Smoothed Particle Magnetohydrodynamics for the Solar Corona

Thomas Matthew Carpenter Knight

Institute of Maths and Physics
Aberystwyth University

Dr. Balázs Pintér - Supervisor

July 2012

This thesis is submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy of Aberystwyth University.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... (candidate)

Date .....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ..... (candidate)

Date .....

Statement 2

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

Acknowledgments

Firstly, I would like to thank my unerringly patient and understanding wife Pippa and my son Cae for both their love and support.

In addition, I must thank the Institute of Maths and Physics staff and, in particular, my supervisors Dr. Balázs Pintér, Prof. Manuel Grande and Dr. Daniel Brown.

I would also like to express my gratitude to my fellow PhD students, past and present. Most especially, David Forster, Jimmy Carter, Martin Vickers, Alex Leatherland, Gemma Guymer, Ian Whittaker, Drew Leonard and Paddy Dixon.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The aim of this Thesis is the construction, by reasoned argument and investigation, of a Smoothed Particle (SP) algorithm and model suitable for the stable, accurate simulation of Solar Coronal phenomena. Before discussing any details, consider the physical and numerical scientific background.

1.1 Solar Corona

Our star, the Sun, is a massive body of ionised gas (plasma) that radiates vast amounts of energy. So massive, and so energetic, is the Sun that the warring forces that govern it give rise to numerous layers of material, each with their own physically distinct characteristics. Referring for context to Figure 1.1, at the centre of the Sun is the Core. A region of such extremes of temperature and pressure that the matter undergoes fusion. Surrounding the Core is the corotating Radiative Zone, so named because the energy is transported by radiating through the opaque medium. As the opacity of the Radiative Zone increases (radially) the thermal energy builds, until it is more efficient for the energy to be transported by convection. In this way the Radiative Zone gives way to the Convective Zone. Sitting atop the Convection Zone is the only visible layer of plasma, and upper surface of the body of the Sun, the Photosphere. From this point on, the layers are referred to as part of the Solar Atmosphere. The physical properties also begin to act strangely. From the Core to the Photosphere the temperature of each layer has been reducing proportionally, depending on its distance from the fusion, until the Photosphere sits at approximately 5.5×10^3 K. The Chromosphere, the first layer of atmosphere sitting directly above the Photosphere, is at approximately 7×10^3 K. The final, outermost layer of the Sun before the solar plasma gives way to interplanetary space is the Solar Corona at approximately 2×10^6 K.

It is this outermost layer that is the focus of this work. The Corona is a very hot and diffuse ($\sim 10^{12}$ kgm^{-3}) fully ionised plasma. It is also highly variable. Suspended within it by powerful and complex magnetic field structures are pockets of cooler, dense plasma. As the magnetic fields evolve, they store an increasing amount of energy with potentially dramatic consequences, from solar flares to prominences to

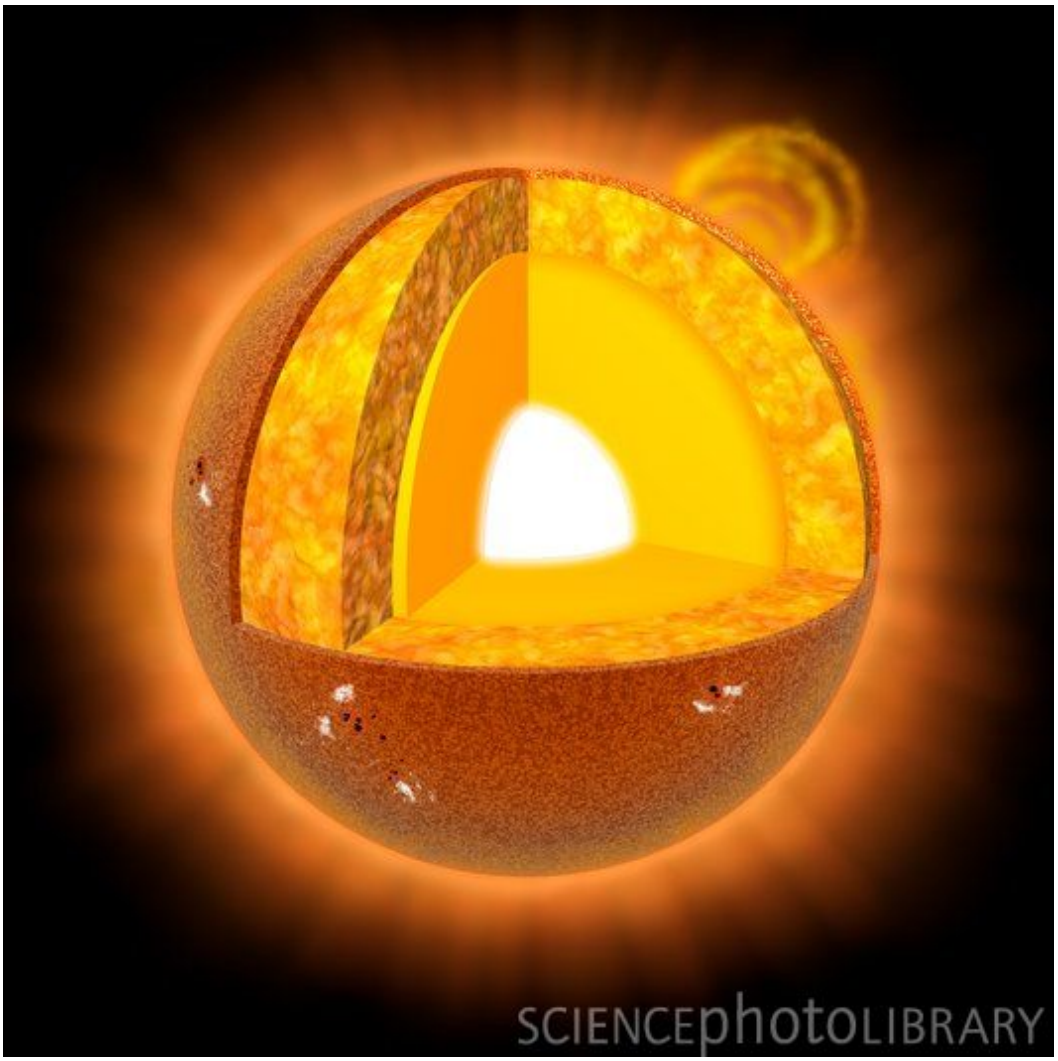

Figure 1.1: A cut-away diagram showing the basic layers of the Sun. Moving out from the centre, the layers are the Core, the Radiative and Convective Zones, the Photosphere, the Chromosphere and the Corona. The final two layers are diffuse, shown here as the opaque peach layer and striated region, respectively.

coronal mass ejections (CMEs). These explosive phenomena accelerate vast amounts of material into interplanetary space, with the potential for striking the planets including Earth. This accounts for the majority of Space Weather. During the active phase of the $\sim$ 11-year solar cycle, the atmosphere can become so distorted by eruptions that there is no stable Corona. Given this dynamic behaviour, the Corona is a focus of much research. However, this work is challenging, and physicists must rely on numerical experiments to overcome, or at least reduce, some of the difficulties.

1.2 Smoothed Particle Hydrodynamics

The smoothed particle (SP) method is a technique that can approximate the behaviour of complex media as numerical algorithms. Presented here is a brief summary of the background and concepts underlying the SP method. A detailed presentation of the technique is presented in the subsequent chapters.

Conceptually, the SP method divides a material into a number of finite elements. These elements are then free to interact as instructed by the macroscopic physics of the original media. This it does with relative ease which has led, in the 45 years since its original presentation by Lucy (1977) and Gingold and Monaghan (1977), to the technique spreading far beyond the bounds of its astrophysical childhood, and into a wide range of disciplines within the academic, industrial and service sectors. It has been applied to; binary fission and instabilities (Gingold and Monaghan, 1978; Monaghan, 2005), variable smoothing lengths (Nelson, 1994; Price and Monaghan, 2004b; Buchlin, 2007), relativistic dynamics (Chow and Monaghan, 1997), compressible turbulence (Monaghan, 2002), elastic materials (Gray et al., 2001) and, most relevantly, large scale (Rosswog and Price, 2007) and small scale (Monaghan and Price, 2004, 2006) MHD applications. The latter distinction, between the scale of MHD problems attempted, is made because the intermediate scale (smaller than stars collisions but greater than tokamak based simulations, such as coronal loops) remains, as yet, unexplored. It is this length scale, $\sim O(\mathrm{Mm})$, in the solar corona which is the focus for the work herein

1.3 Thesis

This thesis is divided into two parts. Chapters 2 and 3 present the established SP method as described by the surrounding literature. The method is divided between the chapters into the rigorous physical components and the numerical/artificial components, respectively. The remaining Chapters (4, 5 and 6) present the original

research portion of this document.

That research is focused on the construction of an SP algorithm and domain built to simulate the transient phenomena located in the Solar Corona. Without being too specific as to the phenomena, certain conclusions can still be drawn about the general conditions faced by the model by, for example, looking at other successful numerical approaches (Chen et al., 2002). For instance, the model must be stable in a regime of high temperatures, low densities, negative pressures, steep gradients, high rate of deformation, and rapidly varying time and length scales. As a finite-element method, SP theory can automatically handle, for example, the high rate of deformation. However, negative stresses and steep gradients could present something of a problem. In addition, the geometry of the solar corona and the fine-grained nature of the phenomena could present a problem for the domain and element parameters.

It is these problems that are addressed though reasoned argument, in the case of the negative pressures (Chapter 3), and investigation, as with the steep gradients (Chapter 5).

Subsequent to a thorough discussion of general SP theory and implementation, this document initially presents original research investigating the nature of the techniques defining variable spatial resolution (variable smoothing length), and discovers a numerical artefact that in the presence of large variations in smoothing length drastically effects the stability of the algorithm (Chapter 4). In order to better understand this artefact, the subsequent research focuses on the quantification of errors in SP algorithms (Chapter 5) leading to the cumbersome, but more accurate Corrected SP (CSP) method. Following an investigation into the behaviour of that algorithm, another is presented that conforms to the high accuracy generated by the CSP method, but strips away that which makes it so computationally weighty. Entitled the CSP-Δh method, it avoids some of the complications of the former method by justifiably dropping those terms that are a function of the gradients of spatial resolution (that is the spatial gradients of the smoothing length). Finally, in Chapter 6, the work defines a novel domain geometry in order to capture the required dynamics with as little wasted computation as possible. Two new boundaries have been established, the first a specification of the periodic boundary and the second a novel boundary that attempts to let information about the internal dynamics out of the domain while ensuring the model evolution remains stable.

Chapter 2

Smoothed Particle Theory

Contents

2.1 Introduction

This chapter concerns the derivation of smoothed particle (SP) theory and the creation of the basic SPMHD algorithm. Note that the components presented here are only those that stem from physical or mathematical argument. The artificial and numerical components of the model are addressed is Chapter 3.

The derivation, in one form or another, is presented in almost every paper concerning SP methods. Many of the points raised in the sections that follow are common to all of them. However, certain papers are particularly illuminating. Most notably Hernquist and Katz (1989); Monaghan (2005); Price (2012), and the series of papers Price and Monaghan (2004a,b, 2005); Price (2010).

2.2 Integral Basis

SP algorithms can be created from two bases. The most recent review paper (Price, 2012) for instance, presented the derivation from density estimate basis. However, for reasons discussed in Section 2.3, presented in detail herein is the more common derivation from the integral basis.

2.2.1 Fundamental Proposition

The aim of any SP algorithm is the simulation of some physical medium or conglomerate of related mediums. To create a model, a conceptual space must be defined (as the simulation domain, Ω) with some number of dimensions, ν, and bound by a surface, $S(\Omega)$, within which the appropriate equations can be solved. In SP theory the assumption is made that any information exists as macroscopic properties or field variables. Thus, for example, any velocities quoted would refer exclusively to the motion of portions of the bulk medium rather than individual physical units (atoms, molecules, etc.) that make up those portions. A list of macroscopic properties would include; velocity, magnetic field, density, mass, thermal energy, pressure, etc. However, colour charge or spin would be examples of non-macroscopic properties as they are clearly isolated to individual particles and are, as such, discontinuous throughout a domain. This assumption clearly restricts the media (or, more specifically, the systems of equations) that can be simulated by an SP algorithm.

Theorem 2.1 *If λ is a scalar macroscopic property that exists throughout the domain, $\lambda(\mathbf{r})$ may be expressed in the form*

$$\lambda(\mathbf{r}) = \int_{\Omega} \lambda(\mathbf{r}')\delta(\mathbf{r} - \mathbf{r}')\mathrm{d}\mathbf{r}' \tag{2.1}$$

where δ is the Dirac delta function.

There can be some confusion regarding the specific form of the delta function, either Dirac or Kronecker, as both can be described in the context of an integral as a function equal to zero for any input excluding for a single value, at which the function is equal to one. To clarify, the Kronecker delta acts on discrete, usually integer, variables

$$\delta^{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

whereas the Dirac delta function (see Figure 2.1) is, more accurately, a continuum of zero with an infinite pulse centered at zero. For instance,

$$\delta(x) = \lim_{a \to 0} \left[\frac{1}{a\sqrt{\pi}} e^{-\frac{x^2}{a^2}} \right] \tag{2.3}$$

such that, conceptually,

$$\delta(x) = \begin{cases} \infty & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

additionally constrained by

$$\int_{-\infty}^{\infty} \delta(x)\mathrm{d}x = 1 \tag{2.5}$$

In a similar manner that may account for the confusion surrounding delta functions, an integral of the Kronecker delta over all possible input values (all integers, hence the integral is replaced by a summation) conforms to

$$\sum_{j=-\infty}^{\infty} \delta^{ij} = 1 \quad \text{for} \quad i \in \mathbb{Z} \tag{2.6}$$

Though the practical distinction here appears insignificant, the implications regarding the theoretical concepts are not. Consider that the integral (2.1) and the Dirac delta function operate on the entire continuum of the domain. Any specific λ, therefore, must be continuous. Thus, the fundamental proposition validates the assumption that λ must be some macroscopic variable.

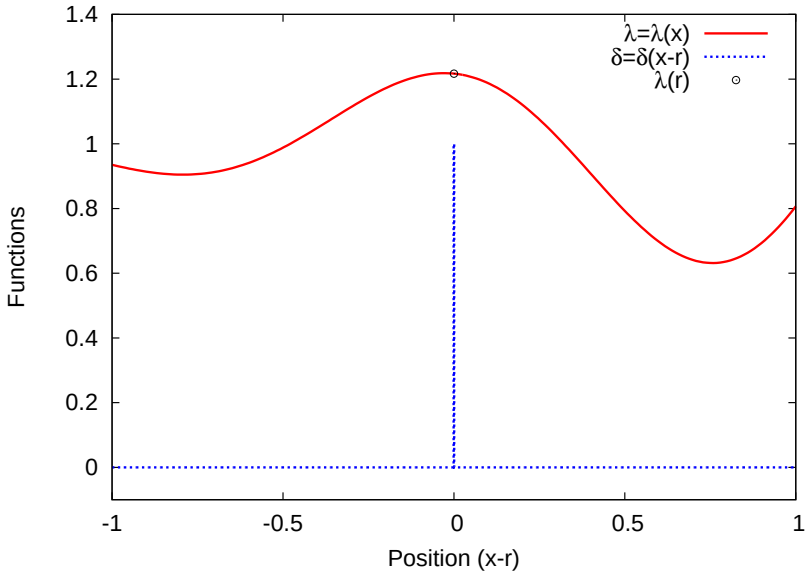

Figure 2.1: A plot showing, over a 1-Dimensional domain $-1 \leq r - x \leq 1$, the Dirac delta function and some continuous property λ as a blue and red line, respectively. Using equation (2.1), with the point of interest at x, $\lambda(x)$ is shown as a black circle.

Theorem 2.1 is the proposition upon which the rest of SP theory is constructed. However, this construction requires numerous steps. First among them is the smoothing approximation.

2.2.2 Smoothing Approximation

The smoothing approximation is one of two unavoidable approximations made in the creation of any SP algorithm. It is essential to the discretisation process required for computation.

Consider some hypothetical discretisation of the domain (Ω) and the information within it. The integral over all points in Ω of equation (2.1) would become a summation over the finite number of locations at which information exists. Given the intended Lagrangian frame of reference, equation (2.1) could be calculated at any one of the infinite number of points contained within Ω, therefore it is unlikely that the unit pulse of the delta function occurs where the information exists, and the only result from the equation would be zero. If, however, the Dirac delta function were *smoothed out* over some small distance, defined relative to the scale of the discretisation, non-zero outputs from the equation would become possible.

Theorem 2.2 *In order to approximate equation (2.1), the Dirac δ must be replaced by some smoothing function, W.*

$$\lambda(\mathbf{r}) \approx [\lambda(\mathbf{r})]_{\text{sm}} = \int_{\Omega} \lambda(\mathbf{r}')W\left(\mathbf{r}-\mathbf{r}', h\right) \mathrm{d}\mathbf{r}' \tag{2.7}$$

where h is the smoothing length. This is the **smoothing approximation**.

This expression (2.7) places a number of constraints on the nature of the smoothing function.

2.2.3 Constraints on the Smoothing Function

The smoothing function, often referred to as the kernel (Monaghan, 2005), can take a number of forms (Fulk and Quinn, 1996). However, its behaviour must conform to particular bounds. Initially, as a direct result of Theorem 2.2, as the smoothing length collapses to zero (and the resolution increases),

$$\lim_{h\to 0} W\left(\mathbf{r}-\mathbf{r}', h\right) = \delta(\mathbf{r}-\mathbf{r}') \tag{2.8}$$

This constraint in fact breaks down to form a number of other constraints. Simplest of all, that the function should be normalised by

$$\int_{\Omega} W \mathrm{d}\mathbf{r}' \equiv 1 \tag{2.9}$$

such that

$$\lim_{h\to 0} \ [\lambda(\mathbf{r})]_{sm} = \lambda(\mathbf{r}) \tag{2.10}$$

Also, since $\delta \geq 0$,

$$W \geq 0 \text{ over } \Omega \tag{2.11}$$

Further to equation (2.11), for the purpose of the SP method (i.e. from a numerical perspective), the smoothing function should have some form of compact support such that the calculations for some arbitrary point require only information stored in the surrounding region, rather than information from the entire domain. That is,

$$W\left(|\mathbf{r}-\mathbf{r}'|, h\right) = 0 \text{ for } \left|\mathbf{r}-\mathbf{r}'\right| > Ch \tag{2.12}$$

where $C > 0$ is some constant. Typically $C = 2$ as the cubic spline is most commonly employed as the smoothing function (see Section 2.2.4). In this way, the smoothing length h must be proportional to the spatial resolution of the simulation.

The compact support can be represented as a spherical domain which is a function of h, $\omega(h)$, about $\mathbf{r}$ such that

$$[\lambda(\mathbf{r})]_{\text{sm}} = \int_{\Omega} \lambda(\mathbf{r}')W \mathrm{d}\mathbf{r}' = \int_{\omega(h)} \lambda(\mathbf{r}')W \mathrm{d}\mathbf{r}' \tag{2.13}$$

Though not implicit in the mathematics, the smoothing function should also be even, that is

$$W\left(\mathbf{r}-\mathbf{r}', h\right) \equiv W\left(|\mathbf{r}-\mathbf{r}'|, h\right) \tag{2.14}$$

Given that the *application* of the smoothing approximation is intended for physical mediums, the integral must conform to physical laws, in particular physical consistency. Consider, using an asymmetric smoothing function in disagreement with equation (2.14) and assuming some variable field λ, the calculation of equation (2.7) would give some answer. Now, reconsider the calculation having altered only the orientation of the coordinate system. The answer must have changed, which is inconsistent with the laws of physics as they should be the same regardless of coordinate system. However, repeating the procedure, assuming a symmetric (and therefore even) smoothing function, the answers will be identical regardless of coordinate system.

There are additional constraints placed on the spatial derivative of the smoothing function. From a numerical perspective, the derivative should be finite and continuous. Also note as a comment on the behaviour, that

$$\nabla^j W\left(|\mathbf{r}-\mathbf{r}'|, h\right) = \frac{\partial W\left(|\mathbf{r}-\mathbf{r}'|, h\right)}{\partial r_j} \tag{2.15}$$

where j denotes the dimensional components of the coordinate system. By convention here $1 \leq j \leq \nu$ and $j \in \mathbb{Z}$, however any consistent discrete value system will suffice. Repeated application of the chain rule of differentiation yields

$$\nabla^j W\left(|\mathbf{r}-\mathbf{r}'|, h\right) = \frac{\partial \mathbf{r}-\mathbf{r}'}{\partial r_j}\frac{\partial |\mathbf{r}-\mathbf{r}'|}{\partial \mathbf{r}-\mathbf{r}'}\frac{\partial W\left(|\mathbf{r}-\mathbf{r}'|, h\right)}{\partial |\mathbf{r}-\mathbf{r}'|} \tag{2.16}$$

so that

$$\nabla^j W\left(|\mathbf{r}-\mathbf{r}'|, h\right) = \frac{r_j - r_j'}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial W\left(|\mathbf{r}-\mathbf{r}'|, h\right)}{\partial |\mathbf{r}-\mathbf{r}'|} \tag{2.17}$$

and therefore the gradient as calculated looking from $\mathbf{r}$ to $\mathbf{r}'$ must be equal and opposite to the gradient as calculated looking from $\mathbf{r}'$ to $\mathbf{r}$, i.e.

$$\nabla W\left(|\mathbf{r}-\mathbf{r}'|, h\right) = -\nabla' W\left(|\mathbf{r}'-\mathbf{r}|, h\right) \tag{2.18}$$

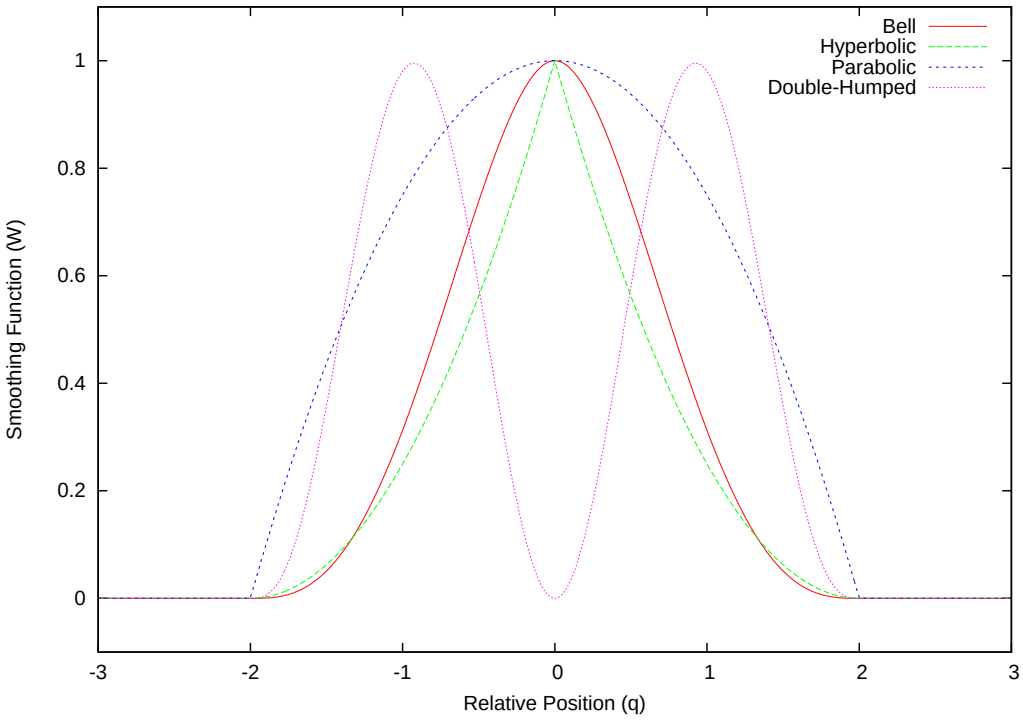

Figure 2.2: A graph showing examples of the four smoothing function categories; bell-, hyperbolic-, parabolic-, and double-humped-shaped. See Table 2.1 for specific examples.

2.2.4 General Smoothing Function Forms

Smoothing functions can be categorised into four main forms; Bell, Hyperbolic, Parabolic and Double-Hump. Specific examples are shown in Table 2.1 and Figure 2.2. However, more generally, most have the form

$$W\left(|\mathbf{r}-\mathbf{r}'|,h\right)=\frac{\sigma_\nu}{h^\nu}F(q) \tag{2.19}$$

where $q=|\mathbf{r}-\mathbf{r}'|/h$ is the relative spatial distance, and σ_ν is the normalisation constant responsible for ensuring equation (2.9). The function F defines the shape of the smoothing function as a function of relative distance.

More information concerning the smoothing function definitions in Table 2.1 and many additional definitions, and an extensive discussion of possible methods of determining a specific functions suitability to act as a smoothing function can be found in Fulk and Quinn (1996).

In the originator paper (Gingold and Monaghan, 1977), three examples of

Catagory	Name	F(q)
Bell	Cosine	$(1-0.25q^2)(1+\cos(0.5\pi q))$
	L Gaussian	$(2-q)e^{-q^2}$
	Quartic-1	$(2+3q)(2-q)^3$
Hyperbolic	$\kappa-2$ Exponential	$e^{-2.25q^2}-e^{-9}$
	'1/X, 2'	$(2+q)^{-1}+0.0625(q-6)$
	'-X^2'	$0.5(q-2)^2$
Parabolic	'-X-exp(-X)'	$2-q-e^{-q}+e^{-2}$
	'4-X^2'	$4-q^2$
	'8-X^3'	$8-q^3$
Double-Humped	Double Q Gauss	$0.25q^2(4-q^2)e^{-q^2}$
	Double T Gauss	$q^2(e^{-q^2}-e^{-4})$
	Double Q-1	$q^2(2+3q)(2-q)^3$

Table 2.1: A small selection of the smoothing functions found in the surrounding literature (Fulk and Quinn, 1996).

smoothing functions were presented. The predominant form for the smoothing function was defined by a Gaussian curve. That is

$$F(q) = e^{-q^2} \tag{2.20}$$

Herein, the smoothing function is defined using bell shaped curve, the cubic spline (also referred to as the M_4-Spline or the W_4 B-Spline Kernel (Monaghan, 2005)), which achieved good scores in the assessment formulae described in Fulk and Quinn (1996) and is the most commonly implemented form given in surrounding literature. Explicitly,

$$W\left(|\mathbf{r}-\mathbf{r}'|,h\right) = \frac{\sigma_\nu}{h^\nu}\mathrm{M}_4(q) \tag{2.21}$$

where

$$M_4(q) = \begin{cases} 1-\frac{3}{2}q^2+\frac{3}{4}q^3 & \text{if} \quad 0 \le q < 1 \\ \frac{1}{4}(2-q)^3 & \text{if} \quad 1 \le q < 2 \\ 0 & \text{if} \quad q \ge 2 \end{cases} \tag{2.22}$$

and the normalisation constants are

$$\sigma_\nu = \begin{cases} {}^2/_3 & \text{if} \quad \nu = 1 \\ {}^{10}/_{7\pi} & \text{if} \quad \nu = 2 \\ {}^1/_\pi & \text{if} \quad \nu = 3 \end{cases} \tag{2.23}$$

From equation (2.17), the corresponding smoothing function derivative is given

by

$$\nabla W\left(|\mathbf{r}-\mathbf{r}'|,h\right)=\frac{r_j-r'_j}{|\mathbf{r}-\mathbf{r}'|}\frac{\sigma_\nu}{h^{\nu+1}}\begin{cases}\frac{9}{4}q^2-3q & \text{if} \quad 0\leq q<1\\ 3q-\frac{3}{4}q^2-3 & \text{if} \quad 1\leq q<2\\ 0 & \text{if} \quad q\geq 2\end{cases} \tag{2.24}$$

Alternatively, the derivative could be generalised by substituting the general form of smoothing function, equation (2.19), into the expanded derivative given by equation (2.17).

$$\begin{aligned}\nabla^j W\left(|\mathbf{r}-\mathbf{r}'|,h\right)&=\frac{r_j-r'_j}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial\left(\sigma_\nu h^{-\nu}F(q)\right)}{\partial|\mathbf{r}-\mathbf{r}'|}\\ &=\frac{r_j-r'_j}{|\mathbf{r}-\mathbf{r}'|}\frac{\sigma_\nu}{h^\nu}\frac{\partial q}{\partial|\mathbf{r}-\mathbf{r}'|}\frac{\partial F(q)}{\partial q}\\ &=\frac{r_j-r'_j}{|\mathbf{r}-\mathbf{r}'|}\frac{\sigma_\nu}{h^{\nu+1}}\frac{\partial F(q)}{\partial q}\end{aligned} \tag{2.25}$$

2.2.5 Error in the Smoothing Approximation

By analysing the smoothing approximation (Theorem 2.2), the error implicit in its definition may be expressed. That definition also serves to prove that such an approximation can be made. Consider the Taylor series expansion of $\lambda(\mathbf{r}')$ about $\mathbf{r}'=\mathbf{r}$, in equation (2.7),

$$[\lambda(\mathbf{r})]_{\text{sm}}=\int_\Omega\left\{\sum_{n=0}^{\infty}\frac{(\mathbf{r}'-\mathbf{r})^n}{n!}\left[\frac{\mathrm{d}^n\lambda(\mathbf{r}')}{\mathrm{d}x^n}\right]_{\mathbf{r}'=\mathbf{r}}\right\}W\mathrm{d}\mathbf{r}' \tag{2.26}$$

Recalling that W is an even function,

$$[\lambda(\mathbf{r})]_{\text{sm}}=\int_\Omega\left\{\sum_{\substack{n=0\\ n\neq\text{odd}}}^{\infty}\frac{(\mathbf{r}'-\mathbf{r})^n}{n!}\left[\frac{\mathrm{d}^n\lambda(\mathbf{r}')}{\mathrm{d}x^n}\right]_{\mathbf{r}'=\mathbf{r}}\right\}W\mathrm{d}\mathbf{r}' \tag{2.27}$$

and expansion gives

$$[\lambda(\mathbf{r})]_{\text{sm}}=\lambda(\mathbf{r})\int_\Omega W\mathrm{d}\mathbf{r}'+\int_\Omega\left\{\sum_{\substack{n=2\\ n\neq\text{odd}}}^{\infty}\frac{(\mathbf{r}'-\mathbf{r})^n}{n!}\left[\frac{\mathrm{d}^n\lambda(\mathbf{r}')}{\mathrm{d}x^n}\right]_{\mathbf{r}'=\mathbf{r}}\right\}W\mathrm{d}\mathbf{r}' \tag{2.28}$$

Given the normalisation constraint, equation (2.9), the expression above col-

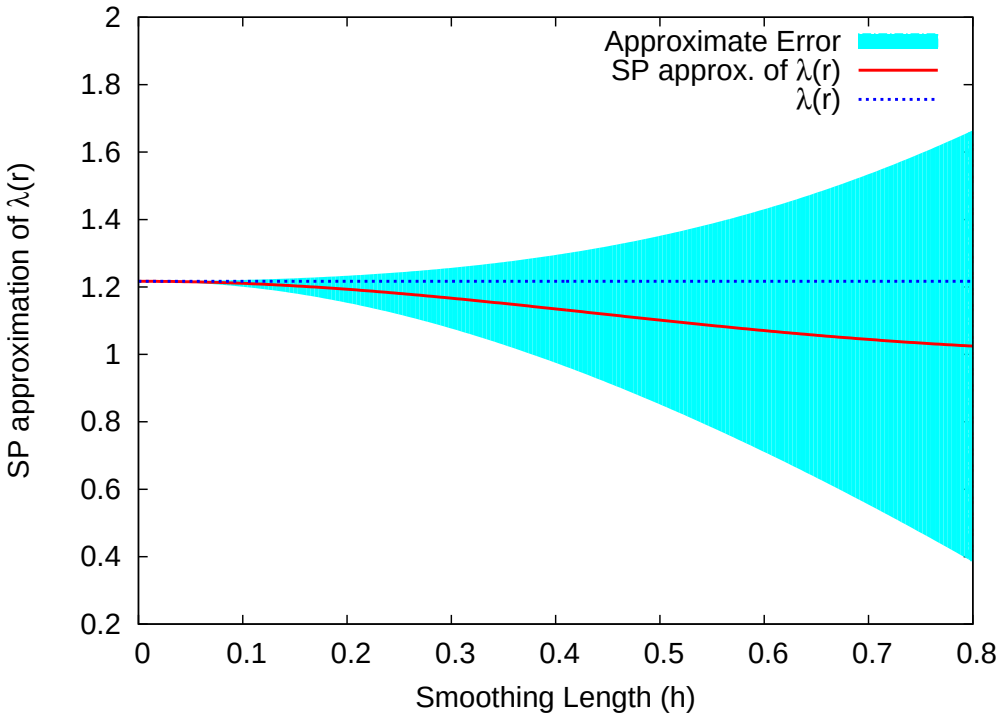

Figure 2.3: A graph demonstrating the ability of equation (2.7) to approximate $\lambda(x)$, as a function of smoothing length. Note also the shaded region as a representation of the error on the approximation (see Section 2.2.5).

lapses to give

$$[\lambda(\mathbf{r})]_{\text{sm}} = \lambda(\mathbf{r}) + O\left(h^2\right) \tag{2.29}$$

Thus the error may now be formalised,

Theorem 2.3 *By defining that some approximation, $[X]_{\text{approx}}$, is the sum of the true value, X, and the error in that estimation, i.e.*

$$[X]_{\text{approx}} = X + \varepsilon_{\text{approx}} \tag{2.30}$$

the error induced by the smoothing approximation (Theorem 2.2) is given by

$$\varepsilon_{\text{sm}} = O\left(h^2\right) \tag{2.31}$$

where $[\lambda(\mathbf{r})]_{\text{sm}} = \lambda(\mathbf{r}) + \varepsilon_{\text{sm}}$. This error is demonstrated in Figure 2.3

2.2.6 Smoothing Approximation of Spatial Derivatives

Given the eventual application of the SP method to equations governed by gradients, the behaviour of the smoothing approximation of derivatives is of particular interest.

Theorem 2.4 *For some macroscopic, vector property at* $\mathbf{r}$, $\mathbf{A}(\mathbf{r})$, *the smoothing approximation of the divergence is*

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \int_{\Omega} \mathbf{A}(\mathbf{r}') \cdot \nabla W \left(|\mathbf{r} - \mathbf{r}'|, h\right) \mathrm{d}\mathbf{r}' \tag{2.32}$$

Similar forms exist as, for instance, the Grad, Curl and Tensor Product Smoothing Approximations. These are the **Zeroth-Order Derivative Smoothing Approximations**.

There are two possible arguments that justify Theorem 2.4. The first, and least evidenced, begins by assuming

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \nabla \cdot [\mathbf{A}(\mathbf{r})]_{\text{sm}} = \nabla \cdot \left(\int_{\Omega} \mathbf{A}(\mathbf{r}') W \mathrm{d}\mathbf{r}' \right) = \int_{\Omega} \nabla \cdot \left(\mathbf{A}(\mathbf{r}') W \right) \mathrm{d}\mathbf{r}' \tag{2.33}$$

The assumption is made that the vector information – where it exists – exists only as statistical sample data and does not form part of a continuum. As such, $\mathbf{A}(\mathbf{r}')$ are constant and therefore,

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \int_{\Omega} W \nabla \cdot \mathbf{A}(\mathbf{r}') + \mathbf{A}(\mathbf{r}') \cdot \nabla W \mathrm{d}\mathbf{r}' = \int_{\Omega} \mathbf{A}(\mathbf{r}') \cdot \nabla W \mathrm{d}\mathbf{r}' \tag{2.34}$$

However, this argument makes many assumptions and fails to identify the issues that occur at the edge of the simulation domain (discussed by several authors (Vignjevic, 2004; Song and Dong, 2010; Price, 2012)).

Alternatively, without these assumptions, let $\lambda(\mathbf{r}) = \nabla \cdot \mathbf{A}(\mathbf{r})$ be substituted into equation (2.7) such that

$$\nabla \cdot \mathbf{A}(\mathbf{r}) \approx [\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \int_{\Omega} \nabla' \cdot \mathbf{A}(\mathbf{r}') W \mathrm{d}\mathbf{r}' \tag{2.35}$$

Recalling the surface of the simulation domain, $S(\Omega)$, this expression can be integrated by parts to give

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \oint_{S(\Omega)} W \mathbf{A}(\mathbf{r}') \mathbf{n} \cdot \mathrm{d}\mathbf{S}' - \int_{\Omega} \mathbf{A}(\mathbf{r}') \cdot \nabla' W \mathrm{d}\mathbf{r}' \tag{2.36}$$

The surface integral in equation (2.36) reduces to zero if the surface of subdomain created by the compact support of the smoothing function, $S\left(\omega(\mathbf{r})\right)$, does not

intersect the surface of the simulation domain. The assumption is made that all points at which the smoothing approximation is calculated are within the domain by a minimum distance no less than Ch. This assumption explicitly defines the SP method's inability to model domain edges without modification. Also recalling the behaviour given by equation (2.18),

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\mathrm{sm}} = -\int_{\Omega} \mathbf{A}(\mathbf{r}') \cdot \nabla' W \mathrm{d}\mathbf{r}' = \int_{\Omega} \mathbf{A}(\mathbf{r}') \cdot \nabla W \mathrm{d}\mathbf{r}' \tag{2.37}$$

Theorem 2.5 Substitution: *Consider some set G that is variable as a function of position and contains all the $N \equiv |G(\mathbf{r})| \equiv |G(\mathbf{r}')|$ properties of the system such that $G_k(\mathbf{r})$ is the k^{th} property, q_k, of the system at $\mathbf{r}$, and let*

$$\frac{\partial G_k(\mathbf{r})}{\partial q_k} = f_k\left(\mathbf{r}, G\right) \tag{2.38}$$

So that

$$\frac{\partial G_k(\mathbf{r})}{\partial q_k} \approx \left[\frac{\partial G_k(\mathbf{r})}{\partial q_k}\right]_{\mathrm{sm}} = \int_{\Omega} \frac{\partial G_k(\mathbf{r}')}{\partial q_k} W \mathrm{d}\mathbf{r}' \tag{2.39}$$

Thus by substitution, and Theorem 2.2,

$$\left[\frac{\partial G_k(\mathbf{r})}{\partial q_k}\right]_{\mathrm{sm}} = \int_{\Omega} f_k(\mathbf{r}', G) W \mathrm{d}\mathbf{r}' = [f_k(\mathbf{r}, G)]_{\mathrm{sm}} \tag{2.40}$$

Therefore it is possible to substitute for the derivative terms in some f_k, the derivative smoothing approximations (Theorem 2.4) while leaving the non-derivative G terms in place.

However, note that $[\nabla \cdot \mathbf{A}]_{\mathrm{sm}}$ can be made more accurate if $\mathbf{A}$ is redefined by

$$\mathbf{A}(\mathbf{r}) = \frac{\Psi(\mathbf{r})\mathbf{A}(\mathbf{r})}{\Psi(\mathbf{r})} \tag{2.41}$$

where Ψ is some scalar function. So that, by the quotient rule,

$$\nabla \cdot \mathbf{A}(\mathbf{r}) = \nabla \cdot \left[\frac{\Psi(\mathbf{r})\mathbf{A}(\mathbf{r})}{\Psi(\mathbf{r})}\right] = \frac{1}{\Psi(\mathbf{r})} \nabla \cdot [\Psi(\mathbf{r})\mathbf{A}(\mathbf{r})] - \frac{\mathbf{A}(\mathbf{r})}{\Psi(\mathbf{r})} \cdot \nabla \Psi(\mathbf{r}) \tag{2.42}$$

Thus, substitution of the zeroth-order derivative smoothing approximations (Theorem 2.4) gives equation (2.44). Alternatively by application of the smoothing approximation and substitution theorem (Theorems 2.2),

$$\nabla \cdot \mathbf{A}(\mathbf{r}) \approx [\nabla \cdot \mathbf{A}(\mathbf{r})]_{\mathrm{sm}} = \frac{1}{\Psi} \int_{\Omega} \nabla' \cdot \left(\Psi' \mathbf{A}'\right) W \mathrm{d}\mathbf{r}' - \frac{\mathbf{A}}{\Psi} \cdot \int_{\Omega} \nabla' \Psi' W \mathrm{d}\mathbf{r}' \tag{2.43}$$

where, for instance, $\lambda' \equiv \lambda(\mathbf{r}')$. Integrating by parts and recalling equation (2.18) (or via the sample points argument) yields,

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \frac{1}{\Psi} \int_{\Omega} \Psi' \mathbf{A}' \cdot \nabla W \mathrm{d}\mathbf{r}' - \frac{\mathbf{A}}{\Psi} \cdot \int_{\Omega} \Psi' \nabla W \mathrm{d}\mathbf{r}' \tag{2.44}$$

Theorem 2.6 *For* $\mathbf{A}(\mathbf{r})$*, the smoothing approximation of the divergence can be redefined as*

$$[\nabla \cdot \mathbf{A}(\mathbf{r})]_{\text{sm}} = \frac{1}{\Psi(\mathbf{r})} \int_{\Omega} \Psi(\mathbf{r}') \left[\mathbf{A}(\mathbf{r}') - \mathbf{A}(\mathbf{r})\right] \cdot \nabla W \left(|\mathbf{r} - \mathbf{r}'|, h\right) \mathrm{d}\mathbf{r}' \tag{2.45}$$

Again, similar forms can be defined for other derivatives. Collectively, these are the **First-Order Smoothing Approximations**.

As a conceptual alternative, Price (2012) make the argument put forth in Theorem 2.6 by taking a Taylor series expansion of the zeroth-order smoothing approximation and subtract the first-order term from the original form. The former is favourable as differing SP equations are found by using different definitions of Ψ. Using the latter, the gradients within the true dynamics equations need expanding. For example, the divergence of velocity (required by the Continuity and Induction equations (2.99) and (2.101), respectively) is usually expanded by

$$\nabla \cdot \mathbf{v} = \nabla \cdot \mathbf{v} - \mathbf{v} \nabla 1 \tag{2.46}$$

or

$$\nabla \cdot \mathbf{v} = \frac{1}{\rho} \nabla \cdot (\rho \mathbf{v}) - \frac{\mathbf{v}}{\rho} \nabla \rho \tag{2.47}$$

The equivalent in the former scheme is $\Psi = 1$ and $\Psi = \rho$, respectively. Using either argument,

Theorem 2.7 *The smoothing approximation, substitution theorem and the first-order derivative smoothing approximation (Theorems 2.2, 2.5 and 2.6) can be used to replace the components in some dynamic equations until they are expressed in terms of the smoothing function,* W*, its derivative,* ∇W*, and the elements of the set* $G(\mathbf{r})$ *only.*

2.2.7 Discretisation Approximation

The descretisation approximation is the second of the two unavoidable approximations made in the creation of any SP algorithm. It is fundamental to their eventual application within a computational framework. At its simplest, in order to simulate

the medium, it must be divided into a number of constituent parts, N_p. These are typically termed the smoothed particles, centoids, interpolation points, pseudoparticles or, simply, particles. Paraphrasing Monaghan (2005), the psuedoparticles may be interpreted by mathematicians as interpolation points, or by physicists as material particles. These pseudoparticles are where information about the medium exists. The integral approximations (Theorems 2.2 and 2.6) must, therefore, be further approximated by a summation in order to be calculable.

Theorem 2.8 *Recognising that a mass element is defined by $m = \rho \mathrm{d}\mathbf{r}$, by substitution the direct transformation of the smoothing approximation, equation (2.7), at $\mathbf{r}_a$ yeilds*

$$\lambda(\mathbf{r}_a) \approx [\lambda(\mathbf{r}_a)]_{\mathrm{SP}} = \sum_{b=1}^{N_p} \frac{m(\mathbf{r}_b)}{\rho(\mathbf{r}_b)} \lambda(\mathbf{r}_b) W\left(|\mathbf{r}_a - \mathbf{r}_b|, h\right) \tag{2.48}$$

where a and b are indices identifying specific pseudoparticles. Equation (2.48) is typically termed the **Summation Approximation**. *A similar transformation of, for instance, the divergence smoothing approximation, equation (2.45), gives $\nabla \cdot \mathbf{A}(\mathbf{r}_a) \approx [\nabla \cdot \mathbf{A}(\mathbf{r}_a)]_{\mathrm{SP}}$ where*

$$[\nabla \cdot \mathbf{A}(\mathbf{r}_a)]_{\mathrm{SP}} = \frac{1}{\Psi(\mathbf{r}_a)} \sum_{b=1}^{N_p} \frac{m(\mathbf{r}_b)\Psi(\mathbf{r}_b)}{\rho(\mathbf{r}_b)} \left[\mathbf{A}(\mathbf{r}_b) - \mathbf{A}(\mathbf{r}_a)\right] \cdot \nabla_a W\left(|\mathbf{r}_a - \mathbf{r}_b|, h\right) \tag{2.49}$$

and $\nabla_a = \left[\frac{\partial}{\partial r_a^j}\right]_{j=1..\nu}$, where j denotes the dimensional components of the coordinate system. Note that ∇_a is the del operator, ∇, as computed in this discrete space (at $\mathbf{r}_a$).

The approximated value at some point $\mathbf{r}_a$ is the amalgom of the appropriate information held at points $\mathbf{r}_b$ (where $b = 1..N_p$) throughout the domain, Ω. However, the compact support of the smoothing function limits number of points that have influence over the value to those within the subdomain $\omega(h)$, see Figure 2.4. This represents a significant reduction in the number of interactions (and, therefore, calculations) required by the algorithm. It is useful, therefore, to express this within the SP equations.

Theorem 2.9 *Herein, define ξ as a set of all unique pseudoparticle indices,*

$$\xi = \{a \,|1 \leq a \leq N_p\} \tag{2.50}$$

Also, let ξ_a be a subset of ξ ($\xi_a \subset \xi$) defined by

$$\xi_a = \{b \,|b \in \xi \wedge |\mathbf{r}_a - \mathbf{r}_b| < Ch\} \tag{2.51}$$

That is, ξ_a is a set of all the pseudoparticles within the subdomain $\omega(h)$ centred on the $\mathbf{r}_a$. In addition, let $\xi_a^j \subseteq \xi_a \subset \xi$ where

$$\xi_a^j = \left\{b \,\middle|\, b \in \xi_a \wedge \left(r_a^j - r_b^j\right) \neq 0\right\} \tag{2.52}$$

These sets allow the approximations to be rewritten so that they reflect this reduction,

$$\sum_{b=1}^{N_p} \rightarrow \sum_{b \in \xi_a} \tag{2.53}$$

For the derivative approximations, given equation (2.17), the contribution of the a^{th} pseudoparticle to itself is always zero. Therefore, for derivative approximations only,

$$\sum_{b=1}^{N_p} \rightarrow \sum_{\substack{b \in \xi_a \\ b \neq a}} \tag{2.54}$$

2.2.8 Notes on Notation

There is a measure of standardisation in SP notation.

The lower case characters a, b, c, i, j, k, n, p, and q are always indices. The indices a, b, and c refer exclusively to pseudoparticle indices; i, j and k usually refer to dimensional components of vectors; and n refers only to the time-step. The indices p and q are used as indices generalising the others where appropriate. Indices occurring in superscript surrounded by chevrons imply an iterative process. As an example, $(X_a^i)^{<n>}$ refers to the value of X in the i^{th} dimension, of the a^{th} pseudoparticle at the n^{th} time-step. Other symbols and characters also have standardised meaning, and they will be introduced as required.

Equations are usually given in a reduced (collapsed) notation. At the simplest,

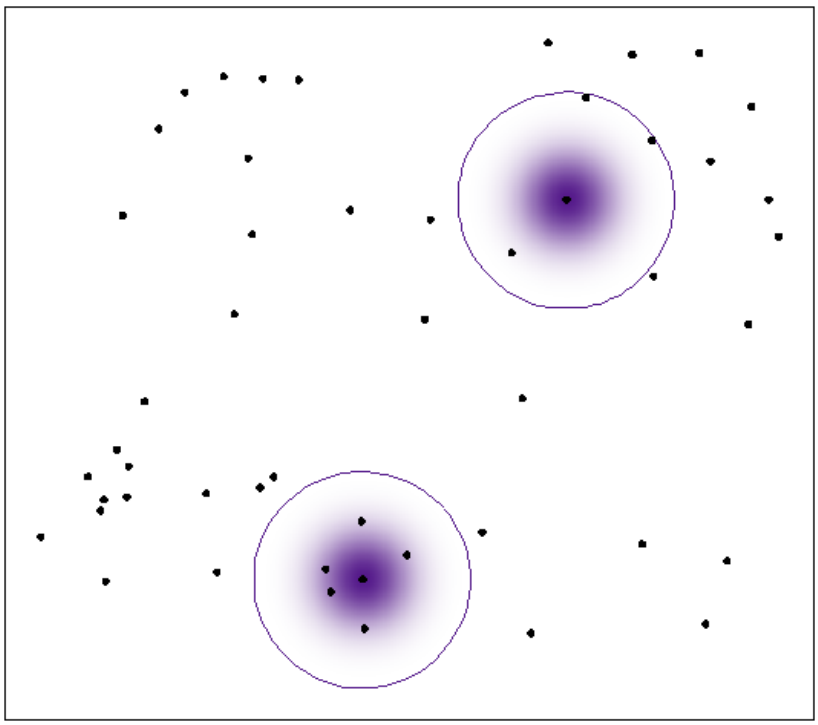

Figure 2.4: Visualisation, for some 2-Dimensional distribution of points, of just two of the pseudoparticles with their subdomains, ω. The purple lines are the surfaces of the subdomains, $S(\omega)$, which are defined by the compact support (Ch) of the smoothing functions, W. The purple haze indicates the relative value of the smoothing function for the two points. The greater the density of colour, the greater the value of the bell-shaped smoothing function.

for all pseudoparticle properties excluding position,

$$A_a = A(\mathbf{r}_a) \tag{2.55}$$

There are also significant shorthands,

$$A_{ab} = A_a - A_b \tag{2.56}$$

Unless $A = W$, in which case,

$$W_{ab} = W\left(|\mathbf{r}_{ab}|, h\right) \tag{2.57}$$

such that

$$\nabla_a^j W_{ab} = \frac{\partial}{\partial r_a^j}\left[W\left(|\mathbf{r}_{ab}|, h\right)\right] = \frac{r_{ab}^j}{|\mathbf{r}_{ab}|}\frac{\partial}{\partial |\mathbf{r}_{ab}|}\left[W\left(|\mathbf{r}_{ab}|, h\right)\right] \tag{2.58}$$

Note the use of equation (2.56) to compress the distance measure. In addition, the average of values of a property from two different pseudoparticles is given by

$$\overline{A_{ab}} = \frac{A_a + A_b}{2} \tag{2.59}$$

Summations, excluding those that result from the discretisation of the integrals in the smoothing approximations, are given in implicit form consistent with index notation. That is, any index that only occurs on one side of an equation implies that there is a summation over all valid values of that index. For instance, the divergence could be given by

$$\nabla \cdot \mathbf{A}_a \equiv \sum_j \nabla^j A_a^j \equiv \nabla^j \mathbf{A}_a^j \tag{2.60}$$

Note there is no summation over the a indices as they occur on both sides of the equal sign.

Bringing together all of the above reductive notation, the divergence approximation, equation (2.49), may be written in the equivalent form,

$$\nabla \cdot \mathbf{A}_a \approx \left[\nabla^j A_a^j\right]_{\mathrm{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a^j \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} A_{ba}^j \nabla_a^j W_{ab} \tag{2.61}$$

2.2.9 SP Identities

Given the SP reduced notation, and the completion of the approximations necessary for implementation of functions within a computational framework, listed herein are the five standard SP approximation identities.

Summation Approximation

$$\lambda \approx [\lambda_a]_{\mathrm{SP}} = \sum_{b \in \xi_a} \frac{m_b}{\rho_b} \lambda_b W_{ab} \tag{2.62}$$

Gradient Approximation

$$[\nabla \lambda_a]_{\mathrm{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} \lambda_{ba} \nabla_a W_{ab} \tag{2.63}$$

where the j^{th} component is

$$\left[\nabla^j \lambda_a\right]_{\text{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a^j \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} \lambda_{ba} \nabla_a^j W_{ab} \tag{2.64}$$

Divergence Approximation

Also called the inner product approximation,

$$\left[\nabla \cdot \mathbf{A}_a\right]_{\text{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} \mathbf{A}_{ba} \cdot \nabla_a W_{ab} \tag{2.65}$$

where the j^{th} element of the summation is

$$\left[\nabla^j A_a^j\right]_{\text{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a^j \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} A_{ba}^j \nabla_a^j W_{ab} \tag{2.66}$$

Tensor Derivative Approximation

Alternatively referred to as the outer product approximation,

$$\left[\nabla \otimes \mathbf{A}_a\right]_{\text{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} \mathbf{A}_{ba} \otimes \nabla_a W_{ab} \tag{2.67}$$

where each element is given by

$$\left[\nabla^j A_a^i\right]_{\text{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a^j \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} A_{ba}^i \nabla_a^j W_{ab} \tag{2.68}$$

Curl Approximation

The k^{th} component of $\nabla \times \mathbf{A}$ is given by

$$\left[(\nabla \times \mathbf{A}_a)^k\right]_{\text{SP}} = \epsilon^{kji} \left[\nabla^j A_a^i\right]_{\text{SP}} \tag{2.69}$$

where

$$\epsilon^{mpq} = \frac{(m-p)(p-q)(q-m)}{2} = \begin{cases} +1 & \text{even permutations of } mpq \\ -1 & \text{odd permutations of } mpq \\ 0 & \text{if any } m, p, \text{ or } q \text{ are equal} \end{cases} \tag{2.70}$$

is the Levi-Civita symbol (also referred to as the permutation, antisymmetric or alternating symbol).

These identities can be substituted into a set of ideal dynamics partial differential equations (PDEs) (e.g. dispersionless Navier-Stokes or Magnetohydrodynamics equations). Care should be taken, however, to ensure that the resulting system of equations is consistent with the Lagrangian and Hamiltonian (see Sections 2.5.3 and 2.5.4). If implemented in this way, the resulting algorithm will conserve linear and angular momentum (Section 2.5.3).

2.3 Density Approximation Basis

Rather than beginning with the integral basis (Section 2.2), the SP algorithm can be derived entirely from the definition of density estimate. There are many benefits to this approach. Primary among them is the clarity of the limited but necessary physical assumptions, quoting Price (2012);

i) That the time integrals and thus the time derivatives, $\frac{\mathrm{d}}{\mathrm{d}t}$*, are computed exactly (though this assumption can in principle be relaxed);*

ii) That the Lagrangian, and by implication the density and thermal energies are differentiable;

iii) That there is no change in entropy, such that the first law of thermodynamics $\mathrm{d}u = P\mathrm{d}V$ *is satisfied, and that the change in particle volume is given by* $\mathrm{d}V = -\frac{m}{\rho^2}\mathrm{d}\rho$*.*

However, that in order to analyse the resulting algorithm, either;

(a) The summations must revert to integrals by assuming a significantly high number density for the pseudoparticles and $m = \rho \mathrm{d}\mathbf{r}$; or

(b) Make additional assumptions pertaining to the behaviour of the smoothing function and other numerical factors.

These is not required by the integral basis as (a) and most (though not all) of the behaviour referenced in (b) is implicit – the assumed behaviour is explicit in Section 2.2.3.

A summary of the density basis for SP theory is presented herein. For a detailed and eloquent description see the most recent review of SP Hydrodynamics (SPH) and Magnetohydrodynamics (SPMHD), Price (2012).

2.3.1 Method Extrapolation

The foundation of the standard SP algorithm under this basis is the following question; Given some scattered point masses, how is the density calculated? Conceptually, the simplest method is to lay a grid over the domain inhabited by the points and by dividing the mass in each cell by the volume, define the density (Figure 2.5a). More complex methods are formed on this fundamental measure, for example the hybrid particle-cell methods Particle-In-Cell and Cloud-In-Cell, of Tskhakaya (2008) and Birdsall and Fuss (1968), respectively. However, these methods are liable to suffer, over simulated time, for the interpolation of properties between the cells and particles.

Alternatively, the grid can be removed by summing the masses within some sphere over the volume of that sphere, such that the density at the centre of the sphere can be calculated. At its simplest

$$\rho(\mathbf{r}) = \frac{\Gamma\left(\frac{\nu}{2}+1\right) M}{\pi^{\frac{\nu}{2}} R^{\nu}} \tag{2.71}$$

where $\Gamma(x)$ is the gamma function, M is the mass within the sphere, and R is the radius of the ν-Dimensional sphere about $\mathbf{r}$. See Figure 2.5b. Both of these methods, particularly the second, can induce significant fluctuations as the relatively distant points move in and out of the individual cell/sphere subdomains.

In order to remove this effect and define the SP density estimation (the third approach), consider that conceptually the sphere of influence continues to exist but the contribution from each individual mass is modified by a weighting function (the smoothing function W). The modification ensures that the further the point mass is from the point of interest, $\mathbf{r}_a$, the less of an influence it has over the computation of density. Recalling SP notation (Section 2.2.8),

$$\rho_a = \sum_{b \in \xi_a} m_b W_{ab} \tag{2.72}$$

This expression can also be found by setting $A = \rho$ in the summation approximation, equation (2.48), and is shown in Figure 2.5.

Consider Figure 2.6. Using the estimation methods, and assuming equal mass per point, the densities have been calculated at each point. Subsequently, the ratio of each density to the average density of the domain $\left(\frac{\text{Total Mass}}{\text{Domain Volume}}\right)$ is interpolated (by the same method, at the same resolution) over the domain. The resulting images show some of the disparities between the methods. The gridded method, Figure 2.6a, respectively over and under samples regions of the domain, spreading

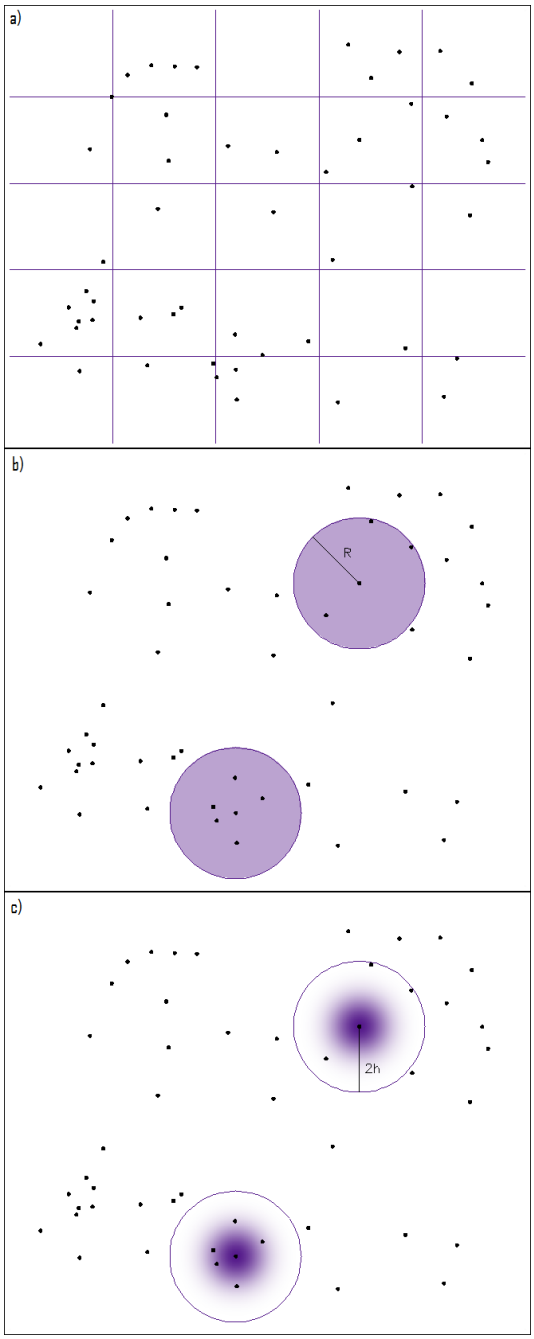

Figure 2.5: Representation of the three methods, presented herein, for estimating density from scattered point mass sources. The panels (top to bottom) show the gridded method, the sphere of influence method and the SP approach. Note that these visualisations must change if the smoothing length, $h(\propto R)$, were allowed to vary from point mass to point mass (see Chapter 4).

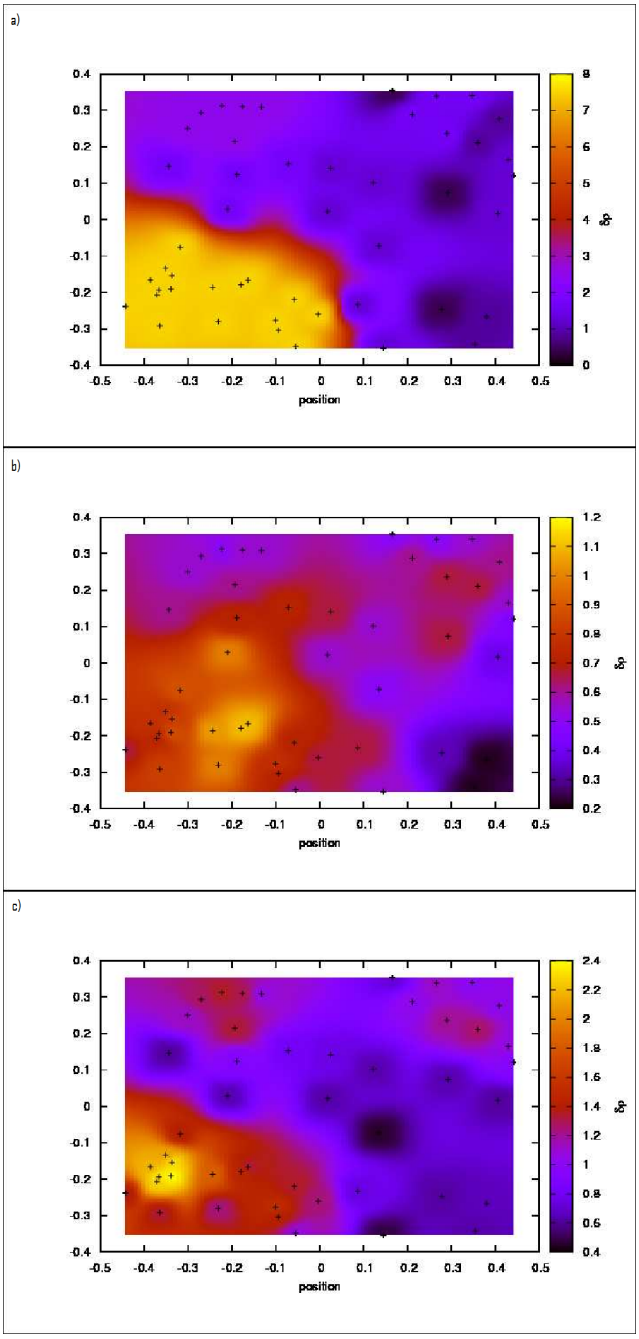

Figure 2.6: Plots showing how each of the estimation methods referenced in Figure 2.5 behave given some example distribution of point masses. Note that the distribution is the same as that in said figure. The density values are calculated, using the appropriate method, and then interpolated to produce the imaged density field via identical interpolation method. The order of the panels reflect the order in Figure 2.5 The upper two panels show significant errors. The first significantly over estimates the density in the lower left quadrant of the domain, and the second smooths out the influence of each individual point mass so that regions without significant clustering have the largest densities. The density field in the final (bottom) panel correlates well (by eye) to the point mass distribution.

the very high density over the lower left quadrant. The sphere sampling method overly smooths out the influence of each mass and the resulting densities deviate very little from the average. In addition, the minimum/maximum densities fail to correspond (by eye) to the highly clustered or sparse points, respectively. Finally, the densities produced by the SP estimation are much more suitable. The high/low points appear to correlate to the degree of clustering. The range in density ratio also appears to reflect the clustering in certain regions, particularly the highest density ratio (~ 2.4) relative to the high number density about its position.

2.3.2 Density Spatial Derivative

Given the definition of the density estimate, it is possible to define its derivative in space. Using the SP density estimate, equation (2.72),

$$\frac{\partial \rho_b}{\partial r_a^j} = \sum_{c \in \xi_b} m_c \frac{\partial W_{bc}}{\partial r_a^j} = \sum_{c \in \xi_b} m_c \left(\delta^{ab} - \delta^{ac} \right) \frac{r_{bc}^j}{|\mathbf{r}_{bc}|} \frac{\partial W_{ab}}{\partial r} \tag{2.73}$$

where $r = |\mathbf{r}_{ab}|$. Defining equation (2.73) allows for the definition of

$$\delta \rho_b = \sum_{c \in \xi_b} m_c \left(\delta^{ab} - \delta^{ac} \right) \delta r_a^j \frac{r_{bc}^j}{|\mathbf{r}_{bc}|} \frac{\partial W_{ab}}{\partial r} \tag{2.74}$$

by variational principles (which are discussed in greater detail in Section 2.5.3.

The subsequent application of the definition of density and the variation in density (equations (2.72) and (2.74), respectively), with the assumptions prefacing this Section (2.3), to the Lagrangian defines the equation of motion (SP acceleration equation). If there are remaining dynamics equations that require SP approximation, the appropriate forms can be found by first writing them in Lagrangian (conservative) form, and factoring in the aforementioned assumptions and definitions of density and density variation.

2.4 Higher-Order Spatial Derivatives

Higher-order derivatives present something of a problem for the usual SP theory, which is based around a bell-shaped, compact smoothing function.

Consider that the most direct approach is to define $\left[\nabla^2 \lambda_a\right]_{\mathrm{SP}}$ analytically by taking the derivative of the summation approximation, and subtracting the second

term of the Taylor series expansion, such that

$$\left[\nabla^2\lambda_a\right]_{\mathrm{SP}} = \sum_{b\in\xi_a}\frac{m_b}{\rho_b}\lambda_{ba}\nabla_a^2 W_{ab} \tag{2.75}$$

However, for $\frac{\partial^{<n>}}{\partial r_a^{j\,<n>}}$, consider that as n increases the gradients increase (generally) and, more importantly, fluctuate between positive and negative values more often. As a result, in order to stabilise the resulting approximation, there needs to be a sufficient sampling density about the smoothing function. That is not to say that h should increase such that a greater number of pseudoparticles sit within the subdomain, ω. Instead, the subdomain should encompass a greater extent of the pseudoparticles through the smoothing functions redefinition, at all levels, to a much higher-order approximation of the Gaussian, as the compact support extends further for high-order smoothing functions. So that, at the maximum order possible, the Gaussian smoothing function encompasses the entire domain to infinity.

Redefinition of the smoothing function requires a significant computational cost, as the set ξ_a expands greatly for each pseudoparticle. For instance, assuming a periodic hexagonal lattice of pseudoparticles and h set to $\frac{4}{5}$ of the pseudoparticle spacing, the lowest possible increase in smoothing function order (from M_4 cubic to M_5 quartic) causes the number of pseudoparticles in each set (the degree of the set) to increase from $|\xi_a| = 7$ to $|\xi_a| = 19$. Instead, W is often replaced (for higher-order derivatives only) by a new smoothing function Y (Brookshaw, 1985; Monaghan, 2005; Price and Monaghan, 2004a; Price, 2012), such that

$$\nabla_a^2 W_{ab} \;\rightarrow\; \nabla_a^2 Y_{ab} \equiv -\frac{2}{r}\frac{\partial W_{ab}}{\partial r} \tag{2.76}$$

where $r \equiv |\mathbf{r}_{ab}|$ and $\frac{\partial W_{ab}}{\partial r}$ is the scalar part of $\nabla_a^j W_{ab}$. The general second-order derivative is, therefore,

$$\left[\nabla^2\lambda_a\right]_{\mathrm{SP}} \approx 2\sum_{b\in\xi_a}\frac{m_b}{\rho_b}\frac{\lambda_{ab}}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}}{\partial|\mathbf{r}_{ab}|} \tag{2.77}$$

Additionally, more advanced expressions have been formulated in order to approximate $\nabla\cdot(\kappa\nabla\lambda)$, such as

$$\left[\nabla\cdot(\kappa_a\nabla\lambda_a)\right]_{\mathrm{SP}} \approx 2\sum_{b\in\xi_a}\frac{m_b}{\rho_b}\frac{\overline{\kappa_{ab}}\lambda_{ab}}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}}{\partial|\mathbf{r}_{ab}|} \tag{2.78}$$

or, alternatively, in order to produce smooth derivatives regardless of discontinuities

in κ,

$$[\nabla \cdot (\kappa_a \nabla \lambda_a)]_{\text{SP}} \approx 2 \sum_{b \in \xi_a} \frac{m_b}{\rho_b} \frac{\kappa_a \kappa_b}{\overline{\kappa_{ab}}} \frac{\lambda_{ab}}{|\mathbf{r}_{ab}|} \frac{\partial W_{ab}}{\partial |\mathbf{r}_{ab}|} \tag{2.79}$$

was proposed by Cleary and Monaghan (1999), and forms the basis of a number of dissipative terms.

2.5 SP Magnetohydrodynamics

It is not the purpose of this document to prove, or expand, magnetohydrodynamic (MHD) theory. Hence, presented herein is a discussion of MHD as it pertains to SP applications only. For further detail relating to MHD, refer to the works by E. R. Priest.

Magnetohydrodynamics (MHD) is a blending of the fluid equations from hydrodynamics (Navier-Stokes Equations) and Maxwell's equations from electromagnetic theory. The *standard* set of MHD equations consists of the following components;

The Mass Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2.80}$$

which ensures conservation of mass.

The Momentum Equation

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla P + \mathbf{J} \times \mathbf{B} + \rho \eta_{\mathbf{v}} \nabla^2 \mathbf{v} - \rho \mathbf{g} + \mathbf{F} \tag{2.81}$$

where $\eta_{\mathbf{v}}$ is the kinematic viscosity coefficient and $\mathbf{F}$ represent any additional external forces.

Ampère's Law

$$\mathbf{J} = \frac{1}{\mu_0} \nabla \times \mathbf{B} \tag{2.82}$$

where μ_0 is the magnetic permeability in a vacuum.

Faraday's Law

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{2.83}$$

Ohm's Law

$$\mathbf{E} + \mathbf{v} \times \mathbf{B} = \chi = \begin{cases} 0 & \text{Ideal MHD} \\ \eta_{\mathbf{B}}\mathbf{J} & \text{Resistive MHD} \\ \frac{1}{ne}\mathbf{J} \times \mathbf{B} & \text{Hall MHD} \\ \cdots & \text{More general forms} \end{cases} \tag{2.84}$$

where $\eta_{\mathbf{B}}$ is the resistivity, n is the charge-carrier density and e is the unit charge. χ varies as a function of the particular interpretation of the MHD equations.

The Solenoidal Condition

$$\nabla \cdot \mathbf{B} = 0 \tag{2.85}$$

which negates monopoles in the magnetic field.

Typically perceived as the *standard* or *basic* form, the resistive MHD equations define $\chi = \eta_{\mathbf{B}}\mathbf{J}$ (in Ohm's Law), such that substitution of Ohm's and Ampère's Laws into Faraday's Law, and subsequent simplification defines;

The Induction Equation (Resistive MHD)

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) + \frac{\eta_{\mathbf{B}}}{\mu_0}\nabla^2\mathbf{B} \tag{2.86}$$

2.5.1 Ideal MHD Equations

The MHD equations can be reduced in complexity to form the much simpler set of equations; the ideal (dissipationless) MHD equations – abbreviated here to the iMHD Equations.

Consider first the momentum equation (2.81). By assuming that viscous motion and external forces (including the gravitational effects) are negligible, the terms $\rho\eta_{\mathbf{v}}\nabla^2\mathbf{v}$, $\mathbf{F}$ and $-\rho\mathbf{g}$ drop out. Writing the acceleration as a total derivative and substituting in the definition of current, the momentum equation reduces to

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla P + \frac{1}{\mu_0}(\nabla \times \mathbf{B}) \times \mathbf{B} \tag{2.87}$$

By expanding the magnetic field term, the iMHD equation of motion is given by

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla P - \frac{\nabla\left(B^2\right)}{2\mu_0} + \frac{1}{\mu_0}\left(\mathbf{B} \cdot \nabla\right)\mathbf{B} = \nabla\mathbf{S} \tag{2.88}$$

such that $\mathbf{S}$ is the stress tensor defined by $S^{ij} = -P\delta^{ij} - \frac{1}{2\mu_0}B^2\delta^{ij} + \frac{1}{\mu_0}B^iB^j$.

The first three spatial gradient terms in equation (2.88) (ie. not $\nabla\mathbf{S}$) have clear physical interpretations.

i) $-\nabla P$ acts to smooth out fluid pressure gradients,

ii) Similarly, $-\frac{\nabla(B^2)}{2\mu_0}$ acts to smooth out magnetic pressure gradients, and

iii) $\frac{1}{\mu_0}(\mathbf{B}\cdot\nabla)\mathbf{B}$ acts to straighten magnetic field lines.

The induction equation (2.86) can also be simplified. Initially, by assuming that magnetic diffusion effects are negligible, the term $\frac{\eta_\mathbf{B}}{\mu_0}\nabla^2\mathbf{B}$ is dropped and the remaining term $\nabla\times(\mathbf{v}\times\mathbf{B})$ can be expanded to give

$$\frac{\partial\mathbf{B}}{\partial t} = (\mathbf{B}\cdot\nabla)\,\mathbf{v} - (\mathbf{v}\cdot\nabla)\,\mathbf{B} + \mathbf{v}\,(\nabla\cdot\mathbf{B}) - \mathbf{B}\,(\nabla\cdot\mathbf{v}) \tag{2.89}$$

Alternatively, this expression can be found by fixing $\chi = 0$ in Ohm's Law and reconstructing the induction equation (with the latter expansion of the vector triple product).

By recalling the solenoidal condition (2.85), rearrangement to form a total derivative gives the iMHD induction equation as

$$\frac{\partial\mathbf{B}}{\partial t} + (\mathbf{v}\cdot\nabla)\,\mathbf{B} = \frac{\mathrm{D}\mathbf{B}}{\mathrm{D}t} = (\mathbf{B}\cdot\nabla)\,\mathbf{v} - \mathbf{B}\,(\nabla\cdot\mathbf{v}) \tag{2.90}$$

Once again, the remaining spatial gradient terms can be interpreted physically, though the process is somewhat more involved.

i) Consider $(\mathbf{B}\cdot\nabla)\mathbf{v}$. Let $B_x \neq 0$, $v_x = v(t)$ and $B_y = B_z = v_y = v_z = 0$ such that

$$\frac{\mathrm{D}B_x}{\mathrm{D}t} \propto B_x\frac{\mathrm{d}v_x}{\mathrm{d}x} \tag{2.91}$$

Thus, if $\frac{\mathrm{d}v_x}{\mathrm{d}x} > 0$ then $\frac{\mathrm{D}B_x}{\mathrm{D}t} > 0$ (and vice versa). Thus, if some volume of fluid is compressed or decompressed parallel to the magnetic field lines, due to acceleration of the fluid, then $(\mathbf{B}\cdot\nabla)\mathbf{v}$ acts to enhance or weaken the $B_\parallel$, respectively.

ii) However, referring to $(\mathbf{B}\cdot\nabla)\mathbf{v}$ again, if the flow is accelerated perpendicular to the field, $(\mathbf{B}\cdot\nabla)\mathbf{v}$ would cause an increase in the field strength of $B_\perp$. Let $v_x = \text{constant}$, $B_x \neq 0$, $v_y = v(t)$ and $B_y = B_z = v_z = 0$, such that

$$\frac{\mathrm{D}B_x}{\mathrm{D}t} \propto B_x\frac{\mathrm{d}v_x}{\mathrm{d}x} \tag{2.92}$$

remains, but since $\frac{\mathrm{d}v_x}{\mathrm{d}x} = 0$, $\frac{\mathrm{D}B_x}{\mathrm{D}t} = 0$ as well. However,

$$\frac{\mathrm{D}B_y}{\mathrm{D}t} \propto B_x \frac{\mathrm{d}v_y}{\mathrm{d}x} \tag{2.93}$$

and if $\frac{\mathrm{d}v_y}{\mathrm{d}x} > 0$ then $\frac{\mathrm{D}B_y}{\mathrm{D}t} > 0$ (and vice versa).

iii) $-\mathbf{B}(\nabla \cdot \mathbf{v})$ acts to enhance or weaken the magnetic field if the fluid converges or diverges, respectively. Thus acting to increase the pressure exerted by the magnetic field such that the momentum equation can force the fluid to diverge, hence smoothing the pressure gradient. This term is analogous to the spatial derivative in continuity equation (2.80), which similarly acts to increase or decrease the density to force the fluid (through the momentum equation) to converge or diverge, respectively, in order to smooth out the *fluid* pressure.

Alternatively, several authors (Price and Monaghan, 2004a; Price, 2012) recommend constructing algorithms via the Lagrangian (conservative) formulation of the induction equation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{B}}{\rho}\right) = \left(\frac{\mathbf{B}}{\rho} \cdot \nabla\right)\mathbf{v} \tag{2.94}$$

In particular, this form is used in conjunction with algorithms derived via the density approximation basis.

Additional equations can be specified given the ideal (dispersionless) nature of the equation set. Specifically, the fluid pressure can be defined as the ideal equation of state,

$$P = (\gamma - 1)\rho u \tag{2.95}$$

where γ is the ratio of specific heats, and u is the specific internal energy per unit mass. Or

$$P = K(s)\rho^{\gamma} \tag{2.96}$$

where s is the entropy. Thus the iMHD energy equation can be defined by

$$\frac{\partial P}{\partial t} + \mathbf{v} \cdot \nabla P + \gamma P \nabla \cdot \mathbf{v} = 0 \tag{2.97}$$

Also note that, still using the ideal equation of state, for standard (resistive) MHD is given by

$$\frac{\partial P}{\partial t} + \mathbf{v} \cdot \nabla P + \gamma P \nabla \cdot \mathbf{v} = (\gamma - 1)\eta_{\mathbf{B}}|\mathbf{J}|^2 \tag{2.98}$$

Alternatively, the model could evolve either the internal energy per unit mass (specific internal energy), u, or the variable K defined by the alternative form of the ideal equation of state.

In summary, the iMHD equations, in a form that is useful for the derivation of SP theory, are;

The Mass Continuity Equation

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla\cdot\mathbf{v} \tag{2.99}$$

The Momentum Equation

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \nabla\mathbf{S}\left[= -\nabla P - \nabla\left(\frac{B^2}{2\mu_0}\right) + \frac{1}{\mu_0}\left(\mathbf{B}\cdot\nabla\right)\mathbf{B}\right] \tag{2.100}$$

The Induction Equation

$$\frac{\mathrm{D}\mathbf{B}}{\mathrm{D}t} = \left(\mathbf{B}\cdot\nabla\right)\mathbf{v} - \mathbf{B}\left(\nabla\cdot\mathbf{v}\right) \tag{2.101}$$

or

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{B}}{\rho}\right) = \left(\frac{\mathbf{B}}{\rho}\cdot\nabla\right)\mathbf{v} \tag{2.102}$$

The Solenoidal Condition

$$\nabla\cdot\mathbf{B} = 0 \tag{2.103}$$

Equation of State

$$P = (\gamma-1)\rho u = K(s)\rho^{\gamma} \tag{2.104}$$

The Energy Equation For example,

$$\frac{\partial P}{\partial t} + \mathbf{v}\cdot\nabla P + \gamma P\nabla\cdot\mathbf{v} = 0 \tag{2.105}$$

2.5.2 SP Application

A specific SP algorithm can be formulated by simple application of the SP identities (listed in Section 2.2.9) to the dynamics equations such that the system properties, tied to each pseudoparticle, can be evolved. The density can be evolved in a number of ways. The simplest is via the summation approximation (2.72), i.e. recalculating the density whenever required with

$$\rho_a = \sum_{b\in\xi_a} m_b W_{ab} \tag{2.106}$$

This form is consistent with both SP derivation bases, however if the domain utilises free surface boundaries (see Section 3.3) then the calculated density will fall off as

it approaches the boundary. In order to avoid this artefact, the density should be evolved by defining the SP continuity equation. By substitution into equation (2.99) of the divergence approximation,

$$\frac{\partial \rho_a}{\partial t} \approx \left[\frac{\partial \rho_a}{\partial t}\right]_{\mathrm{SP}} = -\rho_a \left[\nabla \cdot \mathbf{v}_a\right]_{\mathrm{SP}} \tag{2.107}$$

and thus,

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \frac{\rho_a}{\Psi_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} \tag{2.108}$$

Note the extension of SP notation, whereby all SP temporal derivatives are presented with using $\frac{\mathrm{d}}{\mathrm{d}t}$ regardless of the continuous form (partial, total, etc.). Thus $\frac{\mathrm{d}\rho_a}{\mathrm{d}t} \equiv \left[\frac{\mathrm{D}\rho_a}{\mathrm{D}t}\right]_{\mathrm{SP}}$.

By defining $\Psi = \rho$, the continuity equation may be rewritten as

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \sum_{\substack{b \in \xi_a \\ b \neq a}} m_b v_{ab}^j \nabla_a^j W_{ab} \tag{2.109}$$

which is identical to the form found by taking the derivative with respect to time of the summation approximation of density, equation (2.106). That is

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{b \in \xi_a} m_b W_{ab}\right) = \sum_{b \in \xi_a} m_b \frac{\mathrm{d}W_{ab}}{\mathrm{d}t} \tag{2.110}$$

as $\frac{\mathrm{d}m_a}{\mathrm{d}t} = 0$ and since

$$\begin{aligned} \frac{\mathrm{d}W_{ab}}{\mathrm{d}t} &= \frac{\mathrm{d}|\mathbf{r}_{ab}|}{\mathrm{d}t} \frac{\mathrm{d}W_{ab}}{\mathrm{d}|\mathbf{r}_{ab}|} \\ &= v_{ab}^j \frac{r_{ab}^j}{|\mathbf{r}_{ab}|} \frac{\partial W_{ab}}{\partial |\mathbf{r}_{ab}|} \\ &= v_{ab}^j \nabla_a^j W_{ab} \end{aligned} \tag{2.111}$$

by substitution, the rate of change of density is given by equation (2.109).

However, if $\Psi = 1$ and therefore

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \rho_a \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} \tag{2.112}$$

Colagrossi (2004a) showed that the estimation of density is more stable in situations

with large density ratios. Given his argument, the algorithm herein uses $\Psi = 1$.

Equation (2.112) (and density evolution via the continuity equation in general) has further advantages. In terms of implementation, using the summation approximation (2.106) would require two loops over the particles – the first to calculate the rates of change determining a change in position, and the second to recalculate the density. The continuity equation is itself a rate of change calculation and can, therefore, be computed in tandem with the others, hence reducing the computational cost. In addition, assuming the continuity equation is implemented, the user need not consider again the calculation of density when applying the code to alternate domains and initial conditions. Care must be taken, however, when implementing thermal conductivity (dissipation) – see Section 3.6.2 for details.

Consider now, the evolution of $\mathbf{B}$. Further direct substitution into the induction equation (2.101) yields

$$\frac{\mathrm{D}\mathbf{B}_a}{\mathrm{D}t} \approx \left[\frac{\mathrm{D}\mathbf{B}_a}{\mathrm{D}t}\right]_{\mathrm{SP}} = \left[(\mathbf{B}_a \cdot \nabla)\,\mathbf{v}_a\right]_{\mathrm{SP}} - \mathbf{B}_a\left[\nabla \cdot \mathbf{v}_a\right]_{\mathrm{SP}} \tag{2.113}$$

where the divergence of velocity is approximated in the same manner as the continuity equation. The first term appears more complex. Writing it using index notation, however, simplifies the required substitution, i.e.

$$\left[\frac{\mathrm{D}B_a^i}{\mathrm{D}t}\right]_{\mathrm{SP}} = \left[B_a^j \nabla^j v_a^i\right]_{\mathrm{SP}} - B_a^i\left[\nabla^j v_a^j\right]_{\mathrm{SP}} = B_a^j\left[\nabla^j v_a^i\right]_{\mathrm{SP}} - B_a^i\left[\nabla^j v_a^j\right]_{\mathrm{SP}} \tag{2.114}$$

Observing that each of the SP approximations is an element of the tensor (outer) product approximation, by rearrangement, the SP induction equation is

$$\frac{\mathrm{d}B_a^i}{\mathrm{d}t} = \frac{1}{\Psi_a}\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b\Psi_b}{\rho_b}\left(B_a^i v_{ab}^j - B_a^j v_{ab}^i\right)\nabla_a^j W_{ab} \tag{2.115}$$

The conservative induction equation (2.102) can also be simplified by index notation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{B_a^i}{\rho_a}\right) \approx \left[\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{B_a^i}{\rho_a}\right)\right]_{\mathrm{SP}} = \left[\frac{B_a^j}{\rho_a}\nabla^j v_a^i\right]_{\mathrm{SP}} = \frac{B_a^i}{\rho_a}\left[\nabla^j v_a^i\right]_{\mathrm{SP}} \tag{2.116}$$

and therefore

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{B_a^i}{\rho_a}\right) = \frac{B_a^j}{\rho_a\Psi_a}\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b\Psi_b}{\rho_b} v_{ba}^i \nabla_a^j W_{ab} \tag{2.117}$$

Note that the definition of Ψ should be consistent with the form taken defining it elsewhere, hence herein the SP induction equation is given by

$$\frac{\mathrm{d}B_a^j}{\mathrm{d}t} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b}\left(B_a^i v_{ab}^j - B_a^j v_{ab}^i\right)\nabla_a^j W_{ab} \tag{2.118}$$

The remaining iMHD equations could also be defined by direct substitution of the identities. Given the SP equation of state,

$$P_a = (\gamma - 1)\,\rho_a u_a \tag{2.119}$$

the equation of motion could be defined as

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = \frac{1}{\rho_a \Psi_a}\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} S_{ba}^{ij}\nabla_a^j W_{ab} \tag{2.120}$$

where $S_a^{ij} = -P_a\delta^{ij} - \frac{B_a^2}{2\mu_0}\delta^{ij} + \frac{1}{\mu_0}B_a^i B_a^j$.

However, this form fails to conserve, for instance, angular momentum and is inconsistent with the Lagrangian. The same is true of the total energy equation with respect to the Hamiltonian. It follows, therefore, that these SP equations should be treated separately.

2.5.3 SP Equation of Motion

In order to construct the SP equation of motion, also termed SP acceleration equation, first consider the systems Lagrangian,

$$L = E_T - E_V = \int\left(\frac{1}{2}\rho v^2 - \rho u - \frac{B^2}{2\mu_0}\right)\mathrm{d}\mathbf{r} \tag{2.121}$$

Through the discretisation approximation (Theorem 2.8) the Lagrangian can be expressed in SP form, i.e.

$$L \approx [L]_{\mathrm{SP}} = \sum_b m_b\left[\frac{1}{2}v_b^2 - u_b(\rho_b, s_b) - \frac{1}{2\mu_0}\frac{B_b^2}{\rho_b}\right] \tag{2.122}$$

2.5.3.1 Usual Derivation

At this point, most authors (Bonet and Lok, 1999; Monaghan and Price, 2001; Price and Monaghan, 2004b) make the following argument, minimising the action

$$S = \int [L]_{\mathrm{SP}} \, \mathrm{d}t \tag{2.123}$$

In this way, for small perturbation $\delta\mathbf{r}$,

$$\delta S = \int [\delta L]_{\mathrm{SP}} \, \mathrm{d}t = \int \left(\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{v}} \cdot \delta\mathbf{v} + \frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{r}} \cdot \delta\mathbf{r} \right) \mathrm{d}t = 0 \tag{2.124}$$

Given $\delta\mathbf{r} \equiv \frac{\partial \delta\mathbf{r}}{\partial t} + (\mathbf{v} \cdot \nabla)\, \delta\mathbf{r}$, expansion by integration by parts leads to

$$\delta S = \int \left\{ \left[-\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{v}} \right) + \frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{r}} \right] \cdot \delta\mathbf{r} \right\} \mathrm{d}t + \left[\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{v}} \cdot \delta\mathbf{r} \right]_{t_0}^{t} = 0 \tag{2.125}$$

Thus, assuming $[\delta L(t_0)]_{\mathrm{SP}} = [\delta L(t)]_{\mathrm{SP}} = 0$, in order for equation (2.125) to hold true,

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{v}_a} \right) - \frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{r}_a} = 0 \tag{2.126}$$

That is, the equation of motion must be formed solving the Euler-Lagrange equation. Price (2012) emphasised this argument claiming it made clear the assumptions that must be made in order to formulated the equation of motion. That is, the Lagrangian is differentiable – hence without discontinuous solutions – and that references to *exact conservation* of momentum and energy in fact mean that the errors result solely from numerical time integration, rather than the SP technique. However, the former is implicit in the process of deriving the equation of motion from the Euler-Lagrange equation (2.131), which is a recognised approach to dynamic systems analysis (Lanczos, 1986), and the latter is unnecessary, as any discretised continuous operation introduces errors and it remains, therefore, the responsibility of the author to show a component to be exact, rather than to show that it is not. In addition, there is a further, more troubling phenomena associated with this argument. In order to include the partial derivatives in equation (2.124), for some arbitrary variable $X(z)$,

$$\frac{\partial X}{\partial z} = \frac{\delta X}{\delta z} \tag{2.127}$$

However, consider

$$X(z + \delta z) = X(z) + \delta z \frac{\partial X(z)}{\partial z} + O(\delta z) \tag{2.128}$$

which rearranges to give

$$\frac{\partial X(z)}{\partial z} = \lim_{\delta z \to 0} \frac{X(z+\delta z) - X(z)}{\delta z} + O(\delta z) = \lim_{\delta z \to 0} \frac{\delta X(z)}{\delta z} + O(\delta z) \tag{2.129}$$

Therefore equation (2.124) should be rewritten as

$$\begin{aligned} 0 = \delta S &= \int \left[\left(\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{v}} + O(\delta \mathbf{v}) \right) \cdot \delta \mathbf{v} + \left(\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{r}} + O(\delta \mathbf{r}) \right) \cdot \delta \mathbf{r} \right] \mathrm{d}t \\ &\approx \int \left[\delta L\right]_{\mathrm{SP}} \mathrm{d}t = \int \left(\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{v}} \cdot \delta \mathbf{v} + \frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{r}} \cdot \delta \mathbf{r} \right) \mathrm{d}t \end{aligned} \tag{2.130}$$

Which, inaccurately (see the discussion of conservation later), implicates the SP equation of motion as an additional source of error in the system. In addition, it suggests that the non-existent error is compounded by the numerical temporal integration. This artefact leads to the argument presented herein.

2.5.3.2 Derivation

Begin with the assumption that the Lagrangian is differentiable (shockless), which is consistent with the application of iMHD equations, and present the general Euler-Lagrange equation

$$\frac{\partial L}{\partial q_k} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \dot{q}_k} \right) = 0 \tag{2.131}$$

where q_k is the k^{th} generalised coordinate of the system. For a smoothed particle system, the generalised coordinates are $\mathbf{r}_a$ (for each pseudoparticle a). Thus the equation of motion (singular, as indicated by the number of generalised coordinates) may be found by solving

$$\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{r}_a} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{v}_a} \right) = 0 \tag{2.132}$$

This task is be rather involved, and is presented explicitly in Appendix A. To summarise, the derivatives in the Euler-Lagrange equation (2.132) simplify to give

$$-\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{v}_a} \right) = -\frac{1}{2} \sum_b m_b \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial v_b^2}{\partial \mathbf{v}_a} \right) = -m_a \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} \tag{2.133}$$

and

$$\frac{\partial \left[L\right]_{\mathrm{SP}}}{\partial \mathbf{r}_a} = -\sum_b m_b \left[\frac{\partial u_b}{\partial \mathbf{r}_a} + \frac{1}{2\mu_0} \frac{\partial}{\partial \mathbf{r}_a} \left(\frac{B_b^2}{\rho_b} \right) \right] \tag{2.134}$$

The second requires further simplification. The first term of the summation can

be expanded by

$$\frac{\partial u_b}{\partial \mathbf{r}_a} = \left.\frac{\partial u_b}{\partial \rho_b}\right|_s \frac{\partial \rho_b}{\partial \mathbf{r}_a} \tag{2.135}$$

at constant entropy, s. The first law of thermodynamics dictates

$$\mathrm{d}U = m\mathrm{d}u = \mathrm{d}Q - \mathrm{d}W \tag{2.136}$$

where $\mathrm{d}Q = T\mathrm{d}s = 0$ is the heat added to the fluid body (zero as the entropy is constant), and $\mathrm{d}W = P\mathrm{d}V$ is the work done as the fluid diverges (or undergoes compression). Given that $V = \frac{m}{\rho}$, $\frac{\mathrm{d}V}{\mathrm{d}\rho} = -\frac{m}{\rho^2}$ such that by substitution equation (2.136) can be rewritten as

$$\mathrm{d}u|_s = -\frac{P}{\rho^2}\mathrm{d}\rho \tag{2.137}$$

and thus equation (2.135) becomes,

$$\frac{\partial u_b}{\partial \mathbf{r}_a} = \frac{P_b}{\rho_b^2}\frac{\partial \rho_b}{\partial \mathbf{r}_a} \tag{2.138}$$

where the $\frac{\partial \rho_b}{\partial \mathbf{r}_a}$ can be found analytically from the density summation, or via the continuity equation (2.112). The final term of the summation is rather more complicated, and its manipulation depends on the specific SP induction equation employed – once again, specifics can be found in Appendix A. Note however, that regardless of the form of SP induction equation the eventual SP equation of motion is identical. After substitution back into the Euler-Lagrange equation (2.132), and significant rearrangement,

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b}\left(\frac{\Psi_b}{\Psi_a}S_a^{ij} + \frac{\Psi_a}{\Psi_b}S_b^{ij}\right)\nabla_a^j W_{ab} \tag{2.139}$$

For the previously discussed definitions of Ψ,

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b}\left(S_a^{ij} + S_b^{ij}\right)\nabla_a^j W_{ab} \tag{2.140}$$

and

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = \sum_{\substack{b\in\xi_a \\ b\neq a}} m_b\left(\frac{S_a^{ij}}{\rho_a^2} + \frac{S_b^{ij}}{\rho_b^2}\right)\nabla_a^j W_{ab} \tag{2.141}$$

where $\Psi = 1$ and $\Psi = \rho$, respectively. Note that there are a number of alternative forms presented by various authors, see Price (2012) for a list, described by addi-

tional definitions of Ψ. For example, Hernquist and Katz (1989) suggested $\Psi = \frac{\sqrt{P}}{\rho}$ for the hydrodynamic case, such that

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = -2\sum_{\substack{b\in\xi_a \\ b\neq a}} m_b \frac{\sqrt{P_a P_b}}{\rho_a \rho_b} \nabla_a^i W_{ab} \tag{2.142}$$

The velocities, and therefore positions, of the pseudoparticles can be evolved. The reader should note, however, that transformation back in to continuous form shows that

$$\rho_a \frac{\mathrm{D}\mathbf{v}_a}{\mathrm{D}t} = -\nabla P + \frac{1}{\mu_0}\left(\nabla \times \mathbf{B}_a\right) \times \mathbf{B}_a + \frac{1}{\mu_0}\mathbf{B}_a \nabla \cdot \mathbf{B}_a \tag{2.143}$$

rather than the correct form, given by equation (2.87), which does not include the final term. In the continuum limit, $\nabla \cdot \mathbf{B} = 0$ and therefore equation (2.143) collapses to give the accurate form. However, this is not the case for a finite, discretised system. This issue returned to in Section 3.6.1.

2.5.3.3 Conservation of Momentum

The momentum conservation properties are simply proved. Given the reflective constraint (2.18) on the smoothing function, $\nabla W = -\nabla' W$, the linear momentum can be shown to be conserved by

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_a m_a \mathbf{v}_a\right) = \sum_a m_a \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_a}{\rho_b}\left(\mathbf{S}_a^{ij} + \mathbf{S}_b^{ij}\right)\nabla_a W_{ab} = 0 \tag{2.144}$$

In addition, the angular momentum is also conserved since

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_a \mathbf{r}_a \times m_a \mathbf{v}_a\right) &= \sum_a m_a \left(\mathbf{r}_a \times \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right) \\ &= \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b}\left(\mathbf{S}_a^{ij} + \mathbf{S}_b^{ij}\right)\mathbf{r}_a \times \nabla_a W_{ab} = 0\end{aligned} \tag{2.145}$$

Note that this behaviour further cements the reflective constraint as essential behaviour of SP algorithms.

2.5.4 SP Energy Equations

Discussed here are the two questions relating to an SP algorithm ability to model the flow of energy in a system. Namely,

- How is are the energy states evolved? and,
- Does the model conserve energy?

2.5.4.1 Energy Evolution

The objective of the SP energy equation is to determine the evolution of the thermal (internal) energy distribution such that the fluid pressure can be computed for the SP equation of motion (acceleration equation). There are three possible approaches.

i) The first, and simplest, is to evolve the specific internal energy (internal energy per unit mass), u, directly. It has already been established, for the SP equation of motion, that

$$\left.\frac{\partial u}{\partial \rho}\right|_s = -\frac{P}{\rho^2} \tag{2.146}$$

and therefore

$$\frac{\mathrm{d}u_a}{\mathrm{d}t} = \left.\frac{\partial u_a}{\partial \rho_a}\right|_s \frac{\partial \rho_a}{\partial t} = \frac{P_a}{\rho_a}\left[\nabla \cdot \mathbf{v}_a\right]_{\mathrm{SP}} = \frac{P_a}{\rho_a \Psi_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} \mathbf{v}_{ba} \nabla_a W_{ab} \tag{2.147}$$

However, assuming that the continuity equation (2.112) is implemented rather than the density summation, equation (2.72), then the rate of change of specific internal energy can be found must more efficiently by

$$\frac{\mathrm{d}u_a}{\mathrm{d}t} = -\frac{P_a}{\rho_a^2}\left[\frac{\partial \rho_a}{\partial t}\right]_{\mathrm{SP}} \equiv -\frac{P_a}{\rho_a^2}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} \tag{2.148}$$

By either method, the specific internal energy can be evolved and the fluid pressure be calculated by an appropriate equation of state. Most commonly,

$$P_a = (\gamma - 1)\,\rho_a u_a \tag{2.149}$$

ii) The most common approach is to evolve the total energy per unit mass, e, such that the specific internal energy can be determined by rearranging

$$e = \frac{v^2}{2} + u + \frac{B^2}{2\mu_0 \rho} \tag{2.150}$$

Formulated as the sum of the kinetic, internal and magnetic energy per unit mass. This is a useful derivation, regardless of its implementations, as its defi-

nition allows for the confirmation of energy conservation within SP algorithms. Note that this approach also holds in hydrodynamic models, but that the magnetic field terms are dropped. An explicit derivation is included in Appendix A.

Consider that the total energy of the system is defined by the Hamiltonian,

$$[E]_{\mathrm{SP}} = [H]_{\mathrm{SP}} = \sum_a \mathbf{v}_a \cdot \frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{v}_a} - [L]_{\mathrm{SP}} \tag{2.151}$$

as a funnction of the SP Langrangian (2.122). After simple, but lengthy, manipulation the total energy can be expressed as

$$[E]_{\mathrm{SP}} = \sum_a m_a \left(\frac{v_a^2}{2} + u_a + \frac{B_a^2}{2\mu_0 \rho_a} \right) \tag{2.152}$$

which, as expected, is the sum of all the energy stored at each pseudoparticle.

The rate of change in total energy, E, can be found by taking the comoving time derivative of equation (2.152), hence

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a m_a \left[\frac{1}{2} \frac{\mathrm{d}v_a^2}{\mathrm{d}t} + \frac{\mathrm{d}u_a}{\mathrm{d}t} + \frac{1}{2\mu_0} \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{B_a^2}{\rho_a} \right) \right] \tag{2.153}$$

By expanding the derivatives

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a m_a \left[\frac{1}{2} \frac{\mathrm{d}v_a^2}{\mathrm{d}\mathbf{v}_a} \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} + \frac{\mathrm{d}u_a}{\mathrm{d}t} + \frac{1}{2\mu_0 \rho_a} \frac{\mathrm{d}B_a^2}{\mathrm{d}\mathbf{B}_a} \frac{\mathrm{d}\mathbf{B}_a}{\mathrm{d}t} + \frac{B_a^2}{2\mu_0} \frac{\mathrm{d}\rho_a^{-1}}{\mathrm{d}\rho_a} \frac{\mathrm{d}\rho_a}{\mathrm{d}t} \right] \tag{2.154}$$

Thus, by substitution of the SP equation of motion (2.140), the SP specific internal energy equation (2.148), the SP induction equation (2.118) and the SP continuity equation (2.112), and after significant rearrangement, it is possible to define the rate of change of total energy per unit mass of each pseudoparticle as

$$\frac{\mathrm{d}e_a}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left(\frac{\Psi_b}{\Psi_a} S_a^{ij} v_b^i + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right) \nabla_a^j W_{ab} \tag{2.155}$$

such that

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{\mathrm{d}E_a}{\mathrm{d}t} = \sum_a m_a \frac{\mathrm{d}e_a}{\mathrm{d}t} \tag{2.156}$$

Note that the expansion above, equation (2.154), is required explicitly only for algorithms utilising the non-conservative induction equation (2.101). For the conservative induction equation (2.102), the required expansion would look a

little different. However the expressions for the rate of change resulting from the manipulations are identical.

From the SP total energy equation (2.155), the energy per unit mass, e_a, can be evolved and hence the specific internal energy, u_a, determined such that the pressure can be recalculated by the equation of state (2.149).

iii) As a final alternative, the entropy-conserving SP energy equation (Springel and Hernquist, 2002). This form proposes that, assuming an alternate equation of state,

$$P_a = K(s)\rho^\gamma \tag{2.157}$$

the variable K may be evolved. Refering to Price (2012), its evolution is determined by

$$\frac{\mathrm{d}K_a}{\mathrm{d}t} = \frac{\gamma-1}{\rho_a^{\gamma-1}}\left(\frac{\mathrm{d}u_a}{\mathrm{d}t} - \frac{P_a}{\rho_a^2}\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right) = \frac{\gamma-1}{\rho_a^{\gamma-1}}\left[\frac{\mathrm{d}u_a}{\mathrm{d}t}\right]_{\mathrm{diss}} \tag{2.158}$$

where $\left[\frac{\mathrm{d}u_a}{\mathrm{d}t}\right]_{\mathrm{diss}}$ refers to the dissipative component of the specific internal energy evolution. Hence, for the ideal case, $\frac{\mathrm{d}K_a}{\mathrm{d}t} \equiv 0$.

For monitoring purposes, the specific internal energy can be determined by

$$u_a = \frac{K_a\rho_a^{\gamma-1}}{\gamma-1} \tag{2.159}$$

and subsequently the total energy found by equation (2.150).

Price (2012) comments that, referring to this method as the *entopy-conserving* SP energy equation inaccurately implies that the other forms are not entropy conserving. In fact, assuming a consistent formulation of the energy equation utilised (ie. by one of the methods just described) and that the effects of a potentially variable smoothing length are properly included (see Chapter 4), all the energy equations conserve entropy.

It should also be noted that this form does not perform particularly well in non-ideal models, that is where

$$\left[\frac{\mathrm{d}u_a}{\mathrm{d}t}\right]_{\mathrm{diss}} \neq 0 \tag{2.160}$$

as any definition of the heat conduction will not be strictly physical, and will introduce errors into the system.

2.5.4.2 Conservation of Energy

As with the conservation of momentum, the reflective constraint on the smoothing function ensures that energy is conserved. Consider, again, equation (2.156). By substitution of the SP total energy equation (2.155),

$$\frac{\mathrm{d}\left[E\right]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \sum_b \frac{m_b}{\rho_b} \left(\frac{\Psi_b}{\Psi_a} S_a^{ij} v_b^i + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right) \nabla_a^j W_{ab} \tag{2.161}$$

and given the antisymmetry in the smoothing function derivative (the reflective constraint),

$$\frac{\mathrm{d}\left[E\right]_{\mathrm{SP}}}{\mathrm{d}t} = 0 \tag{2.162}$$

Thus energy is conserved.

2.6 Conclusion

To summarise, the SP theory has been constructed in full from the integral basis making two distinct approximations along the way; the smoothing approximation and the discretisation approximation. Note, however, that only the error due to smoothing, $\varepsilon_{\mathrm{sm}}$, has been estimated, and in that the noise in the system has also been established.

The fundamental components of the SPMHD algorithm have also been defined. Specifically the continuity equation (2.112), induction equation (2.118), SP equation of motion (2.140) and a range of possible SP energy equations (2.148), (2.155) and (2.158). Estimates have been formulated for the density (2.72) and higher order spatial derivatives.

However, there has been no discussion of the implementation of the algorithm. This is the subject of the following Chapter.

Chapter 3

Smoothed Particle Implementation

Contents

3.1 Introduction

Discussed in this Chapter are various facets of the implementation of the SPMHD algorithm as described in the previous Chapter (2). In addition the numerical (rather than strictly physical) modifications to the algorithm are also discussed. The Chapter's structure is intended to reflect the structure of the programs themselves. Very generally, the program is split into two *phases* – the *Initialisation Phase* that is responsible for preparing the domain and simulated plasma, and the *Simulation Phase* which is responsible for the modelled evolution of the plasma. Sections 3.2 and 3.3 discuss elements of the Initialisation Phase, the unitisation procedure and boundaries, respectively. The two subsequent Sections (3.4 and 3.5) concern the accuracy and pace of the modelled evolution by discussing respectively the numerical temporal integration and the optimisation of SP approximation calculations. The final Section 3.6 focuses on the specific numerical modifications of the SPMHD algorithm required to refine the simulation. Specifically discussed are the procedures stabilising the system when subjected to strong negative stress, ensuring the solenoidal constraint and that the model is able to capture MHD shocks.

First, consider the parameters that constrain the implementation. The program is written as a serial code in Fortran 90/95, with various imaging and analysis routines including the PGPLOT library, written in Fortran 77, Gnuplot, Java and bash script. Initially conceived as an extension of an incompressible fluid SPH (serial) code constructed for Knight (2008), it quickly became evident that the program as written required significant modification in order to run in a practical period of time. This lead to a refactorisation of the code, which was streamlined and ran in a much more reasonable period of time. However, in order to achieve this increased pace in a serial framework, the code was *restricted to test-bed problems only.* The code performs this task well, but has by necessity restricted the possible applications and therefore investigations conducted. Conversely, this arrangement has made possible investigations of the nature of the SP approximations – investigations that would have been infeasible in the a restricted time frame (particularly the research in Chapter 5) if implementation were attempted in a parallel context.

3.1.1 Generalised Code Structure Diagrams

In order to provide some wider context for some of the implemented (or implementable) components discussed in this and later sections, presented here are general code structure diagrams. These are flow charts, depicting the interplay between the different sections/components/tasks within the SPMHD program, without the need for pseudocode or quoted code. As a primary example, and to provide the

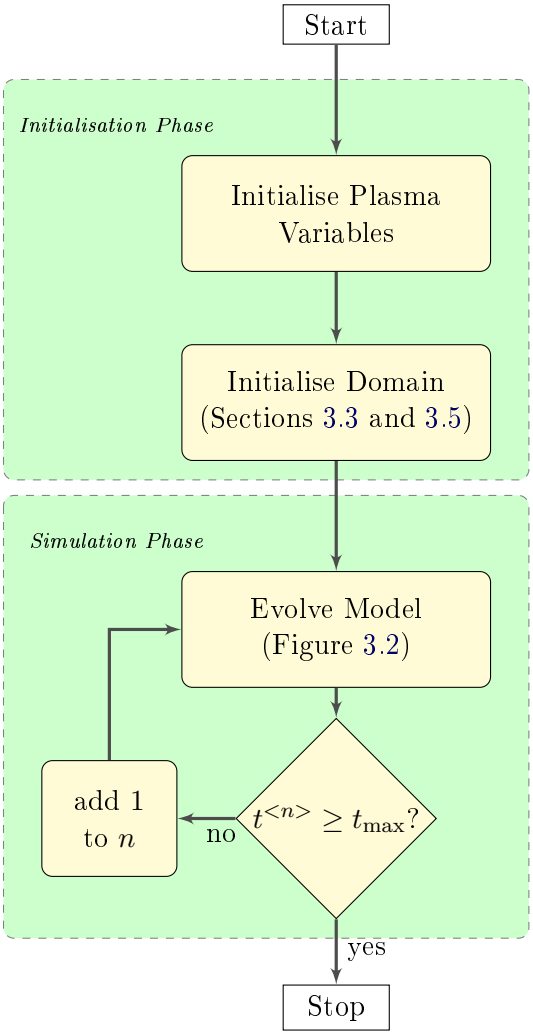

Figure 3.1: A generalised code structure diagram showing the global behaviour of the SPMHD programs.

global context for the program as previously described, see Figure 3.1.

Each diagram will conform to a set pattern. Above and below the main body are the source and sink of the information flow (i.e. where the data comes from, and where the manipulated data goes to). Within the main body of the image, the specific component nodes are given within frames that give some more general context. For instance, in Figure 3.1, the specific node *Initialise Domain* is framed within the *Initialisation Phase* of the program.

As a further example, consider Figure 3.2. Here, the node *Evolve Model*, within the *Simulation Phase* of Figure 3.1, is shown in more detail. The node is broken down into constituent parts. In this case, simply separating the calculation of

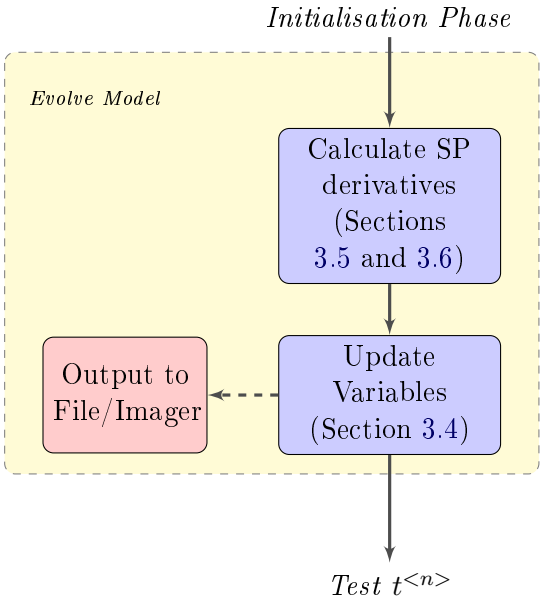

Figure 3.2: A generalised code structure diagram showing the basics that define how the model is evolved from one time step to the next.

SP derivatives, the evolution of variables and the output of data. More advanced examples of the same node can be found in Section 3.4.

3.2 Unitisation

It can be useful, especially in simulations where the values of the properties are particularly disparate, to unitise (that is, make dimensionless) the code. This ensures that the properties are of $O(1)$, and therefore mitigate the effects of truncation error. As a by-product, the equations are also slightly simplified. Consider that a macroscopic variable A can be decomposed so that

$$A = A_0 \hat{A} \tag{3.1}$$

where $\hat{A}$ is the unit value of A, and A_0 is a constant conversion factor.

There are three fundamental conversion factors of which every other conversion factor is a function. They extend from the three fundamental scales: length, time,

and mass. The conversions are

$$t = t_0\hat{t} \tag{3.2}$$

$$m = m_0\hat{m} \tag{3.3}$$

$$\mathbf{r} = l_0\hat{\mathbf{r}} \tag{3.4}$$

Given these factors, calculus operators can also be decomposed into a conversion factor (which will be some function of l_0, t_0 and/or m_0) and an operator applicable to the unit variables. So, given these new identities

$$\nabla = \frac{1}{l_0}\hat{\nabla} \tag{3.5}$$

$$\int \mathrm{d}\mathbf{r} = \frac{1}{l_0^\nu}\int \mathrm{d}\hat{\mathbf{r}} \tag{3.6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{1}{t_0}\frac{\mathrm{d}}{\mathrm{d}\hat{t}} \tag{3.7}$$

the conversion factors of the non-fundamental properties can be found, and the iMHD equations rewritten to function with the unit variables. For example, conversion factors would include

$$v_0 = \frac{l_0}{t_0} \tag{3.8}$$

$$\rho_0 = \frac{m_0}{l_0^\nu} \tag{3.9}$$

$$u_0 = \left(\frac{l_0}{t_0}\right)^2 \tag{3.10}$$

$$P_0 = \frac{m_0}{l_0^{(\nu-2)}t_0^2} \tag{3.11}$$

$$B_0 = \sqrt{\frac{\mu_0 m_0}{l_0^{(\nu-2)}t_0^2}} \tag{3.12}$$

where ν remains the number of dimensions. Note the unavoidable inclusion of μ_0 in B_0, and the consequential elimination of the constant from the dynamics equations. For example, Ampère's law can be rewritten as

$$\hat{\mathbf{J}} = \hat{\nabla} \times \hat{\mathbf{B}} \tag{3.13}$$

and the induction equation (2.101) becomes

$$\frac{\mathrm{D}\hat{\mathbf{B}}}{\mathrm{D}\hat{t}} = (\hat{\mathbf{B}} \cdot \hat{\nabla})\hat{\mathbf{v}} - \hat{\mathbf{B}}(\hat{\nabla} \cdot \hat{\mathbf{v}}) \tag{3.14}$$

Clearly, this simplification is transmitted through the derivations in the previous Chapter (2), and therefore simplifies the SP equations that form the algorithms.

3.3 Boundaries

There are various boundaries, both within and defining the domain, Ω, that could form part of an SP algorithms implementation.

3.3.1 Internal Boundaries

The following three concern the simpler boundaries that require little to no alteration of the standard implementation, thus occurring within the domain Ω as defined by the amalgam of all the pseudoparticle subdomains, ω_a. All three are implemented (where required) herein.

3.3.1.1 Free Surface Boundaries

Assuming an implementation that has not taken any boundaries into account, and that the continuity equation (2.112) has been implemented, then free surface boundaries have been implicitly included. This is one of the key benefits of the SP technique over meshed methods that typically require additional, complex routines in order to discretise and evolve a free surface boundary.

3.3.1.2 Discontinuities

Discontinuities in the initial conditions can also be included without altering the algorithm implementation. However, discontinuities must be *smoothed out* in order to conform to the smoothing and discretisation approximation. Assuming the discontinuity exists in the j^{th} dimension, between some *left* and *right* regions, the property A is initialised using the function

$$A_a = \frac{A_{\text{left}} + A_{\text{right}} e^{\left(\frac{r_a^j - r_0}{\Delta}\right)}}{1 + e^{\left(\frac{r_a^j - r_0}{\Delta}\right)}} \tag{3.15}$$

where r_0 is the position of the discontinuity, and Δ is the pseudoparticle spacing.

However, in order to accurately capture the evolution of the shock, the implementation should include the dissipation terms discussed in Section 3.6.2.

3.3.1.3 Periodic Boundaries

Fundamentally, periodic boundaries constrain the pseudoparticle positions within the box of the domain. They occur in opposing pairs such that, assuming the boundaries are applied in the j^{th} dimension, the positions conform to $r_{\text{L}}^j \leq r_a^j \leq r_{\text{U}}^j$, where r_{L}^j and r_{U}^j are the position of the lower and upper boundaries, respectively. These boundaries are the final form that do not require any ghost/boundary particles. They do, however, require slight modification of the implementation in two key places.

i) Subsequent to the numerical integration modification of position (see Section 3.4.1), the positions must be forced to remain within the upper and lower bounds. That is

$$\left(r_a^j\right)_{\text{new}} = \begin{cases} \left(r_a^j\right)_{\text{old}} - \left(r_{\text{U}}^j - x_{\text{L}}^j\right) & \text{if} \quad \left(r_a^j\right)_{\text{old}} > r_{\text{U}}^j \\ \left(r_a^j\right)_{\text{old}} + \left(r_{\text{U}}^j - x_{\text{L}}^j\right) & \text{if} \quad \left(r_a^j\right)_{\text{old}} < r_{\text{L}}^j \\ \left(r_a^j\right)_{\text{old}} & \text{otherwise} \end{cases} \tag{3.16}$$

This, however, is not the constraint as implemented. Instead, to optimise the procedure so that no conditional statements are required, the domain boundaries and other positions are centred about zero. As a result, the boundary may be enforced by

$$\left(r_a^j\right)_{\text{new}} = \left(r_a^j\right)_{\text{old}} - 2D^j I\left[\frac{\left(r_a^j\right)_{\text{old}}}{D^j}\right] \tag{3.17}$$

where D^j is the distance from the centre of the domain to the positive boundary edge (which is identical to the position of that boundary), and $I(x)$ is a function that truncates x, leaving only the integer component remaining.

ii) When calculated, the distance between two pseudoparticles r_{ab}^j must be checked to see that $r_{ab}^j < D^j$. If it fails this test then it should be reset such that

$$\left(r_{ab}^j\right)_{\text{new}} = 2D^j - \left(r_{ab}^j\right)_{\text{old}} \tag{3.18}$$

Note that once the optimisations discussed in Section 3.5 are implemented, this second modification becomes superfluous.

3.3.2 Domain Edge Boundaries

These boundaries are altogether more complex and add error to the system.

Recall the derivation of the SP approximations. For both the summation and derivative approximations the domain boundary effects were ignored. This occurred implicitly for the summation approximation in the form of the smoothing function constraints. Those definitions only hold within the domain. For instance, recall the specific equation (2.9), the normalisation constraint

$$\int_{\Omega} W \mathrm{d}\mathbf{r}' \equiv 1 \tag{3.19}$$

Consider that, if the extent of the smoothing function intersects the surface of the domain, $S(\Omega)$, then the normalisation constraint does not hold. In fact it would calculate an ever decreasing figure as the centre of the smoothing function moves closer to the surface and an ever increasing proportion of the smoothing function domain, $\omega(h)$, exists beyond $S(\Omega)$. This is what causes the drop off in the density estimate (2.72) previously described as the reason for implementing the continuity equation (2.112) as a preferred method for density evolution. The effect is explicit in the integral basis derivation of smoothed derivative approximations (Section 2.2.6), as the dropping of the surface integral from equation (2.36).

Considerable research has been done on the subject of correcting for these *forgotten* components leading to, among many other techniques, the Normalised Corrected SPH algorithm (Vila, 1999; Vignjevic, 2004).

Presented here are the types of edge boundary rather than the numerous techniques for implementing them. They are constructed in a simple fashion using extra sets of particles that behave in various different ways in order to constrain the simulations.

The following boundaries all require the addition of significant memory to store the properties of ghost (or boundary) particles. These are created simultaneously with the initial plasma conditions and factored into the algorithm at the same time. The ghost particle positions define the extent of the domain, Ω, by creating a barrier (or barriers) that hold the pseudoparticles.

3.3.2.1 Solid Boundaries

A solid boundary, non-physical in the context of the Solar Corona, is formed using an infinite potential generated by the ghost particles. The pseudoparticles are bound

within the domain by adapting the SP equation of motion such that

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \left[\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right]_{\text{fluid}} + \sum_{g=1}^{N_g} \Gamma^i(\mathbf{r}_{ag}) \tag{3.20}$$

where $\Gamma^i(\mathbf{r}_{ag})$ is a boundary function that repels any approaching pseudoparticles, in the i^{th} dimension, and N_g is the number of ghost particles. As a pseudoparticle approaches the ghost particle, the repulsive force steadily increases until $\Gamma(0) \approx \infty$ There are several different versions of this form of boundary, discussed in most SP Hydrodynamics papers pertaining to incompressible fluids, as the simulated medium (typically water) must be retained within some walled structure. Further discussion can be found in Knight (2008).

3.3.2.2 Background Boundaries

This form of boundary creates a set of ghost particles initialised identically to the regular pseudoparticles. However, their properties (including position) are not allowed to evolve as a function of the SP derivatives, but are included as part of the sets ξ_a. As long as the ghost particles are not required to evolve (i.e. no information is past to them) these boundaries perform well.

An adapted form of this boundary is presented in Chapter 6. This is also the only domain edge boundary implemented herein.

3.3.2.3 Data Boundaries

This final form is somewhat esoteric. Notionally, a barrier of ghost particles is created but the properties (other than position) are evolved according to some external numerical or real world data input. In this way, an external driver can be factored into the model.

In the context of a model of the Solar Corona, real world remotely sensed data about the Photosphere and/or Chromosphere can be factored into model geometry as a lower boundary driving the simulated dynamics above.

3.4 Time-stepping

The following sections answer the questions:

- How are the pseudoparticles (plasma) properties evolved over simulated time?
- How can the procedure be optimised, such that an acceptable compromise is found between real world run-time and the accuracy of the results?

3.4.1 Numerical Integration

Numerical integration, in this context, is the procedure by which the required properties of the system are evolved. That is, the modification of variables (lacking analytical expressions describing there relationship to others in the system), A, over simulated time. A is given by the set

$$A = \{\mathbf{r}, \mathbf{v}, \mathbf{B}, \rho, E, D\} \tag{3.21}$$

where D represents a possible evolved dissipation parameter, and E should be supplemented by either e, u or K, depending on the implemented energy equation. Herein, therefore, $E = u$. Also note that, should the smoothing length be allowed to vary by some method whereby an analytical or approximate value of $\frac{\mathrm{d}h}{\mathrm{d}t}$ is known, it should be included as well (see Chapter 4).

In order to understand the fundamental nature of numerical integration, consider the simplest, first-order procedure,

$$A^{<n+1>} = A^{<n>} + \delta_t \frac{\mathrm{d}A^{<n>}}{\mathrm{d}t} \tag{3.22}$$

where δ_t is the time step, and $A^{<n>}$ refers to the value of A at the n^{th} iteration of the routine. In this way, the variables may be evolved, with an error $O(\delta_t)$ and therefore

$$\lim_{\delta_t \to 0} \left[A_a^{<n>}\right]_{\text{SP}} = A_a(t) \tag{3.23}$$

where $t = t^{<0>} + n\delta_t$.

As previously described (Sections 2.5.3 and 2.5.4), and in the literature surrounding the SP theory (Monaghan, 2005; Price, 2012) the exact conservation of energy and momentum by SP approximations are only as *exact* as the temporal numerical integration will allow. Thus δ_t should be kept small. However, consider that the only way to remove the error is to set $\delta_t = 0$. It is impossible, therefore, to evolve the system without error. In addition, in order to create a usable amount of data (ie. over a large enough range of simulated time) in a reasonable period of (real) time δ_t must be of a reasonable magnitude. Thus, a higher-order procedure should be employed to reduce the error without reducing the time step.

Listed below are several second-order methods (therefore with error $O(\delta_t^2)$) commonly employed in SP algorithms.

i) The modified Euler method, also referred to as the midpoint method,

$$A^{<*>} = A^{<n>} + \frac{\delta_t}{2}\frac{\mathrm{d}A^{<n>}}{\mathrm{d}t} \tag{3.24}$$

$$A^{<n+1>} = A^{<n>} + \delta_t\frac{\mathrm{d}A^{<*>}}{\mathrm{d}t} \tag{3.25}$$

where $A^{<*>}$ is a pseudo-half-step. This method requires memory proportional to $N_p\left(|A| + 2N_v\right)$, where N_v is the number of properties stored at each pseudoparticle and $|A|$ is the degree (size) of set A. It also requires the computation of the SP derivatives expressions twice (sequentially).

ii) The leapfrog method,

$$A^{<n+1>} = A^{<n-1>} + 2\delta_t\frac{\mathrm{d}A^{<n>}}{\mathrm{d}t} \tag{3.26}$$

Unlike the modified Euler method, this requires only a single computation of the SP derivatives. However, it also requires a greater amount of memory, in the form of an extra stored time-step, $N_p\left(|A| + 3N_v\right)$ – or alternatively extra computation and a more complex implementation, that will not allow for simple modification to alternate methods.

iii) The predictor-corrector method, also referred to as the improved Euler, Heun, or Crank-Nicolson (trapezoidal) method,

$$A^{<*>} = A^{<n>} + \delta_t\frac{\mathrm{d}A^{<n>}}{\mathrm{d}t} \tag{3.27}$$

$$A^{<n+1>} = A^{<n>} + \frac{\delta_t}{2}\left(\frac{\mathrm{d}A^{<n>}}{\mathrm{d}t} + \frac{\mathrm{d}A^{<*>}}{\mathrm{d}t}\right) \tag{3.28}$$

This form requires equivalent computation as the modified Euler method (twice calculated SP derivatives), but greater storage. The results of both SP derivative calculations must be simultaneously recorded, hence $2N_p\left(|A| + N_v\right)$. It is possible, depending upon the specific SP routines employed, that this memory increase can be avoided. However, this will once again constrain the adaptability of the algorithm.

Higher-order Runge-Kutta methods could also be applied. However, given the significant computational cost of SP approximations, it is necessary to strike a balance between the accuracy of the results and the time taken to create them. If the implementation requires any measure of adaptability, the possible procedure is further constrained. Consider that in the development phase of an algorithm, any modification may significantly increase memory and time requirements prior to

optimisation and the numerical integration with multiple steps will contribute to that pressure. Finally, owing to the serial limitations discussed in Section 3.1, low memory is also a significant consideration. Hence, within the SPMHD algorithm, the modified Euler (or midpoint) method is employed.

Consider that, assuming the properties are written out as often as is necessary, the implementation only requires the properties of each pseudoparticle in two states; at half- and full-step. Labelling these states 1 and 2 respectively, oscillating between the two values allows a single equation to express both equation (3.24) and equation (3.25). That is,

$$A^{<s>} = A^{<2>} + \frac{\delta_t}{3-s}\frac{\mathrm{d}A^{<3-s>}}{\mathrm{d}t} \tag{3.29}$$

where $s = 1..2$. This form facilitates the implementation as depicted in Figure 3.3.

3.4.2 Variable Time-step

In order to optimise the performance of the model, it is logical to leap in simulated time as far ahead as possible without letting the numerical dominate the physical. Notionally, the time step should be recalculated (as often as necessary) such that the fastest possible physical transfer of information in the system is captured over the resolution.

In practice, the fastest possible physical transfer of information is a fast MHD wave with magnitude

$$|\text{wave}| = \frac{1}{2}\sqrt{c^2 + \frac{B^2}{\mu_0\rho} + \frac{2cB^x}{\sqrt{\mu_0\rho}}} + \frac{1}{2}\sqrt{c^2 + \frac{B^2}{\mu_0\rho} - \frac{2cB^x}{\sqrt{\mu_0\rho}}} \tag{3.30}$$

assuming the wave traverses the x-axis in a static medium, and where c is the sound speed defined by

$$c^2 = \frac{\partial P}{\partial \rho} = \frac{P}{\rho} \tag{3.31}$$

As such, the maximum signal velocity between two pseudoparticles a and b (i.e. the speed fastest possible transfer of information along the line-of-sight between the pseudoparticles) is given by

$$v_{\mathrm{sig},t,ab} = \tilde{v}_a + \tilde{v}_b + \beta\left|\mathbf{v}_{ab}\cdot\frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}\right| \tag{3.32}$$

where β is a numerical parameter, usually defined such that $\beta \approx 1$ (in the case of

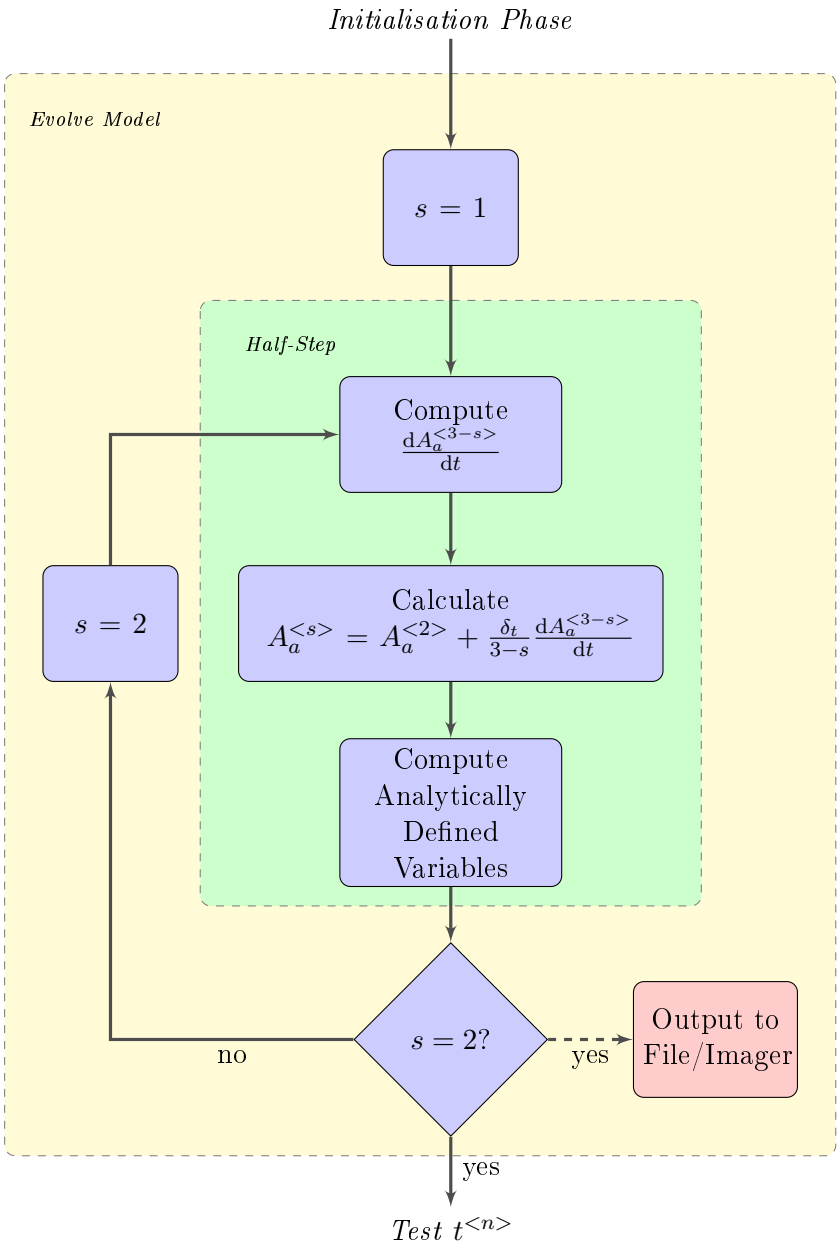

Figure 3.3: A generalised code structure diagram showing the basics that define how the model is evolved from one time step to the next using the modified Euler method.

certain shock capture schemes $\beta = 2$), and $\tilde{v}_a$ is given by

$$\tilde{v}_a = \frac{1}{2}\sqrt{c_a^2 + \frac{B_a^2}{\mu_0\rho_a} + \frac{2c_a}{\sqrt{\mu_0\rho_a}}\mathbf{B}_a \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}} + \frac{1}{2}\sqrt{c_a^2 + \frac{B_a^2}{\mu_0\rho_a} - \frac{2c_a}{\sqrt{\mu_0\rho_a}}\mathbf{B}_a \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}} \quad (3.33)$$

The subscript t in the notation defining the signal velocity communicates that this is the signal velocity as required by the calculation of the time-step. Also note that μ_0 is dropped once the variables become unitised (see Section 3.2).

Given this, and the limit imposed by the forces the pseudoparticles are subjected to, the time step can be defined such that

$$\delta_t = \min\left(\delta_{t,c}, \delta_{t,f}\right) \quad (3.34)$$

constructed as a function of the Courant-Friedrichs-Lewy (CFL) condition and the forces limited time step, equations (3.35) and (3.36) respectively.

$$\delta_{t.c} = C_{\mathrm{CFL}} \min_{\substack{a=1..(N_p-1) \\ b=(a+1)..N_p}} \left(\frac{\overline{h_{ab}}}{v_{\mathrm{sig},t,ab}}\right) \quad (3.35)$$

where $C_{\mathrm{CFL}} \approx \frac{4}{5}$.

$$\delta_{t,f} = \sqrt{\min_{a=1..N_p}\left(\frac{h_a}{|\mathbf{a}_a|}\right)} \quad (3.36)$$

where $\mathbf{a}_a = \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}$.

Equation (3.35) must be computed over all unique, non-zero particle interactions and in terms of implementation, therefore, requires the same loops over the pseudoparticles. However, it need only be calculated once, as a function of the pseudoparticle properties at the n^{th} time step (rather than requiring the pseudo-half-step properties as well). Thus it may be implemented in simultaneously with the SP derivative calculations, see Figure 3.4

3.5 Reducing the Computational Weight of SP Approximations

This section presents schemes implementing the SP derivative approximations. The algorithm is assumed to be composed of the SP continuity equation (2.112), the SP equation of motion (2.140), the SP induction equation (2.118), and the specific internal energy equation (2.148). The general context for these schemes are given by Figures 3.1, 3.3 and 3.4, where the specific node under discussion is labelled

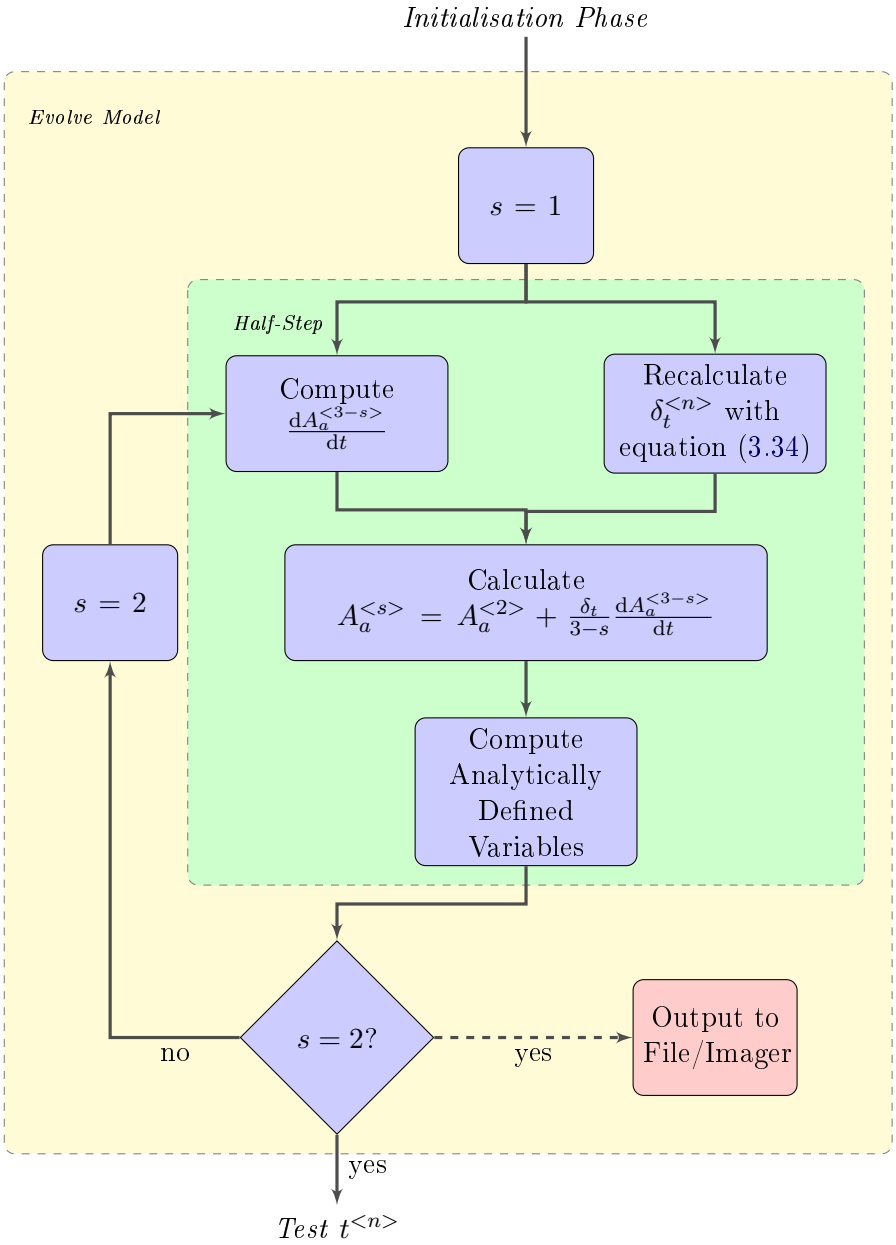

Figure 3.4: A generalised code structure diagram showing the basics that define how the model is evolved from one time step to the next using the modified Euler method and factoring in an automatically recalculated time step, δ_t.

Compute $\frac{\mathrm{d}A^{<3-s>}}{\mathrm{d}t}$ and is held recursively within the nodes Simulation Phase $\rightarrow$ Evolve Model $\rightarrow$ Half-Step.

The simplest scheme is to sweep over the pseudoparticles, $a = 1..N_p$, and compute the contribution of each pseudoparticle, $b = 1..N_p$, to each equation, resulting in a scheme of $O(N_p^2)$. However, this fails to take into account a number of the numerical factors defining the equations. For instance, if $|\mathbf{r}_{ab}| \geq Ch$ then there is no contribution from b to a. In addition the symmetry in the system ensures that there is no contribution of a to b. Extending these principles over the entire calculation, and the scheme presented in Figure 3.5 can be constructed.

Essentially, of the original $N_p \times N_p$ matrix of interactions the simplest scheme encounters, only the upper triangular components are investigated. That is the nested loops are governed by $a = 1..(N_p - 1)$; $b = (a + 1)..N_p$. Within that, only the pseudoparticle pairs that satisfy $|\mathbf{r}_{ab}| < Ch$ are allowed to interact. Then, for each interacting pair, the b^{th} contribution to the a^{th} derivative is computed, ie.

$$\left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\text{new}} = \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\text{old}} + \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_b \tag{3.37}$$

where $A = \{\rho, \mathbf{v}, \mathbf{B}, u\}$, *and* the a^{th} contribution to the b^{th} derivative is computed by

$$\left[\frac{\mathrm{d}A_b^{<3-s>}}{\mathrm{d}t}\right]_{\text{new}} = \left[\frac{\mathrm{d}A_b^{<3-s>}}{\mathrm{d}t}\right]_{\text{old}} + \left[\frac{\mathrm{d}A_b^{<3-s>}}{\mathrm{d}t}\right]_a \tag{3.38}$$

simultaneously.

There are, however, further reductions possible from the SP equations. Consider that the rate of change of specific internal energy (assuming constant entropy) is a function of the rate of change of density,

$$\frac{\mathrm{d}u_a}{\mathrm{d}t} = -\frac{P_a}{\rho_a^2}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} \tag{3.39}$$

Hence, rather than computing both, only the rate of change of density need by factored into the nested scheme such that $A = \{\rho, \mathbf{v}, \mathbf{B}\}$.

Also consider the sets that govern the particle interactions, ξ_a. These could be directly interpreted as neighbour lists, and are relatively easily implemented within the algorithm. However, in order that the lists not need to be updated every cycle of the model, the sets would have to be expanded to incorporate the pseudoparticles from outside the subdomains ω_a – the lists would become quite large, and drain resources. Also, for simple implementations, measures determining that the lists require updating must be *overly sensitive* in order to avoid the risk of omitting neighbours as the simulation evolves, inducing extra computation. That is not to

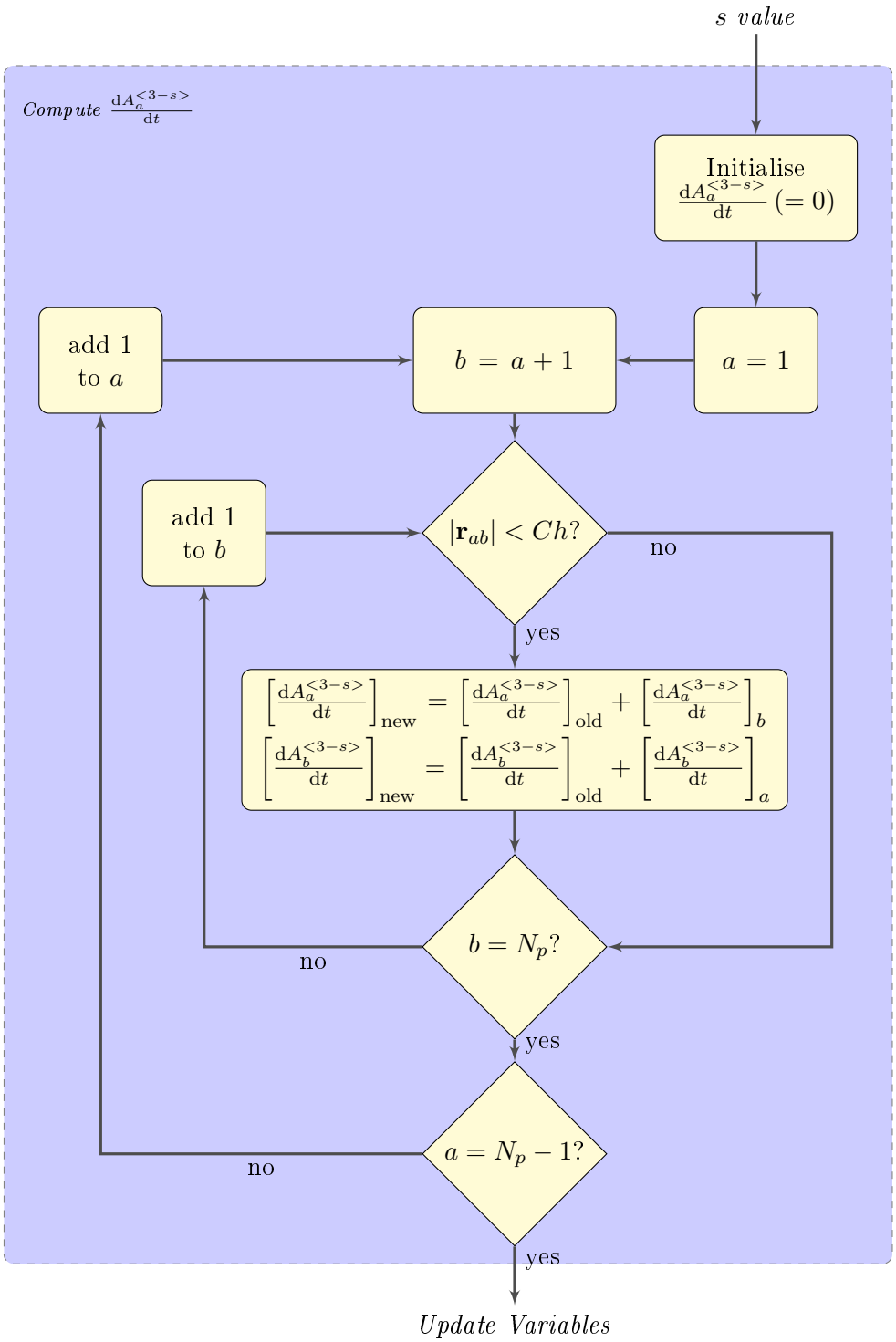

Figure 3.5: A generalised code structure diagram showing the basic structure of the SP derivative calculations.

discount explicit neighbour lists. For a number of implementations, particularly parallel codes, neighbour lists are a natural extension of the existing framework and these concerns, therefore, do not apply (Hernquist and Katz, 1989; Price, 2012; Springel, 2005). However, given the relative cost of maintaining the lists in a serial framework, herein a simpler *Bucket Sort* method is employed.

3.5.1 Bucket Sort

The bucket sort approach has a number of advantages. The algorithm follows these basic steps

1. An imaginary mesh is superimposed over the domain Ω, dividing it into a number of cells (buckets), N_c. Herein the mesh is cubic with equal sized cells and equal widths, lengths and heights.

2. The pseudoparticles are assigned to particular cells by creating a new pseudoparticle property, w_a, that records the unique identifier of the cell.

3. Pseudoparticle interactions are then limited to pairs that exist in neighbouring cells.

4. As the positions evolve, w_a is periodically reassessed as required.

Though the mesh only exists conceptually, three properties are explicitly required; the position of the cell centres $\mathbf{r}_c$; a constant, δ_c, equal to half the width of a cell; and cell neighbour lists, p_c^q (the q^{th} neighbour of the c^{th}*cell*). All of these can be defined in the initialisation phase of the program (see Figure 3.7). The neighbour lists do not suffer from the same issues as the sets ξ_a as the cells are motionless (relative to the Ω and each other). Thus p is fixed for the length of a simulation.

In order to define p, first it is necessary to define the dimensions of the cells. For the implementation, δ_c should be limited such that

$$h < \delta_c < \frac{1}{3} \min_{i=1..\nu} \left(r_{\text{U}}^i \right) \tag{3.40}$$

assuming that Ω is centred on zero, where r_{U}^i is the position of the upper boundary along the i^{th} axis. The upper limit exists to ensure that the automated cell generation routines are always applicable, whereas the lower limit constrains the number of neighbouring cells, N_q, to a constant,

$$N_q = 3^\nu - 1 \tag{3.41}$$

$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$
$\cdots$	$p_c^1 = c - N_j - 1$	$p_c^2 = c - N_j$	$p_c^3 = c - N_j + 1$	$\cdots$
$\cdots$	$p_c^4 = c - 1$	c^{th} cell	$p_c^5 = c + 1$	$\cdots$
$\cdots$	$p_c^6 = c + N_j - 1$	$p_c^7 = c + N_j$	$p_c^8 = c + N_j + 1$	$\cdots$
$\iddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Figure 3.6: A graphical representation of the definition of p_c for the c^{th} cell of the bucket sort of a 2-Dimensional domain. N_j is the number of cells in the j^{th} dimension of the domain.

These are the N_q cells that are adjacent to either a face, edge or corner of some specific cell, such that for the c^{th} cell in a 2-Dimensional domain the array p_c is defined in the manner described by Figure 3.6.

Once the cell properties are defined, it is necessary to sort the pseudoparticles. Having assigned cell identifying indices to w_a, a vector ϕ_a is then created that holds each index identifying specific pseudoparticles, i.e.

$$\phi_a = a \tag{3.42}$$

Subsequently, ϕ is sorted relative to w such that the indices of all the pseudoparticles in cell 1 are follow by the indices in cell 2, then cell 3, and so on up to the indices of all the pseudoparticles in cell N_q. The indices within each cell subsection are also

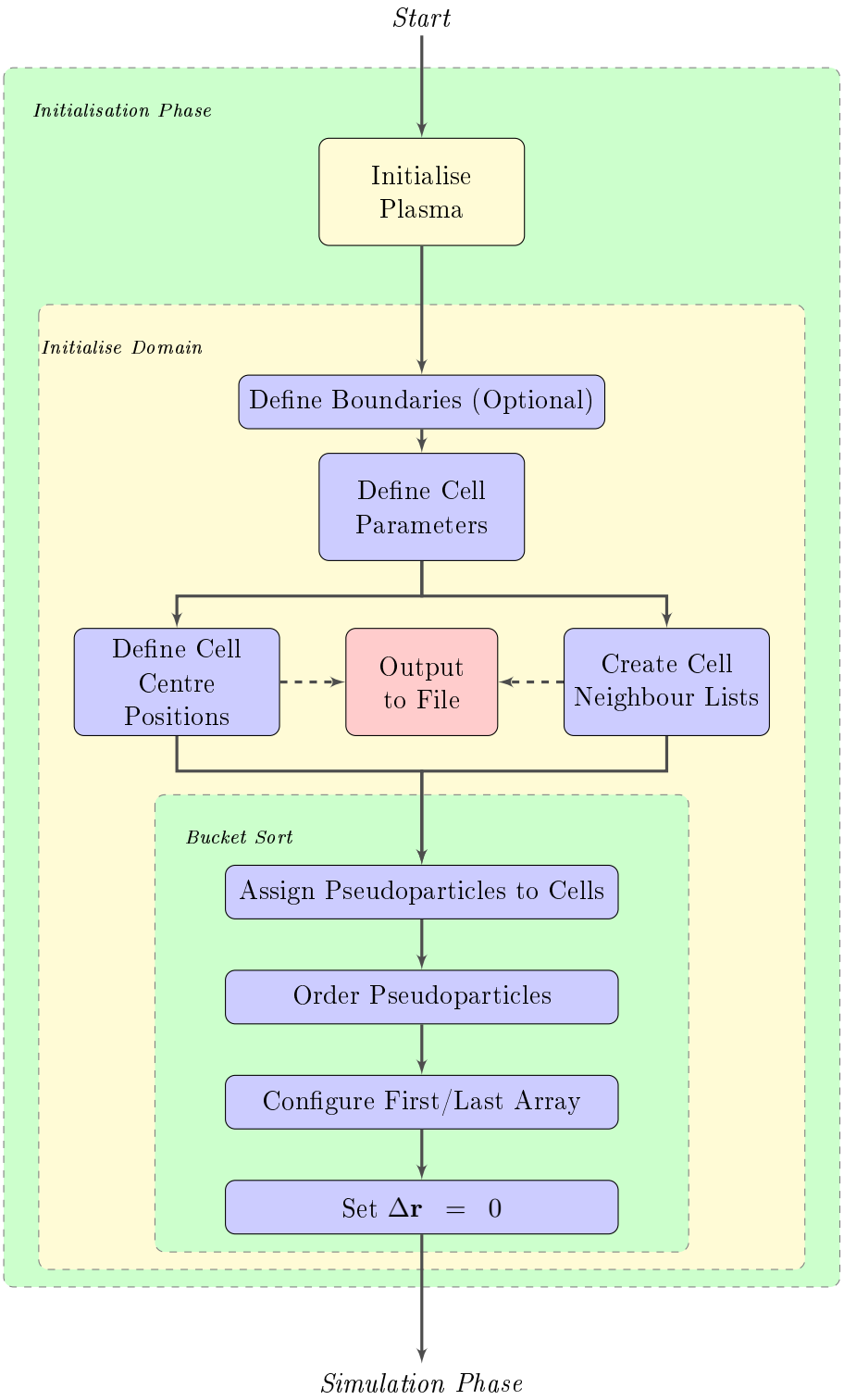

Figure 3.7: A generalised code structure diagram showing the implementation of the Bucket Sort algorithm.

sorted into ascending order (smallest to largest). That is

$$\phi = \begin{bmatrix} \{a|w_a = 1\} \\ \{a|w_a = 2\} \\ \{a|w_a = 3\} \\ \vdots \\ \{a|w_a = N_q - 1\} \\ \{a|w_a = N_q\} \end{bmatrix} \tag{3.43}$$

where $1 \leq a \leq N_p$, and $\{a|w_a = c\}$ is the set of indices of the pseudoparticles assigned to the c^{th} cell, and the sets are sorted such that

$$a_{i+1} > a_i \tag{3.44}$$

is always true, assuming the elements are ordered such that $\{\cdots, a_{i-1}, a_i, a_{i+1}, \cdots\}$. Note that this indicates that the indices assigned to the c^{th} cell are ordered $[\cdots \quad a_{i-1} \quad a_i \quad a_{i+1} \quad \cdots]^T$ within ϕ. Now, vectors f_c and l_c can also be defined as

$$f_c = \begin{cases} 1 + \sum_{k=1}^{c-1} |\{a|w_a = k\}| & \text{if} \quad |\{a|w_a = c\}| \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.45}$$

$$l_c = \begin{cases} \sum_{k=1}^{c} |\{a|w_a = k\}| = f_c + |\{a|w_a = c\}| - 1 & \text{if} \quad |\{a|w_a = c\}| \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.46}$$

where $|\{a|w_a = k\}|$ is the degree of (the number of elements within) the k^{th} set. Hence, f and l stores the positions within ϕ of the lowest and highest pseudoparticle index assigned to each cell, respectively.

As a simple example consider the distribution of pseudoparticles and a 2×2 grid of cells described by Figure 3.8. In this case, the arrays are given by

$$\begin{aligned} \phi &= \begin{bmatrix} 3 & 4 & 10 & 11 & 1 & 5 & 6 & 7 & 12 & 13 & 2 & 8 & 9 & 14 & 15 \end{bmatrix}^{\text{T}} \\ f &= \begin{bmatrix} 1 & 0 & 5 & 11 \end{bmatrix}^{\text{T}} \qquad l = \begin{bmatrix} 4 & 0 & 10 & 15 \end{bmatrix}^{\text{T}} \end{aligned} \tag{3.47}$$

Note that $f_2 = l_2 = 0$ indicating that there are no pseudoparticles in cell 2.

The process of assigning values to w_a, defining and sorting ϕ_a and eventually creating f_c and l_c is collectively referred to as the Bucket Sort in the Figures 3.7 and 3.9.

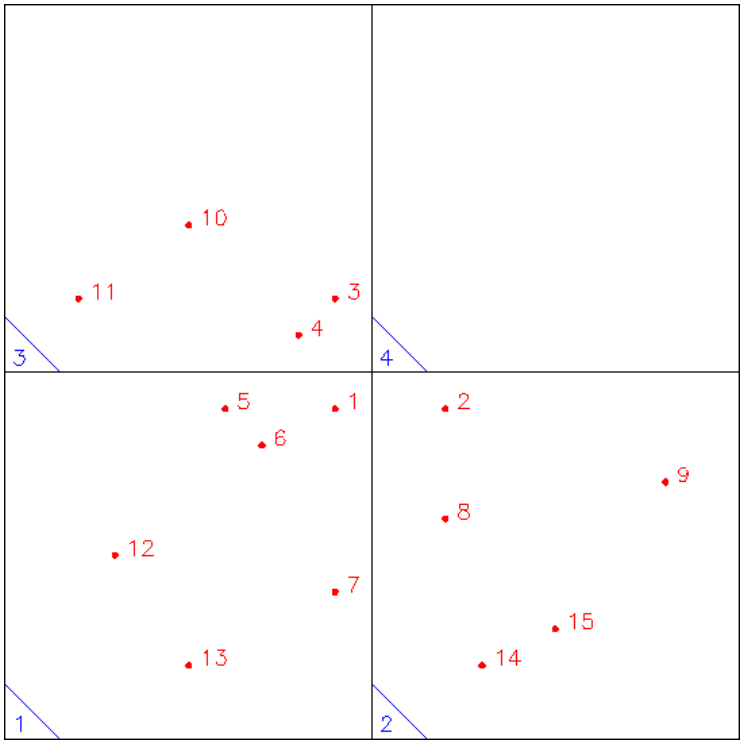

Figure 3.8: A simple example distribution of pseudoparticles with a 2×2 grid (shown in black). The pseudoparticle positions and indices are shown in red and the grid cell indices are shown in blue.

Finally, in order to automatically trigger a reconfiguration of w (and therefore ϕ, f and l), define a pseudoparticle variable $\Delta\mathbf{r}_a$ that records the total displacement (in each dimension) since the last reconfiguration of the a^{th} pseudoparticle. This concept is illustrated in Figure 3.9.

Given these new variables, the sweep over the upper triangular portion of the interaction matrix can be rewritten. Conceptually, while the pseudoparticle pair interaction order remains the same, the sequence appears to jump over many of the potential interactions as the pairs are implicitly too far apart to interact. Specifically

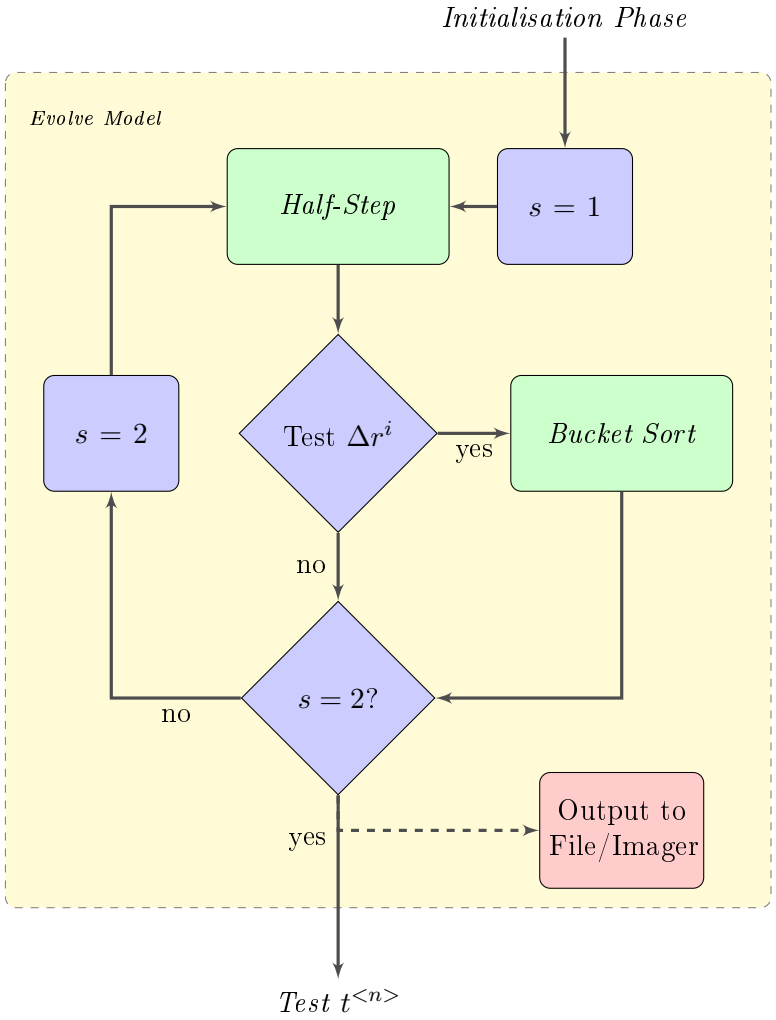

Figure 3.9: A generalised code structure diagram showing the periodic reconfiguration of the Bucket Sort algorithm. The *Test* Δr^i component checks to see that none of the pseudoparticles have travelled further than half the width of the cells in the simulated time elapsed since the previous reconfiguration, ie. $\Delta r^i + 2h > \delta_c$.

the nested structure is shown in Figure 3.10, and are governed by

$$
\begin{array}{l}
\texttt{for } a = 1 \rightarrow (N_p - 1);\ \texttt{do} \\
\quad \texttt{for } b = l_{w_a} \rightarrow f_{w_a};\ -1;\ \texttt{do} \\
\quad\quad \texttt{if } \phi_b = a;\ \texttt{exit} \\
\quad\quad \texttt{if } |\mathbf{r}_{a\phi_b}| \geq Ch;\ \texttt{cycle} \\
\quad\quad \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\text{new}} = \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\text{old}} + \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\phi_b} \\
\quad\quad \left[\frac{\mathrm{d}A_{\phi_b}^{<3-s>}}{\mathrm{d}t}\right]_{\text{new}} = \left[\frac{\mathrm{d}A_{\phi_b}^{<3-s>}}{\mathrm{d}t}\right]_{\text{old}} + \left[\frac{\mathrm{d}A_{\phi_b}^{<3-s>}}{\mathrm{d}t}\right]_{a} \\
\quad \overline{\texttt{for } q = 1 \rightarrow (3^\nu - 1);\ \texttt{do}} \\
\quad\quad \texttt{for } b = l_{p^q_{w_a}} \rightarrow f_{p^q_{w_a}},\ -1;\ \texttt{do} \\
\quad\quad\quad \texttt{if } \phi_b < a;\ \texttt{exit} \\
\quad\quad\quad \texttt{if } |\mathbf{r}_{a\phi_b}| \geq Ch;\ \texttt{cycle} \\
\quad\quad\quad \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\text{new}} = \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\text{old}} + \left[\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}\right]_{\phi_b} \\
\quad\quad\quad \left[\frac{\mathrm{d}A_{\phi_b}^{<3-s>}}{\mathrm{d}t}\right]_{\text{new}} = \left[\frac{\mathrm{d}A_{\phi_b}^{<3-s>}}{\mathrm{d}t}\right]_{\text{old}} + \left[\frac{\mathrm{d}A_{\phi_b}^{<3-s>}}{\mathrm{d}t}\right]_{a}
\end{array}
$$

3.6 Numerical Modification of the SP Magnetohydrodynamics Algorithm

The following sections present the schemes by which the ideal SPMHD algorithms may be extended to correct errors and better capture non-linear behaviour.

3.6.1 Complications Under Negative Stress

As presented in Section 2.5.3, the SP equation of motion equates to an approximation of

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \frac{1}{\rho_a}\left\{-\nabla P + \frac{1}{\mu_0}\left[(\nabla \times \mathbf{B}_a) \times \mathbf{B}_a + \mathbf{B}_a\left(\nabla \cdot \mathbf{B}_a\right)\right]\right\} \tag{3.48}$$

rather than, as it should, equation (2.87)

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \frac{1}{\rho_a}\left[-\nabla P + \frac{1}{\mu_0}\left(\nabla \times \mathbf{B}_a\right) \times \mathbf{B}_a\right] \tag{3.49}$$

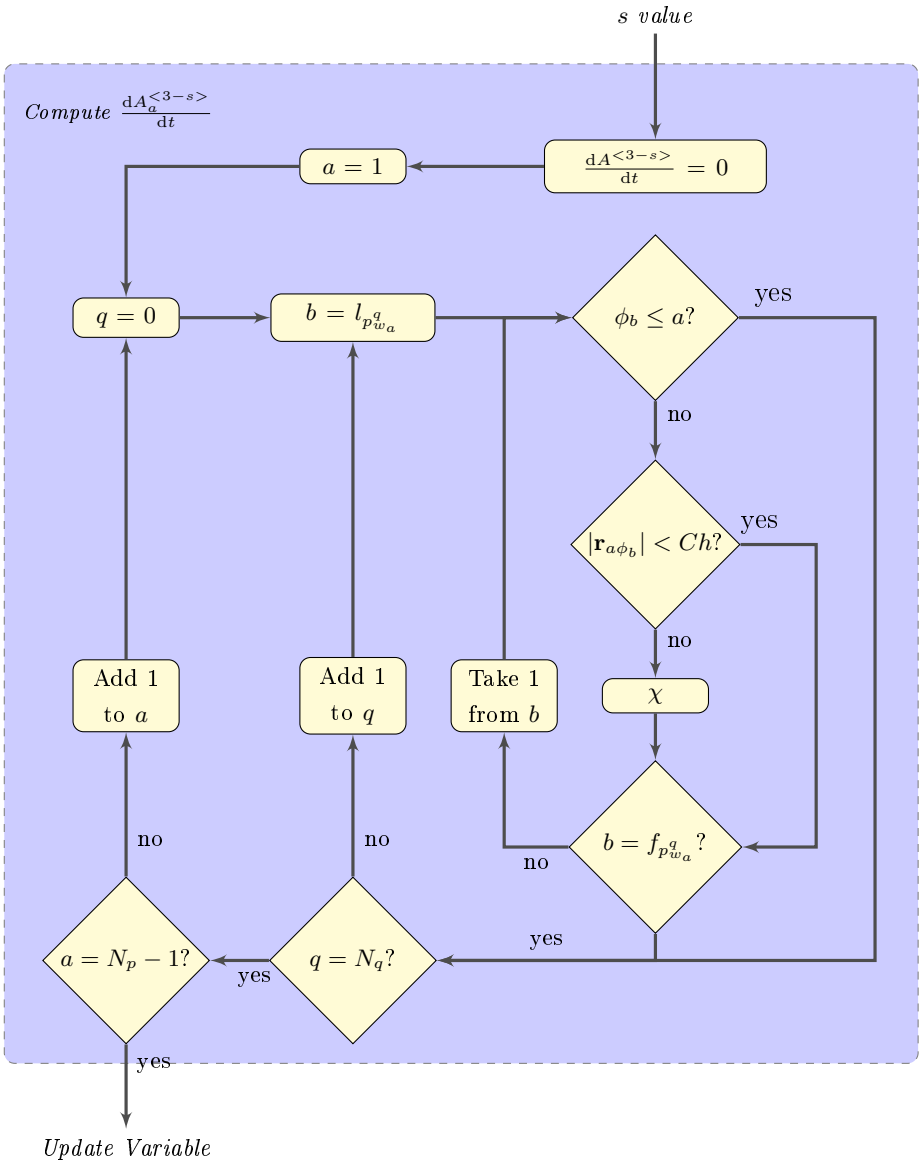

Figure 3.10: A generalised code structure diagram showing the rewritten *Compute* $\frac{dA_a^{<3-s>}}{dt}$ component, optimised to use the Bucket Sort algorithm. The χ node refers to computing the equations $[X_a]_{\text{new}} = [X_a]_{\text{old}} + [X_a]_{\phi_b}$, and $[X_{\phi_b}]_{\text{new}} = [X_{\phi_b}]_{\text{old}} + [X_{\phi_b}]_a$, where $X_y = \frac{dA_y^{<3-s>}}{dt}$. The functions are defined in Section 3.5.

At the continuum limit $[\nabla \cdot \mathbf{B}_a]_{\text{SP}} = 0$, however for finite, discrete (numerical) systems (ie. not exclusively SP methods) $\nabla \cdot \mathbf{B} \neq 0$. In the model this is interpreted as the widely discussed *tensile instability* (Hernquist and Katz, 1989; Vignjevic, 2004; Monaghan, 2005; Price, 2012). Essentially, the system becomes unstable where the stress becomes negative as the pseudoparticles begin to strongly attract one another. Given the inclusion of magnetic field effects, this instability generally manifests itself when the perpendicular component of magnetic stress swamps the combined fluid and magnetic pressure, ie. in 1D $P_{\text{magnetic}} > P$ (Price, 2012). See Børve et al. (2004) for a detailed, multidimensional stability analysis. It was noted by Vignjevic (2004) that, though labelled a *tensile* instability, the cause of the instability in fact occurs in compression as well. However, with accurately specified smoothing length and positive stress the compression instability never manifests in simulations.

In this context, consider the Solar Corona and specifically the plasma parameter β, which is the ratio of the fluid and magnetic pressures, i.e. $\beta = P/P_{\text{magnetic}} = \frac{2\mu_0 P}{B^2}$. In the solar corona, the magnetic pressure, $\frac{B^2}{2\mu_0}$ where $10^{-3}\text{T} \geq |\mathbf{B}| \geq 10^{-2}\text{T}$, massively outweighs the fluid pressure, $P = (\gamma - 1)\rho u = (1 - C_v)C_p \rho T$ where $10^{-13}\text{kg.m}^{-3} \geq \rho \geq 10^{-11}\text{kg.m}^{-3}$ and $T \approx 10^6$. More specifically, the plasma β ranges from approximately 10^{-3} to 10^{-2} for coronal holes and active regions, respectively. Not only is this within the region of the instability, but drastically so. Hence, SP simulations of the corona must be highly susceptible to the aforementioned instability, and some measure of correction is required.

A solution removing the instability must physically conform to $\nabla \cdot \mathbf{B} = 0$ and numerically ensure that the *approximation* of $\nabla \cdot \mathbf{B}$ within the SP equation of motion equals zero (Tóth, 2000). (Price, 2012), interpreting the conclusions drawn by (Tóth, 2000, 2002), stated *"In SPMHD* [such a solution] *is equivalent to requiring both exact derivatives and exact conservation which...does not appear to be possible"*. As such, an approximate numerical solution must be found.

In the context of the limitations of the model, there are four general possible approaches to creating a solution.

i) Create a sphere of exclusion about each pseudoparticle that no other can breach, thus stopping the pseudoparticles from clumping together. This counters only the most obvious detrimental effect of the instability and does not attempt to correct for, or avoid, the problem.

ii) Artificially introduce additional stress to the system so that the total stress at any point is *always positive*, thereby avoiding but not eliminating the instability.

iii) Ere on the physical, and introduce a more accurate formulation that better approximates $\nabla \cdot \mathbf{B} = 0$.

iv) Ere on the numerical, that is the Lagrangian formulation basis, and correct for the inaccurate $\nabla \cdot \mathbf{B}$ as it appears in the SP equation of motion.

Note that iii) and iv) are both the logical outcomes of the line of reasoning that concludes; holding to at least one of the critera of stability is better than none at all.

Approach i) was proposed by Monaghan (2000) was quickly shown to be problematic in high compression simulations and leads to errors in the calculation of sound speed (and therefore time-stepping – see Section 3.4.1). In addition, determining the extent of the exclusion sphere presents a particular problem as a non-physical parameter, particularly for variable smoothing length algorithms.

The remaining propositions are more applicable, but each have advantages and disadvantages. Taking each in turn;

ii) Proposed by Phillips and Monaghan (1985), after identifying the instability itself, this is the simplest fix. Rather than parametrising the additional stress, which would be problematic, the algorithm requires an additional loop over the pseudoparticles in order to determine the maximum stress, $S^{ij}_{\max}$. The resulting stress as implemented within the algorithm, $\left(S^{ij} - S^{ij}_{\max}\right)$ is therefore positive definite but the minimum stress is restricted to zero. Thus stressing the system as little as possible. The resulting modified SP equation of motion is therefore,

$$\frac{\mathrm{d}v^i_a}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} \left(S^{ij}_a - S^{ij}_{\max} \right) + \frac{\Psi_a}{\Psi_b} \left(S^{ij}_b - S^{ij}_{\max} \right) \right] \nabla^j_a W_{ab} \tag{3.50}$$

This is equivalent to the modification

$$\left[\frac{\mathrm{d}v^i_a}{\mathrm{d}t} \right]_{\text{new}} = \left[\frac{\mathrm{d}v^i_a}{\mathrm{d}t} \right]_{\text{old}} - \frac{2S^{ij}_{\max}}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left(\frac{\Psi_b}{\Psi_a} + \frac{\Psi_a}{\Psi_b} \right) \nabla^j_a W_{ab} \tag{3.51}$$

Consider the advantages and disadvantages of this modification;

Advantages

- Simple implementation
- Keeps conservation of momentum
- Removes instability from simulation

- Ideal for external stresses, particularly an imposed magnetic field, that can stabilise the simulation

Disadvantages

- Drops conservation of energy
- Relatively high computational cost of finding $S^{ij}_{\max}$, which gets worse for multidimensional simulations
- Assuming $S^{ij}_{\max}$ is large, induces non-physical effects

iii) The isotropic components of the SP equation of motion, as derived from the Euler-Lagrange equation (2.131), is retained as

$$\left[\frac{\mathrm{d}v^i_a}{\mathrm{d}t}\right]_{\substack{\mathrm{SP}\\ \mathrm{iso.}\\ \mathrm{consv}}} = \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\left(\frac{\Psi_b}{\Psi_a}S^{ii}_a + \frac{\Psi_a}{\Psi_b}S^{ii}_b\right)\nabla^i_a W_{ab} \tag{3.52}$$

preserving with it the conservative properties isolated to the isotropic components. The anisotropic components are replaced by the anisotropic components as determined by direct application of the SP identities (Section 2.2.9), i.e.

$$\left[\frac{\mathrm{d}v^i_a}{\mathrm{d}t}\right]^j_{\substack{\mathrm{SP}\\ \mathrm{aniso.}\\ \mathrm{linear}}} = \frac{1}{\mu_0\Psi_a\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b\Psi_b}{\rho_b}\left(B^i_b B^j_b - B^i_a B^j_a\right)\nabla^j_a W_{ab} \tag{3.53}$$

Thus in combination,

$$\frac{\mathrm{d}v^i_a}{\mathrm{d}t} = \left[\frac{\mathrm{d}v^i_a}{\mathrm{d}t}\right]^j_{\substack{\mathrm{SP}\\ \mathrm{iso.}\\ \mathrm{consv}}} + \left[\frac{\mathrm{d}v^i_a}{\mathrm{d}t}\right]^j_{\substack{\mathrm{SP}\\ \mathrm{aniso.}\\ \mathrm{linear}}} \tag{3.54}$$

such that, by rearrangement,

$$\frac{\mathrm{d}v^i_a}{\mathrm{d}t} = \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\left[\frac{\Psi_b}{\Psi_a}\left(S^{ij}_a\delta^{ij} + \frac{B^i_b B^j_b - B^i_a B^j_a}{\mu_0}\right) + \frac{\Psi_a}{\Psi_b}S^{ij}_b\delta^{ij}\right]\nabla^j_a W_{ab} \tag{3.55}$$

Note the advantages and and disadvantages of this compromise approach (Morris, 1996);

Advantages

- Captures, with relatively small errors, MHD shocks
- Conservation within isotropic component, enabling *remeshing* of pseudoparticle configurations (until errors accumulate, and symmetry is lost)

Disadvantages

- No conservation of energy or momentum
- Non-trivial conversion moving from constant to variable smoothing length
- Only vanishes if $B_a^i B_a^j = B_b^i B_b^j$, and cannot be *turned off*
- For long running simulations can build up significant, cumulative error

iv) Proposed by Børve et al. (2001) and deeply analysed by Børve et al. (2004, 2006), is the explicit subtraction of the solenoidal constraint term as it appears in the SP equation of motion. That is

$$\frac{\tilde{\mathbf{B}}_a}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left(\frac{\Psi_b}{\Psi_a} \mathbf{B}_a + \frac{\Psi_a}{\Psi_b} \mathbf{B}_b \right) \cdot \nabla_a W_{ab} \tag{3.56}$$

Note that $\tilde{\mathbf{B}}_a \neq \mathbf{B}_a$. In fact, Børve et al. (2006) showed that, though $\tilde{\mathbf{B}}_a$ must be parallel to $\mathbf{B}_a$, the magnitude can be scaled such that $|\tilde{\mathbf{B}}_a| < |\mathbf{B}_a|$. In order to analyse the stable regime, they created a measure $\epsilon = \frac{\tilde{\mathbf{B}}_a \cdot \mathbf{B}_a}{B_a^2}$. $\epsilon_{\min}$ is the smallest possible value at which the resulting algorithm is stable. Analysing $\epsilon_{\min}$ as a function of plasma β, it was established that

$$\lim_{\beta \to 0} \epsilon_{\min} = \frac{1}{2} \tag{3.57}$$

and hence $\tilde{\mathbf{B}}$ is usually defined such that $\tilde{\mathbf{B}}_a = \frac{\mathbf{B}_a}{2}$. The resulting modification takes the form,

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} \left(S_a^{ij} - \tilde{B}_a^i B_a^j \right) + \frac{\Psi_a}{\Psi_b} \left(S_b^{ij} - \tilde{B}_a^i B_b^j \right) \right] \nabla_a^j W_{ab} \tag{3.58}$$

where the advantages and disadvantages are;

Advantages

- Can be switched on/off as a function of the instability existence
- Correction term identical to the component of the SP equation of motion that is in error

Disadvantages

- While *on* no conservation of energy or momentum
- Can be reduced in scale (via the $\epsilon_{\min}$ argument) but sill ensure stability

Given the constraints of the Solar Corona, a correction that can be switched off is not a requirement. However, the relative strength of the magnetic field dictates that a stable solution is most important. Therefore, the final proposition is implemented.

3.6.2 Capturing MHD Shocks

Simulating the evolution of discontinuities presents something of a challenge for numerical models. Quoting Price (2008) *"The treatment of flow discontinuities...has been the subject of a vast body of research over the last 50 years"*. In the context of SPMHD, two factors effect the algorithms ability to model shocks,

- The previously discussed (Sections 2.2.6 and 3.3.2) assumption the the pseudoparticles do not approach the surface of the domain, $S(\Omega)$
- The fundamental assumption that the solutions the SP derivative approximations are looking to find are continuous.

Lengthy discussions can be found in a number of papers (Monaghan, 1997; Price and Monaghan, 2004c,a; Price, 2008) and here the issue is presented in brief.

In order to accurately simulate shocks in SP algorithms, a number of dissipation terms must be included. These spread the effects of the discontinuity (also referred to as diffusion) so that the SP approximations see it as a continuous slope, admittedly very steep, spread out over a few Δ_p. Most authors, at this point, quote the widely implemented (but entirely numerical) *artificial viscosity* term Π_{ab} (Monaghan, 1992), and then express the advances since (Bonet and Rodríguez-Paz, 2005; Price, 2008), leading to the general formulation presented by Monaghan (1997). This general formulation, for some property λ, is given by

$$\left[\frac{\mathrm{d}\lambda_a}{\mathrm{d}t}\right]_{\mathrm{diss}} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \alpha_{\lambda,a} v_{\mathrm{sig},\lambda,ab}}{\overline{\rho_{ab}}} \lambda_{ab} \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|} \cdot \nabla_a W_{ab} \tag{3.59}$$

where $\alpha_{\lambda,a}$ is a variable defining the amount of diffusion (discussed later), and $v_{\mathrm{sig},\lambda,ab}$ is the signal velocity *as required by* λ (details to follow). For context, the revised derivative approximations take the form,

$$\left[\frac{\mathrm{d}\lambda_a}{\mathrm{d}t}\right]_{\mathrm{new}} = \left[\frac{\mathrm{d}\lambda_a}{\mathrm{d}t}\right]_{\mathrm{old}} + \left[\frac{\mathrm{d}\lambda_a}{\mathrm{d}t}\right]_{\mathrm{diss}} \tag{3.60}$$

Comparing equation (3.59) with the higher order SP derivatives (Section 2.4), and assuming that the dissipation term represents

$$\left[\frac{\mathrm{d}\lambda_a}{\mathrm{d}t}\right]_{\mathrm{diss}} = \eta_\lambda \left[\nabla^2 \lambda_a\right]_{\mathrm{SP}} \tag{3.61}$$

the parameter η_λ is shown to be particular to the individual pseudoparticle interactions, ie. $\eta_\lambda = \eta_{\lambda,ab}$, and that

$$\eta_{\lambda,ab} \propto \alpha_{\lambda,a} v_{\mathrm{sig},\lambda,ab} |\mathbf{r}_{ab}| \tag{3.62}$$

Given equation (3.59), it is possible to define the SP viscosity and SP total energy dissipation terms as

$$\left[\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right]_{\mathrm{diss}} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\overline{\rho_{ab}}} \alpha_{\mathbf{v},a} v_{\mathrm{sig},\mathbf{v},ab} \left(\mathbf{v}_{ab} \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}\right) \nabla_a W_{ab} \tag{3.63}$$

and

$$\left[\frac{\mathrm{d}e_a}{\mathrm{d}t}\right]_{\mathrm{diss}} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\overline{\rho_{ab}}} (\tilde{e}_a - \tilde{e}_b) \frac{\partial W_{ab}}{\partial |\mathbf{r}_{ab}|} \tag{3.64}$$

respectively, where

$$\tilde{e}_a = \frac{1}{2} \alpha_{\mathbf{v},a} v_{\mathrm{sig},\mathbf{v},ab} \left(\mathbf{v}_a \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}\right)^2 + \alpha_{u,a} v_{\mathrm{sig},u,ab} u_a + \alpha_{\mathbf{B},a} v_{\mathrm{sig},\mathbf{B},ab} \frac{B_a^2}{2\mu_0 \overline{\rho_{ab}}} \tag{3.65}$$

Note also that

$$v_{\mathrm{sig},\mathbf{v},ab} = \begin{cases} \tilde{v}_a + \tilde{v}_b - \beta\chi & \text{if} \quad \chi \leq 0 \\ 0 & \text{if} \quad \chi > 0 \end{cases} \tag{3.66}$$

where $\chi = \mathbf{v}_{ab} \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}$ and

$$\tilde{v}_a = \frac{1}{2}\sqrt{c_a^2 + \frac{B_a^2}{\mu_0 \rho_a} + \frac{2c_a}{\sqrt{\mu_0 \rho_a}} \mathbf{B}_a \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}} + \frac{1}{2}\sqrt{c_a^2 + \frac{B_a^2}{\mu_0 \rho_a} - \frac{2c_a}{\sqrt{\mu_0 \rho_a}} \mathbf{B}_a \cdot \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|}} \tag{3.67}$$

Also, that the thermal signal velocity does not have to equal the true signal velocity, $v_{\mathrm{sig},u,ab} \neq v_{\mathrm{sig},\mathbf{v},ab}$. Price (2008, 2012), for instance, recommend setting $v_{\mathrm{sig},u,ab} = \sqrt{\frac{P_{ab}}{\overline{\rho_{ab}}}}$.

The second law of thermodynamics constrains the forms of the remaining dissipation terms (Price, 2012). Each must be derived such that the contribution to the

entropy, via the total energy equation (2.155), must be positive definite. Thus the specific internal energy dissipation (thermal conduction) term is given by

$$\left[\frac{\mathrm{d}u}{\mathrm{d}t}\right]_{\mathrm{diss}} = -\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\overline{\rho_{ab}}}\left[\alpha_{\mathbf{v},a} v_{\mathrm{sig},\mathbf{v},ab}\frac{\mathbf{v}_{ab}^2}{2} + \alpha_{u,a} v_{\mathrm{sig},u,ab} u_{ab} + \alpha_{\mathbf{B},a} v_{\mathrm{sig},\mathbf{B},ab}\frac{\mathbf{B}_{ab}}{2\mu_0\overline{\rho_{ab}}}\right]\frac{\partial W_{ab}}{\partial|\mathbf{r}_{ab}|} \tag{3.68}$$

Using equations (3.64) and (3.68), it is possible to rearrange and define the dissipation (resistivity) term within the induction equation. Namely,

$$\left[\frac{\mathrm{d}\mathbf{B}_a}{\mathrm{d}t}\right]_{\mathrm{diss}} = \rho_a \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\overline{\rho_{ab}}^2}\alpha_{\mathbf{B},a} v_{\mathrm{sig},\mathbf{B},ab}\mathbf{B}_{ab}\frac{\partial W_{ab}}{\partial t} \tag{3.69}$$

where

$$v_{\mathrm{sig},\mathbf{B},ab} = \frac{1}{2}\sqrt{\left(\frac{B_a^2}{\mu_0\rho_a}\right)^2 + \left(\frac{B_b^2}{\mu_0\rho_b}\right)^2} \tag{3.70}$$

(Price and Monaghan, 2004c). Note that $v_A = B^2/\mu_0\rho$ is the Alfvén speed.

In order that these dissipation terms have sufficient weight to diffuse the shock, but effectively vanish away from the discontinuity, the variable $\alpha_{\lambda,a}$ must be evolved according to

$$\frac{\mathrm{d}\alpha_{\lambda,a}}{\mathrm{d}t} = \mathbb{S}_{\lambda,a} + (\alpha_{\lambda,a} - \sigma_1)\frac{\sigma_2 c_a}{h} \tag{3.71}$$

(Morris and Monaghan, 1997) where $\mathbb{S}_\lambda$ is some source term, σ_1 and σ_2 are parameters. Note that $\frac{h}{\sigma_2 c_a}$ is the decay time, and $\sigma_1 \approx \sigma_2 \approx 0.1$. The source terms are given by

$$\mathbb{S}_{\mathbf{v},a} = \max\left(0, -\left[\nabla\cdot\mathbf{v}_a\right]_{\mathrm{SP}}\right) \tag{3.72}$$

$$\mathbb{S}_{u,a} = \sigma_2 h\left[\nabla^2 u_a\right]_{\mathrm{SP}} \tag{3.73}$$

and

$$\mathbb{S}_{\mathbf{B},a} = (\mu_0\rho_a)^{-\frac{1}{2}}\max\left(\left|\left[\nabla\times\mathbf{B}_a\right]_{\mathrm{SP}}\right|, \left|\left[\nabla\cdot\mathbf{B}_a\right]_{\mathrm{SP}}\right|\right) \tag{3.74}$$

for the viscosity (Morris and Monaghan, 1997), conductivity and resistivity (Price and Monaghan, 2005), respectively.

3.6.3 Ensuring the Solenoidal Constraint

For a text describing a numerical approach to the MHD equations, very little has been written herein concerning the solenoidal constraint ($\nabla \cdot \mathbf{B}_a = 0$), save to simplify the iMHD equation (Section 2.5.1) and in response to the tensile instability (Section 3.6.1). This should not be taken as a sign that the constraint has been ignored. Nor that the constraint is, in any way, unimportant. Indeed, it is the only component of the MHD theory that remains unchanged (version to version) and unalterable. Under no conditions are monopoles allowed in the MHD view of the physical universe.

In SP theory, however, there are no techniques for imposing $\nabla \cdot \mathbf{B} = 0$. Instead, the algorithm must be built to either restrict $\nabla \cdot \mathbf{B}$ to some small value (*prevent*), or to allow the system to evolve without constraint then perform a subsequent sweep over the pseudoparticles *cleaning* the $\nabla \cdot \mathbf{B}$ values. A third option exists; to numerically *ignore* the effect of $\nabla \cdot \mathbf{B}$, trusting the algorithm to keep the value low, and monitor its value. However, this risks producing (at expense) corrupted results and inducing a rewrite of the algorithm to include one of the previous approaches regardless. The emphasised text here (e.g. *prevent*) refer to the labels given in Price (2012). In that same paper, numerous specific methods are described for handling the solenoidal constraint, including the procedure for monitoring the value of $\nabla \cdot \mathbf{B}$.

Referring to Powell et al. (1999) (the technique presented in Section 10.1 of Price (2012)), recall the iMHD induction equation (2.101),

$$\frac{\mathrm{d}\mathbf{B}}{\mathrm{d}t} = (\mathbf{B} \cdot \nabla)\,\mathbf{v} - \mathbf{B}\,(\nabla \cdot \mathbf{v}) \tag{3.75}$$

If follows that the divergence of equation (2.101) is

$$\frac{\partial}{\partial t}(\nabla \cdot \mathbf{B}) + \nabla \cdot (\mathbf{v} \nabla \cdot \mathbf{B}) = 0 \tag{3.76}$$

and, writing $\phi = \frac{\nabla \cdot \mathbf{B}}{\rho}$

$$\frac{\partial \rho\phi}{\partial t} + \rho\psi\nabla \cdot \mathbf{v} = \frac{\mathrm{D}\rho\phi}{\mathrm{D}t} = 0 \tag{3.77}$$

Hence ϕ is a conserved quantity. Thus if the initial, and boundary conditions, are consistent with the solenoidal constraint conservation will be exact in the same sense as the conservation of momentum and energy (i.e. as exact as the numerical temporal integration will allow). Powell et al. (1999) therefore introduced the 8-wave formulation whereby

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \frac{1}{\rho}\left[\nabla \mathbf{S} - \mathbf{B}\,(\nabla \cdot \mathbf{B})\right] \tag{3.78}$$

and

$$\frac{\mathrm{d}e}{\mathrm{d}t} = \frac{1}{\rho}\left[\mathbf{v}\nabla\mathbf{S} - \mathbf{v}\mathbf{B}\left(\nabla\cdot\mathbf{B}\right)\right] \tag{3.79}$$

are the equation of motion and the energy equation, respectively, and the induction equation is given by equation (2.101).

Note that this scheme has been implicitly implemented already, and this is the reason for addressing the solenoidal constraint at this late stage – i.e. that no new correction are required. By utilising the SP induction equation (2.118), and implementing the third of the tensile instability correction techniques, this algorithm is in effect the 8-wave formulation. The implementation herein differs in two places. Firstly, that the magnetic field variable ahead of the divergence in equation (3.78) is replaced by the scaled vector $\tilde{\mathbf{B}}$ (see Section 3.6.1), which has already been shown to be stable. Secondly, that the specific internal energy, rather than the total energy is evolved. This serves to reduce the number of modifications required, as the magnetic field does not directly effect $\frac{\mathrm{d}u_a}{\mathrm{d}t}$ as derived.

3.7 Conclusion

At this stage, the fundamentals exist to construct a functioning algorithm. The resulting model is not only optimised in the sense of programmed efficiency but, more importantly, in the sense that the equations that the algorithm is composed of are the least expensive configuration allowable that ensure the highest accuracy simulations. There are, however, outstanding issues. For example, there is no quantifiable measure of the SP error. Nor is there specialisation of the code to more efficiently simulate solar phenomena. This current SP algorithm could get the job done, but at vast and unnecessary computational cost.

In order to reduce the computational weight of the algorithm, the following Chapter (the first of the three chapters to present original research) considers extending the algorithm to include variable spatial resolution in the form of variable smoothing length.

Chapter 4

Effect and Effectiveness of Variable Smoothing Length

Contents

4.1 Introduction

Conceptually, variable soothing length, h, is one of the simplest aspects of SP theory. The previously constant h is exchanged for a set of values h_a that are allowed to evolve over simulated time, such that simulations may consist of different sized pseudoparticles. In this way, variable h enables variable resolution.

The notion of variable spatial resolution is potentially useful to the solar model, however first consider the standardised benefits of an SP algorithm built with variable smoothing length. Monaghan (2005) and Bonet and Rodríguez-Paz (2005) showed that, due to an increase in model stability under variable h, there is a corresponding increase in the accuracy of data output. In addition, from an implementation perspective, the inclusion of variable h should enable the programs to concentrate their resources on the most dynamic/energetic/complex regions of a domain – thereby reducing either/both memory requirement and run time. It has been shown by numerous authors (Monaghan, 1992; Vila, 1999; Price, 2008) that varying h stabilises (at least in part) rarefaction waves, and waves in general. This is of particular interest in a solar context as the possible phenomena under investigation induce numerous waves (Yokoyama et al., 2001; Delannée et al., 2008).

The method requires some measure of investigation for, once variable smoothing length has been chosen over constant, what is the best method for implementing the variability? There are numerous different interpretations of variable h_a (discussed Section 4.3), and many different measures of *best*. The best could be defined, for instance, as the method with the simplest rational, or the simplest implementation, or greatest consistency, or greatest accuracy, or some balance of those. In addition, there are questions as to the consistency of h_a (Vila, 1999), and discrepancies in my own data that require further investigation (Section 4.4).

In brief, the research in this chapter addresses; the question of consistency (Section 4.3.3), leading to a new formalism for the definition of the smoothing function derivative (Section 4.3.4); and an investigation into a numerical artefact, including possible correction procedures, identified in Section 4.4.

4.2 Dynamic Variation of Smoothing Length in Space and Time

The numerous SP algorithms modified to include variable smoothing length follow a similar template. Subsequent to the derivation of standard SP theory (Chapter 2), an individual smoothing length, h_a, is created for each pseudoparticle to allow variability throughout the domain, Ω. The smoothing lengths are set proportional to a function of some dynamic property of the system, enabling them to evolve over simulated time. Usually

$$h_a \propto \rho^{-\frac{1}{\nu}} \tag{4.1}$$

as with Bonet and Rodríguez-Paz (2005); Monaghan (2005); Price (2008). Occasionally the smoothing lengths will be set relative to some numerical (rather than physical) property, for instance $h_a \propto n_a^{-\frac{1}{\nu}}$, where n_a is the pseudoparticle number density (Hernquist and Katz, 1989), not to be confused with N_a, the set of interacting neighbours. This alteration introduces two new properties of the system that must be accounted for within the algorithm. Namely,

$$\frac{\mathrm{d}h_a}{\mathrm{d}t} \neq 0 \tag{4.2}$$

and

$$\frac{\mathrm{d}h_a}{\mathrm{d}r_a} \neq 0 \tag{4.3}$$

The general effects of these new properties are discussed in Sections 4.2.3 and 4.2.4. First, what justification can there be for the inclusion of variable smoothing lengths, and what is the *correct* definition of h_a?

4.2.1 Rational

Primarily, there are three rationales given in the literature for the modification of an SP algorithm to include variable smoothing length. In order of ascending individual validity, they are experimental, numerical and physical in nature. Each rational is important as they inform the particular definition of smoothing length employed within an algorithm.

4.2.1.1 Accuracy

The simplest, and least restrictive, rational for variable smoothing length results from numerical experiments. Namely that, given data from test cases produced by

models built with and without variable smoothing length, its inclusion is justified if the errors in the former are lower than in the latter. This spurious argument relies on the assumption that the physical regimes of the test cases encompass all the subsequent applications of the algorithm. This is invalid as a test case must be simple enough that the problem must be solvable analytically. An overly simple example would be if the test cases were all 1-Dimensional but the research applications required multidimensional domains. Or, more pertinently, if the test cases rely on an assumption of some large dominance of gas pressure over magnetic pressure (or vice versa), and the subsequent application included simulation of a phenomenon of relatively high gas pressure bound by high magnetic pressure, as with solar filaments.

As the least rigorous, this rational only restricts the definition of the smoothing length to forms that decrease the error in results from simulations. These are, clearly, numerous. More significantly, it places no constraints on the form, or the consistency of the form, of the rate of change of the smoothing length over time or space. Note, however, that this accuracy argument usually occurs in tandem with one of the other rationales, usually to justify a specific definition of h_a.

4.2.1.2 Divergent Material

Consider an argument from a numerical perspective. Gaseous material can diverge greatly and plasmas even more so, given the **B**-field dependence. Within standard SP theory, the smoothing length, h, is constant and equal throughout the simulation domain, Ω. It is possible under divergent conditions, therefore, that the pseudoparticles will separate to the point that the distance between them, $|\mathbf{r}_{ab}|$, is greater than the compact support of the smoothing function (which is related to the smoothing length by $\kappa = Ch$, see Section 2.2.3). That is, the subdomain of each pseudoparticle, ω_a and ω_b, will not intersect. Thus, the pseudoparticles cease interacting with their neighbours leading to significant errors in the interpolation of variables. Even where the divergence is relatively small, the increased separation constitutes a drop in spacial resolution. In order to avoid these problems, the smoothing length (and thus the subdomains), must be allowed to vary such that the number of interactions between pseudoparticles remains roughly constant over time. Or, to reiterate, the smoothing length must be a function of the number density of each pseudoparticle.

$$h_a^{<n>} \propto \frac{1}{n_a^{<n>}} \tag{4.4}$$

where n_a is the number density of pseudoparticle a.

The numerical argument can be extended. Consider some extended, complex phenomena that is not bound to any single axis but curves as it interacts with the

surrounding, smoothly varying, background, for instance a solar filament. If all of the pseudoparticles have the same smoothing length, high enough to capture the complex phenomena, then a great deal of unnecessary computation occurs modelling the background medium. If instead the smoothing length were increased for pseudoparticles in the 'background' portions of the domain, hence reducing the number of particles required, the volume of computation would be reduced. Thus justifying the inclusion of variable smoothing lengths.

Constant Neighbourhood There are two ways of enforcing a constant neighbourhood, or number density. Given some limiting parameter, such as some *ideal* number density, and optimising using the neighbour lists (if included as part of the implementation), redefine each smoothing length at each time-step, $h_a^{<n>}$, such that $n_a^{<n>} \approx n_a^{<n-1>}$. Alternatively, h could be evolved relative to the number of neighbours. For example, the scheme presented by Hernquist and Katz (1989) whereby the smoothing length is determined by

$$h_a^{<n>} = h_a^{<n-1>} \left(\frac{N}{|\xi_a|^{<n-1>}} \right)^{\frac{1}{\nu}} \tag{4.5}$$

where N is some parameter. Or, as the average between $h_a^{<n>}$ and $h_a^{<n-1>}$,

$$h_a^{<n>} = \frac{h_a^{<n-1>}}{2} \left[1 + \left(\frac{N}{|\xi_a|^{<n-1>}} \right)^{\frac{1}{\nu}} \right] \tag{4.6}$$

and $|\xi_a|^{<n>}$, the degree of the set ξ_a at the n^{th} iteration, is found using the predicted value of $h_a^{<n>}$. The variation in $h_a^{<n>}$ is constrained such that $|\xi_a|^{<n>}$ only differs from N by some prescribed tolerance (a few percent).

In this way, the rates of change over time or space have no consistent form.

4.2.1.3 Physical Argument

Finally, if one assumes that the pseudoparticles represent moving packets of material, it follows that since volume is defined by

$$V = \frac{m}{\rho} \tag{4.7}$$

and the mass of each pseudoparticle is held constant, as the density of each varies so too must the volume. This rational asserts that the smoothing length is a proxy for the volume of each pseudoparticle, and must therefore, be variable. This is the natural consequence of the assumption that the sphere of influence of pseudoparticle

a, the subdomain ω_a, is representative of the volume of the material packet. The extent of ω_a is defined by the compact support, κ, of the smoothing function and, given $\kappa = Ch$ (see Section 2.2.3), the smoothing length is thus restricted by

$$h \propto \rho^{-\frac{1}{\nu}} \tag{4.8}$$

where ν is the number of dimensions.

A logical extension of this rational, that assumes nothing about the masses of the particles, must conclude that

$$h \propto V^{\frac{1}{\nu}} \tag{4.9}$$

This form has several advantages over equation (4.8), as the masses can be modified during the initialisation phase to produce a set of pseudoparticles with a smaller range of smoothing lengths (and therefore volumes) which reduces some of the errors discussed later in this chapter (Section 4.4).

4.2.2 Smoothing Length Definition

Given the rigour of the physical rational for the inclusion of variable smoothing length, and the control allowed by defining h such that equation (4.9) is satisfied, the smoothing length herein is defined by

$$h_a = \beta \left(\frac{m_a}{\rho_a} \right)^{\frac{1}{\nu}} \tag{4.10}$$

where β is some scaling constant. There are two important factors of this definition that require some discussion.

4.2.2.1 Limits

It has been shown quantitatively by Vila (1999), and discussed qualitatively by Monaghan (2005) that the variation in h_a must have some upper and lower limit, i.e.

$$C_1 < \frac{h_a^{<n>}}{h_a^{<0>}} < C_2 \tag{4.11}$$

where C_1 and C_2 are some positive constants.

For most simulations, where the density variation is itself limited, equation (4.10) satisfies the limits (4.11) automatically. However, for highly dynamic models, such an assertion cannot be made. In these conditions an upper limit for the smoothing

length is enforced by redefining equation (4.10) by

$$h_a = \beta \left(\frac{m_a}{\alpha + \rho_a} \right)^{\frac{1}{\nu}} \tag{4.12}$$

where β retains its original definition, and α is some suitable constant (Monaghan, 2005). A lower limit can be enforced with a similar redefinition.

4.2.2.2 Initialisation

The smoothing length and density are functions of each other, by equation (4.10) and equation (2.72), and must therefore be defined consistently at $t^{<0>}$. Thus, some root finding algorithm and control routines must be added to the initialisation phase of the simulation.

Let $\rho_a^{(1)}$ be the density as calculated by rearranging the definition of h_a, equation (4.10), and $\rho_a^{(2)}$ be the density as calculated by the summation approximation equation (2.72), i.e.

$$\rho_a^{(1)} = m_a \left(\frac{\beta}{h_a^{(1)}} \right)^{\nu} \tag{4.13}$$

$$\rho_a^{(2)} = \sum_{b \in \xi_a} m_b W_{ab} \left(H^{(2)} \right) \tag{4.14}$$

where $H^{(2)} = H\left(h_a^{(2)}, h_b^{(2)}\right)$. Note that the smoothing length is quoted as $H^{(2)}$ rather than $h_a^{(2)}$ in equation (4.14) to generalise the routine (see Section 4.3 for further details). Similarly, let $h_a^{(1)}$ and $h_a^{(2)}$ be the smoothing lengths found by inversions of these definitions.

The objective, during the initialisation phase, must be that the pseudoparticle properties conform to

$$\rho_a^{(1)} - \rho_a^{(2)} = 0$$
$$h_a^{(1)} - h_a^{(2)} = 0$$

One of these equations – typically the former as equation (4.14) may not be rearranged to give $h_a^{(2)} = h\left(\rho_a^{(2)}\right)$, depending on the definition of H – forms the basis of the root finding algorithm. Most authors (Price and Monaghan, 2004b; Bonet and Rodríguez-Paz, 2005; Price, 2012) quote use of the Raphson-Newton method, though most root finding algorithms will suffice as there is little need for a particularly rapid approach as the routine only runs once, during the initialisation of the plasma.

4.2.3 Temporal Variation

Of the two new system properties, equations (4.2) and (4.3), the former is the easiest to include into the original SP algorithms. At its simplest, when the properties that the smoothing lengths are dependant on vary, the smoothing lengths must be recalculated. Logically, this occurs just after the properties have been evolved by whatever numerical integration technique is in use (see Section 3.4.1).

In the case of a physical property dependant h_a, the rate of change in time can be found analytically by taking the time derivative of the definition of h_a. The derivative can then be used simultaneously with the other rates of change to evolve the system properties through numerical integration. Using equation (4.1) for instance

$$\frac{\mathrm{d}h_a}{\mathrm{d}t} = \frac{\mathrm{d}h_a}{\mathrm{d}\rho_a}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} \approx \frac{\mathrm{d}h_a}{\mathrm{d}\rho_a}\left[\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right]_{\mathrm{SP}} \tag{4.15}$$

and, specifying equation (4.10), equation (4.15) can be expanded to

$$\frac{\mathrm{d}h_a}{\mathrm{d}t} = \left[-\frac{h_a}{\nu\rho_a}\right]\left[\rho_a \sum_{b\in\xi_a} \frac{m_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab}(H)\right] = \frac{h_a}{\nu}\sum_{b\in\xi_a} \frac{m_b}{\rho_b} v_{ba}^j \nabla_a^j W_{ab}(H) \tag{4.16}$$

assuming $\Psi = 1$ (see Section 2.2.6).

A numerically defined h_a must be recalculated each step using formulae similar to the procedure described in Section 4.2.1.2. Both methods introduce a limited number of additional expressions to be computed.

4.2.4 Spatial Variation

A spatially variant h effects the algorithm through the smoothing function. Specifically,

$$W = W\left(|\mathbf{r}_{ab}|, H\right) = W_{ab}(H) \tag{4.17}$$

where, rather than a specified smoothing length, H is used here to indicate that the value is subject to a specific interpretation (see the discussion in Section 4.3). In addition, $\nabla^j W = \nabla_a^j W_{ab}(H) = \frac{\mathrm{d}W_{ab}(H)}{\mathrm{d}r_a^j}$. However,

$$\frac{\mathrm{d}W_{ab}(H)}{\mathrm{d}r_a^j} \neq \frac{\partial W_{ab}(H)}{\partial r_a^j} \tag{4.18}$$

as with constant smoothing length, but rather

$$\frac{\mathrm{d}W_{ab}(H)}{\mathrm{d}r_a^j} = \frac{\partial W_{ab}(H)}{\partial r_a^j} + \frac{\mathrm{d}H}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(H)}{\partial H} \tag{4.19}$$

By substituting the general form of the smoothing function, equation (2.25), the partial derivative with respect to H can be redefined as

$$\begin{aligned}\frac{\partial W_{ab}(H)}{\partial H} &= \frac{\partial}{\partial H}\left(\frac{\sigma_\nu}{H^\nu}F(q)\right)\\ &= \frac{\sigma_\nu}{H^\nu}\frac{\partial F(q)}{\partial H} + \sigma_\nu F(q)\frac{\partial H^{-\nu}}{\partial H}\\ &= \frac{\sigma_\nu}{H^\nu}\frac{\partial q}{\partial H}\frac{\partial F(q)}{\partial q} - \frac{\nu\sigma_\nu F(q)}{H^{\nu+1}}\\ &= -\frac{\sigma_\nu q}{H^{\nu+1}}\frac{\partial F(q)}{\partial q} - \frac{\nu\sigma_\nu F(q)}{H^{\nu+1}}\\ &= -\frac{|\mathbf{r}_{ab}|}{H}\frac{\partial W_{ab}(H)}{\partial r} - \frac{\nu}{H}W_{ab}(H) \end{aligned} \tag{4.20}$$

where $q = |\mathbf{r}_{ab}|/H$. Subsequently, by substituting equation (4.20) and a direction derivative into equation (4.19),

$$\nabla_a^j W_{ab}(H) = \frac{r_{ab}^j}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}(H)}{\partial r} - \frac{\mathrm{d}H}{\mathrm{d}r_a^j}\left(\frac{|\mathbf{r}_{ab}|}{H}\frac{\partial W_{ab}(H)}{\partial r} + \frac{\nu}{H}W_{ab}(H)\right) \tag{4.21}$$

This expression collapses to give

$$\nabla_a^j W_{ab}(H) = \left(\frac{r_{ab}^j}{|\mathbf{r}_{ab}|} - \frac{|\mathbf{r}_{ab}|}{H}\frac{\mathrm{d}H}{\mathrm{d}r_a^j}\right)\frac{\partial W_{ab}(H)}{\partial r} - \frac{\nu}{H}\frac{\mathrm{d}H}{\mathrm{d}r_a^j}W_{ab}(H) \tag{4.22}$$

Without specifying $\frac{\mathrm{d}H}{\mathrm{d}r_a^j}$, which as an unknown gradient must include an SP derivative approximation within its definition, $\nabla_a^j W_{ab}(H)$ is clearly more complex than when calculated with constant smoothing length.

4.3 Multiple Interpretations

In order to appreciate the effect that the increase in complexity of W and ∇W has on the implementation of an algorithm, the variation in interpretation of the smoothing function under variable h conditions must be discussed. The general forms considered here are the four most prolific interpretations from SP literature and a new derivative formalism. These constitute the basis of specific forms that arise during the specific implementations of the method.

4.3.1 Extreme Perspectives

First, consider the following interpretive question. When computing the value of λ_a via an SP summation, equation (2.48), is the information drawn into the central pseudoparticle, a, or donated by the surrounding pseudoparticles, $b \in \xi_a$? While the smoothing length is constant, the answer is entirely arbitrary and the question only exists as a conceptual subtlety. However, when the smoothing length is allowed to vary, different answers imply different definitions of the smoothing function, W. Specifically, if the information is drawn into the a^{th} pseudoparticle the implied smoothing length, for every smoothing function within the summation is $H = h_a$. Alternatively, if the information is donated by the set of neighbouring pseudoparticles, $b \in \xi_a \neq a$, the implication is that W_{ab} is a function of the smoothing length, $H = h_b$. These two answers represent the two fundamental, opposing perspectives, defined by Hernquist and Katz (1989), as the *Gather* and *Scatter* interpretations. The smoothing functions under these interpretations are, respectively,

$$W_{ab}(H) = W_{ab}(h_a) \tag{4.23}$$

and

$$W_{ab}(H) = W_{ab}(h_b) \tag{4.24}$$

Illustrated in Figures 4.1a and 4.1b.

4.3.2 Intermediate Perspectives

The conceptual question can be generalised to instead ask where, during the calculation, does the information flow? The former question (Section 4.3.1) must be considered in order to form the logical bounds for the latter, that is the Gather and Scatter interpretations. Thus it is reasonable to consider intermediate concepts, where information is simultaneously drawn into and donated to the a^{th} pseudoparticle. It is logical to consider, specifically, that the information from each source should be weighted equally. Alternatively, though both the Gather and Scatter interpretations preserve the spherical symmetry of the smoothing function and its derivative, the reversibility condition $\nabla W = -\nabla' W$ is violated (see Section 4.3.3.2). As such, several authors have utilised numerical means. An additional complexity of an intermediate interpretation is the form of the mean. Specifically, a mean can be formed by averaging the smoothing lengths prior to the computation of the smoothing function or vice-versa, by averaging the results of the Gather and Scatter interpretations. For this discussion we label the pair of intermediate perspectives (presented by Hernquist and Katz (1989); Monaghan (2005)) as the *Merged* in-

terpretation, where $H = \overline{h_{ab}} = \frac{1}{2}(h_a + h_b)$, and *Average* interpretation, which is the mean of the Gather and Scattered smoothing functions. These interpretations respectively dictate that the smoothing functions be formulated as

$$W_{ab}(H) = W_{ab}(\overline{h_{ab}}) \tag{4.25}$$

and

$$W_{ab}(H) = \frac{1}{2}\left[W_{ab}(h_a) + W_{ab}(h_b)\right] = \overline{W_{ab}(h_a, h_b)} \tag{4.26}$$

These approaches are illustrated in Figures 4.1c and 4.1d.

The distinction between the Merged and Average interpretations is important as they are not equal, either numerically or conceptually. The latter, for instance, preserves the spherical symmetry of each particles contribution, however the resulting interactions are not spherically symmetric. The Merged interpretation, on the other hand, has no spherical symmetry at all. Unlike Gather and Scatter, both mean interpretations preserve symmetry along the line-of-sight between each pair of neighbouring particles.

4.3.3 Smoothing Function Derivative Interpretations

In the following sections of this document the derivatives of each perspective are considered. These derivatives add further complexity to the construction of an algorithm.

It is assumed here that

$$\frac{\mathrm{d}h_p}{\mathrm{d}r_q^j} = \frac{\mathrm{d}r_p^j}{\mathrm{d}r_q^j}\frac{\mathrm{d}h_p}{\mathrm{d}r_p^j} = \delta^{pq}\frac{\mathrm{d}h_p}{\mathrm{d}r_p^j} \tag{4.27}$$

which is consistant with the SP method as presented in Chapter 2, specifically the derivation of the equation of motion (by solving the Euler-Lagrange equation (2.131) or through variational principles) in Section 2.5.3.

It is also important to recall that the directional derivative form of the partial derivative $\frac{\partial W_{ab}}{\partial r_a^j}$, equation (2.17). To reiterate,

$$\frac{\partial W_{ab}}{\partial r_a^j} = \frac{\partial \mathbf{r}_{ab}}{\partial r_a^j}\frac{\partial |\mathbf{r}_{ab}|}{\partial \mathbf{r}_{ab}}\frac{\partial W_{ab}}{\partial |\mathbf{r}_{ab}|} = \frac{r_{ab}^j}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}}{\partial r} \tag{4.28}$$

where, for simplicity of notation, $r = |\mathbf{r}_{ab}|$ in the final term.

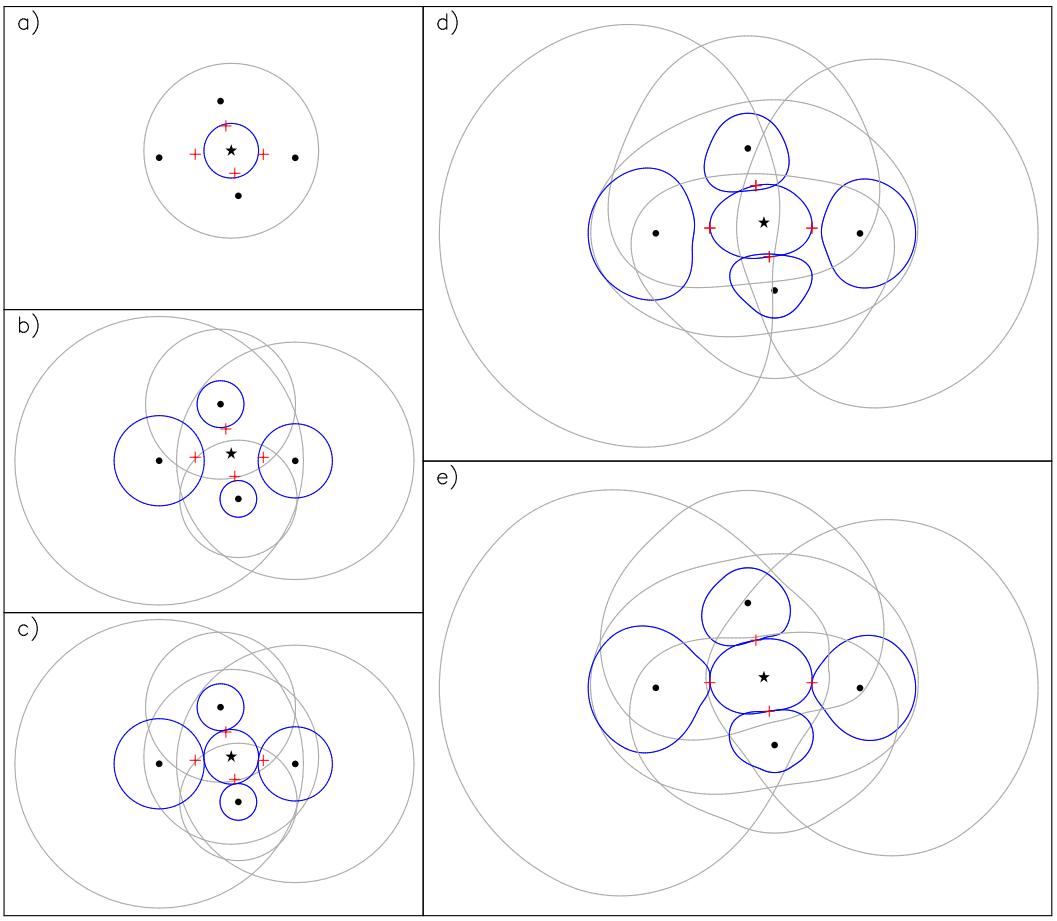

Figure 4.1: A graphic representation of the various interpretations of the smoothing function derivative, $\nabla_a W_{ab}$. Represented in each image plane is the same arrangement of pseudoparticles. The position of the particle of interest, $\mathbf{r}_a$, is given by a fixed black star and the position of all other particles, $\mathbf{r}_b$ (where $b \in \xi_a \neq a$), given by a number of fixed black points. Also displayed as blue and grey curves are, respectively, isolines at the same unspecified value of $\nabla_a W_{ab}$ and the compact support domains, ω_a and ω_b, of each particle factored into the particular interpretation. If $\nabla_a W_{ab} = -\nabla_b W_{ba}$ as required by Newton's third law as well as the derivation of the SP derivative approximations and the acceleration equation, the isolines would intersect at particular locations along the lines of sight between each pseudoparticle pair a and b, given by the red crosses. The image planes a) through d) represent, respectively, the *Gather*, *Scattered*, *Average* and *Merged* interpretations. Image plane e) represents the *Reflected Form*, presented in detail in Section 4.3.4.

4.3.3.1 General Definitions

Consider the derivative smoothing functions of each implementation, by substituting into equation (4.19) and/or equation (4.22). By far the simplest is Gather, formed by direct substitution of $H = h_a$, i.e.

$$\begin{aligned}\nabla_a^j W_{ab}(h_a) &= \frac{\partial W_{ab}(h_a)}{\partial r_a^j} + \frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(h_a)}{\partial h_a} \\ &= \left(\frac{r_{ab}^j}{|\mathbf{r}_{ab}|} - \frac{|\mathbf{r}_{ab}|}{h_a}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\right)\frac{\partial W_{ab}(h_a)}{\partial r} - \frac{\nu}{h_a}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}W_{ab}(h_a)\end{aligned} \tag{4.29}$$

Initially, Scattered appears equally simple, producing by substitution,

$$\begin{aligned}\nabla_a^j W_{ab}(h_b) &= \frac{\partial W_{ab}(h_b)}{\partial r_a^j} + \frac{\mathrm{d}h_b}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(h_b)}{\partial h_b} \\ &= \left(\frac{r_{ab}^j}{|\mathbf{r}_{ab}|} - \frac{|\mathbf{r}_{ab}|}{h_b}\frac{\mathrm{d}h_b}{\mathrm{d}r_a^j}\right)\frac{\partial W_{ab}(h_b)}{\partial r} - \frac{\nu}{h_b}\frac{\mathrm{d}h_b}{\mathrm{d}r_a^j}W_{ab}(h_b)\end{aligned} \tag{4.30}$$

However, given equation (4.27), equation (4.30) simplifies to give

$$\begin{aligned}\nabla_a^j W_{ab}(h_b) &= \left(\frac{r_{ab}^j}{|\mathbf{r}_{ab}|} - \delta^{ab}\frac{|\mathbf{r}_{ab}|}{h_b}\frac{\mathrm{d}h_b}{\mathrm{d}r_b^j}\right)\frac{\partial W_{ab}(h_b)}{\partial r} - \frac{\delta^{ab}\nu}{h_b}\frac{\mathrm{d}h_b}{\mathrm{d}r_b^j}W_{ab}(h_b) \\ &= \frac{r_{ab}^j}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}(h_b)}{\partial r} - \frac{\delta^{ab}\nu}{h_b}\frac{\mathrm{d}h_b}{\mathrm{d}r_b^j}W_{ab}(h_b)\end{aligned} \tag{4.31}$$

which should increase the complexity of the implementation, as the particle loops would have to include the a^{th} pseudoparticles effect on itself, if there is a non-zero smoothing length gradient about it. For the standard SP summation approximation (of some macroscopic variable, A), however, the effect of latter term can be ignored as $A_{aa} = A_a - A_a = 0$ and thus the $[aa]^{\text{th}}$ term of the summation cannot be non-zero no matter what definition the derivative takes when $a = b$. And, given the derivation of the equation of motion follows from the definition of the SP summation approximation, the latter term can be ignored completely. Therefore,

$$\nabla_a^j W_{ab}(h_b) = \frac{r_{ab}^j}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}(h_b)}{\partial r} \tag{4.32}$$

which appears very similar to the original smoothing function derivative, equation (2.17).

The derivative of the Merged perspectives follows a similar argument. Substi-

tuting $H = \overline{h_{ab}}$ into equation (4.19) yields

$$\nabla_a^j W_{ab}(\overline{h_{ab}}) = \frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r_a^j} + \frac{\mathrm{d}\overline{h_{ab}}}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}} \tag{4.33}$$

and, given equation (4.27), the smoothing length derivative expands by

$$\frac{\mathrm{d}\overline{h_{ab}}}{\mathrm{d}r_a^j} = \frac{\mathrm{d}}{\mathrm{d}r_a^j}\left[\frac{1}{2}(h_a + h_b)\right] = \frac{1}{2}\left(\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j} + \frac{\mathrm{d}h_a}{\mathrm{d}r_b^j}\right) = \frac{1}{2}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\left(1 + \delta^{ab}\right) \tag{4.34}$$

to give equation (4.35),

$$\nabla_a^j W_{ab}(\overline{h_{ab}}) = \frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r_a^j} + \frac{1}{2}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\left(1 + \delta^{ab}\right)\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}} \tag{4.35}$$

Repeating the implementation argument, and expanding by equation (4.22), the Merged derivative simplifies to give

$$\nabla_a^j W_{ab}(\overline{h_{ab}}) = \left(\frac{r_{ab}^j}{|\mathbf{r}_{ab}|} - \frac{|\mathbf{r}_{ab}|}{2\overline{h_{ab}}}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\right)\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r} - \frac{\nu}{2\overline{h_{ab}}}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}W_{ab}(\overline{h_{ab}}) \tag{4.36}$$

Finally, the Average derivative is simultaneously simpler and more complex than previous derivatives. Expanding,

$$\nabla_a^j \overline{W_{ab}(h_a, h_b)} = \nabla_a^j\left\{\frac{1}{2}\left[W_{ab}(h_a) + W_{ab}(h_b)\right]\right\} = \frac{1}{2}\left[\nabla_a^j W_{ab}(h_a) + \nabla_a^j W_{ab}(h_b)\right] \tag{4.37}$$

That is, the Average derivative is the mean average of the Gather and Scattered derivatives. Therefore,

$$\nabla_a^j \overline{W_{ab}(h_a, h_b)} = \frac{1}{2}\left[\frac{\partial W_{ab}(h_a)}{\partial r_a^j} + \frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(h_a)}{\partial h_a} + \frac{\partial W_{ab}(h_b)}{\partial r_a^j}\right] \tag{4.38}$$

which collapses to give,

$$\nabla_a^j \overline{W_{ab}(h_a, h_b)} = \frac{\partial}{\partial r_a^j}\overline{W_{ab}(h_a, h_b)} + \frac{1}{2}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(h_a)}{\partial h_a} \tag{4.39}$$

or the expanded form,

$$\nabla_a^j \overline{W_{ab}(h_a, h_b)} = \frac{1}{2}\left[\left(\frac{r_{ab}^j}{|\mathbf{r}_{ab}|} - \frac{|\mathbf{r}_{ab}|}{h_a}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}\right)\frac{\partial W_{ab}(h_a)}{\partial r} + \frac{r_{ab}^j}{|\mathbf{r}_{ab}|}\frac{\partial W_{ab}(h_b)}{\partial r} - \frac{\nu}{h_a}\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j}W_{ab}(h_a)\right] \tag{4.40}$$

4.3.3.2 Effect of Reflective Constraint

Referring back to Chapter 2, if the derivation from first principles holds then,

$$\nabla_a^j W_{ab}(H) = -\nabla_b^j W_{ba}(H') \tag{4.41}$$

becomes a constraint on the system, termed here the *reversibility constraint*, via the derivation of the SP derivative approximations (Section 2.2.6), and again through the derivation of the SP equation of motion (2.140) (Section 2.5.3). It is not clear, however, that equation (4.41) is required by the derivation from density approximation (Section 2.3).

Consider the derivation of $\frac{\partial \rho_b}{\partial r_a^j}$ (Price and Monaghan, 2004b). Given the definition of density, assuming the Gather interpretation,

$$\rho_b = \sum_{c \in \xi_b} m_c W_{bc}(h_b) \tag{4.42}$$

the first step in the derivation is to state that,

$$\frac{\partial \rho_b}{\partial r_a^j} = \frac{\partial}{\partial r_a^j}\sum_{c \in \xi_b} m_c W_{bc}(h_b) = \sum_{c \in \xi_b} m_c \frac{\partial W_{bc}(h_b)}{\partial r_a^j} \tag{4.43}$$

Now, assuming the smoothing length is some function of the mass density, e.g. equation (4.10), leads to the definition

$$\frac{\partial \rho_b}{\partial r_a^j} = \frac{1}{\Omega_b}\sum_{c \in \xi_b} m_c \frac{r_{bc}^j}{|\mathbf{r}_{bc}|}\frac{\partial W_{bc}(h_b)}{\partial r}\left(\delta^{ba} - \delta^{ca}\right) \tag{4.44}$$

where

$$\Omega_a = 1 - \frac{\partial h_a}{\partial \rho_a}\sum_{b \in \xi_a} m_c \frac{\partial W_{ab}(h_a)}{\partial h_a} \tag{4.45}$$

However, the first statement (4.43) assumes that the masses are sample data,

rather than variables and therefore the derivative cannot be given by

$$\frac{\partial \rho_b}{\partial r_a^j} = \sum_{c \in \xi_b} \frac{\partial}{\partial r_a^j} \left(m_c W_{bc}(h_b) \right) = \sum_{c \in \xi_b} m_c \frac{\partial W_{bc}(h_b)}{\partial r_a^j} + W_{bc}(h_b) \frac{\partial m_c}{\partial r_a^j} \tag{4.46}$$

but there is no proof of that assumption. In order for equation (4.46) to collapse to equation (4.43), the latter term must be equal to zero, and since $W \neq 0$ if $|\mathbf{r}_{ab}| < \kappa$,

$$\frac{\partial m_b}{\partial r_a^j} \equiv 0 \tag{4.47}$$

Though, assuming the mass conforms to a Kronecker delta interpretation, similar to equation (4.27),

$$\frac{\partial m_b}{\partial r_a^j} = \delta^{ab} \frac{\partial m_a}{\partial r_a^j} \tag{4.48}$$

and therefore, equation (4.46) reduces to

$$\frac{\partial \rho_b}{\partial r_a^j} = \sum_{c \in \xi_b} m_c \frac{\partial W_{bc}(h_b)}{\partial r_a^j} + \delta^{ab} W_{ac}(h_a) \frac{\partial m_a}{\partial r_a^j} \tag{4.49}$$

which is clearly not equal to the initial statement.

To prove the validity of the statement (4.43), it must be necessary to cause the summation to revert back to the integral form and follow the derivation of the SP derivatives. Thus, equation (4.41) exists for both derivations as the reversibility constraint. However, for each of the interpretations in the previous Section 4.3.3.1, the constraint does not hold.

For instance, consider the simplest interpretation, the Scattered perspective. In order for equation (4.41) to hold, $h_a \equiv h_b$. For a model with variable smoothing length, where $\max(h) - \min(h) \neq 0$, this is impossible. The more complex perspectives would require that $h_a \equiv h_b$ and $\frac{\mathrm{d}h_a}{\mathrm{d}r_a^j} \equiv -\frac{\mathrm{d}h_b}{\mathrm{d}r_b^j}$, or that all the variables involved (the smoothing lengths, particle configuration, the scale of particle separation, etc.) exist in perfect balance at all times, $t^{<n>} = \sum_{m=0}^{n-1} \delta_t^{<m>}$. All of these conditions are very unlikely to be possible. Even if the initial conditions (at $t^{<0>} = 0$) could be prescribed such that the constraint holds, there is no constraint within the dynamics equations that would ensure a valid solution at $t^{<1>} = \delta_t^{<0>}$. This qualitative argument occurs in quantitative form in Vila (1999), and accounts for the first of the 'problems' identified in Section 3.1 of that document.

In addition, without the reflective constraint enforcing Newtons third law, the applicability of the algorithm must come into question. There exists, also, a numerical facet to this problem. Specifically, it must become possible that, of a pair of

pseudoparticles a and b, $\nabla_a^j W_{ab} = 0$ while $\nabla_b^j W_{ba} \neq 0$. And thus a more accurate formulation must be required.

4.3.4 Revised Derivative (Reflected) Formulation

Given that each of the previous perspectives fails to formally ensure the reversibility constraint, proposed here is a revised formulation. This new derivative resulting from my research, is termed the *reflective form* – so named as it averages the outgoing derivative value from a to b with the incoming value from b to a is defined by

$$\overline{\nabla_{ab}^j} W_{ab} = \frac{1}{2}\left[\nabla_a^j W_{ab} - \nabla_b^j W_{ba}\right] \tag{4.50}$$

Note the negative used here. It must be negative to reverse the image of the kernel derivative created from b to a.

Clearly, this form guarantees that the reversibility condition is applied consistently, given

$$\begin{aligned}
\overline{\nabla_{ab}^j} W_{ab} &= -\overline{\nabla_{ba}^j} W_{ba} \\
\frac{1}{2}\left[\nabla_a^j W_{ab} - \nabla_b^j W_{ba}\right] &= -\frac{1}{2}\left[\nabla_b^j W_{ba} - \nabla_a^j W_{ab}\right] \\
\nabla_a^j W_{ab} - \nabla_b^j W_{ba} &= -\left[\nabla_b^j W_{ba} - \nabla_a^j W_{ab}\right] \\
\cancel{\nabla_a^j W_{ab}} - \cancel{\nabla_b^j W_{ba}} &= -\cancel{\nabla_b^j W_{ba}} + \cancel{\nabla_a^j W_{ab}}
\end{aligned}$$

Of considerable note is that this holds for *any* definition of the smoothing function; gather, scattered, average and merged. Recalling

$$\nabla_a^j W_{ab}(H) = \hat{r}_{ab}^j \frac{\partial W_{ab}(H)}{\partial r} + \frac{\mathrm{d}H}{\mathrm{d}r_a^j}\frac{\partial W_{ab}(H)}{\partial H} \tag{4.51}$$

and

$$\frac{\mathrm{d}h_p}{\mathrm{d}r_q^j} = \delta^{pq}\frac{\mathrm{d}h_p}{\mathrm{d}r_p^j} \tag{4.52}$$

under each definition the derivative becomes;

Gather:

$$\begin{aligned}
\overline{\nabla^j_{ab}} W_{ab} =& \frac{1}{2}\left[\nabla^j_a W_{ab}(h_a) - \nabla^j_b W_{ab}(h_b)\right] \\
=& \frac{1}{2}\left[\hat{r}^j_{ab}\frac{\partial W_{ab}(h_a)}{\partial r} + \frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_a)}{\partial h_a} - \hat{r}^j_{ba}\frac{\partial W_{ba}(h_b)}{\partial r} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\frac{\partial W_{ba}(h_b)}{\partial h_b}\right] \\
=& \frac{\hat{r}^j_{ab}}{2}\left[\frac{\partial W_{ab}(h_a)}{\partial r} + \frac{\partial W_{ab}(h_b)}{\partial r}\right] + \frac{1}{2}\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_a)}{\partial h_a} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\frac{\partial W_{ab}(h_b)}{\partial h_b}\right] \\
=& \hat{r}^j_{ab}\frac{\partial \overline{W_{ab}(h_a, h_b)}}{\partial r} + \frac{1}{2}\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_a)}{\partial h_a} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\frac{\partial W_{ab}(h_b)}{\partial h_b}\right]
\end{aligned} \tag{4.53}$$

Scattered:

$$\begin{aligned}
\overline{\nabla^j_{ab}} W_{ab} =& \frac{1}{2}\left[\nabla^j_a W_{ab}(h_b) - \nabla^j_b W_{ab}(h_a)\right] \\
=& \frac{1}{2}\left[\hat{r}^j_{ab}\frac{\partial W_{ab}(h_b)}{\partial r} + \frac{\mathrm{d}h_b}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_b)}{\partial h_b} - \hat{r}^j_{ba}\frac{\partial W_{ba}(h_a)}{\partial r} - \frac{\mathrm{d}h_a}{\mathrm{d}r^j_b}\frac{\partial W_{ba}(h_a)}{\partial h_a}\right] \\
=& \hat{r}^j_{ab}\frac{\partial \overline{W_{ab}(h_a, h_b)}}{\partial r} + \left[\delta^{ab}\frac{\partial W_{ab}(h_b)}{\partial h_b} - \delta^{ab}\frac{\partial W_{ab}(h_a)}{\partial h_a}\right] \\
=& \hat{r}^j_{ab}\frac{\partial \overline{W_{ab}(h_a, h_b)}}{\partial r}
\end{aligned} \tag{4.54}$$

Average:

$$\begin{aligned}
\overline{\nabla^j_{ab}} W_{ab} =& \frac{1}{2}\left[\nabla^j_a \overline{W_{ab}(h_a, h_b)} - \nabla^j_b \overline{W_{ab}(h_b, h_a)}\right] \\
=& \frac{1}{2}\left\{\frac{1}{2}\left[\nabla^j_a W_{ab}(h_a) + \nabla^j_a W_{ab}(h_b)\right] - \frac{1}{2}\left[\nabla^j_b W_{ba}(h_b) + \nabla^j_b W_{ba}(h_a)\right]\right\} \\
=& \frac{1}{4}\left[\hat{r}^j_{ab}\frac{\partial W_{ab}(h_a)}{\partial r} + \frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_a)}{\partial h_a} + \hat{r}^j_{ab}\frac{\partial W_{ab}(h_b)}{\partial r} + \frac{\mathrm{d}h_b}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_b)}{\partial h_b}\right. \\
& \left. -\hat{r}^j_{ba}\frac{\partial W_{ba}(h_b)}{\partial r} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\frac{\partial W_{ba}(h_b)}{\partial h_b} - \hat{r}^j_{ba}\frac{\partial W_{ba}(h_a)}{\partial r} - \frac{\mathrm{d}h_a}{\mathrm{d}r^j_b}\frac{\partial W_{ba}(h_a)}{\partial h_a}\right] \\
=& \frac{\hat{r}^j_{ab}}{4}\left[2\frac{\partial W_{ab}(h_a)}{\partial r} + 2\frac{\partial W_{ab}(h_b)}{\partial r}\right] + \frac{1}{4}\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}(1-\delta^{ab})\frac{\partial W_{ab}(h_a)}{\partial h_a} \\
& - \frac{1}{4}\frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}(1-\delta^{ab})\frac{\partial W_{ab}(h_b)}{\partial h_b} \\
=& \hat{r}^j_{ab}\frac{\partial \overline{W_{ab}(h_a, h_b)}}{\partial r} + \frac{1}{4}\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_a)}{\partial h_a} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\frac{\partial W_{ab}(h_b)}{\partial h_b}\right]
\end{aligned} \tag{4.55}$$

Merged:

$$\begin{aligned}
\overline{\nabla^j_{ab}} W_{ab} =& \frac{1}{2}\left[\nabla^j_a W_{ab}(\overline{h_{ab}}) - \nabla^j_b W_{ab}(\overline{h_{ab}})\right] \\
=& \frac{1}{2}\left[\hat{r}^j_{ab}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r} + \frac{\mathrm{d}\overline{h_{ab}}}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}} - \hat{r}^j_{ba}\frac{\partial W_{ba}(\overline{h_{ab}})}{\partial r} - \frac{\mathrm{d}\overline{h_{ab}}}{\mathrm{d}r^j_b}\frac{\partial W_{ba}(\overline{h_{ab}})}{\partial \overline{h_{ab}}}\right] \\
=& \hat{r}^j_{ab}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r} + \left[\frac{\mathrm{d}\overline{h_{ab}}}{\mathrm{d}r^j_a} - \frac{\mathrm{d}\overline{h_{ab}}}{\mathrm{d}r^j_b}\right]\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}} \\
=& \hat{r}^j_{ab}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r} + \left\{\frac{1}{2}\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a} + \frac{\mathrm{d}h_b}{\mathrm{d}r^j_a}\right] - \frac{1}{2}\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_b} + \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\right]\right\}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}} \\
=& \hat{r}^j_{ab}\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r} + \frac{1}{4}\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\right]\frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}}
\end{aligned} \tag{4.56}$$

Where $(1-\delta^{ab})$ is implicit in each term. Of particular note are the similarities between the all of the formulae. In fact, each of these gather, scattered and average derivatives could be expressed as

$$\overline{\nabla^j_{ab}} W_{ab} = \hat{r}^j_{ab}\frac{\partial \overline{W_{ab}(h_a, h_b)}}{\partial r} + \beta\left[\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a}\frac{\partial W_{ab}(h_a)}{\partial h_a} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b}\frac{\partial W_{ab}(h_b)}{\partial h_b}\right] \tag{4.57}$$

where β if given as $\frac{1}{2}$, 0 and $\frac{1}{4}$ for the gather, scattered and average perspectives, respectively. The only reason the reflected merged derivative has a *slightly* different definition is the interaction of the two smoothing lengths prior to the derivation – a phenomena that occurs during the definition of the other derivatives. In addition, note the relative simplicity of the reflected scattered derivative. It avoids the need to define/derive/approximate the smoothing lengths gradients and, as such, has a far simpler implementation. Also, in simulations with smoothly varying h the difference in gradient,

$$\frac{\mathrm{d}h_a}{\mathrm{d}r^j_a} - \frac{\mathrm{d}h_b}{\mathrm{d}r^j_b} \tag{4.58}$$

for most $[ab]$ values is likely to be very small, and thus the other reflected derivatives approximate the reflected scattered derivative.

For every other definition, a form must be found for the gradient of the smoothing length. As a dynamically varying pseudoparticle property, a definite form cannot be substituted into the formula and, instead, an approximation is required. Assuming no specific definition of the smoothing length, the simplest approach is a SP approximation, i.e.

$$\frac{\mathrm{d}h_p}{\mathrm{d}r^j_p} = \sum_{c=1}^{N_p}\frac{m_c}{\rho_c}h_{cp}\overline{\nabla^j_{pc}} W_{pc} \tag{4.59}$$

where $h_{cp} = h_c - h_p$ in the standard way. When substituted back into any of each of equation (4.53), equation (4.55) and equation (4.56), the reflected derivative for some specific $[ab]$ pair is defined as a function of itself and the derivatives formed between other pairs of pseudoparticles. Thus each must represent large sets of simultaneous equations (an equation for each $a = 1..N_p$; $b = 1..N_p$) and, therefore, the implementation of such a scheme must be relatively involved.

4.3.5 Implementation of Reflected Formulation

The reflective derivative form introduces a new system of equations to solve. For $a = 1..N_p$ and $b = 1..N_p$, there exists a system of unknowns, $\overline{\nabla_{ab}} W_{ab}(H)$, confined by

$$\begin{aligned} \hat{r}^j_{ab} \frac{\partial W}{\partial r} \left(1 - \delta^{ab}\right) = \overline{\nabla^j_{ab}} W_{ab} + \beta \Bigg(& \frac{\partial W_a}{\partial h} \sum_{c=1}^{N_p} \frac{m_c}{\rho_c} h_{ac} \overline{\nabla^j_{ac}} W_{ac} \\ & - \frac{\partial W_b}{\partial h} \sum_{c=1}^{N_p} \frac{m_c}{\rho_c} h_{bc} \overline{\nabla^j_{bc}} W_{bc} \Bigg) \left(1 - \delta^{ab}\right) \end{aligned} \tag{4.60}$$

This system is representative of a generalised form of the reflected gather, average and merged derivatives. The reflected scattered derivative has been omitted as it does not produce a set simultaneous equations and therefore most of the following arguments do not apply. Aside from some minor rearrangement, the previous implicit $(1 - \delta^{ab})$'s have been made explicit and the notation simplified such that

$$\frac{\partial W}{\partial r} = \begin{cases} \frac{\partial \overline{W_{ab}(h_a, h_b)}}{\partial r} & \text{for gather \& average} \\ \frac{\partial W_{ab}(\overline{h_{ab}})}{\partial r} & \text{for merged} \end{cases} \tag{4.61}$$

For the gather and average definitions

$$\frac{\partial W_a}{\partial h} = \frac{\partial W_{ab}(h_a)}{\partial h_a} \tag{4.62}$$

and

$$\frac{\partial W_b}{\partial h} = \frac{\partial W_{ab}(h_b)}{\partial h_b} \tag{4.63}$$

In the case of the merged definition,

$$\frac{\partial W_a}{\partial h} = \frac{\partial W_b}{\partial h} = \frac{\partial W_{ab}(\overline{h_{ab}})}{\partial \overline{h_{ab}}} \tag{4.64}$$

Also note,

$$\beta = \begin{cases} \frac{1}{2} & \text{for gather} \\ \frac{1}{4} & \text{for average \& merged} \end{cases} \tag{4.65}$$

In order to solve this system of simultaneous equations, it must be compressed into a form that resembles

$$A_{ij}x_j = b_i \tag{4.66}$$

This can be achieved if the $N_p \times N_p$ matrix components, $\overline{\nabla_{ab}}W_{ab}(H)$, $\frac{\partial W}{\partial r}$, $\frac{\partial W_a}{\partial h}$ and $\frac{\partial W_b}{\partial h}$ are forced into vectors by dictating that the k^{th} element of the vector is related to the $[ab]^{\text{th}}$ element of the original matrix by

$$k = k(a,b) = (a-1)N_p + b \tag{4.67}$$

It is worth noting that each k identifies a unique interaction (pair of a and b). Explicitly, the terms of equation (4.66) are given by

$$x_{j[=k(p,q)]} = \overline{\nabla^j_{pq}}W_{pq} \tag{4.68}$$

$$b_{i[=k(a,b)]} = \hat{r}^j_{ab}\frac{\partial W}{\partial r} \tag{4.69}$$

and

$$A_{i[=k(a,b)]j[=k(p,q)]} = \begin{cases} 1+\beta\frac{\partial W_a}{\partial h}V_b h_{ab} & \text{if} \quad p=a \wedge q=b \wedge p \neq q \\ \beta\frac{\partial W_a}{\partial h}V_c h_{ac} & \text{if} \quad p=a \wedge q=c \neq b \wedge p \neq q \\ -\beta\frac{\partial W_b}{\partial h}V_c h_{bc} & \text{if} \quad p=b \wedge q=c \wedge p \neq q \\ 0 & \text{otherwise} \end{cases} \tag{4.70}$$

where $V = \frac{m}{\rho}$ and $\wedge$ is a *logical and*.

This produces a system of N_p^2 unknowns, vector x_j, related by N_p^4 coefficients, in matrix A_{ij}, and N_p^2, in vector b_i. The unit memory required to store, rather than solve, this system is $N_p^4 + 2N_p^2$. Clearly, an attempt to solve this system, as it is, requires even more memory and a significant amount of computational time. Both need reducing as much as possible.

The equation offers several opportunities for such reductions. The simplest, and most obvious, come from the Kronecker deltas in equation (4.60). Namely, that

$$\overline{\nabla^j_{ab}}W_{ab} = \overline{\nabla^j_{aa}}W_{aa} = 0 \qquad \text{if} \qquad a = b \tag{4.71}$$

and thus the system of equations reduce to

$$\hat{r}^j_{ab}\frac{\partial W}{\partial r} = \overline{\nabla^j_{ab}}W_{ab} + \beta\left(\frac{\partial W_a}{\partial h}\sum_{\substack{c=1\\c\neq a}}^{N_p}\frac{m_c}{\rho_c}h_{ac}\overline{\nabla^j_{ac}}W_{ac} - \frac{\partial W_b}{\partial h}\sum_{\substack{c=1\\c\neq b}}^{N_p}\frac{m_c}{\rho_c}h_{bc}\overline{\nabla^j_{bc}}W_{bc}\right) \tag{4.72}$$

over $a = 1..N_p;\ b = 1..N_p \neq a$. Note that equation (4.71) is implicit in the original definition of equation (2.18) as

$$\overline{\nabla^j_{ab}}W_{ab} = -\overline{\nabla^j_{ba}}W_{ba} \tag{4.73}$$

and thus $2\overline{\nabla^j_{aa}}W_{aa} = 0$.

In addition, given equation (4.50), solutions are only required for the upper triangle of elements within the $\overline{\nabla^j_{ab}}W_{ab}$ matrix. This manifests itself in the system of equations as, for any $\overline{\nabla^j_{pq}}W_{pq}$, $p < q$ and thus

$$\begin{aligned}\hat{r}^j_{ab}\frac{\partial W}{\partial r} &= \left[1 + \beta h_{ab}\left(\frac{m_b}{\rho_b}\frac{\partial W_a}{\partial h} - \frac{m_a}{\rho_a}\frac{\partial W_b}{\partial h}\right)\right]\overline{\nabla^j_{ab}}W_{ab}\\ &+\beta\left[\frac{\partial W_a}{\partial h}\left(-\sum_{c=1}^{a-1}\frac{m_c}{\rho_c}h_{ac}\overline{\nabla^j_{ca}}W_{ca} + \sum_{\substack{c=a+1\\c\neq b}}^{N_p}\frac{m_c}{\rho_c}h_{ac}\overline{\nabla^j_{ac}}W_{ac}\right)\right.\\ &\left.+\frac{\partial W_b}{\partial h}\left(\sum_{\substack{c=1\\c\neq a}}^{b-1}\frac{m_c}{\rho_c}h_{bc}\overline{\nabla^j_{cb}}W_{cb} - \sum_{c=b+1}^{N_p}\frac{m_c}{\rho_c}h_{bc}\overline{\nabla^j_{bc}}W_{bc}\right)\right]\end{aligned} \tag{4.74}$$

for $a = 1..(N_p - 1);\ b = (a+1)..N_p$. Note also, that the $[ab]^{\text{th}}$ terms have been collected together.

Once again, if the matrices are compressed into vectors, this equation is still similar to equation (4.66). For the $[ab]^{\text{th}}$ element of some matrix, the position in the vector (or row or column of A_{ij}), k, is computed by

$$k = k(a,b) = (a-1)N_p + b - a - \frac{a(a-1)}{2} = (a-1)N_p + b - \frac{a(a+1)}{2} \tag{4.75}$$

For this configuration of the system, the number of unknowns is given by $\frac{N_p(N_p-1)}{2}$, the number of elements of A_{ij} is given by $\left(\frac{N_p(N_p-1)}{2}\right)^2$ and the unit storage memory is

$$\left[\left(\frac{N_p - 1}{2}\right)^2 + 1\right]N_p^2 - N \tag{4.76}$$

Further reductions follow from the set of equations. Consider that for each row of the matrix of coefficients A_{ij} (ie. NOT each unknown) there are a total of $\frac{1}{2}N_p(N_p-1)$ elements but the summations of equation (4.74) only encounter a subset of those. That is the only required coefficients of A_{ij}, where $i=k(a,b)$, are those where

$$j=i=k(a,b) \tag{4.77}$$

or

$$j=\begin{cases} k(a,c) & \text{if} \quad a<c \\ k(c,a) & \text{if} \quad c<a \end{cases} \tag{4.78}$$

or

$$j=\begin{cases} k(b,c) & \text{if} \quad b<c \\ k(c,b) & \text{if} \quad c<b \end{cases} \tag{4.79}$$

over $a=1..(N_p-1)$; $b=(a+1)..N_p$ and $c=1..N_p \neq a \neq b$. Thus, the number of c values is (N_p-2) and the number of *non-existent* elements, on a row of $T=T(N_p)$ elements is given by $T(N_p-2)$. For the system described in equation (4.74),

$$T(n)=\frac{n(n-1)}{2} \tag{4.80}$$

and the number of *existing* elements per row is given by

$$E(N_p)=T(N_p)-T(N_p-2)=2N_p-3\left(=1+\sum_{c=1}^{a-1}1+\sum_{\substack{c=a+1\\c\neq b}}^{N_p}1+\sum_{\substack{c=1\\c\neq a}}^{b-1}1+\sum_{c=b+1}^{N_p}1\right) \tag{4.81}$$

Re-compressing the system of equations, the total number of coefficients from the previous matrix A_{ij} in the new $E(N_p)\times T(N_p)$ matrix of existing coefficients, $\tilde{A}_{ij}$, is

$$E(N_p)T(N_p)=(2N_p-3)\left(\frac{N_p(N_p-1)}{2}\right)=N_p^3-\frac{5}{3}N_p^2+\frac{3}{2}N_p \tag{4.82}$$

This compression is clearly complex, but still deterministic so that functions could be written that relate elements in the compressed vectors and matrices to the original compression, and therefore the originals in equation (4.74). The positions in vectors x_j and b_i, and the rows, i, in $\tilde{A}_{ij}$ are still given by equation (4.75). However, the positions in the columns, j, of $\tilde{A}_{ij}$ are wholly more complex, being some complex stepwise function of a, b and i. It is simpler to, instead, store the values themselves in a second $E(N_p)\times T(N_p)$ matrix, $\tilde{k}_{ij}$. Therefore, the unit storage memory, U, is in the range

$$\frac{N_p}{2}(N_p^2-3N_p+1)\leq U\leq 2N_p(N_p-1)^2 \tag{4.83}$$

Configuration		Number of Interactions/Unknowns
1D		$N_{I,1} = N_x$
2D	Cubic	$N_{I,2c} = 4N_xN_y$
	Hexagonal	$N_{I,2h} = 3N_xN_y$
3D	Cubic	$N_{I,3c} = 9N_xN_yN_z$
	HCP	$N_{I,3h} = 8N_xN_yN_z$

Table 4.1: Formulae defining the number of interactions (and therefore the number of unknowns), N_I, for various particle configurations assuming periodic boundaries. HCP (Hexagonal Close Packing) – repeating layers of 2-Dimensional Hexagonal configurations stacked ABABAB, where B is A with some x and y offset. Also, $N_p = N_xN_yN_z$ where $N_z = 1$ if $\nu < 3$ and $N_y = 1$ if $\nu = 1$, otherwise $N_x, N_y, N_z > 2$. In addition, for Heptagonal and HCP either N_x or N_y must be even, and in the case of the later N_z must be even. All expressions are computed assuming $\kappa \approx 2h$ and $h \approx \frac{4}{5}\Delta_p$.

where the lower limit is for a fully determined implementation and the upper with the addition of a matrix $\tilde{k}_{ij}$.

Even though the remaining elements *exist*, they may still, and are likely to, be zero as the compact support of the smoothing function forces the derivatives $\frac{\partial W_{ab}}{\partial r_a^j}$ and $\frac{\partial W_{ab}}{\partial H}$ to zero. This not only reduces the number of coefficients but also reduces the number of unknowns as the equations (4.74) collapse to $\overline{\nabla_{pq}}W_{pq} = 0$ for many of the possible pairs of p and q in $a = 1..(N_p-1)$; $b = (a+1)..N_p$. These derivatives have become *known* and can be removed from x_j along with the j^{th} column of A_{ij} (note not $\tilde{A}_{ij}$).

If the compression is performed the number of unknowns reduces to the number of non-zero unknowns from the upper triangle of matrix $\overline{\nabla_{ab}}W_{ab}$, which is equal to the number of unique, non-zero particle interactions, N_I. For example figures see Table 4.1 and Table 4.2. Also, the number of elements in A_{ij} would be N_I^2.

However many of the values in A_{ij} would be zero as the non-zero interactions of one particle will not effect most of the other particles. So instead, the number of elements per row is far less, and the total number of elements is significantly reduced. See Table 4.3.

Configuration		Number of Interactions/Unknowns
1D		$N_{I,1} = N_x - 1$
2D	Cubic	$N_{I,2c} = (N_y - 1)[4(N_x - 2) + 5] + N_x - 1$
	Hexagonal	$N_{I,2h} = (N_y - 1)[3(N_x - 1) + 1] + N_x - 1$
3D	Cubic	$N_{I,3c} = (N_z - 1)\{N_x N_y + 2[N_y(N_x - 1) + N_x(N_y - 1)]\} + N_z N_{I,2c}$
	HCP	$N_{I,3h} = (N_z - 1)\{N_x + 2(N_x - 1)[3(N_y - 2) + 4] + (Ny - 1)[3(N_x - 1) + 2]\} + N_z N_{I,3h}$

Table 4.2: Formulae defining the number of interactions (and therefore the number of unknowns), N_I, for various particle configurations assuming non-periodic boundaries. HCP (Hexagonal Close Packing) – repeating layers of 2-Dimensional Hexagonal configurations stacked ABABAB, where B is A with some x and y offset. Also, $N_p = N_x N_y N_z$ where $N_z = 1$ if $\nu < 3$ and $N_y = 1$ if $\nu = 1$, otherwise $N_x, N_y, N_z > 2$. All expressions are computed assuming $\kappa \approx 2h$ and $h \approx \frac{4}{5}\Delta_p$.

$$\begin{aligned}\hat{r}_{ab}^{j}\frac{\partial W}{\partial r} &= \left[1 + \beta h_{ab}\left(\frac{m_b}{\rho_b}\frac{\partial W_a}{\partial h} - \frac{m_a}{\rho_a}\frac{\partial W_b}{\partial h}\right)\right]\overline{\nabla_{ab}^{j}}W_{ab} \\ &+\beta\left[\frac{\partial W_a}{\partial h}\left(-\sum_{\substack{c\in\xi_a \\ c<a}}\frac{m_c}{\rho_c}h_{ac}\overline{\nabla_{ca}^{j}}W_{ca} + \sum_{\substack{c\in\xi_a \\ c>a \\ c\neq b}}\frac{m_c}{\rho_c}h_{ac}\overline{\nabla_{ac}^{j}}W_{ac}\right)\right. \\ &\left.+\frac{\partial W_b}{\partial h}\left(\sum_{\substack{c\in\xi_a \\ c<b \\ c\neq a}}\frac{m_c}{\rho_c}h_{bc}\overline{\nabla_{cb}^{j}}W_{cb} - \sum_{\substack{c\in\xi_a \\ c>b}}\frac{m_c}{\rho_c}h_{bc}\overline{\nabla_{bc}^{j}}W_{bc}\right)\right]\end{aligned} \tag{4.84}$$

for $a = 1..(N_p - 1);\ b = (a + 1)..N_p$.

It should be noted that, prior to this reduction strategy, the size of each array was deterministic. As such arrays could be predefined in the code. In addition, the indices defining each element related – by some regular expression – to specific indices in the original equations (4.60). That is, the values held in the i^{th}, j^{th} and $[ij]^{\text{th}}$ elements of equation (4.66) (b, x and A, respectively) relate to the $[ab]^{\text{th}}$, $[ac]^{\text{th}}$ or $[bc]^{\text{th}}$ terms/variables in equations (4.60). However, now that the size of the arrays has been made a function of a dynamic system property, the pseudoparticle interactions, both the size and relationship between indices cannot be predetermined. It is therefore necessary to store not only the information held by each element, but the

Configuration		$\lvert\mathbf{N_a}\rvert$	$\lvert\mathbf{A}\rvert$ (periodic)	$\lvert\mathbf{A}\rvert$ (non-periodic)
1D		2	$2N_p$	$2(N_p-1)$
2D	Cubic	8	$32N_p$	$24-8N_p-11(N_x+N_y)$
	Hexagonal	6	$18N_p$	$6N_p-4(N_x+N_y)+2$
3D	Cubic	18	$162N_p$	$18N_p+4(N_x+N_y+N_z)$ $-10(N_xN_y+N_xN_z+N_yN_z)$
	HCP	18	$144N_p$	—

Table 4.3: The typical number of neighbours (excluding the particle of interest), $|Na|-1$, and formulae detailing the approximate number of non-zero coefficients in array A_{ij}, $|A|$, as a function of the particle configurations listed in Tables 4.1 and 4.2. The final entry under *3D HCP* – $|A|$ (non-periodic) – is missing as its form fluctuates in tandem with the N_x, N_y and Nz values. Specifically, the form is a function of the divisibility of each by the lowest three primes (1, 2 and 3). As with Tables 4.1 and 4.2, all expressions assume that $\kappa \approx 2h$ and $h \approx \frac{4}{5}\Delta_p$.

i and/or j values as well. Though i and j (in A) could be combined to form k by

$$k = (i-1)N_I + j \tag{4.85}$$

Any information about known x_j need not be stored as, to be known at his point, they must be zero.

No further reductions are possible from the general equations. If there are other possible simplifications, they will come from the specific variable quantities generated during a simulation. For instance, if (assuming the merged definition of smoothing function) $\frac{\partial W_{pq}}{\partial h} = 0$ for some specific pseudoparticle pair p and q, then the $[pq]^{\text{th}}$ equation would reduce to

$$\overline{\nabla^j_{pq}} W_{pq} = \hat{r}^j_{pq} \frac{\partial W_{pq}}{\partial r} \tag{4.86}$$

and therefore the k^{th} (as a function of p and q) unknown, x_k, has become a known! Thus the k^{th} column of A may be removed, provided that the vector b is rewritten such that

$$[b_i]_{\text{new}} = [b_i]_{\text{old}} - A_{ik}x_k \tag{4.87}$$

Alternatively, all the h values in the neighbourhood of some particle of interest may be equal, and therefore

$$h_a = h_b = h_c \tag{4.88}$$

for all indices encountered by equations (4.84). Once again, x_k has become known

and the set of equations may be simplified following a similar rewriting of b.

It is also possible that, after all of these types of effect have been taken into account, some other $[pq]^{\text{th}}$ equation may have been simplified to the point where only one unknown remains, $\overline{\nabla^j_{pq} W_{pq}}$. The $[pq]^{\text{th}}$ equation may now be solved externally of the simultaneous equation solving routine. In such a way – if these considerations are applied – an algorithm could be applied recursively until all possible knowns are found.

These simulation-specific specific routines have the potential to reduce the memory required at the cost of some amount of computation. Whether the benefits of the reduced memory requirements (and hence reduced computation within whatever method is utilised to solve the remaining system of simultaneous equations) out-weigh the impediments of increased computation is entirely dependant on the dynamics of the system. It is certainly unlikely that the variables will adhere to similar, stable values throughout an entire simulation and there will be no knowns to identify, making the identifying routines superfluous. Whether to include these modifications must depend on specific models and implementations.

Assuming nothing about the application of the model, in order to solve the system of equations (4.84), some numerical scheme must be included. Though explicit method exist to invert A and, hence, solve the system of equations (e.g. LUP decomposition) it is more practical to implement some iterative technique better suited to sparse matrices. In addition, iterative methods also allow for easier optimisation for specific problems. Many such algorithms exist (see Table 4.4).

Iterative methods follow from the following argument. Some equation (4.66), $Ax = b$, may be expressed as

$$Qx = (Q - A)x + b \tag{4.89}$$

where Q is the splitting matrix, and an iterative process is formed by rearranging and defining

$$x^{<k>} = Q^{-1}[(I - A)x^{<k-1>} + b] = (I - Q^{-1}A)x^{<k-1>} + Q^{-1}b \tag{4.90}$$

Q is usually chosen such that it simplifies a problem and/or ensures convergence. It can be shown (Kincaid and Cheney, 2002) that equation (4.90) will always converge to the true solution if $\|I - Q^{-1}A\| < 1$. Therefore, any implementation of a numerical solution to equation (4.84) must be written such that it holds to this condition.

Given the relatively high memory requirements of the method, the Gauss-Seidel algorithm was chosen as it is simple but does not involve remembering and *old*,

Method	Q	G
Richardson	I	$A - I$
Jacobi	D	$D^{-1}(C_L + C_U)$
Gauss-Seidel	$D - C_L$	$(D - C_L)^{-1}C_U$
SOR	$\omega^{-1}(D - \omega C_L)$	$(D - \omega C_L)^{-1}(\omega C_U + (1 - \omega)D)$
And so on, with increasing complexity...		

Table 4.4: Description of simple iterative methods for solving simultaneous equations. These are more complex schemes beyond this, most notably, *symmetric successive over relaxation* (SSOR) method – which introduces a half-step $x^{<k-\frac{1}{2}>}$ to each estimation. Here, $D = \text{diag}(A)$, C_L and C_U are the strictly lower and upper parts of A, respectively, $0 \leq \omega \leq 2$ is some weighting factor and Q is the splitting matrix. G is defined by $x^{<k>} = Gx^{<k-1>} + c$ and is, therefore, usually defined by $G = I - Q^{-1}A$. Note that $\|G\| < 1$.

$x^{<k-1>}$ and *new*, $x^{<k>}$, versions of the vector of unknowns. The Gauss-Seidel algorithm is reasonably common, however the inclusion of this method within the SP model is novel and requires some discussion.

At its most basic, the implementation discussed here replaces the *Compute* $\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}$ as previously described (Section 3.5.1) with a new version that breaks down into the following four general steps;

i) Identify the unknowns $\left(\overline{\nabla_{ab}^{j}}W_{ab}\right)$,

ii) Compute the arrays required by the Gauss-Seidel routine

iii) Solve for $\overline{\nabla_{ab}^{j}}W_{ab}$ (run the Gauss-Seidel routine), and

iv) Calculate $\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}$

An unknown exists for every unique, non-zero pseudoparticle interaction (pair $[ab]$), therefore identification of the unknowns merely requires identification of these pseudoparticle pair. This is done in exactly the same way as Figure 3.10, whereby a sweep is conducted over all possible pairs limited by some measure of optimisation (the bucket sort algorithm), and $|\mathbf{r}_{ab}|$ is computed and tested. If the conditions prove favourable, the routine makes it to the node marked as χ and a pair has been identified. For greater efficiency later, all components of the calculation that are a function of distance are also calculated and stored, including the vector b_i for the Gauss-Seidel routine.

Subsequently, the remaining variables required by the Gauss-Seidel routine must be computed. This task is in itself quite complex as the algorithm has been optimised to reduce the memory constraints as far as possible. Consider the compression in isolation. It is achieved by compressing some sparse matrix A_{ij} into three vectors;

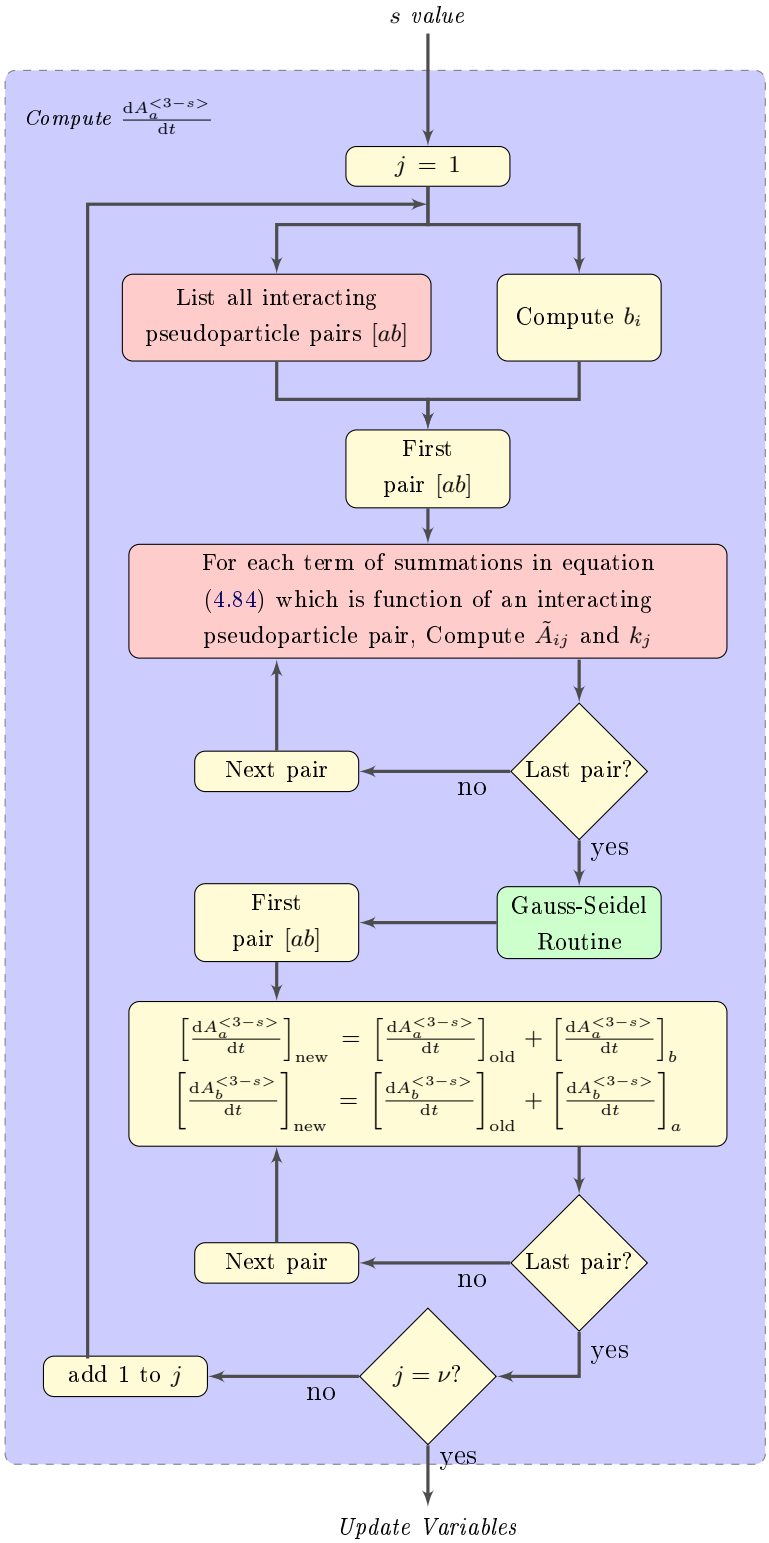

Figure 4.2: A generalised code structure diagram showing the routine implementing the reflected formulation of the smoothing function derivative. The red nodes present summaries of significantly more complex routines. The green node, *Gauss-Seidel Routine*, encapsulates the Gauss-Seidel algorithm as described in Section 4.3.5.

the non-zero values sit in a vector $\tilde{A}_k$, of length N_k; the original j values sit in the equivalent position in a vector of equal size, L_k; and, the final vector, Γ_i, stores the number of non-zero values per row, and is therefore of length N_i (the number of unknowns). For example,

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & a_{25} & a_{26} \\ a_{31} & 0 & a_{33} & a_{34} & 0 & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & 0 \\ a_{61} & 0 & a_{63} & a_{64} & 0 & a_{66} \end{bmatrix} \tag{4.91}$$

is transformed to become

$$\tilde{A} = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{14} \\ a_{21} \\ a_{25} \\ a_{22} \\ a_{26} \\ a_{31} \\ a_{36} \\ a_{34} \\ a_{33} \\ a_{41} \\ a_{42} \\ a_{43} \\ a_{44} \\ a_{45} \\ a_{55} \\ a_{54} \\ a_{61} \\ a_{63} \\ a_{66} \\ a_{64} \end{bmatrix} \quad \& \; L = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 1 \\ 5 \\ 2 \\ 6 \\ 1 \\ 6 \\ 4 \\ 3 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 5 \\ 4 \\ 1 \\ 3 \\ 6 \\ 4 \end{bmatrix} \quad \& \; \Gamma = \begin{bmatrix} 3 \\ 4 \\ 4 \\ 5 \\ 2 \\ 4 \end{bmatrix} \tag{4.92}$$

where Γ_i is the number of non-zero values on the i^{th} row of A. $\tilde{A}$ stores the non-zeros from each row in no particular order, however, the block of non-zeros from

the i^{th} row of A (the i^{th} block) follows the $(i-1)^{\text{th}}$ and precedes the $(i+1)^{\text{th}}$ blocks. To be clear, L_c holds the column, j, in A where the non-zero value $\tilde{A}_c$ occurs. In addition, the i^{th} component of b and and row of A are divided by A_{ii}. The subsequent optimised Gauss-Seidel algorithm is given by

```
input N_i, N_k, Ã, L, Γ, b
x = b
start ∞ loop
    u = x
    c = 0
    for i = 1 .. N_i ; do
        s = 0
        for j = 1 .. Γ_i ; do
            if c = c + 1
            s = s + Ã_c x_{L_c}
        end j
        x_i = b_i − s + x_i
    end i
    if (|x − u| < tolerance); end ∞ loop
jump back to "start ∞ loop"
output x
```

Given that the compressed vectors required by this routine are populated by a complex, multicomponent equations that is itself compressed, it is clear that the implementations of this step is not simple. In general terms, the procedure loops through the previously identified pseudoparticle pairs, then through all those indicated by the summations in equation (4.84). For each pair that has been shown to interact, the appropriate terms in each of the four vectors $(\tilde{A}, L, \Gamma, b)$ are calculated. Once complete, all of this information is fed into Gauss-Seidel routine. One final pass over the interacting pseudoparticle pairs to calculate the rates of change, and the revised computation is complete.

Clearly, this procedure is complex, relative to the standard SP implementation, and also quite unwieldy. Further work is required to either significantly reduce the computational weight (both in run time and memory), or justify the inconsistency making the reflective constraint necessary.

4.4 Numerical Artifact Flow

Consider the visualisation of the results from a Hydrodynamic SP algorithm built with variable smoothing length, given by equation (4.10). Specifically, the magnitude of velocity, $|\mathbf{v}|$, in a 2-Dimensional domain at some arbitrary time, $t > 0$, shown in Figure 4.3. The simulations initial conditions were simple. For a regular distribution of pseudoparticles ($|\mathbf{r}_{ab}| \equiv \Delta$), a smoothed discontinuity in density at $x_0 = 0$ was created, in the standard way (Monaghan, 2005), in the x-axis by

$$\rho_a^{<0>} = \frac{\rho_L + \rho_R e^{\left(x_a^{<0>} - x_0\right)/\Delta}}{1 + e^{\left(x_a^{<0>} - x_0\right)/\Delta}} \tag{4.93}$$

where (for general cases)

$$\Delta = \begin{cases} \Delta_L & \text{if} \quad x_a^{<0>} < x_0 \\ \Delta_R & \text{if} \quad x_a^{<0>} > x_0 \end{cases} \tag{4.94}$$

and variables with subscript L or R are the left and right states, either side of the discontinuity, respectively. Note that there is no definition of Δ at $x_a^{<0>} = x_0$ because, regardless of it's non-zero value, the exponentials in equation (4.93) return a value of $1 (= e^0)$.

In addition the mass is held constant, $\mathbf{v}^{<0>} = 0$, and the initial pressure was forced constant by defining the initial thermal energy such that

$$u_a^{<0>} \propto \frac{1}{\rho_a^{<0>}} \tag{4.95}$$

and also has, therefore, a similar though opposing discontinuity as in equation (4.93).

There is also a Gaussian pressure pulse (with complementary density and thermal energy profiles) at the centre of the left-hand portion of the domain, coordinates $\left\{\left[\frac{1}{4}(\uparrow x + 3 \downarrow x)\right], \left[\frac{1}{2}(\uparrow y + \downarrow y)\right]\right\}$. This phenomena exists without relevance herein, and remains as a reminent of the original intention of the simulation; to test the ability of the SP algorithm to model acoustic wave refraction.

Referring to equation (2.81), the acceleration is a function of the stress gradient. As the pressure is constant there is no stress gradient, and therefore there should be no deviation from the initial velocity, i.e. $\mathbf{v}_a^{<n>} = \mathbf{v}_a^{<0>} = 0$. However, as can be clearly seen in the figure, the velocity in non-zero along $x = 0$ (the discontinuity). All the more troubling is the relative direction of the flow – against the density gradient. In essence the material is trying to flow *up hill*. This phenomena is clearly a non-physical, numerical artefact flow.

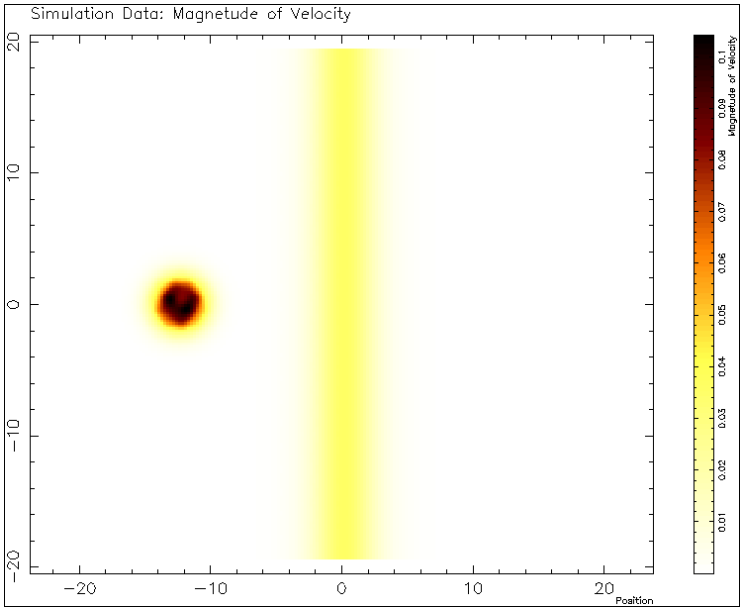

Figure 4.3: A plot of the magnetude of velocity, in 2-Dimensions, of a hydrodynamic SP test simulation (notice the gaussian pressure pulse that is superfluous to the current discussion topic). The domain is split into two halves, left and right of x=0. Each half has dissperate physical properties, but equal pressure (and therefore no cause to accelerate accross the interface between the two states). Clearly, however, there is motion accross the discontinuity. This is source of the phenomenon identified in this document as the Numerical Artefact Flow.

In order to understand the source of this artefact, and therefore understand the ramifications of the artefact, consider initially a 1D system of identical pseudoparticles being modelled using a standard, constant-h, SP algorithm. In such a system, each pseudoparticle would be positioned equidistant from its neighbours and therefore the pseudoparticle subdomains, ω, will be equally embedded into each other, given that $|\mathbf{r}_{ab}| = \overline{\Delta}$ and $h \propto \Delta$. This is shown graphically in Figure 4.4 for ten pseudoparticles, where the black points are the centres of each, $\mathbf{r}_a$, and the curves show the extent of the kernels (the surface of each subdomain, $s(\omega)$), assuming $\kappa = h$. In reality $\kappa > h$, usually by a factor of 2, and has been reduced in Figure 4.4 and Figure 4.5 only for clarity.

Now consider a change to the system, such that $\frac{\partial \rho}{\partial x} = c$, where $c \neq 0$, and define that

$$\frac{\partial \rho}{\partial x} \propto \left(\frac{\partial u}{\partial x} \right)^{-1} \tag{4.96}$$

such that ρu is constant. Under the original definition of h as a constant, there is no change to the pseudoparticle arrangement in Figure 4.4. However, when the

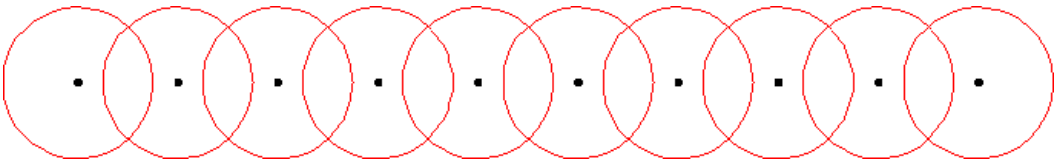

Figure 4.4: Representation of the initial conditions, constructed using original SP theory, of ten pseudoparticles in a 1-dimensional simulation with or without a positive density gradient (left to right). The only properties shown here are the pseudoparticle positions (in black) and the furthest extents of the kernels (in red).

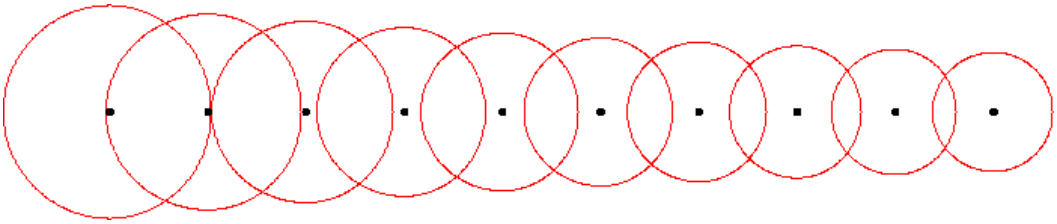

Figure 4.5: Representation of the initial conditions, where the positions are constructed using original SP theory and the smoothing lengths are defined using equation (4.10), of ten pseudoparticles in a 1-dimensional simulation where there is a positive density gradient (left to right). The only properties shown here are the pseudoparticle positions (in black) and the furthest extents of the kernels (in red).

constant from is exchanged for equation (4.10), the smoothing function extents change dramatically, see Figure 4.5. The pseudoparticles are now clearly unequally embedded.

It is this unequal embedding that leads to the artefact. For the purposes of this explanation, and given the initial identification of the numerical artefact, consider only the motion of the pseudoparticles as a function of the SP acceleration equation (2.140). The motion of a single pseudoparticle, a, is given by considering the sum of the line-of-sight stresses place upon it by the interplay with the pseudoparticles, b, that are part of the set ξ_a – its neighbouring pseudoparticles, those within the subdomain ω_a – modified by the relative distance between them, $|\mathbf{r}_{ab}|/H$. The imbalance in embedding, caused by the incorrect positioning, induces a non-physical acceleration against the density gradient towards accurate positions.

It is worth noting that this flow could be interpreted as the mis-initialisation of the code, and that the simulation merely began, and subsequently remained, non-physical. This is the essence of the argument to include a position relaxing algorithm to any SP set-up codes (Price, 2008). However, there is no justification for *assuming*

this is the case, as unless the source of the artefact is investigated there could be issues that the relaxation of position do not solve.

As we shall see, this is in fact the case.

4.4.1 Quantification

In order to identify the source of the artefact, consider a system of pseudoparticles with constant, non-zero stress throughout the domain, $S_a^{ij} = S_b^{ij} = S^{ij} \neq 0$. The stress gradient, and therefore the acceleration, should be zero in such a system, i.e.

$$\left[\frac{\partial S_a^{ij}}{\partial r_a^j}\right]_{\text{SP}} = \sum_{b\in\xi_a} \frac{m_b}{\rho_b}\left(S_a^{ij} + S_b^{ij}\right)\overline{\nabla_{ab}}W_{ab} = 2S^{ij}\sum_{b\in\xi_a}\frac{m_b}{\rho_b}\overline{\nabla_{ab}}W_{ab} = 0 \tag{4.97}$$

Given that $2S^{ij} \neq 0$, for this relation to hold true,

$$\sum_{b\in\xi_a}\frac{m_b}{\rho_b}\overline{\nabla_{ab}}W_{ab} = 0 \tag{4.98}$$

Price (2012) describes this additional term as an implicit re-meshing scheme that exists to maintain some measure of order in the pseudoparticle arrangement. However, it would be fallacious to assume without a rational or a proof that there exists a solution to equation (4.98) for every system. Therefore a new relation must be defined,

$$I_a^j = \sum_{b\in\xi_a} I_{ab}^j = \sum_{b\in\xi_a}\frac{m_b}{\rho_b}\overline{\nabla_{ab}}W_{ab} \tag{4.99}$$

This may be interpreted as the net interaction of particle a in the j^{th} dimension, where the direction of interaction between particles a and b is given by $\text{sgn}\left(I_{ab}^j\right)$ and the magnitude of such interaction is given by $\left|I_{ab}^j\right|$. I_a^j is a quantitative measure of the numerical artefact. Ideally, there should be no net interaction ($I_a^j = 0$).

However, recall that equation (4.99) is found by assuming no variability in the system stress, and that if this were the case that approximating the equations would be superfluous as the system could be described analytically. In reality, the stress could not be separated from the numerical artefact (where it exists). As a result of this analysis though, it is clear that the artefact is a function of the relative interactions of the pseudoparticles. This is of extreme importance as the interactions effect every SP derivative approximation. Take, for example, the SP continuity equation (2.112), induction equation (2.118) and equation of motion equation (2.140), which

can be rewritten in terms of the interactions, I^j_{ab}, as

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \rho_a \sum_{b\in\xi_a} v^j_{ab} I^j_{ab} \tag{4.100}$$

$$\frac{\mathrm{d}B^i_a}{\mathrm{d}t} = \sum_{b\in\xi_a} \left(B^i_a v^j_{ab} - B^j_a v^i_{ab} \right) I^j_{ab} \tag{4.101}$$

$$\frac{\mathrm{d}v^i_a}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{b\in\xi_a} \left(S^{ij}_a + S^{ij}_b \right) I^j_{ab} \tag{4.102}$$

Proving that the numerical artefact not only effects motion, but also degrades the quality of the magnetic field and the density.

4.4.2 Artifact in a 1-Dimensional Domain

For any 1D system of particles, assuming $\kappa \approx 2h$ and $h \approx \frac{4}{5}\Delta_p$, a solution to equation (4.99) exists such that $\delta_a = 0$. This is simply argued. Consider that any particle, p, in the system only interacts with the two particles either side of it, $(p-1)$ and $(p+1)$ and given that $\overline{\nabla_{pp}} W_{pp} \equiv 0$, the criterion (4.99) reduces to;

$$\delta_p = \frac{m_{(p-1)}}{\rho_{(p-1)}} \overline{\nabla_{p,(p-1)}} W_{p,(p-1)} + \frac{m_{(p+1)}}{\rho_{(p+1)}} \overline{\nabla_{p,(p+1)}} W_{p,(p+1)} = 0 \tag{4.103}$$

Thus, if ρ and m are prescribed or are defined by some smooth, physical function of space, and the positions of two adjacent particles are sensibly prescribed, then equation (4.103) can be solved to iteratively position the remaining particles such that they are equally embedded. Consider, for example, that mass and density for all particles are proscribed, and that the positions of $a = p = 2$ and $a = (p-1) = 1$ are prescribed. Therefore the position of particle $a = (p+1) = 3$ can be found by solving;

$$\frac{m_1}{\rho_1} \overline{\nabla_{21}} W_{21} + \frac{m_3}{\rho_3} \overline{\nabla_{13}} W_{13} = 0 \tag{4.104}$$

using a root finding algorithm where the unknown is r_3. Subsequently, particle $a = 4$ can be found by solving;

$$\frac{m_2}{\rho_2} \overline{\nabla_{32}} W_{32} + \frac{m_4}{\rho_4} \overline{\nabla_{34}} W_{34} = 0 \tag{4.105}$$

And so on, until the position of $a = N_p$ is found. For the ten particles discussed above, this would look result in the positions in blue on Figure 4.6.

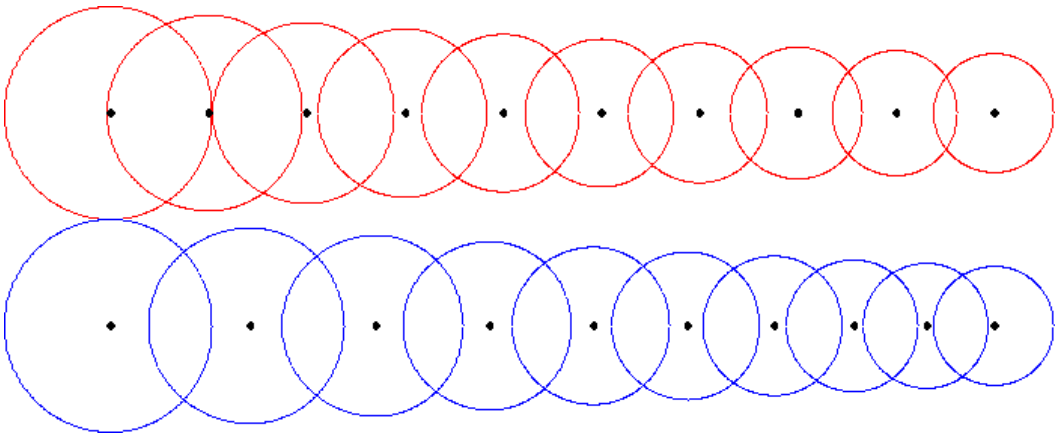

Figure 4.6: Comparison of the initial conditions of the ten particles shown in Figure 4.5, in red, and the corrected conditions calculated using a root-finding algorithm, in blue. Both are 1D systems with positive density gradient (left to right). The black points indicate the particle positions, r_a, and the lines (red and blue, for incorrectly and correctly embedded particle conditions respectively) show the furthest extents of the kernels.

4.4.3 Artifact in Multidimensional Domain

Multi-dimensional systems are altogether more complicated. The following arguments, though only discussed in terms of 2D systems, apply to all multi-dimensional applications of SP theory with variable smoothing lengths.

4.4.3.1 Underdetermined Systems

Assuming that, for some systems initial conditions, there is a solution to equation (4.99) that ensures $I_a^j = 0$, it may still not be possible to discover the exact nature of that solution as, unlike 1D systems, the number of unknowns in most configurations will far outnumber knowns. The knowns take the form of the particle properties that effect the amount of interaction (other than position) which, in combination, equal the number of particles. While the unknowns are the vectors connecting each particle to its neighbours. For a 2D system where m_a and ρ_a are prescribed, consider the approximate placement of the first three particles (Figure 4.7). Initially there are an equal number of unknowns (blue lines representing the vectors) to knowns (red dots representing the particles). However when a realistic number of particles (to accurately model the system) are considered, the number of vectors far outstrips the number of particles, see Figure 4.8. For explicit definitions for various simple, regular configurations see Table 4.3. Quantification of this behaviour is given in Section 4.4.3.3.

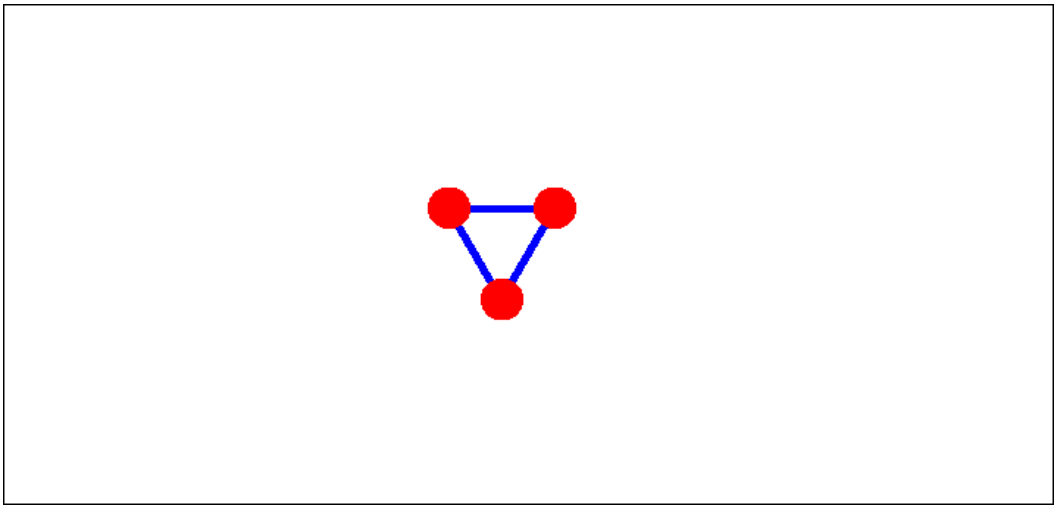

Figure 4.7: A graphic depicting the first three particles (red dots) in a hexagonal lattice and the three vectors (blue lines) connecting them.

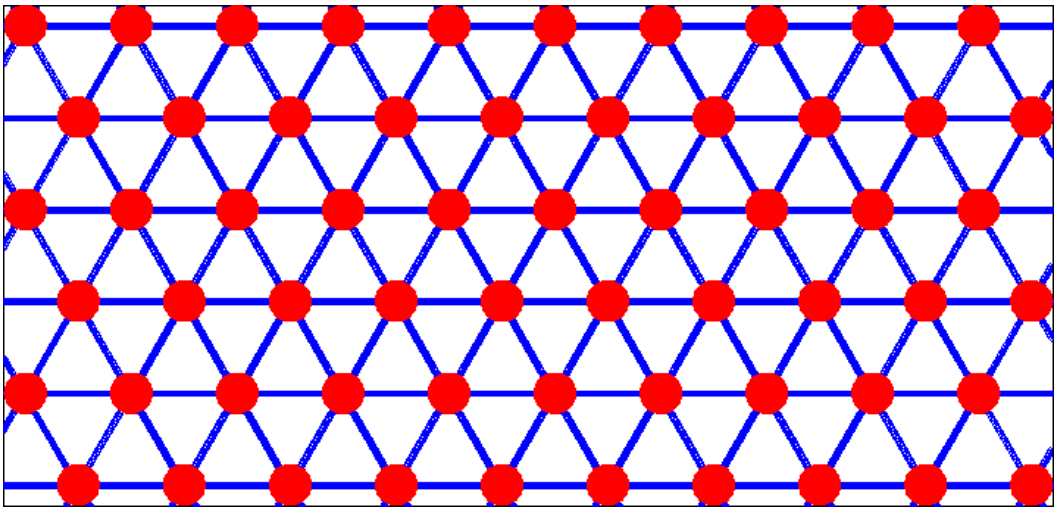

Figure 4.8: A graphic depicting a space full of the particles (red dots) of a hexagonal lattice and the vectors (blue lines) that connect them to their neighbours. It should be noted that the number of vectors is much greater than the number of particles, and that there are, therefore, more unknowns to knowns in the system which infers that the true positions cannot be found analytically.

From this imbalance it is possible to infer that multi-dimensional simulations are underdetermined systems (where there is no analytical solution to equation (4.99)), and that only a numerical approach can be used to consistently solve for the approximate configuration of particles.

4.4.3.2 Restrictive Phenomena

There are further problems with multi-dimensional simulations induced by the artefact flow than just the impossibility of a general analytical solution. Not the least of which is that for almost all multi-dimensional pseudoparticle configurations, equation (4.99) has no valid solutions (where $I_a^j = 0$). Even those, very rare, initial conditions that do satisfy the no-net-interaction constraint are expected to become invalid once the pseudoparticles set in motion. This can be simply argued.

There are two phenomena that restrict the arrangement of pseudoparticles to non-zero net iteration configurations; the inhibition of deformation to prescribed property geometries and the existence of impossible pseudoparticle clusters. Simulations are constructed to model how specific situations evolve over time. Therefore each system of pseudoparticles has some prescribed pattern of properties. Consider seven pseudoparticles of a 2D system in which $0 < \frac{\partial \rho}{\partial x} = \text{const}$, $\frac{\partial \rho}{\partial y} = 0$ and the smoothing lengths are defined by equation (4.10). If the gradients are treated as sacrosanct, the pseudoparticles must be placed equidistantly, as in Figure 4.9a. However, these pseudoparticles are clearly unequally embedded and, therefore, $I_a^j \neq 0$. If the pseudoparticles can be arranged in space so that they are equally embedded, the geometry of the system suffers deformation. In this case the pseudoparticles are forced to curve causing the density gradient to alter dramatically such that there is a non-constant gradient in both x and y, in opposition to the situation intended to be modelled, see Figure 4.9b.

A pseudoparticle cluster is a group of pseudoparticles within a system that all interact with one, single pseudoparticle, often at the centre of the cluster. Impossible clusters are groups of pseudoparticles that cannot be positioned so that for the central pseudoparticle $I_a^j = 0$. Though there are a near infinite number of impossible cluster configurations, each one can be assigned to one of two categories where either the central pseudoparticle is too large (Figure 4.10a) or to small (Figure 4.10b) to allow an arrangement with zero net-interaction. In each figure, five of the six surrounding pseudoparticles have been positioned around the central one, leaving a gap that must be filled by the seventh pseudoparticle. For impossible clusters where the central pseudoparticle is too large, the gap is similarly too large to allow the remaining pseudoparticle to be positioned consistently. The inverse is true of

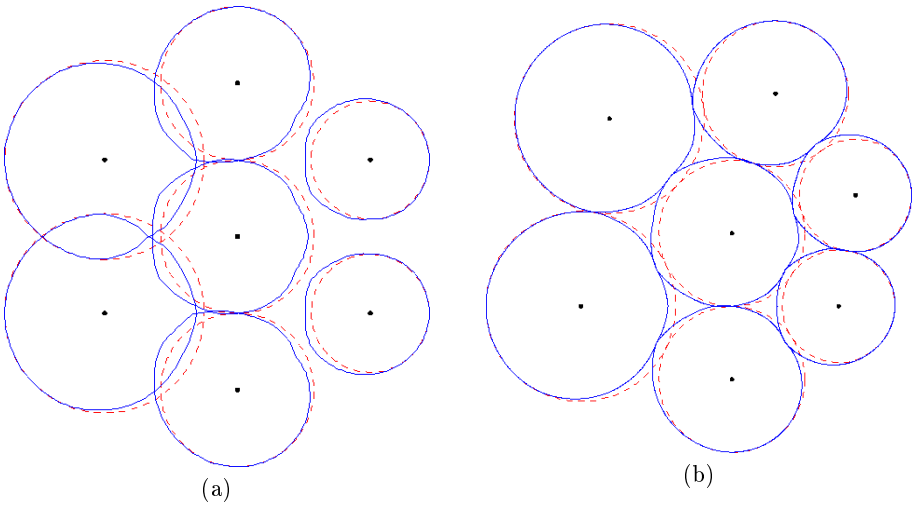

Figure 4.9: Representations of seven interacting pseudoparticles (with variable smoothing length) in a 2D system. Initially the pseudoparticles are arranged in a hexagonal lattice that holds to the initial conditions of the system, $0 < \frac{\partial \rho}{\partial x} = \text{const}$ and $\frac{\partial \rho}{\partial y} = 0$, image (a). However, this results in unequal embedding. The positions are subsequently manipulated such that the pseudoparticles are equally embedded, image (b). In this case, the geometry of the system is deformed and the initial conditions are lost. The black points represent the position of each pseudoparticle, r_a. The lines are depict the extents of equal interaction (such that touching lines infer equal embedding) around each pseudoparticle. For each pseudoparticle, the red, dashed line shows what the extent would be if the pseudoparticle was surrounded by identical pseudoparticles of the same type, while the blue, solid line shows the extent as it has been deformed by the actual pseudoparticles surrounding it.

clusters where the central pseudoparticle is small, leaving a gap that is far too small. Given a sufficiently dynamic system, such as those that require a numerical solution, it is highly likely that at least one impossible cluster of pseudoparticles will exist, and therefore invalidate the pseudoparticle system.

4.4.3.3 Possible Corrective Formulations

Equation (4.99) produces for each particle the net interaction about it. Where the value of I_a^j is non-zero, there must be some error in one or more of the interactions involved. However, it is not possible to know how much error there is in each individual interaction. Fortunately the value of I_a^j not only indicates whether there is an error, but also quantifies that error. It must, therefore, be possible to reconstruct the algorithm in such a way as to reduce, or hopefully remove, the error in interaction.

To reiterate, I_{ab}^j is the effect b has on a, where the direction and magnitude of the interaction is given by $\text{sgn}\left(I_{ab}^j\right)$ and $\gamma_{ab}^j = \left|I_{ab}^j\right|$, respectively. The direction of

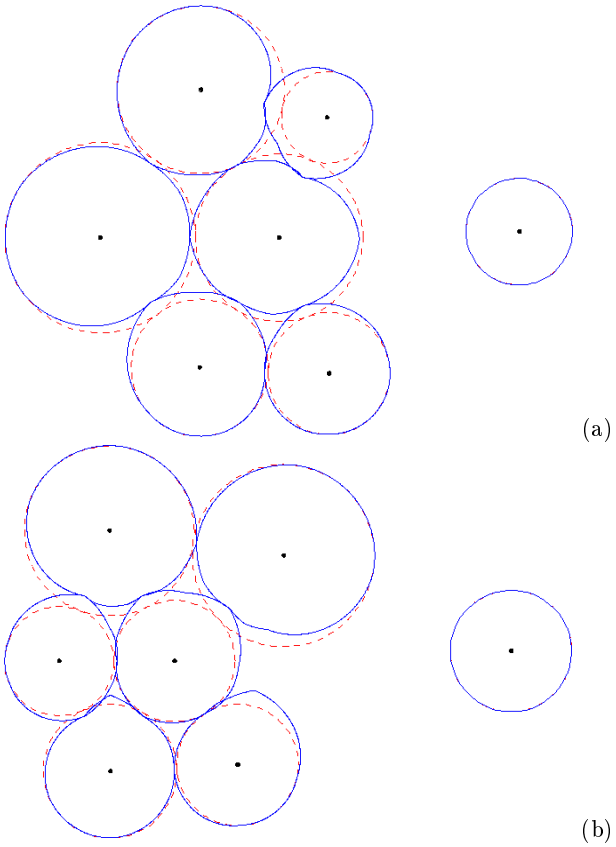

Figure 4.10: Examples of the two subtypes of impossible clusters. Where (a) and (b) are depictions of the large and small central pseudoparticle subtypes, respectively. Note that the seventh pseudoparticle in (a) cannot be made to fill the over-large gap around the similarly large central pseudoparticle. Conversely the seventh pseudoparticle in (b) cannot be made to fit the overly small gap around the similarly small central pseudoparticle. The black points represent the position of each pseudoparticle, r_a. The lines are depict the extents of equal interaction (such that touching lines infer equal embedding) around each pseudoparticle. For each pseudoparticle, the red, dashed line shows what the extent would be if the pseudoparticle was surrounded by identical pseudoparticles of the same type, while the blue, solid line shows the extent as it has been deformed by the actual pseudoparticles surrounding it.

the interaction is not a function of distance, and therefore the error must be in the magnitude of the interaction. Explicitly,

$$I_{ab}^{j} = \left[I_{ab}^{j}\right]_{\text{true}} F_{ab}^{j} \tag{4.106}$$

where F_{ab}^{j} is some destabilising factor. Thus, the obvious modification of algorithm would be

$$I_{ab}^{j} \quad \rightarrow \quad \frac{I_{ab}^{j}}{F_{ab}^{j}} \tag{4.107}$$

Thus, for the net interaction to reduce to zero,

$$I_{a}^{j} = \sum_{b \in \xi_a} I_{ab}^{j} = 0 \quad \rightarrow \quad \tilde{I_{a}^{j}} = \sum_{b \in \xi_a} \frac{I_{ab}^{j}}{F_{ab}^{j}} = 0 \tag{4.108}$$

However, equation (4.108) represents an underdetermined system, and F_{ab}^{j} cannot be found. In addition, there is no simple basis for assumptions as to the values of F_{ab}^{j}.

Consider, instead, the proposition that, I_{ab}^{j} is modified such that some approximation of the error is removed by subtraction, i.e.

$$I_{ab}^{j} \quad \rightarrow \quad I_{ab}^{j} - R_{ab}^{j} I_{a}^{j} \tag{4.109}$$

The change must modify the measure of net interaction, which is no longer equal to I_{a}^{j},

$$I_{a}^{j} = \sum_{b \in \xi_a} I_{ab}^{j} \quad \rightarrow \quad \tilde{I_{a}^{j}} = \sum_{b \in \xi_a} I_{ab}^{j} - R_{ab}^{j} I_{a}^{j} \tag{4.110}$$

which is still an underdetermined system, but with some rearrangement,

$$\tilde{I_{a}^{j}} = \sum_{b \in \xi_a} I_{ab}^{j} - I_{a}^{j} \sum_{b \in \xi_a} R_{ab}^{j} = I_{a}^{j} - I_{a}^{j} \sum_{b \in \xi_a} R_{ab}^{j} \tag{4.111}$$

Therefore,

$$\tilde{I_{a}^{j}} = 0 \quad \rightleftharpoons \quad \sum_{b \in \xi_a} R_{ab}^{j} = 1 \tag{4.112}$$

That is, the net interaction in this modified system must reduce to zero if the sum total of the corrective factors is one. R_{ab}^{j} must, therefore, be the fraction of the total error that can be apportioned to the interaction between the two particles a and b. This forms the basis for assumptions as to the definition of R_{ab}^{j}. There are several possibilities.

Assumption 1: Every b with a non-zero interaction with a in the j^{th} dimension contributes an equal proportion of the net interaction. Note that the number of pseudoparticles, b, is equal to one less than the degree of the j^{th} subset of the interacting pseudoparticles in the neighbourhood about a, $\xi_a^j \in \xi_a$.

$$R_{ab}^j = \frac{1}{\left|\xi_a^j\right| - 1} \tag{4.113}$$

where $\left|\xi_a^j\right|$ is the degree of the subset $\xi_a^j \in \xi_a$.

Assumption 2: The proportion of the net interaction of a that b is responsible for is equal to the ratio of the magnitude of the interaction between a and b over the total interaction at a. Explicitly, the total interaction is given by

$$\gamma_a^j = \sum_{b \in \xi_a} \gamma_{ab}^j = \sum_{b \in \xi_a} |I_{ab}^j| \tag{4.114}$$

and, therefore,

$$R_{ab}^j = \frac{\gamma_{ab}^j}{\gamma_a^j} \tag{4.115}$$

Assumption 3: The proportion of the net interaction of a that is due to b is equal to the ratio of the error in net interaction of b over the net of the net interactions of all the particles that interact with a. The net of net interactions about a is given by

$$\eta_a^j = \sum_{\substack{b \in \xi_a \\ b \neq a}} I_b^j = \sum_{\substack{b \in \xi_a \\ b \neq a}} \sum_{c \in \xi_b} I_{bc}^j \tag{4.116}$$

and

$$R_{ab}^j = \frac{I_b^j}{\eta_a^j} \tag{4.117}$$

Assumption 4: As a logical adaptation of the previous assumption, the proportion of the net interaction of a due to b is equal to the total interaction about b over the net total interaction at a,

$$\chi_a^j = \sum_{\substack{b \in \xi_a \\ b \neq a}} \gamma_b^j = \sum_{\substack{b \in \xi_a \\ b \neq a}} \sum_{c \in \xi_b} \gamma_{ab}^j \tag{4.118}$$

Thus,

$$R_{ab}^j = \frac{\gamma_b^j}{\chi_a^j} \tag{4.119}$$

Assumption N: Further propositions could be created, factoring in the net and total interactions of an increasing number of surrounding particles and/or increasing the complexity of the fraction with each new modification. However, the error is only being spread out further into the other particles, rather than keeping localised. Further modifications may also add considerable computations.

The first assumption is, very much, the simplest. However, it must also be the least accurate as it is improbable that the contributions to the net interaction for all particles can be equally weighted. It is also the simplest to implement. All assumptions require at an additional pass over the particle interactions, to calculate I_a^j and γ_a^j, but the other assumptions ($N > 2$) require at least one more loop through the particles in order to create η_a^j and χ_a^j. The modified algorithms behaviour is difficult to analyse.

Consider first affirmation of the effectiveness of each proposition to eliminate the numerical artefact under conditions of constant stress, regardless of the inaccuracy in pseudoparticle configuration. In reference to Figure 4.11, the four demonstrated configurations are

1. Regularly spaced with particle separations of Δ_p.
2. Regularly spaced, with a single exception at $\frac{1}{2}N_p$ where the particle is offset by some amount less than Δ_p.
3. Regularly spaced, excluding the a reduced gap between $a = \frac{1}{2}N_p$ and $b = \frac{1}{2}N_p + 1$.
4. Regularly spaced configuration modified by a sub-Δ_p random shift.

In addition the volume of each particle, $\frac{m}{\rho}$, is constant.

Note that, as expected, for all configurations there is no deviation from the true solution, $\frac{\partial S}{\partial r} = 0$.

Consider tests with more dynamic stress conditions for each of the configurations in turn. The stress is defined by these, progressively more complex, forms;

1. Linear Variation, $S = S_0 mx$.
2. Sinusoidal Variation, $S = S_0 \sin\left(\frac{2\pi x}{\max(x)}\right)$.
3. Variation defined by the quadratic polynomial, $S = S_0 x^2$.
4. Constant stress offset by some Gaussian variation, $S = S_0\left(1 + e^{\frac{-(x-x_0)^2}{0.05}}\right)$.

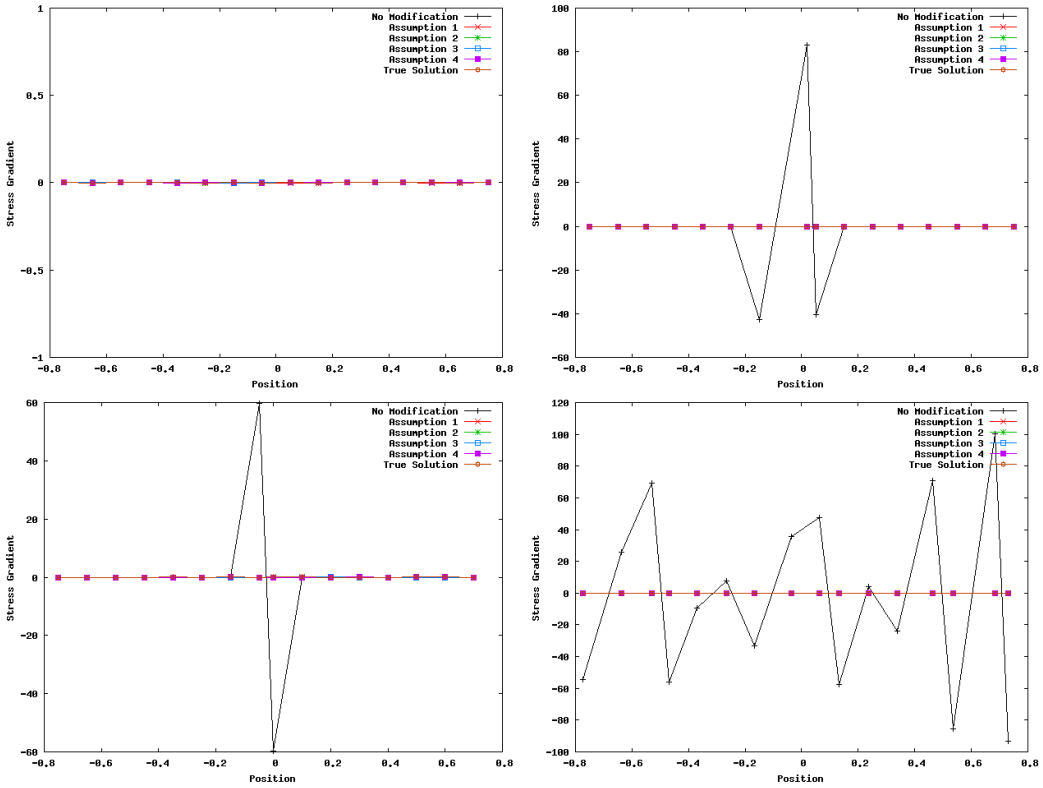

Figure 4.11: Plots of the stress gradients (as calculated by uncorrected SP approximation and with the various corrective methods) for each particle configuration 1 to 4. Each configuration begins as regularly spaced, $\delta_x = \Delta_p$, pseudoparticles (configuration 1) then the pseudoparticle $a = N_p/2$ is shifted by $0 < \delta_p < \Delta_p$ (configuration 2), or $|\mathbf{r}_{ab}| < \Delta_p$ where $a = N_p/2$ and $b = a+1$, i.e. the gap between a and b is reduced (configuration 3), or a random shift, $\pm\Delta_p R/2$ where R is a random number, is applied to each pseudoparticle (configuration 4). Using English directionality (as pertaining to the conventions of print, i.e. reading top to bottom, and left to right within that), each plot shows the stress gradients with configuration 1 to 4, respectively. Notice, in Black, that the original stress gradient is non-zero around the focus of misconfiguration (nowhere in the case of configuration 1 and everywhere in configuration 4) and that each assumption corrects the problem.

See Figure 4.12 for plots of these stresses for each configuration.

There are an infinite number of other misconfigurations, beyond the three presented here, so why discuss these specific forms? Configurations 2 and 3 are the simplest, where the misconfigurations are highly localised, and are asymmetric and symmetric, respectively. They allow investigation of a corrective methods ability to reduce the error, while not smearing out the error beyond the localised frame. In addition, a measure of the effectiveness of each to discern the source of an error in simple cases with and without asymmetry. The final configuration (4) is included to see how each proposition deals with unpredictable net interactions spread out over the domain.

Consider each configuration in turn. Firstly, the simplest has only one particle ($a = N_p/2$) out of place by some distance less than the particle spacing (configuration 2). As such, and as with all of the configurations, the neighbour lists remain the same. Figure 4.13 shows the calculated stress gradients, and the absolute residual once the true solution (as calculated by SP approximation with the correct particle configuration, at the same spatial resolution) was removed. Secondly, exchange the configuration for the set of pseudoparticles separated by $|\mathbf{r}_{ab}| = \Delta_p$, excluding the distance between $a = N_p/2$ and $b = a+1$ (configuration 3). See Figure 4.14 showing the gradient curves and residuals. Finally, redefining the particle configuration such that the original regular set of particles have some random, sub-Δ_p offsets applied individually to each (configuration 4). See Figure 4.15.

Before discussing the specific results, consider the more general observation that the residuals are in several cases larger than the true stress gradients. It is clear, therefore, that none of the assumed corrections are without fault, though the differences are pronounced. In asymmetric conditions, assumption 2 behaves most poorly, but in symmetric conditions assumption 2 performs as well or better than the other methods. It appears for assumption 3, particularly in Figure 4.15, that where the sign of the stress, $\operatorname{sgn}\left(S_a^j\right)$, and the net interaction, $\operatorname{sgn}\left(I_a^j\right)$, are equal then the $O\left(\tilde{I}_a^j\right) = O\left(I_a^j\right)$ and for the majority $\tilde{I}_a^j \approx I_a^j$. This rule is not universal. For the simpler symmetric configuration, Figure 4.14, the observation appears to hold in terms of the order of the two arguments $O\left(\tilde{I}_a^j\right) = O\left(I_a^j\right)$, however rather than the equal signs of stress and net interaction, the sign of all modified net interactions are equal. In the asymmetric configuration, Figure 4.13, the behaviour is similar. However, for most cases it displays the lowest residuals of the various assumptions. The only exception is the gradients generated from Gaussian stress. In this case the central particle ($a = N_p/2$), $O\left(\tilde{I}_a^j\right) = O\left(I_a^j\right)$. Of each corrective proposition, the behaviour of the fourth is the most stable. Throughout all tests $\tilde{I}_a^j < I_a^j$ and

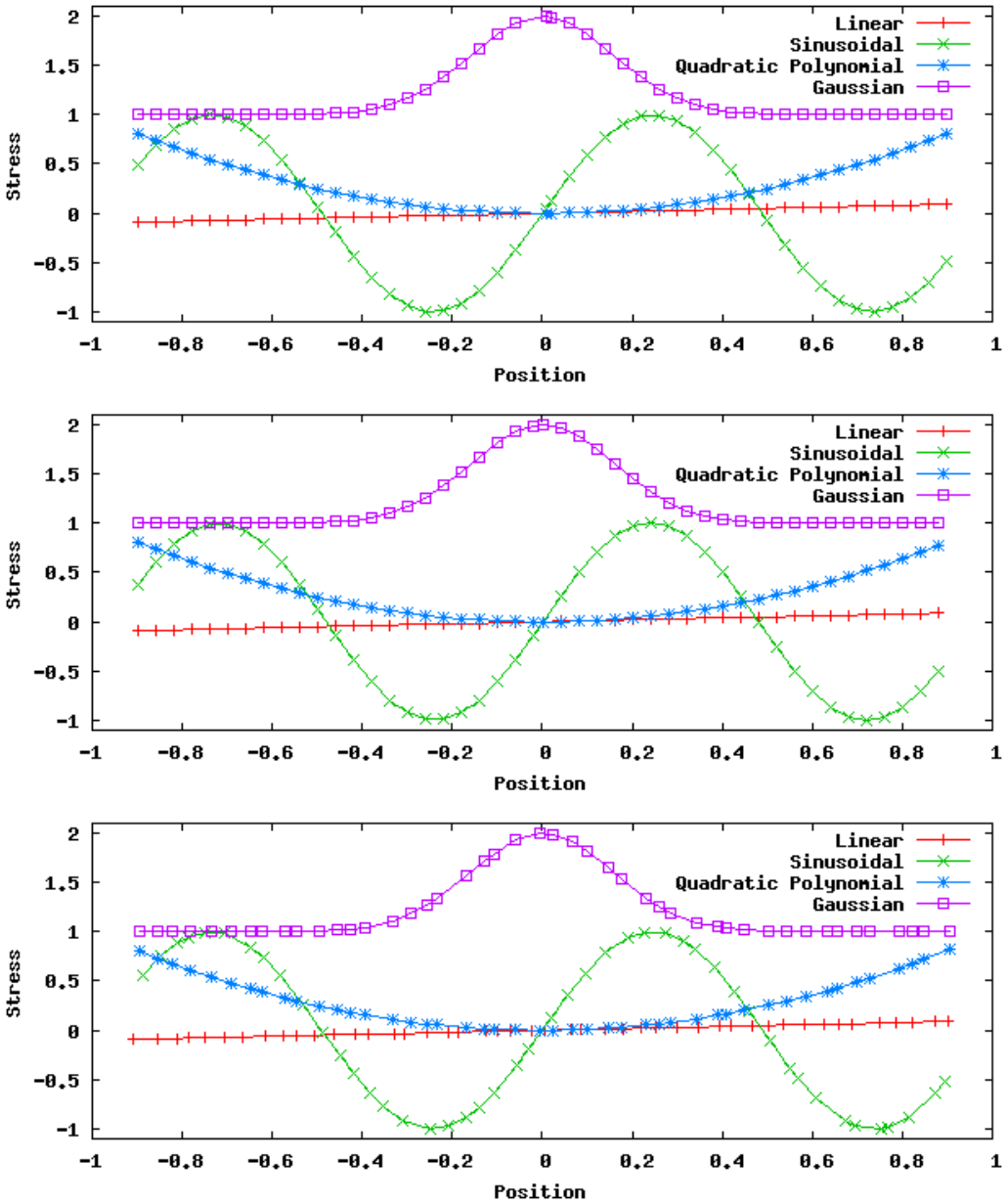

Figure 4.12: Plots showing each stress curve with each pseudoparticle misconfiguration ($N_p \approx 50$). Each configuration begins as regularly spaced, $\delta_x = \Delta_p$, pseudoparticles (configuration 1) then the pseudoparticle $a = N_p/2$ is shifted by $0 < \delta_p < \Delta_p$ (configuration 2), or $|\mathbf{r}_{ab}| < \Delta_p$ where $a = N_p/2$ and $b = a+1$, ie. the gap between a and b is reduced (configuration 3), or a random shift, $\pm\delta_p R/2$ where $0 < R < 1$ is a random number, is applied to each pseudoparticle (configuration 4). Top to bottom the plots show, respectively, the stress curves with configuration 2, 3, and 4.

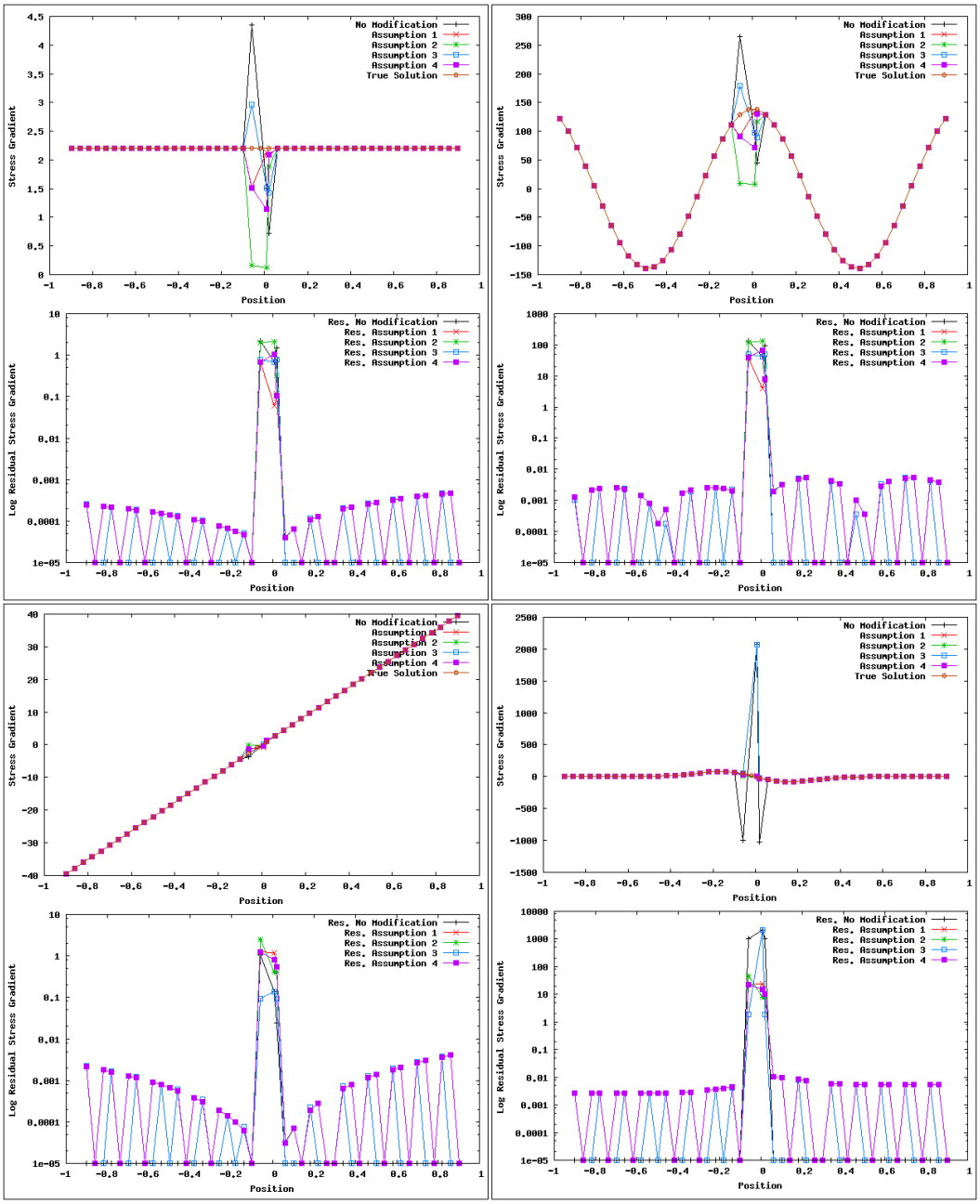

Figure 4.13: Plots showing the stress gradient curves and absolute residuals constructed by subtracting the the true solution, as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, boxed are pairs of stress gradients and residuals for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 2, where the pseudoparticles are regularly spaced, excluding $a = N_p/2$ which is offset by $0 < \delta_p < \Delta_p$. For larger plots see Figures B.1 and B.2.

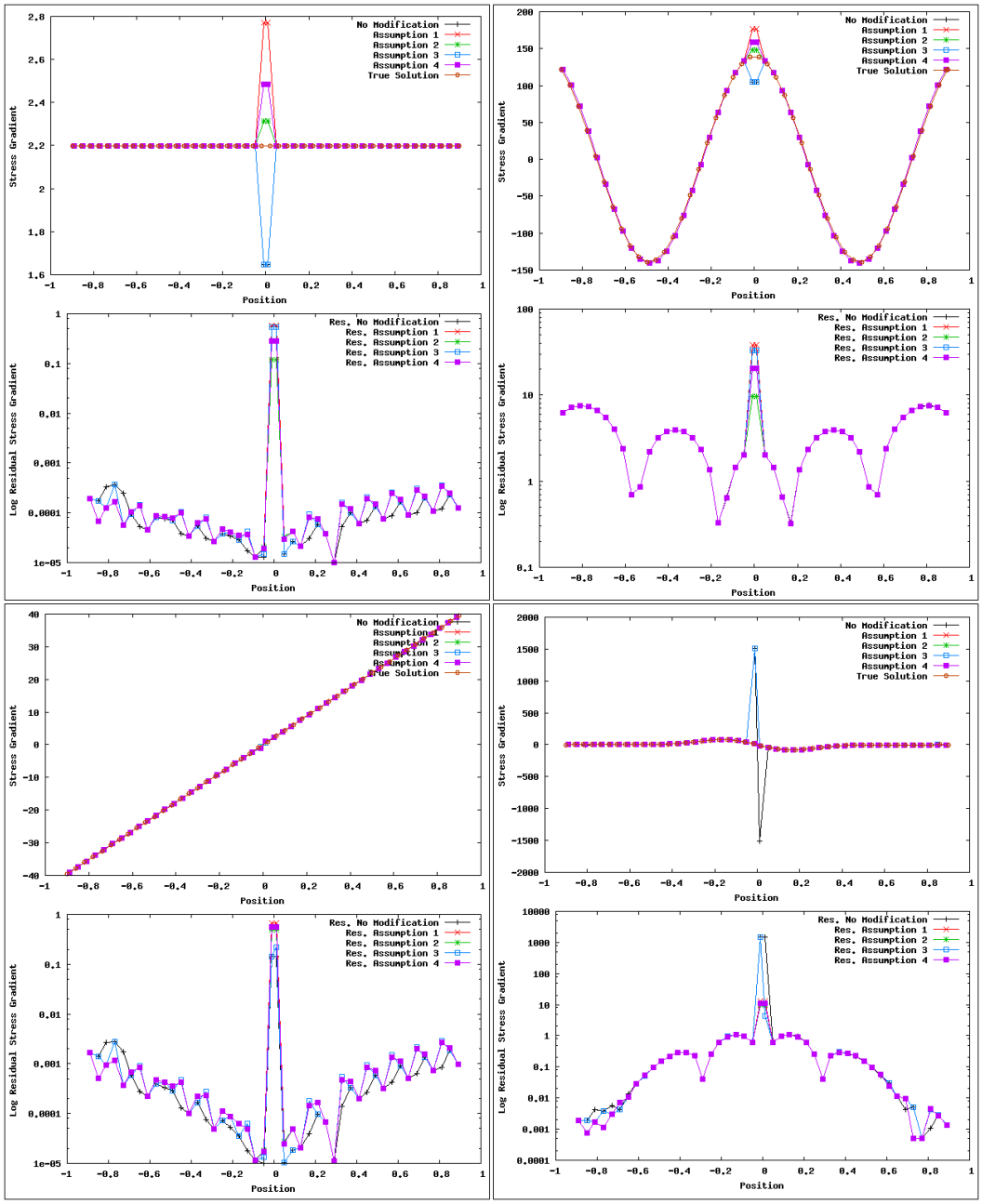

Figure 4.14: Plots showing the stress gradient curves and absolute residuals constructed by subtracting the the true solution, as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, boxed are pairs of stress gradients and residuals for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 3, where the pseudoparticles are regularly spaced, excluding the pair $a = N_p/2$ and $b = a + 1$ where are closer together ($|\mathbf{r}_{ab}| < \Delta_p$). For larger plots see Figures B.3 and B.4.

Figure 4.15: Plots showing the stress gradient curves and absolute residuals constructed by subtracting the the true solution, as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, boxed are pairs of stress gradients and residuals for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 4, where the pseudoparticles are regularly spaced, then offset by $\pm\delta_p R/2$ where $0 < R < 1$ is some random number. For larger plots see Figures B.5 and B.6.

normally $\tilde{I_a^j} << I_a^j$. Somewhat counter-intuitively the proposition that reduces the error most, number 1, is also the least complex and did not take into account any of the quantifiable measures associated with that error. The only exception appears in, seemingly, the simplest test of the algorithm – the linear stress, computed for the symmetric misconfiguration – whereby the net interaction displays little change, $\tilde{I_a^j} \approx I_a^j$.

Clearly, none of these approaches functions as intended in each test. This is of grave concern as, if the errors cannot be reduced, then the numerical artefact flow will create non-physical data to be generated by the algorithm.

4.5 Conclusion

The work in this chapter revealed a lack of consistency among the traditional interpretations of the smoothing function derivative, and restored that consistency by proposing the reflected formulation.

$$\overline{\nabla_{ab}^j} W_{ab} = \frac{1}{2}\left[\nabla_a^j W_{ab} - \nabla_b^j W_{ba}\right] \tag{4.120}$$

However, this accurate restoration came at the cost of computational efficiency.

Evidence and explanation of a numerical artefact was also presented, as well as unsuccessful attempts to correct for the error. The error could be reduced, or at the very least localised, but the cost to the physical dynamics is impossible to quantify. As such, it is clear that an investigation into how errors manifest in the SP model is required. This is the impetus for the research presented in the next chapter.

Chapter 5

Considering the SP Errors and Corrected SP Method

Contents

5.1 Introduction

The complications under negative stress (due to the tensile instability) discussed in Section 3.6.1 and the numerical artefact flow, as well as the difficulty in correcting for the artefact, discussed in the previous chapter (Section 4.4) show that the errors in the SPMHD algorithm can be complex and very difficult to understand. Consequently, this chapter is dedicated to the quantification and mitigation of errors

or error sources. Of particular concern, given its position as a fundamental part of the SP technique, is the undefined error due to the discretisation approximation presented in Chapter 2.

Subsequent to the successful quantification of the discretisation error in Section 5.2.1, a corrective procedure mitigating it is described (the Corrected SP method). Analysis of the technique reveal that it is highly accurate if somewhat computationally intensive. Once properly investigated, by returning to the derivation of the corrective procedure, the method is simplified to form the CSP-Δh (Corrected-SP-built-without-smoothing-length-gradient-terms) method while continuing to mitigate the error due to discretisation.

5.2 SP Errors

Assuming that an SP model has been constructed consistently, and that subsequent simulations are properly configured – ie. boundary and initial conditions are physical – then there are only four sources of error that contribute to the actual error in the data output.

- Error induced by numerical integration,
- Approximations in the model,
- Model instabilities, and
- Computational error (due to truncation, etc.).

These errors can become compounded over simulated time, leading to increasingly inaccurate results. Reducing these errors produces more accurate data, and allows the simulation to remain stable for a longer period of time. The first three sources of error have been touched on in earlier sections of this document (Sections 3.4.1, 2.2.2 and 3.6). The final error source, a form of systematic error, is unavoidable. Of the four sources, discussed here are the errors introduced by approximations made in the construction of the SP model. Specifically, the approximations made creating the SP model are the smoothing and discretisation approximations, Theorems 2.2 and 2.8, respectively.

The error due to the smoothing approximation, ε_{sm}, has already been discussed in Section 2.2.5 and was shown to be $O\left(h^2\right)$ (Theorem 2.3). However by implementing a higher-order smoothing function this estimate can be reduced (Monaghan, 1985). Monaghan (2005) gave a Gaussian example,

$$W\left(r,h\right)=\frac{1}{h\sqrt{\pi}}\left(\frac{3}{2}-q^2\right)e^{-q^2} \tag{5.1}$$

where $q = \frac{r}{h}$, which showed the smoothing approximation to be accurate to $O\left(h^4\right)$. He went on to note that the higher-order smoothing functions, by necessity, were not positive definite and therefore could produce, for instance, negative density values. Hence, to exclude this possibility the smoothing functions conceptually employed here are consistent with an error of $O\left(h^2\right)$.

The error due to the discretisation is more complex, and is the subject of investigation in the next section.

5.2.1 Discretisation Error

Defining the error induced by the discretisation approximation, $\varepsilon_{\mathrm{dc}}$, more involved than the smoothing approximation error, $\varepsilon_{\mathrm{sm}}$, because the pseudoparticles only exist at points within the subdomains ω_a, with no restrictions as to relative position or even the number of pseudoparticles, $|\xi_a|$. Any rigorous analysis is therefore challenging. The most accurate estimates are only applicable to quasi-static fluids, which are rarely modelled as they can commonly be solved analytically (Monaghan, 2005; Colagrossi, 2004b).

By splitting the discretisation of the smoothing approximation (the summation approximation) and the smoothing approximation of derivatives (SP derivatives), some limited analysis is possible. For instance, Monaghan (2005) presented a derivation of the error based on; a set of pseudoparticles equally spaced on an infinite line (in a 1D domain);a Gaussian smoothing function; $\frac{m}{\rho} = \Delta_p$, and; reconstructing $g(x) = \alpha + \beta x$ by the summation approximation and g' with SP gradient approximation. In both reconstruction cases, $\varepsilon_{\mathrm{dc}}$ was shown to be *"exponentially small and negligible if* $h > \Delta_p$*"*. However, these conditions are highly restrictive and not representative of a real simulation.

Due to the parallels that can be drawn between SP and Monte-Carlo methods, the error was initially approximated as $\sim N_p^{-\frac{1}{2}}$ by Gingold and Monaghan (1977). However, the true error was shown to be much less than this. Monaghan (2005) attributed this to the Monte-Carlo estimates ignorance of the dynamics of the system. Consider some distribution of identical pseudoparticles with errors in their positions given by $0 < \delta\mathbf{r} < \Delta_p$, where Δ_p is the pseudoparticle spacing if $\delta\mathbf{r} = 0$. The resulting pressure imbalance causes the pseudoparticles to accelerate, acting to smooth out the pressure gradient. Thus, at the next iteration, the error $|\delta\mathbf{r}|^{<n+1>} \leq |\delta\mathbf{r}|^{<n>}$ and over a number of such iterations, the error reduces dramatically. Monaghan (2005) exerted that *"...the SPH* [pseudo]*particles are* [therefore] *disordered, but in an orderly way"*.

Given this perspective it is possible deduce that, in order to quantify the discreti-

sation error, the errors knowledge of the system must be accounted for. Explicitly defining some all-encompassing, general formula for the discretisation error is therefore not possible, as it would have to ignore the relationship to the specific conditions under approximation. Instead, it must be possible to express the error as a function of the approximated system. Here the problem is redefined such that the approximations for which a definition of the error is required are all approximating derivatives. This is consistent with (and therefore most useful to) the SPMHD equations that evolve the properties via definitions of the rates of change, rather than using the summation approximation. The unknown error can, therefore, be expressed as $\varepsilon_{\mathrm{dc},X}$ where X is some property the derivative of which undergoing approximation.

The following derivation produces an expression defining the discretisation error for vector derivatives.

Theorem 5.1 *Consider first that an approximation,* $[X]_{\mathrm{approx}}$*, is the sum of the true value and any errors, i.e.*

$$[X]_{\mathrm{approx}} = X + \varepsilon_{\mathrm{approx}} = X + \sum_q \varepsilon_q \tag{5.2}$$

where $\varepsilon_{\mathrm{approx}}$ *is the total error, and* ε_q *is the error induced by the* q^{th} *source. In this way, for the SP outer product approximation* $[\nabla \otimes \mathbf{A}_a]_{\mathrm{SP}}$ *where* $\mathbf{A}$ *is some vector,*

$$\left[\nabla^j A_a^i\right]_{\mathrm{SP}} = \nabla^j A_a^i + \varepsilon_{\mathrm{SP}} = \nabla^j A_a^i + \varepsilon_{\mathrm{dc},\mathbf{A}}^{ji} + \varepsilon_{\mathrm{sm}} \tag{5.3}$$

using indice notation. Also consider the definition of the SP outer product identity (2.67), such that by substitution,

$$\left[\nabla^j A_a^i\right]_{\mathrm{SP}} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b}\left(A_b^i - A_a^i\right) \otimes \nabla_a^j W_{ab} \tag{5.4}$$

Now, a Taylor series expansion of A_b^i *about* $\mathbf{r}_b = \mathbf{r}_a$ *yields*

$$\left[\nabla^j A_a^i\right]_{\mathrm{SP}} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b}\left(\left[A_a^i + r_{ba}^k \nabla^k A_a^i + \cdots\right]_{\mathbf{r}_b=\mathbf{r}_a} - A_a^i\right)\nabla_a^j W_{ab} \tag{5.5}$$

Theorem 5.1 (continued) *The zeroth order terms of equation (5.5) clearly cancel and, if the higher order terms are collected, it collapses to give*

$$\left[\nabla^j A_a^i\right]_{\mathrm{SP}} = \nabla^k A_a^i \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} r_{ba}^k \nabla_a^j W_{ab} + O(h^2) \tag{5.6}$$

where the pseudoparticle summation is the SP approximation of $\nabla \otimes \mathbf{r}_a$*. Thus, recalling the definition of the the error due to the smoothing approximation, equation (2.31), it is possible to define*

$$\left[\nabla^j A_a^i\right]_{\mathrm{SP}} = \left[\nabla^j r_a^k\right]_{\mathrm{SP}} \nabla^k A_a^i + O(h^2) = \left[\nabla^j r_a^k\right]_{\mathrm{SP}} \nabla^k A_a^i + \varepsilon_{\mathrm{sm}} \tag{5.7}$$

Finally, by setting equation (5.7) equal to equation (5.3) and subsequent simplification it is possible to show that

$$\varepsilon_{\mathrm{dc},\mathbf{A}}^{ji} = \left(\left[\nabla^j r_a^k\right]_{\mathrm{SP}} - \delta^{jk}\right) \nabla^k A_a^i \tag{5.8}$$

is the **Vector Derivative Discretisation Error**.

An analysis of the other vector derivatives show that the discretisation error in each case is composed of elements of the $\varepsilon_{\mathrm{dc},\mathbf{A}}$ as defined. Explicitly, for the SP divergence approximation,

$$[\nabla \cdot \mathbf{A}_a]_{\mathrm{SP}} = \nabla \cdot \mathbf{A}_a + \varepsilon_{\mathrm{dc},\mathbf{A}}^{jj} + \varepsilon_{\mathrm{sm}} \tag{5.9}$$

where $\varepsilon_{\mathrm{dc},\mathbf{A}}^{jj}$ is the discretisation error. Or, for the k^{th} element of the SP curl approximation,

$$[\nabla \times \mathbf{A}_a]_{\mathrm{SP}}^k = (\nabla \times \mathbf{A}_a)^k + \epsilon^{kji}\varepsilon_{\mathrm{dc},\mathbf{A}}^{ji} + \varepsilon_{\mathrm{sm}} \tag{5.10}$$

where $\epsilon^{kji}\varepsilon_{\mathrm{dc},\mathbf{A}}^{ji}$ is the discretisation error, and ϵ^{kji} is the Levi-Civita symbol (2.70). Note also that by defining $\mathbf{A} = \mathbf{T}^i$, where $\mathbf{T}^i$ is the i^{th} row/column of some tensor $\mathbf{T}$, it is possible to express the discretisation error for simple derivatives of tensors. For example, the error due to discretisation in $\left[\frac{\partial T_a^{ij}}{\partial r^j}\right]_{\mathrm{SP}}$ is

$$\varepsilon_{\mathrm{dc},T}^{ji} = \left(\nabla^j r^k - \delta^{jk}\right) \frac{\partial T_a^{ik}}{\partial r^k} \tag{5.11}$$

In a similar manner to Theorem 5.1, it is also possible to define the discretisation error for the SP gradient approximation.

Theorem 5.2 *Consider the definition of the SP gradient identity (2.63),*

$$\left[\nabla^j \lambda_a\right]_{\mathrm{SP}} = \frac{1}{\Psi_a} \sum_{\substack{b \in \xi_a^j \\ b \neq a}} \frac{m_b \Psi_b}{\rho_b} \left(\lambda_b - \lambda_a\right) \nabla_a^j W_{ab} \tag{5.12}$$

such that another Taylor series expansion (of λ_b about $\mathbf{r}_b = \mathbf{r}_a$) yields

$$\left[\nabla^j \lambda_a\right]_{\mathrm{SP}} = \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left(\left[\lambda_a + r_{ba}^k \nabla^k \lambda_a + \cdots \right]_{\mathbf{r}_b = \mathbf{r}_a} - \lambda_a \right) \nabla_a^j W_{ab} \tag{5.13}$$

Cancelling the zeroth order terms and collecting the higher order terms once again, gives

$$\left[\nabla^j \lambda_a\right]_{\mathrm{SP}} = \nabla^k \lambda_a \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} r_{ba}^k \nabla_a^j W_{ab} + O(h^2) \tag{5.14}$$

where the pseudoparticle summation is the SP approximation of $\nabla \otimes \mathbf{r}_a$. Thus, recalling the definition of the the error due to the smoothing approximation (2.31) and equation (5.2), it is possible to define

$$\left[\nabla^j \lambda_a\right]_{\mathrm{SP}} = \left[\nabla^j r_a^k\right]_{\mathrm{SP}} \nabla^k \lambda_a + \varepsilon_{\mathrm{sm}} = \nabla^j \lambda_a + \varepsilon_{\mathrm{dc},\lambda} + \varepsilon_{\mathrm{sm}} \tag{5.15}$$

Hence,

$$\varepsilon_{\mathrm{dc},\lambda} = \left(\left[\nabla^j r_a^k\right]_{\mathrm{SP}} - \delta^{jk} \right) \nabla^k \lambda_a \tag{5.16}$$

is the **Gradient Descretisation Error**.

5.2.2 Consequence of the Discretisation Error

By writing the discretisation errors in vector form, and collecting them together, it is possible to show that

$$\begin{bmatrix} \varepsilon_{\mathrm{dc},\lambda} \\ \varepsilon_{\mathrm{dc},\mathbf{A}} \\ \varepsilon_{\mathrm{dc},\mathbf{T}} \end{bmatrix} = \left(\left[\nabla \otimes \mathbf{r}_a\right]_{\mathrm{SP}} - \mathbf{I} \right) \begin{bmatrix} \nabla \lambda \\ \nabla \otimes \mathbf{A} \\ \nabla \otimes \mathbf{T} \end{bmatrix} \tag{5.17}$$

Given this expression, it is simpler to interpret the $\varepsilon_{\mathrm{dc}}$ more generally. Clearly two factors contribute to the error; the difference $\left[\nabla \otimes \mathbf{r}_a\right]_{\mathrm{SP}} - \mathbf{I}$ and the true gradient under approximation. In order to understand the former, first note that for the true

value $\nabla \otimes \mathbf{r}_a \equiv \mathbf{I}$, and therefore,

$$[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}} - \mathbf{I} = [\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}} - \nabla \otimes \mathbf{r}_a \tag{5.18}$$

Thus the $\varepsilon_{\mathrm{dc}}$ must be proportional to the systems inability to represent the isotropy of space. The latter leads to the simple conclusion that

$$\begin{bmatrix} \varepsilon_{\mathrm{dc},\lambda} \\ \varepsilon_{\mathrm{dc},\mathbf{A}} \\ \varepsilon_{\mathrm{dc},\mathbf{T}} \end{bmatrix} \propto \begin{bmatrix} \nabla\lambda \\ \nabla \otimes \mathbf{A} \\ \nabla \otimes \mathbf{T} \end{bmatrix} \tag{5.19}$$

Hence, the error must degrade high energy simulations (with large gradients) more so than simpler, low energy models.

In order to better understand the error, consider that the approximation $[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}}$ can be expressed as its true value (the identity matrix, I) modified by a scalar $k > 0$ and matrix D, which respectively express some numerical scaling factor and some measure of the distortion from the ideal pseudoparticle arrangement. That is

$$[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}} = kI + D \approx kI \pm \delta J \tag{5.20}$$

where D has been expanded $\pm\delta J$ (where J is the unit matrix) by assuming that the distortion uniformly induces values in D with a maximum magnetude of $|\delta|$, termed here the distortion coefficient. It is important to note that the two scalars (k and δ) are not mutually exclusive. This may be discussed in terms of information sampling. If too large/small a weighting is placed on the sampling from each neighbouring pseudoparticle both the scaling factor and distortion coefficient would be amplified. Additionally, if the subdomain about $\mathbf{r}_a$ were over-sampled in one direction relative to the sampling in the opposite direction, an increase in the distortion coefficient and a decrease, of unequal magnitude, in the scaling factor would be evident. Thus, the error may be approximated by

$$\begin{bmatrix} \varepsilon_{\mathrm{dc},\lambda} \\ \varepsilon_{\mathrm{dc},\mathbf{A}} \\ \varepsilon_{\mathrm{dc},\mathbf{T}} \end{bmatrix} \approx [(k-1)\mathbf{I} \pm \delta\mathbf{J}] \begin{bmatrix} \nabla\lambda \\ \nabla \otimes \mathbf{A} \\ \nabla \otimes \mathbf{T} \end{bmatrix} \tag{5.21}$$

Now, assuming some static pseudoparticle number density such that the scaling factor remains roughly constant, to keep the error constant as the true gradient increases, the pseudoparticles must be come increasingly ordered. Since this is impossible to enforce, the number density of the pseudoparticles must be increased. The error due to discretisation must, therefore, be responsible for the disproportion-

ately high resolution requirements for dynamic simulations relative to grid-based numerical methods (Hubber et al., 2011).

In the context of a Solar simulation, the discretisation is particularly problematic. Consider that the degree of material transport in the Solar Corona is very high, leading to potentially highly disordered pseudoparticle arrangements. Isolated, this would be distressing enough, however consider that this occurs in combination with significant gradients sufficient to bind the plasma into complex magnetic structures that store vast amounts of energy. In order to keep the number of pseudoparticles low, and therefore increase the pace and efficiency (in terms of physical memory) of the simulations, some measure of correction is required.

5.3 Corrected SP Method

The definition of the discretisation error facilitates the creation of a corrective matrix that can improve the accuracy of the SP method.

5.3.1 General Definition and Error Formulation

Rearrangement of equation (5.15) yields

$$\nabla \otimes \mathbf{A}_a = [\nabla \otimes \mathbf{r}_a]^{-1}_{\mathrm{SP}} \left([\nabla \otimes \mathbf{A}_a]_{\mathrm{SP}} - \varepsilon_{\mathrm{sm}} \right) \tag{5.22}$$

By dropping the smoothing error term, which must be dropped as it is not known explicitly, its possible to define the Corrected SP (CSP) outer product approximation,

$$[\nabla \otimes \mathbf{A}_a]_{\mathrm{CSP}} = [\nabla \otimes \mathbf{r}_a]^{-1}_{\mathrm{SP}} \, [\nabla \otimes \mathbf{A}_a]_{\mathrm{SP}} \tag{5.23}$$

and, therefore

$$[\nabla \otimes \mathbf{A}_a]_{\mathrm{CSP}} = \nabla \otimes \mathbf{A} + \varepsilon_{\mathrm{CSP}} = \nabla \otimes \mathbf{A}_a + [\nabla \otimes \mathbf{r}_a]^{-1}_{\mathrm{SP}} \, \varepsilon_{\mathrm{sm}} \tag{5.24}$$

In a similar manner, corrective procedures can be formulated for each SP derivative approximation, such that

$$\begin{bmatrix} [\nabla \lambda_a]_{\mathrm{CSP}} \\ [\nabla \otimes \mathbf{A}]_{\mathrm{CSP}} \\ [\nabla \otimes \mathbf{T}]_{\mathrm{CSP}} \end{bmatrix} = [\nabla \otimes \mathbf{r}_a]^{-1}_{\mathrm{SP}} \begin{bmatrix} [\nabla \lambda_a]_{\mathrm{SP}} \\ [\nabla \otimes \mathbf{A}]_{\mathrm{SP}} \\ [\nabla \otimes \mathbf{T}]_{\mathrm{SP}} \end{bmatrix} \tag{5.25}$$

and, constructed from these,

$$[\nabla \cdot \mathbf{A}_a]_{\text{CSP}} = [\nabla \otimes \mathbf{A}_a]^{jj}_{\text{CSP}} \tag{5.26}$$

and

$$[\nabla \times \mathbf{A}_a]^{k}_{\text{CSP}} = \epsilon^{kji} [\nabla \otimes \mathbf{A}_a]^{ji}_{\text{CSP}} \tag{5.27}$$

each with an equal error measure, $\varepsilon_{\text{CSP}} = [\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}} \varepsilon_{\text{sm}}$.

Note that these approximations are similar to those that define Normalised Corrected Smoothed Particle Hydrodynamics (NCSPH) as formulated by Vignjevic (2004). Note, however, the manner by which information is passed around the system of equations. The anisotropy of space causes the correct information to be shifted into the incorrect elements of the matrix $[\nabla \otimes \mathbf{A}_a]_{\text{SP}}$, such that in order to recover the correct distribution of that information (i.e. correct for ε_{dc}) the whole matrix must be constructed, even if all that is required by some algorithm is the divergence of $\mathbf{A}_a$. Therefore, the CSP method is not a modification of the smoothing function such that

$$\nabla_a W_{ab} \;\rightarrow\; \tilde{\nabla}_a W_{ab} = [\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}} \nabla_a W_{ab} \tag{5.28}$$

as is the case for NCSPH. Note that, for NCSPH, the smoothing function has already been normalised in discrete space such that

$$W_{ab} = \frac{W(|\mathbf{r}_{ab}|, h)}{\sum_{b \in \xi_a} \frac{m_b}{\rho_b} W(|\mathbf{r}_{ab}|, h)} \tag{5.29}$$

This is the fundamental difference between the CSP method and NCSPH, making CSP superficially more complex to implement, but leading to a more accurate approximation (see next).

5.3.2 Interpretation of the CSP Error

In order to be be useful, the CSP approximation i only required to be more accurate than SP approximation, ie. $\varepsilon_{\text{CSP}} = [\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}} \varepsilon_{\text{sm}} < \varepsilon_{\text{dc},X} + \varepsilon_{\text{sm}}$. However, the aim must be to drive the error as low as possible, the limit of which is the system noise ε_{sm}, therefore

$$\left\| [\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}} \varepsilon_{\text{sm}} \right\| \leq \|\varepsilon_{\text{sm}}\| \tag{5.30}$$

Given that $\|Ax\| \leq \|A\| \, \|x\|$, equation (5.30) must be true if

$$\left\| [\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}} \right\| \leq 1 \tag{5.31}$$

Recall the approximation of $[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}}$, equation (5.20), whereby the matrix is expressed as the sum of the identity matrix multiplied by some scaling factor, k, added to the unit matrix multiplied by a distortion coefficient somewhere in the range of $\pm\delta$. After significant rearrangement, this expression can be rewritten as

$$Sk \geq \frac{1+\nu|\delta|}{1-O\left(\frac{\delta^2}{k^2}\right)} \tag{5.32}$$

giving limits on the scaling factor k relative to the distortion coefficient, δ. In order to arrive at this expression, $|\delta|k^{-1}$ must be a sufficiently small value. One of the benefits of the SP method, is the deformability of the system. It follows, therefore, that $|\delta| << 1$ is unlikely to be true.

However, consider that $[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}}$ can be written in terms of the general form (2.25) as

$$[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}} = -\sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} r^k_{ab} \frac{r^j_{ab}}{|\mathbf{r}_{ab}|} \frac{\sigma_\nu}{h^{\nu+1}} \frac{\mathrm{d}F}{\mathrm{d}q} = \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} r^k_{ab} \frac{r^j_{ab}}{|\mathbf{r}_{ab}|} \frac{\sigma_\nu}{h^{\nu+1}} \left|\frac{\mathrm{d}F}{\mathrm{d}q}\right| \tag{5.33}$$

where $m_b; \rho_b; \sigma_\nu; |\mathbf{r}_{ab}| \geq 0$ for a physical medium and

$$\mathrm{sgn}\left(r^k_{ab} r^j_{ab}\right) = \begin{cases} \pm 1 & \text{if} \quad k \neq j \\ 1 & \text{if} \quad k = j \end{cases} \tag{5.34}$$

Clearly, therefore, the main diagonal elements are sums of positive terms and the off-diagonal elements are sums of both positive and negative terms. Assuming there are a significant number of neighbours, $|\xi_a|$, the number of positive and negative terms should roughly balance, such that $k > \delta$, and therefore equation (5.32) is a valid upper limit on the minimum value of k. Consequently, equation (5.30) holds. That is, the error in the CSP approximation is approximately equal to the system noise.

5.3.3 Creating CSPMHD

The creation of the CSPMHD algorithm, as with all modifications of the SP method, begins with the premise that the SP algorithm as previously defined requires only modification of those elements that differ between the SP method and, in this case, CSP method. All the equations remain the same, it is merely the approximation of spatial gradients that change.

5.3.3.1 Modification of the SP Equations

Beginning with the simplest equations, therefore, the continuity equation (2.99) and induction equation (2.101) become

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = -\rho_a \left[\nabla \cdot \mathbf{v}_a\right]_{\mathrm{CSP}} \tag{5.35}$$

and

$$\frac{\mathrm{d}\mathbf{B}_a}{\mathrm{d}t} = \mathbf{B}_a \left[\nabla \cdot \mathbf{v}_a\right]_{\mathrm{CSP}} - \left[\left(\mathbf{B}_a \cdot \nabla\right)\mathbf{v}_a\right]_{\mathrm{CSP}} \tag{5.36}$$

respectively. Expanding these approximations, in index notation, the equations are

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = -\rho_a \left[\nabla \otimes \mathbf{v}_a\right]^{jj}_{\mathrm{CSP}} \tag{5.37}$$

and

$$\frac{\mathrm{d}B^i_a}{\mathrm{d}t} = B^i_a \left[\nabla \otimes \mathbf{v}_a\right]^{jj}_{\mathrm{CSP}} - B^j_a \left[\nabla \otimes \mathbf{v}_a\right]^{ji}_{\mathrm{CSP}} \tag{5.38}$$

where

$$\left[\nabla \otimes \mathbf{v}_a\right]^{ji}_{\mathrm{CSP}} \equiv \left[\nabla^j v^i_a\right]_{\mathrm{CSP}} = \left(\left[\nabla \otimes \mathbf{r}_a\right]^{-1}_{\mathrm{SP}}\right)^{jk} \left[\nabla^k v^i_a\right]_{\mathrm{SP}} \tag{5.39}$$

and, as per the SP identities,

$$\left[\nabla^j v^i_a\right]_{\mathrm{SP}} = \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho} v^i_{ba} \nabla^j_a W_{ab} \tag{5.40}$$

and

$$\left[\nabla^j r^k_a\right]_{\mathrm{SP}} = \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} r^k_{ba} \nabla^j_a W_{ab} \tag{5.41}$$

Note that, though the complexity has increase somewhat, the number and nature of the approximation calculations remains constant. That is, there are two SP approximation calculations required by both SP and CSP methods, and it is only how those approximations are applied (and the variables undergoing approximation) that change.

If the SP equation of motion were derived using the SP identities directly, as with the SP continuity equation (2.112) and SP induction equation (2.118), the process of substitution would be similarly simple. However, the momentum and energy conserving behaviour of the model should be preserved, and therefore the CSP should be applied to the true SP equation of motion (2.140). This requires further manipulation as the correction matrix defining the CSP method was derived using the SP identities which rely on the difference, and not the sum, of the property

(λ, $\mathbf{A}$, $\mathbf{T}$, etc.) at a and b. Consider that the SP equation of motion, built assuming $\Psi = 1$, can be written such that

$$\frac{\mathrm{d}v_a^i}{\mathrm{d}t} = \frac{1}{\rho_a}\left[\nabla^j S_a^{ij}\right]_{\substack{\mathrm{SP}\\ \mathrm{consv}}} \tag{5.42}$$

where

$$\left[\nabla^j S_a^{ij}\right]_{\substack{\mathrm{SP}\\ \mathrm{consv}}} = \sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\left(S_a^{ij} + S_b^{ij}\right)\overline{\nabla_{ab}^j} W_{ab} \tag{5.43}$$

represents the approximation of $\nabla^j S_a^{ij}$ that ensures conservation, rather than the SP approximation $\left[\nabla^j S_a^{ij}\right]_{\mathrm{SP}}$. This is the element of the equation of motion that must be altered to conform to the CSP method.

Consider that equation (5.43) can be expanded to give

$$\left[\nabla^j S_a^{ij}\right]_{\substack{\mathrm{SP}\\ \mathrm{consv}}} = \sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\left(S_b^{ij} - S_a^{ij}\right)\overline{\nabla_{ab}^j} W_{ab} + 2S_a^{ij}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\overline{\nabla_{ab}^j} W_{ab} \tag{5.44}$$

The first summation is the gradient of the stress as determined by direct application of the SP identities $\left(\left[\nabla^j S_a^{ij}\right]_{\mathrm{SP}}\right)$, and therefore the second summation must represent the numerical procedure ensuring conservation of momentum and the implicit re-meshing scheme discussed by Price (2012). Now, rewrite equation (5.45) such that

$$\left[\nabla^j S_a^{ij}\right]_{\substack{\mathrm{SP}\\ \mathrm{consv}}} = \left[\nabla^j S_a^{ij}\right]_{\mathrm{SP}} + 2S_a^{ij}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\phi_{ba}\overline{\nabla_{ab}^j} W_{ab} = \left[\nabla^j S_a^{ij}\right]_{\mathrm{SP}} + 2S_a^{ij}\left[\nabla^j \phi_a\right]_{\mathrm{SP}} \tag{5.45}$$

where ϕ_a is some purely numerical property that exhibits the following, non-physical behaviour,

$$(\phi_b - \phi_a)_{b\to a} \equiv (\phi_a - \phi_b)_{a\to b} \equiv 1 \tag{5.46}$$

for all pseudoparticle pairs a and b. To be clear, the difference between ϕ_a and ϕ_b is always 1, and the $\mathrm{sgn}\,(\phi_{ba})$ is always positive, no matter whether calculating the effect of pseudoparticle b on a ($b \to a$), or the effect of a on b ($a \to b$). Expanded in this way, it is clear that the transformation to a conservative CSP approximation requires that

$$\left[\nabla^j S_a^{ij}\right]_{\substack{\mathrm{CSP}\\ \mathrm{consv}}} = \left[\nabla^j S_a^{ij}\right]_{\mathrm{CSP}} + 2S_a^{ij}\left[\nabla^j \phi_a\right]_{\mathrm{CSP}} \tag{5.47}$$

and by substitution and rearrangement,

$$\begin{aligned}\left[\nabla^j S_a^{ij}\right]_{\substack{\text{CSP}\\\text{consv}}} &= \left([\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}}\right)^{jk} \left[\nabla^k S_a^{ik}\right]_{\text{SP}} + 2S_a^{ij} \left([\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}}\right)^{jk} \left[\nabla^k \phi_a\right]_{\text{SP}} \\ &= \left([\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}}\right)^{jk} \left(\left[\nabla^k S_a^{ik}\right]_{\text{SP}} + 2S_a^{ij} \left[\nabla^k \phi_a\right]_{\text{SP}}\right) \\ &= \left([\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}}\right)^{jk} \left[\nabla^k S_a^{ik}\right]_{\substack{\text{SP}\\\text{consv}}}\end{aligned} \tag{5.48}$$

Hence, the correction matrix may be applied ahead of all of the spatial derivative components and the CSP equation of motion is given by

$$\left[\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right]_{\text{CSP}} = \frac{1}{\rho_a} \left[\nabla \otimes \mathbf{S}_a\right]_{\substack{\text{CSP}\\\text{consv}}} \tag{5.49}$$

where the spatial derivative is given by the above. This, in combination with the CSP continuity equation (5.37) and induction equation (5.38), provides the final changes to the basic SPMHD defining the CSPMHD algorithm. Note that there remain an equal number of approximations, the outer product of position, velocity and stress.

It is also worth noting that the justification of the application of the corrective matrix only holds for the SP equation of motion defined by $\Psi = 1$. For other definitions, the applicability of the corrective procedure is debatable. Certainly, the expression could not be expanded in the manner described above.

5.3.3.2 Effect on Numerical Artifact Flow

One point should be stated clearly such there can be no misunderstanding. The CSP method only corrects for the discretisation error, it does not remove any errors or instabilities other than those stemming directly from the inaccuracy in the spatial gradients due to discretisation approximation. Other numerical issues, such as the tensile instability and the inability to capture shocks or, necessarily, constrain $\nabla \cdot \mathbf{B}$ to zero must remain. Similarly, however, the modification procedures (if not the expressions themselves) that correct for these problems also remain.

So, what of the numerical artefact flow, as discussed in Section 4.4? The two are certainly connected. Both are functions of the anisotropy that the SP method perceives in the discretised space. However, the correction procedure cannot eliminate the numerical artefact flow. This is simply argued. Given the previous definition (Section 4.4.1) of the numerical artefact flow, it will only vanish if $[\nabla \lambda_a]_{\text{CSP}} = 0$ when λ_a is constant. Let

$$P = [\nabla \otimes \mathbf{r}_a]_{\text{SP}} \tag{5.50}$$

and

$$q = [\nabla \cdot \lambda_a]_{\text{SP}} \tag{5.51}$$

such that the corrective method may be expressed as $P^{-1}q$. It is clear to see that because $P \neq 0$ (and therefore invertible), if the numerical artefact is evident in q, and hence non-zero,

$$P^{-1}q \neq 0 \tag{5.52}$$

Thus the corrective procedure cannot eliminate the numerical artefact flow. It can however reduce the error, as numerical artefact flow is a clear function of the spatial derivative approximations, which are made more exact by the CSP method.

5.3.3.3 Numerical Modifications in CSPMHD

Of the two numerical modifications to the original SPMHD algorithm. the avoidance/removal of the tensile instability (Section 3.6.1) and the dissipation formulae (Section 3.6.2), the former are easier to adapt for CSPMHD. Taking each of the methods in turn,

1. Artificially stressing the system to avoid the negative stress that induces the tensile instability is just as simple in CSPMHD as SPMHD. Specifically,

$$\left[\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right]_{\text{CSP}} = \frac{1}{\rho_a}\left([\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}}\right)^{jk} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b}\left(S_a^{ik} + S_b ik - 2S_{\max}^{ik}\right)\overline{\nabla_{ab}^k W_{ab}} \tag{5.53}$$

2. The approach presented by Morris (1996), proposed calculating the isotropic components of the stress gradients using the conservative SP approximation and the anisotropic components using the standard (linear) SP identities. However, as was discussed in the previous section, the corrective matrix can be applied to both forms and therefore

$$\left[\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right]_{\text{CSP}} = \left[\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right]^j_{\substack{\text{CSP} \\ \text{iso.} \\ \text{consv}}} + \left[\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right]^j_{\substack{\text{CSP} \\ \text{aniso.} \\ \text{linear}}} \tag{5.54}$$

$$= \left([\nabla \otimes \mathbf{r}_a]^{-1}_{\text{SP}}\right)^{jk} \left\{ \left[\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right]^k_{\substack{\text{SP} \\ \text{iso.} \\ \text{consv}}} + \left[\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right]^k_{\substack{\text{SP} \\ \text{aniso.} \\ \text{linear}}} \right\} \tag{5.55}$$

3. This final approach follows a similar pattern, namely that spatial derivatives

in the modification must be composed of a CSP approximation, i.e.

$$\left[\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right]_{\mathrm{new}} = \left[\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t}\right]_{\mathrm{CSP}} + \frac{\tilde{\mathbf{B}}_a}{\rho}\left[\nabla \cdot \mathbf{B}_a\right]_{\mathrm{CSP}} \tag{5.56}$$

These final two modifications demand that an additional approximation in the CSPMHD algorithm. Specifically,

$$\left[\nabla \otimes \mathbf{B}\right]_{\mathrm{SP}} \tag{5.57}$$

computed using the direct (linear) or conservative SP approximations, as appropriate.

The dissipation terms are wholly more complex, and a rigorous treatment has yet to be derived. For the purposes of the work in this document, the terms remain unchanged. If the automatic switch (that evolves the α_λ parameters) is not used, the parameter is set to $0.9 \leq \alpha_\lambda \leq 1$. Where it is, the physics of the system is relied upon to constrain α_λ to reasonable figures. This has worked reasonably well, although a consistent justification is still required.

5.3.4 Implementation of CSPMHD

The implementation of the CSPMHD method is simple, assuming the SPMHD program as previously described (Chapter 3) already exists. The structure of all elements of the code structure remain the same, however the *Compute* $\frac{\mathrm{d}A_a^{<3-s>}}{\mathrm{d}t}$ component has a few additions and minor alterations relative to the structure as given by Figure 3.10. These changes are depicted in Figure 5.1. Specifically, the sweep over the pseudoparticle pairs remains in existence, however the SP approximations calculated change to suit the CSPMHD model. Subsequent to the computation of the SP approximations, an additional sweeep over the pseudoparticles (not the pseudoparticle pairs) is required, to sequentially invert $\left[\nabla \otimes \mathbf{r}_a\right]_{\mathrm{SP}}$, compute the required CSP approximations and finally compute $\frac{\mathrm{d}A^{<3-s>}}{\mathrm{d}t}$.

Note that this is not the most efficient way of calculating the CSPMHD equations in terms of the memory requirements. Assuming parallel computation, the sweep over the pseudoparticle pairs must be simplified, allowing for a merger of the SP and CSP components of the method, reducing the required memory. However CSPMHD is implemented, it will still require an increased amount of memory in order to store the increased number of variables that survive to the outside of the sweep over pseudoparticle pairs.

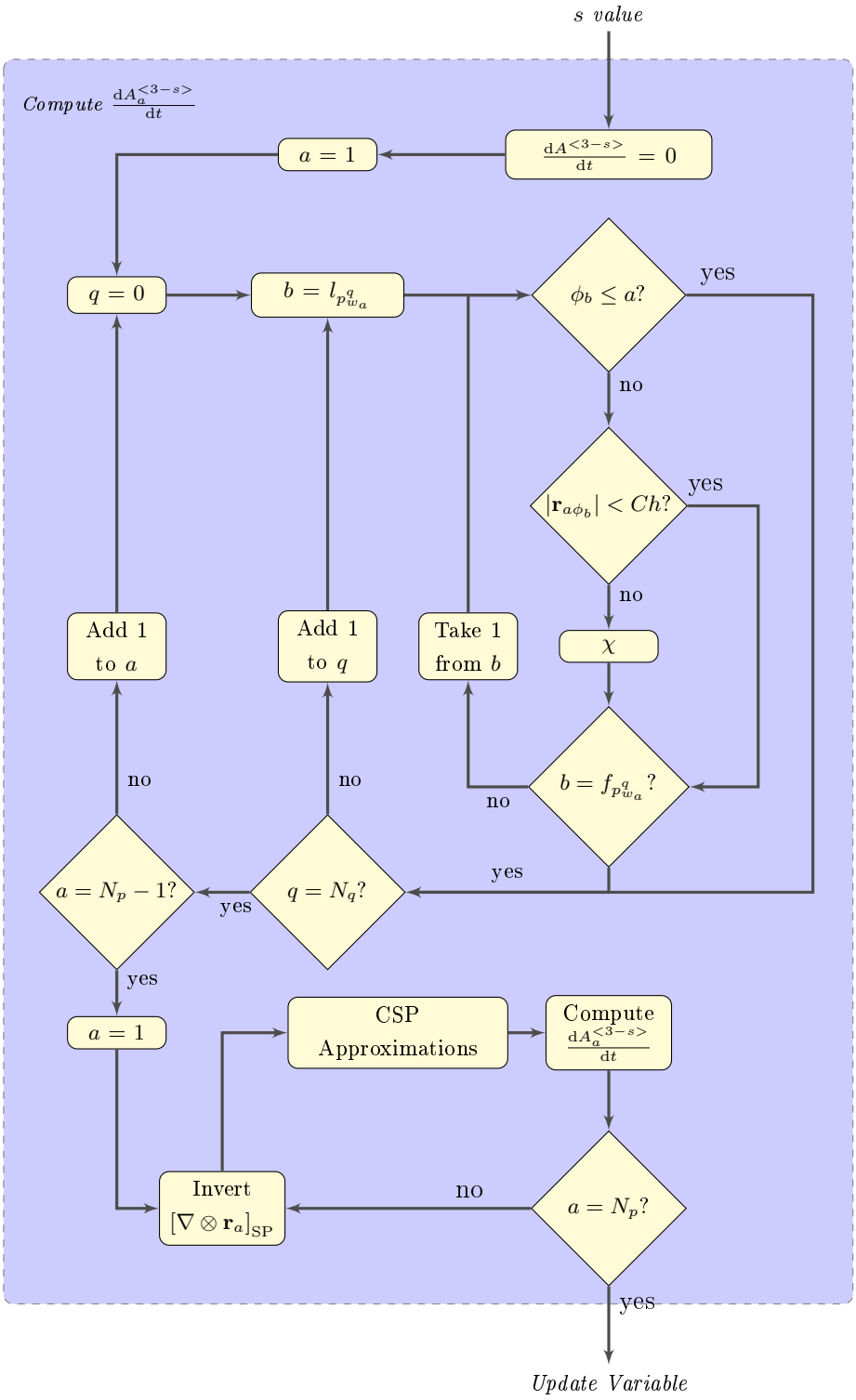

Figure 5.1: A generalised code structure diagram showing the implementation of the CSP algorithm. The node labelled χ computes the SP approximations required by the CSP method. Namely, $[\nabla \otimes \mathbf{v}_a]_{\text{SP}}$, $[\nabla \otimes \mathbf{B}_a]_{\text{SP}}$, $[\nabla \otimes \mathbf{S}_a]_{\text{SP}}$, and $[\nabla \otimes \mathbf{r}_a]_{\text{SP}}$.

5.4 Simplifying the CSP Method

Given the relative complexity and increased memory constraints of CSPMHD, when the effects of variable smoothing length are factored in as well, the resulting algorithm can be very large. However, it is possible to extend the CSP approach in order to significantly reduce the computational weight. Presented next is the derivation, and subsequent analysis, of the CSP-Δh method.

5.4.1 CSP-Δh Method

SP spatial gradient approximations built with variable smoothing length terms can be generalised by

$$[\nabla \otimes \mathbf{A}_a]_{\mathrm{SP}} = K_1 \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \frac{\partial W_{ab}(H)}{\partial \mathbf{r}_a} + K_2 \tag{5.58}$$

where K_1 and K_2 are additional factors stemming from the smoothing length gradient terms in the smoothing function derivative. That is,

$$\nabla_a W_{ab} = \frac{\partial W_{ab}(H)}{\partial \mathbf{r}_a} + \frac{\mathrm{d}H}{\mathrm{d}\mathbf{r}_a} \frac{\partial W_{ab}(H)}{\partial H} \tag{5.59}$$

for the general case.

For example, by substitution,

$$\begin{aligned}
[\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}} &= \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \nabla_a W_{ab}(H) \\
&= \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \left(\frac{\partial W_{ab}(H)}{\partial \mathbf{r}_a} + \frac{\mathrm{d}H}{\mathrm{d}\mathbf{r}_a} \frac{\partial W_{ab}(H)}{\partial H} \right) \\
&= \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \frac{\partial W_{ab}(H)}{\partial \mathbf{r}_a} + \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \frac{\mathrm{d}H}{\mathrm{d}\mathbf{r}_a} \frac{\partial W_{ab}(H)}{\partial H}
\end{aligned} \tag{5.60}$$

such that, in this case, $K_1 = 1$ and

$$K_2 = \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \frac{\mathrm{d}H}{\mathrm{d}\mathbf{r}_a} \frac{\partial W_{ab}(H)}{\partial H} \tag{5.61}$$

Equation (5.58) can be simplified further by defining the partial approximation

$$\left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{A}_a\right]_{\mathrm{SP}} = \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \mathbf{A}_{ba} \otimes \frac{\partial W_{ab}(H)}{\partial \mathbf{r}_a} \tag{5.62}$$

such that

$$[\nabla \otimes \mathbf{A}_a]_{\mathrm{SP}} = K_1 \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{A}_a\right]_{\mathrm{SP}} + K_2 \tag{5.63}$$

Now consider a repeat of the procedure that defines the CSP approximation, but begin with the expression above (5.62). The Taylor series approximation about $\mathbf{A}_b$ shows that

$$\left[\frac{\partial A_a^i}{\partial r^j}\right]_{\mathrm{SP}} = \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[\left(A_a^i + r_{ab}^k \nabla^k A_a^i + \cdots\right) - A_a^i\right] \frac{\partial W_{ab}(H)}{\partial r_a^j} \tag{5.64}$$

$$= \nabla^k A_a^i \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} r_{ab}^k \frac{\partial W_{ab}(H)}{\partial r_a^j} + O(h^2) \tag{5.65}$$

by substitution of equations (5.62) and (2.31) therefore,

$$\left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{A}_a\right]_{\mathrm{SP}} = \nabla \otimes \mathbf{A}_a \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]_{\mathrm{SP}} + \varepsilon_{\mathrm{sm}} \tag{5.66}$$

and earrangement yields

$$\nabla \otimes \mathbf{A}_a = \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]_{\mathrm{SP}}^{-1} \left(\left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{A}_a\right]_{\mathrm{SP}} - \varepsilon_{\mathrm{sm}}\right) \tag{5.67}$$

Thus, combinations of the partial SP approximation is able to be manipulated to produce the true derivative. However, the smoothing error is still not defined explicitly so that, by dropping it, a new correction procedure may be defined. Namely,

$$[\nabla \otimes \mathbf{A}_a]_{\mathrm{CSP}-\Delta h} = \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]_{\mathrm{SP}}^{-1} \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{A}_a\right]_{\mathrm{SP}} \tag{5.68}$$

whereby the full derivative can be formulated on the background of variable smoothing lengths, but without the additional strain on the smoothing function computation. Also given its similarity to the original CSP method, this new method has the same implementation. For the same reason it is also entitled the CSP-without-smoothing-length-gradient method, abbreviated for convenience to CSP-Δh .

As a further comment on the implementation, note that as the smoothing lengths

tend to some constant, the CSP-Δh method converges with the CSP method as built with constant smoothing length with minimal additional costs (just the storage and evolution of h for each pseudoparticle). The CSP-Δh approach also allows for the rapid variation of smoothing function interpretation (Section 4.3). In addition, assuming either of the intermediate perspectives is implemented, the method must conform to the reflective constraint.

5.4.2 Error in the CSP-Δh Approximation

Given the similarity to the CSP method, it is unsurprising that the errors are also very similar. The error in CSP-Δh is given by

$$\varepsilon_{\mathrm{CSP}-\Delta h} = \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]^{-1}_{\mathrm{SP}} \varepsilon_{\mathrm{sm}} \tag{5.69}$$

As with CSP, the aim must be to limit the error to system noise, ensuring that

$$\left\| \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]^{-1}_{\mathrm{SP}} \right\| \leq 1 \tag{5.70}$$

and therefore

$$\left\| \left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]^{-1}_{\mathrm{SP}} \varepsilon_{\mathrm{sm}} \right\| \leq \|\varepsilon_{\mathrm{sm}}\| \tag{5.71}$$

where $\|\varepsilon_{\mathrm{sm}}\| = O(h^2)$ is the system noise.

The matrix can be decomposed in the same way as the inversion of the full derivative (Section 5.2.2), i.e.

$$\left[\frac{\partial}{\partial \mathbf{r}} \otimes \mathbf{r}_a\right]_{\mathrm{SP}} = k\mathbf{I} + \mathbf{D} \approx k\mathbf{I} \pm \delta\mathbf{J} \tag{5.72}$$

where $k\mathbf{I}$ is the true solution (the identity matrix) multiplied by some scaling factor, and $\mathbf{D} \approx \pm\delta\mathbf{J}$ is the matrix of distortions placed on the approximation of space by the disorder of the pseudoparticles, which has been approximated by further decomposition into the unit matrix and some distortion coefficient $\pm\delta$. From this, k must still conform to the same criterion as CSP (relative to the δ here) and therefore, also conform to the inequality (5.71).

This is unsurprising. Consider that both the error in the full SP derivative and the CSP correction factor are both functions of the warping of the isotropy of space, and that the dropping of the K_1 and K_2 terms cause further warping of that isotropy. The error in the partial SP derivatives and CSP-Δh correction factor,

therefore, simply represent a different anisotropy and, by consistent formulation, correct for it in precisely the same manner. It is equivalent to the full CSP method being calculated for a more disordered arrangement of pseudoparticles, hence very similar errors.

5.4.3 Relative Accuracy in the CSP-Δh Approximation

Consider now the relative accuracy of each method. This shall be demonstrated by simulating the same problem repeatedly for each method, and a wide range of spatial resolutions. Those methods include the SP and CSP method, built with and without variable smoothing length, and the CSP-Δh method – all built with and without artificial dissipation. Thus there are 10 different algorithms; SP, SP+Δh, SP+AV, SP+Δh+AV, CSP, CSP+Δh, CSP+AV, CSP+Δh+AV, CSP-Δh , and CSP-Δh +AV.

The problem considered is the classic Sod shock tube test as described by (Sod, 1978). Specifically, a discontinuity exists between two left and right states; on the left $\{P, \rho, \mathbf{v}, \gamma, h\} = \left\{1, 1, 0, \frac{7}{5}, h_0\right\}$, and on the right $\{P, \rho, \mathbf{v}, \gamma, h\} = \left\{\frac{1}{10}, \frac{1}{8}, 0, \frac{7}{5}, \sim 8h_0\right\}$. Once allowed to evolve, a rarefaction wave forms on the left of the original discontinuity and a shock forms ahead of the advancing pressure wave. Figure 5.2 shows an example of the Sod test density, velocity and pressure output at $t = 0.245$s for the simplest (SP) algorithm.

Once simulated, in order to analyse the large multidimensional data set, the output of each model is then consolidated into a single measure of error for the variables ρ, $\mathbf{v}$ and P. The error is found by computing the area between the true solution, and the simulated data. This is achieved by approximating the smooth area as a series of trapezoids, where the vertices are the determined by the pseudoparticle data for each adjacent pair $(x_1, [A_1]_{\mathrm{SP}})$ and $(x_2, [A_2]_{\mathrm{SP}})$, and vertical intersections with the true solution, (x_1, A_1) and (x_2, A_2), and/or the possible intersection of the straight lines

$$y = \left(\frac{[A_2]_{\mathrm{SP}} - [A_1]_{\mathrm{SP}}}{x_2 - x_1}\right)(x - x_1) + [A_1]_{\mathrm{SP}} \tag{5.73}$$

$$y = \left(\frac{A_2 - A_1}{x_2 - x_1}\right)(x - x_1) + A_1 \tag{5.74}$$

The areas between the true and SP solutions are highlighted in the example plots Figure 5.2 and Figure 5.3, which are the results of the raw SP model and of the CSP-Δh +AV model, respectively.

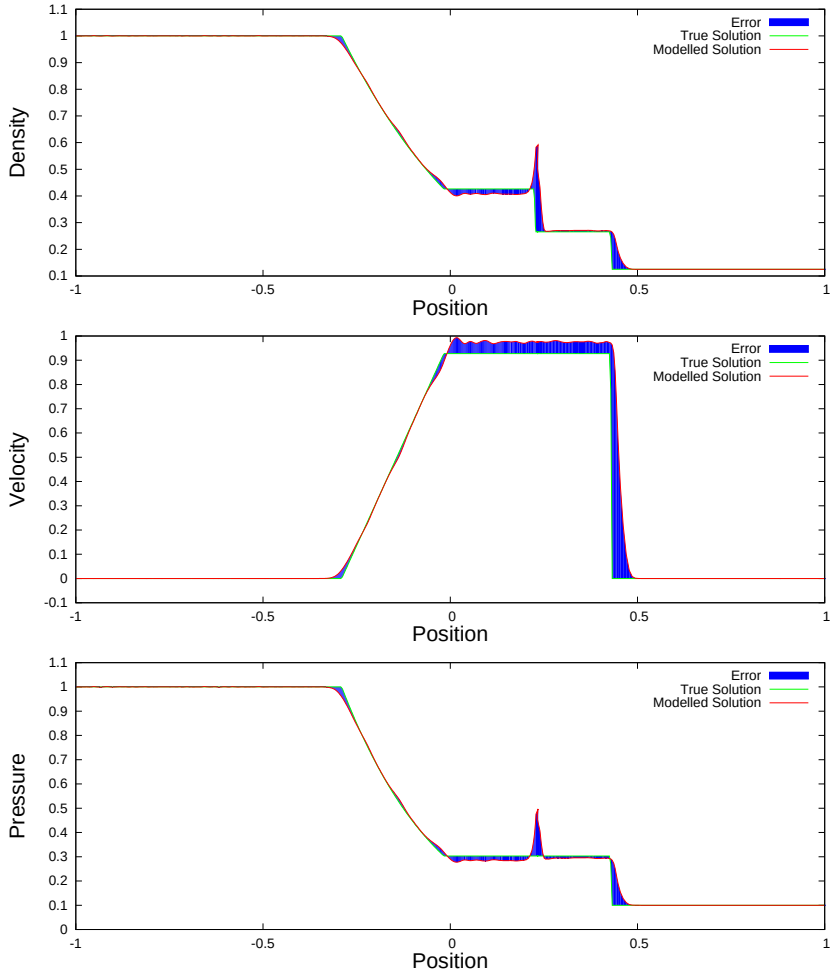

Figure 5.2: Example plots showing the true and SP solutions of the density, velocity and pressure for the Sod shock tube test. This simplest of algorithms has the most trouble with the shock interface, massively overestimating the step. In addition, note the large area between the true and SP velocity solutions, indicating the overestimation of the shock propagation speed. Also highlighted is the area between the two solutions. This is used as a measure of the error for this simulation.

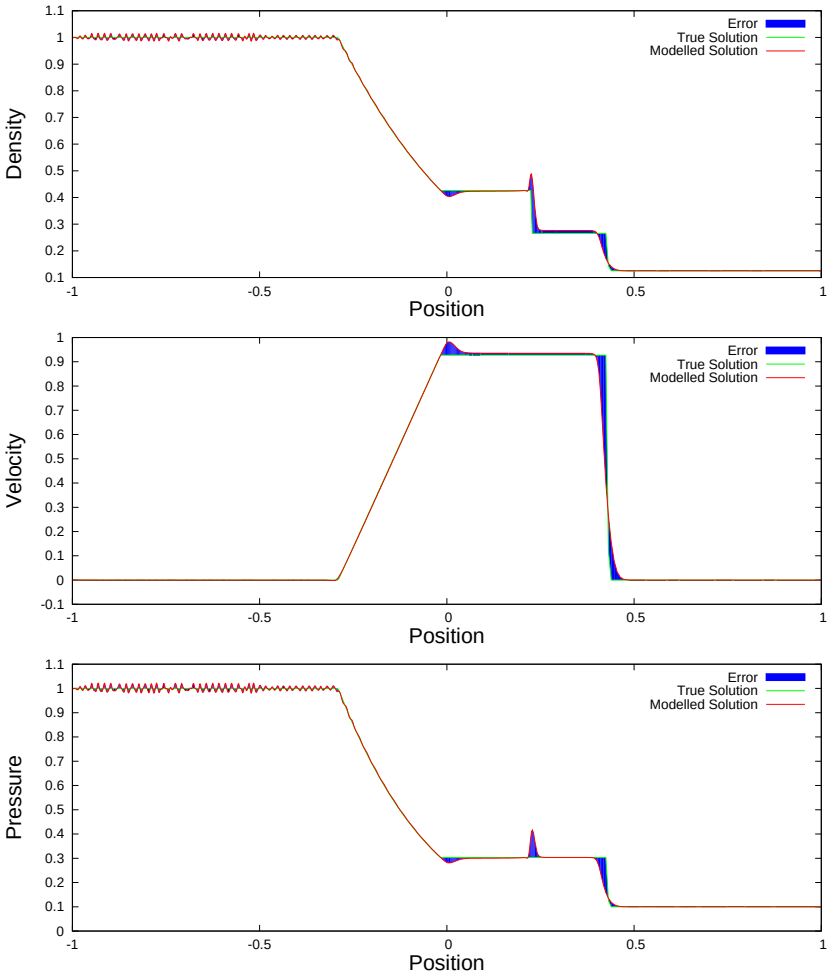

Figure 5.3: Example plots showing the true and CSP-Δh +AV solutions of the density, velocity and pressure for the Sod shock tube test. Intended for comparison with the plots in Figure 5.2, these plots highlight the increased accuracy of the CSP-Δh method over the SP method.

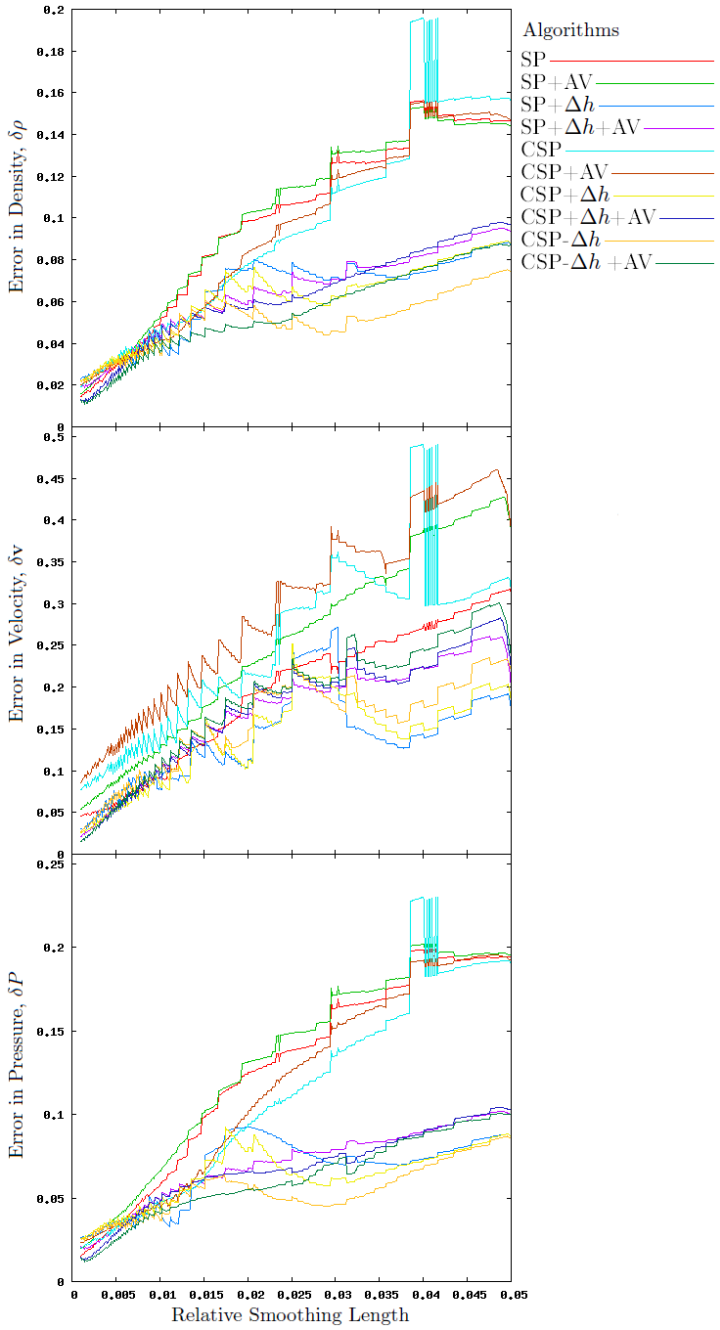

Figure 5.4: Summary of the error in each algorithm plotted against the relative smoothing length, h_0, as a proxy for the spatial resolution. The properties considered are density, velocity and pressure. The algorithms are the SP method with/without variable smoothing length, the CSP method with/without variable smoothing length, and the CSP-Δh method. Each are considered with/without artificial dissipation.

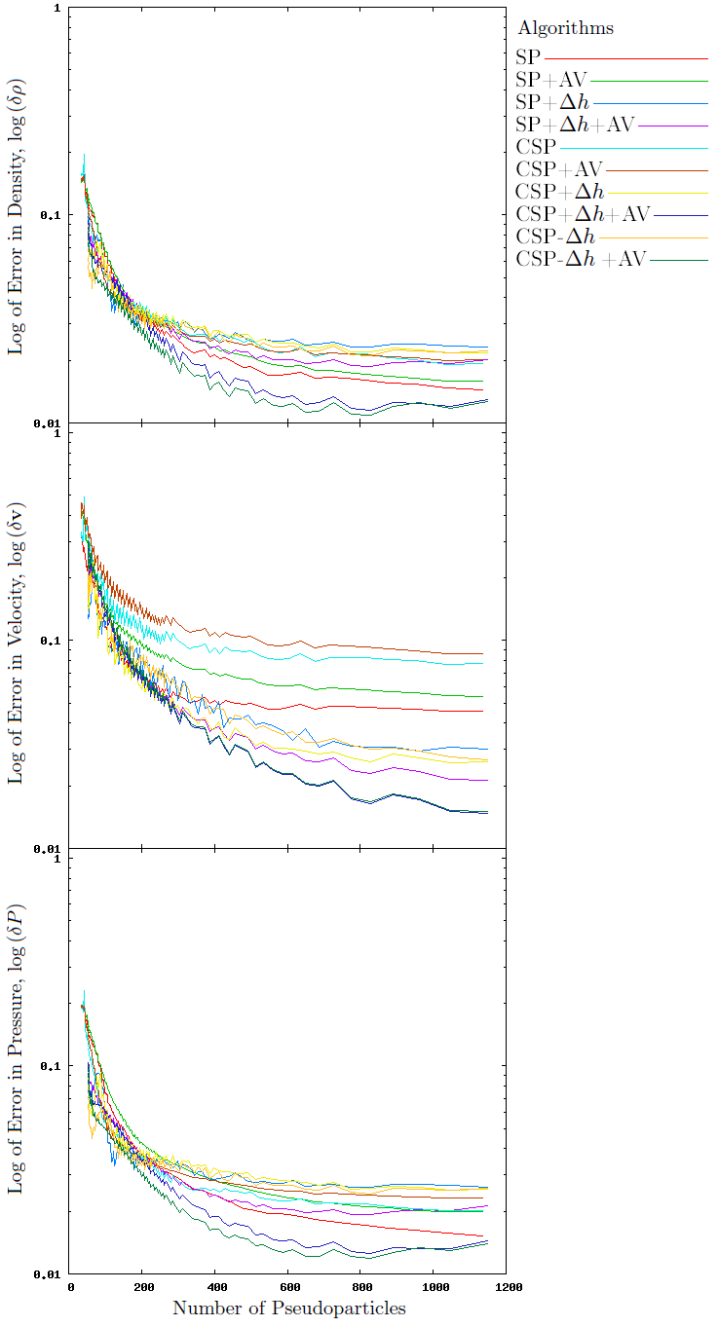

Figure 5.5: Summary of the error in each algorithm plotted against the number of pseudoparticles, N_p, as an measure of the amount of computation required. The properties considered are density, velocity and pressure. The algorithms are the SP method with/without variable smoothing length, the CSP method with/without variable smoothing length, and the CSP-Δh method. Each are considered with/without artificial dissipation.

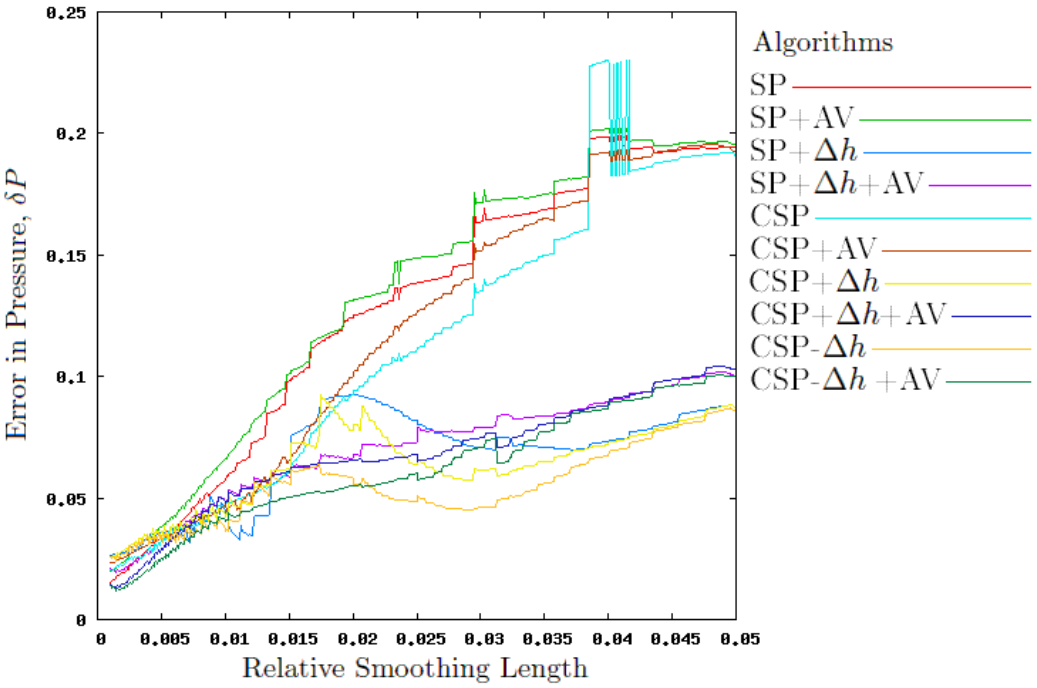

Figure 5.6: Graphs showing the error in pressure for each algorithm plotted against the relative smoothing length, h_0, as a proxy for the spatial resolution. The algorithms are the SP method with/without variable smoothing length, the CSP method with/without variable smoothing length, and the CSP-Δh method. Each are considered with/without artificial dissipation. Note the two populations separating the methods built with and without variable smoothing length, and the significantly small error, at relatively high smoothing length, in the raw CSP-Δh method.

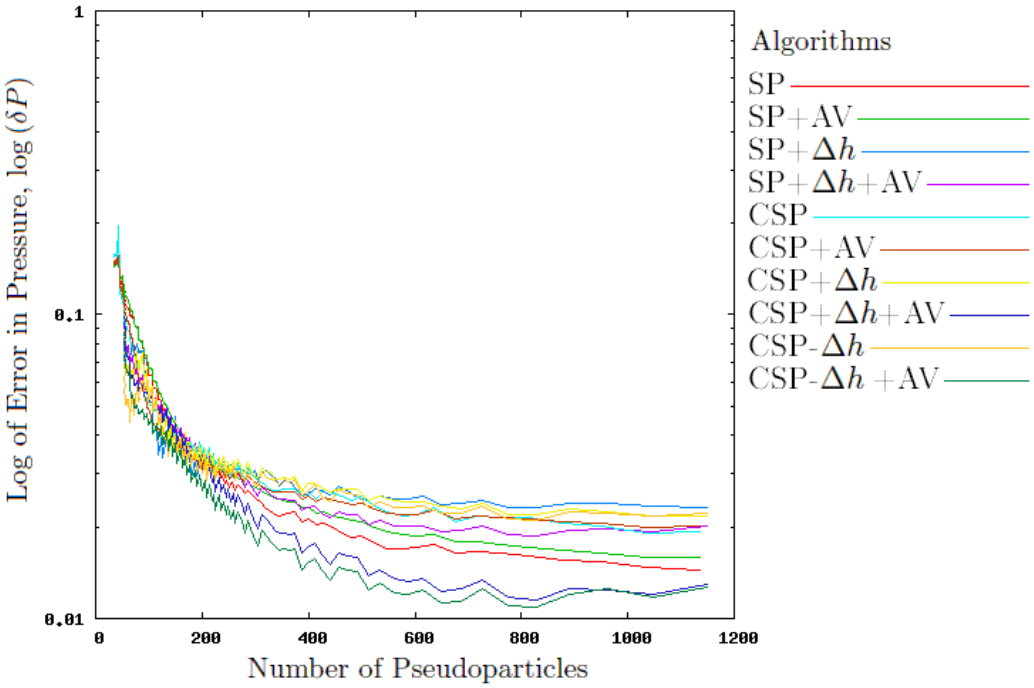

Figure 5.7: Graphs showing the error in density for each algorithm plotted against the number of pseudoparticles, N_p, as an measure of the amount of computation required. The algorithms are the SP method with/without variable smoothing length, the CSP method with/without variable smoothing length, and the CSP-Δh method. Each are considered with/without artificial dissipation. Note that the lowest errors are exhibited by the CSP with variable smoothing length and CSP-Δh (both with artificial dissipation) in the region $N_p > 200$. Also note that each method tends (approximately) towards some constant error as N_p increases.

These errors are then plotted against the relative smoothing length, h_0, or the number of pseudoparticles, N_p, so that the relative accuracy of each algorithm can be determined. The results are summarised in Figures 5.4 and 5.5. Figures 5.6 and 5.7 show specific plots pulled from the summaries.

Consider first Figure 5.6. There are two clear populations (at high smoothing length) separating the methods built with and without variable smoothing length, indicating that algorithms built with variable smoothing length, no matter the remaining details of the application, allow for a lower resolution with equivalent accuracy. Also consider the particularly small error, at relatively high smoothing length, of the raw CSP-Δh method. Finally note that as the smoothing length tends to zero the error in each method appears to converge. Figure 5.7 shows this to be an artefact of the relative image scale. The errors for each method tend (approximately) towards individual constant errors as N_p increases. Most significantly, note that the two lowest errors are exhibited by the CSP with variable smoothing length and CSP-Δh methods (both including artificial dissipation) in the region $N_p > 200$. Rather unexpectedly, but convincingly, the smallest errors are exhibited by the CSP-Δh method. These show that much lower pseudoparticle numbers (and therefore lower computational weight) is possible with the variable smoothing length CSP and the CSP-Δh methods. This experimentally supports the analysis conducted by Vila (1999), in which he concluded that "[CSP] *turns out to be more robust than the standard* [SP] *methods, and also less expensive since we can use higher values of the ratio* $(\Delta x)/h$. *This decreases the number of neighbours...and the cost of the method also decreases proportionally.*"

5.5 Conclusion

It was the intention of this work to better understand the source of errors in SP simulations. This was achieved by analysing and eventually defining the error due to the discretisation approximation as

$$\begin{bmatrix} \varepsilon_{\mathrm{dc},\lambda} \\ \varepsilon_{\mathrm{dc},\mathbf{A}} \\ \varepsilon_{\mathrm{dc},\mathbf{T}} \end{bmatrix} = \left([\nabla \otimes \mathbf{r}_a]_{\mathrm{SP}} - \mathbf{I}\right) \begin{bmatrix} \nabla\lambda \\ \nabla \otimes \mathbf{A} \\ \nabla \otimes \mathbf{T} \end{bmatrix} \tag{5.75}$$

The definition has several troubling factors. The error is a function of the anisotropy of space, which can only be reduced by significantly increasing the number of pseudoparticles. In addition, and far more worryingly, the error is proportional to the gradients being approximated.

The error must, for highly dynamic simulations like the Corona, cause a particular problem. As such the CSP method was presented that eliminated the error. However, its implementation proved complex and weighty. Hence the novel CSP-Δh method was subsequently presented that not only eliminated the error but the cause of the computational weight in the CSP algorithm as well.

Finally, with a stable and (more importantly) accurate algorithm it is now possible to construct a simulation. Thus the construction of a suitable domain for the simulation of phenomena within the Corona can now be discussed in the following Chapter.

Chapter 6

Domain Construction for Solar Phenomena

Contents

6.1 Introduction

With the completion of an accurate SPMHD algorithm (actually, CSP-Δh MHD) simulations can be constructed. At this point, the research could take a number of paths including, for example, comparison of additional MHD codes with CSP-Δh MHD. However, given this documents focus on the eventual simulation of coronal phenomena, of the several possible paths available those that make appropriate steps towards that goal were given preference. In this final research chapter an investigation of the simulation domain, Ω, is conducted.

Specifically, using the context provided by the geometry of the Solar Corona and the constraints of the algorithm, an efficient domain is constructed using appropriate boundary conditions.

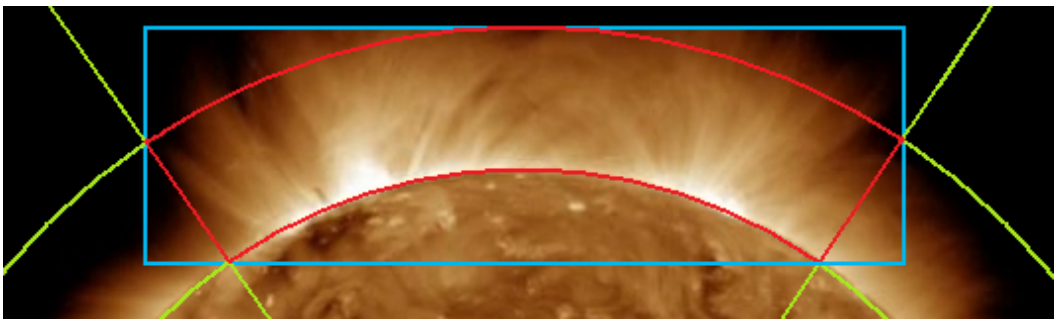

Figure 6.1: Graphical representation of a curved surface model and rectangular domain in 2-Dimensions about the coronal loops of two active regions on the Sun (base image credit: NASA). Shown in blue are the bounds of a rectangular domain, and in red are the bounds of a curved surface model. Of particular note is the variation in the physics of the medium adjacent to the blue boundaries. Also, the amount of non-coronal material encompassed by those boundaries.

6.2 Curved Surface Model

Consider the simulation of mega-metre scale phenomena in the solar corona. These phenomena would include, for example, Coronal Bright Fronts (CBFs, also referred to as EIT waves), Filaments/Prominences and Coronal Mass Ejections (CMEs). All of these curve appreciably about the surface of the Sun. It is contended here that to model these large, sub-solar scale dynamics a new domain must be created.

Consider some general, 2-Dimensional phenomenon simulated in a rectangular domain, shown in blue in Figure 6.1. Notice that the physics throughout the domain is highly variable, and any solution will required the simulation of multiple fluids – at the very least coronal plasma and a single state of sub-coronal plasma. More detrimentally, the physics and the *orientation* of the physics also varies along the length of the boundaries. This means that periodic and free surface boundaries cannot be implemented, leaving only solid boundaries. However, these solid boundaries are difficult to utilise as they require ghost particles (requiring large resources - particularly in 3-Dimensions) and some complex definition of their physical behaviour that is likely to be subject to the evolution of the simulated plasma. All of these are oportunities to introduce error into the simulation. Also note that, assuming the interesting portion of the model is the evolution of the phenomena, there is a large amount of wasted simulated space (and therefore memory in the form of additional pseudoparticles).

In order to keep the required computation as low as possible, the extent of the domain should be limited to the region required by the dynamics (shown in red

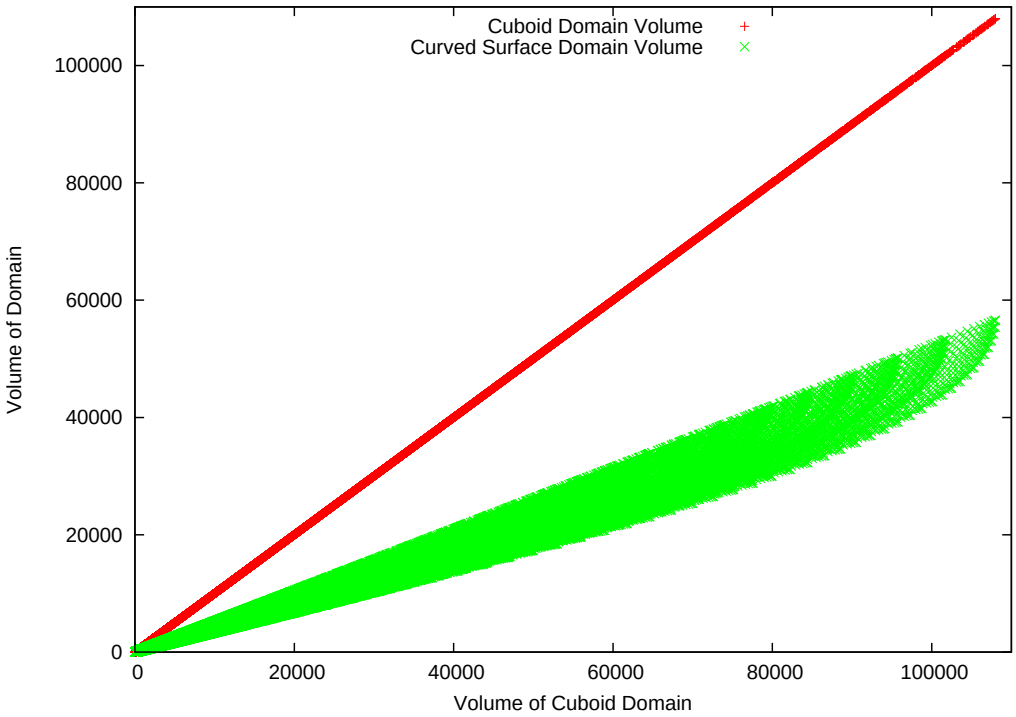

Figure 6.2: A scatter plot comparing the volume (as a proxy for the approximate number of pseudoparticles) required to encompass a given subset of the surface. That subset is defined by spherical polar coordinates. The surface exists between the angles $0 < \phi, \theta < \frac{\pi}{2}$, a lower radial boundary set at $r = 1$ and an upper boundary defined between $1 < R \leq 30$. Notionally, the radial distances are defined in units of solar radii, $R_{\odot}$. Note that, despite the spread in curved surface domain values, the reduction in volume (i.e. pseudoparticles) is approximately 50%.

in Figure 6.1). That is, rather than placing a large rectangular domain about the phenomenon it is supplanted by a domain that better conforms to the geometry of the system. Consider a 3-Dimensional equivalent of the 2D domain shown in Figure 6.1 (such as that in Figure 6.3). The scatter plot shown in Figure 6.2 illustrates the potential reduction in volume of this new domain compared with the volume of a cuboid capable of encompassing the new domain. Note that there is some spread in the value of the new domain volume (against cuboid volume) as it is a function of the spherical geometry. Even with this divergence, the new domain approximately halves the required simulated volume.

However, this new domain still has to allow

i) The formation of sensible boundary conditions, and

ii) Consistency with the geometry of the algorithm governing the dynamics.

The first point is relatively simple to achieve, though is a function of the required boundaries. For instance, a background boundary can be warped into almost any required shape (as long as it curves over an appreciable distance relative to the smoothing length), however periodic boundaries are more stable when implemented with flat boundary surfaces.

The second point, demanding consistency between the domain and the dynamics algorithm, places certain limitations on the domain. Given the spherical nature of the body on which the phenomena take place (i.e. the Sun), it would appear efficient to express the CSP/CSP-Δh equations in spherical coordinates. However, the equivalent spherical correction factor in the CSP or CSP-Δh methods cannot be separated from the SP approximations. Hence the Corrected models must be implemented using Cartesian coordinates, which limits the form of the domain. Specifically, the boundaries are much easier to implement if they are built symmetrical about the Cartesian axes (see Figure 6.4).

Consider Figure 6.3. Represented in the centre of the frame is some general domain that sits on the curved surface of the sun. It is bound by the intersection of six planes. Four of those plane are flat and intersect each other at the centre of the sun. They exist to limit the angular extent of the domain by creating an open-ended square-based pyramid that extends out into interplanetary space. This is termed, here, the *wedge* and is characterised by the angles 2θ and 2ϕ which define the angular separation of the North-South planes, and the East-West planes. These are shown in Figure 6.4. The remaining two boundaries are concentric spherical shells, centred on the sun's centre. The smaller sits at the interface between the lower Corona/upper Chromosphere and the body of the sun, as determined by the user – though it should be low enough that the phenomena is within the domain. The upper surface, being further out in a curved space should be set a low as possible without interfering directly with the modelled dynamics. This is important as even a small increase in height can introduce a large number of new pseudoparticles. This latter constraint can be weakened by implementing an algorithm built variable smoothing lengths, though the numerical artefact flow can influence the results.

The defining characteristic of the implementation of this domain must be the nature of the boundaries. Assuming that the curved surfaces are some type (or types) of edge boundary (see Section 6.4), consider the plane boundaries.

6.3 Periodic Geometry

Here it is assumed that the four planes represent some form of angular periodicity within the model. Specifically, the angular components of the pseudoparticle

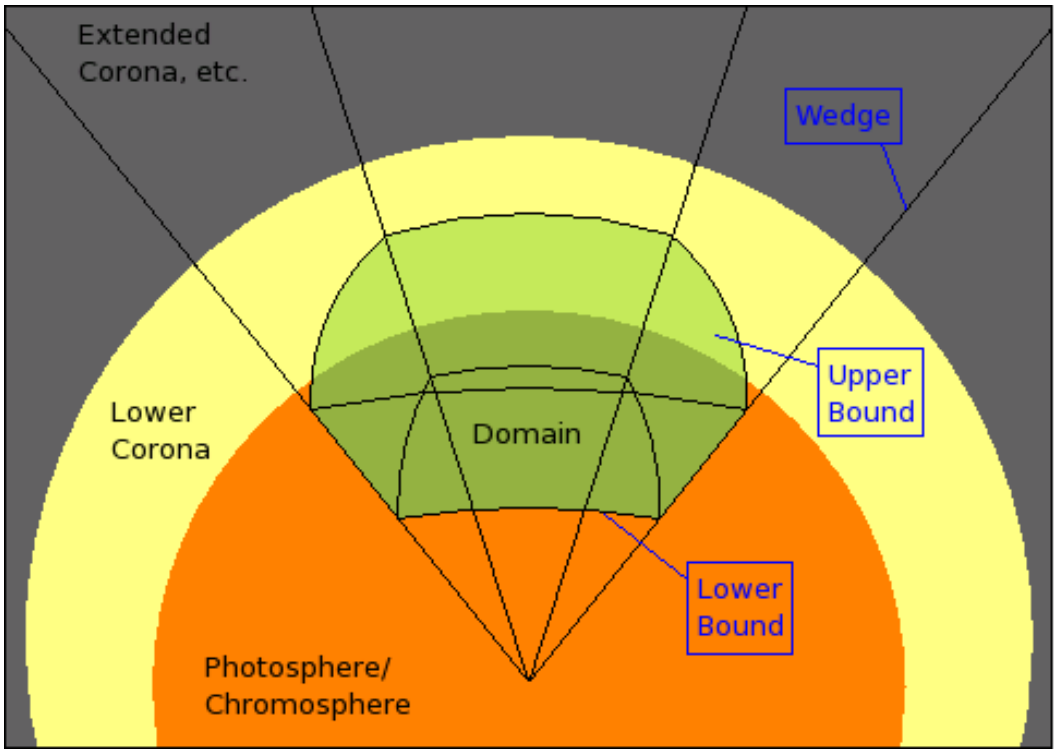

Figure 6.3: A pictographic representation of the sola/curved surface domain. The orange ball represents the surface of the sun, the yellow layer about it is the lower Corona and/or upper chromosphere, and the grey region about that is everything from the upper Coronal out into the Solar Wind and the heliosphere beyond. Also presented, highlighted green, is the curved surface domain. A region of space, the extent of which is constrained by upper and lower curved boundaries conforming to the intersection of the upper and lower limits of the yellow region and a *wedge*. That wedge is a baseless square-based pyramid extending to infinity, with its apex fixed to the centre of the sun.

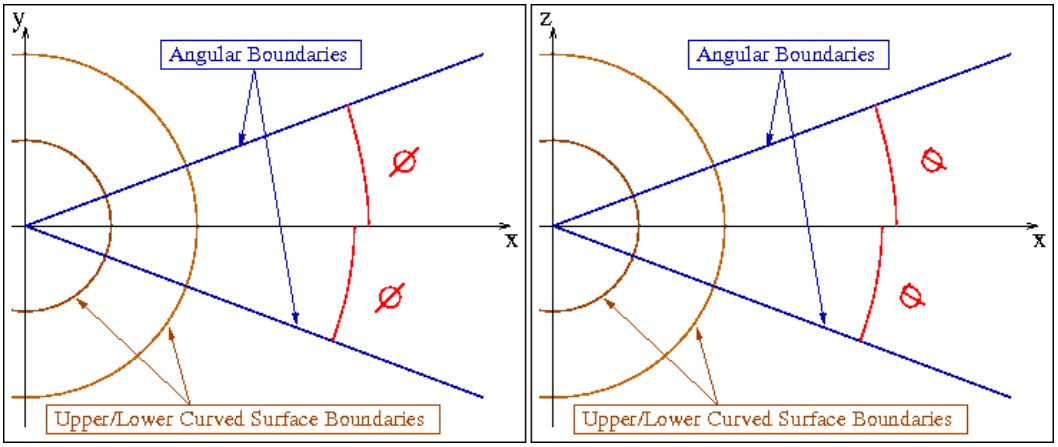

Figure 6.4: Plots showing the xy- and xz-plane intersections with the curved surface domain. Shown in brown are the upper and lower curved surface boundaries. In blue are the periodic angular boundaries, defined by the angles ϕ_B and θ_B (shown in red) made with the x-axis.

position (expressed in spherical coordinates) are constrained such that

$$|r_a^\theta| \leq \theta_B < \frac{\pi}{2} \tag{6.1}$$

$$|r_a^\phi| \leq \phi_B < \frac{\pi}{2} \tag{6.2}$$

Consider some pseudoparticle moving through the domain, and then through the Eastern plane. To an external observer, the pseudoparticle should blink out of existence and reappear coming through the Western plane. However it should, relative to each plane, have the same relative properties of direction. Thus it is not enough to translate the particle in a straight line from where it is in the east to where it should be in the west. All of its vector properties need to be rotated about the centre of the Sun (the intersection point of all four flat planes).

6.3.1 Usual Periodicity

This rotation is achieved, if the case of an east-west (or vice versa) transformation by

$$[A_a]_{\text{new}} = R_z(\phi)\,[A_a]_{\text{old}} \tag{6.3}$$

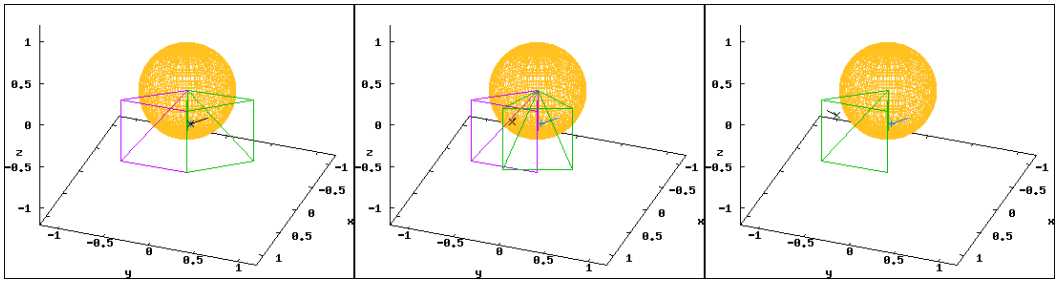

Figure 6.5: The panels here show (left to right) the rotation of a single pseudoparticle (with some vector property) from the Eastern ghost wedge back into the domain wedge. For clarity the wedges are shown as sealed square-based pyramids, though in reality they are open to simulated space.

where $A_a = \{\mathbf{r}_a, \mathbf{v}_a, \mathbf{B}_a\}$, R_z is the rotation matrix (gimbal about z) given by

$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

and $\phi = -2\phi_B I\left[\frac{r_a^y}{r_a^x \tan\phi_B}\right]$ where $I(x)$ is again a function that truncates x, leaving only the integer component remaining. Conceptually, a ghost wedge appears, to the East or West depending on which of the two boundaries the pseudoparticle violated, around the errant pseudoparticle then the ghost wedge takes possession of the pseudoparticle and the wedge is rotated until the ghost wedge occupies the exact position of the domain wedge. This rotation is shown in Figure 6.5.

A similar rotation is required for pseudoparticles violating the Northern or Southern boundary so that they too are periodicly bound. Specifically,

$$[A_a]_{\text{new}} = R_y(\theta)\,[A_a]_{\text{old}} \tag{6.5}$$

where R_y is the rotation matrix (gimbal about y) given by

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{6.6}$$

and $\theta = 2\theta_B I\left[\frac{r_a^z}{r_a^x \tan\theta_B}\right]$. This rotation is shown in Figure 6.6.

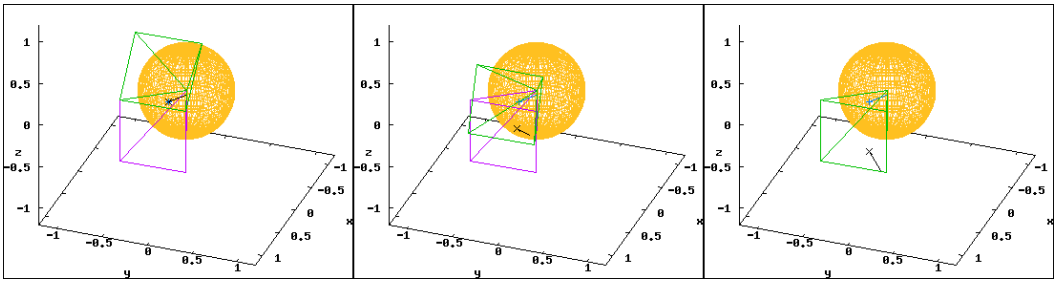

Figure 6.6: The panels here show (left to right) the rotation of a single pseudoparticle (with some vector property) from the Northern ghost wedge back into the domain wedge. For clarity the wedges are shown as sealed square-based pyramids, though in reality they are open to simulated space.

6.3.2 Complex Periodicity

Consider a pseudoparticle violating two of the periodic boundaries simultaneously (both North or South *and* East or West). Were the boundaries defined on a Cartesian coordinate system then the periodicity could be enforced by applying the translations from North/South ghost wedges and East/West ghost wedges sequentially. However the domain wedges cannot tessellate about their apexes. Alternatively, consider that the curved geometry ensures that applying the rotations about the y- then z-axes is not the same as applying the rotations about the z- then y-axes, i.e.

$$R_y(\theta)R_z(\phi)A_a \neq R_z(\phi)R_y(\theta)A_a \tag{6.7}$$

Thus, if a pseudoparticle drifts through a corner, it must be captured by some new corner ghost wedge.

Given that the pseudoparticle has violated the limits in r_a^θ and r_a^ϕ, the corrections here are more complex. First consider the definition two new angular properties, as functions of θ_B and ϕ_B,

$$\alpha_B = \arctan\left(\frac{\tan\theta_B}{\tan\phi_B}\right) \tag{6.8}$$

$$\beta_B = 2\arctan\left(\sqrt{\tan^2\theta_B + \tan^2\phi_B}\right) \tag{6.9}$$

where, for ease of explanation, β_B is twice the angle between the x-axis and the vector joining the origin and $\mathbf{r}'$ which is a point anywhere along the intersection between two boundary planes. α_B is the angle between the z-axis and the vector joining the origin and $(0, y', z')$ where the y and z coordinates are components of the point $\mathbf{r}'$. With these definitions, the periodicity can be enforced by the multiple

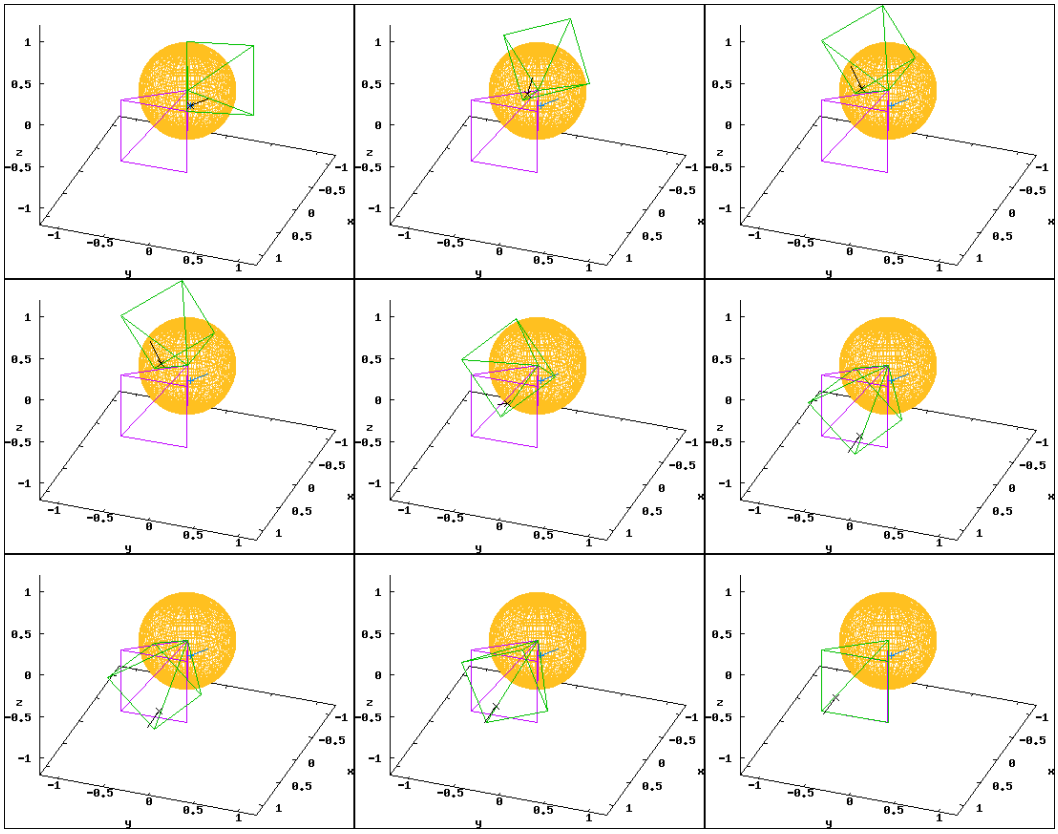

Figure 6.7: The panels here (read as one would read text on a page, i.e. top row left to right, followed by second row left to right, etc.) shows the multi-stage rotation of a single pseudoparticle (with some vector property) from the North-Eastern (Edge Interface) ghost wedge back into the domain wedge. The top row shows the rotation from A_a to $R_x(\alpha)A_a$, the middle row shows the rotation from there to $R_y(\beta)R_x(\alpha)A_a$, and the lower three panels show the final rotation to $R_x(-\alpha)R_y(\beta)R_x(\alpha)A_a$. For clarity the wedges are shown as sealed square-based pyramids, though in reality they are open to simulated space.

rotations,

$$[A_a]_{\text{new}} = R_x(-\alpha)R_y(\beta)R_x(\alpha)\,[A_a]_{\text{old}} \tag{6.10}$$

where R_x is the rotation matrix (gimbal about x) given by

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{6.11}$$

and the two angles are $\alpha = \text{sgn}\,(r_a^y r_a^z)\,\alpha_B$ and $\beta = \text{sgn}\,(r_a^z)\,\beta_B$. This series of rotations is shown in Figure 6.7.

6.3.3 Implementation of Angular Periodicity

Wherever the distance between pseudoparticle pairs is calculated within the implemented algorithm, the rotational periodicity must be taken into account. For an implementation built with a bucket sort, this can be made easier. Assuming that the cells (though *not* the contents) remain static, then only those pseudoparticles sorted into cells near/against a boundary could possibly interact with the pseudoparticles near/against the opposing boundary. By creating a list (explicit or implicit) of those cells, all the remain cells, and the pseudoparticles within them continue to evolve as per the previously defined Cartesian algorithm, with no knowledge required of the boundaries.

For those listed cells, it should be made clear whether they interact with the boundary intersections (corners) or planes (walls) of the domain, as the rotations required are different – functions of θ_B and ϕ_B, or α_B and β_B respectively. Once the required rotations are identified, the procedure for calculation is identical.

To illustrate consider two pseudoparticles a and b that lie against the North and South boundaries, respectively, such that with the periodicity $a \in \xi_b$ and $b \in \xi_a$. When calculating the contribution of b to a, b must appear to be in the Northern ghost wedge, with the appropriate rotations to all vector properties. This is, essentially, putting b in the reference frame of a. The opposite must be done when calculating the contribution of a to b. That is, a must appear to be in the Southern ghost wedge and hence in the reference frame of b.

6.4 Edge Boundaries

There is no symmetry between the upper and lower curved surface boundaries thus, assuming the simulated plasma must be confined by both, they must be configured using ghost particles. As the source of the energy that feeds the dynamics in the Corona, it is logical to implement a data boundary as the lower surface to act as a driver. This data could either be artificial or remote-sensed from the Sun. In the latter there are possible complications due to line of site integration, though these can be avoided by reducing the angular extent of the domain.

The upper surface is more difficult to configure. In part, this is due to the fact that it can be set to almost any height away from the solar surface. Placing the boundary far away limits the amount the boundary configuration can impinge on the simulated dynamics. On the other hand, the further away the boundary is placed, the more pseudoparticles are required to populate the additional volume. In this 3D space the increase is proportional to $H^3 + 3HL(H + L)$, where L is the

height of the lower boundary above the centre of the Sun and H is the height of the upper boundary above the lower boundary. Clearly what is required is some measure of compromise between the two, or additional resources to compensate the extra workload (either of simulating more plasma or cleaning up the error caused by the boundary).

The upper surface is also more complex because of the Solar Wind. A large amount of the material in the Corona is jettisoned into interplanetary space in the form of the solar wind, as well as massive transients. Large transients are less problematic from the perspective of the boundaries as, assuming some form of artificial solar flare or CME occurs, it is more than likely the intention of the simulation is to study those phenomena, and therefore the boundary will be set back from the transient. However, the constant stream of subsonic solar wind must impinge on and influence the nature of the upper boundary. At the extreme, placing a solid boundary (Section 3.3) at the top of the simulation would drastically effect the simulation. Equally, a free surface boundary would be inappropriate as, with out information being passed back through the medium of the solar wind, the pseudoparticles would rapidly become supersonic and significantly warp any modelled phenomena. Again, this suggests compromise, logically in the form of a background boundary.

6.4.1 Interpolated and Background Boundaries

Consider the background boundary. As an edge boundary, it relies on ghost particles (ghosts) set up along the edge of the boundary, possibly in depth, affixed with particular behaviour. In the case of the background boundary, that behaviour is relatively simple. Each ghost has the same range of properties and behaviour as a fluid pseudoparticle with one difference – a lack of evolution. The properties, as defined during the initialisation phase of the model do not change over the course of simulated time, ie.

$$A_g^{<n>} \equiv A_g^{<0>} \tag{6.12}$$

for any n, where $A = \{\mathbf{r}_g, \mathbf{v}_g, \mathbf{B}_g, \rho_g, u_g, P_g, \gamma, h_g, \cdots\}$.

To avoid instabilities and non-physical behaviour, the boundary must be positioned away from the expected dynamic behaviour so that no information pertaining to the dynamics is passed to the boundary. Thus, the ghosts should only interact with fluid pseudoparticles with properties that are also unchanging. Take, for example, the Sod shock tube tests run in Section 5.4.3. Given the quantifiable wave speeds, the dimensions of the dynamic portion of the simulation is easily calculated at some time $t_{\max}$, and therefore the theoretical maximum extent of the domain can be defined. However, to ensure no transfer of information to the ghosts, the

boundary is fixed relatively far from the centre of the domain.

This approach is computationally costly as the extra domain space must be populated by pseudoparticles. This is particularly problematic for multidimensional simulations, as they suffer from the same issue as defence-in-depth (proportional to r^2 - see Section 6.4). It also assumes that the spatial scale of the dynamics is predictable and/or that the simulation will run for a defined (usually short) period of simulated time.

In an attempt to counter these issues, consider some form of *interpolated boundary* where the ghosts properties are interpolated from the local fluid pseudoparticles. Assuming some kernel interpolation method,

$$K_i = \frac{\sum_j K_j Y(\mathbf{r}_i - \mathbf{r}_j)}{\sum_j Y(\mathbf{r}_i - \mathbf{r}_j)} \tag{6.13}$$

where, for SP-type methods, the summation is conducted over the local pseudoparticles and $Y = \frac{m}{\rho} W$ (Section 2.2), such that

$$A_g^{<n>} = \left[\sum_{a \in \xi_g} \frac{m_a}{\rho_a} W_{ga} \right]^{-1} \sum_{a \in \xi_g} \frac{m_a}{\rho_a} A_a^{<n>} W_{ga} \tag{6.14}$$

where $\xi_g = \{a | g \in \xi_a\}$ However, this too has issues. It can, for instance, artificially draw energy from the system.

6.4.2 Formulation of the Interpolated-Background Boundary

Consider some transient pulse impacting the boundary (Figure 6.8). The upper panel shows the impact of the pulse, and the manner by which the background ghosts artificially cause steepening of the pulse profile, inevitably slowing the rate at which information leaves the domain. The central panel represents the impact of the pulse on the interpolated ghosts which artificially accelerates the outflow of information from the simulation.

Consider now the definition of the interpolated-background boundary. The ghost properties are computed by

$$A_g^{<n>} = (1 - \eta) A_g^{<0>} + \eta \left[\left(\sum_{b \in \xi_g} \frac{m_a}{\rho_a} W_{ga} \right)^{-1} \sum_{a \in \xi_g} \frac{m_a}{\rho_a} A_a^{<n>} W_{ga} \right] \tag{6.15}$$

where ξ_g keeps its prior meaning, and η is some external parameter. Conceptually, the boundary has been re-imagined as some form of rubber sheet which is tied to

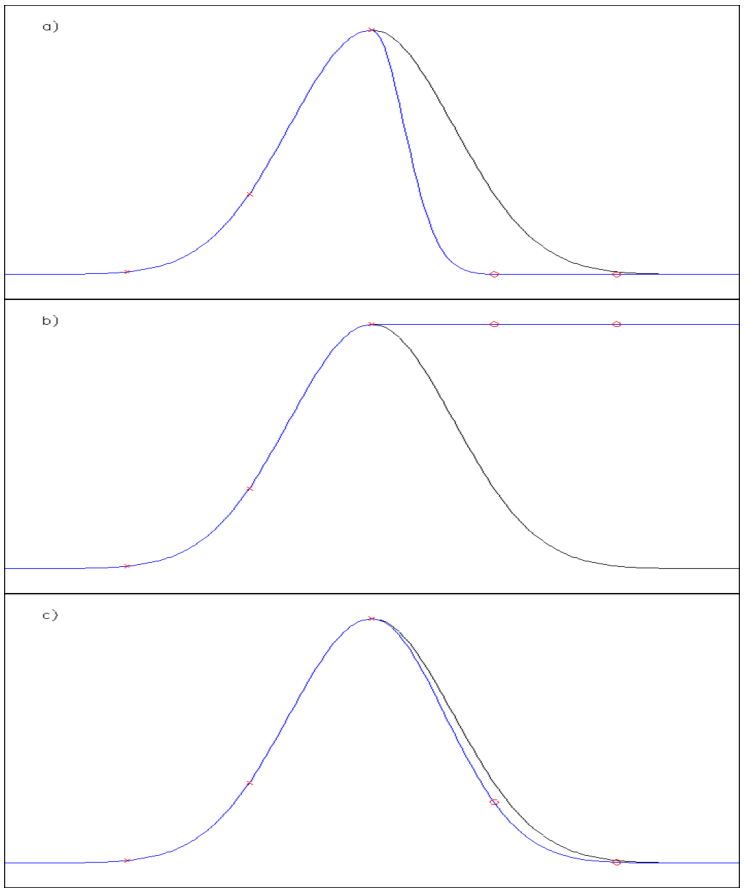

Figure 6.8: The panels (top to bottom) show the background, interpolated, and interpolated-background boundary, respectively, being struck by some transient pulse. Specifically, each shows the property response of the two-deep ghost particles. Notice how the top two responses respectively damp or accelerate the pulse.

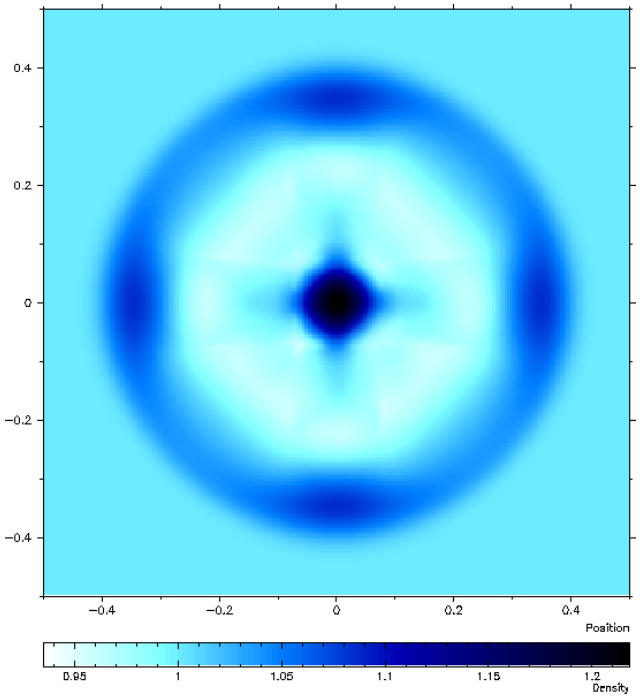

Figure 6.9: Plot showing example density data for the simulation used to evaluate the stiffness coefficient, η, for the interpolated-background boundary.

the initial background boundaries but is able to flex in order to match incoming property profiles. In this context, η determines the relative stiffness of the rubber sheet. When $\eta = 1$, the boundary flexes to entirely match the surrounding fluid, and when $\eta = 0$ the boundary becomes entirely inflexible and is ignorant of the state of the fluid. Clearly, also, at these extreme values the boundary behaves in the same way as the interpolated and background boundaries, respectively. The interpolated-background boundary is shown in the lower panel of Figure 6.8.

6.4.3 Determining the Stiffness Coefficient, η

The stiffness coefficient, η, is a parameter and must, therefore, be determined by experiment. Presented here is a simple analysis of a circular pressure wave repeatedly striking the boundary (see Figure 6.9), with η changing with each cycle. Released from the centre of a unit square (2D) domain, the wave impacts the boundary approximately 2 simulated seconds after release. Subsequently the boundary will either actively remove or inject information, depending on the η value.

The error in the results is determined by calculating the absolute difference between the results of the simulations, as described above, and the same modelled phenomena computed on a domain three times the size of the unit domain. This

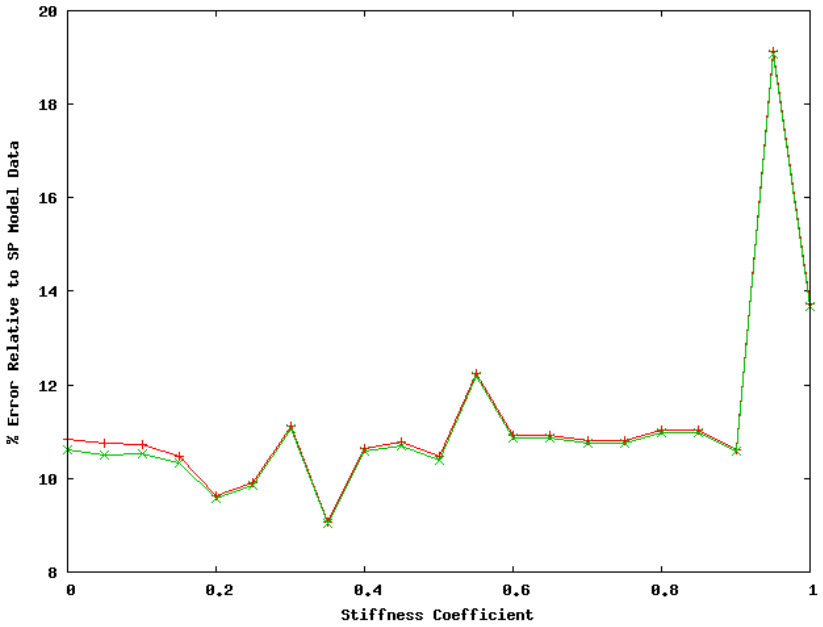

Figure 6.10: A graph showing the average percentage error in the properties of an simulation conducted in a unit box bound by the interpolated-background boundary relative to the same model run in a domain three times the size against the stiffness coefficient, η. The pseudoparticle behaviour in the larger domain simulation provide a benchmark for the expected behaviour of the pseudoparticles in the boundary regions of the smaller simulation. The two lines represent the errors as calculated on axis running North-South (red) and East-West (green). High η values cause larger errors and low η values cause a loss of symmetry.

second simulation produces data about how the true fluid pseudoparticles would respond in the boundary region. The errors are assessed, for each η value tested, along a North-South and East-West line. The results can be seen in Figure 6.10.

Though the stiffness coefficient varies over a large range, the errors are approximately equal at about 10% for most of that range, the exception occurring at high η values where the error is significantly higher. High stiffness coefficient corresponds to a larger interpolated component of the ghost properties. Consider the hypothesis that the large jump in error corresponds to the point at which the information is being drawn out of the simulation faster than the wave is transiting. Without higher resolution in η, the hypothesis cannot be tested, but assuming that the interpolated boundary causes more information to flow out of the simulation then it is just the sort of behaviour that should be expected. Once the rate at which information is being drawn out of the system (which cannot be measured) exceeds the wave speed, the wave should accelerate as if in the presence of an external driver. The simulations also maintain reasonable symmetry until, conversely, η drops to low values,

where the errors separate by $> 0.5\%$. The difference is small, but gets progressively more pronounced as η decreases, possibly implying some causal relationship. Assuming there is a link between the two, this could be attributed to the damping effect of the background boundary (see top panel Figure 6.8). However, more data is required to extend the argument any further.

Given the present data, $\eta \approx 0.35$ appears to have the lowest error, however, almost all errors are within 2% of the the error at $\eta = 0.35$.

6.5 Conclusion

The research in this chapter described the construction and bounds of a domain that took advantage of the spherical geometry of the solar coronal environment. That domain was, however, limited by the nature of the CSP-Δh MHD technique, which can only be implemented in Cartesian coordinates. Despite the restriction, the reduced volume (and hence reduced computational requirements) equated to approximately 50% of the cuboid domain required to encompass the same portion of the solar surface.

In addition, relatively complex periodicities were defined that wrap themselves to the solar surface accounting for a some of the volume reduction. A new interpolated-background boundary was also developed and investigated that defined the properties of ghost particles as

$$A_g^{<n>} = (1-\eta)\, A_g^{<0>} + \eta \left[\left(\sum_{b \in \xi_g} \frac{m_a}{\rho_a} W_{ga} \right)^{-1} \sum_{a \in \xi_g} \frac{m_a}{\rho_a} A_a^{<n>} W_{ga} \right] \tag{6.16}$$

The modelled phenomenon can now be confined within the domain, while allowing the streaming of solar wind material in accordance with the physics of the simulation.

A logical extension to the work here is to further investigate the definition of the stiffness coefficient, η, under a range of different physical and domain conditions.

Chapter 7

Conclusion

7.1 Summary

As a measure of success, the construction of an algorithm and domain is complete. However, such is the complexity of the intended simulated media, that is is only complete at the simple test stage. A full 3-Dimensional run of the algorithm, at a resolution that would allow for meaningful results is simply out of the range of the machine power available for this project.

As well as completing the model, a number of other discoveries and conclusions have been made. First among them was the definition of the reflected formalism that ensures forces act equally and opposite (Chapter 4). The new formalism, while consistent with the physics was never-the-less very cumbersome to implement. In that same chapter, a numerical artefact was identified, described and explained. However, no adequate correction could be found.

Using this error as impetus, the other obvious but undefined error was identified as the discretisation error and quantified. It was determined to be a function of the true gradients being approximated, which presented a problem for a coronal model as the gradients can be very steep. To avoid this, potentially, very large error the SP algorithm was, with the now defined error, expanded upon to form the Corrected SP method. This proved to have a very low error approximately equal to the system noise. It was, however, yet another cumbersome (though nowhere near as demanding as the reflected formalism) addition to the implementation. As the program was becoming impractical to run, the short term aim became to find some way of slimming down the algorithm without reintroducing potentially large errors. This chapter culminated with the description of the novel CSP-Δh algorithm, which is approximately as hefty to implement as the CSP algorithm, with one important difference – the omission of all smoothing length gradient terms. This reduction causes the pre-corrected components of the algorithm run as efficiently as the original SP method. Factoring in the possible reduction in resolution (and hence, number of pseudoparticles) as the method is more accurate, the runtime is approximately equal to the raw SP algorithm.

Finally, with a completed algorithm, the geometry and parameters of the domain

were considered. The domain was defined as the space confined by the intersection two spheres and a square-based pyramid. The two spheres are concentric and are centred on the apex of the pyramid, which has a vertical height greater that the radius of the largest sphere. Also created was a novel periodicity routine and a parametrise edge boundary to bind coronal phenomena within the domain.

7.2 Future Work

There are a number of possible paths for the research to progress. A detailed stability analysis of the CSP-Δh method and an assessment of the interpolated-background boundary would, for example, be of great interest. However, primary among the possible tasks is the comparison of the CSP-Δh MHD algorithm with other established MHD codes. Unfortunately, most established codes are highly parallel and thus able to handle more complex applications with their increased memory and processing capacity. This discrepancy between the implementation of each algorithm is problematic, and as such the logical first step on route to result comparisons is the parallelisation of the algorithm.

An additional argument for the parallelisation of the algorithm is the capacity required to simulate solar coronal phenomena. Without that capacity, the fine-grained nature of the plasma physics would be obscured by the low resolution of the domain.

Appendix A

Extended Derivations

A.1 Equation of Motion

The definition of the a Lagrangian – of a closed system – is $L = E_T - E_V$, where E_T is the total kenetic energy and E_V is the total potential energy. For this closed system the Lagrangian can be written as

$$L = \int_\Omega \left(\frac{1}{2}\rho v^2 - \rho u - \frac{B^2}{2\mu_0} \right) \mathrm{d}\mathbf{r} \tag{A.1}$$

This can be transformed into SP form, since a mass element is given by $\rho \mathrm{d}\mathbf{r}$, and reduced to give A.2 by

$$\begin{aligned} L \approx [L]_{\mathrm{SP}} &= \sum_b \frac{m_b}{\rho_b} \left[\frac{1}{2}\rho_b v_b^2 - \rho_b u_b(\rho_b, s_b) - \frac{1}{2\mu_0} B_b^2 \right] \\ &= \sum_b m_b \left[\frac{1}{2} v_b^2 - u_b(\rho_b, s_b) - \frac{1}{2\mu_0} \frac{B_b^2}{\rho_b} \right] \end{aligned} \tag{A.2}$$

Given a Lagrangian, the equations of motion may be found by solving the Euler-Lagrange equation, ie.

$$\frac{\partial L}{\partial q_k} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \dot{q}_k} \right) = 0 \tag{A.3}$$

where q_k is the k^{th} generalised coordinate of the system. For a smoothed particle system, the generalised coordinates are $\mathbf{r}_a$ (for each pseudoparticle a). Thus the equation of motion (singular, as indicated by the number of generalised coordinates) may be found by solving

$$\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{r}_a} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{v}_a} \right) = 0 \tag{A.4}$$

where

$$\frac{\partial [L]_{\mathrm{SP}}}{\partial \mathbf{r}_a} = \sum_b m_b \left\{ \frac{1}{2} \frac{\partial v_b^2}{\partial \mathbf{r}_a} - \frac{\partial u_b}{\partial \mathbf{r}_a} - \frac{1}{2\mu_0} \frac{\partial}{\partial \mathbf{r}_a} \left(\frac{B_b^2}{\rho_b} \right) \right\} \tag{A.5}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\left[L\right]_{\mathrm{SP}}}{\partial\mathbf{v}_a}\right)=\sum_b m_b\left\{\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial u_b}{\partial\mathbf{v}_a}\right)\right.$$
$$\left.-\frac{1}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial\mathbf{v}_a}\left[\frac{B_b^2}{\rho_b}\right]\right)\right\} \tag{A.6}$$

Now, by substituting equation (A.5) and equation (A.6) into equation (A.4);

$$\begin{aligned}
0=&\frac{\partial\left[L\right]_{\mathrm{SP}}}{\partial\mathbf{r}_a}-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\left[L\right]_{\mathrm{SP}}}{\partial\mathbf{v}_a}\right)\\
=&\sum_b m_b\left\{\frac{1}{2}\frac{\partial v_b^2}{\partial\mathbf{r}_a}-\frac{\partial u_b}{\partial\mathbf{r}_a}-\frac{1}{2\mu_0}\frac{\partial}{\partial\mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right)\right\}\\
&-\sum_b m_b\left\{\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial u_b}{\partial\mathbf{v}_a}\right)\right.\\
&\left.-\frac{1}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial\mathbf{v}_a}\left[\frac{B_b^2}{\rho_b}\right]\right)\right\}\\
=&\sum_b m_b\left\{\left[\frac{1}{2}\frac{\partial v_b^2}{\partial\mathbf{r}_a}-\frac{\partial u_b}{\partial\mathbf{r}_a}-\frac{1}{2\mu_0}\frac{\partial}{\partial\mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right)\right]\right.\\
&\left.-\left[\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial u_b}{\partial\mathbf{v}_a}\right)-\frac{1}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial\mathbf{v}_a}\left[\frac{B_b^2}{\rho_b}\right]\right)\right]\right\}\\
=&\sum_b m_b\left\{\frac{1}{2}\frac{\partial v_b^2}{\partial\mathbf{r}_a}-\frac{\partial u_b}{\partial\mathbf{r}_a}-\frac{1}{2\mu_0}\frac{\partial}{\partial\mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right)-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)\right.\\
&\left.+\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial u_b}{\partial\mathbf{v}_a}\right)+\frac{1}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial\mathbf{v}_a}\left[\frac{B_b^2}{\rho_b}\right]\right)\right\}
\end{aligned}$$

Note, $u_a=u_a(\rho_a,s_a)$, $\rho_a=\rho_a(\mathbf{r}_a)$ and $\mathbf{B}_a=\mathbf{B}_a(\mathbf{r}_a)$. Therefore

$$\begin{aligned}
0=&\sum_b m_b\left\{\frac{1}{2}\frac{\partial \cancel{v_b^2}}{\partial\mathbf{r}_a}-\frac{\partial u_b}{\partial\mathbf{r}_a}-\frac{1}{2\mu_0}\frac{\partial}{\partial\mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right)-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)\right.\\
&\left.+\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \cancel{u_b}}{\partial\mathbf{v}_a}\right)+\frac{1}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial\cancel{\mathbf{v}_a}}\left[\frac{\cancel{B_b^2}}{\rho_b}\right]\right)\right\}\\
=&\sum_b m_b\left\{-\frac{\partial u_b}{\partial\mathbf{r}_a}-\frac{1}{2\mu_0}\frac{\partial}{\partial\mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right)-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)\right\}\\
=&-\frac{1}{2}\sum_b m_b\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial\mathbf{v}_a}\right)-\sum_b m_b\left[\frac{\partial u_b}{\partial\mathbf{r}_a}+\frac{1}{2\mu_0}\frac{\partial}{\partial\mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right)\right]
\end{aligned} \tag{A.7}$$

Now, consider the three following manipulations.

Given that

$$\frac{\partial A_b}{\partial A_a} = \begin{cases} 0 & \text{if} \quad a \neq b \\ \frac{\partial A_a}{\partial A_a} & \text{if} \quad a = b \end{cases} \tag{A.8}$$

The first summation can be reduced to

$$\begin{aligned} -\frac{1}{2}\sum_b m_b \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_b^2}{\partial \mathbf{v}_a}\right) &= -\frac{1}{2}\sum_b m_b \frac{\mathrm{d}}{\mathrm{d}t}\left(2\delta^{ab}\mathbf{v}_b\right) \\ &= -\sum_b m_b \delta^{ab}\frac{\mathrm{d}\mathbf{v}_b}{\mathrm{d}t} \\ &= -m_a \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} \end{aligned} \tag{A.9}$$

The first term of the second summation can be expanded by

$$\frac{\partial u_b}{\partial \mathbf{r}_a} = \left.\frac{\partial u_b}{\partial \rho_b}\right|_s \frac{\partial \rho_b}{\partial \mathbf{r}_a} \tag{A.10}$$

at constant entropy, s. The first law of thermodynamics dictates

$$\mathrm{d}U = m\mathrm{d}u = \mathrm{d}Q - \mathrm{d}W \tag{A.11}$$

where $\mathrm{d}Q = T\mathrm{d}s = 0$ is the heat added to the fluid body (zero as the entropy is constant), and $\mathrm{d}W = P\mathrm{d}V$ is the work done as the fluid diverges (or undergoes compression). Given that $V = \frac{m}{\rho}$, $\frac{\mathrm{d}V}{\mathrm{d}\rho} = -\frac{m}{\rho^2}$ such that by substitution equation (A.11) can be rewritten as

$$\mathrm{d}u|_s = -\frac{P}{\rho^2}\mathrm{d}\rho \tag{A.12}$$

and thus equation (A.10) becomes,

$$\frac{\partial u_b}{\partial \mathbf{r}_a} = \frac{P_b}{\rho_b^2}\frac{\partial \rho_b}{\partial \mathbf{r}_a} \tag{A.13}$$

The following can be greatly simplified using the conservative induction equation (**??**). Consider, $\frac{B_b^2}{\rho_b} = UV$, where $U = B_b^2$ and $V = \rho_b^{-1}$, such that

$$\begin{aligned} \frac{\partial}{\partial \mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right) &= \frac{\partial}{\partial \mathbf{r}_a}(UV) \\ &= U\frac{\partial V}{\partial \mathbf{r}_a} + V\frac{\partial U}{\partial \mathbf{r}_a} \\ &= U\frac{\partial V}{\partial \rho_b}\frac{\partial \rho_b}{\partial \mathbf{r}_a} + V\frac{\partial U}{\partial \mathbf{B}_b}\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a} \end{aligned}$$

by substitution therefore,

$$\frac{\partial}{\partial \mathbf{r}_a}\left(\frac{B_b^2}{\rho_b}\right) = B_b^2\frac{\partial(\rho_b^{-1})}{\partial \rho_b}\frac{\partial \rho_b}{\partial \mathbf{r}_a} + \frac{1}{\rho_b}\frac{\partial(B_b^2)}{\partial \mathbf{B}_b}\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a} - \frac{B_b^2}{\rho_b^2}\frac{\partial \rho_b}{\partial \mathbf{r}_a} + \frac{2\mathbf{B}_b}{\rho_b}\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a} \tag{A.14}$$

Thus, by substituting equation (A.9), equation (A.13) and equation (A.14) into equation (A.7),

$$0 = -m_a\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} - \sum_b m_b\left[\frac{P_b}{\rho_b^2}\frac{\partial \rho_b}{\partial \mathbf{r}_a} - \frac{B_b^2}{2\mu_0\rho_b^2}\frac{\partial \rho_b}{\partial \mathbf{r}_a} + \frac{\mathbf{B}_b}{\mu_0\rho_b}\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a}\right] \tag{A.15}$$

$\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a}$ can be found by

$$\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a} = \frac{\delta \mathbf{B}_b}{\delta \mathbf{r}_a} = \frac{\delta \mathbf{B}_b}{\delta \mathbf{r}_a}\frac{\delta t}{\delta t} = \frac{\delta t}{\delta \mathbf{r}_a}\frac{\delta \mathbf{B}_b}{\delta t} = \frac{\delta t}{\delta \mathbf{r}_a}\frac{\mathrm{d}\mathbf{B}_b}{\mathrm{d}t} \tag{A.16}$$

Substituting in the SP induction equation (2.118),

$$\begin{aligned}
\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a} =& \frac{\delta t}{\delta \mathbf{r}_a}\left\{\frac{1}{\Psi_b}\sum_{\substack{c\in\xi_b\\ c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left[\mathbf{B}_b\left(\mathbf{v}_{bc}\cdot\nabla_b W_{bc}\right) - \mathbf{v}_{bc}\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\right]\right\}\\
=& \frac{1}{\Psi_b}\sum_{\substack{c\in\xi_b\\ c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left[\mathbf{B}_b\left(\frac{\delta t}{\delta \mathbf{r}_a}\mathbf{v}_{bc}\cdot\nabla_b W_{bc}\right) - \frac{\delta t}{\delta \mathbf{r}_a}\mathbf{v}_{bc}\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\right]\\
=& \frac{1}{\Psi_b}\sum_{\substack{c\in\xi_b\\ c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left[\mathbf{B}_b\left(\frac{\delta t}{\delta \mathbf{r}_a}\frac{\partial \mathbf{r}_{bc}}{\partial t}\cdot\nabla_b W_{bc}\right) - \frac{\delta t}{\delta \mathbf{r}_a}\frac{\partial \mathbf{r}_{bc}}{\partial t}\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\right]\\
=& \frac{1}{\Psi_b}\sum_{\substack{c\in\xi_b\\ c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left[\mathbf{B}_b\left(\frac{\delta t}{\delta \mathbf{r}_a}\frac{\delta \mathbf{r}_{bc}}{\delta t}\cdot\nabla_b W_{bc}\right) - \frac{\delta t}{\delta \mathbf{r}_a}\frac{\delta \mathbf{r}_{bc}}{\delta t}\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\right]\\
=& \frac{1}{\Psi_b}\sum_{\substack{c\in\xi_b\\ c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left[\mathbf{B}_b\left(\left(\frac{\delta \mathbf{r}_b}{\delta \mathbf{r}_a} - \frac{\delta \mathbf{r}_c}{\delta \mathbf{r}_a}\right)\cdot\nabla_b W_{bc}\right)\right.\\
&\left. - \left(\frac{\delta \mathbf{r}_b}{\delta \mathbf{r}_a} - \frac{\delta \mathbf{r}_c}{\delta \mathbf{r}_a}\right)\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\right]\\
=& \frac{1}{\Psi_b}\sum_{\substack{c\in\xi_b\\ c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left[\mathbf{B}_b\left(\left(\delta^{ab} - \delta^{ac}\right)\nabla_b W_{bc}\right) - \left(\delta^{ab} - \delta^{ac}\right)\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\right]
\end{aligned}$$

$$\frac{\partial \mathbf{B}_b}{\partial \mathbf{r}_a} = \frac{1}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \mathbf{B}_b \left(\left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \right) - \frac{1}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \left(\mathbf{B}_b \cdot \nabla_b W_{bc} \right) \tag{A.17}$$

And, $\frac{\partial \rho_b}{\partial \mathbf{r}_a}$ can be found in a similar way.

$$\frac{\partial \rho_b}{\partial \mathbf{r}_a} = \frac{\delta \rho_b}{\delta \mathbf{r}_a} = \frac{\delta \rho_b}{\delta \mathbf{r}_a} \frac{\delta t}{\delta t} = \frac{\delta t}{\delta \mathbf{r}_a} \frac{\delta \rho_b}{\delta t} = \frac{\delta t}{\delta \mathbf{r}_a} \frac{\mathrm{d} \rho_b}{\mathrm{d} t} \tag{A.18}$$

This time, by substitution of the continuity equation (2.112)

$$\begin{aligned}
\frac{\partial \rho_b}{\partial \mathbf{r}_a} &= \frac{\delta t}{\delta \mathbf{r}_a} \left\{ \frac{\rho_b}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\mathbf{v}_b - \mathbf{v}_c \right) \cdot \nabla_b W_{bc} \right\} \\
&= \frac{\rho_b}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \frac{\delta t}{\delta \mathbf{r}_a} \left(\mathbf{v}_b - \mathbf{v}_c \right) \cdot \nabla_b W_{bc} \\
&= \frac{\rho_b}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \frac{\delta t}{\delta \mathbf{r}_a} \left(\frac{\partial \mathbf{r}_b}{\partial t} - \frac{\partial \mathbf{r}_c}{\partial t} \right) \cdot \nabla_b W_{bc} \\
&= \frac{\rho_b}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \frac{\delta t}{\delta \mathbf{r}_a} \left(\frac{\delta \mathbf{r}_b}{\delta t} - \frac{\delta \mathbf{r}_c}{\delta t} \right) \cdot \nabla_b W_{bc} \\
&= \frac{\rho_b}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\frac{\delta \mathbf{r}_b}{\delta \mathbf{r}_a} - \frac{\delta \mathbf{r}_c}{\delta \mathbf{r}_a} \right) \cdot \nabla_b W_{bc} \\
&= \frac{\rho_b}{\Psi_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc}
\end{aligned} \tag{A.19}$$

Alternaitvely, $\frac{\partial \rho_b}{\partial \mathbf{r}_a}$ can be found analytically by

$$\frac{\partial \rho_b}{\partial \mathbf{r}_a} = \frac{\partial}{\partial \mathbf{r}_a} \left(\sum_{\substack{c \in \xi_b \\ c \neq b}} m_c W_{bc} \right) = \sum_{\substack{c \in \xi_b \\ c \neq b}} m_c \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \tag{A.20}$$

However, this form is inconsistant with the non-conservative induction equation (2.118).

Now, by substituting equation (A.17) and equation (A.19) into equation (A.15),

and rearranging slightly gives

$$
\begin{aligned}
0 = -m_a \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} - \sum_b m_b \Bigg[& \frac{P_b}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \\
& - \frac{B_b^2}{2\mu_0 \Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \\
& + \frac{\mathbf{B}_b}{\mu_0 \Psi_b \rho_b} \cdot \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \mathbf{B}_b \left(\left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \right) \\
& - \frac{\mathbf{B}_b}{\mu_0 \Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \left(\mathbf{B}_b \cdot \nabla_b W_{bc} \right) \Bigg]
\end{aligned}
$$

Further rearrangement yeilds,

$$
\begin{aligned}
m_a \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = & - \sum_b \frac{m_b P_b}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \\
& + \frac{1}{2\mu_0} \sum_b \frac{m_b B_b^2}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \\
& - \frac{1}{\mu_0} \sum_b \frac{m_b \mathbf{B}_b}{\Psi_b \rho_b} \cdot \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \mathbf{B}_b \left(\delta^{ab} - \delta^{ac} \right) \nabla_b W_{bc} \\
& + \frac{1}{\mu_0} \sum_b \frac{m_b \mathbf{B}_b}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \left(\delta^{ab} - \delta^{ac} \right) \left(\mathbf{B}_b \cdot \nabla_b W_{bc} \right) \\
m_a \frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = & - \sum_b \frac{m_b P_b}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \delta^{ab} \nabla_b W_{bc} \\
& + \sum_b \frac{m_b P_b}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \delta^{ac} \nabla_b W_{bc} \\
& + \frac{1}{2\mu_0} \sum_b \frac{m_b B_b^2}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \delta^{ab} \nabla_b W_{bc} \\
& - \frac{1}{2\mu_0} \sum_b \frac{m_b B_b^2}{\Psi_b \rho_b} \sum_{\substack{c \in \xi_b \\ c \neq b}} \frac{m_c \Psi_c}{\rho_c} \delta^{ac} \nabla_b W_{bc} \cdots
\end{aligned}
$$

$$
\begin{aligned}
\cdots &- \frac{1}{\mu_0}\sum_b \frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\cdot\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\mathbf{B}_b\delta^{ab}\nabla_b W_{bc}\\
&+ \frac{1}{\mu_0}\sum_b \frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\cdot\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\mathbf{B}_b\delta^{ac}\nabla_b W_{bc}\\
&+ \frac{1}{\mu_0}\sum_b \frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\delta^{ab}\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\\
&- \frac{1}{\mu_0}\sum_b \frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\delta^{ac}\left(\mathbf{B}_b\cdot\nabla_b W_{bc}\right)\\
m_a\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} =&- \frac{m_aP_a}{\Psi_a\rho_a}\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\nabla_a W_{ac} + \frac{m_a\Psi_a}{\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_bP_b}{\Psi_b\rho_b}\nabla_b W_{ba}\\
&+ \frac{m_aB_a^2}{2\mu_0\Psi_a\rho_a}\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\nabla_a W_{ac} - \frac{m_a\Psi_a}{2\mu_0\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_bB_b^2}{\Psi_b\rho_b}\nabla_b W_{ba}\\
&- \frac{m_a\mathbf{B}_a}{\mu_0\Psi_a\rho_a}\cdot\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\mathbf{B}_a\nabla_a W_{ac} + \frac{m_a\Psi_a}{\mu_0\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\cdot\mathbf{B}_b\nabla_b W_{ba}\\
&+ \frac{m_a\mathbf{B}_a}{\mu_0\Psi_a\rho_a}\sum_{\substack{c\in\xi_b\\c\neq b}}\frac{m_c\Psi_c}{\rho_c}\left(\mathbf{B}_a\cdot\nabla_a W_{ac}\right) - \frac{m_a\Psi_a}{\mu_0\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\left(\mathbf{B}_b\cdot\nabla_b W_{ba}\right)\\
m_a\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} =&- \frac{m_aP_a}{\Psi_a\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\Psi_b}{\rho_b}\nabla_a W_{ab} - \frac{m_a\Psi_a}{\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_bP_b}{\Psi_b\rho_b}\nabla_a W_{ab}\\
&+ \frac{m_aB_a^2}{2\mu_0\Psi_a\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\Psi_b}{\rho_b}\nabla_a W_{ab} + \frac{m_a\Psi_a}{2\mu_0\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_bB_b^2}{\Psi_b\rho_b}\nabla_a W_{ab}\\
&- \frac{m_a\mathbf{B}_a}{\mu_0\Psi_a\rho_a}\cdot\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\Psi_b}{\rho_b}\mathbf{B}_a\nabla_a W_{ab} - \frac{m_a\Psi_a}{\mu_0\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\cdot\mathbf{B}_b\nabla_a W_{ab}\\
&+ \frac{m_a\mathbf{B}_a}{\mu_0\Psi_a\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\Psi_b}{\rho_b}\left(\mathbf{B}_a\cdot\nabla_a W_{ab}\right) + \frac{m_a\Psi_a}{\mu_0\rho_a}\sum_{\substack{b\in\xi_a\\b\neq a}}\frac{m_b\mathbf{B}_b}{\Psi_b\rho_b}\left(\mathbf{B}_b\cdot\nabla_a W_{ab}\right)
\end{aligned}
$$

Once the summations have been manipulated such that the Kronecker delta's are

no longer present, the equation can be rearranged to give the acceleration equation,

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} =& \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_a}\left\{-\frac{\Psi_b P_a}{\Psi_a} - \frac{\Psi_a P_b}{\Psi_b} + \frac{\Psi_b B_a^2}{2\mu_0\Psi_a} + \frac{\Psi_a B_b^2}{2\mu_0\Psi_b}\right.\\
&\left. -\frac{\Psi_b \mathbf{B}_a\cdot\mathbf{B}_a}{\mu_0\Psi_a} - \frac{\Psi_a \mathbf{B}_b\cdot\mathbf{B}_b}{\mu_0\Psi_b} + \frac{\Psi_b \mathbf{B}_a\mathbf{B}_a}{\mu_0\Psi_a} + \frac{\Psi_a \mathbf{B}_b\mathbf{B}_b}{\mu_0\Psi_b}\right\}\nabla_a W_{ab}\\
\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} =& \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_a}\left\{-\frac{\Psi_b P_a}{\Psi_a} - \frac{\Psi_a P_b}{\Psi_b} + \frac{\Psi_b B_a^2}{2\mu_0\Psi_a} + \frac{\Psi_a B_b^2}{2\mu_0\Psi_b}\right.\\
&\left. -\frac{\Psi_b v B_a^2}{\mu_0\Psi_a} - \frac{\Psi_a B_b^2}{\mu_0\Psi_b} + \frac{\Psi_b \mathbf{B}_a\mathbf{B}_a}{\mu_0\Psi_a} + \frac{\Psi_a \mathbf{B}_b\mathbf{B}_b}{\mu_0\Psi_b}\right\}\nabla_a W_{ab}\\
\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} =& \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_a}\left\{-\frac{\Psi_b P_a}{\Psi_a} - \frac{\Psi_a P_b}{\Psi_b} - \frac{\Psi_b B_a^2}{2\mu_0\Psi_a}\right.\\
&\left. -\frac{\Psi_a B_b^2}{2\mu_0\Psi_b} + \frac{\Psi_b \mathbf{B}_a\mathbf{B}_a}{\mu_0\Psi_a} + \frac{\Psi_a \mathbf{B}_b\mathbf{B}_b}{\mu_0\Psi_b}\right\}\nabla_a W_{ab}
\end{aligned}$$

Now, given $S_a^{ij} = -P_a\delta^{ij} + \frac{1}{\mu_0}\left(B_a^i B_a^j - \frac{B_a^2}{2}\delta^{ij}\right)$ the acceleration equation can be simplified to

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\left[\frac{\Psi_b}{\Psi_a}\mathbf{S}_a + \frac{\Psi_a}{\Psi_b}\mathbf{S}_b\right]\nabla_a W_{ab} \tag{A.21}$$

So, if $\Psi = 1$,

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \frac{1}{\rho_a}\sum_{\substack{b\in\xi_a\\ b\neq a}} \frac{m_b}{\rho_b}\left[\mathbf{S}_a + \mathbf{S}_b\right]\nabla_a W_{ab} \tag{A.22}$$

or, if $\Psi = \rho$

$$\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} = \sum_{\substack{b\in\xi_a\\ b\neq a}} m_b\left[\frac{\mathbf{S}_a}{\rho_a^2} + \frac{\mathbf{S}_b}{\rho_b^2}\right]\nabla_a W_{ab} \tag{A.23}$$

A.2 Total Energy Equation

The Hamiltonian represents the conserved total energy, E, and is given by

$$H = E = \sum_a \mathbf{v}_a\cdot\frac{\partial L}{\partial \mathbf{v}_a} - L \tag{A.24}$$

$[E]_{\mathrm{SP}}$ can be found by substituting equation (A.2) into the equation above, and

is given by equation equation (A.25).

$$
\begin{aligned}
[E]_{\mathrm{SP}} &= \sum_a \mathbf{v}_a \cdot \frac{\partial}{\partial \mathbf{v}_a}\left[\sum_b m_b\left(\frac{1}{2}v_b^2 - u_b - \frac{1}{2\mu_0}\frac{B_b^2}{\rho_b}\right)\right] \\
&\quad - \sum_b m_b\left(\frac{1}{2}v_b^2 - u_b - \frac{1}{2\mu_0}\frac{B_b^2}{\rho_b}\right) \\
&= \sum_a\left[\frac{1}{2}\sum_b m_b\mathbf{v}_a\cdot\frac{\partial v_b^2}{\partial \mathbf{v}_a} - \sum_b m_b\mathbf{v}_a\cdot\frac{\partial}{\partial\mathbf{v}_a}\left(u_b + \frac{1}{2\mu_0}\frac{B_b^2}{\rho_b}\right)\right] \\
&\quad - \sum_b m_b\left(\frac{1}{2}v_b^2 - u_b - \frac{1}{2\mu_0}\frac{B_b^2}{\rho_b}\right) \\
&= \sum_a\left[\frac{1}{2}\sum_b m_b\mathbf{v}_a\cdot\left(2\mathbf{v}_b\delta^{ab}\right) - \sum_b m_b\mathbf{v}_a\cdot(0)\right] \\
&\quad - \sum_b m_b\left(\frac{1}{2}v_b^2 - u_b - \frac{1}{2\mu_0}\frac{B_b^2}{\rho_b}\right) \\
&= \sum_a m_a v_a^2 - \sum_a m_a\left(\frac{1}{2}v_a^2 - u_a - \frac{1}{2\mu_0}\frac{B_a^2}{\rho_a}\right) \\
&= \sum_a m_a\left[v_a^2 - \left(\frac{1}{2}v_a^2 - u_a - \frac{1}{2\mu_0}\frac{B_a^2}{\rho_a}\right)\right]
\end{aligned}
$$

Therefore

$$
[E]_{\mathrm{SP}} = \sum_a m_a\left(\frac{1}{2}v_a^2 + u_a + \frac{1}{2\mu_0}\frac{B_a^2}{\rho_a}\right) \tag{A.25}
$$

The comoving time derivative is given by

$$
\begin{aligned}
\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} &= \sum_a m_a\left\{\frac{1}{2}\frac{\mathrm{d}(v_a^2)}{\mathrm{d}t} + \frac{\mathrm{d}u_a}{\mathrm{d}t} + \frac{1}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{B_a^2}{\rho_a}\right)\right\} \\
&= \sum_a m_a\left\{\frac{1}{2}\frac{\mathrm{d}(v_a^2)}{\mathrm{d}\mathbf{v}_a}\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} + \frac{\mathrm{d}u_a}{\mathrm{d}\rho_a}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} + \frac{B^2}{2\mu_0}\frac{\mathrm{d}}{\mathrm{d}\rho_a}\left(\frac{1}{\rho_a}\right)\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right. \\
&\qquad \left. + \frac{1}{2\mu_0\rho_a}\frac{\mathrm{d}(B_a^2)}{\mathrm{d}\mathbf{B}_a}\frac{\mathrm{d}\mathbf{B}_a}{\mathrm{d}t}\right\} \\
&= \sum_a m_a\left\{\mathbf{v}_a\frac{\mathrm{d}\mathbf{v}_a}{\mathrm{d}t} + \frac{P_a}{\rho_a^2}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} + \frac{\mathbf{B}_a}{\mu_0\rho_a}\frac{\mathrm{d}\mathbf{B}_a}{\mathrm{d}t} - \frac{B_a^2}{2\mu_0\rho_a^2}\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right\}
\end{aligned}
$$

By substitution of pre-existing SP dynamic equations,

$$\begin{aligned}\frac{\mathrm{d}\left[E\right]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a m_a \Bigg\{ & \frac{\mathbf{v}_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a}\mathbf{S}_a + \frac{\Psi_a}{\Psi_b}\mathbf{S}_b\right] \nabla_a W_{ab} \\ + & \frac{P_a}{\Psi_a \rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} \mathbf{v}_{ab} \cdot \nabla_a W_{ab} - \frac{B_a^2}{2\mu_0 \Psi_a \rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} \mathbf{v}_{ab} \cdot \nabla_a W_{ab} \\ + & \frac{\mathbf{B}_a}{\mu_0 \Psi_a \rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} \left[\mathbf{B}_a \left(\mathbf{v}_{ab} \cdot \nabla_a W_{ab}\right) - \mathbf{v}_{ab} \left(\mathbf{B}_a \cdot \nabla_a W_{ab}\right)\right] \Bigg\}\end{aligned} \tag{A.26}$$

or in indice notation,

$$\begin{aligned}\frac{\mathrm{d}\left[E\right]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a m_a \Bigg\{ & \frac{v_a^i}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} S_a^{ij} + \frac{\Psi_a}{\Psi_b} S_b^{ij}\right] \nabla_a^j W_{ab} \\ + & \frac{P_a}{\Psi_a \rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} - \frac{B_a^2}{2\mu_0 \Psi_a \rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} \\ + & \frac{B_a^i}{\mu_0 \Psi_a \rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} \left[B_a^i \left(v_{ab}^j \nabla_a^j W_{ab}\right) - v_{ab}^i \left(B_a^j \nabla_a^j W_{ab}\right)\right] \Bigg\}\end{aligned} \tag{A.27}$$

and by rearranging the final three terms

$$\begin{aligned}\frac{\mathrm{d}\left[E\right]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \Bigg\{ & v_a^i \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} S_a^{ij} + \frac{\Psi_a}{\Psi_b} S_b^{ij}\right] \nabla_a^j W_{ab} \\ + & \frac{P_a}{\Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} - \frac{B_a^2}{2\mu_0 \Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} \\ - & \frac{B_a^i}{\mu_0 \Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^i B_a^j \nabla_a^j W_{ab} + \frac{B_a^i}{\mu_0 \Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} B_a^i v_{ab}^j \nabla_a^j W_{ab} \Bigg\}\end{aligned} \tag{A.28}$$

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \Bigg\{ v_a^i \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} S_a^{ij} + \frac{\Psi_a}{\Psi_b} S_b^{ij} \right] \nabla_a^j W_{ab} + \frac{P_a}{\Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} + \frac{B_a^2}{2\mu_0 \Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^j \nabla_a^j W_{ab} - \frac{B_a^i}{\mu_0 \Psi_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b \Psi_b}{\rho_b} v_{ab}^i B_a^j \nabla_a^j W_{ab} \Bigg\} \tag{A.29}$$

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left\{ \frac{\Psi_b}{\Psi_a} S_a^{ij} v_a^i + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i + \frac{\Psi_b}{\Psi_a} P_a v_{ab}^i \delta^{ij} - \frac{\Psi_b B_a^i B_a^j}{\mu_0 \Psi_a} v_{ab}^i + \frac{\Psi_b B_a^2}{2\mu_0 \Psi_a} v_{ab}^i \delta^{ij} \right\} \nabla_a^j W_{ab} \tag{A.30}$$

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left\{ \frac{\Psi_b}{\Psi_a} \left(P_a v_{ab}^i \delta^{ij} - \frac{B_a^i B_j^j}{\mu_0} v_{ab}^i + \frac{B_a^2}{2\mu_0} v_{ab}^i \delta^{ij} + S_a^{ij} v_a^i \right) + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right\} \nabla_a^j W_{ab} \tag{A.31}$$

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left\{ \frac{\Psi_b}{\Psi_a} \left[\left(P_a \delta^{ij} - \frac{B_a^i B_j^j}{\mu_0} + \frac{B_a^2}{2\mu_0} \delta^{ij} \right) v_{ab}^i + S_a^{ij} v_a^i \right] + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right\} \nabla_a^j W_{ab} \tag{A.32}$$

By substitution of $S^{ij} = -P\delta^{ij} - \frac{B^2}{2\mu_0}\delta^{ij} + \frac{1}{\mu_0} B^i B^j$,

$$\frac{\mathrm{d}\,[E]_{\mathrm{SP}}}{\mathrm{d}t} = \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left\{ \frac{\Psi_b}{\Psi_a} \left[-S_a^{ij} v_{ab}^i + S_a^{ij} v_a^i \right] + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right\} \nabla_a^j W_{ab} \tag{A.33}$$

$$= \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left\{ \frac{\Psi_b}{\Psi_a} \left[-S_a^{ij} v_a^i + S_a^{ij} v_b^i + S_a^{ij} v_a^i \right] + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right\} \nabla_a^j W_{ab} \tag{A.34}$$

$$= \sum_a \frac{m_a}{\rho_a} \sum_{\substack{b\in\xi_a \\ b\neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} S_a^{ij} v_b^i + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right] \nabla_a^j W_{ab} \tag{A.35}$$

Thus the total energy of some particle a;

$$\frac{\mathrm{d}E_a}{\mathrm{d}t} = \frac{m_a}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} S_a^{ij} v_b^i + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right] \nabla_a^j W_{ab} \tag{A.36}$$

Therefore the total energy per unit mass is;

$$\frac{\mathrm{d}e_a}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[\frac{\Psi_b}{\Psi_a} S_a^{ij} v_b^i + \frac{\Psi_a}{\Psi_b} S_b^{ij} v_a^i \right] \nabla_a^j W_{ab} \tag{A.37}$$

So, if $\Psi = 1$,

$$\frac{\mathrm{d}e_a}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[S_a^{ij} v_b^i + S_b^{ij} v_a^i \right] \nabla_a^j W_{ab} \tag{A.38}$$

or, if $\Psi = \rho$

$$\frac{\mathrm{d}e_a}{\mathrm{d}t} = \frac{1}{\rho_a} \sum_{\substack{b \in \xi_a \\ b \neq a}} \frac{m_b}{\rho_b} \left[\frac{S_a^{ij} v_b^i}{\rho_a^2} + \frac{S_b^{ij} v_a^i}{\rho_b^2} \right] \nabla_a^j W_{ab} \tag{A.39}$$

Appendix B

Additional Figures

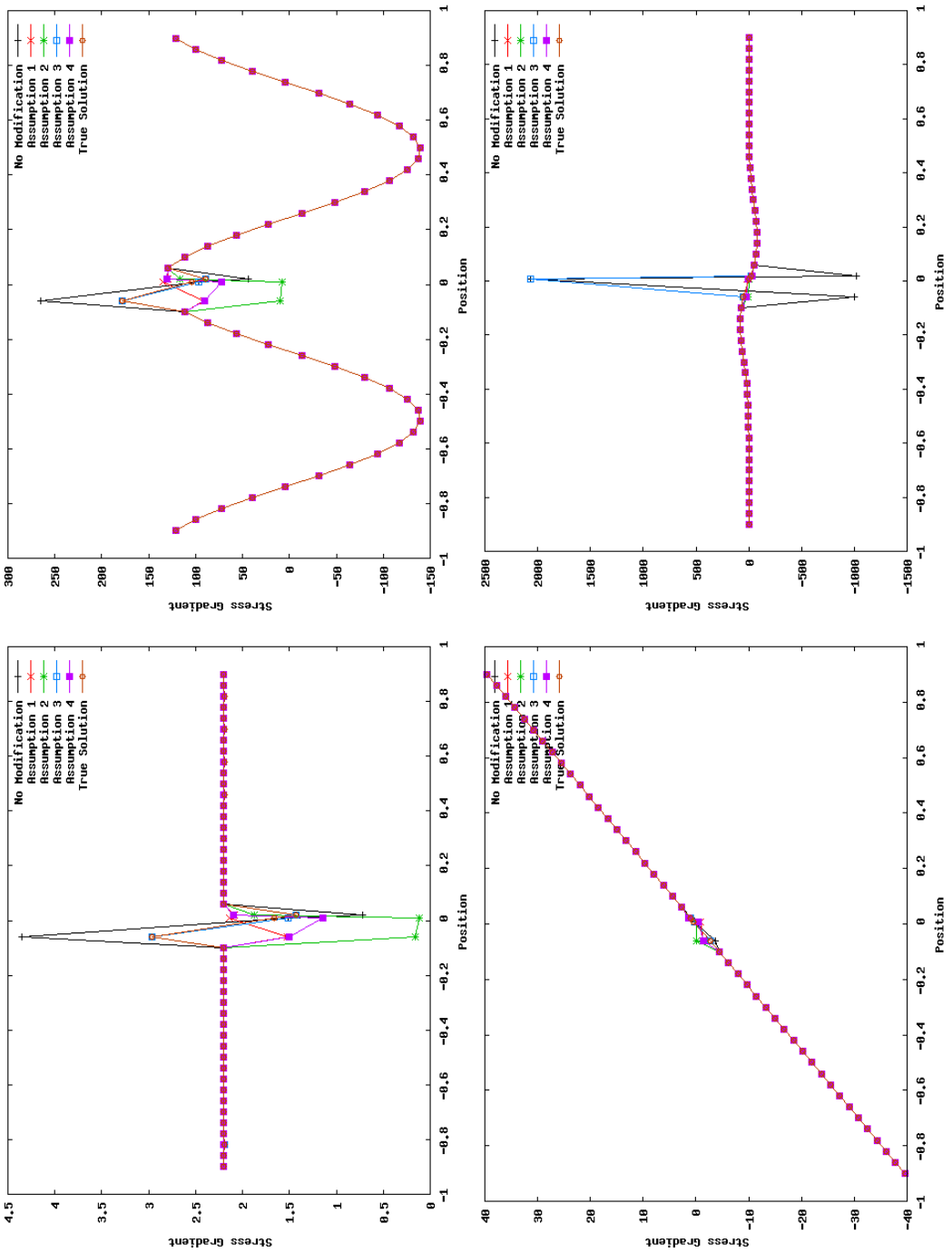

Figure B.1: Plots showing the stress gradient curves as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, the plots show stress gradients for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 2, where the pseudoparticles are regularly spaced, excluding $a = N_p/2$ which is offset by $0 < \delta_p < \Delta_p$.

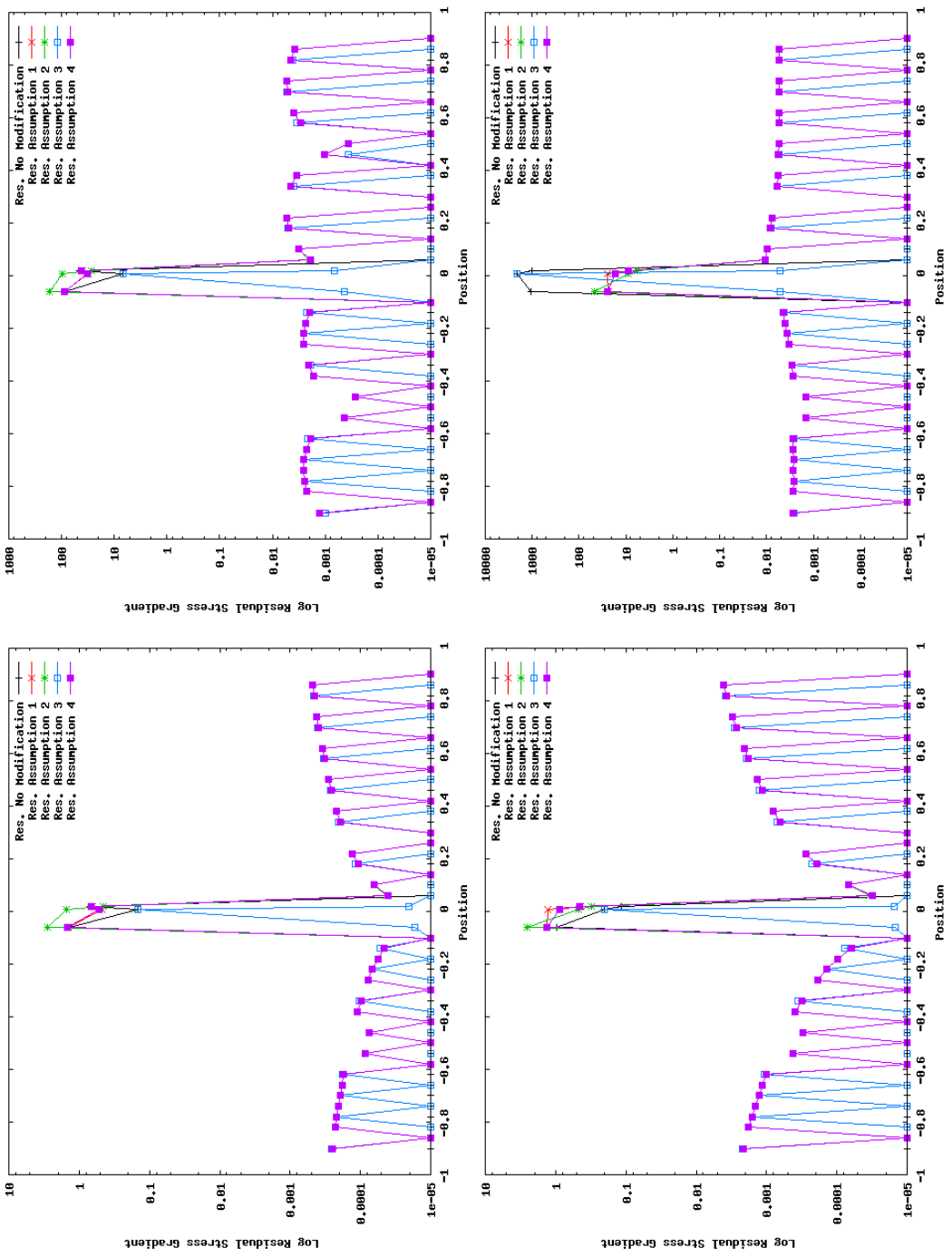

Figure B.2: Plots showing the absolute residual stress gradients constructed by subtracting the the true solution, as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, the plots show the residual stress for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 2, where the pseudoparticles are regularly spaced, excluding $a = N_p/2$ which is offset by $0 < \delta_p < \Delta_p$.

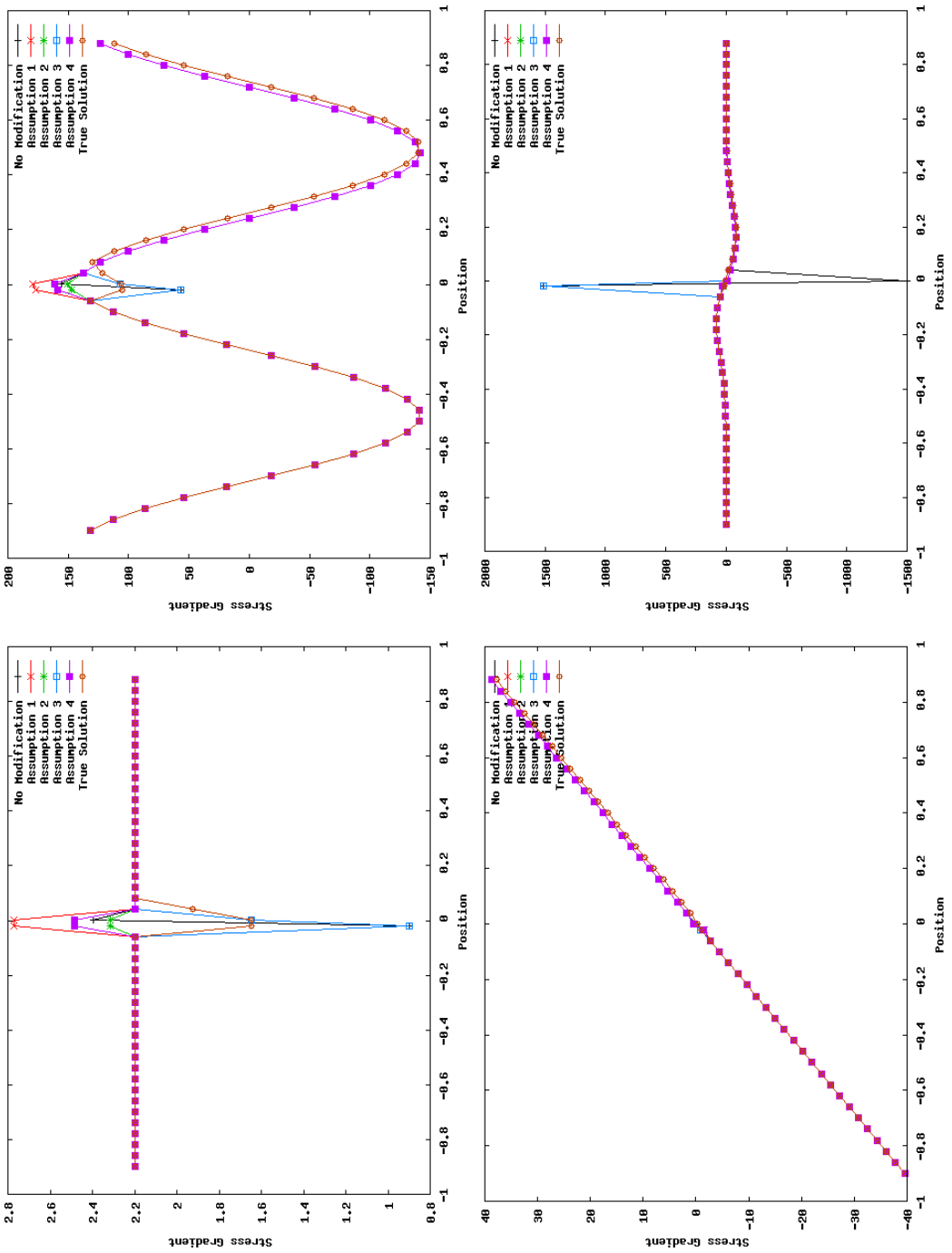

Figure B.3: Plots showing the stress gradient curves as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, the plots show stress gradients and residuals for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 3, where the pseudoparticles are regularly spaced, excluding the pair $a = N_p/2$ and $b = a+1$ where are closer together ($|\mathbf{r}_{ab}| < \Delta_p$).

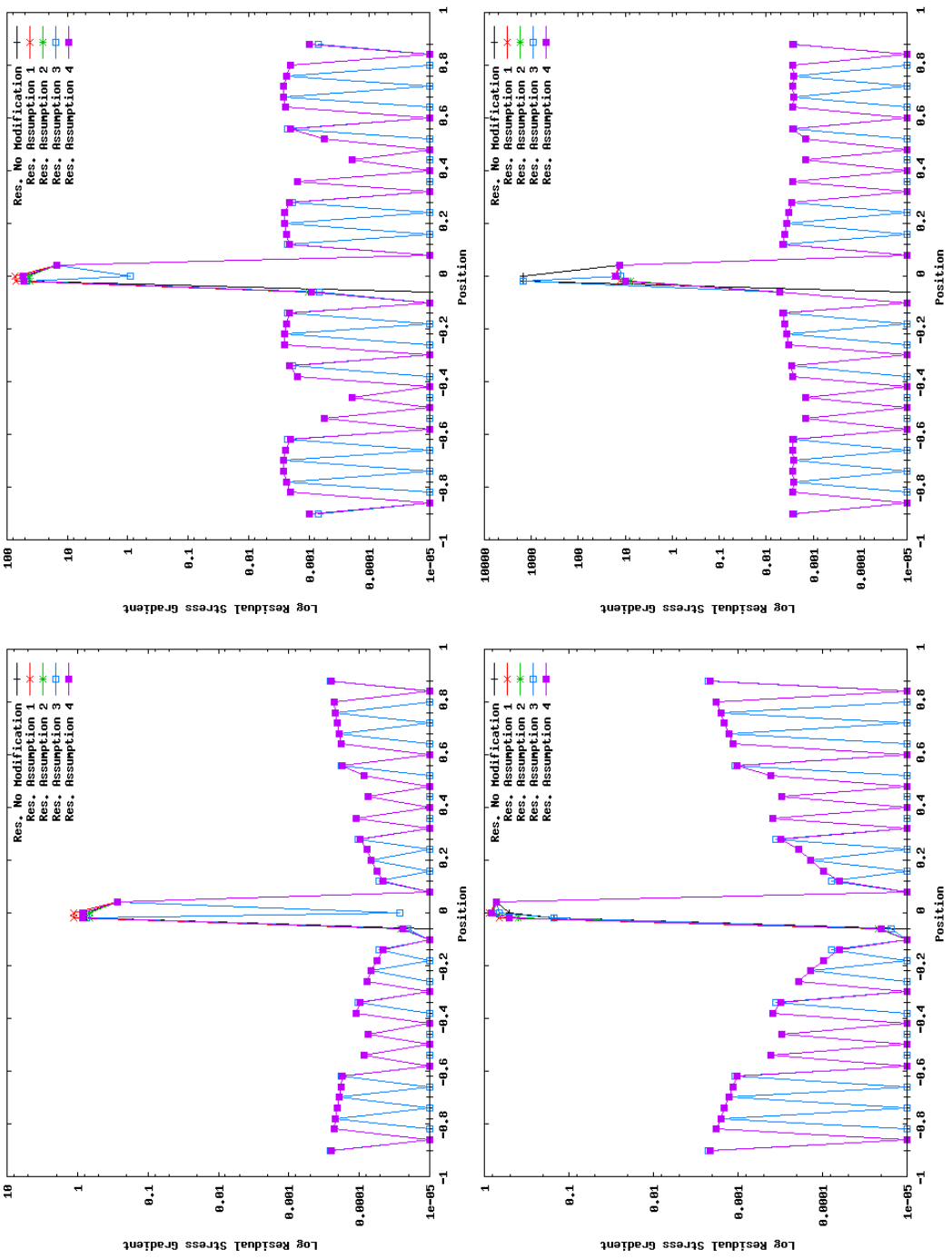

Figure B.4: Plots showing the absolute residual stress gradients constructed by subtracting the the true solution, as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, the plots show the residual stress for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 3, where the pseudoparticles are regularly spaced, excluding the pair $a = N_p/2$ and $b = a + 1$ where are closer together ($|\mathbf{r}_{ab}| < \Delta_p$).

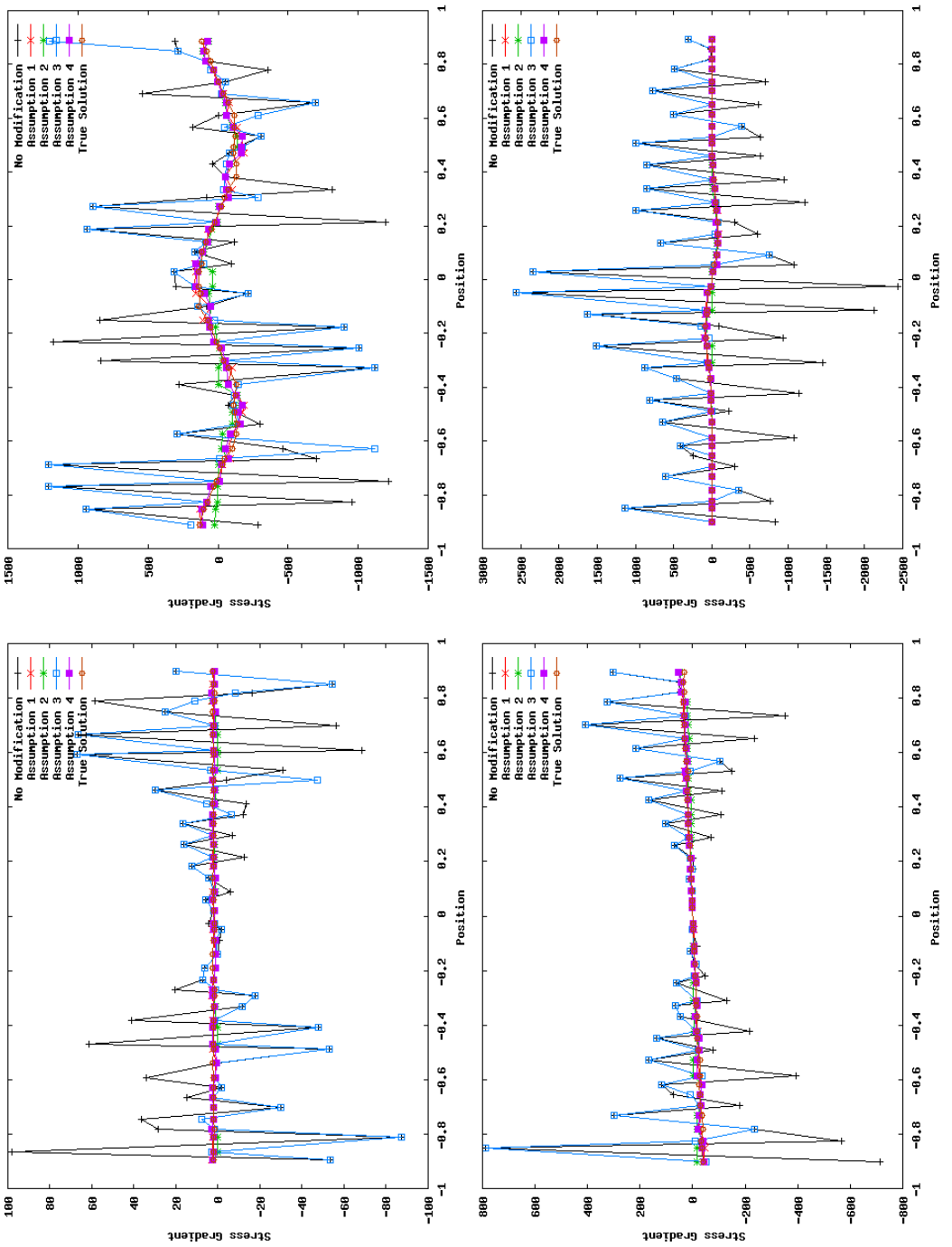

Figure B.5: Plots showing the stress gradient curves as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, the plots show stress gradients for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 4, where the pseudoparticles are regularly spaced, then offset by $\pm\delta_p R/2$ where $0 < R < 1$ is some random number.

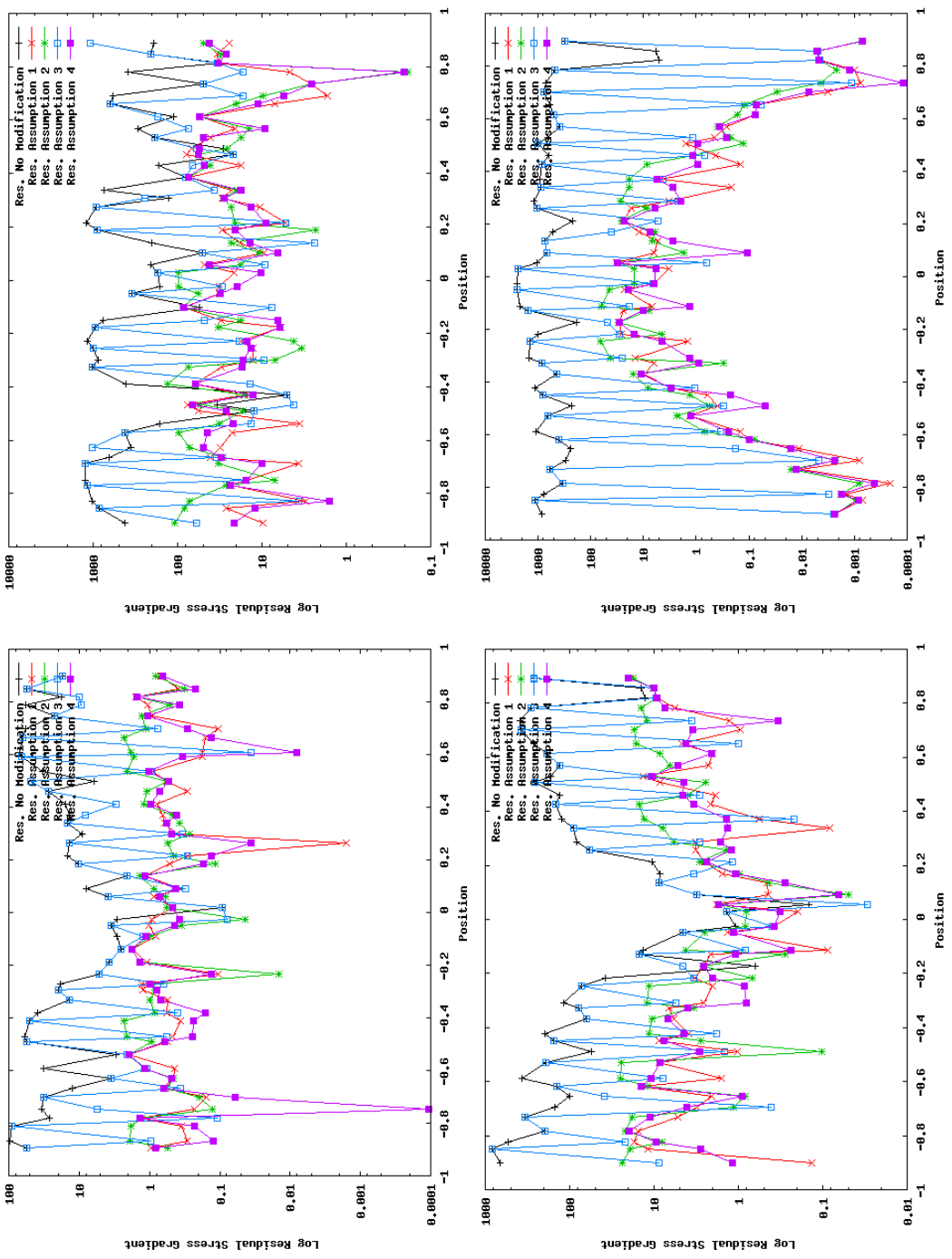

Figure B.6: Plots showing the absolute residual stress gradients constructed by subtracting the the true solution, as calculated by SP approximation with the correct particle configuration at the same spatial resolution. Using English directionality, the plots show the residual stress for linear, sinusoidal, quadratic polynomial and Gaussian stress. All are calculated with configuration 4, where the pseudoparticles are regularly spaced, then offset by $\pm\delta_p R/2$ where $0 < R < 1$ is some random number.

Bibliography

Birdsall, C. K. and Fuss, D. (1968). Cloud-in-cell computer experiments in 2 and 3 dimensions. *Bulletin of the American Physical Society*, 13(12):1746–&.

Bonet, J. and Lok, T. (1999). Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in applied mechanics and engineering*, 180(1):97–115.

Bonet, J. and Rodríguez-Paz, M. (2005). Hamiltonian formulation of the variable-h sph equations. *Journal of Computational Physics*, 209(2):541–558.

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

Børve, S., Omang, M., and Trulsen, J. (2004). Two-dimensional mhd smoothed particle hydrodynamics stability analysis. *The Astrophysical Journal Supplement Series*, 153:447.

Børve, S., Omang, M., and Trulsen, J. (2006). Multidimensional mhd shock tests of regularized smoothed particle hydrodynamics. *The Astrophysical Journal*, 652:1306.

Brookshaw, L. (1985). A method of calculating radiative heat diffusion in particle simulations. In *Proceedings of the Astronomical Society of Australia*, volume 6, pages 207–210.

Buchlin, E. (2007). Intermittent heating of the solar corona by mhd turbulence. *Nonlinear Processes in Geophysics*, 14:649–654.

Chen, P., Wu, S., Shibata, K., and Fang, C. (2002). Evidence of eit and moreton waves in numerical simulations. *The Astrophysical Journal Letters*, 572:L99.

Chow, E. and Monaghan, J. J. (1997). Ultrarelativistic sph. *Journal of Computational Physics*, 134:296–305.

Cleary, P. and Monaghan, J. (1999). Conduction modelling using smoothed particle hydrodynamics. *Journal of Computational Physics*, 148(1):227–264.

Colagrossi, A. (2004a). *A Meshless Lagrangian Method for Free-Surface and Interface Flows with Fragmentation*. PhD thesis, Universita di Roma, La Sapienza.

Colagrossi, A. (2004b). *A Meshless Lagrangian Method for Free-Surface and Interface Flows with Fragmentation.* PhD thesis, Universita di Roma, La Sapienza.

Delannée, C., Török, T., Aulanier, G., and Hochedez, J. (2008). A new model for propagating parts of eit waves: a current shell in a cme. *Solar Physics*, 247(1):123–150.

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

Gingold, R. and Monaghan, J. (1977). Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389.

Gingold, R. A. and Monaghan, J. J. (1978). Binary fission in damping rotating polytropes. *Mon.Not.R.Astron.Soc.*, 184:481–499.

Gray, J. P., Monaghan, J. J., and Swift, R. P. (2001). Sph elastic dynamics. *Comput.Methods Appl.Mech.Engrg.*, 190:6641–6662.

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

Hubber, D. A., Falle, S. A. E. G., and Goodwin, S. P. (2011). Convergence of sph and amr simulations. In *COMPUTATIONAL STAR FORMATION*, number 270 in IAU Symposium Proceedings Series, pages 429–432. CAMBRIDGE UNIV PRESS. 270th Symposium of the International-Astronomical-Union on Computational Star Formation, Barcelona, SPAIN, MAY 31-JUN 04, 2010.

Kincaid, D. and Cheney, E. (2002). *Numerical analysis: mathematics of scientific computing*, volume 2. Amer Mathematical Society.

Knight, T. M. C. (2008). Constructing and modelling hydrodynamic processes using smoothed particle algorithms. Master's thesis, University of Wales, Aberystwyth.

Lanczos, C. (1986). *The Variational Principles of Mechanics.* 4th ed. edition.

Lucy, L. B. (1977). Numerical approach to testing of fission hypothesis. *Astronomical Journal*, 82:1013–1024.

Monaghan, J. (1985). Extrapolating b splines for interpolation. *Journal of Computational Physics*, 60(2):253–262.

Monaghan, J. (1992). Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30:543–574.

Monaghan, J. (1997). Sph and riemann solvers. *Journal of Computational Physics*, 136(2):298–307.

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

Monaghan, J. (2005). Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703.

Monaghan, J. and Price, D. (2001). Variational principles for relativistic smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 328(2):381–392.

Monaghan, J. J. (2002). Sph compressible turbulence. *Mon.Not.R.Astron.Soc.*, 335:843–852.

Monaghan, J. J. and Price, D. J. (2004). Toy stars in one dimension. *Mon.Not.R.Astron.Soc.*, 350:1449–1456.

Monaghan, J. J. and Price, D. J. (2006). Toy stars in two dimensions. *Mon.Not.R.Astron.Soc.*, 365:991–1006.

Morris, J. and Monaghan, J. (1997). A switch to reduce sph viscosity. *Journal of Computational Physics*, 136(1):41–50.

Morris, J. P. (1996). *Analysis of Smoothed Particle Hydrodynamics with Applications*. PhD thesis, Department of Mathematics, Monash University.

Nelson, R. P. (1994). A conservative formulation of sph with variable smoothing lengths. *Mem.S.A.It.*, 65:1161–1165.

Phillips, G. and Monaghan, J. (1985). A numerical method for three-dimensional simulations of collapsing, isothermal, magnetic gas clouds. *Monthly Notices of the Royal Astronomical Society*, 216:883–895.

Powell, K., Roe, P., Linde, T., Gombosi, T., and De Zeeuw, D. (1999). A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *Journal of Computational Physics*, 154(2):284–309.

Price, D. (2008). Modelling discontinuities and kelvin-helmholtz instabilities in sph. *Journal of Computational Physics*, 227(24):10040–10057.

Price, D. (2010). Smoothed particle magnetohydrodynamics–iv. using the vector potential. *Monthly Notices of the Royal Astronomical Society*, 401(3):1475–1499.

Price, D. (2012). Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794.

Price, D. and Monaghan, J. (2004a). Smoothed particle magnetohydrodynamics–i. algorithm and tests in one dimension. *Monthly Notices of the Royal Astronomical Society*, 348(1):123–138.

Price, D. and Monaghan, J. (2004b). Smoothed particle magnetohydrodynamics–ii. variational principles and variable smoothing-length terms. *Monthly Notices of the Royal Astronomical Society*, 348(1):139–152.

Price, D. and Monaghan, J. (2005). Smoothed particle magnetohydrodynamics–iii. multidimensional tests and the $\nabla \cdot \mathbf{B} = 0$ constraint. *Monthly Notices of the Royal Astronomical Society*, 364(2):384–406.

Price, D. J. and Monaghan, J. J. (2004c). Smoothed particle hydrodynamics: Some shocking results... *International Workshop on Magnetic Fields and Star Foramtion Univ Complutense de Madrid, Madrid, SPAIN*, 292:279–283.

Rosswog, S. and Price, D. (2007). Magma: a 3d, lagrangian magnetohydrodynamics code for merger applications. *Mon.Not.R.Astron.Soc.*, 379:915–931.

Sod, G. A. (1978). Survey of several finite-difference methods for systems of non-linear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31.

Song, B. and Dong, L. (2010). A new boundary treatment method for sph and application in fluid simulation. In *Information and Computing (ICIC), 2010 Third International Conference on*, volume 4, pages 82–85. IEEE.

Springel, V. (2005). The cosmological simulation code gadget-2. *Monthly Notices of the Royal Astronomical Society*, 364(4):1105–1134.

Springel, V. and Hernquist, L. (2002). Cosmological smoothed particle hydrodynamics simulations: the entropy equation. *Monthly Notices of the Royal Astronomical Society*, 333(3):649–664.

Tóth, G. (2000). Theâ·< i> b</i>= 0 constraint in shock-capturing magnetohydrodynamics codes. *Journal of Computational Physics*, 161(2):605–652.

Tóth, G. (2002). Conservative and orthogonal discretization for the lorentz force. *Journal of Computational Physics*, 182(1):346–354.

Tskhakaya, D. (2008). The particle-in-cell method. In Fehske, H., Schneider, R., and Weisse, A., editors, *COMPUTATIONAL MANY-PARTICLE PHYSICS*, volume 739 of *Lecture Notes in Physics*, pages 161–189. SPRINGER-VERLAG BERLIN. Summer School on Computational Many-Particle Physics, Greifswald, GERMANY, SEP 18-29, 2006.

Vignjevic, R. (2004). Review of development of the smooth particle hydrodynamics (sph) method.

Vila, J. (1999). On particle weighted methods and smooth particle hydrodynamics. *Mathematical models and methods in applied sciences*, 9(2):161–210.

Yokoyama, T., Akita, K., Morimoto, T., Inoue, K., and Newmark, J. (2001). Clear evidence of reconnection inflow of a solar flare. *The Astrophysical Journal Letters*, 546:L69.

Smoothed Particle Magnetohydrodynamics for the Solar Corona

Abstract:

The focus of the work herein is the construction, by reasoned argument and investigation, of a Smoothed Particle (SP) algorithm and model suitable for the stable, accurate simulation of Solar Coronal phenomena.

The SP method is a numerical technique that approximates a medium and the equations that govern the mediums behaviour such that they form a finite element, Lagrangian model suitable for computation. In this case, that medium is the the Solar Corona, the highly dynamic outer layer of the Sun's atmosphere.

Criteria as to what properties define a *suitable* simulation are, in themselves, a matter for debate. However, general criteria can be established by referencing other numerical models of the Corona and looking at the short-comings of other, more general SP algorithms. For example, quantifiably high accuracy, variable spatial resolution and a degree of optimisation for computational efficiency.

Subsequent to a through discussion of general SP theory and implementation, this document initially presents research investigating the nature of the techniques defining variable spatial resolution (variable smoothing length), and discovers a numerical artefact that in the presence of large variations in smoothing length drastically effects the stability of the algorithm. In order to better understand this artefact, the subsequent research focuses on the quantification of errors leading to the cumbersome, but more accurate Corrected SP (CSP) method. Following an investigation into the behaviour of that algorithm, another is presented that conforms to the high accuracy generated by the CSP method, but strips away that which makes the it so computationally weighty. Entitled the CSP-Δh method, it avoids some of the complications of the former method by justifiably dropping those terms that are a function of the gradients of spatial resolution (that is the spatial gradients of the smoothing length). Finally, the work defines a novel domain geometry in order to capture the required dynamics with as little wasted computation as possible. Two new boundaries have been established, the first a specification of the periodic boundary and the second a novel boundary that attempts to let information about the internal dynamics out of the domain while ensuring the model evolution remains stable.