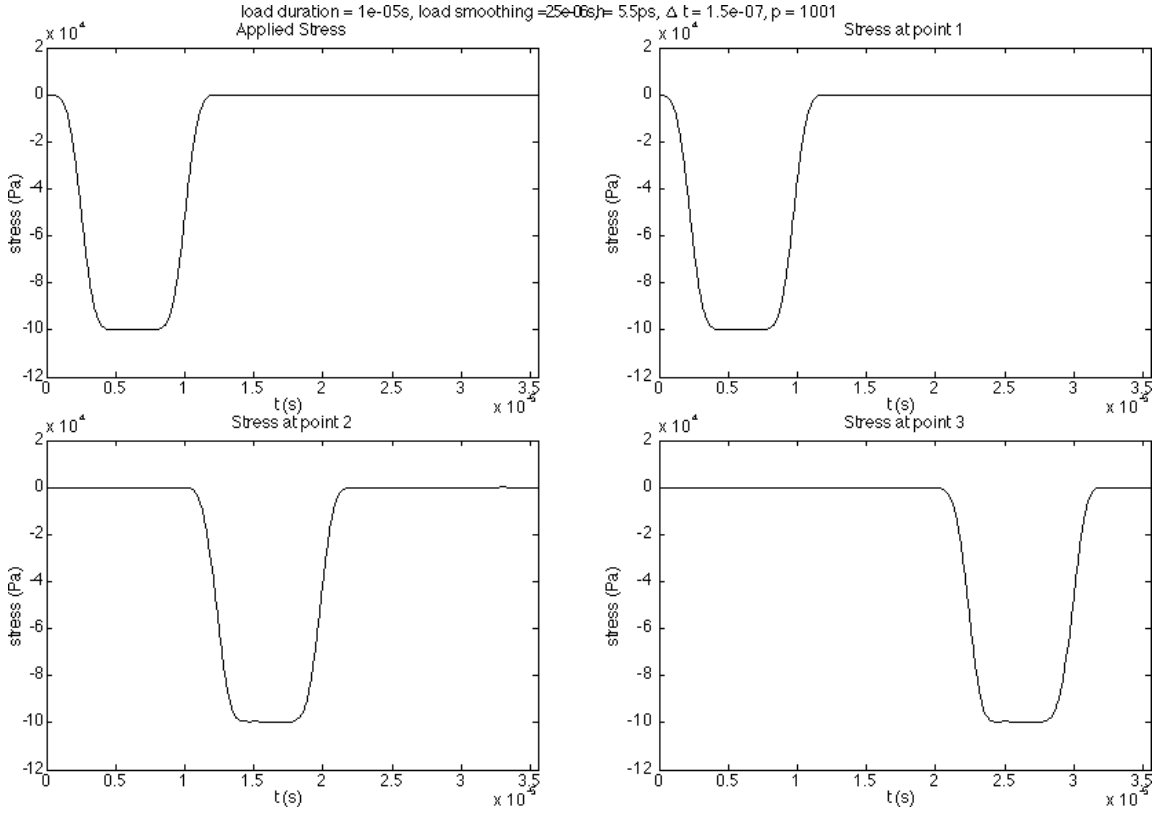


Figure 4.7: Wave propagation - large pulse smoothing

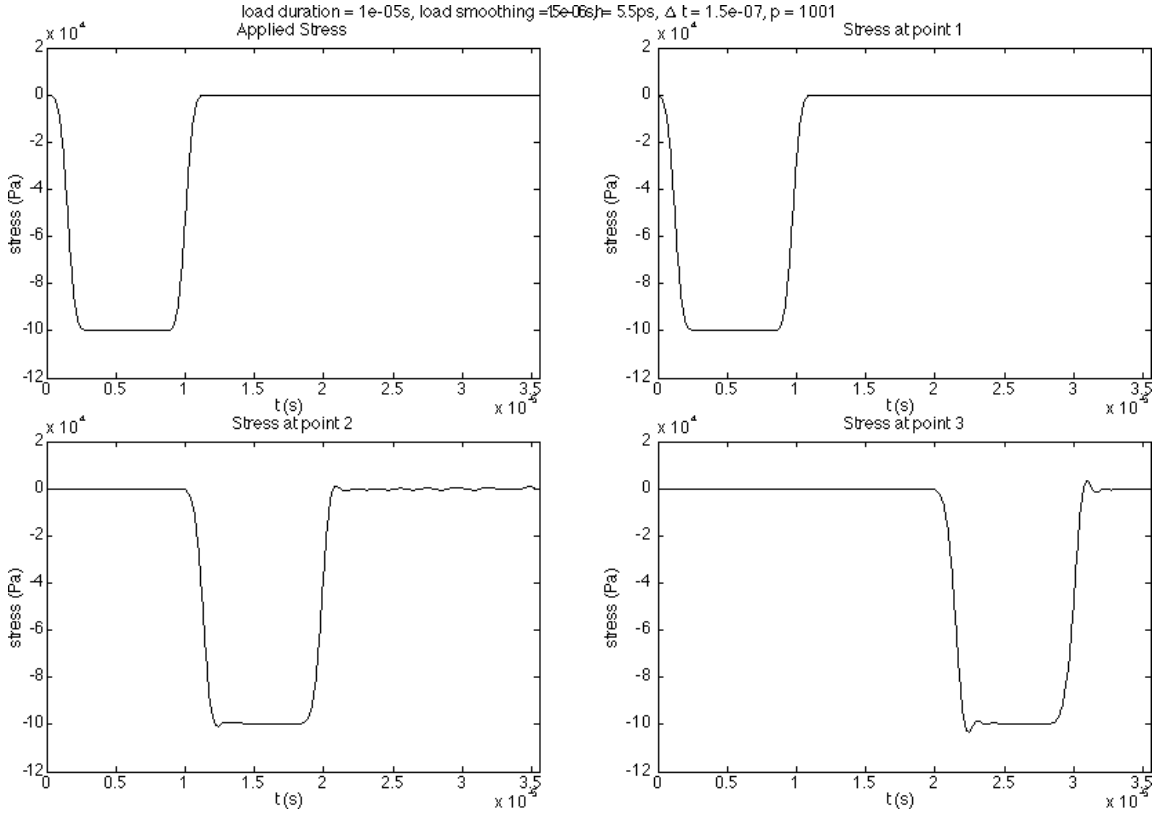


Figure 4.8: Wave propagation - intermediate pulse smoothing

any treatment of elastodynamics, with many authors implementing an artificial viscosity or upwinding scheme to deal with the phenomenon [7].

If no smoothing at all is used there is significant numerical noise, as in Figure 4.9. This noise is greater than that seen in the finite element method. This is likely due to the vast error introduced in the determination of gradients for square waves with SPH.

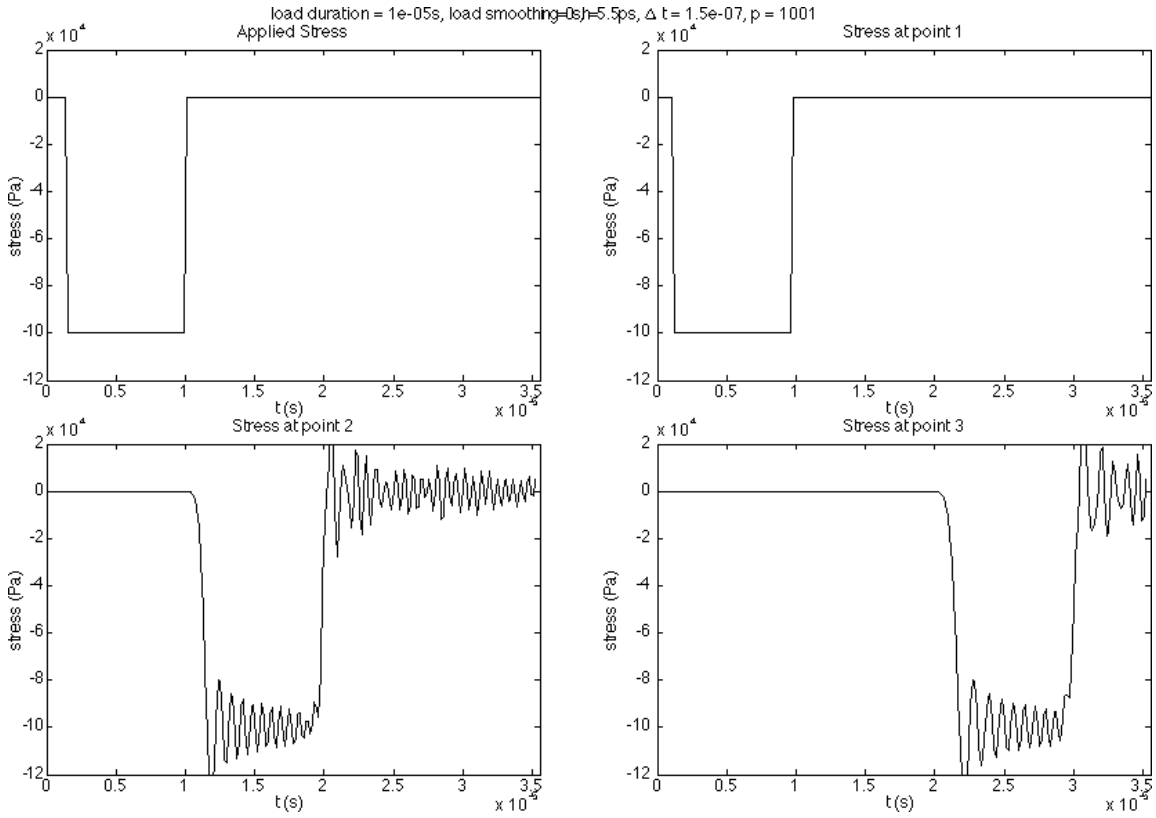


Figure 4.9: Wave propagation - no pulse smoothing

If it is essential to have very rapid rise and fall times, the number of particles used in the discretisation must be increased such that the number of particles within the transient portion effectively remove the discontinuity in the wave.

The remainder of the validation will make use of the intermediate load pulse. This is a good compromise between modelling rapid changes in the wave and maintaining a stable solution.

4.6.3.3 Time integration stability

Any explicit time integration scheme is conditionally stable, with the stability condition defined by a Courant condition. This condition is described as a function of a length scale. In finite elements this length scale is related to element size, while in SPH this length scale is less well defined.

An approximate stability constraint can be constructed by ensuring that the wave can not propagate further than the distance covered by any particles cover. This can be written as

$$\Delta t_{optimal} = \frac{h}{c} \tag{4.47}$$

where c is the smaller value between c_b and c_L.

For the configuration used this results in a critical time step of 165 ns.

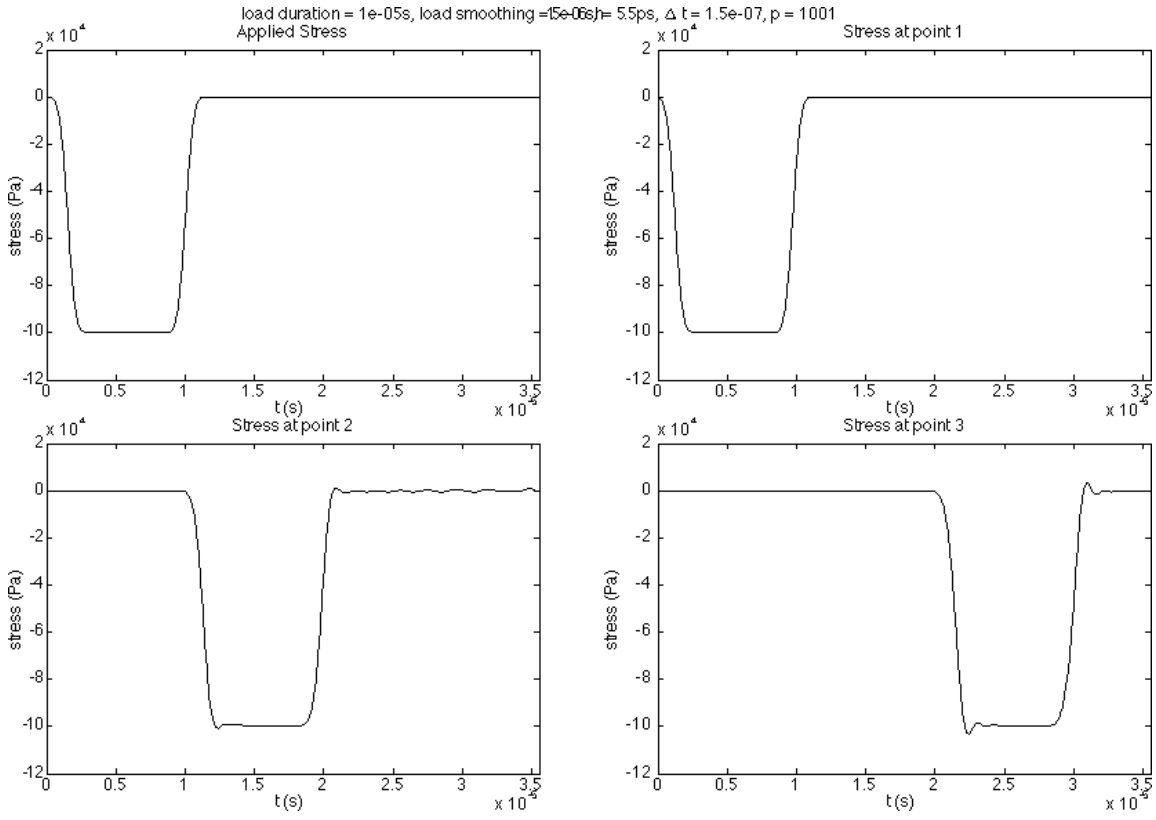


Figure 4.10: Below stability condition

The system is totally stable below the stability limit, as in Figure 4.10, with the wave travelling along the bar at the expected wave speed. Because the critical time step is assumed, a small additional increase in time step is possible. Interestingly, this solution, shown in Figure 4.11, contains *less* numerical noise than the shorter time step, although this is probably due to accumulation of error in each step.

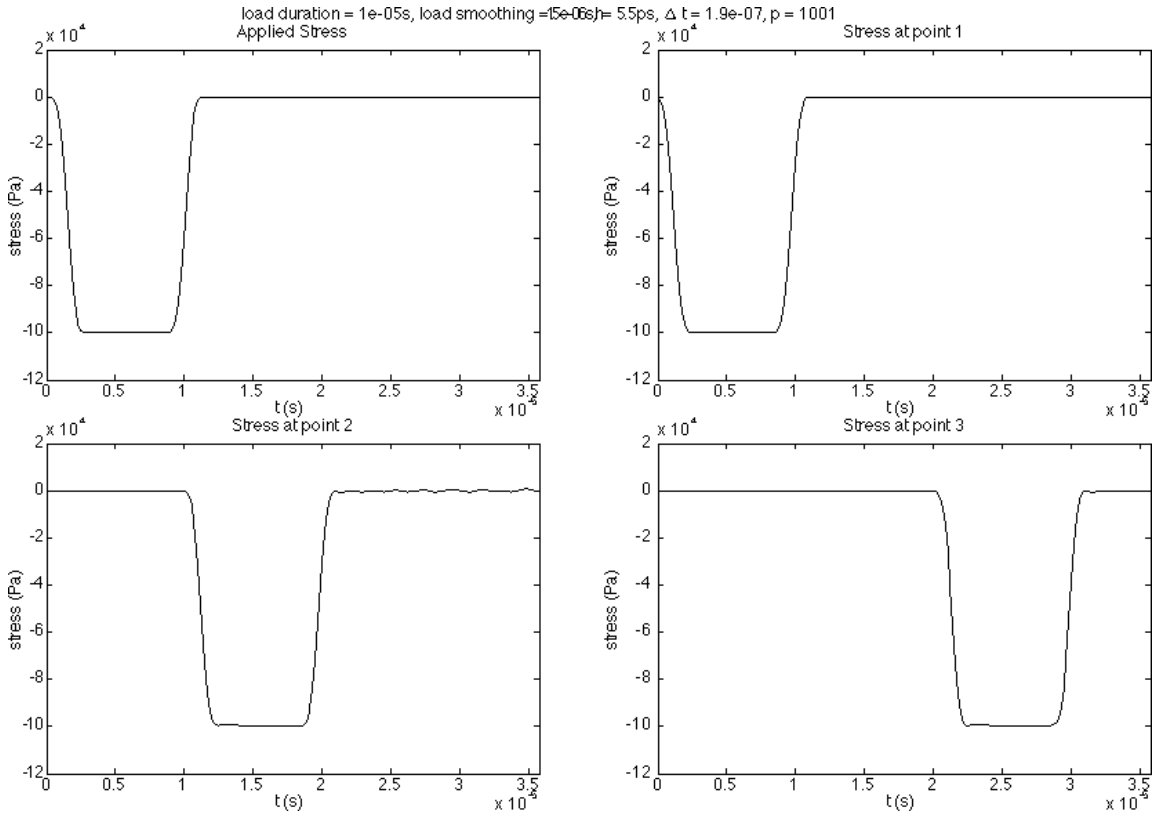


Figure 4.11: At stability condition

When the time step exceeds some value the system becomes unstable, with small perturbations growing unboundedly, as seen in Figure 4.12. For this reason it is essential to have some indication of the stability limit. It is proposed to use a time step slightly less than that given in equation (4.47).

4.6.3.4 Boundary conditions

Having ascertained that the system behaves as expected for the propagation of waves without encountering a boundary or material change one can now evaluate the behaviour after interaction with a boundary.

Using the configuration suggested, the total simulation time is increased, allowing the wave to interact with the right hand boundary. Here the displacement at the boundary is fixed, and total reflection of the wave is expected according to equation (4.43).

Notice in Figure 4.13 that the displacement at point 4 is held fixed exactly. Further, as was noted above, the loading condition on the left boundary is also applied exactly. This demonstrates that both Neumann and Dirichlet boundary conditions are imposed fully.

The wave reflects, but there is some minor error accumulation as the oscillations in the reflected wave interact with the incident wave. This is to be expected, and might be

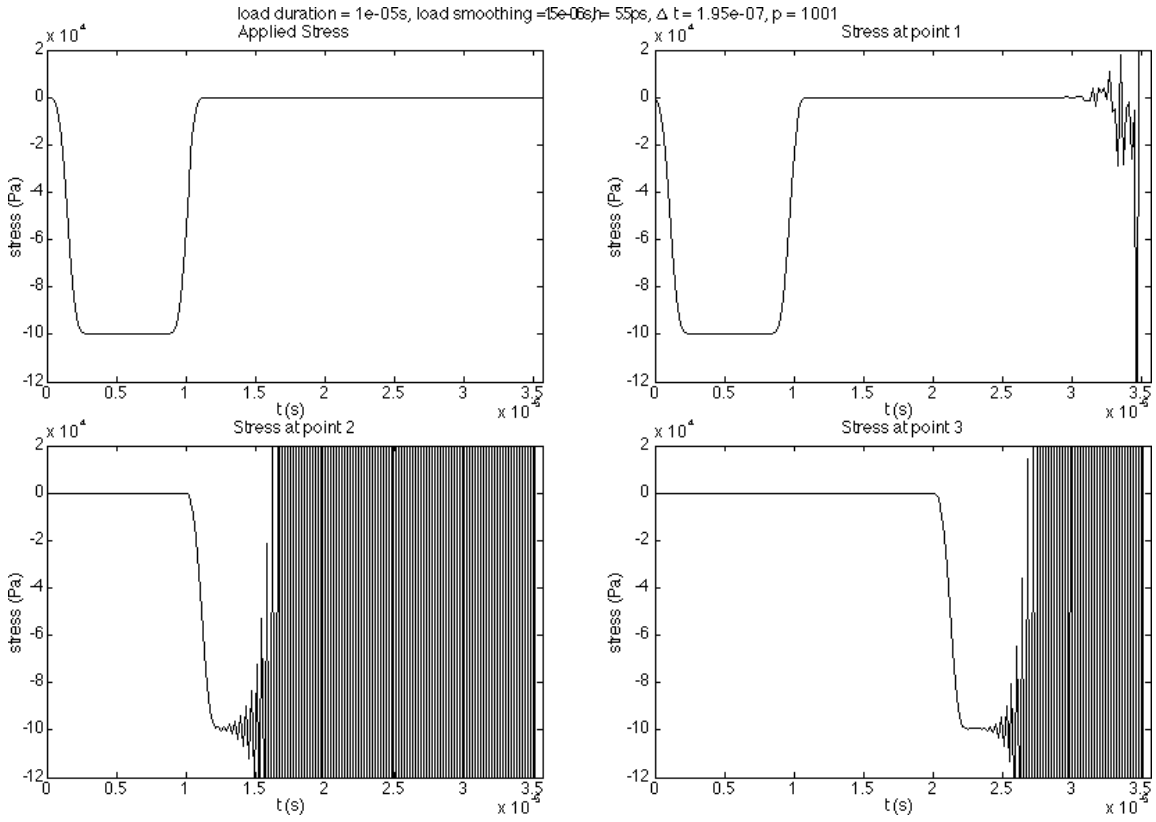


Figure 4.12: Above stability condition

reduced with the inclusion of artificial viscosity or damping.

The imposition of a Neumann boundary condition on the right end of the bar can be achieved by setting the end to be stress-free. This is done in Section 4.7.

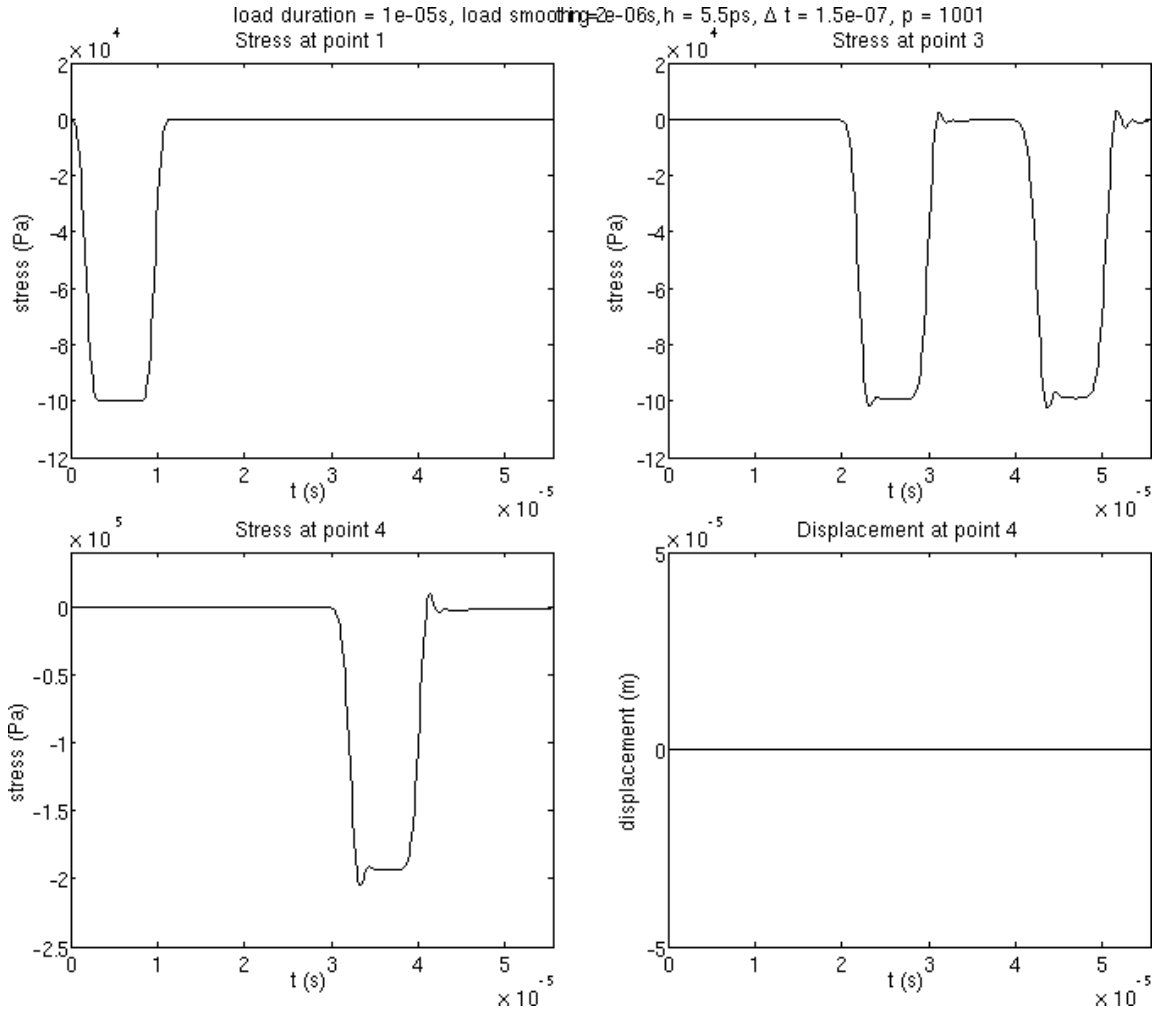
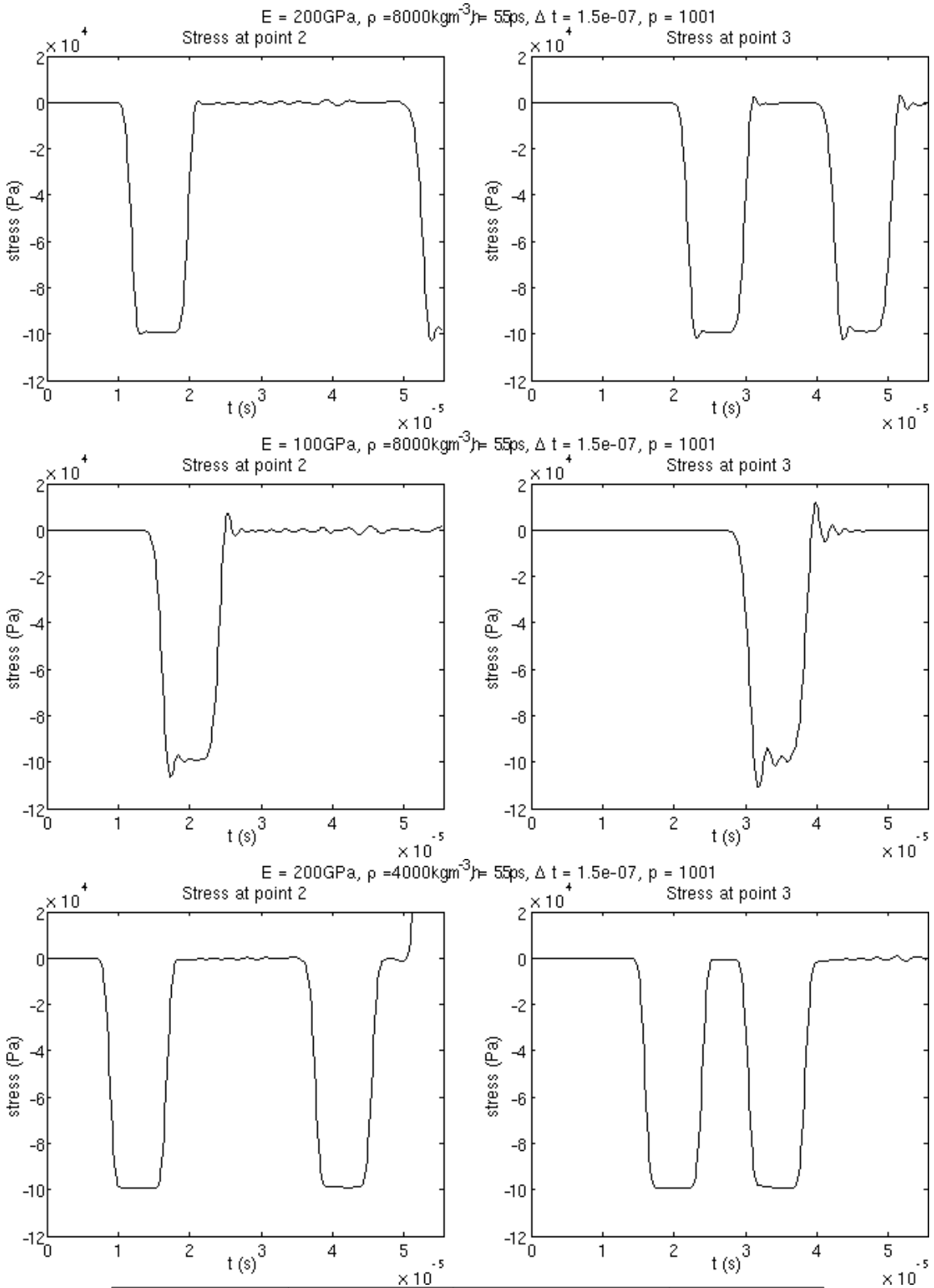


Figure 4.13: Wave reflection

4.6.3.5 Dependence on material parameters

The validation thus far has been for a specific set of material characteristics. It has been shown that the wave speed is not dependent on the smoothing length, pulse shape, or boundary conditions. The wave speed should, however, be dependent on the density and Young's modulus of the material.

Three material sets are compared in Figure 4.14, where each of the material properties is varied. The configuration for the tests will be as above, but with each of the properties defined below. In the third configuration, the tesile wave reflected off the free end is visible at stress point two after 50 μs.



Configuration	Density	Young's Modulus	Wave speed
1	8000 $kg.m^{-3}$	200 GPa	5801 $m.s^{-2}$
2	8000 $kg.m^{-3}$	100 GPa	4102 $m.s^{-2}$
3	4000 $kg.m^{-3}$	200 GPa	8204 $m.s^{-2}$

Figure 4.14: Wave Propagation - various materials

It is interesting to note that the smoothness of the approximation is dependent on the wave speed of the material in question. This is directly linked to the selection of the appropriate time step. Selecting a time step that matches the distance travelled by the wave with the cover function, minimal numerical error is introduced. If too large a time step is introduced the system becomes unstable, but when a very small time step is used numerical noise is introduced, but this noise should not lead to instability.

4.6.3.6 Heterogeneous material

The results examined thus far deal with wave propagation and total reflection of the wave. Equation (4.43) also deals with partial transmission and reflection of waves. This occurs when two portions of the domain are not "impedance matched'. This can occur through a change in either density or Young's modulus.

To demonstrate the effect the validation problem is modified such that the density of the bar can be changed at the midpoint of the bar. This effectively models two bars of different densities joined at the ends and is shown in Figure 4.15.

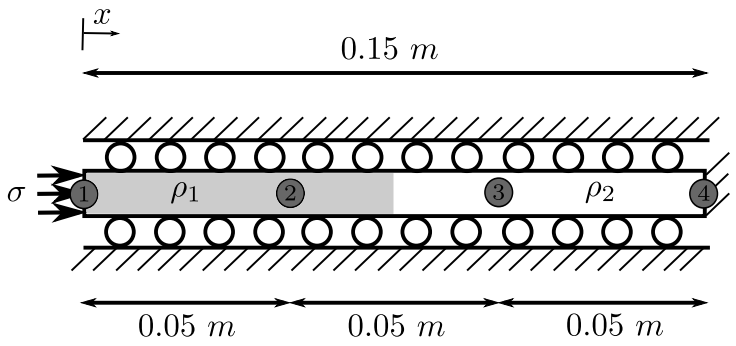


Figure 4.15: Validation problem - density change midspan

Where a drop in density is experienced, as in Figure 4.16, a portion of the wave is reflected with its sense inverted (a compressive wave is partially reflected as a tensile one), and a wave is transmitted where the magnitude of the wave increases, and the wave travels at a greater speed.

Where a density increase is experienced, as in Figure 4.17, a portion of the wave is reflected in the same sense, and a portion of the wave is transmitted with a lower magnitude. The increase in density is coupled with a decrease in the wave speed.

If the density is set to be a very large number the system should behave as if the bar were truncated at the midpoint, with no displacement being permitted beyond this point. This is demonstrated in Figure 4.18.

It is not possible to set the density to be a very small number to approximate a free end

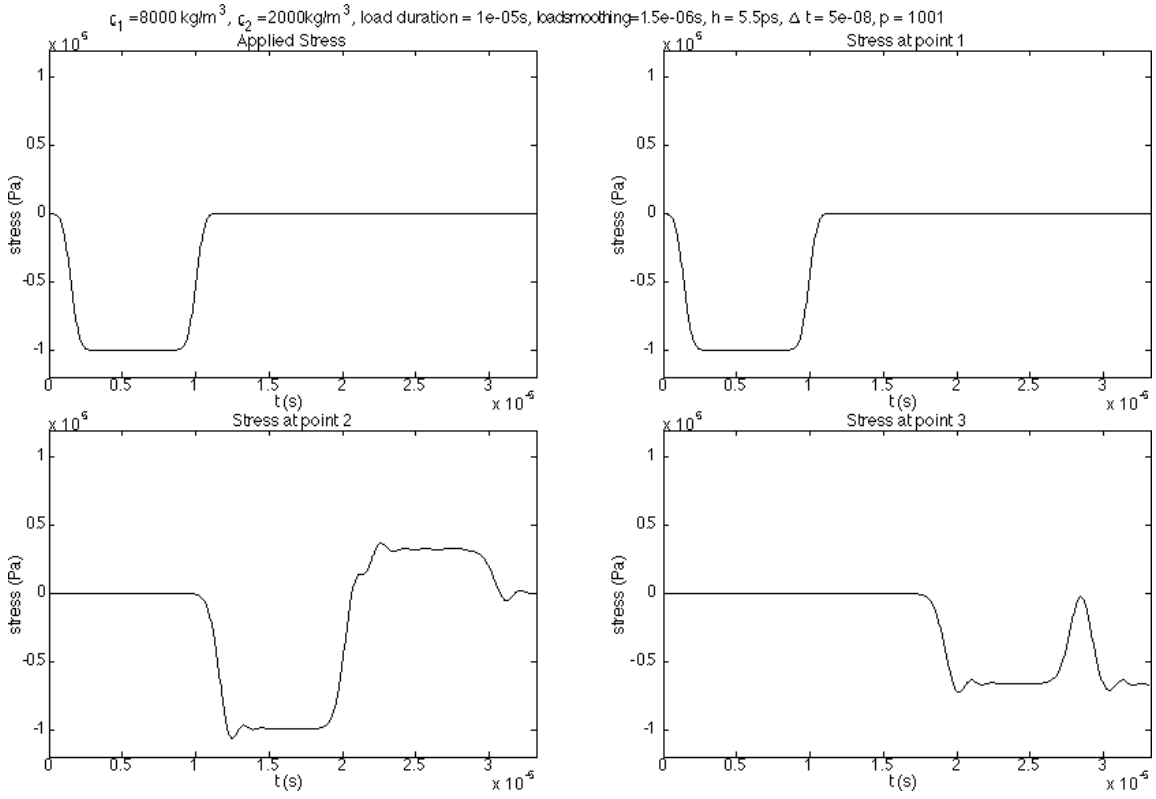


Figure 4.16: Partial transmission - decrease in density

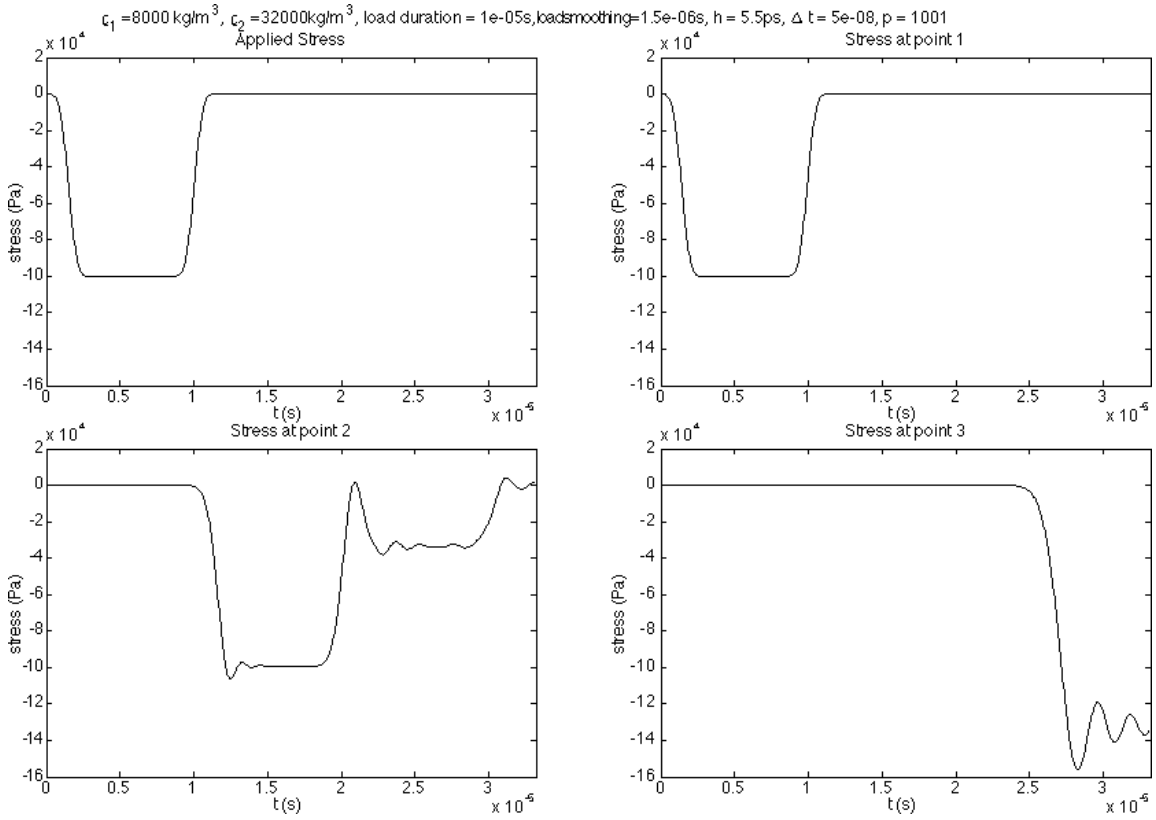


Figure 4.17: Partial transmission - increase in density

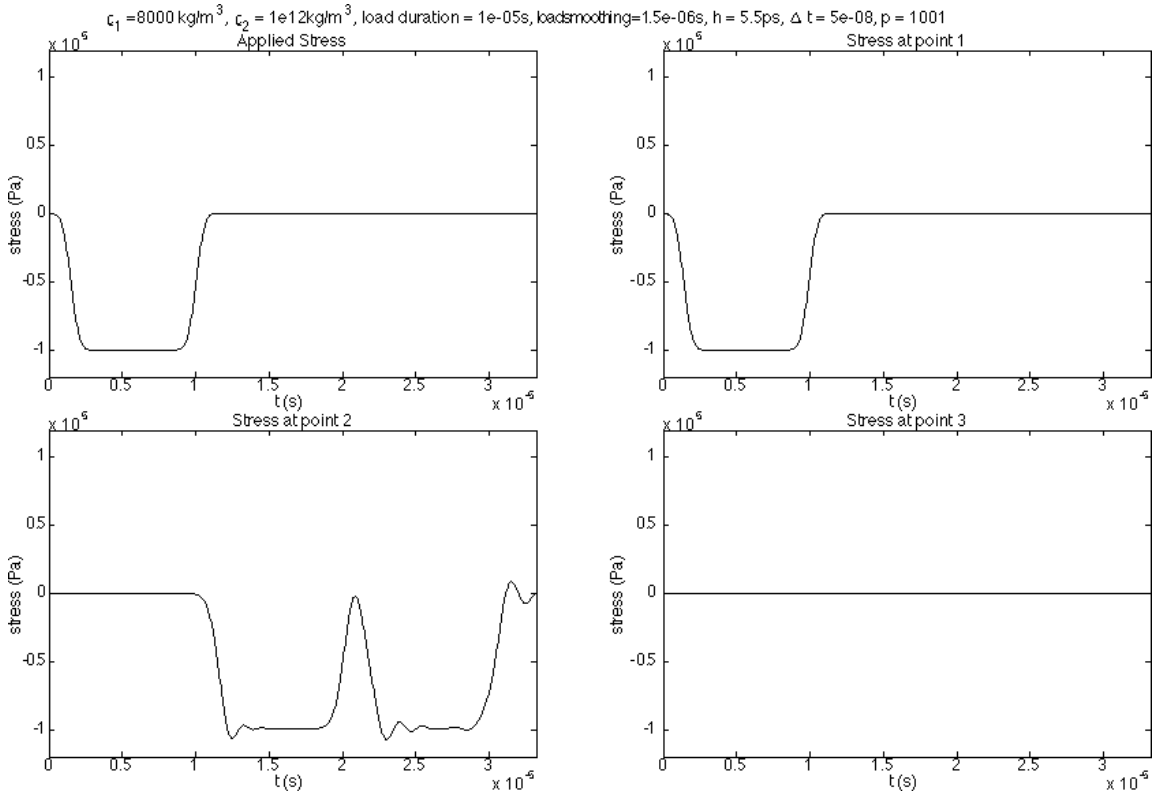


Figure 4.18: Partial transmission - very large increase in density

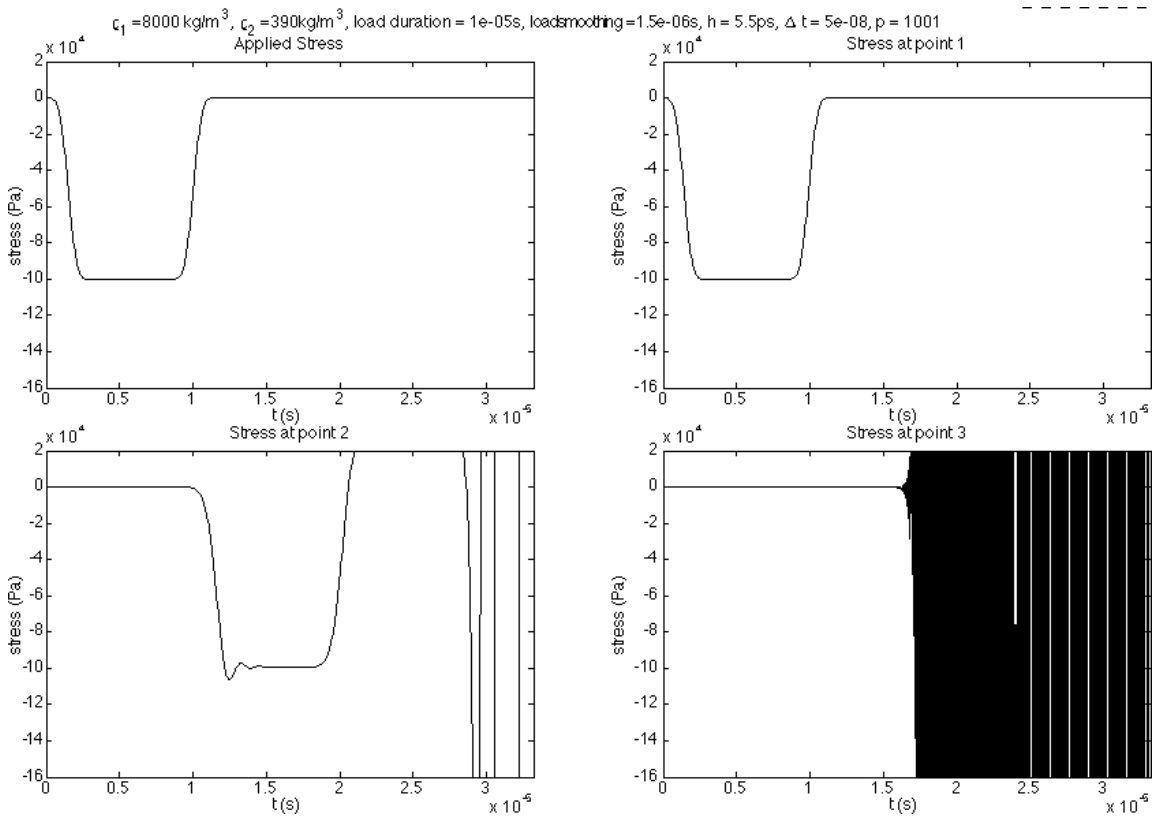


Figure 4.19: Partial transmission - very large decrease in density

because the wave speed in that portion of the bar increases dramatically, and the required time step becomes prohibitively small. This behaviour can be seen in Figure 4.19.

4.7 Tensile instability

In addition to the issues relating to boundary conditions, standard SPH implementations are prone to a numerical instability that arises in tensile regions. This has resulted in the phenomenon being commonly known as the *tensile instability*.

4.7.1 Cause of tensile instability

Tensile instability has garnered a lot of interest. The instability was formally documented in two independent von Neumann stability analyses [4, 29]. Tensile instability was one of the primary foci of a study by the SANDIA laboratories. They found that the instability was related to the calculation of gradients in SPH [4].

Although SPH can be used to solve any PDE, the issue of tensile instability is discussed in terms of stress. It is important to note however, that any property dependent on gradients may be similarly affected.

The criterion for onset of the instability is when [8]

$$W''\sigma > 0 \ . \tag{4.48}$$

where σ is the one-dimensional Cauchy stress.

The reason for this criterion can be easily seen for the stress related case. When $W'' > 0$ and $\sigma > 0$ the system is in tension. If two particles are moved apart in this region, one expects the tensile load to increase. W' decreases because of the positive value of W'', this results in the calculated stress σ decreasing. This a-physical behaviour is the same as that seen in the tests performed earlier.

A similar process can occur if, $W'' < 0$ and $\sigma < 0$. Here the system is in compression. If two particles are moved towards each other, the stress is expected to decrease (increase in magnitude in a negative sense). This does not happen however, as the negative W'' leads to an increase in W'. This results in a decrease in the compressive stress magnitude, which is, again, a-physical.

Although it has been shown that a "compressive instability' exists, it is seldom seen. This is because particles seldom fall close enough to another particle such that is it in the $W'' < 0$ range. In fact, with the one dimensional cubic B-spline, 67% of all positions

within the cover are in the tensile instability range, and particles are extremely unlikely to enter the compressive instability range. This number can be even higher, with 82% of all positions within the one dimensional mollifier cover function falling within the tensile instability range.

Because of the close link between cover function shape and the susceptibility to instability, one can create a cover function that does not go into an unstable zone, provided the domain remains in either tension or compression. Beyond this several attempts at calculating gradients in a manner different to standard SPH, termed conservative smoothing [30,31] have been made. In addition, the instability has been tackled through the addition of dissipative terms in the formulation [32], or of addition of "stress particles" [33].

While these solutions all offer potential solutions to the tensile instability, all require significant deviation from standard SPH. In keeping with the desire to implement as basic an SPH code as possible another solution was required. This was found in the total Lagrangian approach to SPH.

4.7.2 Total and updated Lagrangian approaches

Large strain problems in solid mechanics are typically cast in a Lagrangian framework, where the behaviour of a body is expressed relative to some original configuration. In an updated Lagrangian system the "original configuration" is taken to be that of the body at the start of a time step, with the behaviour of the body during the time step expressed relative to this position. This differs from a total Lagrangian approach where the "original configuration" is that at the start of the simulation, and the displacement in *every* subsequent time step is defined relative to it.

When working with the assumption of infinitesimal deformation there is no mathematical difference between the two approaches. There is however a difference in terms of SPH implementation. While displacements may be small enough to be considered "infinitisimal", a small change in the position of a particle relative to other particles can shift the gradient calculation into a state of instability.

Vignjevic *et al.* [3] present the use of a total Lagrangian implementation as a potential remedy for tensile instability. This approach is natural to follow, as it allows the computation of the particles' influence on each other prior to the time integration. The algorithm presented in Section 4.5 follows this reasoning.

It is possible, however, to recast the algorithm in an updated Lagrangian manner to allow one to identify the tensile instability. This is done by moving the computation or the "SPH matrices" to within each time step. This adds significant computational overhead, and should not generally be used.

4.7.3 Tensile linear elastic results

The short duration simulations in Section 4.6.3 do not show any significant instabilities in either compression or tension. This is not to say that the instability is necessarily absent, but rather that it has not been observed, matching with that postulated by Vignjevic *et al.* [3].

In order to demonstrate one of the manifestations of the tensile instability the results of an updated Lagrangian approach are compared to a total Lagrangian one. Here, one expects the results to be identical, but the updated Lagrangian implementation will be shown to be a-physical.

Simulations similar to those in Section 4.6.3, but in tension appear to behave correctly. A much closer investigation is required to see the onset of the instability in this configuration. Here a bar, constrained in the transverse direction, is pre-stressed by applying a linearly varying displacement, as shown in Figure 4.20. Such a configuration results in a constant stress field along the length of the bar. The two ends are kept at a constant displacement for the duration of the simulation, which should result in a totally stable configuration.

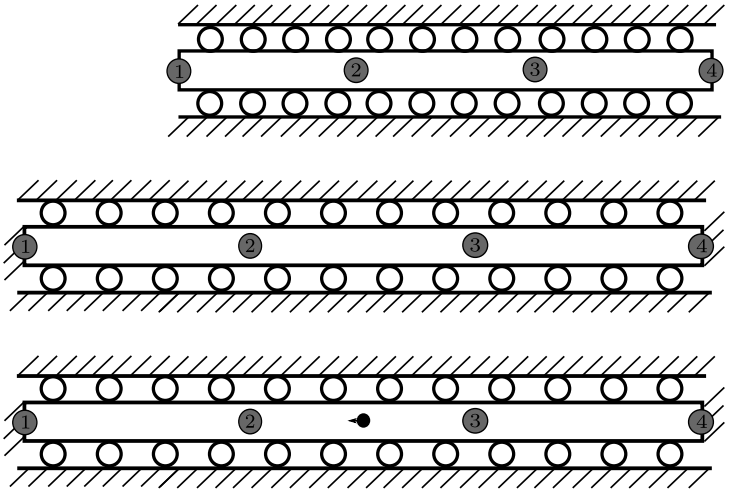


Figure 4.20: Configuration to test for tensile instability

At some time, a slight perturbation is introduced into the system by applying a five percent variation in the displacement of the centre particle. This increase in local displacement should trigger strain wave propagation, with all particle motion tracking that of the perturbation. The simulation is kept shorter than the time expected for this wave to reach the ends of the bar.

Figures 4.21 and 4.22 show the displacement of the particles that constitute the bar through time for both total and updated Lagrangian implementations. The middle seven particles are additionally tracked, enabling the behaviour to be more clearly identified.

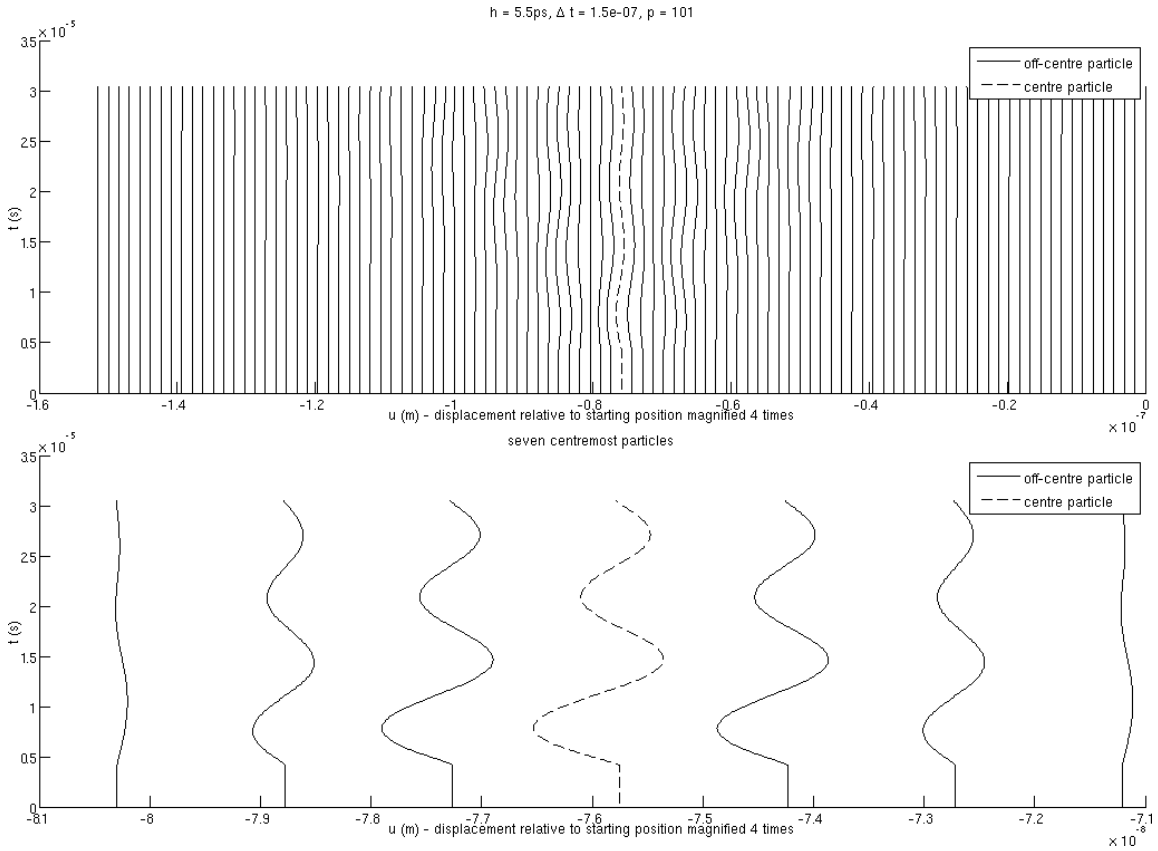


Figure 4.21: Linear Elastic tensile results - total Lagrangian implementation

The total Lagrangian implementation behaves as expected. Before the perturbation the system is static, and after the perturbation the particle displacement tracks the motion of the centre particle (where the perturbation was introduced). This is not the case in the updated Lagrangian implementation however, where adjacent particles move out of phase with each other. This results in particles which are under increasing tension moving closer together at times, which is clearly a-physical.

It is possible, for perturbations of increased duration, to get particles to pair together under tension. This particle "clumping" is well documented [3, 8, 33] and is one of the indications of the presence of the tensile instability, although in this case, the instability is not destructive. In this case, the large magnitude of the existing tensile stress ensures that the stress waves seen are near that expected.

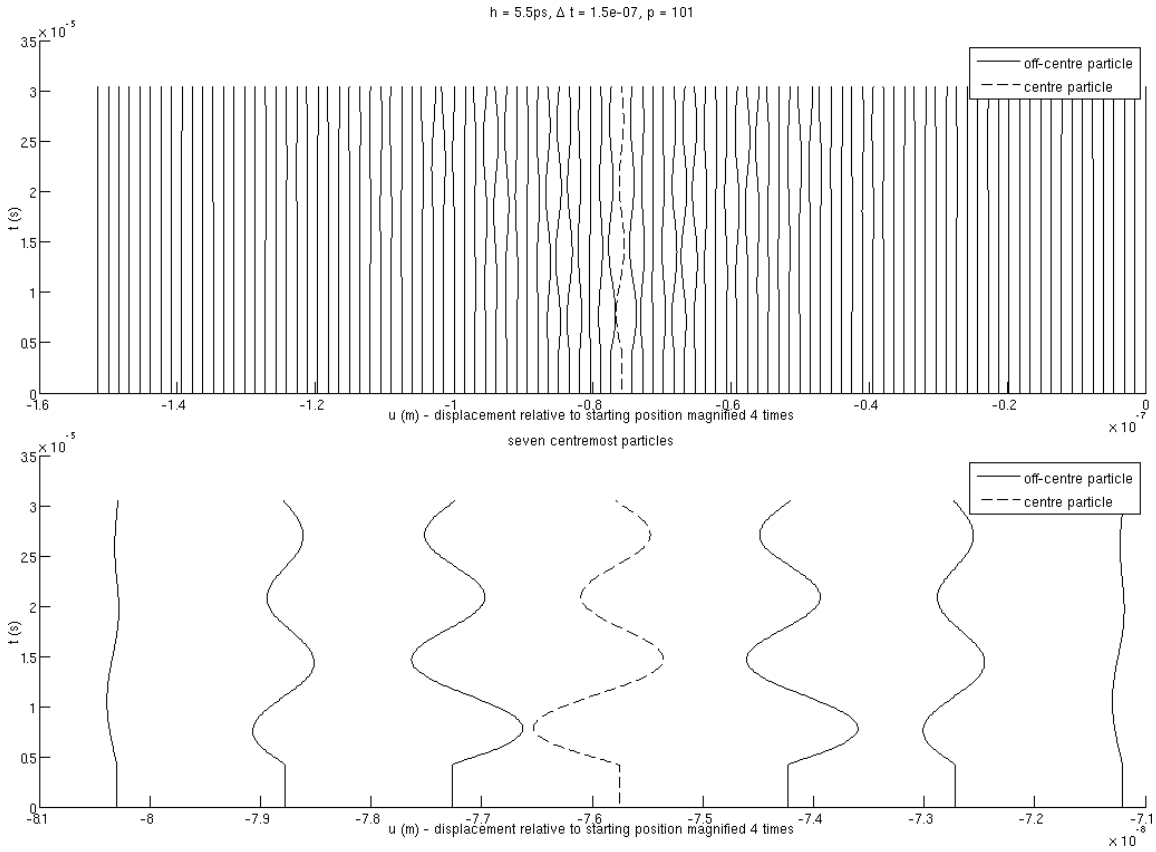


Figure 4.22: Linear elastic tensile results - updated Lagrangian implementation

Chapter 5

SPH and nonlinear elasticity

The implementation presented in Chapter 4 is for linear elastodynamics, where the SPH approximation is done in the original configuration, which is in line with a total Lagrangian methodology. This methodology appears to be less prone to tensile instability, despite the fact that there should be no appreciable difference between the total and updated Lagrangian methodologies for the infinitesimal displacement approximation.

In this chapter the SPH approach developed in Chapter 4 is extended to the case of nonlinear elastic finite deformation problems, as set out in Chapter 3.

5.1 Behaviour of nonlinear materials and code validation

The evaluation problem presented in Chapter 4 serves well because of the simplicity of the wave equations in linear elastic solids, especially in one dimension. Wave propagation becomes increasingly complicated with the addition of higher dimensions, which result in non-scalar motion, and with nonlinear material models.

Nonlinear material models add complexity because the local wave speed varies depending on the deformation of the material at the point in question. This results in complicated behaviour which, while tractable for some nonlinear models, is hard to use as a measure for performance.

The dynamic behaviour of the system can be evaluated for small displacements where the behaviour tends towards that expected for linear elastic materials. While such a validation test does not capture the full non-linearity of the material, it does ensure that the dynamic system performs as expected with the new material model.

In order to evaluate the nonlinear nature of the material one can avoid looking at the detail of a dynamic system by performing a *quasi-static* test. Here, the loading conditions are applied at a rate many times less than the natural frequency of the body being loaded. This ensures that the dynamic effects of the loading are small - relative to the overall behaviour. This technique was used for a validation problem that follows the work of Ogden in his presentation of the compressible Neo-Hookean material model [24].

5.1.1 Behaviour of Neo-Hookean materials

In his evaluation of material models for rubber-like materials Ogden shows curves of Cauchy stress relative to the principal stretches of the body [34], and compares these with experimental results. In this work, the behaviour is found directly from the strain energy function because it is defined in terms of the principal stretches. The normal Cauchy stress in the first direction is the only one presented for clarity. The strain energy function used for the SPH validation is defined in terms of the invariants of $\boldsymbol{C}$, the right Cauchy-Green tensor.

The stress relative to stretch is evaluated by applying a simple triaxial stretch, where the motion is defined to be only in the Cartesian directions, ensuring the principle stretch directions will be along these axes.

The deformation gradient is thus defined in terms of the principal stretches $\lambda_1, \lambda_2, \lambda_3$ as

$$\boldsymbol{F} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \quad . \tag{5.1}$$

The right Cauchy-Green tensor and the Jacobian are thus defined as

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{F}^T \boldsymbol{F} \\ &= \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \end{aligned} \tag{5.2}$$

and

$$J = \lambda_1 \lambda_2 \lambda_3 \quad . \tag{5.3}$$

The first Piola-Kirchhoff stress can then be found to be

$$\boldsymbol{P} = \mu \begin{bmatrix} \lambda_1 - 1/\lambda_1 & 0 & 0 \\ 0 & \lambda_2 - 1/\lambda_2 & 0 \\ 0 & 0 & \lambda_3 - 1/\lambda_3 \end{bmatrix} + \lambda \left(\ln \lambda_1 \lambda_2 \lambda_3\right) \begin{bmatrix} 1/\lambda_1 & 0 & 0 \\ 0 & 1/\lambda_2 & 0 \\ 0 & 0 & 1/\lambda_3 \end{bmatrix} \tag{5.4}$$

using equation (3.64). This can be converted to Cauchy stress using equation (3.33)

$$\boldsymbol{T} = J^{-1}\boldsymbol{P}\boldsymbol{F}^{T} \tag{3.33}$$

$$\boldsymbol{T} = \frac{\mu}{\lambda_1\lambda_2\lambda_3}\begin{bmatrix} \lambda_1^2 - 1 & 0 & 0 \\ 0 & \lambda_2^2 - 1 & 0 \\ 0 & 0 & \lambda_3^2 - 1 \end{bmatrix} + \frac{\lambda}{\lambda_1\lambda_2\lambda_3}\ln\left(\lambda_1\lambda_2\lambda_3\right)\boldsymbol{I} \quad . \tag{5.5}$$

To make the presentation of this relationship more tractable it is assumed that a plane strain condition is applied, where the displacement in the third dimension is held constant resulting in a stretch of $\lambda_3 = 1$.

A family of curves can then be generated by varying λ_1 while keeping λ_2 constant for each curve. In such a scenario the material is pre-stressed in the second dimension, and the normal Cauchy stress in the first direction is plotted as a function of the stretch in that direction. A family of such curves is presented for a compressible material in Figure 5.1 and for a nearly incompressible material in Figure 5.2.

By applying boundary conditions that control the displacement in the second and third dimensions a large strain quasi-static simulation can be used to validate the code against these results. This is done for the one-dimensional case in Section 5.3 and in two dimensions in Section 5.4.

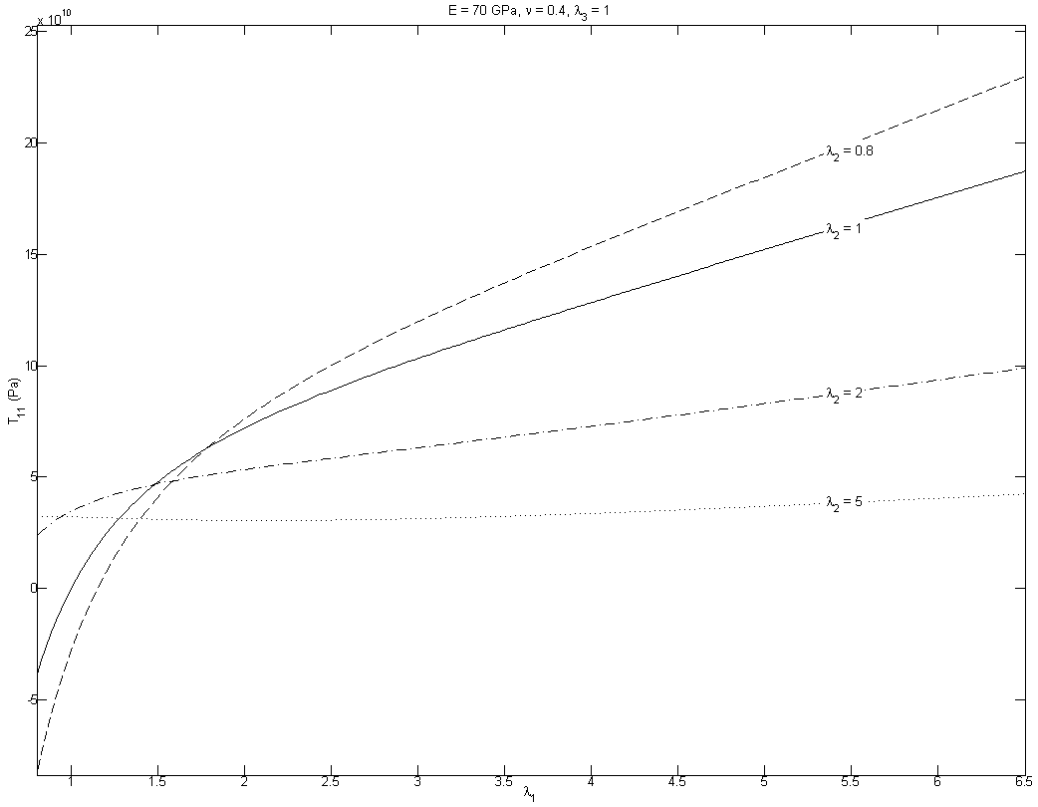


Figure 5.1: Behaviour of Compressible Neo-Hookean material

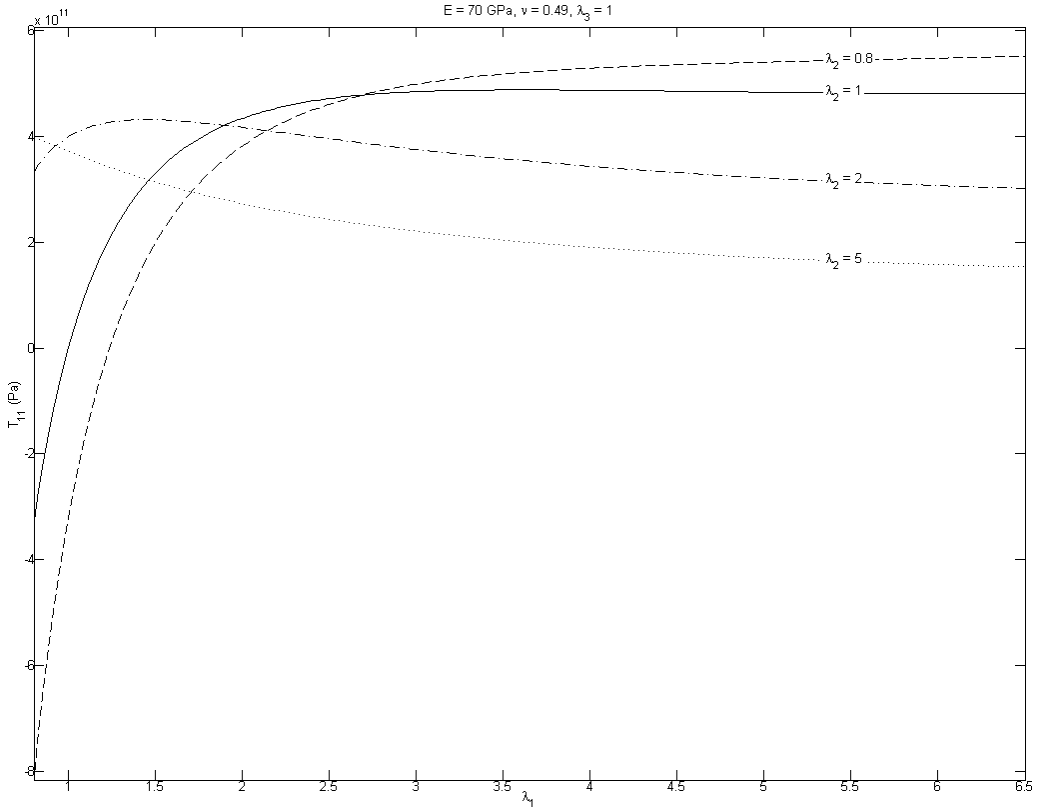


Figure 5.2: Behaviour of nearly incompressible Neo-Hookean material

5.2 SPH implementation

As in Section 4.4, the link between time steps in the dynamic simulation is the acceleration, which can be computed using equation (3.73) for the finite strain case. This can be re-arranged as

$$\frac{\partial \boldsymbol{v}}{\partial t} = \frac{Div\ \boldsymbol{P}}{\rho_0} + \boldsymbol{b}\ . \tag{5.6}$$

If the body forces $\boldsymbol{b}$ are assumed to be negligible the acceleration can be found from the divergence of the first Piola-Kirchhoff stress and the density in the original configuration. Because a total Lagrangian approach has been taken, the density at each particle does not vary. $\boldsymbol{P}$ is function of the deformation gradient, which itself is a function of displacement. By finding the SPH approximation to each inter-connected equation, an SPH approximation of the acceleration can be found.

The tensor mechanics making up these equations are computed component-wise as this facilitates vectorised coding in MATLAB. As a result, the gradient and divergence operations are described in terms of the directional derivatives of the variables. The directional derivative in the x_1 direction is shown, but the equivalent derivative in the other directions can be found with a suitable substitution.

We thus define the SPH approximation of the derivative of displacement in the x_1 direction as

$$< \frac{\partial \boldsymbol{u}}{\partial x_1} >_p = -\sum_{q=1}^{N} (\boldsymbol{u})_q \frac{\partial W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right)}{\partial x_1} \Delta V_q\ . \tag{5.7}$$

This approximation is then used to compute $\boldsymbol{F}$, its determinant J, and the right Cauchy-Green tensor $\boldsymbol{C}$. These are then used to compute the stress using equation (3.64). The divergence of this stress is then computed component-wise from the directional derivatives of the stress according to

$$< \frac{\partial \boldsymbol{P}}{\partial x_1} >_p = -\sum_{q=1}^{N} (\boldsymbol{P})_q \frac{\partial W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right)}{\partial x_1} \Delta V_q\ . \tag{5.8}$$

Finally the updated displacement of the particles can be computed using the same numerical scheme outlined in equation (4.4); that is

$$\boldsymbol{u}_{n+1} = \Delta t^2 \frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} + 2\boldsymbol{u}_n - \boldsymbol{u}_{n-1}\ . \tag{4.4}$$

5.2.0.1 Boundary conditions

Care must be taken in enforcing boundary conditions in nonlinear analysis, as the reference configuration must be used instead of the more "natural" current configuration. Thus, the first Piola-Kirchhoff stress must be used for Neumann boundary conditions. Dirichlet conditions must also be translated from the $\boldsymbol{x}$ to the $\boldsymbol{X}$ configuration. Similarly, if the results are required in the current configuration, a conversion must be performed.

Boundary conditions are dealt with in different ways in the one- and two-dimensional implementations of SPH. In one dimension the boundary conditions are enforced using the methodology presented in Section 2.8.3. This allows both the Dirichlet and Neumann boundary conditions to be set exactly.

This is not possible in higher dimensions (except in special cases), as each ghost particle affects more than one boundary particle. As discussed in Section 2.9, the Neumann conditions can only be implemented based on the secondary SPH approximation of the already computed gradients. This results in any applied Neumann boundary condition affecting the motion of a particle into the next time step, but not that particle's current displacement. With small enough time steps this effect is near negligible.

It is also important to note that Neumann boundary conditions must be set in terms of directional derivatives in the principal directions of the problem. Any traction must thus be resolved into its component form.

Once again, the algorithm implemented makes use of the modified mollifier function presented in Section 2.2.2.

5.3 1D SPH nonlinear elasticity

The one dimensional implementation of Section 5.2 can be arranged into the algorithm below. The performance of the algorithm is then assessed relative to the two validation tests discussed in Section 5.1.

Algorithm 5.1 below is for the one-dimensional case only, where the tensorial computations are reduced to a single dimension. The matrices $\mathbf{W}$ and $\mathbf{X}$ are the "SPH matrices", and define the SPH approximation of every particle to every other particle, as per Section 2.6. Any vector of values multiplied by one of these matrices finds the SPH approximation or SPH gradient approximation of these values in a vectorised manner.

Again, all values such as $\mathbf{u}$ and $\mathbf{P}$ should be read as the vector of these values for *every* particle unless indicated with a subscript denoting which particular particle is in question.

It is assumed that the left hand boundary is a Neumann boundary, and the right hand boundary is a Dirichlet one to allow for the inclusion of both types of boundary condition in the algorithm. The values D and N represent the denominators in equations (2.42) and (2.44) respectively and are used in the application of the Dirichlet and Neumann boundary conditions. Defining these values before the time stepping loop saves considerable computational time.

Algorithm 5.1: SPH nonlinear elasticity (Neo-Hookean)

. **Data**: Initial geometry, $\overline{\mathbf{u}}(t)$, $\overline{\mathbf{t}}(t)$, Δt

set ,$\mathbf{u}_0$, $\mathbf{u}_1$,$\dot{\mathbf{u}}_0$, $\dot{\mathbf{u}}_1$,$\mathbf{T}_0$and $\mathbf{T}_1$ for all p ; // set initial values

set ΔV for all p

Compute $\mathbf{W}$ and $\mathbf{X}$ // calculate SPH matrices

$$\mathsf{D} = \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left[W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right)\right]^2 \Delta V_q$$

$$\mathsf{N} = \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \nabla W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \Delta V_q$$

while $n < n\_total$ **do** // for each time step

  $\eta_{\text{left}} = \left(\overline{\mathbf{t}}(t) - \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left(\boldsymbol{u}(\boldsymbol{x}_q)\right) \nabla W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \Delta V_q\right) / \mathsf{N}$ // Γ_N

  $\eta_{\text{right}} = \left(\overline{\mathbf{u}}(t) - \sum_{q=1}^{\mathcal{B}+\mathcal{I}+\mathcal{G}} \left(\boldsymbol{u}(\boldsymbol{x}_q)\right) W\left(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{x}_q, h\right) \Delta V_q\right) / \mathsf{D}$ // Γ_D

  $\mathbf{u} = \mathbf{u} + \eta_{\text{right}} W\left(\boldsymbol{x}_{\Gamma_N} - \boldsymbol{x}_q, h\right) + \eta_{\text{left}} W\left(\boldsymbol{x}_{\Gamma_D} - \boldsymbol{x}_q, h\right)$ // set $\mathcal{G}$

  $< \dfrac{\partial \boldsymbol{u}}{\partial X} >= \mathbf{X}\boldsymbol{u}$ // calculate gradient

  $F =< \dfrac{\partial \boldsymbol{u}}{\partial X} > +1$

  $J = F$

  $C = F^2$

  $\mathbf{P}_{11} = \mu * (F - 1/F) + \lambda * \ln F/F$

  $< \dfrac{\partial \boldsymbol{P}_{11}}{\partial X} >= \mathbf{X}\mathbf{P}_{11}$ // calculate gradient

  $\mathbf{a} =< \dfrac{\partial \mathbf{P}_{11}}{\partial X} > /\rho$

  $\mathbf{u}_{n+1} = \Delta t^2 \mathbf{a} + 2\mathbf{u} - \mathbf{u}_{n+-1}$

  $n = n + 1$; // increment time step

end

return $\mathbf{u}$, $\mathbf{T}$ *for all* n

5.3.1 Dynamic validation

The dynamic validation followed a process similar to that followed in Section 4.6.3. The same test problems were used, and resulted in similar behaviour. This is to be expected as the Neo-Hookean material model behaves according to Hooke's law at small strains.

The validation problem was the same as that used in Section 4.6.3. The configuration is shown again in Figure 5.3 for convenience. The bar is set to be Neo-Hookean with material parameters

$$\rho = 8000 \ ,$$
$$E = 200 \ GPa \ ,$$
$$\nu = 0.3 \ .$$

If the material had been linear this would have resulted in a wave speed of

$$c_L = 5801 \ m.s^{-1} \ .$$

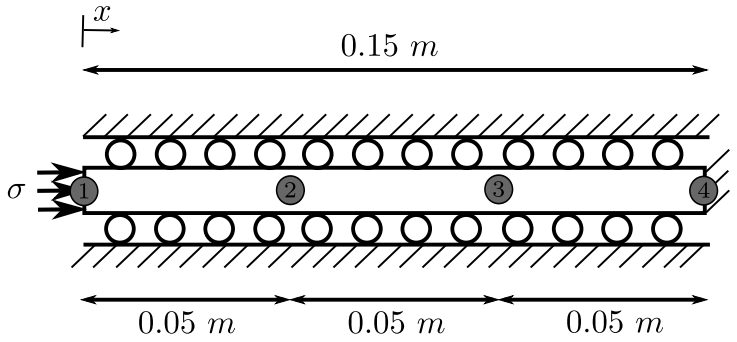


Figure 5.3: Validation problem - 1D strain

All the validation problems followed a similar structure to those in Section 4.6.3. The results were very similar, and a single test result is shown in Figure 5.4 as an example of the behaviour. In this problem the material is exposed to a compressive stress pulse.

5.3.2 Large strain validation

The ability of the implementation to accurately model large strain behaviour was tested by taking a $15mm$ bar of the material and elongating it to $90mm$. This was done over a period of 30 times the time needed for a strain wave to propagate through the material if it were undergoing small strains. It should be noted that the one-dimensional nature of this implementation results in the stretch in the second and third directions to be one.

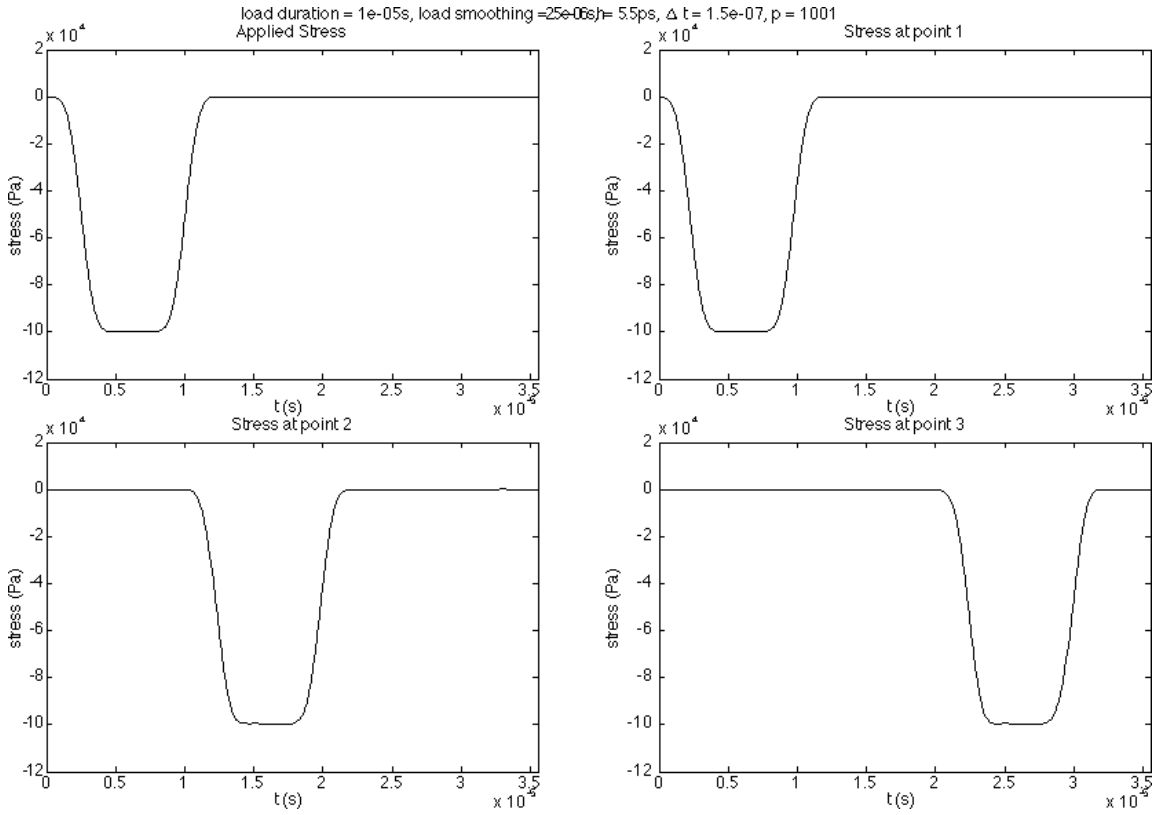


Figure 5.4: Wave propagation in Neo-Hookean material

The results shown in Figure 5.5 show some oscillation about the analytical curves for the static case. This is likely due to the transient portion of the loading. An increase in the loading time of the bar results in oscillations of a smaller magnitude.

If the Poisson's ratio of the bar is increased to 0.49, as in Figure 5.6 the behaviour still matches the exact solution, but the solution becomes unstable after a stretch of 3.6 is achieved. This is likely due to the high stresses resulting in accelerations beyond the capabilities of the time integration to capture although this has not been validated extensively.

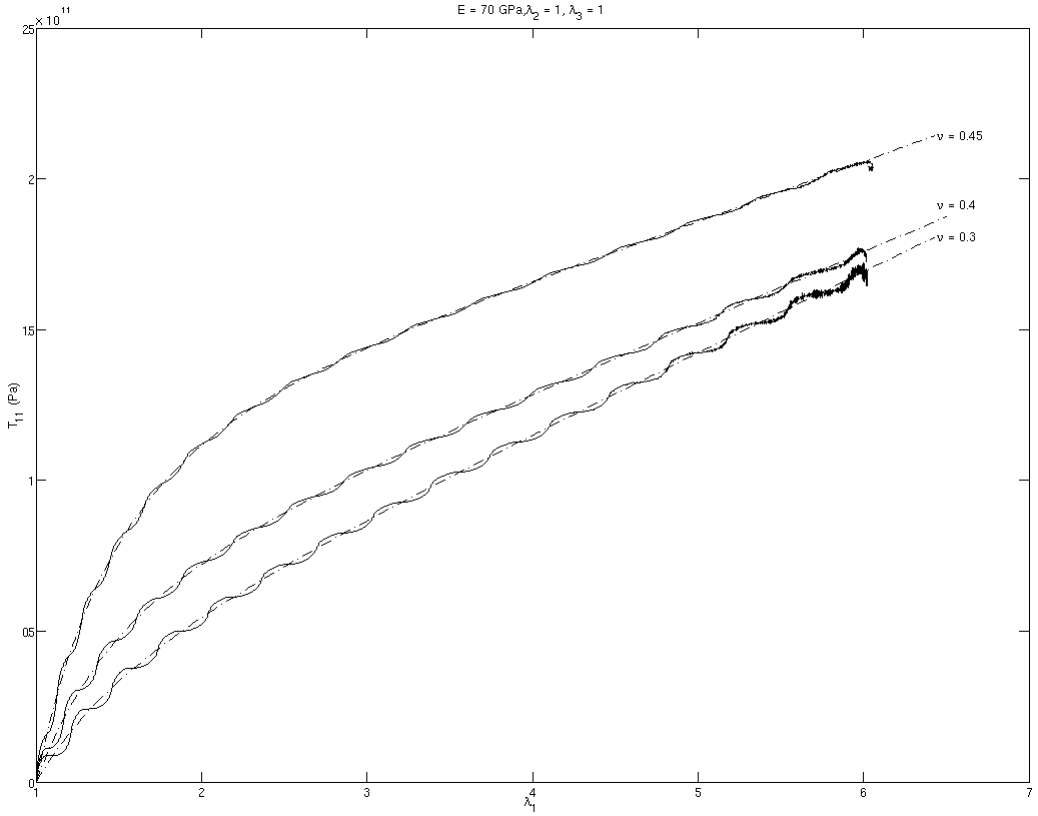


Figure 5.5: Large deformation of a slightly compressible material

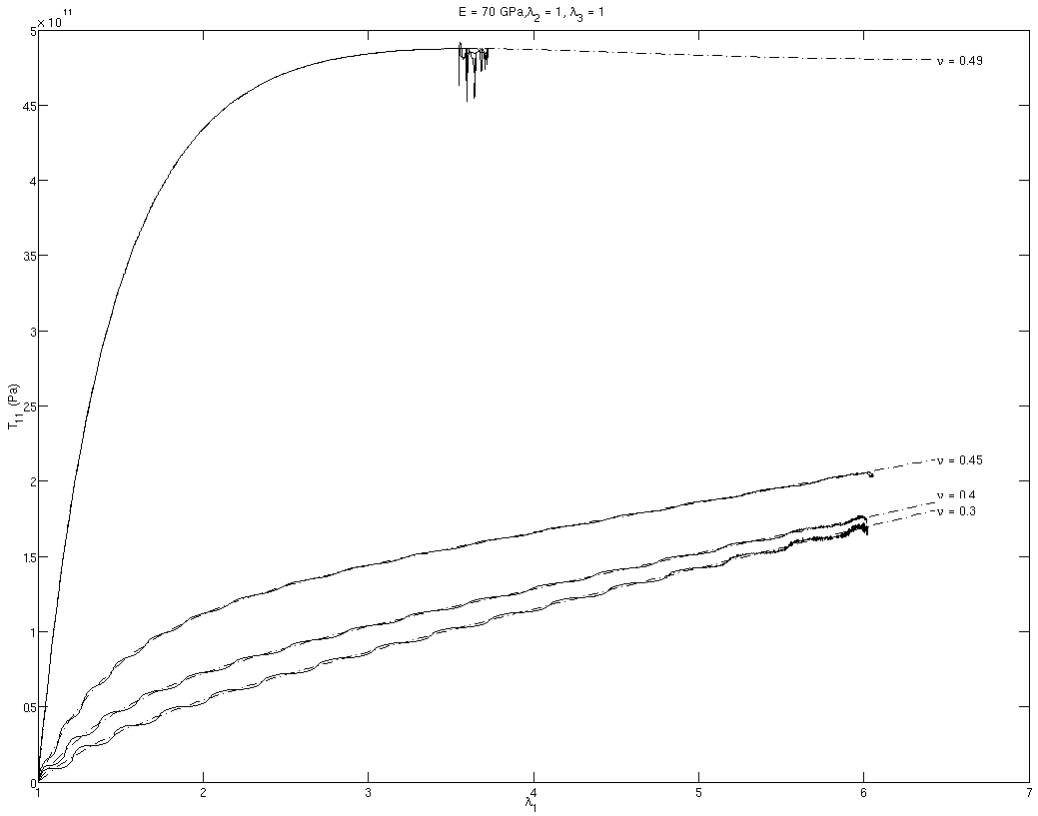


Figure 5.6: Large deformation of a nearly incompressible material - $\nu = 0.49$

5.4 2D SPH nonlinear elasticity

The implementation of two-dimensional SPH was done to confirm that the proposed methodologies work in higher dimensions, while still keeping the problem size tractable. The formulation is very similar to that for one dimension, except that the tensorial nature of the higher dimension problem must be considered. Additionally, the imposition of boundary conditions on the system must change, as the method used in the one-dimensional implementation does not work for all possible cases in higher dimension problems.

5.4.1 SPH implementation

The tensor algebra required in the implementation is done component-wise to facilitate the vectorised computation of SPH approximations, as discussed in Section 2.6. This allows an operation like Grad $\boldsymbol{u}$ to be computed from the directional derivatives

$$\begin{aligned}
\frac{\partial \mathbf{u}_1}{\partial x_1} &= \mathbf{X}\mathbf{u}_1 \\
\frac{\partial \mathbf{u}_1}{\partial x_2} &= \mathbf{X}\mathbf{u}_1 \\
\frac{\partial \mathbf{u}_2}{\partial x_1} &= \mathbf{X}\mathbf{u}_2 \\
\frac{\partial \mathbf{u}_2}{\partial x_2} &= \mathbf{X}\mathbf{u}_2
\end{aligned} \tag{5.9}$$

where $\mathbf{X}$ is the SPH directional derivative matrix to give the approximation of the derivative in the x_1 direction, and $\mathbf{Y}$ is the SPH directional derivative matrix to give the approximation of the derivative in the x_2 direction. $\mathbf{u}_1$ and $\mathbf{u}_2$ are the vectors describing the components of displacement for each particle.

5.4.1.1 Boundary conditions

Boundary conditions are imposed using the approach detailed in Section 2.9. Each type of boundary condition must have its own modified SPH matrix to allow for the control of the ghost particles associated with that type of boundary condition. This results in six such matrices, these corresponding to the application of the displacement in the two primary directions and the imposition of the gradient of either displacement in either direction.

The modified SPH matrices associated with Dirichlet boundaries are denoted $\mathbf{N}$, while those associated with Neumann boundary conditions with gradient in the $\boldsymbol{x}_1$ direction are

denoted $\mathbf{N}$, and those with gradient in the x_2 direction are denoted $\mathbf{M}$. The displacement on which they act is indicated with a † for the $\mathbf{u}_1$ displacement, and with a ‡ in the $\mathbf{u}_2$ displacement.

For instance: the modified SPH matrix associated with boundaries where the u_1 displacement is set is denoted $\mathbf{D}^{\dagger}$.

Each modified SPH matrix is created according to equation (2.46). All entries in the rows and columns of the SPH matrix are set to zero for all particles not included on the boundary in question (or its associated ghost particles), except for the one on the diagonal which is set to one. Rows and columns associated with the boundary region are left untouched.

These matrices and their inverses are ideally computed prior to the time-stepping portion of the code. The only time this is not done is if the regions in question vary in time. This method of pre-computation for the simulation of two-dimensional nonlinear elasticity is used in Algorithm 5.2.

Algorithm 5.2: SPH nonlinear elasticity (Neo-Hookean)

1 . **Data**: Initial geometry, $\overline{\mathbf{u}}(t)$, $\bar{\mathbf{t}}(t)$, Δt
2 set ,$\boldsymbol{u}_0$, $\boldsymbol{u}_1$,$\dot{\mathbf{u}}_0$, $\dot{\mathbf{u}}_1$,$\boldsymbol{P}_0$and $\boldsymbol{P}_1$ for all p ; // set initial values
3 set ΔV for all p
4 Compute $\mathbf{W}$, $\mathbf{X}$and $\mathbf{Y}$ // calculate SPH matrices
5 Compute $\mathbf{D}^{\dagger}$, $\mathbf{D}^{\ddagger}$,$\mathbf{N}^{\dagger}$, $\mathbf{N}^{\ddagger}$,$\mathbf{M}^{\dagger}$, $\mathbf{M}^{\ddagger}$
6 Compute $\mathbf{D}^{\dagger^{-1}}$, $\mathbf{D}^{\ddagger^{-1}}$,$\mathbf{N}^{\dagger^{-1}}$, $\mathbf{N}^{\ddagger^{-1}}$,$\mathbf{M}^{\dagger^{-1}}$, $\mathbf{M}^{\ddagger^{-1}}$
7 **while** $n < n\_total$ **do** // for each time step
8 $< \mathbf{u}_1 >= \mathbf{W}\mathbf{D}^{\dagger^{-1}} \left(\overline{\boldsymbol{u}_1} + \boldsymbol{u}_1^{\mathcal{I}}\right)$
9 $< \mathbf{u}_2 >= \mathbf{W}\mathbf{D}^{\ddagger^{-1}} \left(\overline{\boldsymbol{u}_1} + \boldsymbol{u}_2^{\mathcal{I}}\right)$
10 $< \dfrac{\partial \mathbf{u}_1}{\partial X_1} >= \mathbf{X}\mathbf{u}_1$
11 $< \dfrac{\partial \mathbf{u}_1}{\partial X_2} >= \mathbf{Y}\mathbf{u}_1$
12 $< \dfrac{\partial \mathbf{u}_2}{\partial X_1} >= \mathbf{X}\mathbf{u}_2$
13 $< \dfrac{\partial \mathbf{u}_2}{\partial X_2} >= \mathbf{Y}\mathbf{u}_2$
14 $< \dfrac{\partial \mathbf{u}_1}{\partial X_1} >= \mathbf{W}\mathbf{N}^{\dagger^{-1}} \left(\overline{\boldsymbol{t}_1} + < \dfrac{\partial \mathbf{u}_1}{\partial X_1} >^{\mathcal{I}}\right)$
15 $< \dfrac{\partial \mathbf{u}_1}{\partial X_2} >= \mathbf{W}\mathbf{M}^{\dagger^{-1}} \left(\overline{\boldsymbol{t}_1} + < \dfrac{\partial \mathbf{u}_1}{\partial X_2} >^{\mathcal{I}}\right)$
16 $< \dfrac{\partial \mathbf{u}_2}{\partial X_1} >= \mathbf{W}\mathbf{N}^{\dagger^{-1}} \left(\overline{\boldsymbol{t}_2} + < \dfrac{\partial \mathbf{u}_2}{\partial X_1} >^{\mathcal{I}}\right)$
17 $< \dfrac{\partial \mathbf{u}_2}{\partial X_2} >= \mathbf{W}\mathbf{M}^{\dagger^{-1}} \left(\overline{\boldsymbol{t}_2} + < \dfrac{\partial \mathbf{u}_2}{\partial X_2} >^{\mathcal{I}}\right)$
18 $\boldsymbol{F} = \text{Grad } \boldsymbol{u} + \boldsymbol{I}$ // component-wise
19 $J = \det \boldsymbol{F}$
20 $\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}$ // component-wise
21 $\boldsymbol{P} = \mu * (\boldsymbol{F} - \boldsymbol{F}\boldsymbol{C}^{-1}) + \lambda * \ln \text{J}\boldsymbol{F}\boldsymbol{C}^{-1}$ // component-wise
22 $< \dfrac{\partial \mathbf{P}_{11}}{\partial X_1} >= \mathbf{X}\mathbf{P}_{11}$
23 $< \dfrac{\partial \mathbf{P}_{12}}{\partial X_2} >= \mathbf{Y}\mathbf{P}_{12}$
24 $< \dfrac{\partial \mathbf{P}_{21}}{\partial X_1} >= \mathbf{X}\mathbf{P}_{21}$
25 $< \dfrac{\partial \mathbf{P}_{22}}{\partial X_2} >= \mathbf{Y}\mathbf{P}_{22}$
26 $\boldsymbol{a} = \text{Div } \boldsymbol{P}/\rho$ // component-wise
27 $\mathbf{u}_{n+1} = \Delta t^2 \mathbf{a} + 2\mathbf{u} - \mathbf{u}_{n+-1}$ // component-wise
28 $n = n + 1$; // increment time step
29 **end**
30 **return** $\boldsymbol{u}$, $\boldsymbol{T}$ *for all* n

5.4.2 Dynamic validation

Two-dimensional code allows for a test configuration that can either be in one-dimensional stress or strain, assuming uni-axial loading. If the sides of the bar are allowed to move in the transverse direction (and kept stress free), a state of one-dimensional stress is reached, but if motion is restricted in the transverse direction a state of one-dimensional strain is reached.

One-dimensional strain is easier to analyse as it follows the behaviour seen in the dynamic validation tests of the other implementations. If the same configuration as in Figure 5.3 is used, and the bar is subjected to a tensile pulse the response seen in Figure 5.7 follows the trends seen in the one-dimensional implementations. There is little difference through the transverse direction (a thickness of six particles), and this confirms the one-dimensional nature of the wave.

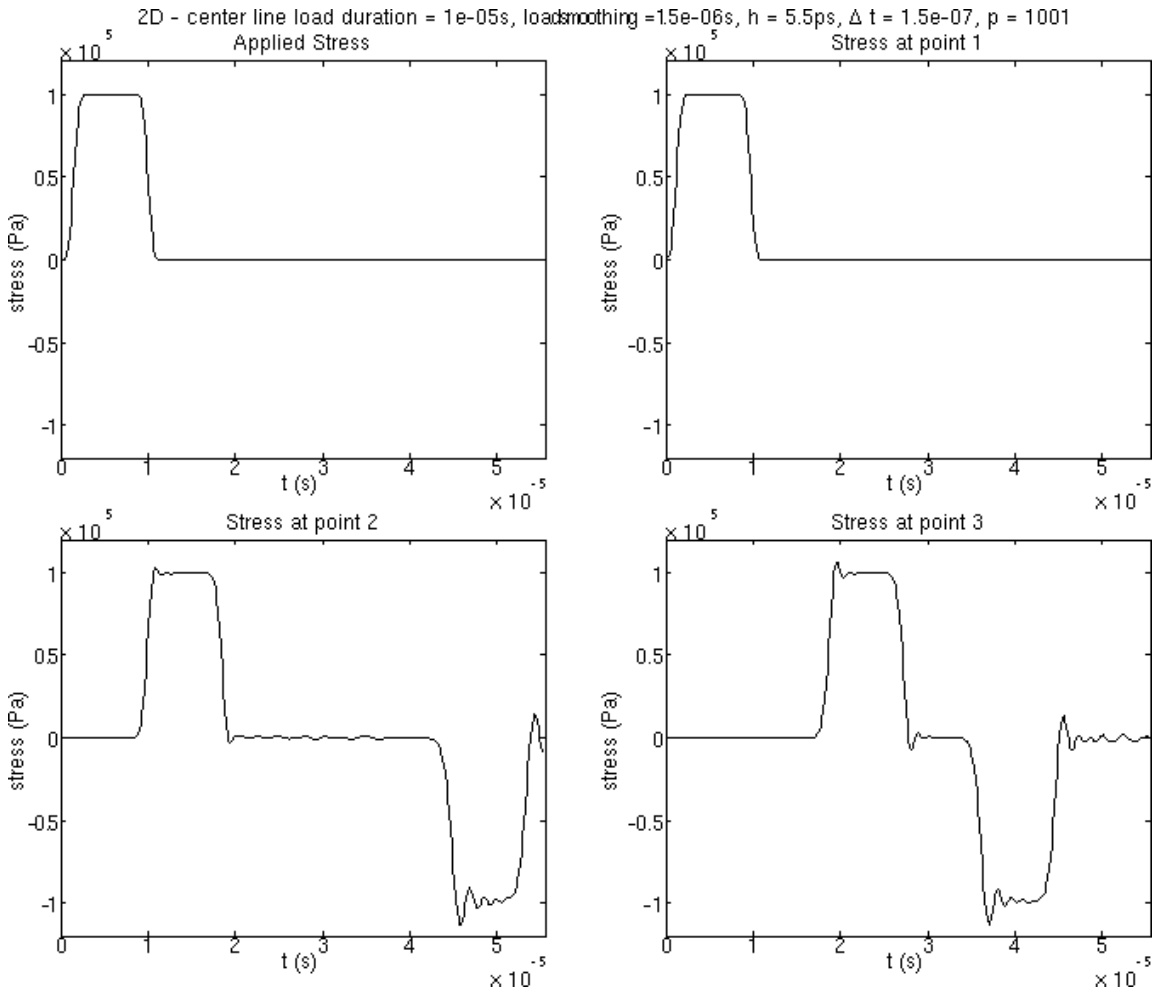


Figure 5.7: Tensile wave propagation in Neo-Hookean material - strain wave

If the bar is instead configured to be in a state of one-dimensional stress, as in Figure 5.8, the wave speed expected changes from c_L to c_b. In addition to this change in wave

speed, one-dimensional stress waves are prone to dispersion [28]. The short duration of the validation test, shown in Figure 5.9, results in no noticeable dispersion, although a detailed study of the dispersion characteristics would be beneficial to understanding the behaviour of the SPH approximation.

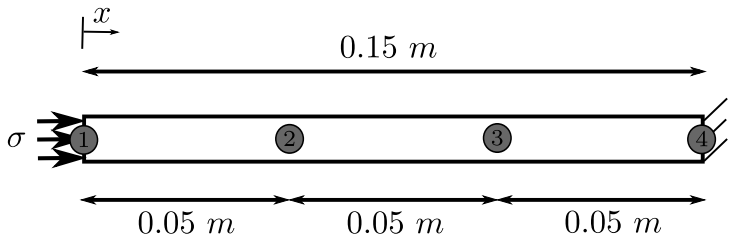


Figure 5.8: Validation problem - 1D stress

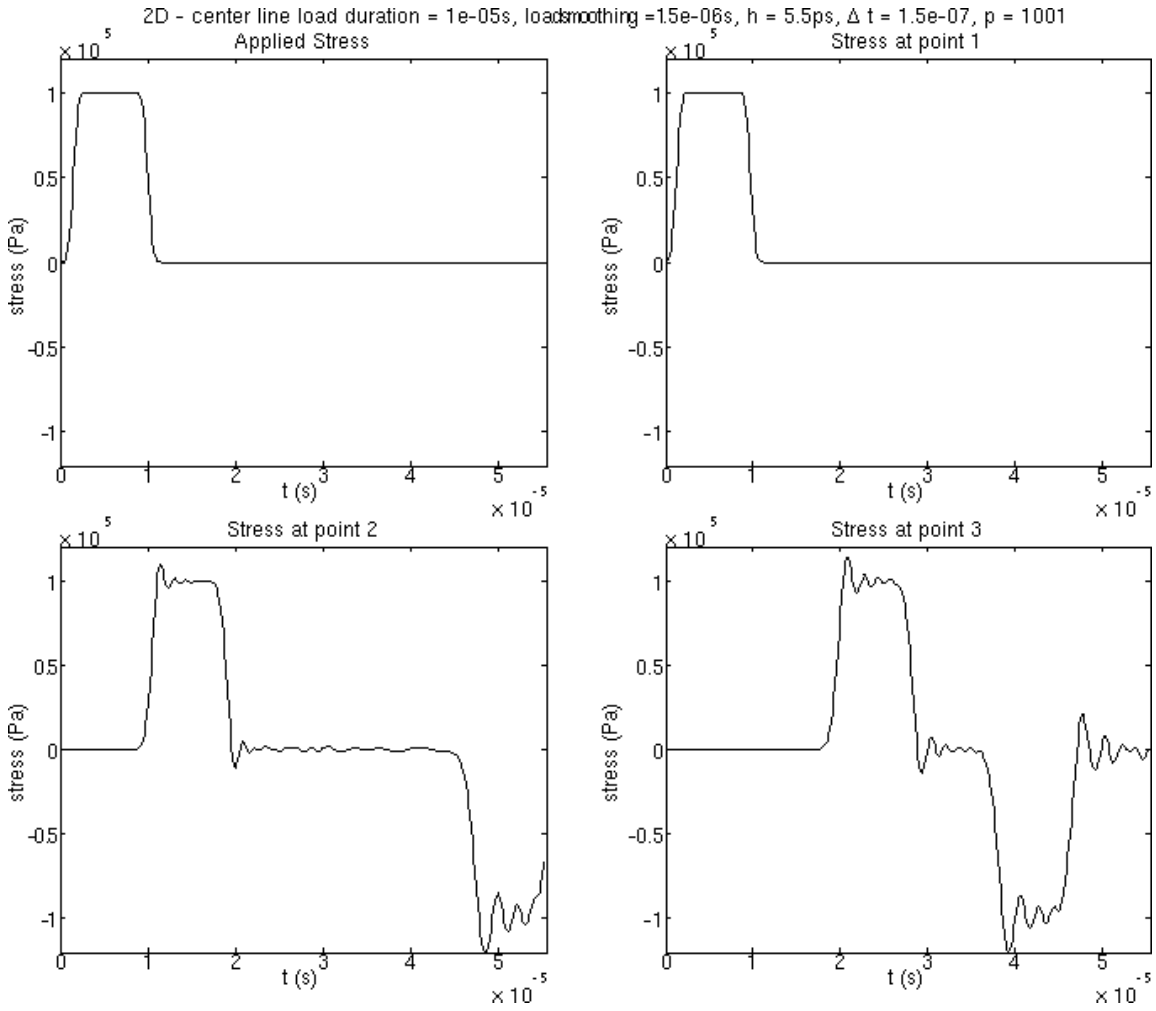


Figure 5.9: Tensile wave propagation in Neo-Hookean material - stress wave

5.4.3 Large strain validation

In order to keep computational times to a manageable level the large strain validation was done on a very small mesh. This reduced mesh size greatly increases the fundamental frequency of the test domain, allowing the deformation to occur over a relatively short duration. The test domain was defined to be 31 particles in length and six in the transverse direction.

Three test cases are presented. The first fixes the stretch in the transverse direction at one (Figure 5.10). The second fixes the stretch in the transverse direction at two, a test possible now that two dimensions are available (Figure 5.11). The final test starts with the test domain undeformed, with the stretch in the primary direction increased linearly while also linearly increasing the stretch in the transverse direction so that it has a stretch of two when the first stretch is six (Figure 5.12). Here the exact solution for the entire path has not been computed but the expected solution should be bounded by the analytical curves for $\lambda_2 = 1$ and $\lambda_2 = 2$, starting at the former when $\lambda_1 = 1$ and being coincident with the latter at the end of the simulation.

The curves presented are for the particle in the centre of the domain. It is possible for this particle to not achieve the stretch imposed on the boundary if equilibrium is not fully attained. This can be seen in Figure 5.12 where the test curve does not end coincident with the curve for $\lambda_2 = 2$.

Interestingly for all the results presented the curve deviates slightly from the analytical solution. Most notably, the stretch in the transverse direction does not reach the value expected in the third test case. It is likely that this error is an artefact of the error introduced into the system by only having six particles in the transverse direction. Another possibility is that the time integration scheme does not perform well with such large deformations. The scheme used becomes unstable if the deformation of the domain is so extreme the current configuration results in local "re-ordering" of particles in the current configuration.

Despite this limitation, the SPH implementation approximates the behaviour of the material well, even at extreme stretches. This performance is impressive considering that finite element models fail without re-meshing due to distortion of elements. It is possible that some amount of "distortion error" has manifested in the simulation, resulting in the result moving away from the exact solution.

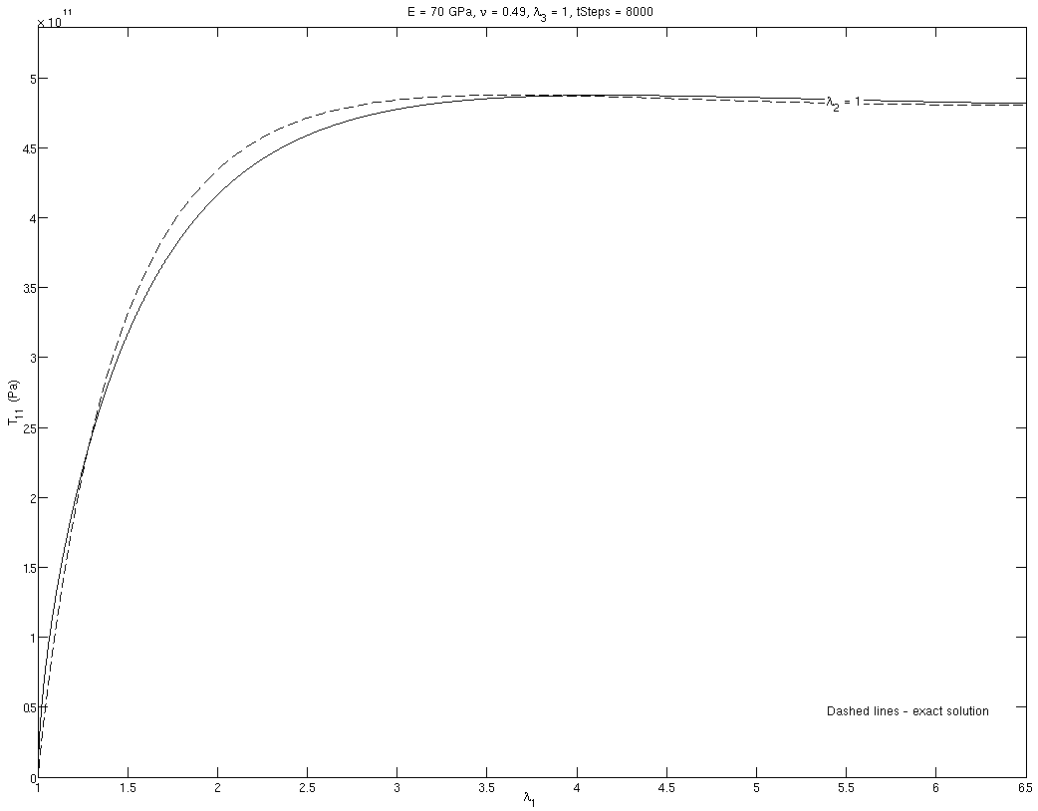


Figure 5.10: Large deformation - $\lambda_2 = 1$

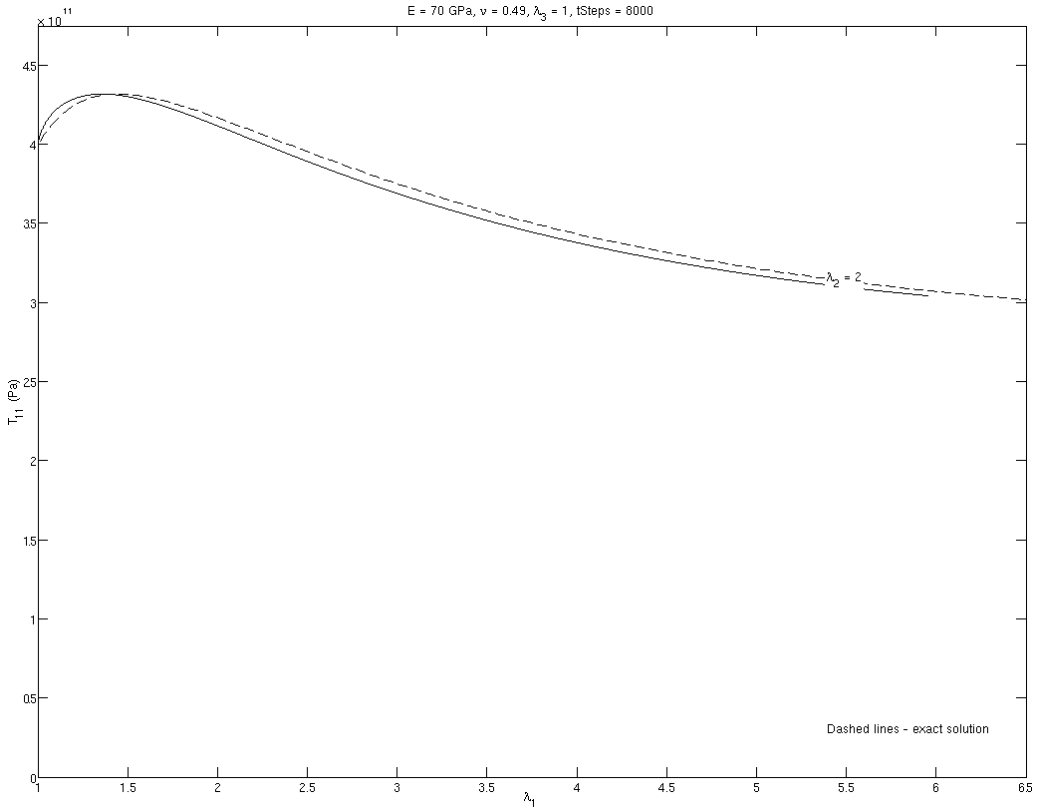


Figure 5.11: Large deformation - $\lambda_2 = 2$

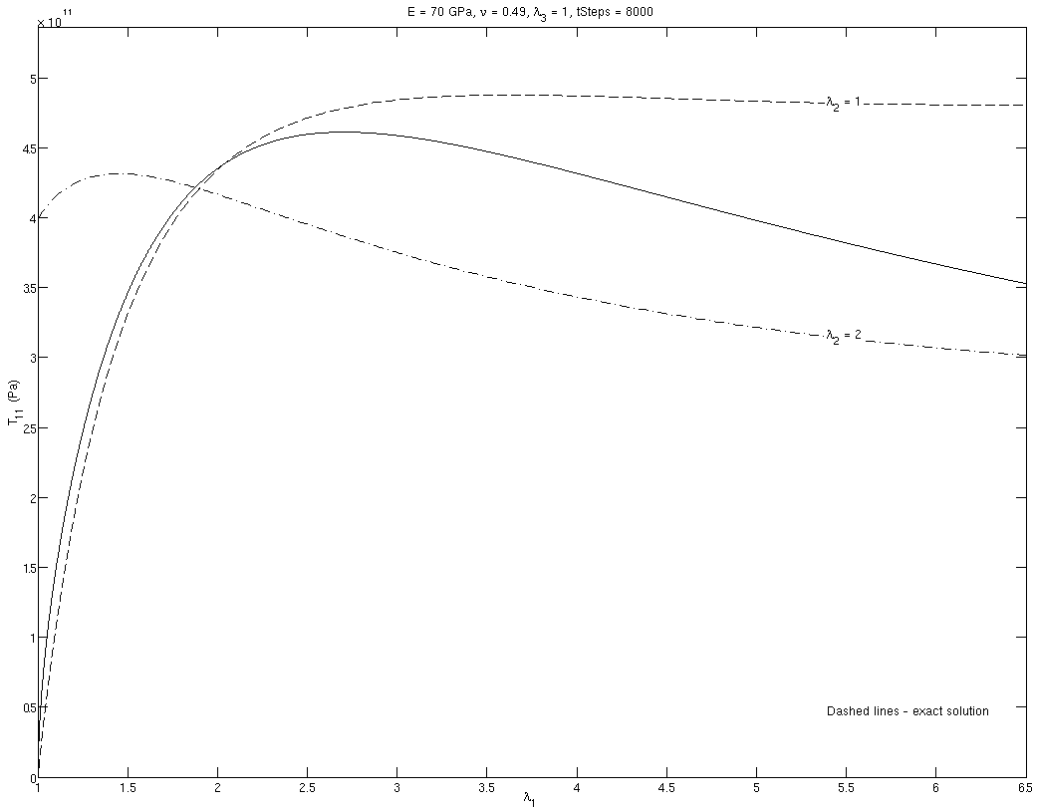


Figure 5.12: Large deformation - λ_2 linearly increased from $\lambda_2 = 1$ to $\lambda_2 = 2$

5.4.4 Complex geometry

While the tests cases presented above serve to confirm the implementation behaves as expected, numerical methods are seldom needed on such simple problems. As an example of a more complex problem that can be solved using SPH an additional problem is included.

This problem is not compared to existing results, but is instead discussed after the computation, as would happen with any mature technique. The problem posed is simple, but consists of a curved, stress-free boundary, which should serve to test the code well.

A sheet with a circular hole in it is subjected to uniaxial tensile loading and the stress distribution in the sheet is desired. This problem can be approximated using quarter symmetry as shown in Figure 5.13.

To facilitate reasonable computational times, a rectangular mesh of 21 by 21 particles is used, where the particles within the "hole" region are set to be ghost particles. The sheet is loaded dynamically from rest with a traction equivalent to $5MPa$.

Figure 5.14 shows several contour "snapshots" of the dynamic loading process, where the advancing wave can clearly be seen interacting with the hole. Finally, the system comes to rest with a stress concentration at the edge of the hole. This stress concentration results

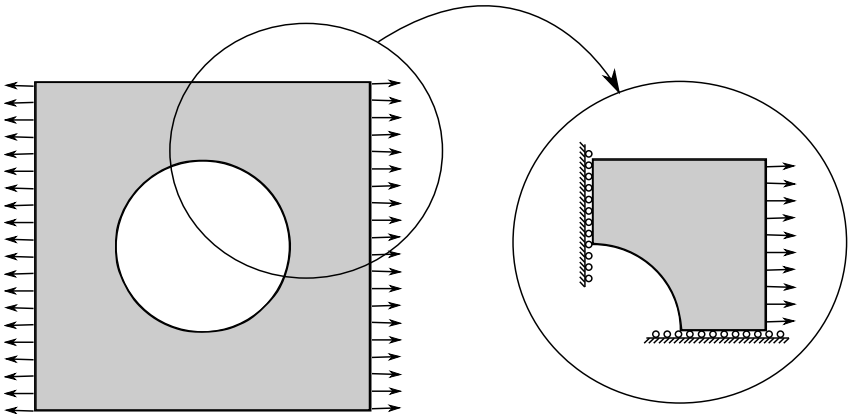


Figure 5.13: Sheet with hole in middle. $E = 70\,MPa,\ \nu = 0.49$

in a local maximum principal stress of $60MPa$. These results appear reasonable, although a convergence analysis with increasing numbers of particles should be completed. The results are also qualitatively similar to those obtained using linear elastic finite elements, as in [35], with the stress concentration developing in the correct place, and being of the correct order of magnitude.

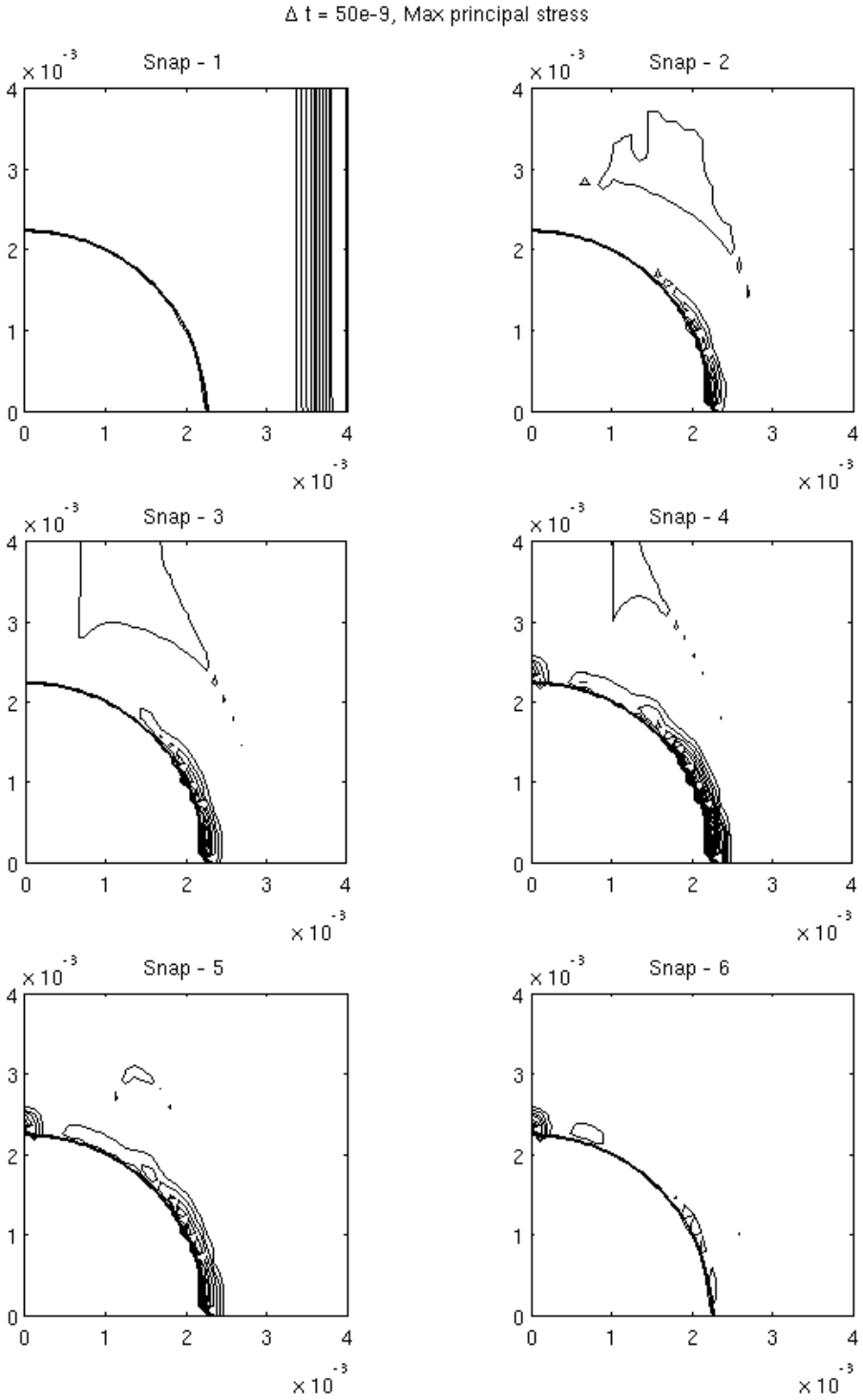


Figure 5.14: Propagation of stress through sheet with hole

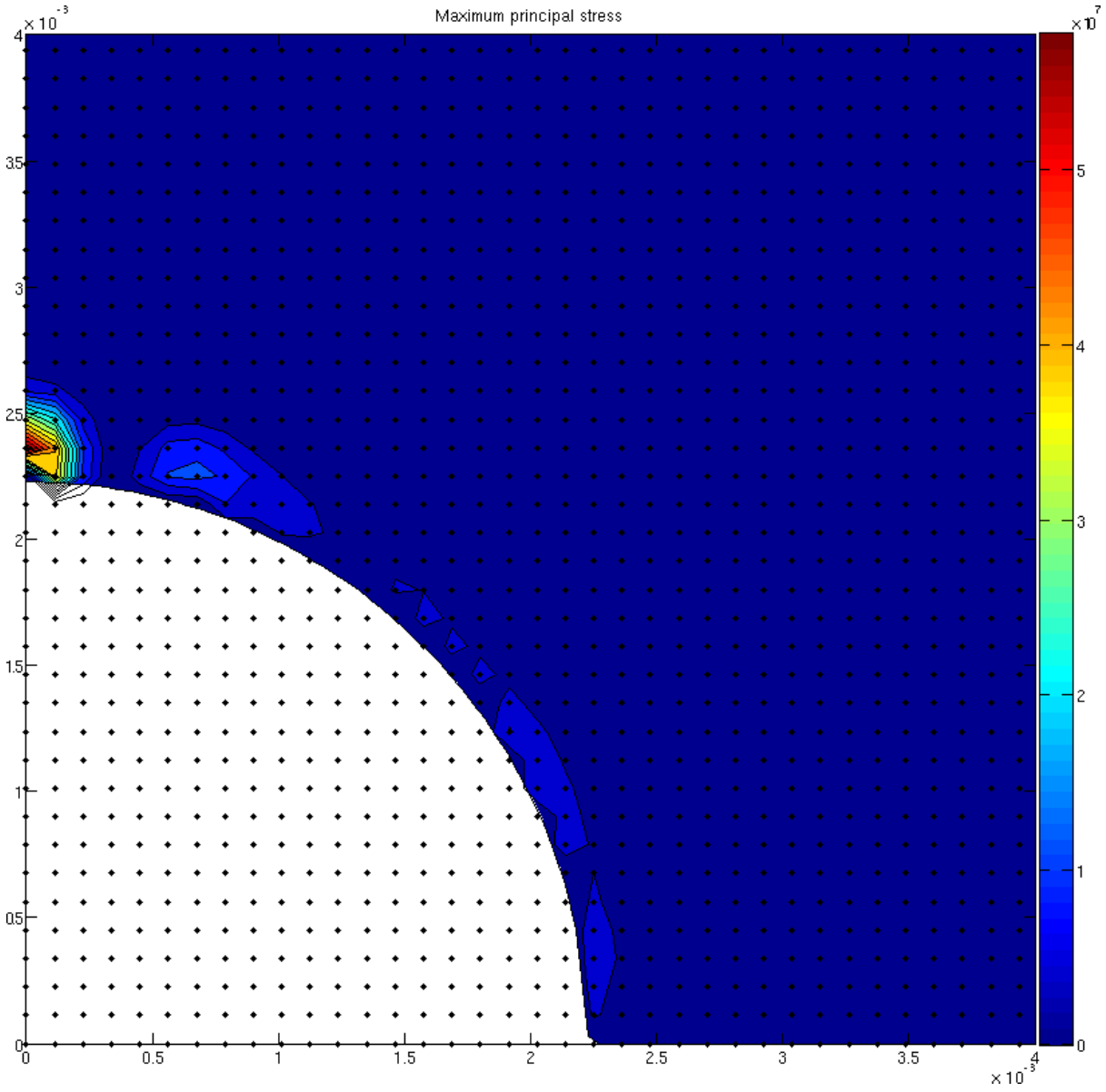


Figure 5.15: Final stress distribution - sheet with hole

Chapter 6

Conclusions and Recommendations

This thesis has focused on the development of the SPH technique. The findings of the thesis are discussed as a summary of the investigations undertaken. These findings are then used to define recommendations for future work.

6.1 The SPH formulation

The development of the SPH method was completed from fundamental assumptions. This allowed for the arrangement of the SPH approximation at every particle to be described as a matrix operation. While it is likely that such an approach has been taken before, the SPH literature typically only provides the approximation at each individual particle. The matrix arrangement is key for an efficient implementation of SPH and should generally be exploited.

Several cover functions were discussed, including a new function, the modified mollifier, that has both compact support and is infinitely differentiable. These are highly valued properties, and the function was used in all the SPH implementations.

Interpolation

The SPH approximation of functions is key to the technique's success. The performance of the technique was evaluated by using SPH to interpolate known functions. This interpolation was found to be acceptable, provided care is taken in the selection of the cover function, cover size, and particle distribution. The performance of the approximation is poor near the discretisation boundary, a feature which must be considered when the method is used for other applications.

6.2 Boundary value problems

The primary use of the SPH approximations is for solving boundary value problems. This is possible as approximations to functions and their derivatives, are directly computed from the position of the particles.

Boundary conditions

Key to solving these problems is the application of boundary values. This is an issue, however, as the classic SPH approximation degrades near the boundary.

Several techniques have been presented to handle near-boundary approximations. Two variations on the classical ghost particle approach are presented. These approaches are used in the SPH implementation presented and appear to function as expected. In one-dimensional problems both the Neumann and Dirichlet conditions can be exactly enforced, while in higher dimensions the Neumann condition can be enforced in a secondary SPH approximation of the gradients.

6.3 SPH for solid mechanics

The SPH approximation was then used to successfully model both linear and non-linear materials. The mathematical description of the problems at hand were discussed in detail, allowing for the SPH approximation of these mathematical descriptions.

The approximations were implemented for one-dimensional linear elastodynamics and both one and two-dimensional nonlinear elastodynamics. These approximations were done in an explicit manner to facilitate ease of computation.

Tensile instability

Evidence of the tensile instability highly commented on in the literature was not present in the direct implementation of the approximations. It was determined that this may have been because the implementation followed the total Lagrangian formulation which is believed to remove the instability.

Validation

The code was validated against one-dimensional wave theory. Generally the implementation behaved as expected, albeit with some numerical noise present. Techniques for dealing with the numerical noise were not investigated as it was determined to be outside of the scope of this project.

Additionally, the nonlinear behaviour was tested quasi-statically against that predicted for a known displacement field. Here the results were good, especially in the one-dimensional case. The two-dimensional case suffered from some accumulation of error, perhaps introduced by large strains of up to 700%.

Finally, a single two-dimensional example problem, albeit on a small scale, but with curved boundaries and mixed boundary conditions was presented to show the capabilities of the method.

6.4 Recommendations

Based on the investigation presented, several recommendations can be made. These are separated into recommendations regarding the work done in this project and into general recommendations regarding the use of SPH.

6.4.1 Future work

The validation of the implementations and the use thereof was limited in this project. This was primarily due to the extensive memory usage and thus computational time encountered. This can most directly be attributed to the choice of MATLAB as the implementation tool. MATLAB is suitable for small problems to be used in methodology development, but rapidly degrades for large problems. Any detailed investigation into the algorithms presented must be done using code written specifically to deal with the large matrices created. It is important to note that the SPH matrices used are the size of the number of particles squared. This leads to rapidly ballooning memory demands, and must be accounted for.

Investigation into how the memory demands might be reduced should also be performed. This will likely be found in the exploitation of the compact support of the cover functions. Smaller SPH matrices that are representative of the larger problem will allow reduced memory use and potential for direct parallelisation.

Because of the structure of the algorithm, all matrix operations in the dynamic code are multiplications. This standard use of the explicit methodology can be used for parallelisation even with the SPH matrix representing the entire domain.

A detailed characterisation of the algorithm has been performed, but not for randomly distributed particles,. This must be undertaken before the code is generally used. This is especially noted in the lack of a formal stability analysis and a convergence analysis. Additionally, it was not possible to test every conceivable boundary configuration. Improved methods for applying boundary conditions must be sought, and care must be taken that the boundary conditions are indeed satisfied.

While analysis into the energy and momentum conservation of the dynamic simulations was done in passing, a dedicated analysis must still be undertaken. This is essential to understanding how errors propagate into the system.

Artificial viscosity and other damping methods should be considered to aid the stability of the dynamic simulations. Methods like upwinding may be possible through the use of non-symmetric cover functions.

6.4.2 SPH in nonlinear mechanics

The problems discussed in this project are a long way from those for which meshless methods are suited. In general there is no particle re-ordering to contend with, and the constitutive models are fairly simple. The complexity of an SPH implementation will likely increase dramatically with more complex models. For the validation test presented finite elements presents a more rigorous approach, with similar, or better performance.

The non-local nature of SPH may lend itself well to non-local formulations and the modified mollifier may find some use in this field, either in an SPH approximation or with some other meshless method. Investigation into how portions of the method may be integrated into other methodologies must be ongoing.

6.5 A few final words

As a basic introduction to meshless methods this project has been a success. A fairly robust algorithm has been created, and can be used as the building blocks for further research into meshless methods. Until a fully rigorous treatment of SPH or one of its derivatives is found, the search for better schemes must continue.

To facilitate ease of use as a first step in understanding meshless methods, sample code has been provided on the included compact disc. The code chosen was that used to generate some of the results presented in this project.

References

[1] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. Technical report, Northwestern University, 1996.

[2] T. Rabczuk, T. Belytschko, and S.P. Xiao. Stable particle methods based on Lagrangian kernels. *Computer Methods in Applied Mechanics and Engineering*, 193:1421–1444, 2004.

[3] Rade Vignjevic, Juan R. Reveles, and James Campbell. SPH in a total Lagrangian formalism. *Computer Modelling in Engineering and Sciences*, 14(3):181–198, 2006.

[4] J.W. Swegle, S.W. Attaway, M.W. Heinstein, F.J. Mello, and D.L. Hicks. An analysis of smoothed particle hydrodynamics. Technical Report SAND93-2513, SANDIA National Labs, 1994.

[5] T.E. De Vuyst, R. Vignjevic, and J.C. Campbell. Coupling between meshless and finite element methods. *International Journal of Impact Engineering*, 31:1054–1064, 2005.

[6] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astrophysical Journal*, 82:1013–1024, 1977.

[7] G.M. Zhang and R.C. Batra. Modified smoothed particle hydrodynamics method and its application to transient problems. *Computational Mechanics*, 34:37–146, 2004.

[8] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics*. World Scientific, 2003.

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

[10] Z. Wang, Y. Lu, H. Hao, and K. Chong. A fully coupled numerical analysis approach for buried structures subjected to subsurface blast. *Computers and Structures*, 83:339–356, 2005.

[11] G.R. Johnson. Linking of Lagrangian particle methods to standard finite element methods for high velocity impact computations. *Nuclear Engineering and Design*, 150:265–274, 1994.

[12] G.R. Johnson, R.A. Stryk, and S.R. Beissel. SPH for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering*, 139:347–373, 1996.

[13] S. Hiermaier, D. Konke, A.J. Stilp, and K. Thoma. Computational simulation of the hypervelocity impact of *al*-spheres on thin plates of different materials. *International Journal of Impact Engineering*, 20:363–374, 1997.

[14] J.J. Monaghan and J.C. Lattanzio. A refined particle method for astrophysical problems. *Astronomy and Astrophysics*, 82:1–15, 1985.

[15] B. D. Reddy. *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Elements*. Springer, 1998.

[16] Shaofan Li and Wing Kam Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55:1 – 34, January 2002.

[17] N. J. Quinlan, M Basa, and M. Lastiwka. Truncation error in mesh-free particle methods. *International Journal for Numerical Methods in Engineering*, 66:2064–2085, 2006.

[18] J.K. Chen, J.E. Beraun, and C.J. Jih. An improvement for tensile instability in smoothed particle hydrodynamics. *Computational Mechanics*, 23:279–287, 1999.

[19] M Basa, N. J. Quinlan, and M. Lastiwka. Robustness and accuracy of SPH formulations for viscous flow. *International Journal for Numerical Methods in Fluids*, 60(1127):1148, 2009.

[20] W.G. Hoover. *Smoothed Particle Applied Mechanics - The State of the Art*. World Scientific, 2006.

[21] J. Bonet and R.D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge university press, 1997.

[22] J. Bonet and S. Kulasegaram. A simplifed approach to enhance the performance of smooth particle hydrodynamics methods. *Applied Mathematics and Computation*, 126:133–155, 2002.

[23] R. W. Ogden. Large deformation isotropic elasticity - on the correlation of theory and experiment for incompressible rubberlike solids. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 326(1567):565–584, 1972.

[24] R. W. Ogden. Large deformation isotropic elasticity - on the correlation of theory and experiment for compressible rubberlike solids. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 328(1575):567–583, 1972.

[25] R. W. Ogden. Recent advances in the phenomenological theory of rubber elasticits. *Rubber Chemistry and Technology*, 59:361–383, 1986.

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

[27] R.J. Wasley. *Stress Wave Propagation in Solids, an Introduction.* Marcel Dekker, 1973.

[28] J.D. Achenbach. *Wave Propagation in Elastic Solids.* North-Holland publishing Company, 1973.

[29] D. P. Balsara. von Neumann stability analysis of smoothed particle hydrodynamics - suggestions for optimal algorithms. *Journal of Computational Physics*, 121:357–372, 1995.

[30] C. Guenther, D. L. Hicks, and J. W. Swegle. Conservative smoothing versus aritificial viscocity. Technical Report SAND94-1853, SANDIA National Labs, 1994.

[31] Y. Wen, D. L. Hicks, and J. W. Swegle. Stabilising SPH with conservative smoothing. Technical report, SANDIA National Labs, 1994.

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

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

[34] R. W. Ogden. *Non-Linear Elastic Deformations.* Dover, 1997.

[35] J. Fish and T. Belytschko. *A First Course in Finite Elements.* Wiley, 2008.

Seventh South African Conference on Computational and Applied Mechanics
SACAM10
Pretoria, 10−13 January 2010

SMOOTHED PARTICLE HYDRODYNAMICS FOR NONLINEAR SOLID MECHANICS

Ernesto B. Ismail [*,†,‡,1] **and B. Daya Reddy**[†,2]

[*]Department of Mechanical Engineering
[‡]Blast Impact and Survivability Research Unit (BISRU)
[†]Centre for Research in Computational and Applied Mechanics (CERECAM)
University of Cape Town, Private Bag X3, Rondebosch, Cape Town, 7701, South Africa,
[1]ernesto.ismail@uct.ac.za, [2]daya.reddy@uct.ac.za.

Keywords: SPH, Smoothed Particle Hydrodynamics, elastodynamics, modified mollifier

Abstract

SPH is used here to approximate the solution to the equations governing nonlinear elastodynamics in one and two dimensions. The SPH implementation is based on a total Lagrangian formulation for reasons of stability [1]. The standard SPH method is discussed, and the discrete SPH approximation is presented in matrix form. This approach allows for the vectorisation of the method, greatly improving computational efficiency.

A new cover function - the modified mollifier - is introduced and used in computations. This has compact support and is infinitely differentiable, allowing it to be applied to higher-order problems.

A methodology for the imposition of both Neumann and Dirichlet boundary conditions is developed. This methodology works exactly for both conditions in one dimension, but Neumann boundary conditions can only be imposed approximately in higher dimensions.

Validation of the methodology presented is done through comparison with results in the literature, as well as with one-dimensional wave propagation theory. Special attention is given to problems involving time-dependent loading conditions. Additionally, a two-dimensional example problem with curved boundaries and mixed boundary conditions is presented to show the capabilities of the method.

1 Introduction

Nearly all complex problems in elastodynamics are currently solved computationally, through the use of some numerical approach. For standard elasticity problems, as well as a host of other, more complex problems, the finite element method (FEM) is the current tool of choice. FEM relies on a fixed discretisation of the material, with a domain being broken up into elements having nodes at least at their vertices. A shortcoming in the modelling of fluids or of solids that undergo large deformations is that the discretised ‘mesh’ quickly becomes distorted, resulting

in poor field approximations [2]. Dynamic re-meshing of highly distorted elements is currently implemented by FEM users to circumvent these problems.

Meshless methods aim to avoid the poor approximations associated with mesh distortion by removing the interconnection of nodes. A domain is discretised into nodes, but no strict connections are enforced. The integrity of the domain is maintained through the application of some interrelation of the nodes. No difficulties are experienced due to mesh distortion, although there is a computational cost associated with the interrelation rule.

Smoothed Particle Hydrodynamics is a fairly simple example of a meshless method. It was first developed in the late 1970s to model astrophysical problems in three-dimensional space [3]. Due to the success of SPH in this field, it was extended to applications in computational mechanics by 1990 [cited in [4]]. It has since been utilised successfully to model various mechanics problems ranging from simple elasticity to complex fracture [5].

SPH has a remarkably simple formulation once fully derived, and one might expect an implementation to be direct. This is indeed the case for the most simple of cases, but problems are encountered early on. The first of these has to do with the imposition of *boundary conditions*. As SPH was developed for unbounded domains (as in astrophysics), the boundary term that arises in the formulation is treated as unimportant (as shall be discussed in Section 2.3). Some special treatment of boundary conditions must be employed for bounded domains. This is in contrast to the finite element method where boundary conditions are naturally enforced. The second, and more vexing issue is that of the so-called *tensile instability* which can cause unbounded growth of the solution.

The aim of this work is to create an approximation of nonlinear elastodynamics, using the most basic SPH formulation, in a total Lagrangian formulation. The structure of the rest of the paper is as follows. First the general SPH approximation is introduced. This includes the description of a new cover function. The SPH approximation is cast in a matrix form, and this is used to provide a treatment of boundary conditions. Following this the SPH approximation is applied to time dependent nonlinear elasticity, and the computational results are evaluated against analytical solutions.

2 SPH approximations

The basis for SPH is an approximation of the sampling relation

$$f(x) = \int_{\Omega} f(\boldsymbol{x}') \, \delta(\boldsymbol{x} - \boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' \tag{1}$$

where $\delta(\boldsymbol{x} - \boldsymbol{x}')$ is the Dirac delta and Ω is the domain. The Dirac delta can be approximated by some function W, that satisfies the normalisation constraint, but which is defined over some finite distance h, referred to as the cover of the function. W is defined to have the properties

$$\int W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}' = 1. \tag{2}$$

The replacement of the Dirac delta with a cover function leads to an approximation of the function called the *kernel approximation*. This is denoted by angular brackets,<>. Thus

$$f(\boldsymbol{x}) \simeq < f > (\boldsymbol{x}) = \int_{\Omega} f(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}' \tag{3}$$

$$< f > (\boldsymbol{x}_p) \simeq \sum_{q=1}^{N} f(\boldsymbol{x}_q) W(\boldsymbol{x}_p - \boldsymbol{x}_q, h) \Delta V_q \tag{4}$$

where equation (4) is a discrete approximation of (3) and p is the particle at which the approximation is made, N is the total number of particles, q, that fall within the support of the cover function W and ΔV_q is the volume (or area in two dimensions) of the support of the cover function.

The gradient to a function can be likewise be approximated by substituting f for its derivative and integrating by parts, giving

$$< \nabla f > (\boldsymbol{x}_p) \simeq - \sum_{q=1}^{N} f(\boldsymbol{x}_q) \nabla' W(\boldsymbol{x}_p - \boldsymbol{x}_q, h) \Delta V_q \, . \tag{5}$$

For convenience, the differential operator acting on W is denoted ∇'.

2.1 The modified mollifier cover function

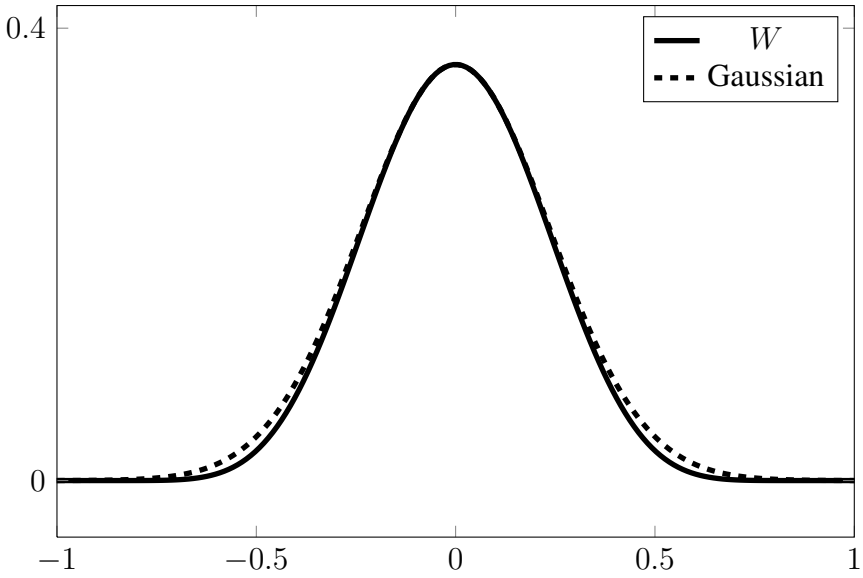


Figure 1: A modified mollifier cover function compared to a Gaussian cover function

In this work a new smoothing function, defined by

$$W(r, h) = \begin{cases} e^{-1/(1-r^2)}(1 - r^2)^8 & \text{for } 0 \leq r < 1 \\ 0 & \text{for } r \geq 1 \end{cases} \tag{6}$$

is introduced and used. It is based on the classical mollifier function $W(r, h) = e^{-1/(1-r^2)}$ which arises in distribution theory (see, for example, [6]). The modified mollifier has C^{∞} continuity and bears close resemblance to the Gaussian function $W(r, h) = e^{-r^2}$. The function as stated is not normalised, and this is done numerically in the implementation.

2.2 Vectorised SPH

The SPH approximation may be vectorised to allow compact notation and implementation. This may be done by defining a matrix relating every particle to every other, via the cover function. In such a matrix the column indicates the particle about which the shape function is centred; the row, which particle it is with respect to.

In order to do so we need to define a vector $\mathbf{f}$ such that

$$\mathbf{f}^T = \left\{ \begin{array}{ccc} f_1 & \dots & f_N \end{array} \right\} \tag{7}$$

where $f_p \, (p = 1, \dots, N)$ represents the value of f at $\boldsymbol{x}_p$. We then define a matrix $\mathbf{W}$ such that

$$\mathbf{W} = \left[\left\{ \begin{array}{c} W_{11} \\ \vdots \\ W_{N1} \end{array} \right\} \quad \cdots \quad \left\{ \begin{array}{c} W_{1N} \\ \vdots \\ W_{NN} \end{array} \right\} \right] \bullet \times \left\{ \begin{array}{c} \Delta V_1 \\ \vdots \\ \Delta V_N \end{array} \right\} \tag{8}$$

where W_{pq} is the value of the cover function at p relative to q. Similarly ΔV_q is the volume contained by the cover function of particle q. In (8), $\mathbf{W}$ is formed by multiplying each column vector by the vector $\{\Delta V_1 \dots \Delta V_N\}^T$.

If we multiply $\mathbf{W}$ and $\mathbf{f}$ we find that

$$\mathbf{Wf} = \left\{ \begin{array}{c} \sum_{q=1}^{N} f\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \\ \vdots \\ \sum_{q=1}^{N} f\left(\boldsymbol{x}_q\right) W\left(\boldsymbol{x}_p - \boldsymbol{x}_q, h\right) \Delta V_q \end{array} \right\} = \left\{ \begin{array}{c} < f_1 > \\ \vdots \\ < f_N > \end{array} \right\} \tag{9}$$

which returns the vector of SPH approximations to f.

We can similarly define a matrix $\mathbf{X}$ by

$$\mathbf{X} = \left[\left\{ \begin{array}{c} \dfrac{\partial W_{11}}{\partial x_1} \\ \vdots \\ \dfrac{\partial W_{N1}}{\partial x_1} \end{array} \right\} \quad \cdots \quad \left\{ \begin{array}{c} \dfrac{\partial W_{1N}}{\partial x_1} \\ \vdots \\ \dfrac{\partial W_{NN}}{\partial x_1} \end{array} \right\} \right] \bullet \times \left\{ \begin{array}{c} \Delta V_1 \\ \vdots \\ \Delta V_N \end{array} \right\} \tag{10}$$

which define the SPH directional derivative with respect to x_1.

2.3 Boundary Conditions

Consider the SPH region shown in Figure 2 consisting of four particles, two of which lie on the boundary. We wish to find the SPH approximation of a function f subject to boundary conditions at the two boundary particles. The conventional ghost particle methodology [5] sets the ghost particles associated with each boundary particle to the value desired. The SPH approximation at the boundary particle will be closer to the desired boundary value had the ghost particles not been included, but may not be exact. Our task is to find what values the ghost particles must be set to such that the Dirichlet boundary conditions are applied exactly.

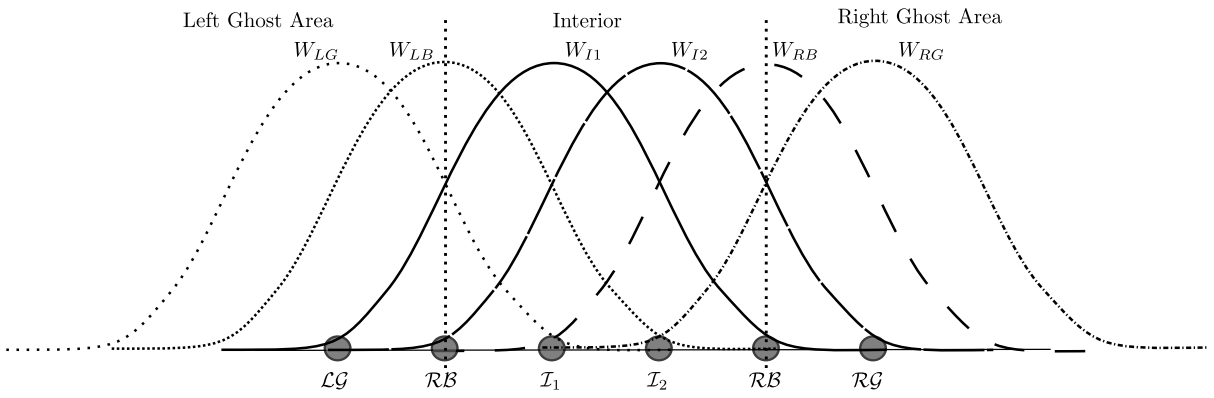


Figure 2: SPH discretisation - interaction of particles with ghost particles

This approach can be generalised for any number of internal, boundary, and ghost particles, denoted $\mathcal{I}$,$\mathcal{B}$ and $\mathcal{G}$ respectively, as

$$\left\{ \begin{array}{c} f_{\mathcal{I}} \\ f_{\mathcal{B}-\text{new}} \\ f_{\mathcal{G}-\text{new}} \end{array} \right\} = \underbrace{\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & W_{\mathcal{B}\mathcal{B}} & W_{\mathcal{B}\mathcal{G}} \\ 0 & W_{\mathcal{G}\mathcal{B}} & W_{\mathcal{B}\mathcal{B}} \end{array} \right]^{-1}}_{\text{modified SPH matrix}} \left\{ \begin{array}{c} f_{\mathcal{I}} \\ \overline{f}_{\mathcal{B}} \\ \overline{f}_{\mathcal{G}} \end{array} \right\} . \tag{11}$$

The computation of the inverse of this modified "SPH matrix" can be expensive, but can be computed once for a simulation, and used repeatedly to compute the value of the ghost particles required to enforce the Dirichlet boundary condition once the SPH approximation is made by calculating

$$\left\{ \begin{array}{c} < f >_{\mathcal{I}} \\ < f >_{\mathcal{B}} \\ < f >_{\mathcal{G}} \end{array} \right\} = \underbrace{\left[\begin{array}{ccc} W_{\mathcal{I}\mathcal{I}} & W_{\mathcal{I}\mathcal{B}} & W_{\mathcal{I}\mathcal{G}} \\ W_{\mathcal{B}\mathcal{I}} & W_{\mathcal{B}\mathcal{B}} & W_{\mathcal{B}\mathcal{G}} \\ W_{\mathcal{G}\mathcal{I}} & W_{\mathcal{G}\mathcal{B}} & W_{\mathcal{B}\mathcal{B}} \end{array} \right]}_{\text{SPH matrix}} \left\{ \begin{array}{c} f_{\mathcal{I}} \\ f_{\mathcal{B}-\text{new}} \\ f_{\mathcal{G}-\text{new}} \end{array} \right\} . \tag{12}$$

It is important to note that this technique would work to set the Dirichlet boundary conditions even if the ghost particles are omitted from the problem. The ghost particles are retained, however, as they are essential to ensure the accuracy of the approximation at near-boundary internal particles.

The disadvantage of this approach is that the technique does not work directly for gradients. The value of the first derivative of any cover function is zero over the particle itself. This leads to any modified "SPH gradient matrix" being not positive-definite, and thus not having a unique inverse. The gradient can only be set approximately, by ensuring that a secondary SPH approximation of the computed gradients satisfies the boundary condition.

3 SPH implementation for nonlinear elastodynamics

The momentum equation for a continuum is given by

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} = \frac{Div\, \boldsymbol{P}}{\rho_0} + \boldsymbol{b} \tag{13}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff strees, $\boldsymbol{u}$ is the displacement and ρ_0 is the density in the reference configuration. The body forces $\boldsymbol{b}$ are assumed to be negligible for this work.

For this work the compressible neo-Hookean model, an adaptation of a material model used to model incompressible rubber-like materials [7], is used. For this material the first Piola-Kirchhoff strees is given by

$$\boldsymbol{P} = \mu \left(\boldsymbol{F} - \boldsymbol{F}\boldsymbol{C}^{-1} \right) + \lambda \left(\ln J \right) \boldsymbol{F}\boldsymbol{C}^{-1} \quad . \tag{14}$$

where $\boldsymbol{F}$ is the deformation gradient, $\boldsymbol{C}$ is the right Cauchy-Green tensor defined by $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$, and J, the Jacobian, is the determinant of $\boldsymbol{F}$ [7, 8].

We define the SPH approximation of the displacement gradient in the x_i direction as

$$< \frac{\partial \boldsymbol{u}}{\partial x_i} >_p = - \sum_{q=1}^{N} (\boldsymbol{u})_q \frac{\partial W \left(\boldsymbol{x}_p - \boldsymbol{x}_q, h \right)}{\partial x_i} \Delta V_q \; . \tag{15}$$

This approximation is then used to compute $\boldsymbol{F}$, its determinant J, and the right Cauchy-Green tensor $\boldsymbol{C}$. These are then used to compute the stress using equation (14). The divergence of this stress is then computed component-wise from the directional derivatives of the stress according to

$$< \frac{\partial \boldsymbol{P}}{\partial x_i} >_p = - \sum_{q=1}^{N} (\boldsymbol{P})_q \frac{\partial W \left(\boldsymbol{x}_p - \boldsymbol{x}_q, h \right)}{\partial x_i} \Delta V_q \quad . \tag{16}$$

The updated displacement of the particles can be computed using a central difference approach

$$\boldsymbol{u}_{n+1} = \Delta t^2 \frac{\partial^2 \boldsymbol{u}_n}{\partial t^2} + 2\boldsymbol{u}_n - \boldsymbol{u}_{n-1} \; . \tag{17}$$

This time integration results in a *explicit* algorithm which is computationally efficient as it does not require matrix inversion.

4 Computational validation

4.1 Small-strain wave propagation

The dynamic behaviour of the approximation is validated through comparison with one-dimensional linear elastodynamic wave theory. It should be noted that for small strains the neo-Hookean material model behaves in a linear manner. For ease of comparison a non-dispersive strain-wave is considered.

If a bar is constrained such that displacement is only possible in the axial direction a one-dimensional strain state is achieved. If such a bar is subjected to a normal surface traction $p(t)$, the one-dimensional Cauchy Stress T_{11} is given by the wave function

$$T_{11} = -p \left(t - \frac{x}{c_b} \right) \tag{18}$$

where c_b is the bar wave speed defined as

$$c_b^2 = \frac{E}{\rho} \tag{19}$$

and E is the Young's modulus of the material [9].

For the validation a neo-Hookean bar of length $0.15\ m$ with density $\rho = 8000$, Young's modulus $E = 200\ GPa$, and Poisson's ratio $\nu = 0.3$ is modelled. The bar is loaded by applying a strain pulse to the left end of the bar equivalent to a stress of

$$\boldsymbol{T}_{11} = -100\ kPa\ .$$

The right end of the bar is set to be rigidly fixed, which results in a total reflection of any wave [9].

To facilitate the validation, four points along the bar are identified where stress history will be compared. These points are at each end of the bar as well as at points at $0.05\ m$ and $0.1\ m$ along the length of the bar, as shown in Figure 3.

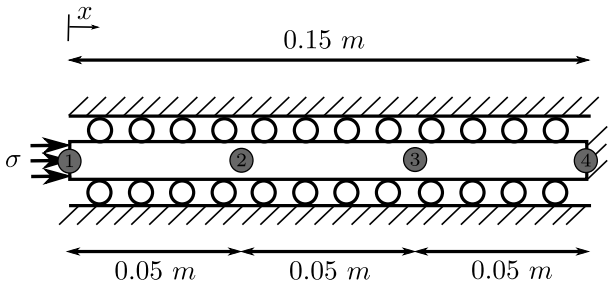


Figure 3: Validation problem

Figure 4 shows the response of the material. Note the numerical noise that is generated is common in all approximations of rapid loading conditions, unless some form of numerical damping is considered.

4.2 Large strain behaviour

The performance of the SPH approximation of the neo-Hookean material is done by comparison with analytical solutions to known strain states. The stress relative to stretch is evaluated by applying a simple triaxial stretch, where the motion is defined to be only in the Cartesian directions, ensuring the principal stretch directions will be along these axes.

The deformation gradient, $\boldsymbol{F}$, the right Cauchy-Green tensor $\boldsymbol{C}$ and Jacobian J are thus defined in terms of the principal stretches $\lambda_1, \lambda_2, \lambda_3$ as

$$\boldsymbol{F} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, \quad \boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} = \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \text{ and } \quad J = \lambda_1\lambda_2\lambda_3\ . \tag{20}$$

The first Piola-Kirchhoff stress is found to be

$$\boldsymbol{P} = \mu \begin{bmatrix} \lambda_1 - 1/\lambda_1 & 0 & 0 \\ 0 & \lambda_2 - 1/\lambda_2 & 0 \\ 0 & 0 & \lambda_3 - 1/\lambda_3 \end{bmatrix} + \lambda \left(\ln \lambda_1\lambda_2\lambda_3\right) \begin{bmatrix} 1/\lambda_1 & 0 & 0 \\ 0 & 1/\lambda_2 & 0 \\ 0 & 0 & 1/\lambda_3 \end{bmatrix} \tag{21}$$

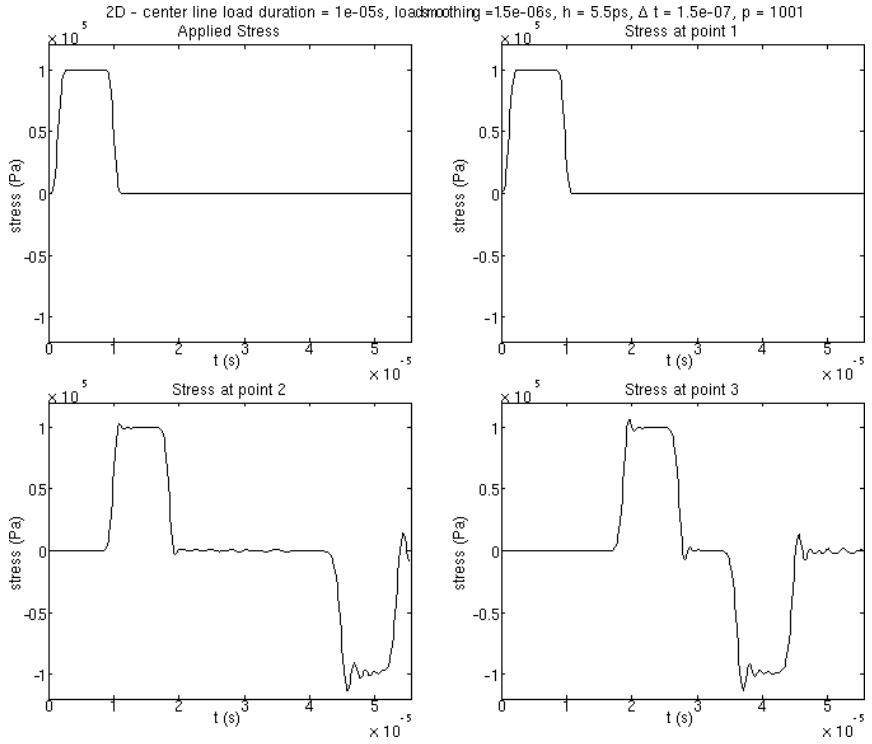


Figure 4: Tensile wave propagation in neo-Hookean material - strain wave

using equation (14).

To make the presentation of this relationship more tractable plane strain conditions are assumed, where the displacement in the third dimension is held constant resulting in a stretch of $\lambda_3 = 1$.

A family of curves can then be generated by varying λ_1 while keeping λ_2 constant for each curve. In such a scenario the material is pre-stressed in the second dimension, and the normal Cauchy stress in the first direction is plotted as a function of the stretch in that direction. The ability of the implementation to accurately model large strain behaviour was tested by taking a $15mm$ sheet of the material and elongating it to $90mm$. This was done over a period of 30 times the time needed for a strain wave to propagate through the material if it were undergoing small strains. This was done in a one-dimensional manner, resulting in a stretch of one in the second and third directions.

Figure 5 shows a comparison between the dynamic behaviour of a compressible sheet of neo-Hookean material as predicted by an SPH simulation and the static curves produced analytically as described above. A close correlation can be seen, with dynamic effects causing the oscillations about the analytical curves. It is important to note the large stretches achieved without any re-discretisation.

5 Conclusions

The SPH approximation to a function has been defined and cast into a matrix formulation which can be used to directly implement SPH as a method for approximating partial differential equations. This approximation is used to model the behaviour of an idealised nonlinear material using a new cover function. The computational model is compared to linear wave propagation

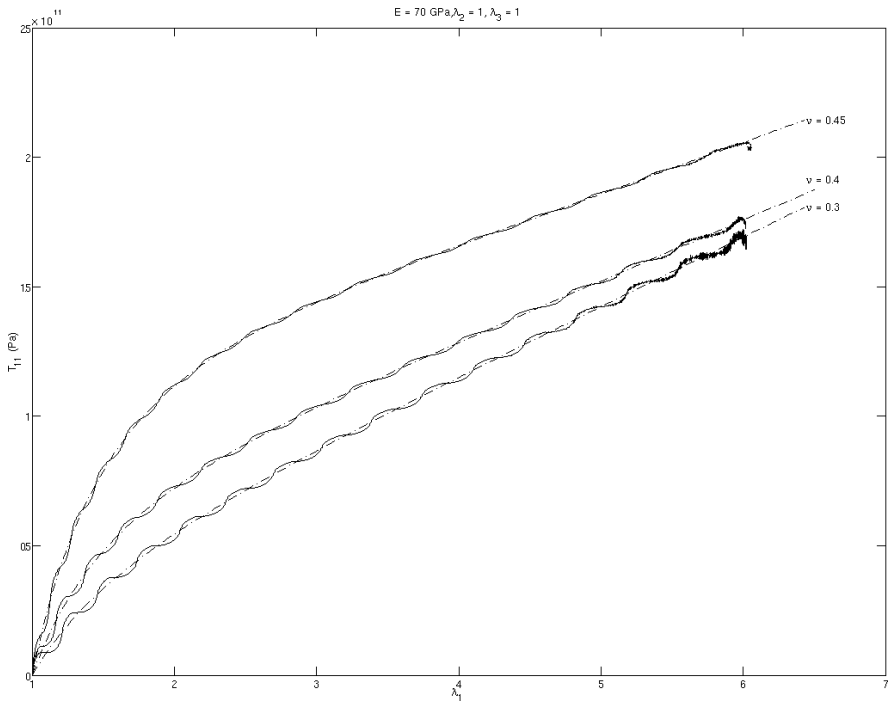


Figure 5: Large deformation of a slightly compressible material

theory (not described in this paper) and to static analytical behaviour. Generally the behaviour is captured well, and large stretches are possible without the need for dynamic re-discretisation.

Additionally this work has been used to model complex two-dimensional shapes, again not presented here. While the results are acceptable, with correctly imposed boundary conditions, investigation into non-structured discretisations and more complex material models is required.

References

[1] R. Vignjevic, J. R. Reveles, and J. Campbell. SPH in a total Lagrangian formalism. *Computer Modelling in Engineering and Sciences*, 14(3):181–198, 2006.

[2] T.E. De Vuyst, R. Vignjevic, and J.C. Campbell. Coupling between meshless and finite element methods. *International Journal of Impact Engineering*, 31:1054–1064, 2005.

[3] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astrophysical Journal*, 82:1013–1024, 1977.

[4] G.M. Zhang and R.C. Batra. Modified smoothed particle hydrodynamics method and its application to transient problems. *Computational Mechanics*, 34:37–146, 2004.

[5] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics*. World Scientific, 2003.

[6] B. D. Reddy. *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Elements*. Springer, 1998.

[7] R. W. Ogden. Large deformation isotropic elasticity - on the correlation of theory and experiment for compressible rubberlike solids. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 328(1575):567–583, 1972.

[8] J. Bonet and R.D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge university press, 1997.

[9] J.D. Achenbach. *Wave Propagation in Elastic Solis*. North-Holland publishing Company, 1973.