

Imperial College London
Department of Earth Science and Engineering

GENERALISED IMMERSED BOUNDARY
METHODS FOR FINITE-DIFFERENCE
GEOPHYSICAL WAVEFIELD MODELLING

EDWARD CAUNT

December 2023

Supervised by: Dr. Rhodri Nelson, Dr. Gerard Gorman, Dr. Fabio Luporini

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Earth Science and Engineering
of Imperial College London
and the Diploma of Imperial College London

Declaration

I herewith certify that the material in this thesis that is not my own work has been properly acknowledged and referenced.

Edward Caunt

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International Licence (CC BY-NC-ND).

Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Topography has considerable impact on seismic wavefields, producing complex patterns of reflections and diffractions. Careful representation is paramount in numerical models of such phenomena; omission or poor-quality implementation severely degrades simulation accuracy, carrying into any functionality built atop such models. Whilst structured finite-difference approaches are widely used in seismic simulation, they are difficult to reconcile with such geometries, producing surface representations which are artificially blocky, resulting in nonphysical behaviour.

At the large scales involved in seismic simulation, increasing model resolution becomes unfeasibly computationally expensive, necessitating efficient algorithms. Immersed boundary approaches embed curvilinear surfaces within regular computational grids, naturally accommodating irregular topography by suitably continuing fields beyond the interior domain, yielding accurate results with coarser discretisations than would be otherwise possible. By formulating a generalised immersed boundary, a consistent methodology can be applied across a wide range of boundary conditions and geometries, avoiding the need to develop a new numerical treatment for each particular problem.

Removing the need for interventions based on domain knowledge, it becomes possible to automatically devise immersed boundary treatments for a given discretisation and set of boundary conditions. A software framework, Schism, was designed for this purpose, creating separation of concerns between numerical model specification and underlying implementation by generating low-level boundary treatments from a high-level symbolic specification.

With this novel approach, topography implementation for a wide range of equations was explored, including several problems of seismological interest for which no published examples exist. Mountainous real-world topography could be accurately accommodated with relatively coarse discretisations, suppressing spurious scattering whilst capturing complex reflections and other free-surface effects generated by such irregular profiles. This approach offers a path towards implementation of complex geometries at scale across a diversity of geophysical applications.

To all my friends

Acknowledgements

In completing this work, I have drawn on the support of friends, family, and colleagues, without whom this would not have been possible.

I would like to extend my deepest thanks to Rhodri Nelson for his support and guidance over the course of my PhD. You have been unfailing in your support, even in unprecedented circumstances, and would not have started this project, nor could I have finished it without you. Here's to many more years of science and "just one beer". I would also like to express my gratitude to Gerard Gorman for his vision and passion for the project, and to Fabio Luporini for his help and encouragement throughout. I will be forever grateful for your supervision and all I have learned over the course of this work.

I would also like to thank Mathias Louboutin for sharing his encyclopedic knowledge of seismic imaging, numerical methods, and computing at all manner of unsociable hours. Please log out of Slack and get some sleep. I would also like to express my gratitude to the rest of the Devito team, including Zoe Leibowitz, George Bisbas, and Kene Nwegbu who I have been lucky enough to work alongside over the course of my PhD. Seeing the work that you have done has been truly inspiring and I wish you all the best for everything the future holds.

I am grateful to the wider community for their input, support, and interest in my work throughout this project. In particular, I would like to extend my thanks to Tim MacArthur for stress-testing my work in his own experiments, vindicating the design and capabilities of the approach.

To Rita Kounoudis, Luke Giudici, Will Scott, Evie Baker, Cat Harris, Alex Campbell and everyone else who has shared this journey with me:

your company has brightened my PhD experience immeasurably. It has been a pleasure seeing the science you have done and I look forward to seeing everyone's next steps as this era comes to a close. To my long-suffering flatmate Kat, thanks for putting up with my indecipherable ramblings and for being a sounding board for so many of my problems. Your support has meant so much to me, and I doubt I would be here without you.

Finally, to my family, and my friends in London and elsewhere, thank you for supporting me throughout this chapter of my life, and here's to the next one.

Contents

1. Introduction	15
1.1. Overview	15
1.2. Thesis Outline and Contributions	17
1.3. Dissemination	20
1.4. Background	21
1.4.1. Seismology and Wave Equations	21
1.4.2. Wave Equations in Linearly Elastic Media	23
2. General Mathematical Approach for Immersed Boundaries	33
2.1. Design Considerations	38
2.2. Context and Approach	39
2.3. General Approach for Constructing Immersed Boundary Derivative Operators	41
2.3.1. Constructing FD Approximations	41
2.3.2. FD With Higher-Dimensional Basis Functions	43
2.3.3. Imposition of Boundaries	46
2.3.4. Approximating Boundary Conditions	48
2.3.5. Boundary-Adjacent Support Regions	51
2.3.6. Reconciliation with the Interior Numerical discreti- sation	52
2.3.7. Extension to Multiple Fields	54
2.4. Summary	58

3. Immersed Boundaries and the 2nd-Order Acoustic Wave Equation	61
3.1. The Acoustic Wave Equation and its Discretization	62
3.2. The Free Surface	62
3.2.1. Polynomial Approximation of Free-Surface Constraints	63
3.2.2. Application to Example Geometry	66
3.2.3. Reduction to 1D	70
3.2.4. Convergence	72
3.3. The Rigid Surface	74
3.4. Application to Real-World Topography	76
3.4.1. 2D 2nd-order acoustic free-surface	77
3.4.2. 2D 2nd-order acoustic rigid-surface	78
3.4.3. 3D 2nd-order acoustic free-surface	79
3.4.4. 3D 2nd-order acoustic rigid-surface	81
3.5. Conclusions	81
3.6. Future Work	83
4. Immersed Boundaries and the 1st-Order Acoustic Wave Equation	85
4.1. Equations and Discretization	86
4.2. The Free Surface	88
4.2.1. Polynomial Approximation	88
4.2.2. Boundary Points	90
4.3. Stencil Formation	90
4.3.1. Geometrical Handling of Subgrids	90
4.4. Application to Example Geometry	91
4.4.1. Stencil Size and Implications	93
4.5. Convergence Testing	94
4.6. Examples	96
4.6.1. Application to Simple Geometry	96
4.6.2. 2D 1st-order acoustic free-surface	97
4.6.3. 3D 1st-order acoustic free-surface	99
4.7. Conclusions	100
4.8. Future Work	102

5. Immersed Boundaries, VTI, and TTI	103
5.1. Duveneck/Plessix VTI	105
5.1.1. Free-Surface Immersed Boundaries	105
5.1.2. Comparing Isotropic and Anisotropic Conditions . .	108
5.1.3. Real-Topography Example	111
5.2. Fletcher VTI	112
5.2.1. Free-Surface Immersed Boundaries	113
5.2.2. Real-Topography Example	116
5.3. Fletcher TTI	117
5.3.1. Free-Surface Immersed Boundaries	118
5.3.2. Real-Topography Example	121
5.4. Conclusions	121
5.5. Future Work	122
6. Immersed Boundaries and Elastic Wave Equations	125
6.1. 2nd-Order Elastic Wave Equations	127
6.1.1. Application to Simple Topography	129
6.1.2. Application to Real-World Topography	129
6.2. VTI Free Surface	133
6.2.1. Application to Simple Topography	135
6.2.2. Application to Real-World Topography	135
6.3. TTI Free Surface	138
6.3.1. Application to Simple Topography	139
6.3.2. Application to Real-World Topography	141
6.4. Long-Time Stability	143
6.4.1. Characterising Instabilities	143
6.4.2. Proposed Rectification	145
6.5. 1st-Order Elastic	151
6.6. Free-Surface Boundary Conditions	152
6.7. Examples	153
6.7.1. Instability with Staggered discretisation	157
6.7.2. Application of Unstaggered Scheme	158
6.8. Conclusions	160
6.9. Future Work	161

7. Implementation of the Mathematical Framework	165
7.1. API Approach and Considerations	166
7.1.1. Intent of the Software Framework	166
7.1.2. API Example	168
7.1.3. API Overview	172
7.1.4. Numerical Scheme Specification	173
7.1.5. Geometry Specification and Handling	174
7.1.6. Boundary Condition Specification and Handling . .	175
7.2. Generation of Immersed Boundary Operators	177
7.2.1. Identifying Boundary-Adjacent Points	177
7.2.2. Symbolic Representation of Constraints	178
7.2.3. Bridging the Gap Between Symbolic and Numeric .	179
7.2.4. Matrix Assembly and Inversion	180
7.2.5. Coefficient Calculation	181
7.2.6. Boundary Operator Expressions	182
7.2.7. Support Regions	183
7.3. Testing, Documentation, and Availability	183
7.4. Future Research Directions	184
8. Conclusions	189
8.1. Summary	189
8.2. Limitations	192
8.3. Future Research Directions	192
A. Imaging and Gradients	197
References	200

Chapter 1

Introduction

1.1. Overview

Partial differential equations (PDEs) are used to model the behaviour of numerous phenomena of interest across fields spanning computational fluid dynamics (CFD), electromagnetics, seismic modelling, and beyond. Obtaining exact solutions for many target problems is challenging, becoming mathematically onerous long before a sufficient level of complexity to represent real-world scenarios is attained; no known solution exists in some cases. FD methods approximate derivatives within the governing PDE as a weighted sum of discrete point values, commonly referred to as a stencil. Whilst arranging these points on a structured, rectilinear grid lends itself to more straightforward implementation, the physical shape of the problem often does not conform to this rectangular grid. Stencils will consequently require points located beyond the bounding surface of the computational domain.

To avoid the need for unstructured grids and the associated complexity and computational cost, FD operators can be modified to immerse a curvilinear surface within the structured grid. By ensuring underlying basis functions respect the boundary condition at the boundary position, a sharp, accurate boundary representation is achieved. This approach sidesteps the need for mesh generation and enables efficient, structured FD methods to be applied throughout the wider domain, reducing computational cost. Whilst simply treating points according to their position on the interior or exterior may seem a more straightforward option, the

resultant stair-step representation in which the surface jumps from one node to the next severely limits model accuracy [Brehm and Fasel, 2013]. These immersed boundary approaches are widely used in CFD for simulating interaction between fluids and complex geometries, having been initially applied for modelling blood flow within the heart [Peskin, 1972]. They have been explored in contexts ranging from flow-induced sound [Seo and Mittal, 2011] to the aerodynamics of insect wings [Vargas et al., 2008].

In the context of geophysical wavefield modelling, the wide range of equations used to approximate the behaviour of seismic waves, and the embedding of these wavesolvers within higher-level algorithms present a challenge to devising and implementing suitable immersed boundary treatments. By introducing a generalised mathematical method, immersed boundary treatments for some particular set of boundary conditions can be devised via a consistent methodology in an equation-agnostic manner.

A software framework was developed to implement the mathematical method, designed to compliment the abstraction of the FD method offered by the domain-specific language (DSL) and compiler Devito [Louboutin et al., 2019, Loporini et al., 2019]. Modified FD stencils implementing the immersed boundary are automatically generated from a high-level expression of the boundary geometry and conditions, these being demonstrated for a range of equations of interest. These equations span a wide range of complexity from constant-density isotropic acoustic to anisotropic elastic, demonstrating the broad applicability of the approach.

Abstracting the user from low-level implementation manages the complexity inherent in immersed boundary implementations, creating a separation of concerns between the requisite boundary specification and how it is to be achieved. As this thesis will demonstrate, this approach of generalisation, abstraction, and automation delivers significant advantages to topography implementation and immersed boundaries more generally, these being:

Consistency The process of devising an immersed boundary treatment can be carried out via a consistent methodology without domain-specific knowledge or intuition of the physical problem.

Flexibility By abstracting the boundary treatment from both the equations

solved and the conditions imposed, immersed boundaries of the class described can be formulated for a wide range of problems. Application across the wide range of boundary conditions associated with topography in seismological problems demonstrates the efficacy of this approach.

Usability The construction of a software framework implementing the method improves the accessibility of the immersed boundary method and creates a separation of concerns between the mathematical or geophysical problem and its underlying numerical implementation. This enables straightforward inclusion of immersed boundary treatments in higher-level applications without the need for deep involvement in low-level implementation.

Reproducibility Results are reproducible without the need to re-implement the mathematical method or numerical solver to ensure their veracity, due to the open-source nature of the framework.

1.2. Thesis Outline and Contributions

This work takes a three-pronged approach to implementing irregular boundaries in geophysical wavefield models, these elements being

- Application of immersed boundaries for topography implementation in problems of seismological interest.
- Engineering of software abstractions by which the underlying numerical methods and overarching application layer can be coupled.
- Development of a general class of immersed boundary treatments encompassing the requirements of the problems of interest whilst being amenable to automatic generation from a compact specification.

The novel aspect of this work is the combination of these elements: applications centred in geophysical wavefield modelling form the main motivation and contextualise the work, the mathematical aspect enables generalisation across these problems, whilst the software design allows this to be effectively leveraged. Over the course of this work, these threads

of investigation informed one another, high-level requirements dictating lower-level formulations, whilst the low-level mathematical approach and subsequent automation accelerated investigations at the topmost level. Abstraction and automation were powerful tools for building geophysical models, enabling a greater depth of investigation than would otherwise have been possible.

In addressing these topics, the following ground is covered:

- Characterisation of the information distinguishing a particular immersed boundary approach of the general type developed.
- Investigation of the necessary abstractions and routines involved in automatic generation of immersed boundary operators.
- Evaluation of the accuracy and suitability of both the immersed boundary approach developed and the conditions which it is used to impose.
- Investigation of the suitability of immersed boundaries as a means of implementing challenging real-world topography in geophysical wavefield models across wide range of complexity.

This thesis is structured as follows: Chapter 2 establishes the mathematical background and outlines the generalised immersed boundary approach developed. This includes the use of boundary constraints to fit the basis, and modification of interior stencils in the vicinity of the boundary to obtain the immersed boundary operators.

Application to specific problems is subsequently detailed in Chapters 3 through 6. These chapters form a hierarchy of complexity, incrementally addressing increasingly involved physics, mathematical formulations, and numerical discretizations spanning from isotropic acoustic in Chapter 3 to anisotropic elastic wave equations detailed in Chapter 6.

Chapters 3 and 4 detail the application of the immersed boundary approach to the isotropic acoustic wave equation in 2nd and 1st-order formulations respectively, considering free and rigid-surface boundary conditions associated with topography in models of seismic wave and infrasound propagation. These equations serve as a testbed to validate the approach's applicability across disparate boundary conditions, the repre-

sensation of multidimensional conditions, and its applicability to conditions specifying some relationship between fields.

This work is built upon in Chapter 5, exploring the suitability of immersed boundaries for including topography in FD models solving a range of anisotropic pseudoacoustic equations, of interest due to their widespread use in seismic imaging contexts. The generality of the approach facilitated its application across proposed formulations of varying levels of sophistication, including those for which immersed boundary treatments have not been previously published, and enabled comparison of various free-surface conditions.

Chapter 6 applies the immersed boundary approach to the elastic free surface. Seismic waves comprise both compressive (primary or P wave) and shear (secondary or S wave) modes; at interfaces such as the Earth's surface, conversion between the two occurs alongside simple reflection. The framework developed was used to implement irregular topography with multiple formulations of the elastic wave equation, and the stabilisation of such boundary treatments was explored.

Finally Chapter 7 details the workings and API of Schism, the software framework developed to implement the mathematical method, which was used to produce all examples provided throughout this work. The user interface for the package and its integration with Devito are outlined alongside the major algorithmic components behind the automated generation of immersed boundary operators. Details of the package including distribution, documentation, and testing are also provided.

This thesis advances the state of the art by presenting both a novel approach to immersed boundary formulation and implementation and a practical means of including irregular topography in a range of geophysical wavefield models. This work leverages abstraction of the immersed boundary from the physical problem and subsequent automation of the mathematical method to cover an unprecedented diversity of seismological problems. Applications of immersed boundaries to many of the conditions and equations studied in this work have not previously been presented in the literature to the author's knowledge.

The key novel contributions herein:

- Development of an immersed boundary approach which can be ap-

plied consistently across equations and boundary conditions without the need for domain-specific intervention, including cases where boundary conditions specify some relationship between fields.

- Implementation of a software framework to automatically devise immersed boundary treatments of this class from a high-level specification and generate modified FD operators incorporating this boundary.
- Application of immersed boundary treatments to implement real-world topography in a wide range of problems of geophysical interest, including rigid surface boundary conditions for acoustic wave propagation, and free surface implementations for the TTI wave equations of Fletcher et al. [2009] and anisotropic elastic wave equations, among others.
- Demonstration of the power of abstraction and automation in managing complexity introduced by immersed boundary treatments, removing barriers to their usage in higher-level workflows.

1.3. Dissemination

Publications An abridged version of the material presented in Chapters 2, 3, and 4 forms the basis of a publication currently submitted to the journal *Geophysics*. A preprint of the paper can be found on the repository ArXiv at <https://arxiv.org/abs/2309.03600>.

- E. Caunt, R. Nelson, F. Luporini, G. Gorman. *A Novel Immersed Boundary Approach for Irregular Topography with Acoustic Wave Equations*, 2023

Presentations Development of the mathematical method and software framework was the product of several revisions, the last of which, detailed in this work, entirely superseded previous efforts. The work presented herein formed the basis for the following presentation:

- E. Caunt, R. Nelson, G. Gorman, F. Luporini. *High-Fidelity Topography Implementation for Finite-Difference Models of Infrasonic Propagation*. In AGU Fall Meeting Abstracts (Vol. 2022, pp. S56B-07).

The prototype mathematical and software framework in its numerous iterations formed a foundation for many of the concepts explored in this work. A list of presentations pertaining to this work is as follows:

- E. Caunt, R. Nelson, F. Luporini, G. Gorman. *Devitoboundary: An Open-Source Tool for Topography Implementation in Finite-Difference Wavesolvers with Devito*. Poster presentation, Rice EHPC 2022.
- E. Caunt, R. Nelson, G. Gorman, F. Luporini. *An Open-Source Tool for Accurate Topography Implementation within Finite Difference Wave Solvers*. In AGU Fall Meeting Abstracts (Vol. 2021, pp. S15E-0295).
- E. Caunt, R. Nelson, F. Luporini, G. Gorman. *Implementation of Irregular Topography in Finite-Difference Acoustic Wave Models*. Presented at EAGE Annual 2021.
- E. Caunt, R. Nelson, F. Luporini, G. Gorman. *A Flexible, High-Level Abstraction for Topography Implementation Within Finite Difference Wave Solvers*. Poster presentation, Rice OGHPC 2021.
- E. Caunt, R. Nelson, F. Luporini, G. Gorman. *3D Immersed Boundary Generation From Topography Point Clouds: an Implementation in Devito*. Poster presentation, Rice OGHPC 2020.

Software Schism, an open-source software framework developed as a plugin for Devito, was developed as a part of this work, underpinning all examples presented. It is available on Github and is archived on Zenodo.

Github: <https://github.com/devitocodes/schism>

Zenodo: <https://zenodo.org/record/8167794>

1.4. Background

1.4.1. Seismology and Wave Equations

Rapid increases in computing power have enabled the proliferation of wave-equation-based geophysical algorithms, particularly in exploration

seismology. Velocity model building is increasingly carried out using full-waveform inversion (FWI) [Warner et al., 2013] rather than ray tomography or normal move-out (NMO) corrections, whilst migration of horizons commonly uses reverse-time migration (RTM) [Liu et al., 2012] in the place of Kirchhoff migration algorithms. These compute-intensive workflows surpass their predecessors in resolving subsurface structure.

FWI, summarised by Virieux and Operto [2009], is an iterative, nonlinear data-fitting procedure which attempts to find a subsurface parameterisation which minimises misfit between the resultant simulated and real-world datasets, widely credited to Tarantola [1984]. Whilst Mora [1986], Pica et al. [1990], Crase et al. [1990] demonstrated some of the first applications to real datasets, application to practical 3D problems did not come to fruition for many years afterwards [Ben-Hadj-Ali et al., 2008, Warner et al., 2013]. Wavefield modelling varies wildly between implementations, featuring numerous equations and solvers. In the simplest case, these may involve acoustic or anisotropic pseudoacoustic wave equations [Plessix and Cao, 2011, Warner et al., 2013, Hobro et al., 2014, Agudo et al., 2018], although elastic wave equations are increasingly common [Zhang et al., 2023]; a tradeoff between kinematic realism and computational cost is always present [Warner et al., 2013, Hobro et al., 2014]. Furthermore, whilst explicit FD schemes are commonplace, frequency-domain approaches [Ben-Hadj-Ali et al., 2008], and even FE and spectral-element (SE) methods [Borisov et al., 2018, Roberts et al., 2021b] are occasionally used, depending upon requirements.

Similarly, RTM forward propagates source wavefields whilst backward propagating recorded wavefields, the two coinciding at subsurface reflectors [Baysal et al., 1983]. It naturally handles multipathing and overturned raypaths [Fowler et al., 2010], lending superior performance in complex geological contexts. By calculating an imaging condition for each shot and stacking these, a high-resolution section is obtained. Detailed subsurface models obtained from FWI are commonly used for parameterisation [Zhang et al., 2023].

Outside of exploration seismology, such solvers are used in geophysical contexts including earthquake modelling [Lee et al., 2009, DI Michele et al., 2022], travel-time tomography [Park and Pyun, 2021], source localisation [Kim and Lees, 2014, Fee et al., 2021] and characterisation [Kim

et al., 2015], and teleseismic FWI [Monteiller et al., 2013, 2015, Wang et al., 2016, Beller et al., 2018]. Seismic modelling also enables detailed investigation of particular earthquake events [DI Michele et al., 2022] or hypothetical scenarios [Lee et al., 2009], playing a key role in seismic hazard analysis. Applications also exist in nuclear test monitoring [Frankel and Leith, 1992, Rodgers et al., 2010], non-destructive testing [Köhn et al., 2016], and medical imaging contexts [Guasch et al., 2020].

Whilst rarely an end in itself, wavefield modelling is found in numerous applications in seismology and beyond, representing a diverse class of scientific problems.

1.4.2. Wave Equations in Linearly Elastic Media

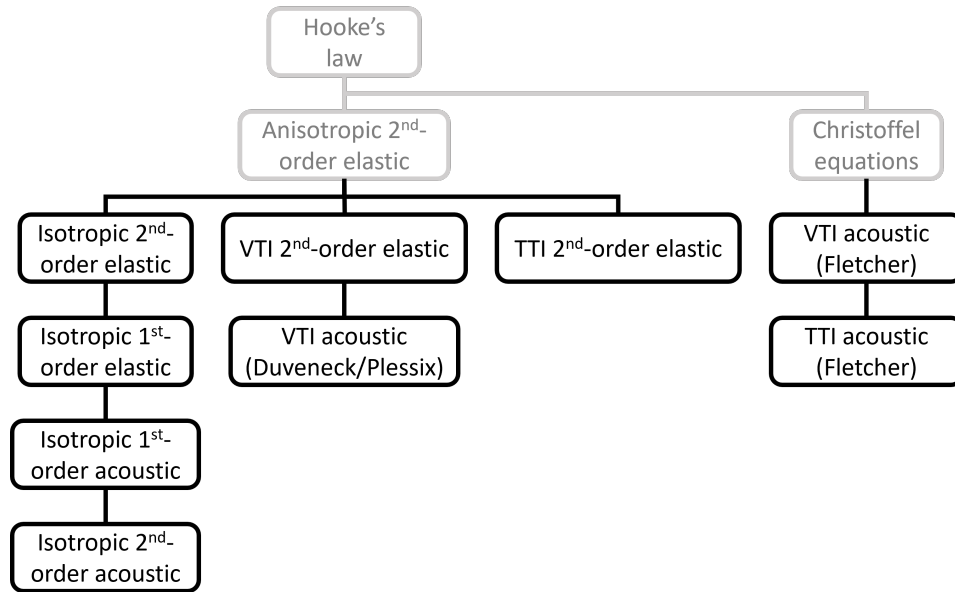


Figure 1.1.: A diagram of relationships between equations considered in this work, each being derived from that above. Equations to which immersed boundary treatments are applied in this work are shown in solid black.

Geophysical wavefield modelling relies on a wide range of wave equations, spanning acoustic, anisotropic pseudoacoustic [Fletcher et al., 2009, Louboutin et al., 2018], viscoacoustic [Carcione et al., 1988], elastic (isotropic and anisotropic) [Virieux, 1986, Yang et al., 2015], and viscoelastic [Carcione, 1993] propagation. An overview of relationships between equations considered in this work is shown in Figure 1.1.

The core of linear elasticity is Hooke's law, compactly specified in N dimensions as

$$\tau_{ij} = \sum_{k=1}^N \sum_{l=1}^N C_{ijkl} e_{kl}, \quad (1.1)$$

$$i, j = 1, 2, 3, \dots,$$

where τ_{ij} , C_{ijkl} , and e_{kl} are components of the stress, stiffness, and strain tensors respectively.

Strains are related to particle displacement via

$$e_{ij} = \frac{\partial u_i}{\partial x_j}, \quad (1.2)$$

where $\mathbf{u}$ is the particle displacement vector. Note that $\mathbf{e}$ is symmetrical, so $\frac{\partial u_i}{\partial x_j} = \frac{\partial u_j}{\partial x_i}$. Substituting Equations 1.1 and 1.2 into the equations of motion

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \sum_{j=1}^N \frac{\partial \tau_{ij}}{\partial x_j}, \quad (1.3)$$

where ρ is density, assuming components of C_{ijkl} are locally-constant or slowly-varying, yields

$$\frac{\partial^2 u_i}{\partial t^2} = \frac{1}{\rho} \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N C_{ijkl} \frac{\partial^2 u_k}{\partial x_j \partial x_l}, \quad (1.4)$$

this being the 2nd-order formulation of the elastic wave equation in the general anisotropic case.

The stiffness tensor $\mathbf{C}$ is more easily illustrated by reindexing with Voight notation, enabling the fourth-order tensor to be represented as a square 2D matrix of size $\frac{N(N+1)}{2}$. In 3D, the mapping is as follows:

ij or kl	α or β
11	1
22	2
33	3
32 = 23	4
31 = 13	5
12 = 21	6

Table 1.1.: Correspondence between natural and Voight notation indices in 3D. Note that ij maps onto α and kl onto β . The reverse mapping can also be applied.

The stiffness tensor in Voight notation, $\mathbf{C}_{\alpha\beta}$, remains symmetrical. For the general anisotropic (triclinic) case, this is

$$\mathbf{C}_{\alpha\beta} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ & & C_{33} & C_{34} & C_{35} & C_{36} \\ & & & C_{44} & C_{45} & C_{46} \\ & & & & C_{55} & C_{56} \\ & & & & & C_{66} \end{bmatrix}. \quad (1.5)$$

A special case of interest to geophysics is that of transverse isotropy. Whilst the causes and nature of this anisotropy are covered in more detail in Chapter 5, the mathematics are sufficient for now. From Thomsen [1986], the stiffness tensor for transverse isotropy with a vertically-oriented symmetry axis (vertical transverse isotropy or VTI) is:

$$\mathbf{C}_{\alpha\beta} = \begin{bmatrix} C_{11} & (C_{11} - 2C_{66}) & C_{13} & & & \\ & C_{11} & C_{13} & & & \\ & & C_{33} & & & \\ & & & C_{44} & & \\ & & & & C_{44} & \\ & & & & & C_{66} \end{bmatrix}. \quad (1.6)$$

Zero elements are also omitted. Thomsen [1986] defines two anisotropy parameters:

$$\epsilon \equiv \frac{C_{11} - C_{33}}{2C_{33}}, \quad (1.7)$$

and

$$\delta \equiv \frac{(C_{13} + C_{44})^2 - (C_{33} - C_{44})^2}{2C_{33}(C_{33} - C_{44})}. \quad (1.8)$$

In 3D, an additional parameter γ is required

$$\gamma \equiv \frac{C_{66} - C_{44}}{2C_{44}}. \quad (1.9)$$

From here, two equations explored in this work can be derived. Rotat-

ing the stiffness tensor using matrix $\mathbf{R}$

$$\mathbf{C}'_{ijkl} = \sum_{p=1}^N \sum_{q=1}^N \sum_{r=1}^N \sum_{s=1}^N R_{pi} R_{qj} R_{rk} R_{sl} \mathbf{C}_{pqrs}, \quad (1.10)$$

the symmetry axis is tilted from the vertical [Barucq et al., 2014], yielding tilted transverse isotropy (TTI). The rotated tensor is dense, unlike that in Equation 1.6, essentially being a special case of triclinic anisotropy. In combination with Equation 1.4, one obtains the second-order formulation of the TTI elastic wave equation.

Equations for acoustic propagation featuring VTI were proposed by [Duveneck et al., 2008], subsequently lightly modified by [Plessix and Cao, 2011]. The lower right quadrant of the stiffness matrix given by Equation 1.6 corresponds to shear components, and is thus zeroed. Defining horizontal and vertical stresses as

$$\tau_H = \tau_{11} + \tau_{22}, \quad (1.11)$$

and

$$\tau_V = \tau_{33}, \quad (1.12)$$

one obtains

$$\tau_H = \rho c_p^2 \left((1 + 2\epsilon) \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \right) + \sqrt{1 + 2\delta} \frac{\partial u_z}{\partial z} \right), \quad (1.13)$$

and

$$\tau_V = \rho c_p^2 \left(\sqrt{1 + 2\delta} \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \right) + \frac{\partial u_z}{\partial z} \right), \quad (1.14)$$

where c_p is P-wave velocity. Using Equation 1.3, a system of second-order partial differential equations can be derived:

$$\begin{aligned} \frac{\partial^2 \tau_H}{\partial t^2} &= c_p^2 \left((1 + 2\epsilon) \left(\frac{\partial^2 \tau_H}{\partial x^2} + \frac{\partial^2 \tau_H}{\partial y^2} \right) + \sqrt{1 + 2\delta} \frac{\partial^2 \tau_V}{\partial z^2} \right) \\ \frac{\partial^2 \tau_V}{\partial t^2} &= c_p^2 \left(\sqrt{1 + 2\delta} \left(\frac{\partial^2 \tau_H}{\partial x^2} + \frac{\partial^2 \tau_H}{\partial y^2} \right) + \frac{\partial^2 \tau_V}{\partial z^2} \right). \end{aligned} \quad (1.15)$$

Further equations can be derived from Equation 1.4. The isotropic stiff-

ness tensor is given as:

$$\mathbf{C}_{\alpha\beta} = \begin{bmatrix} C_{33} & (C_{33} - 2C_{44}) & (C_{33} - 2C_{44}) & & & \\ & C_{33} & (C_{33} - 2C_{44}) & & & \\ & & C_{33} & & & \\ & & & C_{44} & & \\ & & & & C_{44} & \\ & & & & & C_{44} \end{bmatrix}, \quad (1.16)$$

with

$$C_{33} = \lambda + 2\mu, \quad (1.17)$$

and

$$C_{44} = \mu, \quad (1.18)$$

where λ and μ are the Lamé parameters. A coupled system of second-order PDEs to solve the isotropic elastic wave equation can be obtained using Equations 1.4 and 1.16. Note that these equations are considerably more compact than their anisotropic counterparts as the stiffness matrix is largely sparse.

The isotropic elastic wave equation is more commonly formulated as the first-order coupled system of stress and particle velocity introduced by Virieux [1986]. As particle velocity is the first derivative of particle displacement with respect to time, Equation 1.3 can alternatively be presented as

$$\rho \frac{\partial v_i}{\partial t} = \sum_{j=1}^N \frac{\partial \tau_{ij}}{\partial x_j}, \quad (1.19)$$

whilst substituting Equation 1.2 into Equation 1.1 and taking the first derivative with respect to time yields

$$\frac{\partial \tau_{ij}}{\partial t} = \sum_{k=1}^N \sum_{l=1}^N C_{ijkl} \frac{\partial v_k}{\partial x_l}. \quad (1.20)$$

In combination with the stiffness tensor from Equation 1.16, Equations 1.19 and 1.20 form the coupled system.

In the acoustic case, shear modulus (μ) is zero, and the trace of $\boldsymbol{\tau}$ is equal to pressure p . Given the relationship between Lamé parameters

and P-wave velocity

$$c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad (1.21)$$

Equations 1.19 and 1.20 yield

$$\rho \frac{\partial \mathbf{v}}{\partial t} = \nabla p, \quad (1.22)$$

and

$$\frac{\partial p}{\partial t} = \rho c_p^2 \nabla \cdot \mathbf{v}, \quad (1.23)$$

respectively, this coupled system being the 1st-order formulation of the isotropic acoustic wave equation. Taking the first derivative of Equation 1.23 with respect to time and substituting in Equation 1.22, one obtains a second-order equation for the variable-density acoustic wave equation:

$$\frac{\partial^2 p}{\partial t^2} = \rho c_p^2 \nabla \cdot \left(\frac{1}{\rho} \nabla p \right). \quad (1.24)$$

Due to the insensitivity of transmitted waves to density, and resulting difficulties in inverting for this parameter [Virieux and Operto, 2009], density is often assumed to be constant, yielding

$$\frac{\partial^2 p}{\partial t^2} = c_p^2 \nabla^2 p, \quad (1.25)$$

this being the most common form of the second-order isotropic acoustic wave equation.

Unlike the VTI formulation proposed by Duveneck et al. [2008], the TTI formulation of Fletcher et al. [2009] is derived via the Christoffel equations. Taking Equation 1.4 and substituting the trial solution

$$u_k = U_k \exp \left(i\omega \left(t - \frac{1}{c} \sum_{j=1}^N n_j x_j \right) \right), \quad (1.26)$$

into both sides, one obtains

$$\sum_{k=1}^N (G_{ik} - \rho c^2 \delta_{ik}) U_k = 0, \quad (1.27)$$

where

$$G_{ik} = C_{ijkl}n_jn_l. \quad (1.28)$$

This can alternatively be expressed as the linear system

$$\begin{bmatrix} G_{11} - \rho c^2 & G_{12} & G_{13} \\ G_{12} & G_{22} - \rho c^2 & G_{23} \\ G_{13} & G_{23} & G_{33} - \rho c^2 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = 0. \quad (1.29)$$

Switching to Voight notation and taking the VTI stiffness tensor from Equation 1.6, this becomes

$$\begin{bmatrix} C_{11}n_1^2 + C_{44}n_3^2 - \rho c^2 & 0 & (C_{13} + C_{44})n_1n_3 \\ 0 & C_{66}n_1^2 + C_{44}n_3^2 - \rho c^2 & 0 \\ (C_{13} - C_{44})n_1n_3 & 0 & C_{44}n_1^2 + C_{33}n_3^2 - \rho c^2 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = 0. \quad (1.30)$$

Note that the shear mode polarised in the x_2 direction can be separated from the compressive mode, which is coupled to the remaining shear modes polarised in the perpendicular plane. In Fourier space, the wavefront-normal vectors are given as

$$\begin{aligned} n_1 &= \frac{ck_x}{\omega} \\ n_3 &= \frac{ck_z}{\omega}, \end{aligned} \quad (1.31)$$

where k_i refers to the wavenumber corresponding to dimension x_i . Defining the density-normalised stiffness

$$a_{ij} = \frac{C_{ij}}{\rho}, \quad (1.32)$$

enables Equation 1.30 to be rewritten as

$$\begin{bmatrix} a_{11}k_x^2 + a_{44}k_z^2 - \omega^2 & (a_{13} + a_{44})k_xk_z \\ (a_{13} + a_{44})k_xk_z & a_{44}k_x^2 + a_{33}k_z^2 - \omega^2 \end{bmatrix} \begin{bmatrix} U_1 \\ U_3 \end{bmatrix} = 0. \quad (1.33)$$

Equation 1.33 can be treated as an eigenvalue problem, the eigenvalue being ω^2 :

$$\left(\begin{bmatrix} a_{11}k_x^2 + a_{44}k_z^2 & (a_{13} + a_{44})k_xk_z \\ (a_{13} + a_{44})k_xk_z & a_{44}k_x^2 + a_{33}k_z^2 \end{bmatrix} - \omega^2 \mathbf{I} \right) \begin{bmatrix} U_1 \\ U_3 \end{bmatrix} = 0. \quad (1.34)$$

Setting the determinant of this matrix to zero yields

$$0 = \omega^4 - ((a_{11} + a_{44})k_x^2 + (a_{33} + a_{44})k_z^2)\omega^2 + a_{11}a_{44}k_x^4 + a_{33}a_{44}k_z^4 + (a_{11}a_{33} + a_{44}^2 - (a_{13} + a_{44})^2)k_x^2k_z^2. \quad (1.35)$$

Substituting the relationships between density-normalised stiffness components and seismic wavespeeds

$$\begin{aligned} a_{33} &= c_p^2 \\ a_{44} &= c_s^2 \\ a_{11} &= c_{px}^2 = c_p^2(1 + 2\epsilon) \\ a_{44} + \frac{(a_{13} + a_{44})^2}{a_{33} - a_{44}} &= c_{pn}^2 = c_p^2(1 + 2\delta), \end{aligned} \quad (1.36)$$

into Equation 1.33 yields

$$0 = \omega^4 - ((c_{px}^2 + c_s^2)(k_x^2 + k_y^2) + (c_p^2 - c_s^2)k_z^2)\omega^2 + c_{px}^2c_s^2(k_x^2 + k_y^2)^2 + c_p^2c_s^2k_z^4 + (c_p^2(c_{px}^2 - c_{pn}^2) + c_s^2(c_p^2 + c_{pn}^2))(k_x^2 + k_y^2)k_z^2. \quad (1.37)$$

Inverse Fourier transforms of $(k_x^2 + k_y^2)$ and k_z^2 yield differential operators

$$\begin{aligned} H_1 &= \frac{\partial^2}{\partial z^2} \\ H_2 &= \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \end{aligned} \quad (1.38)$$

respectively, Equation 1.37 becoming

$$\begin{aligned} 0 &= \frac{\partial^4}{\partial t^4} - ((c_{px}^2 + c_s^2)H_2 + (c_p^2 - c_s^2)H_1)\frac{\partial^2}{\partial t^2} + c_{px}^2c_s^2H_2H_2 \\ &\quad + c_p^2c_s^2H_1H_1 + (c_p^2(c_{px}^2 - c_{pn}^2) + c_s^2(c_p^2 + c_{pn}^2))H_2H_1. \end{aligned} \quad (1.39)$$

Defining an auxiliary function q , parameterised by scalar value α

$$q = \frac{(c_{pn}^2 - c_s^2)H_2}{\alpha \frac{\partial^2}{\partial t^2} - c_s^2H_2 - c_p^2H_1} p, \quad (1.40)$$

a coupled system can be formed by multiplying Equation 1.39 by p :

$$\begin{aligned}\frac{\partial^2 p}{\partial t^2} &= c_{px}^2 H_2 p + \alpha c_p^2 H_1 q + c_s^2 H_1 (p - \alpha q) \\ \frac{\partial^2 q}{\partial t^2} &= \frac{c_{pn}^2}{\alpha} H_2 p + c_p^2 H_1 q - c_s^2 H_2 \left(\frac{p}{\alpha} - q\right)\end{aligned}\quad (1.41)$$

Note that the equation governing q is a rearrangement of Equation 1.40.

Velocity parameters c_{pn} and c_{px} are P-wave velocities as determined by normal move-out and in the plane of isotropy respectively. Nonzero c_s can be set to stabilise the numerical scheme and prevent SV wavefront triplications [Fletcher et al., 2009], parameterised by some constant σ :

$$\sigma = \frac{c_p^2}{c_s^2}(\epsilon - \delta). \quad (1.42)$$

As contrast in σ across some interface corresponds to a reflection coefficient of SV-waves, c_s is set such that σ is constant [Fletcher et al., 2009]. It has been found empirically that triplications in the shear wave artefact are removed for values less than $\sigma = 0.8$ [Fletcher et al., 2009]. Nonzero c_s is unnecessary in isotropic or elliptically anisotropic media [Fletcher et al., 2009]. Note that by setting $v_s = 0$ and $\alpha = \frac{1}{\sqrt{1+2\delta}}$, one obtains the Equations 1.13 and 1.14.

Whilst the above equations are for VTI media, they can be adapted to TTI, rotating differential operators H_1 and H_2 to obtain

$$\begin{aligned}H_1 &= \sin^2\theta \cos^2\phi \frac{\partial^2}{\partial x^2} + \sin^2\theta \sin^2\phi \frac{\partial^2}{\partial y^2} + \cos^2\theta \frac{\partial^2}{\partial z^2} + \sin^2\theta \sin 2\phi \frac{\partial^2}{\partial x \partial y} \\ &\quad + \sin 2\theta \sin \phi \frac{\partial^2}{\partial y \partial z} + \sin 2\theta \cos \phi \frac{\partial^2}{\partial x \partial z} \\ H_2 &= \nabla^2 - H_1,\end{aligned}\quad (1.43)$$

where θ is the angle between the symmetry axis and the vertical and ϕ is the azimuth of the symmetry axis measured from the x axis.

Note that all the above often feature a forcing term on the right-hand side, typically representing some point source or set thereof in seismic contexts. Details of common wavelets used in seismic processing contexts can be found in Ryan [1994].

Chapter 2

General Mathematical Approach for Immersed Boundaries

Irregular topography produces complex path effects which turn clean, defined reflections into cascades of overlapping arrivals, causing waves to diffract and scatter, focus and defocus [Takemura et al., 2015, Reinoso et al., 1996, Boore, 1972, Griffiths and Bollinger, 1979]. The sum of these effects corresponds to considerably altered wavefield dynamics versus the flat case; omission is recognised to adversely affect model accuracy [Monteiller et al., 2013, Li and Yao, 2020, Borisov et al., 2018], often severely [Nuber et al., 2016] although this may be acceptable in some applications [Bleibinhaus and Rondenay, 2009]. When applied to imaging problems, unrealistic propagation paths result in artefacts in the processed image [Bleibinhaus and Rondenay, 2009, Nuber et al., 2016, Borisov et al., 2018] which risk being mistaken for real geological features, whilst considerable amplification discrepancies are also observed [Lee et al., 2009, Asimaki and Mohammadi, 2018]. Simplistic corrections to account for topographic variation perform poorly in wave-equation-based workflows [Nuber et al., 2016, Huiskes et al., 2017, Royle et al., 2021]. Sharp, smooth, and correctly positioned surfaces are required; phase and amplitude errors are liable to be introduced otherwise [Seo and Mittal, 2011, Hu, 2016]. As such, topography must be accounted for in the underlying numerical solver.

Unstructured meshes suitably conformed to the topography offer an

immediate solution. Finite-element methods (FEMs) have been demonstrated as effective for modelling wave propagation in the presence of variable topography (e.g. Zhebel et al., 2014, Dupros et al., 2010, Borisov et al., 2018, Liu et al., 2014, Mulder and Shamasundar, 2016), and have been applied to include complex terrain in models of earthquake peak ground acceleration [Lee et al., 2009, DI Michele et al., 2022], in shallow structure characterization applications [Romdhane et al., 2011], and to imaging problems such as full-waveform inversion (FWI) [Roberts et al., 2021b, Shin et al., 2013, Monteiller et al., 2015, Qu et al., 2017]. Unstructured finite-difference methods (FDMs) have also been developed [Takekawa et al., 2015, Martin et al., 2015], and successfully used to perform FWI on synthetic datasets [Wu et al., 2022], alongside RTM [Wu et al., 2021] and least-squares RTM [Deng et al., 2023].

Unstructured mesh generation remains problematic however, particularly for complex models or at scale [Slotnick et al., 2014] and requires *a priori* knowledge of seismic velocities [Borisov et al., 2018, Roberts et al., 2021a,b]. Generation of meshes for complex and uncertain geological models is a non-trivial problem [Peter et al., 2011, Shin et al., 2013], balancing several competing priorities including resolution requirements, cost, and gradation rate [Roberts et al., 2021b]. Compounding this issue is the reconciliation of geological structure with high-quality meshes [Pellerin et al., 2014, Anquez et al., 2019]. Furthermore, the geometry of geological structures is prone to generating ill-conditioned sliver elements as units get pinched out resulting in unreasonably small timesteps Roberts et al. [2021b]. Borisov et al. [2020] notes that mesh generation is sufficiently arduous that adaption for different frequency ranges is omitted in their study. Automated mesh generation for seismic wave propagation in models containing arbitrary horizons remains an open problem, with unstructured approaches rarely used in production.

Numerous strategies have been explored for topography implementation in lieu of unstructured meshes; structured FD methods are commonplace, being computationally cheap, straightforwardly implemented, and eschewing grid generation [Liu and Sen, 2009], with aggressive optimisation possible [Luporini et al., 2019, Louboutin et al., 2019]. The simplest approach, achieved through a low-impedance layer at the surface, [Schultz, 1997, Zeng et al., 2012] trade accuracy for simplicity and intu-

ition; satisfactory results are obtainable with care [Zeng et al., 2012], although convergence is found to be sub-second-order, regardless of interior discretisation [Zhebel et al., 2014, Symes and Vdovina, 2009] and error is potentially egregious for complex geometries [Graves, 1996, Zahradnik et al., 1993]. The aforementioned staircasing of the surface produces this error; locally-refined subgrids can yield improvement [Opsal and Zahradnik, 1999, Tavelli et al., 2019], although such refinement is non-trivial [Lai and Peskin, 2000, Goldstein et al., 1993], often requiring careful filtering and interpolation routines [Zhang et al., 2013b]. The degree of mislocation is directly proportional to resolution, given that the interface is moved to the nearest grid point, incurring first-order error [Mulder, 2017]. Furthermore, the solution is non-differentiable across discontinuities, incurring additional second-order error [Mulder, 2017]. Nonphysical scattering resulting from non-smooth representations of the boundary pollute the solution [de Groot-Hedlin, 2004, Zeng et al., 2012, Mulder, 2017, Gao et al., 2015]; empirical studies find that the resolution required to suppress these effects is multiple times finer than that dictated by numerical dispersion, even for low-order schemes [Bohlen and Saenger, 2006].

Slightly more sophisticated are staircased image methods, several variations having been explored [Robertsson, 1996, Boore, 1972, Hayashi et al., 2001, Ohminato and Chouet, 1997, Ripperger et al., 2003]. However, despite explicitly enforcing the free surface, diffractions remain [Muir et al., 1992, Hu, 2016] alongside a time shift proportional to the difference between the interface and computational grids [Symes and Vdovina, 2009]. With a curvilinear transformation, topography can be mapped to a horizontal plane in the iteration space avoiding staircasing [Tessmer et al., 1992, Zhang and Chen, 2006, Zhang et al., 2012], enabling boundary conditions to be imposed exactly on the surface via a range of approaches [Hestholm and Ruud, 1994, Hestholm et al., 1999, Hestholm and Ruud, 2002, Tarrass et al., 2011, de la Puente et al., 2014, Petersson and Sjögreen, 2015]. This is perhaps more intuitively conceptualised as solving the wave equation on a grid which has been deformed to match the topography. Accuracy is comparable to FEM [Zhang et al., 2012], and various wave equations have been solved with such schemes [Zhang and Chen, 2006, Zhang et al., 2012, Hestholm and Ruud, 1994, Hestholm et al., 1999, Hestholm

and Street, 2000, Sun et al., 2017, de la Puente et al., 2014]. Transformed equations are more expensive than their Cartesian counterparts [Zhang and Chen, 2006], although suitable transformations allow for Cartesian FD grids in the lower regions of the model [Qu et al., 2017]. However, deriving such mappings can still be challenging, particularly if properties such as a conformal grid or surface-orthogonality are required. This approach has been widely used to simulate topographic scattering of seismic waves [Hestholm and Ruud, 2002] and has also been used to carry out FWI on synthetic data featuring topography [Qu et al., 2017]. However, a smooth conformal mapping may be challenging to obtain or yield locally small cells, limiting the maximum timestep [Shragge, 2014] in much the same manner as sliver elements in FEM.

The immersed boundary method presents an alternative approach to the irregular free surface, retaining the Cartesian grid, whilst accurately representing topography as a smooth surface. Immersed boundary approaches to capture scattering effects introduced by realistic, complex geometries have been developed in both computational acoustics and aeroacoustics [Seo and Mittal, 2011, Zhao et al., 2022, Bilbao, 2022, Sun and Jiang, 2012], and geophysical wavefield modelling [Mulder, 2017, Mulder and Huiskes, 2017, Hu, 2016, Gao et al., 2015, Almuheidib and Toksoz, 2015, Li and Yao, 2020, Li et al., 2022] contexts. They have been applied to second- [Zhang et al., 2013a, Mulder, 2017, Li and Yao, 2020] and first-order acoustic wave equations [Mulder and Huiskes, 2017, Hu, 2016], alongside some success with the isotropic elastic wave equation [Lombard et al., 2008, Almuheidib and Toksoz, 2015, Gao et al., 2015]. Pronounced topographic variation is straightforwardly handled; Mulder [2017] shows a test case using topography from El Capitan in Yosemite, which would present a formidable challenge to coordinate-transform approaches.

However, only a handful of studies considering RTM [Zhang et al., 2013a] and FWI [Huiskes et al., 2017, Li et al., 2022] have been produced to date, and frameworks to effectively leverage immersed boundaries have been neglected. Furthermore, approximations suited to a narrowly defined application are commonplace, often reposing the problem as an image method such as that detailed in Robertsson [1996]. Multidimensional conditions or those imposing some relationship between fields are typically replaced by suitable near-equivalents, requiring domain knowledge

and application-specific intervention. This deduction requires a wider understanding of the problem and hinders extension.

This is at odds with the proliferation of domain-specific languages (DSLs) and code generation, with projects such as Devito [Luporini et al., 2019, Louboutin et al., 2019] and Firedrake [Rathgeber et al., 2016] increasingly gaining a toehold in computational acoustics [Stoltzfus et al., 2021], geophysical wavefield modelling [Louboutin et al., 2018], and scientific numerical modelling more widely [Kärnä et al., 2018]. Such approaches represent a new paradigm for numerical modelling, introducing abstraction layers between underlying numerical method and specification, increasing productivity [Louboutin et al., 2019] and creating separation of concerns between overarching problem and implementation [Rathgeber et al., 2016]. Low-level solver kernels are automatically generated from a concise, high-level specification, with sophisticated optimisations applied during the lowering process to yield performant, portable code [Luporini et al., 2019, Louboutin et al., 2019]. Developing a suitable abstraction and automation approach hinges on a general method for solving a given problem; this cannot be easily achieved with the application-specific immersed boundary implementations developed to date.

Whilst parameterisations may change, a generalised approach will follow from specification to implementation via a consistent set of steps. Thus, a method by which immersed boundary treatments to fit some particular specification can be devised is proposed; contrast this with devising a specific immersed boundary treatment. Immersed boundary operators of this class can be obtained in an equation- and boundary-condition-agnostic manner, enabling automated generation of appropriate boundary treatments across a range of problems and boundary conditions, necessitated by varied equations detailed in Figure 1.1.

The novel immersed boundary approach developed in this work addresses issues of existing methods via the following:

Increased flexibility Application of a consistent methodology, regardless of governing equations or boundary conditions was a key design criterion when developing the method. This focus avoids the challenges associated with extending a narrowly targeted approach to new applications; the required flexibility is inherent in the method,

avoiding the need for reworking.

High-level mathematical abstraction and amenability to automation By avoiding the need for domain knowledge or low-level intervention in the numerical scheme to impose the boundary condition, implementation of the boundary treatment can be automated and the level of abstraction at which the immersed boundary can be specified is raised. This enables the construction of a high-level framework implementing the method.

Application-agnosticism By considering how treatments can be devised for an abstract class of problems, suitable immersed boundaries can be obtained to suit any problem in this class. This contrasts with existing approaches which seek to integrate immersed boundaries into some particular existing numerical model or application. Consequently, problems do not need to be explicitly targeted for the method to be applicable.

A generalised approach enables a consistent methodology to be applied across varied problems, in turn enabling the automation and abstraction associated with DSLs and code-generation paradigms to be applied to tackle a wide range of problems.

2.1. Design Considerations

This generalised approach necessitates the following design considerations:

Requirements of a generalised methodology Applicability across multiple equations requires compatibility with a wide variety of stencils, arbitrary boundary conditions (which may be multidimensional or dictate a relationship between multiple fields), and arbitrary geometry in any number of dimensions. Handling this range of parameterisations via a consistent methodology is key to automatically devising immersed boundary treatments of the type described. However, this precludes many of the prior assumptions regarding the boundary geometry or constraints, commonly leveraged to simplify the

formulation of immersed boundary approaches. A boundary representation which enables natural representation of both generalised constraints and complex boundary geometries is thus required.

Target applications and degree of generality The set of boundary conditions encountered within the target applications of the immersed boundary approach developed informs the formulation of the generalised boundary constraint. Whilst attempting to encapsulate all of the numerous wave equations and boundary conditions relevant to seismology would be overambitious, too restrictive a set is liable to limit the applicability of the method, requiring careful consideration.

Accuracy and stability Each component of the immersed boundary treatment must incur consistent error with the overarching scheme to achieve full-order convergence. The method must avoid approximation prior to discretisation to prevent inadvertent introduction of errors, whilst ensuring that the boundary representation is accurate and consistent with the interior discretisation. Furthermore, stability is a concern: several prior efforts emphasise the need for stabilisation efforts [Mulder, 2017, Peller et al., 2006, Dhamankar et al., 2015, Zhao et al., 2022]. Whilst some involve rigorous analysis based on knowledge of the full discretisation [Brehm et al., 2015, Mulder, 2017], in practice ad-hoc interior point criteria [Hu, 2016, Mulder, 2017, Mulder and Huiskes, 2017] or filtering [Dhamankar et al., 2015, Zhao et al., 2022] are used. With the abstracted approach proposed, there is no knowledge of the full discretisation, making consistent stability a challenge to be addressed.

2.2. Context and Approach

The immersed boundary approach outlined in this chapter is an extension of those detailed in Mulder [2017], Mulder and Huiskes [2017], Huiskes et al. [2017]. The key innovation within these publications is the encapsulation of the immersed boundary through spatially-variant stencil coefficients in the vicinity of the surface. Grid-aligned 1D extrapolations, introduced by Zhang et al. [2013a], are constrained using 1D approximations

of the boundary conditions. For the acoustic free surface, the support region of this extrapolant is contained within that of the derivative operator of the same order; the two operations can consequently be combined into a single stencil.

Boundary treatments detailed in Brehm and Fasel [2013], Brehm et al. [2015] also encapsulate boundary forcing within the FD stencils themselves, rather than relying on forcing values at external points (known as ghost nodes). Unlike in Mulder [2017] however, there is no concept of extrapolating fields beyond the boundary. A Dirichlet boundary condition is included within the series expansions from which the derivative stencil is formed, explicitly enforcing the condition upon the approximating basis.

Early stages of the derivation process highlighted in Brehm and Fasel [2013] share similarities with that detailed in the proceeding section, although the concept of extrapolations baked into the stencil is retained from Mulder [2017]. To address shortcomings of 1D extrapolants, namely the inability to handle multidimensional boundary conditions or cross-derivative stencils, extrapolant construction is extended to higher dimensions and multiple fields.

Drawing upon mesh-free FD methods for seismic modelling [Martin et al., 2015, Li et al., 2017, Wu et al., 2022], the work of Takekawa et al. [2015] in particular, multidimensional Taylor-series basis functions were adopted and the boundary considered as a set of locally-unstructured gridpoints rather than a continuous, locally-planar surface. Use of high-dimensional Taylor-series basis functions - also used in Takekawa and Mikada [2015, 2016a,b] - in the place of more common radial-basis functions as in Martin et al. [2015], Cai et al. [2022], yields a linear system to obtain stencil coefficients similar to that used in the conventional, structured FD case. Derivatives of the Taylor-series basis are straightforwardly calculated and incorporated into this linear system, enabling the basis to be constrained with a range of boundary conditions [Takekawa and Mikada, 2019].

2.3. General Approach for Constructing Immersed Boundary Derivative Operators

The process of designing immersed boundary operators shares many similarities with the construction of FD operators in the conventional case. To provide a foundation for the proposed immersed boundary approach, formation of FD approximations is considered more typically. The intention is to highlight the range of approximations resulting from choices during the derivation process.

2.3.1. Constructing FD Approximations

FDMs discretise fields as a set of point values located at nodes on a grid; updating these according to the governing equations approximates the evolution of the underlying continuous function. These values constrain a basis with some specified error relative to the continuous function, diminishing as resolution increases.

To evaluate derivatives within the governing PDEs, the basis is used to form approximations of these derivative operators. As with the basis itself, these will have some error relative to their true value. To outline how these approximations are made in the typical case, consider some function f in a 1D space; approximating with an M th-order Taylor series expansion at some point x_0 yields

$$f(x) = \sum_{m=0}^M \frac{(x - x_0)^m}{m!} \frac{\partial^m f(x_0)}{\partial x^m} + O((x - x_0)^{M+1}), \quad (2.1)$$

concisely represented as

$$\mathbf{a} \cdot \boldsymbol{\delta} = f(x), \quad (2.2)$$

where

$$\mathbf{a}^T = \left(1, x - x_0, \dots, \frac{(x - x_0)^M}{M!} \right), \quad (2.3)$$

and

$$\boldsymbol{\delta}^T = \left(f(x_0), \frac{\partial f}{\partial x}(x_0), \dots, \frac{\partial^M f}{\partial x^M}(x_0) \right). \quad (2.4)$$

For some even M , expansions at $M + 1$ discrete points, labelled $x_{-M/2}$

through $x_{M/2}$, enable the formation of the linear system

$$\mathbf{A}\delta = \begin{pmatrix} f(x_{-M/2}) \\ \vdots \\ f(x_{M/2}) \end{pmatrix}. \quad (2.5)$$

To illustrate, for $M = 4$, the system is as follows

$$\begin{pmatrix} 1 & x_{-2} - x_0 & \frac{(x_{-2}-x_0)^2}{2} & \frac{(x_{-2}-x_0)^3}{6} & \frac{(x_{-2}-x_0)^4}{24} \\ 1 & x_{-1} - x_0 & \frac{(x_{-1}-x_0)^2}{2} & \frac{(x_{-1}-x_0)^3}{6} & \frac{(x_{-1}-x_0)^4}{24} \\ 1 & 0 & 0 & 0 & 0 \\ 1 & x_1 - x_0 & \frac{(x_1-x_0)^2}{2} & \frac{(x_1-x_0)^3}{6} & \frac{(x_1-x_0)^4}{24} \\ 1 & x_2 - x_0 & \frac{(x_2-x_0)^2}{2} & \frac{(x_2-x_0)^3}{6} & \frac{(x_2-x_0)^4}{24} \end{pmatrix} \begin{pmatrix} f(x_0) \\ \frac{\partial f}{\partial x}(x_0) \\ \frac{\partial^2 f}{\partial x^2}(x_0) \\ \frac{\partial^3 f}{\partial x^3}(x_0) \\ \frac{\partial^4 f}{\partial x^4}(x_0) \end{pmatrix} = \begin{pmatrix} f(x_{-2}) \\ f(x_{-1}) \\ f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix} \quad (2.6)$$

which can be rearranged in the form

$$\mathbf{A}^{-1} \begin{pmatrix} f(x_{-M/2}) \\ \vdots \\ f(x_{M/2}) \end{pmatrix} = \delta, \quad (2.7)$$

obtaining derivatives as a weighted sum of discrete function values: a stencil. With the assumption of a regular grid, relative positions of points become fixed regardless of where the derivative is taken, yielding

$$\begin{pmatrix} 1 & -2 & 2 & -\frac{4}{3} & \frac{2}{3} \\ 1 & -1 & \frac{1}{2} & -\frac{1}{6} & \frac{1}{24} \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & \frac{1}{2} & \frac{1}{6} & \frac{1}{24} \\ 1 & 2 & 2 & \frac{4}{3} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} f(x_0) \\ \Delta x \frac{\partial f}{\partial x}(x_0) \\ \Delta x^2 \frac{\partial^2 f}{\partial x^2}(x_0) \\ \Delta x^3 \frac{\partial^3 f}{\partial x^3}(x_0) \\ \Delta x^4 \frac{\partial^4 f}{\partial x^4}(x_0) \end{pmatrix} = \begin{pmatrix} f(x_{-2}) \\ f(x_{-1}) \\ f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix}, \quad (2.8)$$

for the above case, the inverse being

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} \\ -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} \\ -\frac{1}{2} & 1 & 0 & -1 & \frac{1}{2} \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix} \begin{pmatrix} f(x_{-2}) \\ f(x_{-1}) \\ f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix} = \begin{pmatrix} f(x_0) \\ \Delta x \frac{\partial f}{\partial x}(x_0) \\ \Delta x^2 \frac{\partial^2 f}{\partial x^2}(x_0) \\ \Delta x^3 \frac{\partial^3 f}{\partial x^3}(x_0) \\ \Delta x^4 \frac{\partial^4 f}{\partial x^4}(x_0) \end{pmatrix}. \quad (2.9)$$

One may observe that the leftmost matrix contains weights for FD stencils of every derivative order up to that of the basis.

2.3.2. FD With Higher-Dimensional Basis Functions

For higher dimensions, a suitably higher-dimensional polynomial is required, formed as a product of per-dimension 1D series of the form

$$f(x) = \sum_{m=0} a_m x^m, \quad (2.10)$$

in 3D yielding

$$f(x) = \sum_{m=0} \sum_{n=0} \sum_{l=0} a_{mnl} x^m y^n z^l. \quad (2.11)$$

Given this, the process outlined above is extensible to higher dimensions as in Takekawa et al. [2015]. In pursuit of consistency and accuracy, it is crucial to maintain consistent error throughout any numerical scheme; polynomials used in such schemes must be of an order matching the spatial discretisation employed. To this end, for an order M spatial discretisation, the 3D expansion is truncated as follows:

$$f(x) = \sum_{m=0}^{M_x} \sum_{n=0}^{M_y} \sum_{l=0}^{M_z} a_{mnl} x^m y^n z^l + O(|\Delta x|^{(M+1)}), \quad (2.12)$$

subject to

$$M_x + M_y + M_z = M, \quad (2.13)$$

thus removing any terms with order greater than M .

An N-dimensional Taylor series is a robust candidate for such applications, retaining similarities to the aforementioned 1D case and being suitably problem-agnostic. In 3D, this is given by

$$f(x) = \sum_{m=0} \sum_{n=0} \sum_{l=0} \frac{(x-x_0)^m (y-y_0)^n (z-z_0)^l}{m!n!l!} \delta_{m,n,l}, \quad (2.14)$$

where

$$\delta_{m,n,l} = \frac{\partial^{m+n+l}}{\partial x^m \partial y^n \partial z^l} f(x_0). \quad (2.15)$$

Truncating this expansion as in Equation 2.13, the polynomial expansion at points within the support region can be represented as a matrix-vector

multiplication, as in Equation 2.2. Reflecting the higher dimensionality and corresponding increased number of terms, $\mathbf{a}$ is given by

$$\mathbf{a}^T = \left(1, (x - x_0), \dots, (z - z_0), \frac{(x - x_0)^2}{2}, \right. \\ \left. (x - x_0)(y - y_0), \dots, \frac{(z - z_0)^2}{2}, \dots, \right. \\ \left. \frac{(x - x_0)^M}{M!}, \dots, \frac{(z - z_0)^M}{M!} \right), \quad (2.16)$$

whilst

$$\delta^T = \left(1, \frac{\partial}{\partial x}, \dots, \frac{\partial}{\partial z}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial x \partial y}, \dots, \frac{\partial^2}{\partial z^2}, \dots, \frac{\partial^M}{\partial x^M}, \dots, \frac{\partial^M}{\partial z^M} \right) f(\mathbf{x}_0). \quad (2.17)$$

As in the 1D case, expansions taken at some set of points are used to form a linear system. However, increased dimensionality implies increased flexibility regarding the distribution of these points. Consider some set of arbitrarily-distributed function values. In one dimension it is reasonably simple to construct a polynomial expansion of any desired order to fit this data, provided the number of linearly independent data points equals or exceeds the coefficients in the truncated expansion. However, when constraining the higher-dimensional basis, this linear independence requires suitable point distribution.

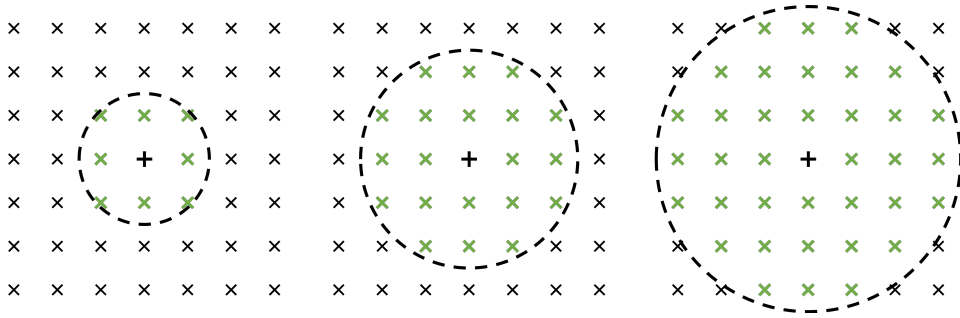


Figure 2.1.: A selection of 2D support region footprints with radii of 1.5, 2.5, and 3.5 respectively. The vertical black cross designates the stencil centre (the position at which the stencil is applied), green crosses are interior points, and the dotted black line shows the extent of the support region.

A hyperspherical support region footprint, similar to that used by Takekawa et al. [2015], is proposed as a suitable choice for a large range of boundary conditions. Extensible to any number of dimensions, this choice of sup-

port region has a defined centre and can be straightforwardly expanded and contracted as necessary without introducing drastic behavioural changes. Furthermore, any given number of points within the support region will be located as close to the stencil centre as possible. Whilst this work limits itself to Cartesian structured grids, such support regions are nominally equally suited to structured and unstructured data as no assumption is required regarding the underlying data structure. In this case, points can be located by index, removing the need for nearest-neighbour searches to find points within the support region. Support regions of various radii are shown in Figure 2.1: a radius of $(M + 1)/2\Delta x$ ensures sufficient information to constrain a polynomial basis on an even, structured grid.

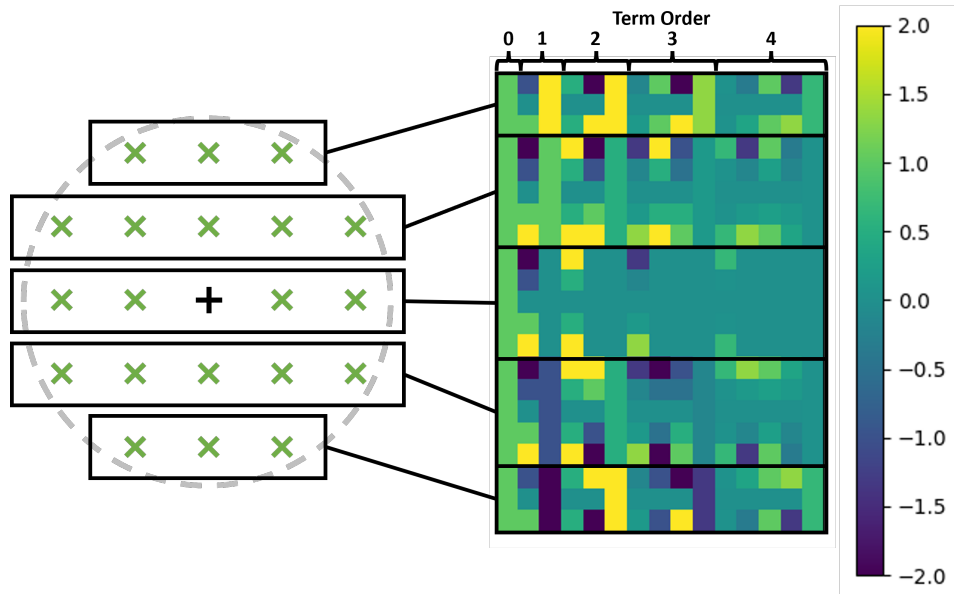


Figure 2.2.: A 2D support region footprint for a 4th-order basis consisting of points X and its correspondence to the matrix A (the structure of which is shown on the right). The support region radius is marked by a dashed grey line. The stencil points in each box correspond to the matrix rows in the connected box, containing coefficients of the polynomial expansion at that point. The order of derivatives to which coefficients correspond increases from left to right. Colours within the matrix plot correspond to the values of those elements, with yellows and blues representing positive and negative values respectively. The vertical black cross designates the stencil centre: the position at which the stencil is applied and expansion point for the polynomial.

Whilst the approach readily extends to any number of dimensions, the proceeding example considers the 2D case for clarity and brevity. Correspondence between a 2D support region for a 4th-order discretisation and

$\mathbf{A}$ is shown in Figure 2.2. Note that this matrix is tall, and thus the linear system

$$\mathbf{A}\delta = f(\mathbf{X}), \quad (2.18)$$

where

$$\begin{aligned} \delta^T = & \left(1, \Delta x \frac{\partial}{\partial x}, \Delta y \frac{\partial}{\partial y}, \Delta x^2 \frac{\partial^2}{\partial x^2}, \Delta x \Delta y \frac{\partial^2}{\partial x \partial y}, \Delta y^2 \frac{\partial^2}{\partial y^2}, \right. \\ & \Delta x^3 \frac{\partial^3}{\partial x^3}, \Delta x^2 \Delta y \frac{\partial^3}{\partial x^2 \partial y}, \Delta x \Delta y^2 \frac{\partial^3}{\partial x \partial y^2}, \Delta y^3 \frac{\partial^3}{\partial y^3}, \\ & \Delta x^4 \frac{\partial^4}{\partial x^4}, \Delta x^3 \Delta y \frac{\partial^4}{\partial x^3 \partial y}, \Delta x^2 \Delta y^2 \frac{\partial^4}{\partial x^2 \partial y^2}, \\ & \left. \Delta x \Delta y^3 \frac{\partial^4}{\partial x \partial y^3}, \Delta y^4 \frac{\partial^4}{\partial y^4} \right) f(\mathbf{x}_0). \end{aligned} \quad (2.19)$$

will be overdetermined. This system is solved with the Moore-Penrose pseudoinverse [Moore, 1920] (equivalent to a least-squares inverse in this case) to obtain the derivative vector. Inverting this system yields expressions for FD stencils using the aforementioned 2D basis.

2.3.3. Imposition of Boundaries

Consider some continuous scalar function within a domain. This could be a standalone scalar function or individual component of some vector field. In the context of wavefield modelling, this may represent pressure, a component of the particle velocity vector, or some stress tensor component. The evolution of the field is dictated not only by the governing system of PDEs but also the boundary conditions imposed. In the case of seismic wavefield modelling, these demark a half-space, bounded by the ground-air interface. Imposing suitable conditions on this interface is necessary for the wavefield to interact with topography in a realistic manner.

Whilst the process of forming stencils is quite straightforward in free space, the boundary presents an issue: derivative operators may be truncated, leaving stencil points on the exterior. Forming FD operators in such situations requires the incorporation of additional constraints, in this case derived from the boundary conditions. This concept has precedent in the design of FD operators, addressing lack of information via other design criterion, dispersion minimisation [Tam and Webb, 1993] being one such example.

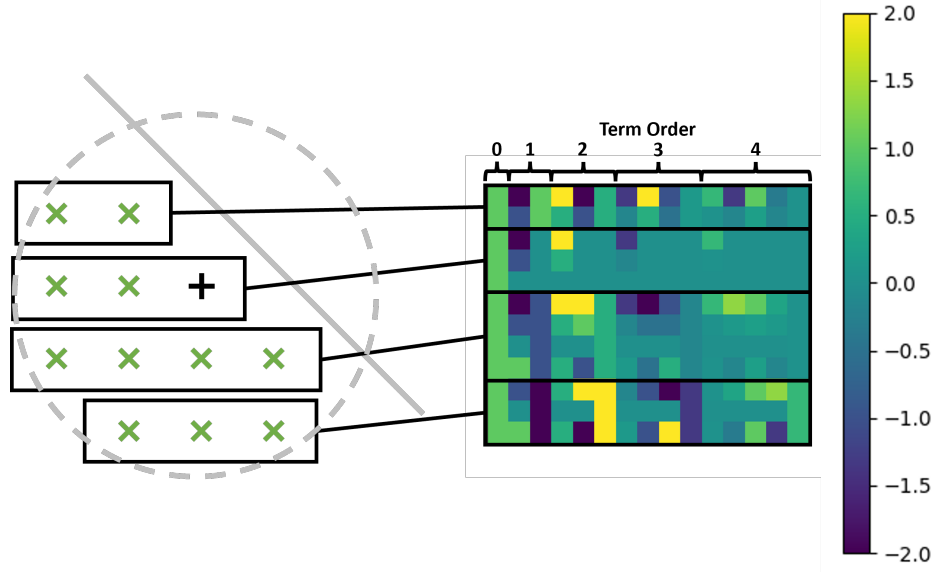


Figure 2.3.: The effect of truncation of the stencil footprint shown in Figure 2.2 by a 45° boundary. The support region radius is marked by a dashed grey line and the solid grey line shows the boundary inserted. Note that the points removed from the stencil correspond to rows removed from $\mathbf{A}$.

Figure 2.3 shows the effect of a boundary cutting through the support region shown in Figure 2.2 on the linear system: $\mathbf{A}$ is underdetermined in its current form.

Immersed boundary treatments extend the function to the edge of the domain in a manner that respects the boundary position and conditions, while maintaining consistency with the error introduced by the spatial discretisation. This is achieved through placing additional constraints upon the basis at boundary points, reducing the number of interior points required and imposing the appropriate behaviour. In doing so, additional rows are added to $\mathbf{A}$. The process of forming these rows given boundary conditions and the points at which they are imposed follows in a similar vein to that of forming polynomial approximations of interior function values.

2.3.4. Approximating Boundary Conditions

In 3D, meaningful linear boundary conditions imposed upon a single field have the general form

$$\sum_{m=0}^{M_x} \sum_{n=0}^{M_y} \sum_{l=0}^{M_z} \alpha_{mnl}(\mathbf{x}_b) \frac{\partial^{m+n+l} f}{\partial x^m \partial y^n \partial z^l}(\mathbf{x}_b) = g(\mathbf{x}_b) \quad (2.20)$$

where M_x , M_y , and M_z are as in Equation 2.13. The function $g(\mathbf{x}_b)$ is a forcing value present at the boundary point $\mathbf{x}_b$. This function is equal to zero for all applications considered in this work, but there is no reason why the method detailed could not be used to impose boundary conditions featuring nonzero $g(\mathbf{x}_b)$ (those specifying a consistent nonzero flux for example). Conditions of this form will not yield the trivial expression $0 = g(\mathbf{x}_b)$ when approximated with the basis. The coefficient $\alpha_{mnl}(\mathbf{x}_b)$ may vary with position.

Substituting f for its Taylor-series approximation yields

$$\sum_{m=0}^{M_x} \sum_{n=0}^{M_y} \sum_{l=0}^{M_z} \sum_{i=0}^{M_x} \sum_{j=0}^{M_y} \sum_{k=0}^{M_z} \beta_{mnljk} \frac{\partial^{i+j+k} f}{\partial x^i \partial y^j \partial z^k}(\mathbf{x}_0) = g(\mathbf{x}_b), \quad (2.21)$$

where

$$\beta_{mnljk} = \alpha_{mnl}(\mathbf{x}_b) \frac{\partial^{m+n+l}}{\partial x^m \partial y^n \partial z^l} \frac{(x - x_0)^i (y - y_0)^j (z - z_0)^k}{i!j!k!}. \quad (2.22)$$

As is the case with Taylor-series approximations at interior points, this is expressible as a dot product:

$$\left(\sum_{m=0}^{M_x} \sum_{n=0}^{M_y} \sum_{l=0}^{M_z} \alpha_{mnl}(\mathbf{x}_b) \frac{\partial^{m+n+l}}{\partial x^m \partial y^n \partial z^l} \mathbf{a} \right) \cdot \boldsymbol{\delta} = g(\mathbf{x}_b). \quad (2.23)$$

Note that the derivative vector is identical to that used for Taylor-series expansion at interior points, and thus some set of interior and boundary constraints can be encapsulated as the multiplication of the derivative vector by some matrix of coefficients. There is no distinction between interior and boundary constraints: it is apparent that interior constraints can be formed from Equation 2.20, and both can be represented as rows of the linear system.

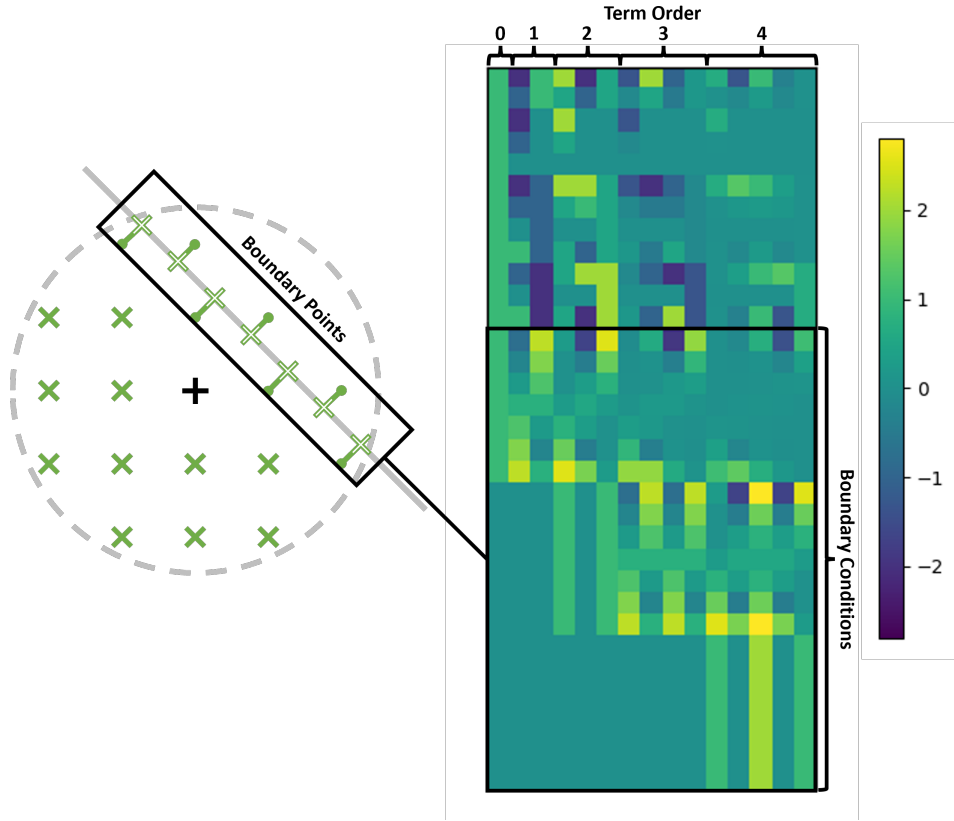


Figure 2.4.: The effect of adding boundary constraints to the truncated support region shown in Figure 2.3. The support region radius is marked by a dashed grey line and the solid grey line shows the boundary inserted. Boundary points where conditions are imposed are shown as hollow green crosses. Solid green dots show the centre of FD cells containing a boundary point; the normal from this point to the boundary is shown as a green line. Boundary conditions increase in order towards the bottom of the matrix. Note that the highest-order boundary conditions here are invariant with position and thus redundant. This is because the polynomial approximation of these conditions produces an expression dependent upon neither x nor y .

With individual constraints for both interior points and boundary conditions, a linear system can be obtained with which to fit the basis, not unlike that in Equation 2.5. Figure 2.4 shows the effect of adding free-surface boundary conditions to the truncated stencil shown in Figure 2.3. Each boundary constraint introduces an additional row to the matrix, with the free-surface boundary conditions introducing three constraints per boundary point. Consequently, the matrix is once again overdetermined, and thus nominally has sufficient information to constrain the

derivatives. Note however, that it may be the case that such a matrix is still not full-rank. For example, particular boundary constraints may contain redundant information, as seen in the lowermost rows of the matrix or lack information regarding particular derivatives.

With interior function values and some J boundary conditions imposed at the boundary points $\mathbf{X}_b$, the linear system

$$\mathbf{A}\delta = \begin{pmatrix} f(\mathbf{X}) \\ g_1(\mathbf{X}_b) \\ \vdots \\ g_J(\mathbf{X}_b) \end{pmatrix}, \quad (2.24)$$

can be formed, with rows of the form given in Equations 2.2 and 2.23, as shown in Figure 2.4. $\mathbf{X}$ is the set of interior points used to construct the interpolant and vectors $g_1(\mathbf{X}_b)$ through $g_J(\mathbf{X}_b)$ contain forcing values corresponding to each boundary condition and forcing point.

As a tangible example, consider constructing a second-order interpolant in 1D from two interior points (labelled x_1 and x_2) and a boundary point upon which the boundary conditions $f(x_b) = 0$ and $\frac{\partial^2 f}{\partial x^2}(x_b) = 0$ are imposed. This particular case yields the linear system

$$\begin{pmatrix} 1 & (x_1 - x_0) & \frac{(x_1 - x_0)^2}{2} \\ 1 & (x_2 - x_0) & \frac{(x_2 - x_0)^2}{2} \\ 1 & (x_b - x_0) & \frac{(x_b - x_0)^2}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f(x_0) \\ \frac{\partial f}{\partial x}(x_0) \\ \frac{\partial^2 f}{\partial x^2}(x_0) \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ 0 \\ 0 \end{pmatrix}. \quad (2.25)$$

Whilst higher-order accuracy and higher dimensionality increase the size of this system (making it somewhat unwieldy to show here) the process of constructing this system remains the same. Note here that whilst this particular system is overdetermined, removing a single interior point from this example gives the approach detailed in Mulder [2017] and Mulder and Huiskes [2017] in which all boundary constraints are selected then supplemented with interior constraints to obtain a square system for which an inverse can be found.

2.3.5. Boundary-Adjacent Support Regions

The process of selecting a suitable set of interior and boundary conditions is worthy of some consideration here. Whilst the method used by Mulder [2017], Mulder and Huiskes [2017] yields easy-to-invert systems and is intuitive in 1D, in higher dimensions more choices are available.

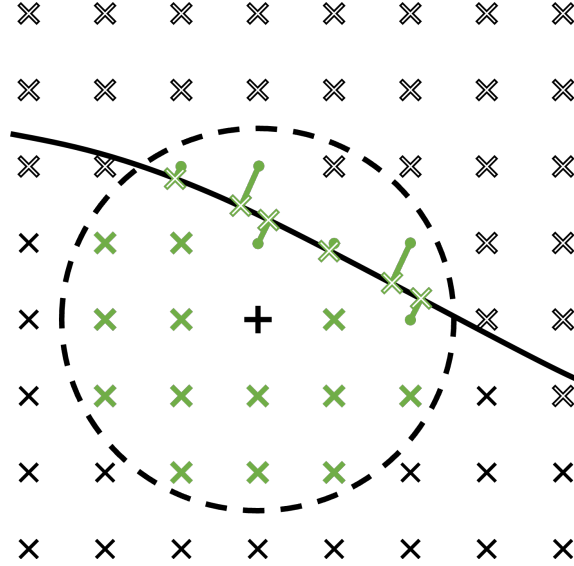


Figure 2.5.: Construction of the support region in the vicinity of the boundary. Symbols are as in Figures 2.1 and 2.4. The solid black line represents the boundary surface. Hollow black crosses are considered to be outside or too close to the boundary for the purposes of constructing the extrapolant (as defined in the proceeding section).

To preserve the Cartesian topology of the grid (thereby allowing all points to be directly indexed), boundary points are taken as the intersection of the boundary normal at a grid point and the boundary, assuming this lies within a point-centred hyperrectangle whose sides correspond to a single grid increment in each dimension. Such a support region is shown in Figure 2.5. From here, a cutoff parameter η is defined. Interior points where a boundary point lies within a gridpoint-centred hyperrectangle with sides of length $2\eta\Delta x_n$ (where Δx_n is the grid increment in the n^{th} dimension) are excluded for the purposes of constructing the extrapolant to avoid instability due to excessively closely-spaced constraints. Note that whilst suitable values of η have been empirically determined in this work, more rigorous analysis would be beneficial (although this may be

challenging, as in Mulder and Huiskes, 2017). An initial support region radius of $(M + 1)/2$ is selected, expanded incrementally if insufficient information to constrain the extrapolation is contained therein.

If the basis is sufficiently constrained, $\text{rank}(A)$ will equal the number of expansion terms. The required rank is $M + 1$ in 1D, $(M + 1)(M + 2)/2$ in 2D, and $(M + 1)(M + 2)(M + 3)/6$ in 3D respectively, the latter two corresponding to the $(M + 1)^{\text{th}}$ triangular and tetrahedral numbers respectively.

If one does not intend to be so strict about maintaining formal order, the order of the polynomial basis can be reduced instead, as in Mulder [2017], Mulder and Huiskes [2017]. It is anticipated that this may be advisable with some boundary conditions and geometries to avoid stencil footprints becoming excessively large in edge cases.

2.3.6. Reconciliation with the Interior Numerical discretisation

Whilst the method described above can be used to directly obtain FD stencils, sudden switches in derivative approximation may lead to instability, and it is thus desirable to retain the interior discretisation throughout the computational domain. The truncation of interior operators applied in the vicinity of the boundary can be addressed by using the boundary-constrained N-dimensional polynomial basis to approximate function values at required exterior points. The process of projecting this basis onto some set of required exterior points $\mathbf{X}_e$ can be represented by the matrix-vector multiplication

$$\mathbf{B}\delta = \tilde{f}(\mathbf{X}_e), \quad (2.26)$$

where $\mathbf{B}$ contains the terms associated with each derivative in the Taylor series evaluated at the respective points in $\mathbf{X}_e$ and $\tilde{f}$ refers to the approximated function. Derivatives are approximated as

$$\delta = \mathbf{A}^+ \begin{pmatrix} f(\mathbf{X}) \\ g_1(\mathbf{X}_b) \\ \vdots \\ g_J(\mathbf{X}_b) \end{pmatrix}, \quad (2.27)$$

where $\mathbf{A}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{A}$. Note that this is the inverse of Equation 2.24.

Using the 1D example shown in Equation 2.25, assume two exterior points, designated x_3 and x_4 are required to complete the stencil operator applied at the specified position. $\mathbf{B}$ will then be of the form

$$\mathbf{B} = \begin{pmatrix} 1 & (x_3 - x_0) & \frac{(x_3 - x_0)^2}{2} \\ 1 & (x_4 - x_0) & \frac{(x_4 - x_0)^2}{2} \end{pmatrix}, \quad (2.28)$$

and thus Equation 2.26 will yield approximations of $f(x_3)$ and $f(x_4)$ as functions of $f(x_1)$, $f(x_2)$ (the other entries in the right-hand side vector of Equation 2.25 are zero). More generally, this process yields exterior function values as functions of interior values and boundary conditions.

Substituting these into the interior stencil as necessary, a modified version of the interior operator is obtained. This operator has a support region consisting of the unity of the set of interior points in the original stencil ($\mathbf{X}_i$), the set of those in the circular stencils used for extrapolation, and the set of boundary points where conditions are enforced.

A stencil expression approximating some derivative of f can be expressed as

$$\mathbf{w} \cdot f(\mathbf{X}) = d(f), \quad (2.29)$$

where d is some arbitrary continuous derivative operator applied to f and $\mathbf{w}$ contains the weights of the stencil used to approximate $d(f)$, corresponding to each function value in the vector $f(\mathbf{X})$. Separating the corresponding vector of stencil weights $\mathbf{w}$ into two subvectors $\mathbf{w}_i$ containing weights for points on the interior and $\mathbf{w}_e$ for points on the exterior, the expression for the modified stencil d can be obtained via

$$\begin{pmatrix} \mathbf{w}_i \\ \mathbf{w}_e \end{pmatrix} \cdot \begin{pmatrix} f(\mathbf{X}_i) \\ \tilde{f}(\mathbf{X}_e) \end{pmatrix} = d. \quad (2.30)$$

Note that by constructing modified operators in this manner, the extrapolated values are used without being explicitly calculated, with the extrapolation process baked into the stencil coefficients. Furthermore, derivative stencils centred at different gridpoints will have independent extrapolations. This independence and the local nature of these extrapolation oper-

ators ensures that the linear systems constructed remain relatively small and simplifies the process of obtaining a solution to the system (both criterion prioritised by Hu, 2016).

2.3.7. Extension to Multiple Fields

Where multiple fields are present, individual Taylor Series are used to approximate each. Supposing K separate fields, labelled f_1 through f_K are present within a model (temporarily ignoring boundary conditions) the polynomial fitting process can be expressed as

$$\begin{pmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_K \end{pmatrix} \begin{pmatrix} \delta_1 \\ \vdots \\ \delta_K \end{pmatrix} = \begin{pmatrix} f_1(\mathbf{X}_1) \\ \vdots \\ f_K(\mathbf{X}_K) \end{pmatrix}, \quad (2.31)$$

where $\mathbf{A}_k$ is the matrix containing coefficients associated with each derivative in the polynomial expansion approximating field f_k at the set of points $\mathbf{X}_k$. The vector δ_k contains the various derivatives of f_k , analogous to the single-field case. The left-hand matrix is block-diagonal, and thus the linear system can be split into K smaller systems, each to be solved individually. In the case that boundary conditions concern only a single field apiece, this remains true, and one can still separate the system in this manner. The intuitive implication of this is the independence of polynomial expansions approximating each field in the absence of any constraints which would otherwise link them.

However, if boundary conditions impose some particular relationship between fields, a polynomial approximation of one of these fields will require information regarding the other fields present in the boundary condition for the condition to be respected. As in the single-field case, each boundary condition will be approximated with Taylor-series expansions, although note that in this case, each field present within the constraint will have its own expansion. Each of these series contains derivatives of its respective function, contained within the corresponding δ_k as in Equation 2.31. As such, the Taylor-series approximation of a constraint specifying some relationship between multiple fields will contain derivatives of multiple fields, and thus when represented as a dot-product, the vector δ will consist of multiple δ_k .

In the context of the linear system shown in Equation 2.31, rows corresponding to such boundary conditions will span multiple previously separate blocks, thereby linking them. As the process of fitting the extrapolant relies on the inversion of the left-hand-side matrix, it follows that these linked blocks will need to be inverted in tandem as they are no longer separated. As a boundary condition row in such a scenario maps multiple δ_k onto a single boundary forcing value, and each of the previously separate blocks maps its respective δ_k onto the corresponding $f_k(\mathbf{X}_k)$, the inverse of the resultant block will consequently map all $f_k(\mathbf{X}_k)$ present onto any given derivative.

As a simple example, consider some pair of 1D functions f and h upon which the condition

$$f(x_b) + h(x_b) = 0, \quad (2.32)$$

is to be applied at the boundary. Approximating this constraint with 2nd-order Taylor series expanded around some x_0 yields

$$\begin{aligned} & f(x_b) + (x_b - x_0) \frac{\partial f}{\partial x}(x_b) + \frac{(x_b - x_0)^2}{2} \frac{\partial^2 f}{\partial x^2}(x_b) \\ & + h(x_b) + (x_b - x_0) \frac{\partial h}{\partial x}(x_b) \\ & + \frac{(x_b - x_0)^2}{2} \frac{\partial^2 h}{\partial x^2}(x_b) = 0. \end{aligned} \quad (2.33)$$

Representing this as a dot product of two vectors

$$\begin{pmatrix} \mathbf{a}_f \\ \mathbf{a}_h \end{pmatrix} \cdot \begin{pmatrix} \delta_f \\ \delta_h \end{pmatrix} = 0 \quad (2.34)$$

where

$$\mathbf{a}_f^T = \mathbf{a}_h^T = \left(1, (x_b - x_0), \frac{(x_b - x_0)^2}{2} \right), \quad (2.35)$$

$$\delta_f^T = \left(f(x_b), \frac{\partial f}{\partial x}(x_b), \frac{\partial^2 f}{\partial x^2}(x_b) \right), \quad (2.36)$$

and

$$\delta_h^T = \left(h(x_b), \frac{\partial h}{\partial x}(x_b), \frac{\partial^2 h}{\partial x^2}(x_b) \right). \quad (2.37)$$

To avoid confusion, note that $\mathbf{a}_f = \mathbf{a}_h$ results from the boundary condition applied and will generally not be the case, depending upon on constraints

applied. A single point of f in this form is given as

$$\begin{pmatrix} \mathbf{a}_f \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \delta_f \\ \delta_h \end{pmatrix} = 0. \quad (2.38)$$

Approximations of h at interior points can be formed in a similar manner. Assembling values of f and h at some set of points into a linear system using the above methodology, it is apparent that the row corresponding to the boundary condition links two otherwise separate blocks pertaining to interior values of f and h .

The boundary serves both to truncate the support region (as no data lies beyond it) and to introduce additional constraints to the polynomial fitting. This profoundly alters the linear system that must be solved to fit the polynomial, particularly where boundary conditions linking fields are present: in the case of the particle velocity boundary condition for the free surface for example.

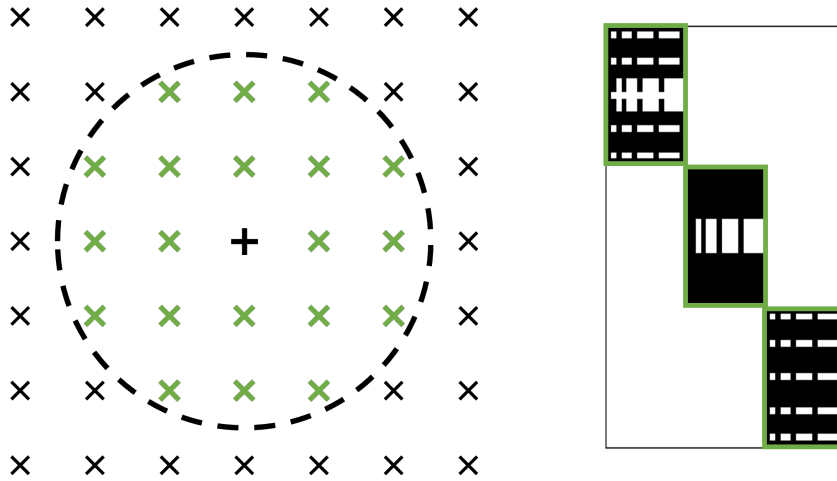


Figure 2.6.: A circular support region in free space and its corresponding matrix structure for the first-order acoustic wave equation (nonzero elements are black, zero elements are white). Staggered particle velocity subgrids are omitted to prevent excessive cluttering. Individual blocks are highlighted in green, and from top-left to bottom-right correspond to pressure, horizontal particle velocity, and vertical particle velocity. It is clear that this matrix can be split into three smaller systems which can be solved individually; each polynomial can be fitted independently of the others.

In free space, the stencil has an uninterrupted circular footprint comprising entirely of interior points. The corresponding matrix has a block-

diagonal structure, as shown in Figure 2.6, with each block corresponding to one of the fields. Each block can be inverted separately, meaning that the resultant polynomial extrapolations have no impact on one another. Note that splitting the matrix up in this manner considerably reduces the computational cost of finding the matrix inverse (or pseudoinverse as required), and it is thus desirable to do so where possible.

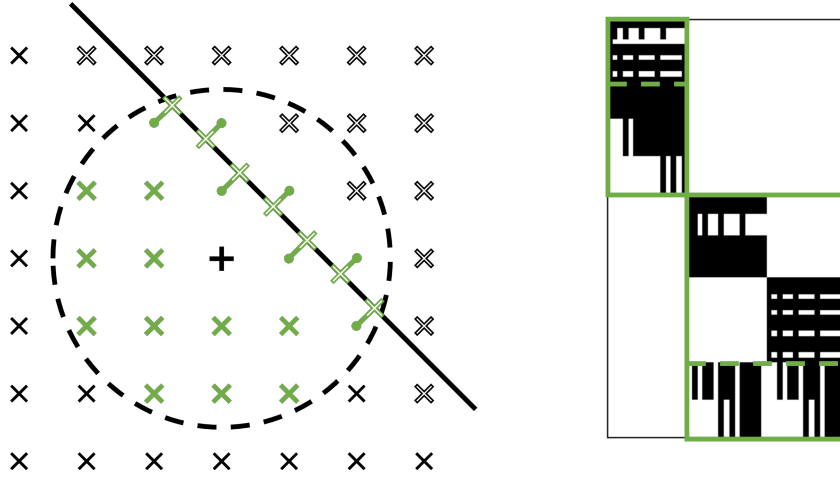


Figure 2.7.: A circular support region cut by a diagonal boundary and its corresponding matrix structure for the first-order acoustic wave equation (nonzero elements are shown in black). Staggered particle velocity subgrids are omitted to minimise clutter. Blocks are highlighted with solid green boxes, with the dashed green lines separating rows corresponding to interior points and those corresponding to boundary conditions. Note that the previously separate particle velocity blocks have become merged, as the zero-divergence boundary condition (and associated higher-order conditions) span both particle velocity fields.

Figure 2.7 shows the effect of inserting a 45° free-surface boundary cutting across the support region: both the internal structure of the blocks and the overarching structure of the matrix are altered. Notably, inclusion of particle velocity boundary conditions merges the corresponding blocks, meaning that they must be inverted in tandem, and resulting polynomial extrapolations will be dependent on both particle velocity fields. As the pressure boundary conditions only concern the pressure field, this block retains its independence as before, although some rows are lost due to corresponding to now-external points, and several additional rows are added by the introduction of boundary conditions.

Whilst the matrices corresponding to such support regions will have consistently overdetermined blocks in free space, this may not be the case

when boundaries are introduced. If a block becomes underdetermined, the radius of the support regions for functions contained therein can be expanded, thereby adding additional rows to the block until sufficient information is present.

Solving the resultant linear system gives derivatives of the basis (now multiple bases) at the expansion point as in the single-field case. Given these, the function can be continued beyond the boundary to obtain values at exterior points required by interior stencils. This is achieved in the same manner as the single-field case. These function values will be some weighted sum of boundary forcing values and interior values of all fields linked to the function of interest via boundary constraints. Consequently, any stencil formed using these approximated values will span all of these fields as well.

With this approach, the need to devise application-specific approximations is removed, enabling the application of a consistent method across a wide range of boundary conditions. As such, for some given set of derivative operators and boundary conditions (within the parameters discussed above), a scheme of this class can be generated.

2.4. Summary

A generalised approach to immersed boundary formulation enables a consistent methodology to be applied across a wide range of equations without a need to derive a new treatment for each. By defining an overarching class and a formulation for schemes of this class, immersed boundary treatments can be rapidly obtained for a variety of applications, and the foundations are laid for automation of this approach.

The approach developed is applicable across any number of dimensions, for arbitrary-order accuracy, and can be applied to essentially any base stencil. This is essential to achieve the application-agnosticism required for the abstraction of the mathematical method. Furthermore, the method intentionally avoids the need for knowledge of the governing application at the low level. This should be reflected in the high-level specification of the scheme: the boundary conditions imposed and the stencils to which the treatment is to be applied. Formulation of a general constraint ensures that the approach can be used to impose a wide range of bound-

ary conditions of interest, and that it can be straightforwardly extended to include conditions which specify some relationship between multiple fields.

The use of hyperspherical support regions for the extrapolant and multiple boundary points at which constraints are applied ensures that the extrapolations formed reflect not only boundary position but also shape and curvature.

With the approach proposed, it is possible to raise the level of abstraction, allowing immersed boundary treatments to be specified independently of the governing equations. The treatment is characterised by surface geometry, underlying discretisation, and the set of boundary conditions imposed. Whilst these reflect the application at hand, application information is not required on the level of creating the immersed boundary treatment itself.

Chapter 3

Immersed Boundaries and the 2nd-Order Acoustic Wave Equation

Whilst unable to capture propagation of shear phases, phase conversions at interfaces, and other phenomena including anisotropy, shear-wave splitting, and polarity effects, the second-order acoustic wave equation remains useful in infrasound modelling [Kim and Lees, 2014] and medical imaging [Guasch et al., 2020].

Whilst extension to incrementally more complex wave equations has been explored in the past, notably by the series of Mulder [2017], Mulder and Huiskes [2017], Huiskes et al. [2017], use of 1D extrapolants hindered application of the boundary treatment to cross-derivatives. The common antisymmetric mirroring approach [Li and Yao, 2020] suffers similar issues, failing to consider the physics behind the behaviour implemented, hindering extension to other conditions. The method outlined by Chapter 2 is designed to be agnostic to both boundary conditions and underlying discretization. Free and rigid surfaces represent disparate boundary conditions; the approach developed should be equally applicable to both and this demonstration indicates its wider applicability.

Demonstrations of free and rigid immersed surfaces are presented, and their accuracy is assessed through comparisons to existing approaches and convergence testing. The approach is then applied to a selection of 2D and 3D test cases featuring realistic geometry to validate its suitability

in practical applications.

3.1. The Acoustic Wave Equation and its Discretization

In seismic modelling contexts, Equation 1.25 is typically discretized with an explicit timestepping scheme, replacing the time derivative with a second-order central difference approximation:

$$\frac{p^{t+1} - 2p^t + p^{t-1}}{\Delta t^2} = c^2 \nabla^2 p. \quad (3.1)$$

Higher-order timestepping tends to be unstable. Rearranging for the forward timestep

$$p^{t+1} = 2p^t - p^{t-1} + \Delta t^2 c^2 \nabla^2 p \quad (3.2)$$

is obtained. This is the update equation used to estimate the field at the next timestep with each iteration.

The Laplacian is discretized using central difference approximations: such approximations of second derivatives are not prone to ringing or checkerboarding [Corrêa et al., 2002]. The immersed boundary treatment will be applied to the individual second-derivative stencils which form this operator.

3.2. The Free Surface

The computational domain for geophysical wavefield modelling problems is typically a half-space with either the upper or lower surface corresponding to that of the Earth; boundary conditions imposed on this surface correspond with the material discontinuity. For waves propagating in the subsurface, the much lower impedance of air results in a reflection coefficient of approximately -1. This condition is commonly imposed through antisymmetric mirroring of pressure across the surface, creating an inward-propagating wave with reversed polarity, interfering with the outgoing wave to enforce zero pressure on the boundary.

Given the condition

$$p(t, \mathbf{x}_b) = 0 \quad (3.3)$$

the governing equations imply additional, higher-order boundary conditions; imposing these conditions improves stability of boundary treatments based on higher-order polynomials Mulder [2017]. Taking the second derivative of Equation 3.3 with respect to time and substituting Equation 1.25, one obtains

$$\nabla^2 p(t, \mathbf{x}_b) = 0. \quad (3.4)$$

Repeating this process for Equation 3.4 and assuming locally-constant c_p yields

$$\nabla^4 p(t, \mathbf{x}_b) = 0; \quad (3.5)$$

this can be iterated indefinitely to obtain incrementally higher-order boundary conditions of the form

$$\nabla^{2m} p(t, \mathbf{x}_b) = 0 \quad m \in \mathbb{Z}^+. \quad (3.6)$$

Note that conditions of higher order than the approximating basis contain no information, corresponding to the trivial expression $0 = 0$.

3.2.1. Polynomial Approximation of Free-Surface Constraints

Boundary constraints are formed by substituting the polynomial basis into the boundary conditions; characteristic rows of the linear system corresponding to each of the above boundary conditions up to the order of the polynomial basis can be obtained. For the free surface, each boundary point will correspond with $M/2 + 1$ rows in the linear system.

Considering a fourth-order discretization in 2D, the corresponding basis

approximating p is given by

$$\begin{aligned}
\tilde{p}(t, x, y) \approx & p(t, x_0, y_0) + (x - x_0) \frac{\partial p}{\partial x}(t, x_0, y_0) \\
& + (y - y_0) \frac{\partial p}{\partial y}(t, x_0, y_0) + \frac{(x - x_0)^2}{2} \frac{\partial^2 p}{\partial x^2}(t, x_0, y_0) \\
& + (x - x_0)(y - y_0) \frac{\partial^2 p}{\partial x \partial y}(t, x_0, y_0) + \frac{(y - y_0)^2}{2} \frac{\partial^2 p}{\partial y^2}(t, x_0, y_0) \\
& + \frac{(x - x_0)^3}{6} \frac{\partial^3 p}{\partial x^3}(t, x_0, y_0) + \frac{(x - x_0)^2(y - y_0)}{2} \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0) \\
& + \frac{(x - x_0)(y - y_0)^2}{2} \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0) + \frac{(y - y_0)^3}{6} \frac{\partial^3 p}{\partial y^3}(t, x_0, y_0) \\
& + \frac{(x - x_0)^4}{24} \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0) + \frac{(x - x_0)^3(y - y_0)}{6} \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + \frac{(x - x_0)^2(y - y_0)^2}{4} \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + \frac{(x - x_0)(y - y_0)^3}{6} \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0) + \frac{(y - y_0)^4}{24} \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0).
\end{aligned} \tag{3.7}$$

Substituting into Equation 3.3 yields

$$\begin{aligned}
0 = & p(t, x_0, y_0) + (x_b - x_0) \frac{\partial p}{\partial x}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial p}{\partial y}(t, x_0, y_0) + \frac{(x_b - x_0)^2}{2} \frac{\partial^2 p}{\partial x^2}(t, x_0, y_0) \\
& + (x_b - x_0)(y_b - y_0) \frac{\partial^2 p}{\partial x \partial y}(t, x_0, y_0) + \frac{(y_b - y_0)^2}{2} \frac{\partial^2 p}{\partial y^2}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)^3}{6} \frac{\partial^3 p}{\partial x^3}(t, x_0, y_0) + \frac{(x_b - x_0)^2(y_b - y_0)}{2} \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)(y_b - y_0)^2}{2} \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0) + \frac{(y_b - y_0)^3}{6} \frac{\partial^3 p}{\partial y^3}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)^4}{24} \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0) + \frac{(x_b - x_0)^3(y_b - y_0)}{6} \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)^2(y_b - y_0)^2}{4} \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)(y_b - y_0)^3}{6} \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0) + \frac{(y_b - y_0)^4}{24} \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0).
\end{aligned} \tag{3.8}$$

Similarly Equations 3.4 and 3.5 yield

$$\begin{aligned}
0 = & \frac{\partial^2 p}{\partial x^2}(t, x_0, y_0) + \frac{\partial^2 p}{\partial y^2}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^3 p}{\partial x^3}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^3 p}{\partial y^3}(t, x_0, y_0) + \frac{(x_b - x_0)^2}{2} \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0) \\
& + (x_b - x_0)(y_b - y_0) \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + ((x_b - x_0)^2 + (y_b - y_0)^2) \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + (x_b - x_0)(y_b - y_0) \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0) + \frac{(y_b - y_0)^2}{2} \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0),
\end{aligned} \tag{3.9}$$

and

$$0 = \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0) + 2 \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) + \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0), \tag{3.10}$$

respectively.

In the context of the dot product given in Equation 2.2,

$$\begin{aligned}
\delta^T = & \left(p(t, x_0, y_0), \frac{\partial p}{\partial x}(t, x_0, y_0), \frac{\partial p}{\partial y}(t, x_0, y_0), \frac{\partial^2 p}{\partial x \partial y}(t, x_0, y_0), \right. \\
& \frac{\partial^2 p}{\partial y^2}(t, x_0, y_0), \frac{\partial^3 p}{\partial x^3}(t, x_0, y_0), \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0), \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0), \\
& \frac{\partial^3 p}{\partial y^3}(t, x_0, y_0), \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0), \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0), \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0), \\
& \left. \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0), \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0) \right).
\end{aligned} \tag{3.11}$$

For Equation 3.3, $\mathbf{a}$ is given by

$$\mathbf{a}^T = \left(1, (x_b - x_0), (y_b - y_0), \frac{(x_b - x_0)^2}{2}, (x_b - x_0)(y_b - y_0), \frac{(y_b - y_0)^2}{2}, \frac{(x_b - x_0)^3}{6}, \frac{(x_b - x_0)^2(y_b - y_0)}{2}, \frac{(x_b - x_0)(y_b - y_0)^2}{2}, \frac{(y_b - y_0)^3}{6}, \frac{(x_b - x_0)^4}{24}, \frac{(x_b - x_0)^3(y_b - y_0)}{6}, \frac{(x_b - x_0)^2(y_b - y_0)^2}{4}, \frac{(x_b - x_0)(y_b - y_0)^3}{6}, \frac{(y_b - y_0)^4}{24} \right), \quad (3.12)$$

whilst Equations 3.4 and 3.5 correspond to

$$\mathbf{a}^T = \left(0, 0, 0, 1, 0, 1, (x_b - x_0), (y_b - y_0), (x_b - x_0), (y_b - y_0), \frac{(x_b - x_0)^2}{2}, (x_b - x_0)(y_b - y_0), (x_b - x_0)^2 + (y_b - y_0)^2, (x_b - x_0)(y_b - y_0), \frac{(y_b - y_0)^2}{2} \right), \quad (3.13)$$

and

$$\mathbf{a}^T = \left(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 1 \right), \quad (3.14)$$

respectively.

With these, three rows are constructed for each boundary point used to constrain the basis. Note that the row corresponding with the fourth-order condition is invariant in space, containing the same information regardless of boundary point location.

3.2.2. Application to Example Geometry

To illustrate this process in a practical case, consider the case of a fourth-order stencil approximating $\frac{\partial^2 p}{\partial y^2}$, truncated by an arc-shaped boundary, as shown in Figure 3.1. It is clear that it will not be possible to form this stencil as one would in free space as two requisite points are outside the function domain.

To rectify this, a circular support region of radius $2.5\Delta x$ is extended from the stencil centre, as in Figure 3.7, containing 5 boundary points and 11 interior points. Each of these boundary points will have three matrix

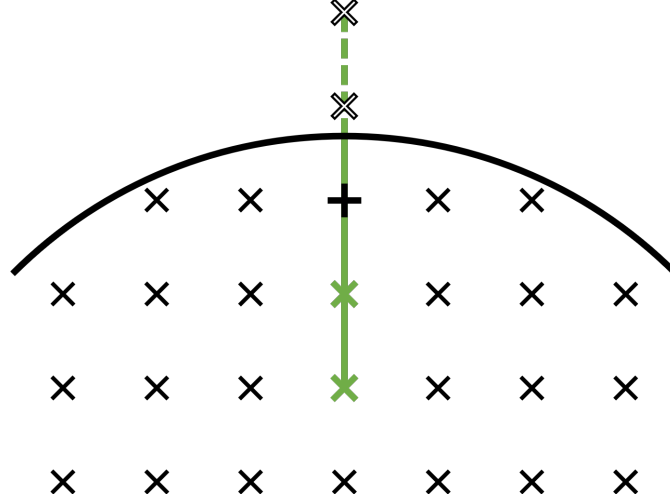


Figure 3.1.: The stencil footprint of a 4th-order-accurate centred $\frac{\partial^2 p}{\partial y^2}$ stencil truncated by an arc-shaped boundary. The bold black cross is the stencil position, whilst pale green crosses are other interior points within the stencil. Values at both hollow black crosses are the both required by the stencil, but are located outside the computational domain.

rows associated with it, one for each boundary condition imposed at that point.

Assembling boundary conditions at all points, 15 rows of boundary constraints are obtained, shown in Figure 3.3. Note however that these constraints are not necessarily unique (most prominently the zero biharmonic condition as aforementioned), and may contain overlapping information. A cutoff parameter of $\eta = 0.5$ for the pressure field was empirically determined to ensure stability when imposing the free surface boundary condition, as in Mulder [2017]. Whilst rigorous analysis would be beneficial, Mulder and Huiskes [2017] found this to be extremely impractical for all but the smallest of problems. In 1D, the cutoff specified in Mulder [2017] is equivalent to that used in this work. Eligible points within the support region are used to form the main body of the matrix shown in Figure 3.4.

Assembling these two submatrices, the resultant matrix has more rows than columns. The Moore-Penrose pseudoinverse is calculated, its structure shown in Figure 3.5. A weighted-least-squares approach was explored to prioritise points close to the boundary and lower-order boundary conditions, although this was not found to yield any discernible ben-

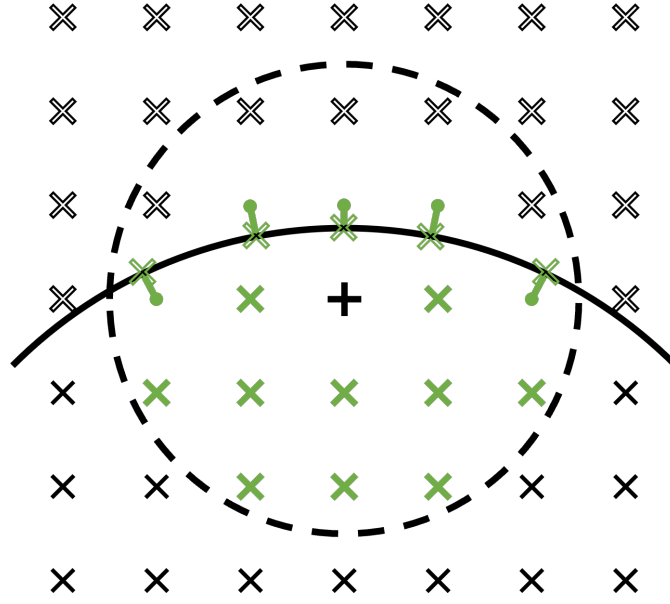


Figure 3.2.: Support region for a 2D polynomial fitted with a combination of available interior points and boundary points.

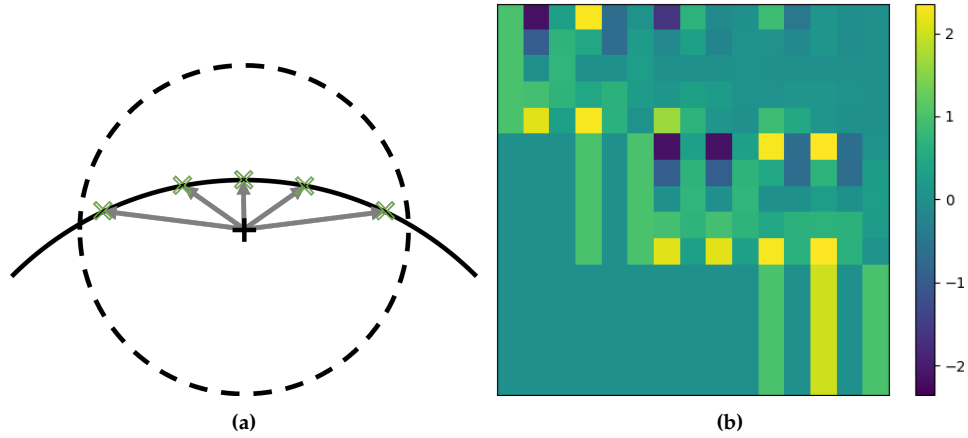


Figure 3.3.: The 5 boundary points within the support region and structure of the corresponding submatrix.

efit and was prone to producing extremely ill-conditioned linear systems.

To project the basis fitted onto the exterior points required by the interior stencil, the matrix $\mathbf{B}$ is formed by evaluating Equation 3.7 at each. The matrix $\mathbf{BA}^+$ maps the function values at the interior points within the support region and boundary forcing values onto extrapolated function values at the two required exterior points, and its structure is shown in Figure 3.6. Note that the last 15 columns of this matrix correspond to

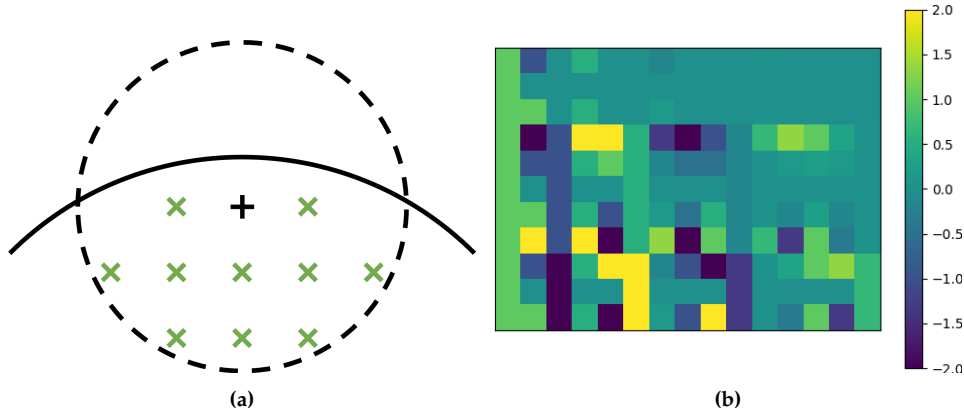


Figure 3.4.: The 11 interior points within the support region and structure of the corresponding submatrix.

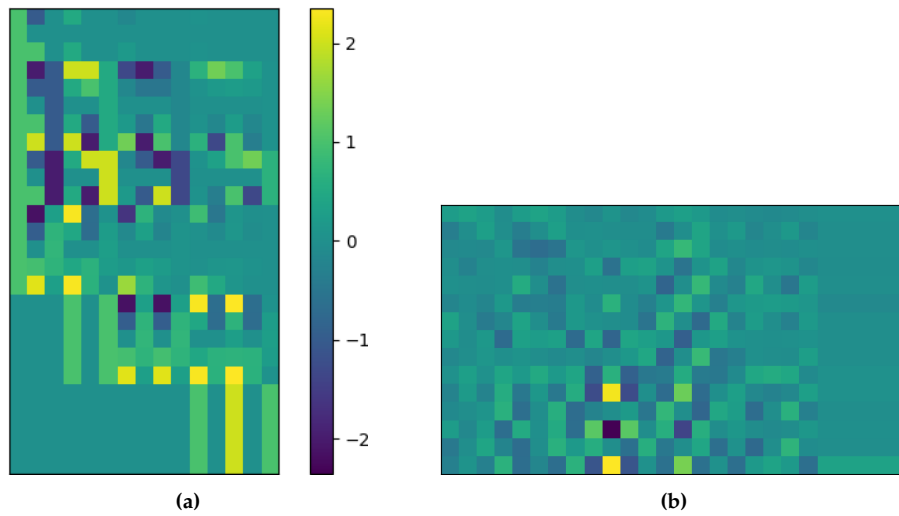


Figure 3.5.: The assembled matrix A (left) and its Moore-Penrose pseudoinverse A^+ (right), the latter of which maps various function values and forcing values onto derivatives.

the boundary forcing values, all of which are zero, and can thus be discounted. The first 11 columns contain weights for the operators extrapolating the function onto the exterior points. These are used to complete the interior stencil, yielding the modified stencil shown in Figure 3.7.

It is apparent that in this case, the boundary modification of the stencil only slightly alters the weights of interior points in the original stencil, as the weights associated with the extrapolation are relatively small compared to the interior stencil weights. However, the footprint of the stencil



Figure 3.6.: A visualisation of the matrix $\mathbf{BA}^+$, mapping interior values and boundary forcing values onto the two requisite exterior points.

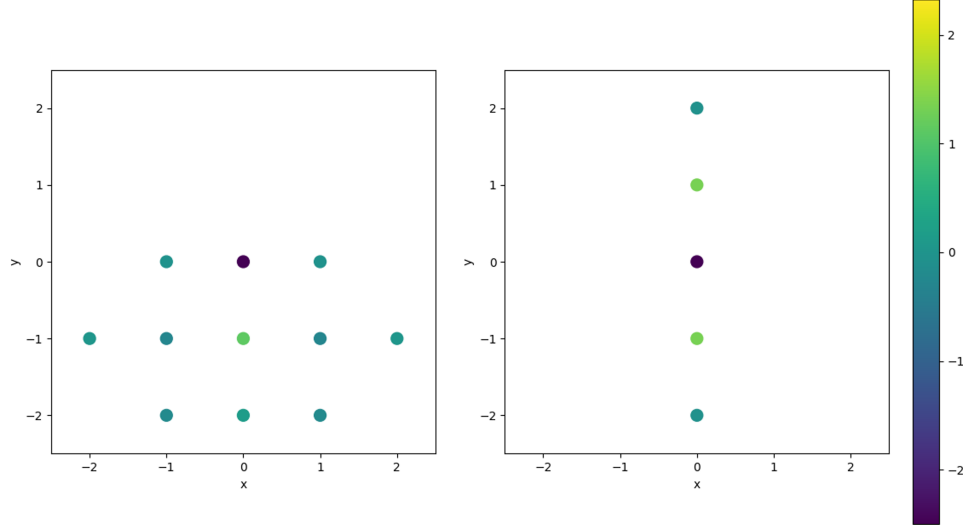


Figure 3.7.: Comparison of the unmodified $\frac{\partial^2 p}{\partial y^2}$ stencil and its boundary-modified counterpart for the arc-shaped free-surface boundary shown above.

is changed significantly with additional interior points, albeit with relatively small weights.

3.2.3. Reduction to 1D

Through particular choices made when applying the method described, other, equally feasible stencils can be obtained, including the 1D approximation detailed in Mulder [2017]. Considering the 1D case, the aforementioned free-surface boundary conditions reduce to

$$p(t, x_b) = 0, \quad \frac{\partial^2 p}{\partial x^2}(t, x_b) = 0, \quad \frac{\partial^4 p}{\partial x^4}(t, x_b) = 0, \quad \dots \quad (3.15)$$

and so forth. Now suppose some case is encountered where a boundary truncates a stencil as in 3.1. Selecting the $M/2$ closest points to the boundary at distance greater than $\eta\Delta x$, $\mathbf{A}$ can be constructed such that it is equidetermined, enabling derivatives to be obtained via an inverse.

Thus the method described by Mulder [2017] can be considered a special case of the overarching method described.

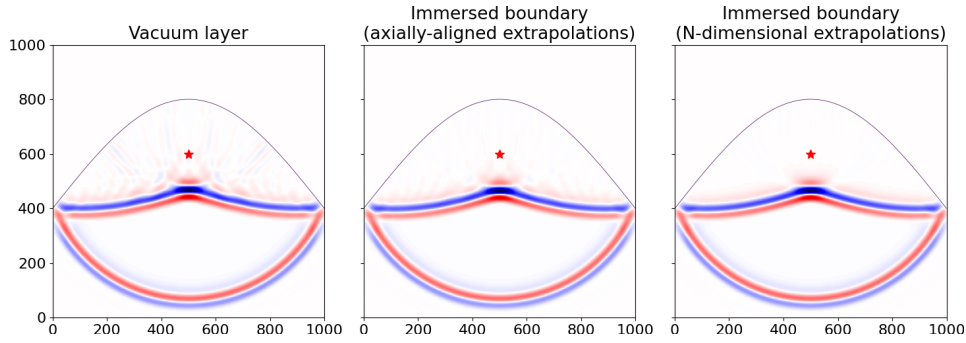


Figure 3.8.: Snapshots at 700ms of a wavefront reflecting off a sinusoidal hill upon which a free surface has been imposed. The leftmost subfigure features a free surface implemented with a vacuum layer approach, whilst the others feature immersed boundary treatments. The middle subfigure features an immersed boundary based on axially-aligned 1D extrapolations as in Mulder [2017], whilst in the rightmost figure, the immersed boundary is implemented using N-dimensional extrapolations with a circular support region. A Ricker source was injected at the point labelled with a red star. Wavefield amplitudes are normalised against the maximum absolute value in each subfigure.

Figure 3.8 compares the proposed immersed boundary treatment based on an N-dimensional basis with circular support on one based on axially-aligned 1D extrapolations and a naive vacuum approach for a 2D second-order acoustic example featuring a sinusoidal free surface. The vacuum layer is far noisier than either immersed boundary treatment, producing pronounced spurious diffractions which pollute the wavefield considerably. These are mostly suppressed by the immersed boundary treatment based on 1D extrapolations, although some minor artefacts are visible immediately behind the wavefront, these being entirely absent in the case where N-dimensional extrapolations are used. N-dimensional extrapolants yield visually-superior results to the other two approaches, implying improved representation of the surface geometry and boundary conditions imposed. Note that 1D extrapolations were found to produce bright spots and significant artefacts when used to implement the rigid surface.

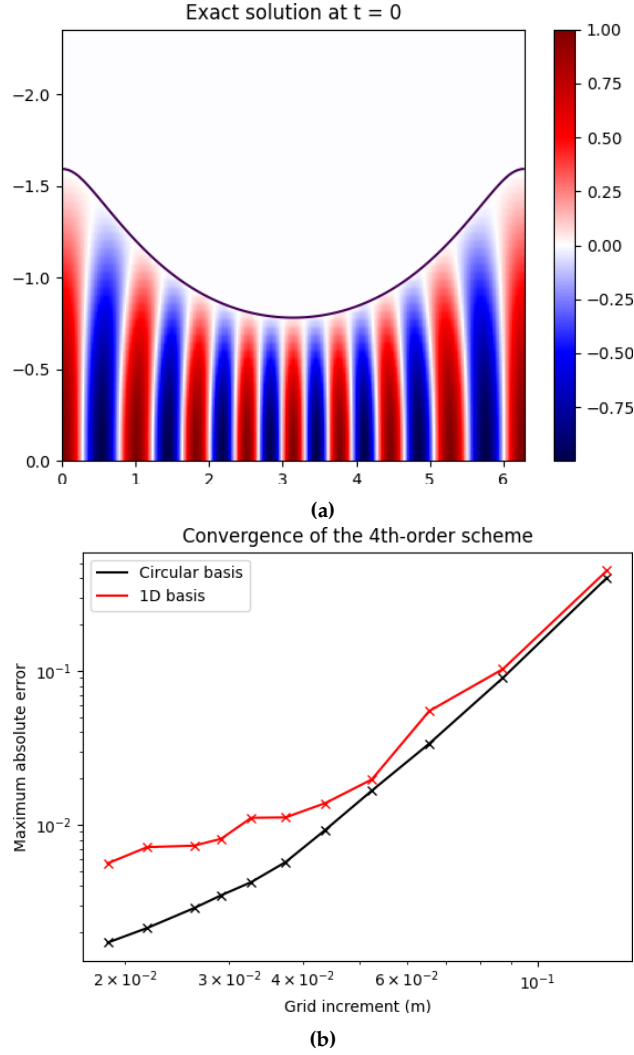


Figure 3.9.: The exact solution at $t = 0$ is shown in the Subfigure 3.9a. The solid black line is the free-surface, the left and right sides of the domain have periodic boundary conditions applied, and the wavefield is mirrored across the lower boundary. A comparison of the convergence behaviour of the method proposed to one using 1D approximations is shown on the right.

3.2.4. Convergence

To examine convergence behaviour of the proposed approach for the free-surface, a setup initially presented by Mulder [2017], subsequently used in Mulder and Huiskes [2017] was replicated. An exact solution exists for this example, allowing for the error in any numerical solution to be evaluated; the solution at $t = 0$ is shown in Figure 3.9a.

The pressure field is given as

$$p(t, x, y) = \cos(mx_1 - \alpha t) \cos\left(\frac{\pi y_1}{2}\right), \quad (3.16)$$

where

$$x_1 = x - A \sin(x) \cosh(y), \quad (3.17)$$

and

$$y_1 = y - A \cos(x) \sinh(y). \quad (3.18)$$

The parameter α is given by

$$\alpha = c_0 \sqrt{m^2 + \left(\frac{\pi}{2}\right)^2}, \quad (3.19)$$

whilst the celerity field is given as

$$c(x, y) = c_0 \left((1 - A \cos(x) \cosh(y))^2 + (A \sin(x) \sinh(y))^2 \right)^{\frac{1}{2}}. \quad (3.20)$$

In the convergence tests shown, this solution was parameterised with $c_0 = 1$, $m = 8$, and $A = \frac{1}{4}$. The free surface is defined by $y_1 = -1$ and the solution is periodic in $x \in [0, 2\pi)$. Zero pressure is imposed at $y = 0$.

Figure 3.9 shows convergence behaviour of a scheme based on ND extrapolations, comparing against a scheme based on axially-aligned 1D extrapolations.

For a fourth-order spatial discretization, the reduction in observed error with respect to grid increment was found to be initially just short of fourth-order, flattening around a grid increment of 0.02 as the spatial error is eclipsed by second-order timestepping error for finer grids. As the timestep was set at 10% of the critical value, this implies that the error introduced by the immersed boundary was minimal in all cases and that in many cases, topography implementation will cease to be the accuracy bottleneck when this immersed boundary treatment is applied. Reducing the timestep enabled the continuation of the approximate fourth-order trend to finer grids, but accumulation of floating-point error again resulted in a similar flattening albeit at a smaller grid increment. The immersed boundary approach based on N-dimensional extrapolations was found to yield reduced error versus that of Mulder [2017] for all grid increments tested,

particularly at finer resolutions, albeit with similar convergence behaviour.

3.3. The Rigid Surface

Considering a wave approaching the surface from the opposite direction, topography acts as a rigid surface, with a reflection coefficient of approximately 1. Given Equation 1.22, one would expect particle velocity to become very small when transitioning from a medium with small values of ρ to one in which it is much greater. This can be approximated by imposing zero particle velocity for an incident acoustic wave propagating in the air. This approach is commonly used for modelling infrasound interaction with topography [Heimann and Karle, 2006, Lacanna et al., 2014]. Acoustic-seismic coupling represents a fairly small energy transfer [Novoselov et al., 2020], and the approximation avoids timestep reduction associated with large subsurface wavespeeds.

The rigid surface condition is given as

$$\frac{\partial p}{\partial x_i}(t, \mathbf{x}_b) = 0. \quad (3.21)$$

As with the free surface, higher-order conditions can be derived. Taking the second derivative of Equation 3.21 with respect to time yields

$$\frac{\partial}{\partial x_i} \frac{\partial^2 p}{\partial t^2}(t, \mathbf{x}_b) = 0. \quad (3.22)$$

Substituting Equation 1.25 and assuming locally constant c_p results in

$$\nabla^2 \frac{\partial p}{\partial x_i}(t, \mathbf{x}_b) = 0; \quad (3.23)$$

this process can once again be repeated indefinitely.

Considering the rigid surface in 2D, approximating p with the 4th-order

Taylor series given in Equation 3.7, the $\frac{\partial p}{\partial x}$ constraint becomes

$$\begin{aligned}
0 = & \frac{\partial p}{\partial x}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^2 p}{\partial x^2}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^2 p}{\partial x \partial y}(t, x_0, y_0) + \frac{(x_b - x_0)^2}{2} \frac{\partial^3 p}{\partial x^3}(t, x_0, y_0) \\
& + (x_b - x_0)(y_b - y_0) \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0) + \frac{(y_b - y_0)^2}{2} \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)^3}{6} \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0) + \frac{(x_b - x_0)^2(y_b - y_0)}{2} \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)(y_b - y_0)^2}{2} \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) + \frac{(y_b - y_0)^3}{6} \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0),
\end{aligned} \tag{3.24}$$

whilst the $\frac{\partial p}{\partial y}$ constraint is similarly

$$\begin{aligned}
0 = & \frac{\partial p}{\partial y}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^2 p}{\partial x \partial y}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^2 p}{\partial y^2}(t, x_0, y_0) + \frac{(x_b - x_0)^2}{2} \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0) \\
& + (x_b - x_0)(y_b - y_0) \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0) + \frac{(y_b - y_0)^2}{2} \frac{\partial^3 p}{\partial y^3}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)^3}{6} \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) + \frac{(x_b - x_0)^2(y_b - y_0)}{2} \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)(y_b - y_0)^2}{2} \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0) + \frac{(y_b - y_0)^3}{6} \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0).
\end{aligned} \tag{3.25}$$

Taking the Laplacian of each to obtain the third-order boundary conditions yields

$$\begin{aligned}
0 = & \frac{\partial^3 p}{\partial x^3}(t, x_0, y_0) + \frac{\partial^3 p}{\partial x \partial y^2}(t, x_0, y_0) \\
& + (x_b - x_0) \frac{\partial^4 p}{\partial x^4}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + (x_b - x_0) \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0)
\end{aligned} \tag{3.26}$$

and

$$\begin{aligned}
0 = & \frac{\partial^3 p}{\partial x^2 \partial y}(t, x_0, y_0) + \frac{\partial^3 p}{\partial y^3}(t, x_0, y_0) \\
& + (x_b - x_0) \frac{\partial^4 p}{\partial x^3 \partial y}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + (x_b - x_0) \frac{\partial^4 p}{\partial x \partial y^3}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^4 p}{\partial y^4}(t, x_0, y_0)
\end{aligned} \tag{3.27}$$

respectively. This corresponds with four characteristic rows of $\mathbf{A}$ associated with each boundary point for a 4th-order discretization. Unlike the free surface, the row corresponding with the highest-order boundary condition will be variable in space, reducing the amount of redundant information in the matrix by comparison.

3.4. Application to Real-World Topography

The first real-world topography example is based on an East-West profile across the summit of Mount St Helens, Washington, USA. Summit collapse during the 1980 eruption and subsequent lava dome formation within the crater resulted in near-vertical crater walls and a mixture of concavity and convexity on the crater floor. As a stratovolcano, Mount St Helens has steep, uneven flanks rising over a kilometer from the surrounding landscape, making it an ideal stress test for the method proposed.

The signed-distance function (SDF) representing the topography was generated from a publicly available digital elevation model (DEM) obtained by the Shuttle Radar Topography Mission. This representation enables the surface to be located with much finer precision than the FD discretization, despite its discretization on the same grids, and the SDF can be constructed from extremely high-resolution digital elevation models (DEMs) without the requirement to downsample or alias the raw data to match the FD grid. This is beneficial for real-world applications where such data obtained from satellites and drones may be structured or unstructured (or compounds of multiple such datasets), heavily oversampled versus the discretization required for numerical-dispersion-free wavefield propagation, and often with much greater vertical precision. In this case, the DEM was obtained with a lateral resolution of one second of arc,

approximately equalling 30m at this latitude, although north-south and east-west resolutions are unequal. However, sampling for both was sufficient to calculate the respective SDF, discretized with an even 30m grid spacing in all dimensions.

The overall model domain is 5.1km in height, and 9.6km in width (for all horizontal dimensions). Material properties are kept consistent throughout the model domain, such that all topographic interactions observed are a product of the boundary treatment rather than any material discontinuity: a convention continued throughout the examples presented. Variable material parameters are a separable concern and are essentially trivial to implement (particularly with the abstraction layers used in this study). Discussion of the effects of material heterogeneity versus topographic variation can be found in Takemura et al. [2015]. Furthermore, their introduction runs the risk of inconsistency between implicit interfaces represented by material parameters and the explicit interface encapsulated by the immersed boundary. On this basis, it is recommended that the transition across the surface should be encapsulated solely by the boundary treatment and the contrast in material parameters at this surface should be omitted.

3.4.1. 2D 2nd-order acoustic free-surface

As aforementioned, a free-surface boundary condition is imposed on the surface. The remaining edges of the computational domain have zero Dirichlet boundary conditions imposed for simplicity. In practice, one would apply absorbing boundary conditions along these edges, but as with variable material parameters, this is a separable concern and straightforwardly combined with the method presented in this work.

A Ricker wavelet with a peak frequency of 8Hz is injected below the lava dome, at an elevation of 1250m: a location loosely reminiscent of tremors induced by shallow magma movement within the conduit. Placing the source close to the surface maximises interaction between wavefield and topography. The spatial discretization used is fourth-order accurate: a precedent continued to all other examples in this chapter.

We see in Figure 3.10 that the uneven topography results in several distinct reflections, with further minor reverberations trailing the main

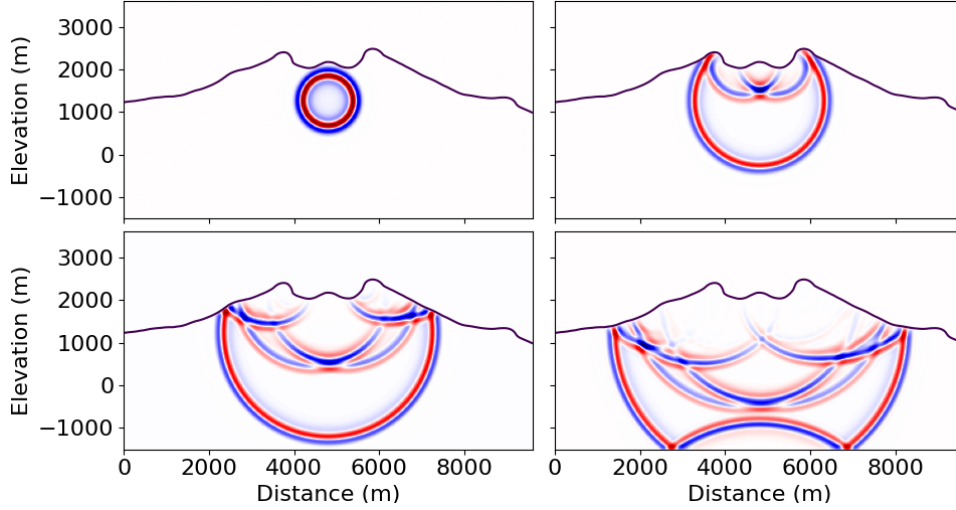


Figure 3.10.: Snapshots of the wavefield interacting with a free surface at 375ms, 750ms, 1125ms, and 1500ms. Celerity is 2.5km/s throughout the model. The black line designates the isosurface $s(x, y) = 0$ on the SDF, coinciding with the surface. The wavefield shown in each snapshot is normalised by the maximum absolute value in that snapshot.

wavefront. Also apparent is the focusing and defocusing effect of concave and convex topography respectively, and the diffraction of the wavefront around obstructions. This yields a much more complex series of arrivals than would be observed for a flat surface, although the horizontally propagating wavefront is only mildly distorted, in agreement with previously published findings [Takemura et al., 2015].

3.4.2. 2D 2nd-order acoustic rigid-surface

As previously discussed, for acoustic waves propagating in the air, the Earth represents an extremely dense and essentially immobile surface: particle velocity is zero at the interface, this being a rigid surface. The Mount St Helens profile used in the previous example once again features, although the surface now forms the lower bound of the domain.

In an imitation of a typical infrasound propagation scenario, a 1Hz Ricker source is placed at an elevation of 2600m above the lava dome. Located only slightly above the crater rim, this location was chosen to capture reverberation within and diffraction around the crater rim.

Figure 3.11 shows the propagating wavefield, including the multiple distinct reflected wavefronts from the crater walls and floor. Note the

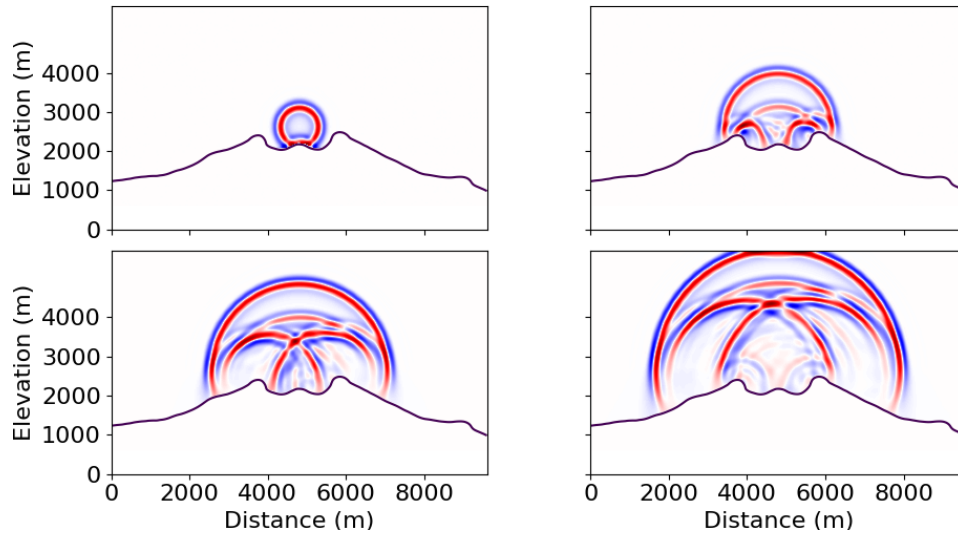


Figure 3.11: Snapshots of the wavefield at 2.5s, 5s, 7.5s, and 10s. Celerity is 350m/s throughout the model. Parameters are rescaled to better reflect infrasound propagation problems to which this boundary condition is applicable. The Courant number is consistent with previous examples. The wavefield shown in each snapshot is normalised by the maximum absolute value in that snapshot.

reversed polarity of these wavefronts versus those found in the previous example. Considerable reverberation within crater is observed, alongside the wavefront diffracting over obstacles (most notably the crater rim).

3.4.3. 3D 2nd-order acoustic free-surface

Whilst in some cases, wave propagation within a 3D physical domain can be adequately approximated along some suitable transect, this generally relies on some kind of consistency within the omitted dimension. Such approximations are possibly less suitable in the setting of a volcanic edifice, containing strong topographic variation along all directions: particularly true for Mount St Helens due to the collapsed northern flank. If realistic wavefield propagation is required in such settings, then full 3D modelling may be necessary.

The setup for this example was much the same as the 2D free-surface example. Figure 3.12 shows the results of wavefield interaction with the volcanic topography. The first arrivals radiate outwards with minimal interruption, diffracting around obstacles in their path, trailed by a series of

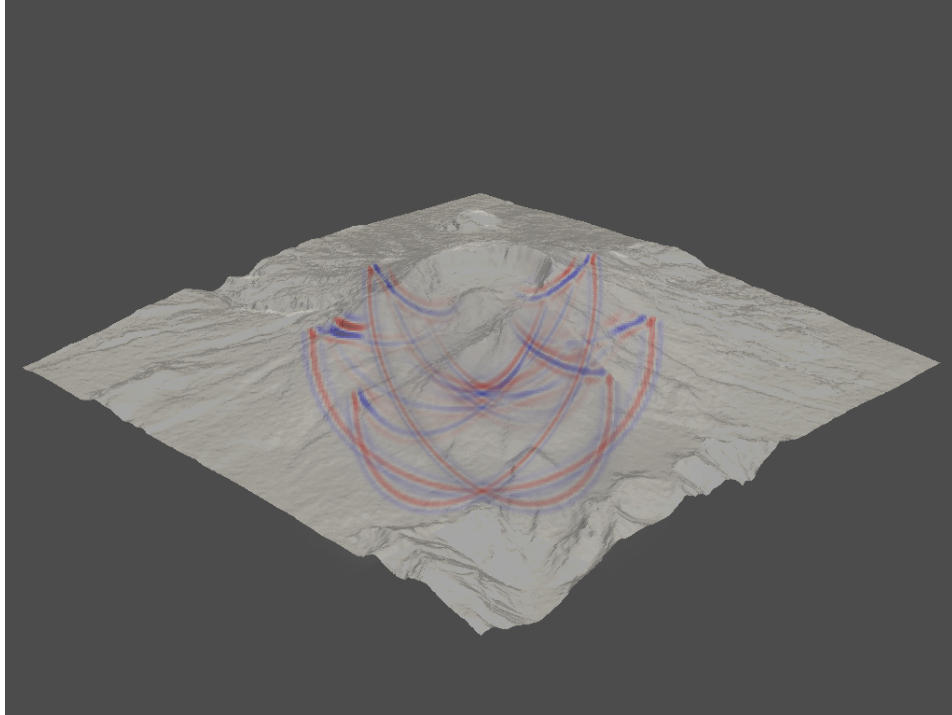


Figure 3.12.: Render of the 3D wavefield at 1125ms and topography. Slices of the wavefield are shown aligned and diagonal to each compass direction for clarity. Wavefield transparency is scaled with amplitude to emphasise the wavefronts.

wavefronts reflected from larger, flatter surfaces forming a layered series of distinct arrivals. These are followed by minor reverberations, decreasing in amplitude. Only the first arrivals remaining relatively unscathed, illustrating the shortcomings of a flat surface in adequately reproducing wavefield behaviour observed in rough terrain. This is in broad agreement with the consensus that inaccuracies caused by poor topography handling result in unrealistic propagation and degraded seismic imaging.

Wavefield slices shown in Figure 3.13 are at first glance similar to those of their 2D approximation in Figure 3.10: the position, shape, and relative amplitude of major wavefronts are the same. On closer inspection however, greater complexity emerges: the main wavefronts contain additional reflections and smaller-scale differences are visible, particularly around the crater rim where some out-of-plane reverberations are visible.

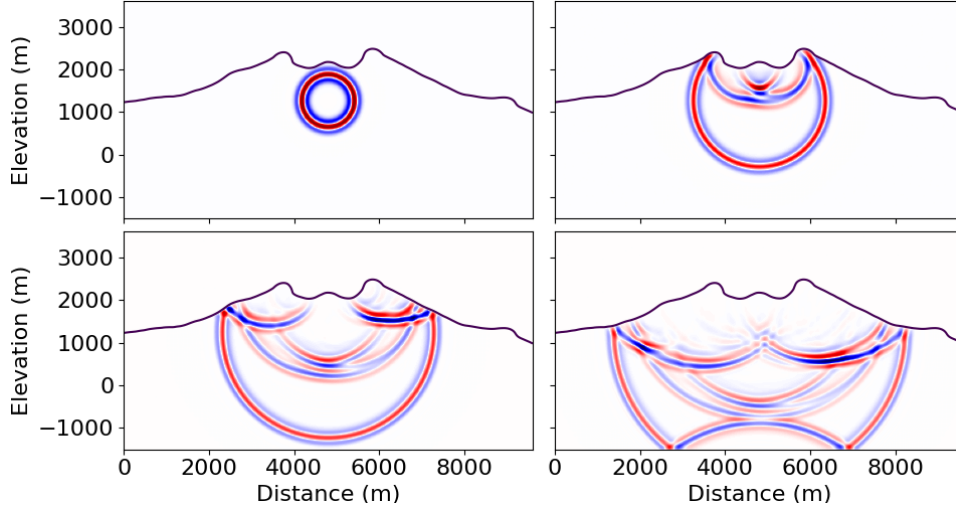


Figure 3.13: Slices through the wavefield on the profile used for the 2D examples. Snapshots were taken at 375ms, 750ms, 1125ms, and 1500ms respectively. Celerity is 2.5km/s throughout the model. The black line designates the isosurface $s(x, y, z) = 0$ on the SDF, coinciding with the surface. The wavefield shown in each snapshot is normalised by the maximum absolute value in that snapshot.

3.4.4. 3D 2nd-order acoustic rigid-surface

The previous rigid-surface example is similarly extended to 3D, the results of which are shown in Figure 3.14. In this case, likely due to the concavity of the surface in the vicinity of the source, out-of-plane reflections are considerably more pronounced, particularly within the confines of the caldera where complex and protracted reverberations can be clearly seen. Again, the most prominent reflected wavefronts become more confused in 3D when compared to the 2D wavefield on the same plane, exhibiting less coherency and continuity due to the wide range of paths taken by its constituent reflections.

These effects are particularly apparent when comparing Figure 3.15 to Figure 3.11, demonstrating the strongly 3D nature of wavefield propagation in this scenario.

3.5. Conclusions

The mathematical method developed was demonstrated for imposing both free and rigid surface boundary conditions with the second-order formu-

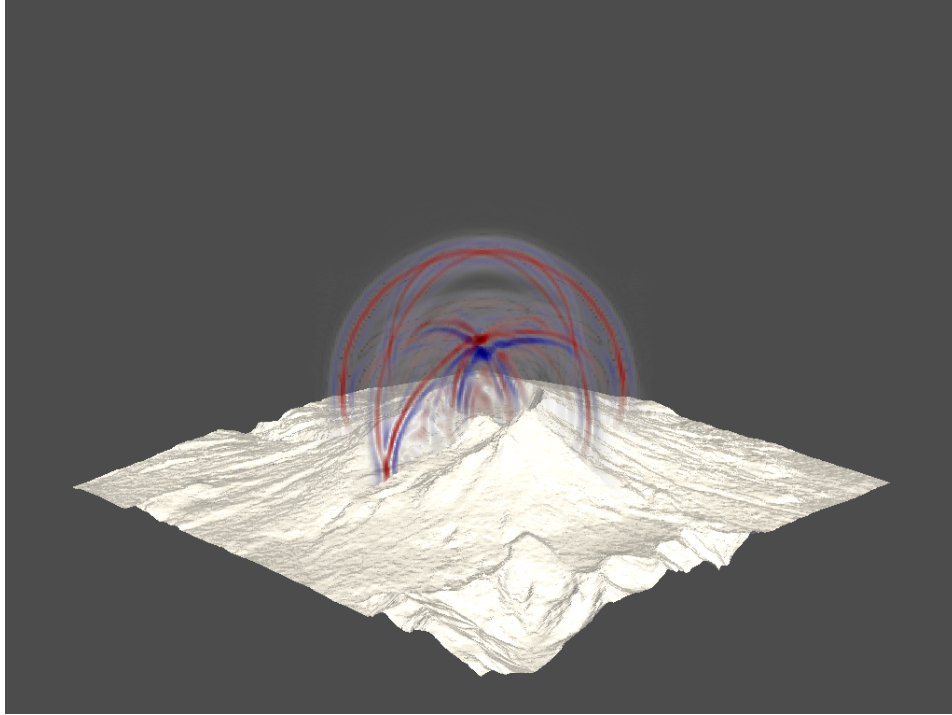


Figure 3.14.: Render of the 3D wavefield at 7.5s and topography. Slices of the wavefield are shown aligned and diagonal to each compass direction for clarity. Wavefield transparency is scaled with amplitude as before.

lation of the isotropic acoustic wave equation. The disparity of these constraints indicates that the method is suitably flexible to be applied consistently to both of these applications, this being encouraging for extension to more complex physics and boundary conditions. The other aspect highlighted is that whilst the governing equation informs the boundary conditions imposed, it does not directly affect the immersed boundary treatment itself beyond the stencils to which it should be applied. The implementation proposed generalises straightforwardly between dimensions and is suited to complex real-world geometries, implying suitability for practical applications.

Convergence testing demonstrates full-order convergence, with reduced error compared to the use of 1D, axially-aligned extrapolations. Furthermore, the spurious scattering introduced by staircased boundary representations is suppressed, enabling resolution to be determined by dispersion characteristics of the numerical scheme. Test cases show the effects of topographic scattering from real-world terrain, yielding results in line

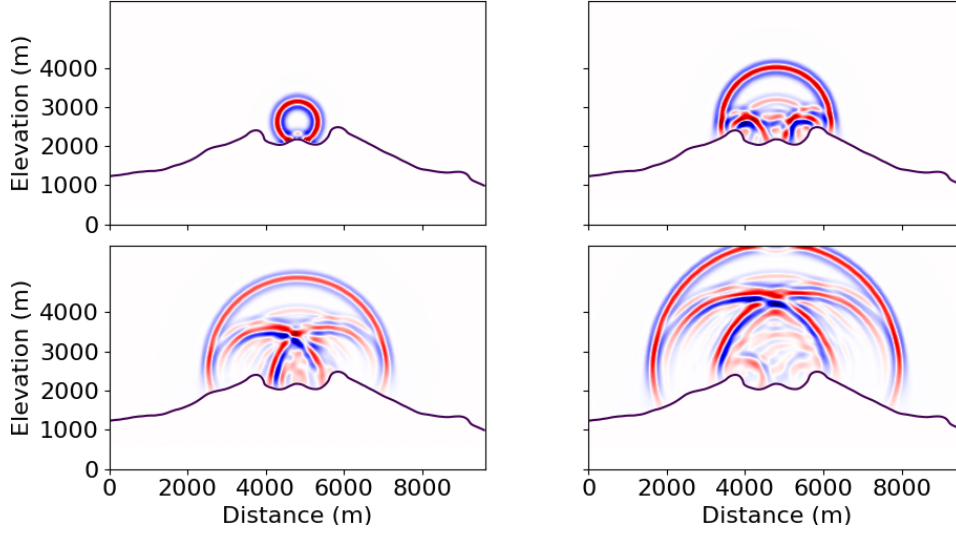


Figure 3.15: Snapshots of the wavefield at 2.5s, 5s, 7.5s, and 10s. Celerity is 350m/s throughout the model. The wavefield shown in each snapshot is normalised by the maximum absolute value in that snapshot.

with published findings.

3.6. Future Work

Future research directions include considering how the footprint of the extrapolant can be reduced whilst retaining formal order. One avenue of exploration would be the more intelligent selection of points with which to construct the support region. For example, it may be possible to distribute interior points in an optimal manner, perhaps dependent upon boundary geometry and the conditions imposed, such that fewer such points are required to form the stencil. Alternatively, it may be possible to distribute boundary points more optimally, reducing the number of interior points required. Suitable heuristics to determine the quality of the support region and algorithms for its methodical improvement would need to be devised.

In a similar vein, exploration of basis functions beyond Taylor series extrapolations may prove fruitful, achieving adequate accuracy whilst potentially reducing the number of coefficients in the expansion. This could again result in smaller support regions being required, translating to smaller linear systems and reduced computational cost for fitting the

extrapolant. This would potentially enable higher-order discretizations. It is also worth exploring the use of more sophisticated criteria with which to constrain the basis, potentially in the vein of the dispersion-relation-preserving schemes commonly used in seismic and aeroacoustics contexts [Tam and Webb, 1993].

Finally, further convergence testing for a wider range of spatial discretization orders and the rigid surface would provide a better understanding of the accuracy of the approach. Furthermore, comparison to vacuum-layer and coordinate transform implementations for various topographic profiles would aid in demonstrating the relative advantages of an immersed boundary approach.

Chapter 4

Immersed Boundaries and the 1st-Order Acoustic Wave Equation

The particle velocity field in the first-order acoustic wave equation introduces the requirement to naturally handle vector boundary conditions. Imposition of the particle velocity component of the free surface condition is considered, drawing on the method outlined in Chapter 2 to devise a suitable boundary treatment, exhibited in a series of test cases featuring analytical and real-world geometries.

Faced with multiple components of some vector- or tensor-valued function discretised at different nodes, it is necessary to impose boundary conditions at consistent positions on the surface to prevent inaccuracies or instabilities. Furthermore, the representation of the boundary condition must be consistent with the true boundary geometry: an aspect considered in previous work concerning the first-order acoustic wave equation. Mulder and Huiskes [2017] reused the 1D, axially-aligned extrapolations used in Mulder [2017], conceding that some additional numerical error is introduced, as this assumes axially-perpendicular boundaries. However, the tradeoff here is the relative simplicity of the scheme, this being fairly similar to the preceding Mulder [2017]. Hu [2016] considers the curvature of the boundary via a local cylindrical coordinate system, noting that assuming a locally planar surface can introduce significant error where topography is irregular.

In contrast, the approach proposed encapsulates boundary geometry information by applying constraints to the multidimensional bases at a suitable set of points, thereby creating some implicit surface where these constraints approximately hold. The distribution of these points and the boundary conditions imposed upon them encapsulate local boundary geometry naturally, avoiding incurring numerical error at this stage. Doing so removes the need for domain-specific approximations of boundary conditions, such as the 1D approximations considered by Mulder and Huiskes [2017], Hu [2016]. The 1D nature of these approximations also requires suitable 1D forms of the boundary conditions to be devised, necessitating domain knowledge. Whilst such approaches are demonstrably effective, the requirement of domain knowledge in designing the numerical scheme hinders generalisation to more complex physics. Ideally, this domain information should only influence the boundary conditions imposed, and not the methodology used to do so, maintaining the abstraction between application and the underlying numerical method.

Beyond previously-mentioned works, papers concerning application of immersed boundaries to this particular set of equations are scarce, an exception being Bilbao [2022]. There are no references therein considering this particular set of equations. The immersed boundary detailed in Bilbao [2022] is diffuse (in contrast to the sharp immersed boundary implementations considered in this work), this being a considerably different implementation to that considered in this work and produces leakage through the boundary. As to why there are so few publications considering application of immersed boundary treatments to this particular set of equations remains unclear.

4.1. Equations and Discretization

Given Equations 1.23 and 1.22 it is apparent that the evolution of p is dependent upon $\mathbf{v}$ and vice versa, addressed by stepping each equation forward in turn. To achieve second-order temporal accuracy, the velocity fields are offset by half a temporal increment in time such that updated particle velocities are calculated at time $t + \Delta t/2$. Updating fields in this manner is commonly referred to as a leapfrog scheme, the fields jumping

past one another in turn. Discretizing the temporal derivatives yields

$$\frac{p^{t+1} - p^t}{\Delta t} = \rho c^2 \nabla \cdot \mathbf{v}^{t+1/2} \quad (4.1)$$

and

$$\frac{\mathbf{v}^{t+1/2} - \mathbf{v}^{t-1/2}}{\Delta t} = \frac{1}{\rho} \nabla p^t, \quad (4.2)$$

where the indices indicate the timestep relative to the current position. Rearranging these to obtain the update equations gives

$$p^{t+1} = p^t + \Delta t \rho c^2 \nabla \cdot \mathbf{v}^{t+1/2} \quad (4.3)$$

and

$$\mathbf{v}^{t+1/2} = \mathbf{v}^{t-1/2} + \frac{\Delta t}{\rho} \nabla p^t \quad (4.4)$$

are obtained. Calculating $\mathbf{v}^{t+1/2}$ using Equation 4.4 enables the subsequent calculation of p^{t+1} via Equation 4.3, thereby stepping the fields forward to the next timestep. Given that a second-order central difference is calculated using points one forward and one behind, such an update would omit information from the current timestep. To avoid this, a two-point forward difference can be used. However, a two-point forward difference is only first-order accurate when evaluating a derivative at either of the constituent points; the leapfrog scheme evaluates the derivative between these points, achieving second-order temporal accuracy.

Systems of equations containing exclusively first-order derivatives are commonly solved by distributing the various fields across staggered subgrids. Centred derivatives cannot be used to calculate the spatial first derivatives without the emergence of even-odd decoupling modes, colloquially known as checkerboard error [Zhang et al., 2012]. Whilst filtering or dissipation can rectify this [Zhang et al., 2012], this behaviour is inherent to centred first derivatives [Corrêa et al., 2002]. As such various staggering schemes have been devised to discretise wave equations. In the case of the first-order acoustic wave equation, pressure is discretised onto the unstaggered reference grid, whilst velocity components are each offset by half a grid increment in their respective dimensions from the reference pressure node. Pressure derivative stencils are thus forward-staggered, whilst derivatives of particle velocity are backward-staggered.

4.2. The Free Surface

Free-surface conditions are required for both pressure and particle velocity fields. The pressure free-surface condition given by Equation 3.3 also applies in this case. Given Equation 1.23, it is apparent that the condition

$$\nabla \cdot \mathbf{v}(t, \mathbf{x}_b) = 0, \quad (4.5)$$

also holds. Taking the first derivative of Equation 3.4 with respect to time and substituting Equation 1.23 yields the third-order particle velocity condition, this being

$$\nabla^2 \nabla \cdot \mathbf{v}(t, \mathbf{x}_b) = 0, \quad (4.6)$$

a process which can be repeated for each subsequent pressure boundary condition.

4.2.1. Polynomial Approximation

In 2D, two derivatives require approximation, these being $\frac{\partial v_x}{\partial x}$ and $\frac{\partial v_y}{\partial y}$. Taking $\frac{\partial v_x}{\partial x}$ and approximating v_x with a 4th-order 2D Taylor series yields

$$\begin{aligned} \frac{\partial v_x}{\partial x}(t, x, y) \approx & \frac{\partial v_x}{\partial x}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^2 v_x}{\partial x^2}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^2 v_x}{\partial x \partial y}(t, x_0, y_0) \\ & + \frac{(x_b - x_0)^2}{2} \frac{\partial^3 v_x}{\partial x^3}(t, x_0, y_0) + (x_b - x_0)(y_b - y_0) \frac{\partial^3 v_x}{\partial x^2 \partial y}(t, x_0, y_0) \\ & + \frac{(y_b - y_0)^2}{2} \frac{\partial^3 v_x}{\partial x \partial y^2}(t, x_0, y_0) + \frac{(x_b - x_0)^3}{6} \frac{\partial^4 v_x}{\partial x^4}(t, x_0, y_0) \\ & + \frac{(x_b - x_0)^2(y_b - y_0)}{2} \frac{\partial^4 v_x}{\partial x^3 \partial y}(t, x_0, y_0) \\ & + \frac{(x_b - x_0)(y_b - y_0)^2}{2} \frac{\partial^4 v_x}{\partial x^2 \partial y^2}(t, x_0, y_0) \\ & + \frac{(y_b - y_0)^3}{6} \frac{\partial^4 v_x}{\partial x \partial y^3}(t, x_0, y_0), \end{aligned} \quad (4.7)$$

whilst doing the same for v_y in $\frac{\partial v_y}{\partial y}$ yields

$$\begin{aligned}
\frac{\partial v_y}{\partial y}(t, x, y) \approx & \frac{\partial v_y}{\partial y}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^2 v_y}{\partial x \partial y}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^2 v_y}{\partial y^2}(t, x_0, y_0) + \frac{(x_b - x_0)^2}{2} \frac{\partial^3 v_y}{\partial x^2 \partial y}(t, x_0, y_0) \\
& + (x_b - x_0)(y_b - y_0) \frac{\partial^3 v_y}{\partial x \partial y^2}(t, x_0, y_0) \\
& + \frac{(y_b - y_0)^2}{2} \frac{\partial^3 v_y}{\partial y^3}(t, x_0, y_0) + \frac{(x_b - x_0)^3}{6} \frac{\partial^4 v_y}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)^2(y_b - y_0)}{2} \frac{\partial^4 v_y}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + \frac{(x_b - x_0)(y_b - y_0)^2}{2} \frac{\partial^4 v_y}{\partial x \partial y^3}(t, x_0, y_0) \\
& + \frac{(y_b - y_0)^3}{6} \frac{\partial^4 v_y}{\partial y^4}(t, x_0, y_0).
\end{aligned} \tag{4.8}$$

For the third-order condition, the Laplacian of 4.7 is

$$\begin{aligned}
\nabla^2 \frac{\partial v_x}{\partial x}(t, x, y) \approx & \frac{\partial^3 v_x}{\partial x^3}(t, x_0, y_0) + \frac{\partial^3 v_x}{\partial x \partial y^2}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^4 v_x}{\partial x^4}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^4 v_x}{\partial x^3 \partial y}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^4 v_x}{\partial x^2 \partial y^2}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^4 v_x}{\partial x \partial y^3}(t, x_0, y_0),
\end{aligned} \tag{4.9}$$

whilst the Laplacian of 4.8 is

$$\begin{aligned}
\nabla^2 \frac{\partial v_y}{\partial y}(t, x, y) \approx & \frac{\partial^3 v_y}{\partial x^2 \partial y}(t, x_0, y_0) + \frac{\partial^3 v_y}{\partial y^3}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^4 v_y}{\partial x^3 \partial y}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^4 v_y}{\partial x^2 \partial y^2}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^4 v_y}{\partial x \partial y^3}(t, x_0, y_0) \\
& + (y_b - y_0) \frac{\partial^4 v_y}{\partial y^4}(t, x_0, y_0).
\end{aligned} \tag{4.10}$$

These two boundary conditions yield two rows for the linear system for each boundary point used to fit the extrapolant.

4.2.2. Boundary Points

The particle velocity boundary conditions directly result from the zero pressure condition and consistency between these two sets of conditions must be ensured, particularly given that conflicting boundary forcing may introduce instability. All components of this condition should be imposed at a common point. To this end, it is proposed that boundary points should be selected as discussed in Section 2.3.5, that being relative to the reference subgrid (the pressure grid in this case) regardless of the location of the fields upon which the boundary condition is imposed. It follows that a consistent expansion point should be used for each of the bases regardless of the location of the functions they approximate: all polynomials are expanded around the reference grid point with index corresponding to the current stencil position. This enables boundary conditions to be located relative to the reference grid.

4.3. Stencil Formation

4.3.1. Geometrical Handling of Subgrids

To simplify geometry handling and keep indexing consistent between fields, the footprint of the stencil for each field is shifted according to the offset of its particular subgrid.

With the introduction of these subgrids, it is necessary to keep track of the relative position of the boundary for consistent exterior-interior sectioning. Whilst one could nominally utilise a single SDF on the reference grid and estimate boundary position relative to the offset points, this is potentially unreliable. An individual SDF is defined on each subgrid for robust interior-exterior sectioning. Note however: only the SDF discretised on the reference subgrid is used for boundary point location.

For purposes of selecting interior points, proximity to boundary points is considered, excluding interior points if a boundary point lies within a surrounding hyperrectangle with sides $2\eta\Delta x_n$. The cutoff for pressure η_p was set to 0.5, whilst the cutoff for particle velocity η_v was zero. Whilst setting $\eta_v = 0.5$ did not introduce instability, nonphysical rippling was observed trailing the reflected wavefront. It is suspected that this may result from insufficient constraint of the extrapolants at the boundary.

4.4. Application to Example Geometry

For illustration, construction of the linear system used to fit the particle velocity extrapolant is carried out for a simple curved geometry, similar to that detailed in Chapter 3. To limit the size of matrices involved, a second-order discretization will be considered. For this particular geometry, the support region required to obtain sufficient information to constrain the basis has radius Δx greater than in the case of the 2nd-order acoustic wave equation free-surface constraints.

The boundary condition imposed is given by Equation 4.5. Approximating with 2D Taylor series yields

$$\begin{aligned} \frac{\partial v_x}{\partial x}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^2 v_x}{\partial x^2}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^2 v_x}{\partial x \partial y}(t, x_0, y_0) \\ + \frac{\partial v_y}{\partial y}(t, x_0, y_0) + (x_b - x_0) \frac{\partial^2 v_y}{\partial x \partial y}(t, x_0, y_0) + (y_b - y_0) \frac{\partial^2 v_y}{\partial y^2}(t, x_0, y_0) = 0. \end{aligned} \quad (4.11)$$

From this, a single row of the linear system is obtained for each boundary point

The support region for each particle velocity component is as shown in Figure 4.1 alongside the corresponding $\mathbf{A}$. The structure of $\mathbf{A}^+$ is shown in Figure 4.2, this being a dense matrix.

Given $\mathbf{A}^+$ maps values of both velocity components and boundary forcing values onto derivatives of both basis functions, each of these derivatives will be approximated as a weighted sum of values of both components. Consequently, any extrapolations using these bases will also be in terms of both v_x and v_y and thus stencils completed using these extrapolations will span both these fields.

For this particular geometry, the $\frac{\partial v_y}{\partial y}$ stencil corresponding to the point marked with a hollow triangle is truncated by the boundary. Whilst this point is on the exterior, the stencil applied at this index evaluates onto the corresponding pressure node which is located on the interior. Conversely, stencils which evaluate derivatives at exterior points do not need calculating, and can simply be omitted as there is no need to update values at these points. Boundary-modified stencils approximating this derivative are shown in Figure 4.3.

The particle velocity free surface and accompanying higher-order con-

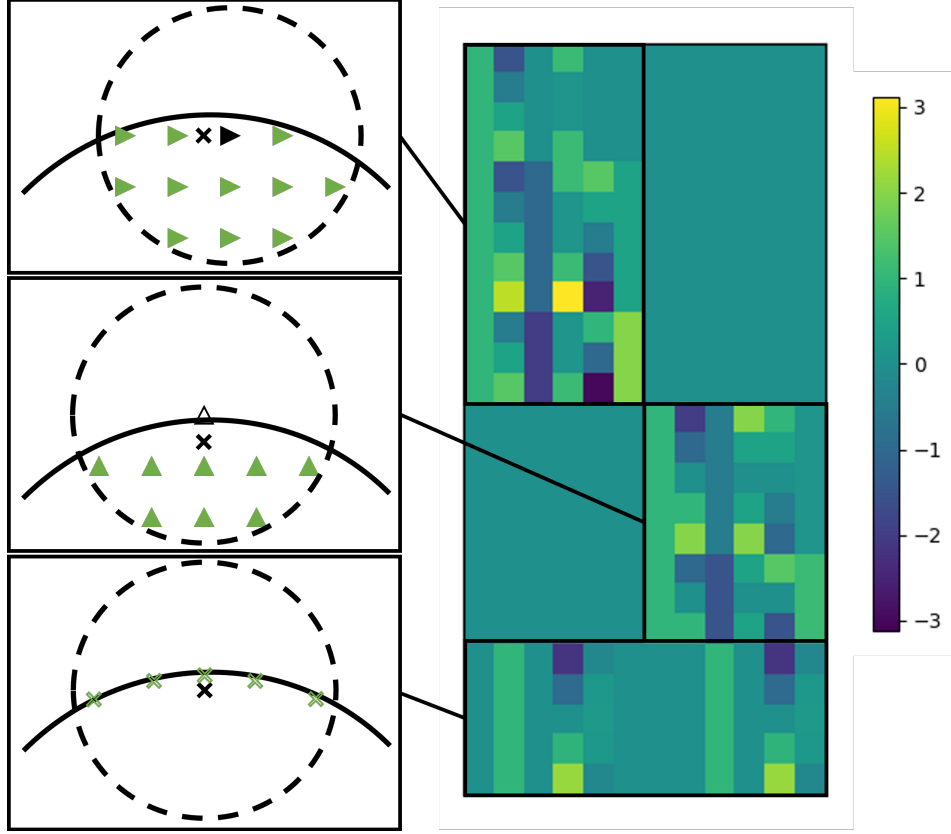


Figure 4.1: Assembling A for the particle velocity fields. Horizontal arrows correspond with v_x points used to construct the extrapolant, whilst vertical arrows are the respective v_y points. The hollow triangle is the centre of the v_y support region. Whilst the nominal position of the stencil is located on the exterior, its corresponding pressure node is located within the boundary. The black cross designates the expansion point. For clarity, only the subgrids of concern are shown. Note how the interior points of each field correspond with an independent block, whilst the boundary condition rows span both.

ditions concern multiple fields. As this results in $M/2$ conditions per boundary point to constrain N fields, the number of boundary conditions within a support region of some given radius is likely to be low compared to the pressure field (for which $1 + M/2$ conditions per boundary point are available to constrain a single field). As such, depending on geometry, more boundary and interior points may be required to adequately constrain the extrapolant than for the pressure stencils, requiring local use of a larger support region at these points.

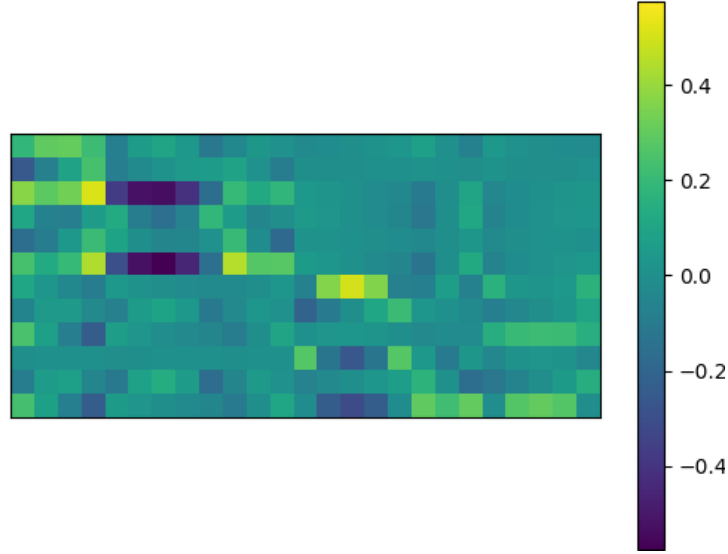


Figure 4.2.: The structure of the Moore-Penrose pseudoinverse of $\mathbf{A}$ shown in Figure 4.1.

4.4.1. Stencil Size and Implications

Larger stencils translate to higher operation count, increased memory pressure, and reduced memory locality in FD kernels, adversely impacting performance. Particular boundary conditions or combinations thereof may contain relatively little information regarding wavefield behaviour at the boundary, or alternatively result in $\mathbf{A}$ which are extremely mixed-determined, leaving some parameters overconstrained whilst others are largely lacking.

The zero-divergence boundary conditions are one such example. By inspection of Equations 4.7 and 4.8, it is apparent that some columns in the boundary condition submatrix will be consistently zero or one regardless of boundary position. As there is no information on the values of particular polynomial coefficients, additional interior points are required to constrain these coefficients.

This potential runaway growth in stencil extent is concerning, particularly in higher dimensions or where stencils contain multiple fields. Ideally a strategy to either identify such points *a priori* or limit resultant stencil size is desired. Whilst the former approach remains beyond the scope of this work, rudimentary means of addressing the latter were devised. Following the example of Mulder [2017], Mulder and Huiskes [2017], an

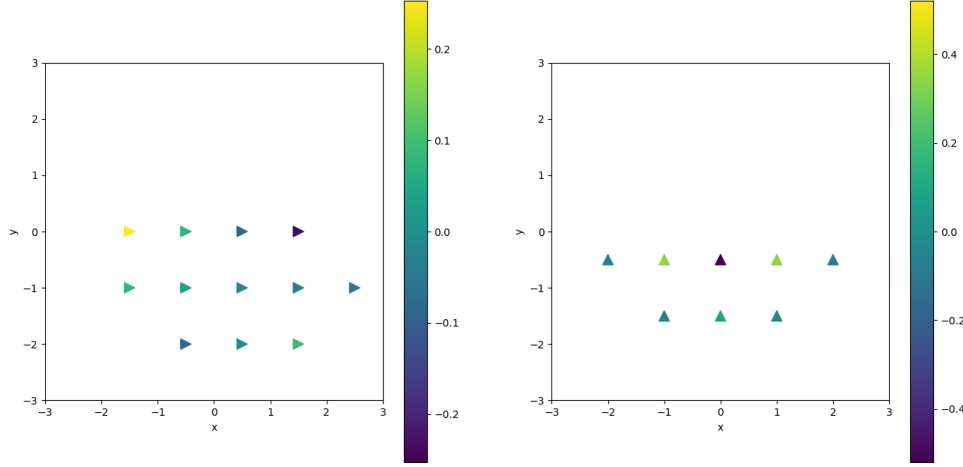


Figure 4.3.: The v_x and v_y portions of the boundary-modified derivative stencil approximating $\frac{\partial v_y}{\partial y}$. The stencil footprint in the v_x field is shown in the left subfigure, whilst the right shows the v_y footprint. Vestiges of the original stencil weights are somewhat visible in this modified stencil: the largest weight is imposed on what was the backward point of the original finite difference.

alternative strategy was devised whereby polynomial order of the extrapolant was reduced rather than expanding the support region. This approach achieves an over or equi-determined linear system by reducing the width of $\mathbf{A}$, in contrast to the expansion strategy which increases its height. This approach was found to be beneficial when implementing the particle velocity free-surface conditions in 3D with higher-order discretizations to limit maximum stencil extent. Note that this approach does not maintain formal order throughout the numerical scheme, although this may be acceptable in some applications.

It is postulated that there may be alternative combinations of support region and basis function which are less prone to this issue; further exploration in this regard may prove fruitful. However, this remains beyond the scope of this work and is left to future research.

4.5. Convergence Testing

To explore the convergence behaviour of the boundary treatment devised, the test considered in Section 3.2.4 was replicated with the 1st-order acoustic wave equation.

The particle velocity solutions are

$$v_x = -\frac{m}{\alpha} (1 - A \cos(x) \cosh(y)) \cos(mx_1 - \alpha t) \cos\left(\frac{\pi y_1}{2}\right) + \frac{\pi}{2\alpha} A \sin(x) \sinh(y) \sin(mx_1 - \alpha t) \sin\left(\frac{\pi y_1}{2}\right), \quad (4.12)$$

and

$$v_y = \frac{m}{\alpha} A \sin(x) \sinh(y) \cos(mx_1 - \alpha t) \cos\left(\frac{\pi y_1}{2}\right) + \frac{\pi}{2\alpha} (1 - A \cos(x) \cosh(y)) \sin(mx_1 - \alpha t) \sin\left(\frac{\pi y_1}{2}\right). \quad (4.13)$$

Density was set to 1 for this test and zero particle velocity flux was imposed at $y = 0$.

As before, a 4th-order spatial discretization was used and the timestep was set to 10% of the critical value. Results are shown in Figure 4.4.

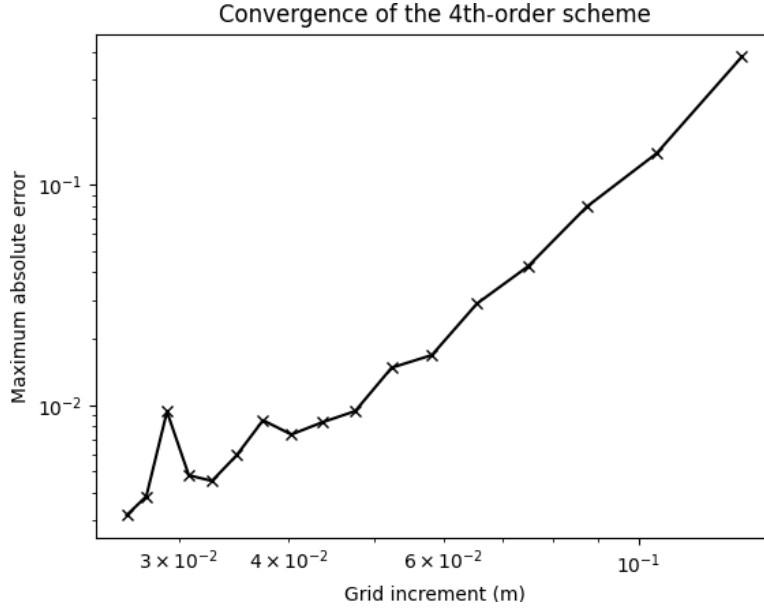


Figure 4.4.: Convergence of the numerical scheme for the 1st-order formulation of the acoustic wave equation using the same setup shown in Figure 3.9a.

As in the previous test, initial convergence is just short of 4th-order, gradually flattening around a grid increment of 0.01, at which point convergence is around 2nd-order as timestepping error saturates the solution. Note that convergence is somewhat less smooth beyond this point with a handful of blackspots where error is anomalously high compared to the

prevailing trend. Broadly speaking, maximum error in the scheme continues to fall as the discretization is refined. Investigation of these anomalies found them to be specific to particular grid sizes. Adding or removing a single node is sufficient to prevent the more prominent spike, although the cause of this behaviour remains to be determined. Overall, it appears that the error introduced by the boundary is rapidly eclipsed by other sources as the discretization is refined, implying that it is unlikely to be a significant source of error in practical applications.

4.6. Examples

The following examples increase in complexity from a simple analytical geometry to a complex 3D topography based on real terrain. The immersed boundary treatment developed is applied geometries representative of those found in practical applications, and the field-linking introduced by the boundary treatment is demonstrated in both 2D and 3D.

4.6.1. Application to Simple Geometry

As an initial demonstration, a simple 2D geometry, featured in Section 3.2.3, was used. As before, homogeneous material parameters were used throughout the computational domain to avoid conflating behaviour introduced by material contrasts with that stemming from the boundary treatment itself. The apparent simplicity of this geometry belies the resultant numerical complexity: boundary curvature and relative position are different for each gridpoint and the boundary cuts between pressure nodes and their respective particle velocity nodes in places. Placing the source on each side and reversing the interior-exterior sectioning (by taking the negative of the discretised SDFs), behaviour with both convex and concave surfaces is assessed.

The results of wave propagation with a source both above and below the surface are shown in Figures 4.5 and 4.6 respectively, injecting a Ricker wavelet into the pressure field in both. For the convex example, the surface was moved downwards to increase the size of the computational domain. In both cases, the reflection is smooth with the expected geometry.

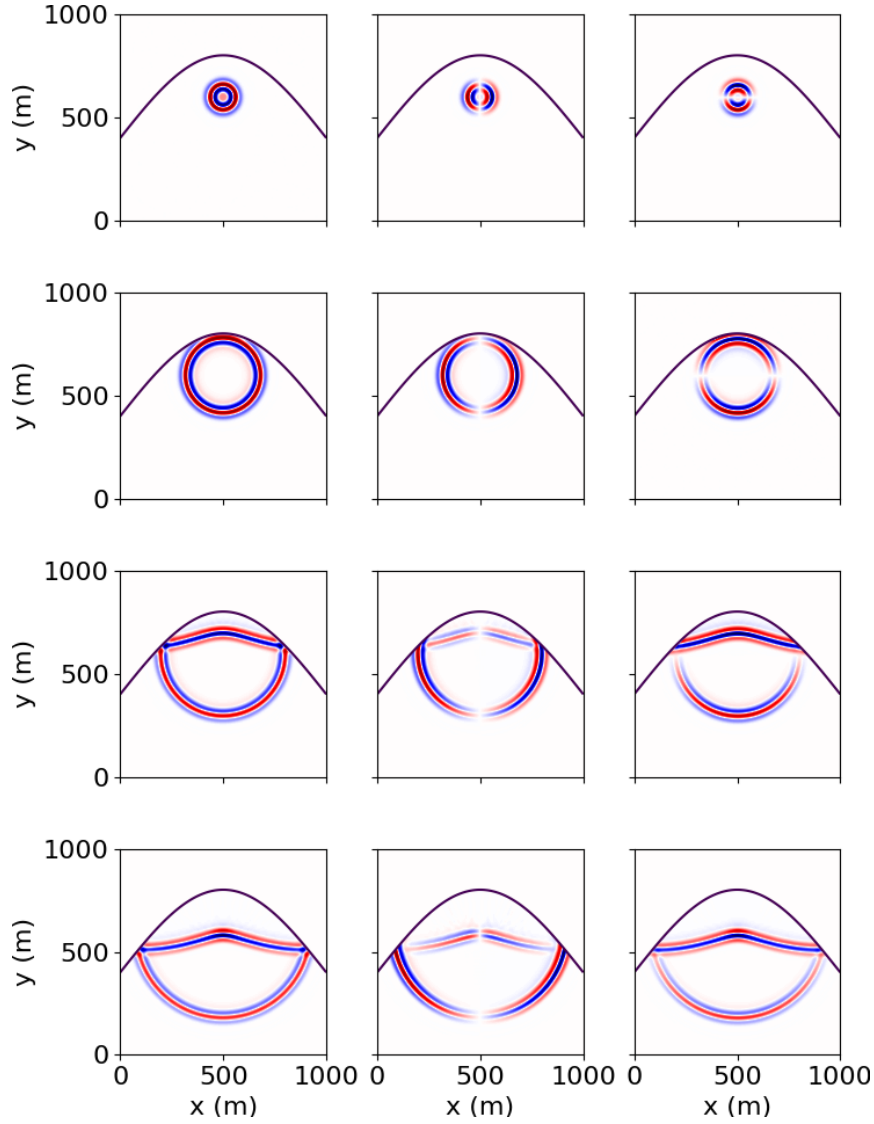


Figure 4.5.: Snapshots of the pressure (left), x particle velocity (middle), and y particle velocity (right) wavefields at evenly-spaced intervals for a wave approaching a concave sinusoid surface.

4.6.2. 2D 1st-order acoustic free-surface

The Mt St Helens geometry from the prior 2nd-order acoustic free-surface examples was revisited. As in the case of the 2nd-order acoustic examples, whilst real-world material parameters are spatially-variant, this is independent of the immersed boundary treatment. The initial 2D example is shown in Figure 4.7.

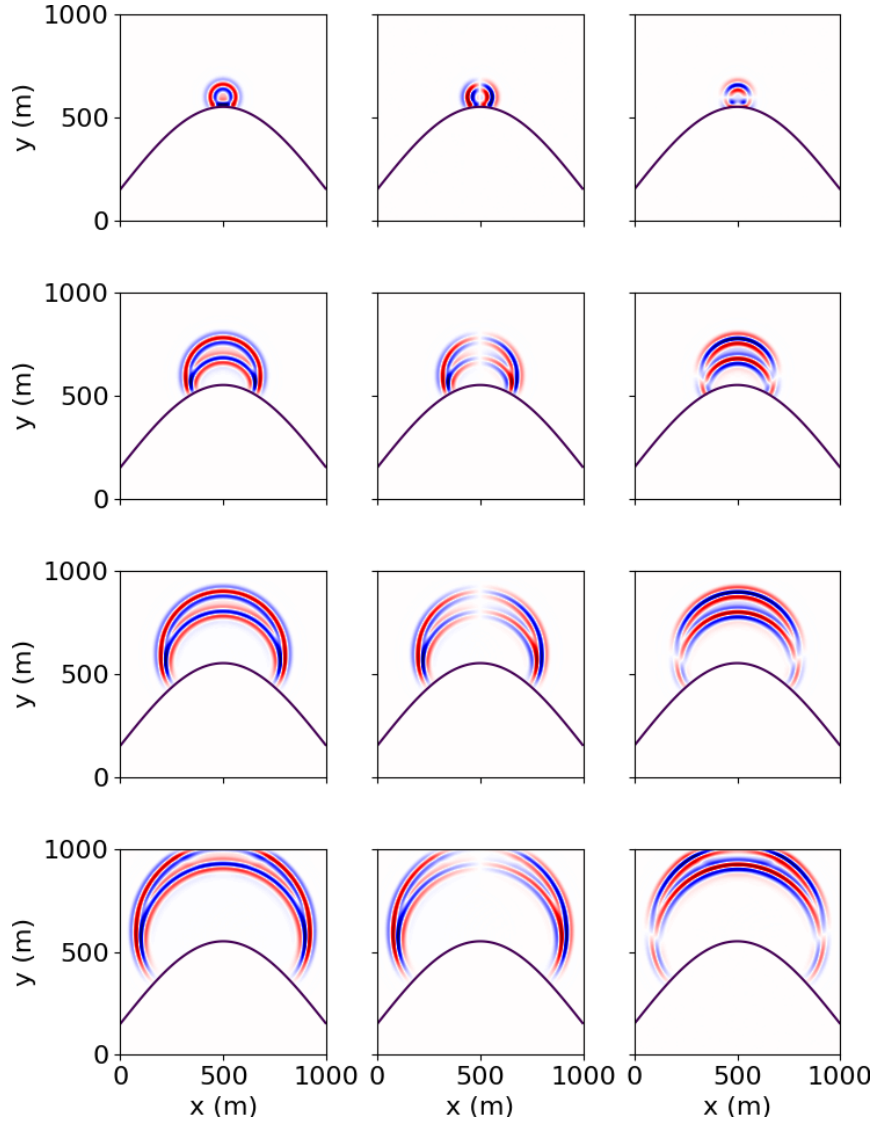


Figure 4.6.: Snapshots of the pressure (left), x particle velocity (middle), and y particle velocity (right) wavefields at evenly-spaced intervals for a wave approaching a convex sinusoid surface.

The reflected wavefront is considerably distorted by its interaction with the irregular-free surface, but remains smooth and free of staircasing error. The radiation pattern of first arrivals is only minimally impacted by the presence of topography, in agreement with previous studies [Takemura et al., 2015]. The reflection geometry is extremely similar to the example detailed in Chapter 3, with the additional particle velocity fields highlighting the partitioning of energy between horizontal and vertical particle

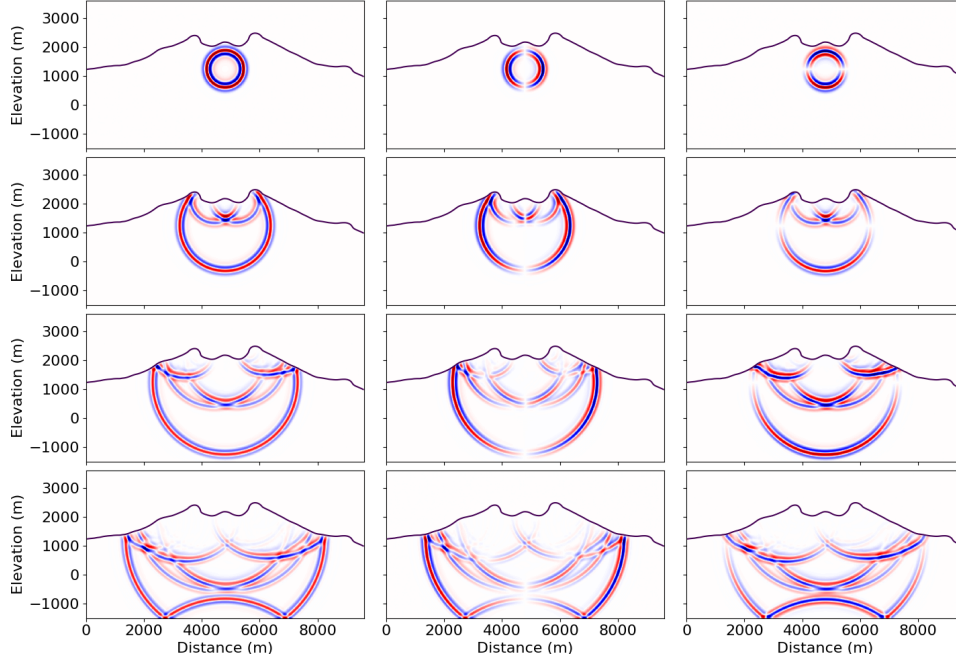


Figure 4.7.: Snapshots of the pressure(left), x particle velocity (middle), and y particle velocity (right) wavefields at 375ms, 750ms, 1125ms, and 1500ms. Celerity is 2.5km/s throughout the model, density is homogeneous throughout. The black line designates the isosurface $s(t, x, y) = 0$ on the SDF, coinciding with the surface. Note that all fields are normalised.

motion, and its relation to reflector orientation. The accuracy of the vector boundary condition implementation is clear in the clean, artefact-free reflections observed.

4.6.3. 3D 1st-order acoustic free-surface

The 2D model was extended to 3D to demonstrate the capabilities of this approach, again reusing the prior parameterisations. From Figure 4.8, it is apparent that the pattern of radiation is identical that for the 2nd-order formulation (note that the wavelet shape changes between the two, as the source signature was kept the same). Wavefield snapshots are shown in Figure 4.9. The y particle velocity field highlights the strongly 3D nature of interaction with topography, with energy reflected perpendicular to the plane of the transect strongly apparent. As with the 2nd-order formulation, reflections become considerably more complex and confused in the presence of 3D topography.

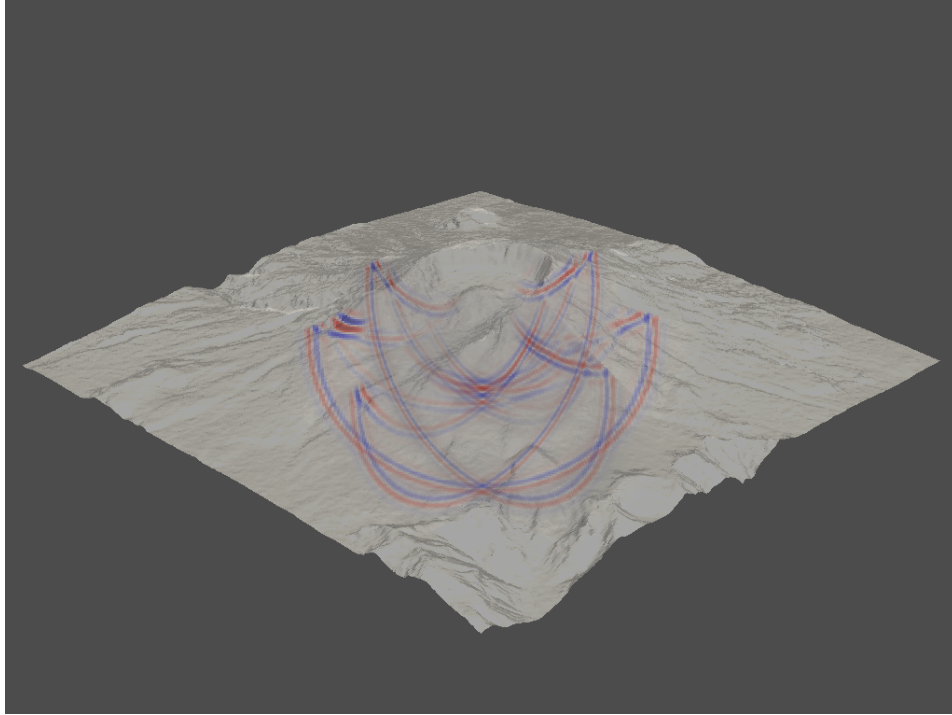


Figure 4.8.: Render of the 3D pressure wavefield at 1125ms and topography. Slices of the wavefield are shown aligned and diagonal to each compass direction for clarity. Wavefield transparency is scaled with amplitude to emphasise the wavefronts.

It was found that undesirably large particle velocity stencils were required at a small handful of points, highlighting limitations of the Taylor series as a basis. A basis-order-reduction strategy was thus adopted for this particular test case to rein in stencil size where insufficient information was present to construct the extrapolant.

4.7. Conclusions

Immersed boundary treatments based on the method developed in Chapter 2 have been demonstrated for implementing free surfaces with the 1st-order formulation of the isotropic acoustic wave equation. The condition to be imposed on the particle velocities dictates the relationship between these fields rather than some property of an individual field. By incorporating this relationship when fitting the basis functions used for extrapolation, this condition is handled naturally without the need to devise

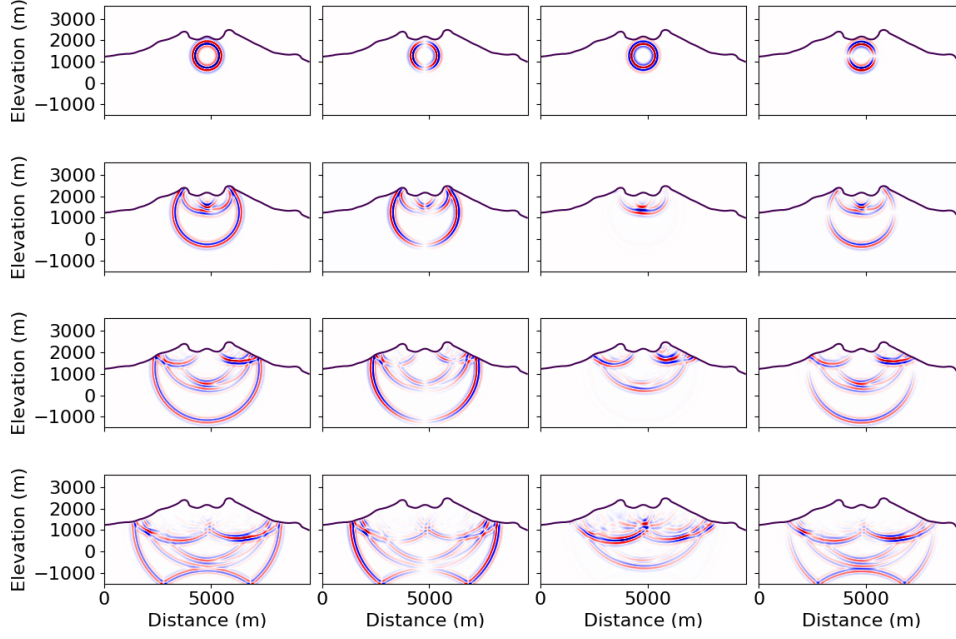


Figure 4.9.: Snapshots of the pressure (left), x particle velocity (middle left), y particle velocity (middle right), and z particle velocity (right) wavefields at 375ms, 750ms, 1125ms, and 1500ms. The y axis is oriented into the page. Celerity is 2.5km/s throughout the model, density is homogeneous throughout. The black line designates the isosurface $s(x, y, z) = 0$ on the SDF, coinciding with the surface.

suitable approximations at the surface. As the extrapolant requires information from all particle velocity components to satisfy this condition, the modified stencils applied in the vicinity of the boundary similarly span all of these fields. This yields an accurate and stable numerical scheme: an encouraging indicator for the suitability of the method to more complex boundary conditions where similar relationships are to be imposed.

The staggered discretization required to solve this 1st-order system introduces additional geometric complexities, with pressure and particle velocity nodes within a single FD cell being on opposite sides of the boundary, and the need to consistently impose boundary conditions on fields discretised onto different subgrids. Despite this, application to real-world topography yields realistic scattering patterns consistent with those detailed in Chapter 3. However, large particle velocity stencils are required at some points due to the reduced number of constraints per boundary point and increased number of fields per constraint when compared to those for the pressure field.

4.8. Future Work

Proposed future work is largely aimed at rectifying this stencil size issue. More efficient assembly and solving of the linear systems would somewhat mitigate the computational cost of evaluating these large stencils. However, it may be preferable to supplement the boundary conditions imposed with additional constraints based on domain knowledge. For example, particle velocity tangential to the free surface is zero in the acoustic case. These should nominally reduce size of the support region required to constrain the extrapolant, although this remains to be confirmed.

It is also worth revisiting ideas regarding alternate basis functions and more intelligent support region construction discussed in the conclusions of Chapter 3. More effective selection of boundary and interior constraints may enable the extrapolant to be fitted using fewer interior points. Suitable heuristics would be needed to determine the quality of the constraints imposed and their effect on properties of the linear system such as its condition number. What these heuristics may be and how they generalise across sets of constraints remains an open question however.

Chapter 5

Immersed Boundaries, VTI, and TTI

Seismic anisotropy is a commonly observed subsurface phenomenon: a product of micro- and macro-scale geological structure, producing non-spherical radiation patterns, distorting wavefronts, and time-shifting arrivals versus the isotropic case. Bedding planes formed by sequential deposition of sediment layers result in anisotropic elastic properties with seismic wavespeed dependent on propagation direction relative to the beds. Layering on scales much smaller than the wavelength of the incident wave results in a macro-scale homogeneous anisotropy [Backus, 1962, Thomsen, 1986]. Similar anisotropic effects can also arise from the preferred orientation of anisotropic mineral grains (mica in shale formations for example) during compaction and sedimentation [Thomsen, 1986]. This mode of anisotropy is known as transverse isotropy (TI), wherein isotropy is present in the bedding plane with a normal axis of symmetry.

Regional stresses, crustal shortening for example, or local forces such as salt tectonism bend, buckle, and rework geological units, tilting and folding horizontally-deposited beds into far more complex structures (fold and thrust belts for example). Thus comes the distinction between vertical transverse isotropy (VTI) and tilted transverse isotropy (TTI), the latter locally rotating the symmetry axis to follow the tilt of the strata. This tilt is spatially-variant, corresponding with folds and buckles introduced by tectonic stresses, potentially discontinuous across faults, intrusions, and unconformities.

Whilst typically weak, accounting for wavespeed variations of around 10-20% [Thomsen, 1986], anisotropy's effect on seismic wave propagation is considerable [Backus, 1962, Thomsen, 1986], particularly in seismic imaging contexts. In the case of FWI, data-matching required by the inversion procedure requires the kinematics of the dominant recorded phases to be accurately reproduced [Warner et al., 2013], especially for datasets containing both wide and narrow azimuth arrivals [Warner et al., 2013] (the former commonly considered requisite for successful FWI). Mismatched traveltimes between near-normal and wide-angle reflections due to differing vertical and horizontal velocities have long been recognised [Backus, 1962]; failure to account for these effects in FWI leads to degraded imaging [Plessix and Cao, 2011, Prioux et al., 2011]. Whilst isotropic propagators in anisotropic contexts can near-match true data, particularly at later FWI iterations [Plessix and Cao, 2011], the disproportionate effect of TI on short-offset reflections produces underestimated velocities or vertical stretching in the deep structure sampled by these raypaths [Prioux et al., 2011]. In RTM, omitting anisotropy where it is present results in mislocated reflectors and poor horizontal continuity in the resultant image [Bowling et al., 2009].

The combination of TTI, strong lateral velocity contrasts, and irregular topography presents considerable challenge to RTM [Shragge, 2016] and FWI [Wang et al., 2021] algorithms, with relatively few successful case studies. Topography handling approaches for anisotropic pseudoacoustic wave equations have predominantly focused on the use of curvilinear transformations [Shragge, 2016, Dong et al., 2017, Royle et al., 2021]. Two broad classes of transformations can be delineated: those based on purely vertical stretching (the flattening transform) and those with more sophisticated conformal or pseudo-orthogonal mappings. Flattening transforms, described independently in Tessmer et al. [1992], Hestholm et al. [1999], are relatively straightforward to apply and are used in both Shragge [2016] and Royle et al. [2021]. Whilst conformal mappings are theoretically more accurate, avoiding errors stemming from non-orthogonal grids, they are not always straightforwardly obtained [Rao and Wang, 2013]. As such, grid-optimisation routines to improve the orthogonality of the curvilinear grid are occasionally used [Dong et al., 2017]. However, it is necessary to transform the anisotropy axis accordingly such that it remains correctly

aligned relative to physical coordinate directions. This likely explains the replacement of the conformal mapping used by Shragge [2013] with a flattening transform when extending their method to TTI in Shragge [2016].

In contrast, an immersed boundary approach alleviates many of these problems: regular Cartesian grids simplify both equations and model parameterisation, and highly irregular geometries remain straightforward to implement. However, this approach has been almost entirely overlooked with the exception of Huiskes et al. [2017] extending Mulder [2017], Mulder and Huiskes [2017] to the VTI equations of Plessix and Cao [2011].

This chapter demonstrates the application of the generalised immersed boundary methodology detailed in Chapter 2 to multiple pseudoacoustic VTI and TTI formulations. A wide variety of formulations are used, the TTI formulation of Fletcher et al. [2009] being a common example. The VTI equations detailed Duveneck et al. [2008] and Fowler et al. [2010] were used to incrementally increase the complexity of physics handled by the immersed boundary and verify its suitability across a range of problems. Furthermore, the effect of isotropic versus anisotropic free-surface boundary conditions in combination with approximations made in the derivation of these equations is explored: a comparison enabled by the generality of the framework and degree of automation.

5.1. Duveneck/Plessix VTI

Whilst a multitude of wave equations for modelling pure P-wave propagation with VTI and TTI have been proposed, parameterising the anisotropy via the parameters δ and ϵ , originally introduced by Thomsen [1986], is commonplace. The equations presented by Duveneck et al. [2008] and their derivation are given in 1.13 and 1.14. Note that Plessix and Cao [2011] derive a slightly different set of second-order equations from the same first-order system, although they are equivalent in the presence of constant ϵ , δ , and ρ .

5.1.1. Free-Surface Immersed Boundaries

The zero-stress boundary condition described in Chapter 3 should be imposed upon both τ_H and τ_V to produce the desired behaviour and ensure

stability at the free surface. Whilst higher-order conditions can be derived from the zero-stress boundary condition and governing system of PDEs, Huiskes et al. [2017] impose higher-order conditions taken from the isotropic case. The effect of this substitution on wavefield behaviour is unclear. However, these isotropic boundary conditions represent much more compact expressions than their anisotropic counterparts, potentially reducing computational cost and complexity whilst yielding more concise mathematics. It is thus worth determining whether this simplification has a detrimental effect on boundary behaviour, and whether this represents an acceptable compromise given the reduction in complexity achieved.

In 2D, Equations 1.13 and 1.14 reduce to

$$\begin{aligned}\frac{\partial^2 \tau_H}{\partial t^2} &= c_p^2 \left((1 + 2\epsilon) \frac{\partial^2 \tau_H}{\partial x^2} + \sqrt{1 + 2\delta} \frac{\partial^2 \tau_V}{\partial y^2} \right) \\ \frac{\partial^2 \tau_V}{\partial t^2} &= c_p^2 \left((1 + 2\epsilon) \frac{\partial^2 \tau_H}{\partial x^2} + \frac{\partial^2 \tau_V}{\partial y^2} \right).\end{aligned}\tag{5.1}$$

Given the governing equations and conditions of zero vertical and horizontal stress on the boundary, higher-order boundary conditions incorporating anisotropy can be derived. The zero-stress boundary conditions

$$\begin{aligned}\tau_H(t, \mathbf{x}_b) &= 0 \\ \tau_V(t, \mathbf{x}_b) &= 0,\end{aligned}\tag{5.2}$$

imply that

$$\begin{aligned}\frac{\partial^2 \tau_H}{\partial t^2}(t, \mathbf{x}_b) &= 0 \\ \frac{\partial^2 \tau_V}{\partial t^2}(t, \mathbf{x}_b) &= 0,\end{aligned}\tag{5.3}$$

must also hold. Equations 5.1 and 5.3 imply

$$\begin{aligned}(1 + 2\epsilon) \frac{\partial^2 \tau_H}{\partial x^2}(t, \mathbf{x}_b) + \sqrt{1 + 2\delta} \frac{\partial^2 \tau_V}{\partial y^2}(t, \mathbf{x}_b) &= 0 \\ \sqrt{1 + 2\delta} \frac{\partial^2 \tau_H}{\partial x^2}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_V}{\partial y^2}(t, \mathbf{x}_b) &= 0.\end{aligned}\tag{5.4}$$

Higher-order conditions can be defined by repeating this process. The fourth derivative of τ_H and τ_V with respect to time must be zero on the surface. Taking the second derivative of Equation 5.1 with respect to

time and setting the right-hand side equal to zero yields the fourth-order boundary conditions

$$\begin{aligned} (1 + 2\epsilon) \frac{\partial^2}{\partial x^2} \frac{\partial^2 \tau_H}{\partial t^2}(t, \mathbf{x}_b) + \sqrt{1 + 2\delta} \frac{\partial^2}{\partial y^2} \frac{\partial^2 \tau_V}{\partial t^2}(t, \mathbf{x}_b) &= 0 \\ \sqrt{1 + 2\delta} \frac{\partial^2}{\partial x^2} \frac{\partial^2 \tau_H}{\partial t^2}(t, \mathbf{x}_b) + \frac{\partial^2}{\partial y^2} \frac{\partial^2 \tau_V}{\partial t^2}(t, \mathbf{x}_b) &= 0. \end{aligned} \quad (5.5)$$

Substituting $\frac{\partial^2 \tau_H}{\partial t^2}$ and $\frac{\partial^2 \tau_V}{\partial t^2}$ from Equation 5.1 gives

$$\begin{aligned} (1 + 2\epsilon) \frac{\partial^2}{\partial x^2} \left(c_p^2 (1 + 2\epsilon) \frac{\partial^2 \tau_H}{\partial x^2}(t, \mathbf{x}_b) + c_p^2 \sqrt{1 + 2\delta} \frac{\partial^2 \tau_V}{\partial y^2}(t, \mathbf{x}_b) \right) \\ + \sqrt{1 + 2\delta} \frac{\partial^2}{\partial y^2} \left(c_p^2 \sqrt{1 + 2\delta} \frac{\partial^2 \tau_H}{\partial x^2}(t, \mathbf{x}_b) + c_p^2 \frac{\partial^2 \tau_V}{\partial y^2}(t, \mathbf{x}_b) \right) &= 0 \\ \sqrt{1 + 2\delta} \frac{\partial^2}{\partial x^2} \left(c_p^2 (1 + 2\epsilon) \frac{\partial^2 \tau_H}{\partial x^2}(t, \mathbf{x}_b) + c_p^2 \sqrt{1 + 2\delta} \frac{\partial^2 \tau_V}{\partial y^2}(t, \mathbf{x}_b) \right) \\ + \frac{\partial^2}{\partial y^2} \left(c_p^2 \sqrt{1 + 2\delta} \frac{\partial^2 \tau_H}{\partial x^2}(t, \mathbf{x}_b) + c_p^2 \frac{\partial^2 \tau_V}{\partial y^2}(t, \mathbf{x}_b) \right) &= 0. \end{aligned} \quad (5.6)$$

Assuming locally-constant c_p , ϵ , and δ allows the inner and outer spatial derivatives to be combined, yielding

$$\begin{aligned} (1 + 2\epsilon)^2 \frac{\partial^4 \tau_H}{\partial x^4}(t, \mathbf{x}_b) + (1 + 2\epsilon) \sqrt{1 + 2\delta} \frac{\partial^4 \tau_V}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) \\ + (1 + 2\delta) \frac{\partial^4 \tau_H}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + \sqrt{1 + 2\delta} \frac{\partial^4 \tau_V}{\partial y^4}(t, \mathbf{x}_b) &= 0 \\ (1 + 2\epsilon) \sqrt{1 + 2\delta} \frac{\partial^4 \tau_H}{\partial x^4}(t, \mathbf{x}_b) + (1 + 2\delta) \frac{\partial^4 \tau_V}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) \\ + \sqrt{1 + 2\delta} \frac{\partial^4 \tau_H}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + \frac{\partial^4 \tau_V}{\partial y^4}(t, \mathbf{x}_b) &= 0. \end{aligned} \quad (5.7)$$

The higher-order anisotropic conditions in Equations 5.4 and 5.7 differ considerably from their isotropic counterparts, although zeroth-order conditions are identical in both cases. Qualitatively similar wavefield behaviour is thus expected.

5.1.2. Comparing Isotropic and Anisotropic Conditions

To quantify the difference in behaviour induced by anisotropic versus isotropic higher-order boundary conditions, a model was parameterised with homogeneous material properties: $c_p = 1\text{km s}^{-1}$, $\delta = 0.1$, and $\epsilon = 0.25$ for anelliptic anisotropy. The model domain was a 1km square with $\Delta x = \Delta y = 10\text{m}$. Equation 5.1 was discretised with a fourth-order-accurate spatial discretisation and second-order explicit timestepping. A Ricker source was positioned 100m below the free surface to maximise interaction between the boundary treatment and generated shear wave artefacts. Free surface conditions were imposed on the sinusoid hill geometry first used in Section 3.2.3 whilst zero Dirichlet conditions were imposed on all other sides of the domain. The simulation was run for 600ms to minimise interaction between the wavefront and Dirichlet boundary conditions, with a timestep of 5ms .

For comparison, two separate models were created, one based on the isotropic free-surface conditions given in Equations 3.4 and 3.5, whilst in the other, anisotropic conditions given in Equations 5.4 and 5.7 were imposed. In both cases, the conditions in Equation 5.2 were imposed. Immersed boundary treatments were devised imposing each set of conditions according to the method described in Section 2.3.

Whilst Equations 3.4 and 3.5 concern only a single field, Equations 5.4 and 5.7 impose a relationship between τ_H and τ_V fields. Consequently, imposition of the anisotropic higher-order conditions requires values of both τ_H and τ_V to construct an appropriate extrapolant. Similarly to the particle-velocity boundary conditions discussed in Chapter 4, the boundary conditions given in Equations 5.4 and 5.7 mean that corresponding boundary-modified stencils must contain points from multiple fields. This increases both the size of the linear system which must be solved and the resultant modified stencils, making anisotropic free-surface conditions more computationally expensive to impose via the prescribed method than isotropic.

Figure 5.1 compares the effects of these two sets of boundary conditions, showing wavefields for each at 600ms alongside the difference between the two. Anisotropy flattens the circular wavefront into more of a diamond shape whilst a prominent shear wave artefact (a product of the governing

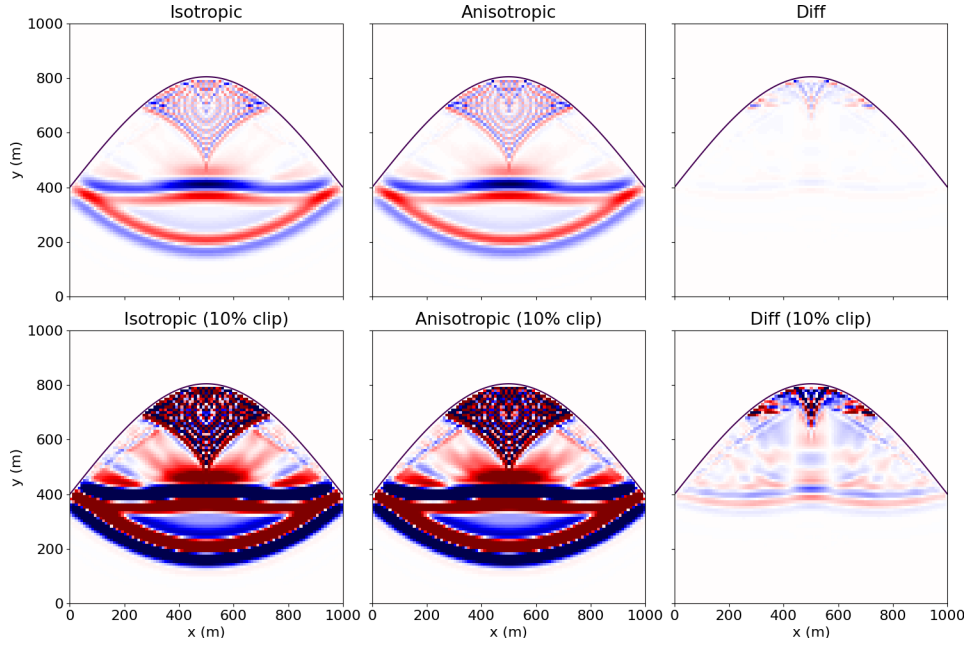


Figure 5.1.: Comparing effects of isotropic free-surface conditions (shown in the left column) to those specified in Equations 5.4 and 5.7 (shown in the middle column). The difference between the two is shown in the right column using the same colour scale. The upper row shows wavefield snapshots at 600ms, with all wavefields normalised against the maximum absolute value of the τ_H wavefield in the isotropic case. To emphasise small differences between the two wavefields, the lower row of subfigures show the same wavefields clipped to 10% of the maximum absolute value of the τ_H wavefield in the isotropic case.

equations) radiates from the source location. This artefact is subject to considerable numerical dispersion due to its short wavelength. Little difference is immediately apparent between the effects of the isotropic and anisotropic boundary conditions: not unexpected as the zeroth-order conditions are identical in both cases. Clipping the wavefield to 10% of the maximum absolute value of the isotropic field makes the differences more observable, highlighting some degree of phase error in the reflected wavefront introduced by the use of isotropic boundary conditions in the place of anisotropic. However, the most substantial difference is present where the shear wave artefact reflects from the boundary. Given that differences between the two are predominantly in nonphysical wavefield features, it is anticipated that any error introduced by isotropic higher-order boundary conditions will be small compared to errors introduced by approxima-

tions made in deriving of the governing equations. The physical portion of the wavefield exhibits a considerably smaller misfit. Note that for both sets of boundary conditions, a faint pseudoshear artefact appears where one would expect P-S conversion at an elastic free surface, this being more apparent in the clipped wavefields.

Given both sets of boundary conditions are derived from Equation 5.2, convergence of the two wavefields is expected as resolution increases. To investigate, the resolution of the grid was doubled, reducing the spatial increment from 10m to 5m whilst holding the Courant number constant. Comparison of the two sets of conditions was repeated, the result of which is shown in Figure 5.2.

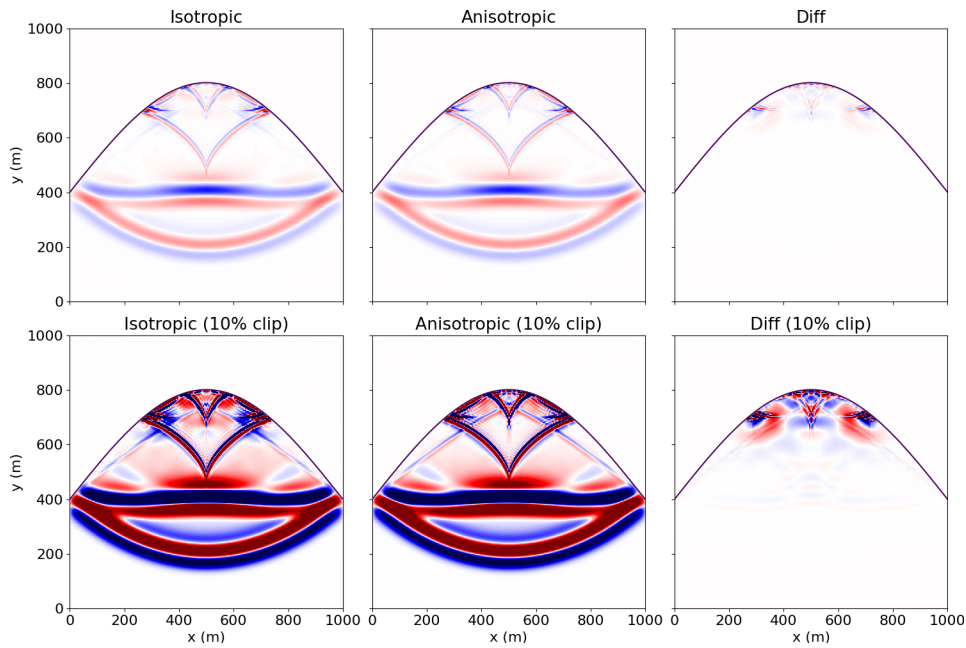


Figure 5.2.: Repeating the experiment shown in Figure 5.1 with twice the spatial and temporal resolution. Subfigures are arranged as in Figure 5.1, again using the maximum absolute value of the τ_H field in the case featuring the isotropic free surface for normalisation and clipping.

The phase error in the reflected wavefront observed between the isotropic and anisotropic wavefronts effectively vanishes at this refined resolution, although differences between the two wavefields persist where the shear wave artefact interacts with the surface. Given that the shear wave artefact is nonphysical, this behaviour is expected. When wavefield values are clipped, a faint long-wavelength artefact can be observed where the shear

wave artefact interacts with the boundary, this being more apparent in the case where isotropic boundary conditions are imposed. It is of note that this artefact has a considerably longer wavelength than the incident shear wave artefact from which it originates.

Use of the free-surface conditions derived from the isotropic wave equations in the place of those derived from the anisotropic governing equations when solving the equations of Duveneck et al. [2008] can have a notable effect on reflections from the free surface. However, given that the linear systems formed and resultant stencils applied are larger in the anisotropic case, and the behaviour of isotropic and anisotropic conditions converges with increased resolution, a trade-off between accuracy and computational cost is present. Accurate free-surface behaviour requires either fine resolution or imposition of anisotropic boundary conditions, although the optimal choice to minimise computational cost whilst retaining sufficient accuracy would require consideration on a case by case basis.

5.1.3. Real-Topography Example

To evaluate the performance of the boundary treatment in the presence of more irregular, realistic topography profiles, the Mt St Helens topography profile introduced in Chapter 3 was reused. Physical extent, temporal and spatial resolutions, and material parameters detailed in Section 3.4 were reused, with the addition of ϵ and δ equal to 0.25 and 0.1 respectively. A Ricker wavelet with a peak frequency of 8Hz was injected at an elevation of 1250m, centred in the x axis. Equation 5.1 was discretised with fourth-order accuracy in space and a second-order-accurate explicit timestepping scheme. Anisotropic free-surface conditions given by Equations 5.2, 5.4, and 5.7 were imposed on the surface.

Snapshots of the τ_H wavefield over the first 1500ms of propagation are shown in Figure 5.3. Comparing this to results featuring the same 2D topography profile in Chapter 3, the expected flattening of the wavefront is apparent, as horizontally-propagating waves travel faster. The pattern of reflections is similarly altered, although many of the reflected wavefronts can be discerned in both cases. Faint artefacts corresponding to the P-S conversion can be discerned in places. Applying the generalised

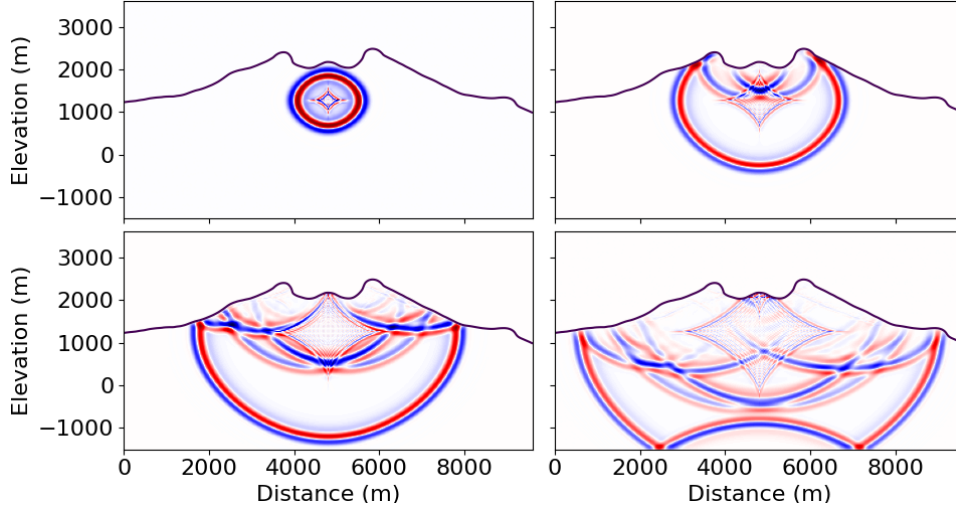


Figure 5.3.: Solving the VTI equations of Duveneck et al. [2008], Plessix and Cao [2011] in the presence of the 2D Mt St Helens topographic profile. Note that the axis scales are equal, and the shape of the radiating wavefield is a result of the anisotropy. Snapshots of the τ_H wavefield are shown at at 375ms, 750ms, 1125ms, and 1500ms. Values are normalised against the maximum absolute value in each snapshot.

immersed boundary method described in Chapter 2 to Equation 5.1 and the corresponding free-surface conditions enables the substantial and irregular topography in this particular profile to be handled without issue, implying that it is well-suited for this particular application.

5.2. Fletcher VTI

An alternative formulation for pseudoacoustic wave equations featuring transverse isotropy is that proposed by Fletcher et al. [2009], given in Equation 1.41. As with the works of Duveneck et al. [2008], Plessix and Cao [2011], a pair of coupled, second-order equations are used to approximate P-wave propagation in an anisotropic medium. Whilst Fletcher et al. [2009] addresses TTI, prior work is referenced therein on the subject of VTI (although this conference abstract is apparently unavailable) and VTI represents the trivial case where $\theta = 0$.

One advantage of the Fletcher et al. [2009] formulation is that it does not require $\epsilon \geq \delta$ to hold throughout the model domain to obtain a stable wavefield [Fowler et al., 2010]: an instability inherent in the physics rather

than resulting from the discretisation [Fowler et al., 2010]. Whilst $c_{sy} = 0$ simplifies the equations, reducing the number of derivatives present, it does not remove shear waves as one might intuitively expect, producing considerable shear wave artefacts around compressive sources [Grechka et al., 2004]. Note that with $c_{sy} = 0$, these equations become a scaled version of those in Duveneck et al. [2008].

5.2.1. Free-Surface Immersed Boundaries

In 2D, the operators introduced in Equation 1.38 reduce to

$$\begin{aligned} H_1 &= \frac{\partial^2}{\partial y^2} \\ H_2 &= \frac{\partial^2}{\partial x^2}. \end{aligned} \tag{5.8}$$

Given Equation 5.3, the right-hand side of Equation 1.41 must also equal zero on the boundary, yielding the second-order conditions:

$$\begin{aligned} c_{px}^2 \frac{\partial^2 p}{\partial x^2}(t, \mathbf{x}_b) + \alpha(c_{pz}^2 - c_s^2) \frac{\partial^2 q}{\partial y^2}(t, \mathbf{x}_b) + c_s^2 \frac{\partial^2 p}{\partial y^2}(t, \mathbf{x}_b) &= 0 \\ \frac{v_{pn}^2 - c_s^2}{\alpha} \frac{\partial^2 p}{\partial x^2}(t, \mathbf{x}_b) + c_{pz}^2 \frac{\partial^2 q}{\partial y^2}(t, \mathbf{x}_b) + c_s^2 \frac{\partial^2 q}{\partial x^2}(t, \mathbf{x}_b) &= 0. \end{aligned} \tag{5.9}$$

Taking the second derivative of Equation 5.9 with respect to time yields

$$\begin{aligned} c_{px}^2 \frac{\partial^2}{\partial x^2} \frac{\partial^2 p}{\partial t^2}(t, \mathbf{x}_b) + \alpha(c_{pz}^2 - c_s^2) \frac{\partial^2}{\partial y^2} \frac{\partial^2 q}{\partial t^2}(t, \mathbf{x}_b) + c_s^2 \frac{\partial^2}{\partial y^2} \frac{\partial^2 p}{\partial t^2}(t, \mathbf{x}_b) &= 0 \\ \frac{v_{pn}^2 - c_s^2}{\alpha} \frac{\partial^2}{\partial x^2} \frac{\partial^2 p}{\partial t^2}(t, \mathbf{x}_b) + c_{pz}^2 \frac{\partial^2}{\partial y^2} \frac{\partial^2 q}{\partial t^2}(t, \mathbf{x}_b) + c_s^2 \frac{\partial^2}{\partial x^2} \frac{\partial^2 q}{\partial t^2}(t, \mathbf{x}_b) &= 0, \end{aligned} \tag{5.10}$$

from which, assuming locally constant material parameters, one obtains

$$\begin{aligned}
& c_{px}^4 \frac{\partial^4 p}{\partial x^4}(t, \mathbf{x}_b) + \alpha c_{px}^2 (c_{pz}^2 - c_s^2) \frac{\partial^4 q}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + c_{px}^2 c_s^2 \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) \\
& + (c_{pz}^2 - c_s^2)(v_{pn}^2 - c_s^2) \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + \alpha c_{pz}^2 (c_{pz}^2 - c_s^2) \frac{\partial^4 q}{\partial y^4}(t, \mathbf{x}_b) \\
& + \alpha c_s^2 (c_{pz}^2 - c_s^2) \frac{\partial^4 q}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + c_s^2 c_{px}^2 \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) \\
& + \alpha c_s^2 (c_{pz}^2 - c_s^2) \frac{\partial^4 q}{\partial y^4}(t, \mathbf{x}_b) + c_s^4 \frac{\partial^4 p}{\partial y^4}(t, \mathbf{x}_b) = 0 \\
& \frac{c_{px}^2 (v_{pn}^2 - c_s^2)}{\alpha} \frac{\partial^4 p}{\partial x^4}(t, \mathbf{x}_b) + (v_{pn}^2 - c_s^2)(c_{pz}^2 - c_s^2) \frac{\partial^4 q}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) \\
& + \frac{c_s^2 (v_{pn}^2 - c_s^2)}{\alpha} \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + \frac{c_{pz}^2 (v_{pn}^2 - c_s^2)}{\alpha} \frac{\partial^4 p}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) \\
& + c_{pz}^4 \frac{\partial^4 q}{\partial y^4}(t, \mathbf{x}_b) + c_{pz}^2 c_s^2 \frac{\partial^4 q}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + \frac{c_s^2 (v_{pn}^2 - c_s^2)}{\alpha} \frac{\partial^4 p}{\partial x^4}(t, \mathbf{x}_b) \\
& + c_{pz}^2 c_s^2 \frac{\partial^4 q}{\partial x^2 \partial y^2}(t, \mathbf{x}_b) + c_s^4 \frac{\partial^4 q}{\partial x^4}(t, \mathbf{x}_b) = 0
\end{aligned} \tag{5.11}$$

as the fourth-order boundary conditions. These conditions are considerably more complex than those for the isotropic case or indeed Equation 5.7. The process of deriving higher-order conditions can be repeated to derive expressions up to the order of the spatial discretisation.

Comparison of isotropic and anisotropic higher-order conditions for the Fletcher et al. [2009], Fowler et al. [2010] VTI formulation was carried out as in the previous section. Parameterisation was reused with the addition of $\sigma = 0.5$. A comparison of the effects of isotropic higher-order conditions and those given in Equations 5.9 and 5.11 is shown in Figure 5.4. The introduction of nonzero c_s removes the diamond shear wave artefact surrounding the source. As the majority of the misfit between the results of anisotropic and isotropic boundary conditions was present in this shear wave artefact in tests using Equation 5.1, the overall difference between the two wavefields is much smaller in this case. The difference in the reflected P-wave wavefront shown in Figure 5.4 is comparable to 5.1, even sharing the same distinct pattern (there are vertical bands of near-zero misfit either side of the centre of the wavefront in both). As in Figure 5.2, doubling the spatial and temporal resolution causes the apparent con-

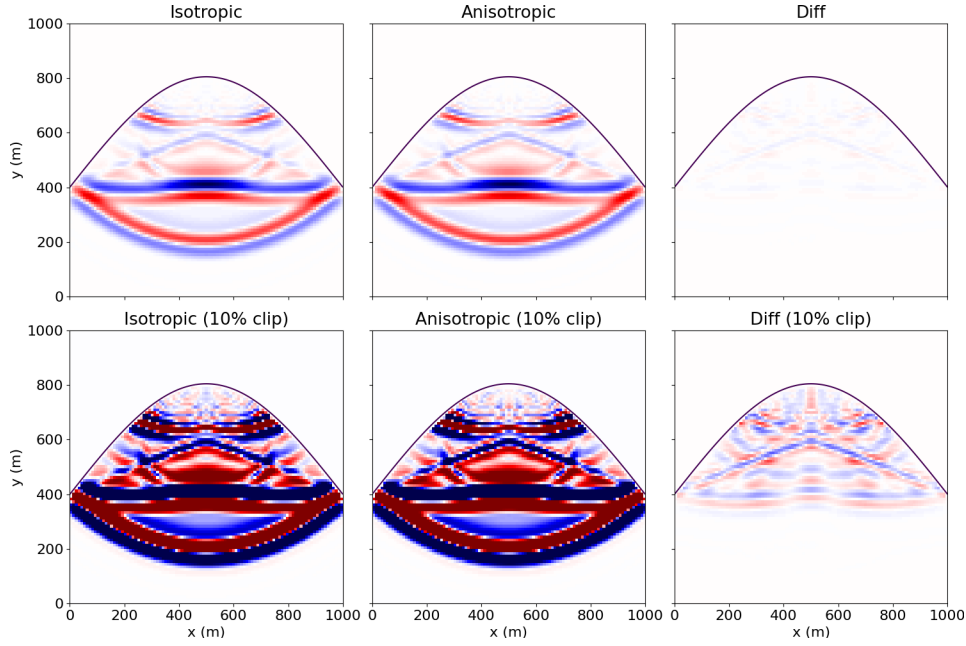


Figure 5.4.: Comparing effects of isotropic free-surface conditions (shown in the left column) to those specified in Equations 5.9 and 5.11 (shown in the middle column). The difference between the two is shown in the right column using the same colour scale. The upper row shows wavefield snapshots at 600ms, with all wavefields normalised against the maximum absolute value of the p wavefield in the isotropic case. To emphasise small differences between the two wavefields, the lower row of subfigures show the same wavefields clipped to 10% of the maximum absolute value of the p wavefield in the isotropic case.

vergence of solutions calculated with isotropic and anisotropic boundary conditions, shown in Figure 5.5.

The shear wave conversion artefacts observed in both Figures 5.4 and 5.5 where the wavefield interacts with the free-surface are of interest. Introducing nonzero c_s results in a markedly different geometry for the apparent P-S conversion wavefront versus that observed with Equation 5.1, being much more clearly defined and closer to the reflected P wavefront. Nonzero c_s also increases the wavelength of this feature, reducing the degree of numerical dispersion in the final wavefield. Incidence of the shear artefact upon the surface generates a coherent, near-horizontal reflection. It is postulated that this reflection is an artefact corresponding to the S-P conversion expected in the elastic case.

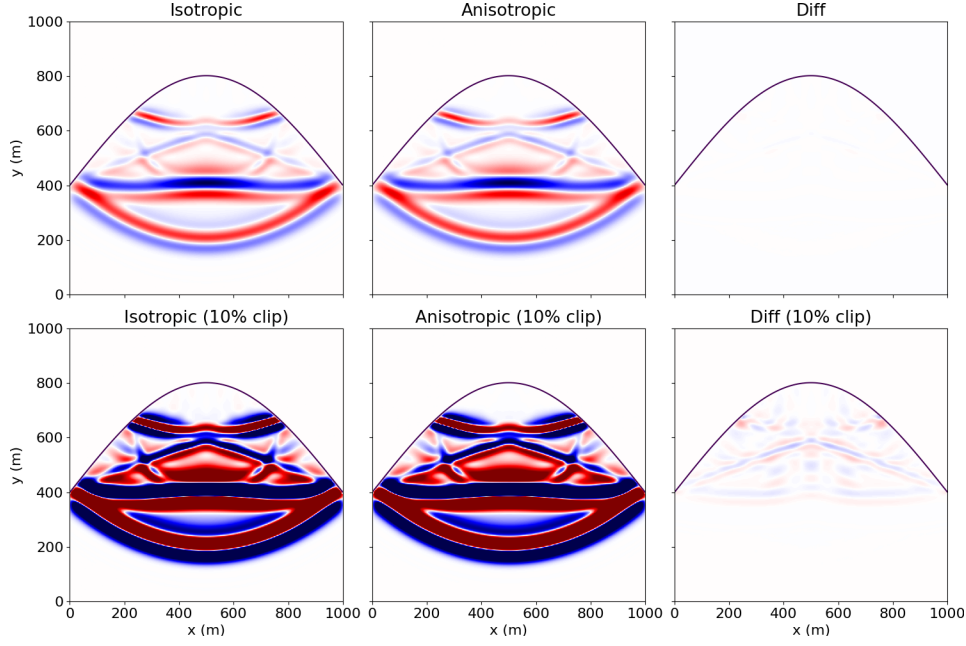


Figure 5.5.: Repeating the experiment shown in Figure 5.4 with twice the spatial and temporal resolution. Subfigures are arranged as in Figure 5.4, again using the maximum absolute value of the p field in the case featuring the isotropic free surface for normalisation and clipping.

5.2.2. Real-Topography Example

To examine the effect of nonzero c_s in the presence of topography, a model parameterised as in Section 5.1.3 was created, setting c_s according to $\sigma = 0.5$. Higher-order boundary conditions given in Equations 5.9 and 5.11 are imposed on the surface alongside the zero-stress condition. Snapshots of the wavefield are shown in Figure 5.6.

P-waves reflected from the free surface have near-identical geometries to those in Figure 5.3. However, additional wavefronts associated with the more pronounced P-S conversion artefact can be observed trailing these waves. The aforementioned S-P conversions associated with the residual shear waves are apparent in the final snapshot, particularly where this wavefront interacts with the crater rim. Whilst the amplitude of these is fairly large, they are well-separated from the main reflected P-wave arrivals and are thus expected to be relatively easily accounted for in any overarching algorithm. Qualitatively, the results of this test are promising, demonstrating compatibility between the proposed immersed boundary

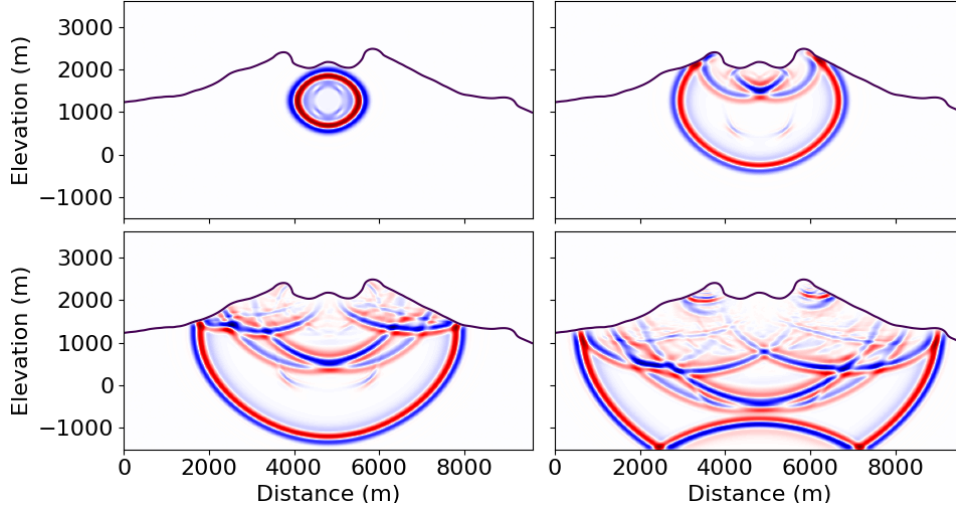


Figure 5.6.: Solving the VTI equations of Fletcher et al. [2009], Fowler et al. [2010] in the presence of the 2D Mt St Helens topographic profile. Snapshots are at the same timesteps as in Figure 5.3.

treatment and one of the more common VTI formulations. Reflection geometries are consistent with previous experiments and no aberrations are apparent from the irregular topography appears.

5.3. Fletcher TTI

The VTI equations of Fowler et al. [2010] can be extended to TTI by rotating the differential operators H_1 and H_2 given in Equation 5.8 [Fletcher et al., 2009]. In 2D, the rotated operators given in Equation 1.43 simplify to

$$\begin{aligned} H_1 &= \sin^2\theta \frac{\partial^2}{\partial x^2} + \cos^2\theta \frac{\partial^2}{\partial y^2} + \sin 2\theta \frac{\partial^2}{\partial x \partial y}, \\ H_2 &= \nabla^2 - H_1. \end{aligned} \quad (5.12)$$

The nonzero shear wave velocity, c_s , used by Fletcher et al. [2009] is found to improve stability in the presence of sudden changes in dip or dip azimuth; zero c_s is found to be prone to instability when the shear wave artefact is incident upon such interfaces [Fletcher et al., 2009]. Whilst long-time instabilities can still occur, even after smoothing the model [Duveneck and Bakker, 2011], these equations are still commonly used, offering a tradeoff between stability and computational cost. Formulating

the equations to contain derivatives of material parameters Duveck and Bakker [2011], Louboutin et al. [2018] yields greater stability at the expense of increased computational cost [Duveck and Bakker, 2011].

5.3.1. Free-Surface Immersed Boundaries

As with the previously-discussed VTI equations, one can apply either boundary conditions from the isotropic case, or derive higher-order conditions from the governing equations. The presence of rotated operators increases the complexity of these governing equations, producing unwieldy higher-order boundary conditions. This increased complexity may become problematic in high-order discretisations, introducing a trade-off between managing complexity and accuracy. Achieving acceptable accuracy without the need to derive complex higher-order boundary conditions is desirable, although it is important to first verify that isotropic boundary conditions behave comparably to anisotropic ones.

Given zero stress on the surface, the second-order conditions given in Equation 5.9 must also hold, albeit using the rotated operators from Equation 5.12. Expanding operators H_1 and H_2 , the second-order boundary conditions

$$\begin{aligned}
& c_{px}^2 \left((1 - \sin^2\theta) \frac{\partial^2 p}{\partial x^2}(t, \mathbf{x}_b) + (1 - \cos^2\theta) \frac{\partial^2 p}{\partial y^2}(t, \mathbf{x}_b) + \sin 2\theta \frac{\partial^2 p}{\partial x \partial y}(t, \mathbf{x}_b) \right) \\
& + \alpha (c_{pz}^2 - c_s^2) \left(\sin^2\theta \frac{\partial^2 q}{\partial x^2}(t, \mathbf{x}_b) + \cos^2\theta \frac{\partial^2 q}{\partial y^2}(t, \mathbf{x}_b) + \sin 2\theta \frac{\partial^2 q}{\partial x \partial y}(t, \mathbf{x}_b) \right) \\
& + c_s^2 \left(\sin^2\theta \frac{\partial^2 p}{\partial x^2}(t, \mathbf{x}_b) + \cos^2\theta \frac{\partial^2 p}{\partial y^2}(t, \mathbf{x}_b) + \sin 2\theta \frac{\partial^2 p}{\partial x \partial y}(t, \mathbf{x}_b) \right) = 0 \\
& \frac{v_{pn}^2 - c_s^2}{\alpha} \left((1 - \sin^2\theta) \frac{\partial^2 p}{\partial x^2}(t, \mathbf{x}_b) + (1 - \cos^2\theta) \frac{\partial^2 p}{\partial y^2}(t, \mathbf{x}_b) + \sin 2\theta \frac{\partial^2 p}{\partial x \partial y}(t, \mathbf{x}_b) \right) \\
& + c_{pz}^2 \left(\sin^2\theta \frac{\partial^2 q}{\partial x^2}(t, \mathbf{x}_b) + \cos^2\theta \frac{\partial^2 q}{\partial y^2}(t, \mathbf{x}_b) + \sin 2\theta \frac{\partial^2 q}{\partial x \partial y}(t, \mathbf{x}_b) \right) \\
& + c_s^2 \left((1 - \sin^2\theta) \frac{\partial^2 q}{\partial x^2}(t, \mathbf{x}_b) + (1 - \cos^2\theta) \frac{\partial^2 q}{\partial y^2}(t, \mathbf{x}_b) + \sin 2\theta \frac{\partial^2 q}{\partial x \partial y}(t, \mathbf{x}_b) \right) = 0,
\end{aligned} \tag{5.13}$$

are obtained. With the aid of the SymPy symbolic algebra package, fourth-order boundary conditions were derived by taking the second derivative of Equation 5.13 with respect to time before substituting Equation 1.41 and assuming locally-constant material parameters. The expressions produced

were too verbose to present here however.

Taking parameters from the test case shown in Figure 5.4 with the addition of $\theta = 45^\circ$, comparison of isotropic and anisotropic boundary conditions was repeated for the TTI equations of Fletcher et al. [2009]. The results of these two boundary treatments are compared in Figure 5.7.

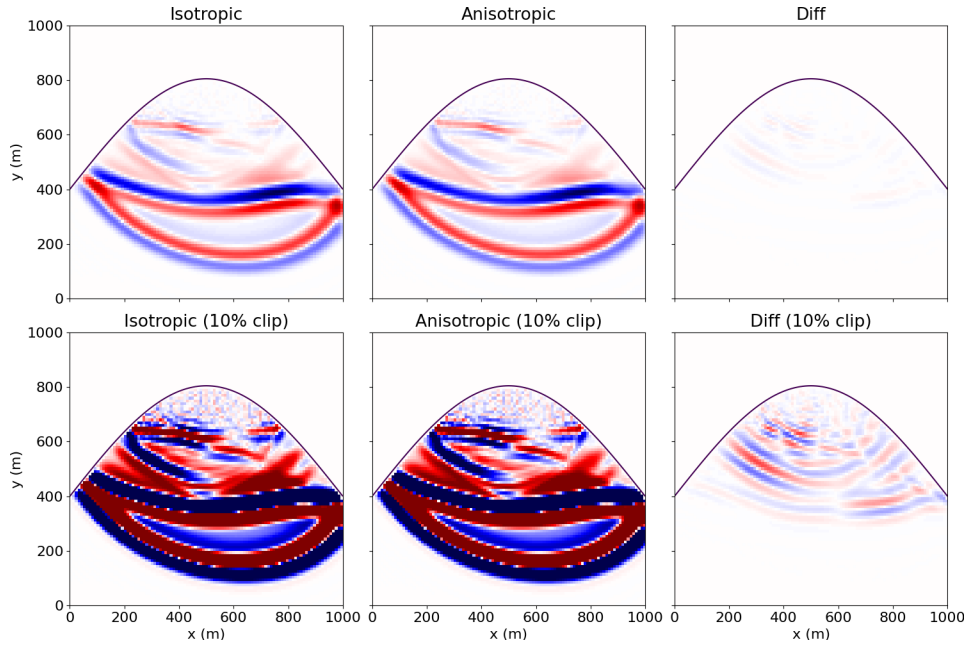


Figure 5.7.: Comparing effects of isotropic free-surface conditions (shown in the left column) to those specified in Equations 5.9 and 5.11 (shown in the middle column) with the TTI equations of [Fletcher et al., 2009]. The difference between the two is shown in the right column using the same colour scale. The upper row shows wavefield snapshots at 600ms, with all wavefields normalised against the maximum absolute value of the p wavefield in the isotropic case. To emphasise small differences between the two wavefields, the lower row of subfigures show the same wavefields clipped to 10% of the maximum absolute value of the p wavefield in the isotropic case.

As in the case of Fletcher VTI, use of isotropic higher-order boundary conditions in place of those derived via the governing equations results in a minor misfit in the reflected P-wavefront, alongside small differences in the behaviour of the residual S-wave field. Note that this misfit is largest in the orientation of the fast axis (oriented top left to bottom right), near-zero diagonal to the fast and slow axes (on the vertical), and smaller and reversed on the slow axis (relative to that observed on the fast axis). However, the scale of this difference is small relative to the amplitude of

the wavefield.

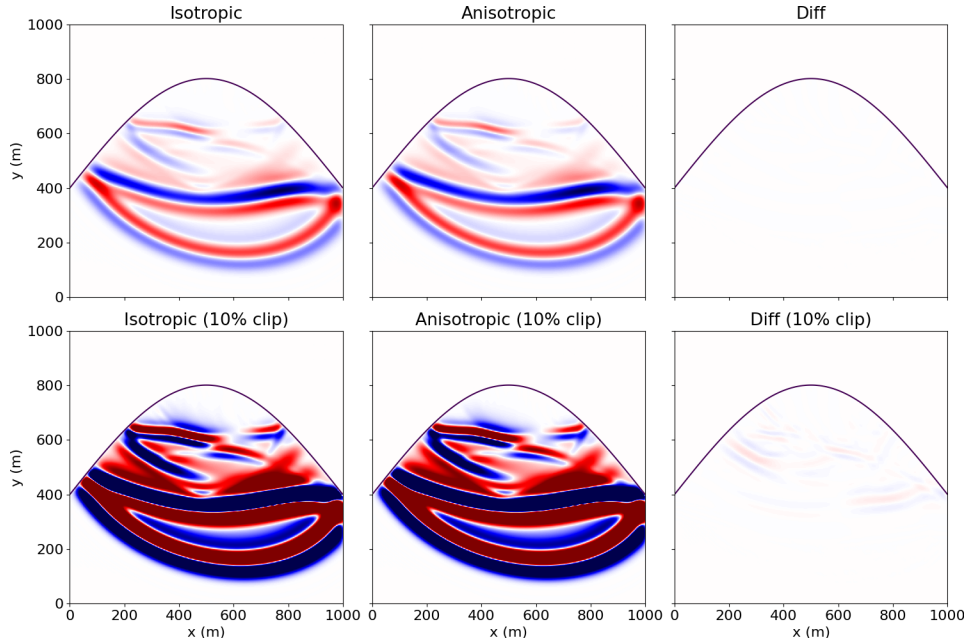


Figure 5.8.: Repeating the experiment shown in Figure 5.7 with twice the spatial and temporal resolution. Subfigures are arranged as in Figure 5.7, again using the maximum absolute value of the p field in the case featuring the isotropic free surface for normalisation and clipping.

To check that the behaviour of the two sets of boundary conditions converges as expected as resolution is refined, the comparison was repeated with a grid increment of 5m whilst holding the Courant number constant. Results are shown in Figure 5.8. As expected, considerable reduction in the difference between the results of the two sets of boundary conditions is observed. There is also reduced noise trailing the residual shear wavefield and associated conversions produced at the free surface in both cases.

Given the complexity of the anisotropic boundary conditions for this particular set of equations and the fairly minimal effect upon the outcome (particularly when the wavefield is well-resolved), isotropic boundary conditions are likely to suffice for many applications. Misfit between wavefields where each set of boundary conditions is applied converges to zero as resolution is refined, implying that the isotropic approximation is reasonable. Furthermore, the governing equations themselves are an approximation made in pursuit of simplicity; approximating the free surface in a similar manner is potentially justifiable.

5.3.2. Real-Topography Example

In light of the complexity introduced by anisotropic higher-order conditions and their minimal effect upon the wavefield, for the real-world topography test case, isotropic boundary conditions given in Equations 3.4 and 3.5 were applied on the surface. The model was parameterised as in the test case shown in Figure 5.6 with the addition of $\theta = 45$ deg. The discretisation and geometry were kept the same, the only changes being the equations solved and boundary conditions imposed by the immersed boundary treatment. Snapshots are shown in Figure 5.9. As in previous examples, the geometry of reflected wavefronts appears reasonable and free of artefacts.

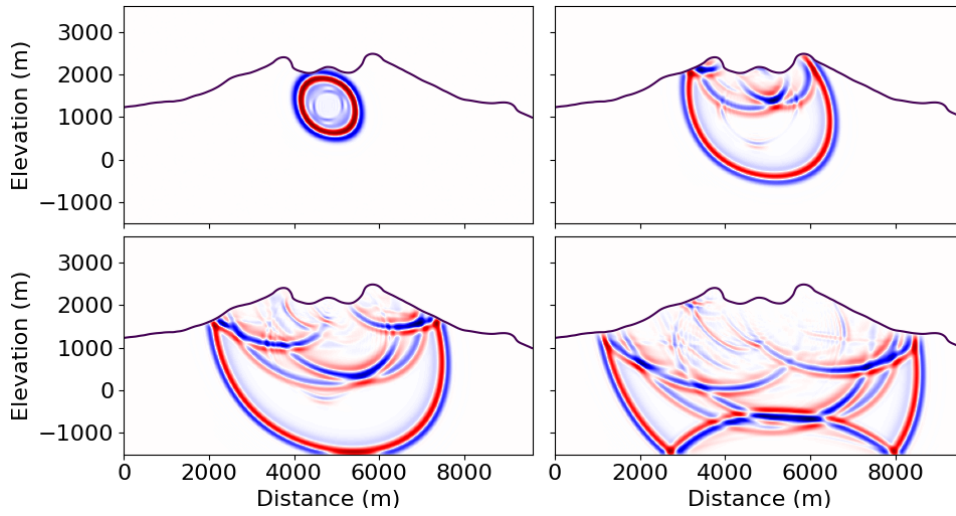


Figure 5.9.: Solving the TTI equations of Fletcher et al. [2009] in the presence of the 2D Mt St Helens topographic profile. Snapshots are as in previous examples.

5.4. Conclusions

This Chapter demonstrates that the generalised immersed boundary approach developed in Chapter 2 can be readily applied to a range of wave equations formulated for P-wave propagation in the presence of transverse isotropy. To the author’s knowledge, an immersed free surface for a pseudoacoustic TTI equation has not been previously detailed in the literature, despite widespread use of such equations in imaging workflows including RTM and FWI and the recognised challenges of topogra-

phy. The effectiveness of the generalised immersed boundary treatment in handling the pronounced, uneven topography in test cases featuring real-world geometries makes this approach appealing for practical applications. Furthermore, the immersed boundary treatment was found to be stable for all boundary conditions and wave equations investigated in this Chapter.

Deriving higher-order counterparts to the free-surface condition using the anisotropic wave equations considered in this Chapter results in considerably more complex sets of conditions than encountered in the isotropic case. As these conditions dictate a relationship between the p and q (or τ_H and τ_V) fields, this results in larger, more computationally expensive stencils. This cost is incurred in both the generation of stencils and within the FD kernel itself. The difference between the effects of isotropic and anisotropic free surfaces was found to be minimal, converging to zero as resolution is refined. There is a consequent trade-off between computational cost, complexity, and accuracy to be explored, the optimal mix likely being application-dependent. Whilst the framework was capable of generating boundary treatments for all conditions considered, the isotropic conditions were found to be sufficient for implementing free surfaces with the equations considered herein.

5.5. Future Work

The approach taken would benefit from validation to ensure confidence in the behaviour of the generalised boundary treatment when applied to these equations. In a half space, there should be no difference between rotating the plane of anisotropy in one direction and rotating the topographic profile and associated source-receiver array in the other [Huiskes et al., 2017]. Shot gathers generated by these two approaches should be consistent with one another. Holding some fixed angle between surface and anisotropy and rotating both together should also be expected to yield consistent gathers. Carrying out this test with both planar and irregular surfaces would ensure consistency in the behaviour of the immersed free surface.

Further test cases in 3D, potentially featuring combined geological and topographic models would be of use as a stepping stone to practical ap-

plications such as imaging algorithms for settings with pronounced, irregular topography. Using the generalised immersed boundary treatment to include topography in imaging workflows based on VTI and TTI wave equations would aid in understanding its footprint in such applications, and how this interacts with anisotropic effects.

For complex subsurface models [Duveneck and Bakker, 2011], or particular imaging algorithms [Louboutin et al., 2018], more sophisticated TTI wave equations are desirable. A common feature of these equations is derivatives not of a single field but of a product of fields or some compound expression. TTI wave equations such as those proposed by Duveneck and Bakker [2011] or Louboutin et al. [2018] are generally more stable in the presence of rapidly varying subsurface properties associated with the geological processes which produce pronounced topography, containing derivatives of these properties within the update equations [Duveneck and Bakker, 2011]. Rigorously derived adjoint equations used for backpropagation in RTM and FWI feature a similar structure [Louboutin et al., 2018]. Given that TTI wave equations of this type yield more reliable and accurate results when used in overarching algorithms, they present a desirable target for immersed boundary topography implementations.

The generalised mathematical approach outlined in Chapter 2 is nominally applicable to the equations of Duveneck and Bakker [2011] and Louboutin et al. [2018] among others: exploring these equations would encourage adoption of immersed boundaries to implement topography in the imaging algorithms for which these equations are predominantly used. More complex differential operators, whilst resulting in a more verbose specification of the governing PDEs, will not contain additional derivatives for which modified stencils are to be constructed (although there may be a need to decide on suitable continuations of material properties). However, given the increased complexity of these acoustic TTI formulations, derivation of higher-order boundary conditions is unlikely to be as straightforward as in the examples detailed in this Chapter. Exploring the applications of the generalised immersed boundary treatment to such equations is anticipated to be an informative exercise for further extension of the mathematical and software frameworks.

Chapter 6

Immersed Boundaries and Elastic Wave Equations

Imaging algorithms including FWI, RTM, and least-squares RTM (LSRTM) yield higher resolution and fewer artefacts when based upon elastic propagators [Feng and Schuster, 2017, Agudo et al., 2018], with kinematic inaccuracies resulting from simulating P-waves translating to amplitude mismatches and missing arrivals in the simulated data [Agudo et al., 2018]. Whilst this can be addressed by specialised workflows such as matching filters [Agudo et al., 2018] or residual error estimation [Hobro et al., 2014], computational advancements have made fully elastic workflows increasingly feasible.

Topographic effects are rarely accounted for in elastic imaging workflows due to the additional computational complexities introduced by their inclusion [Nuber et al., 2016]. Accurate, efficient forward solvers remain elusive [Nuber et al., 2016], with the majority of existent studies using FEM solvers to handle topographic variation [Romdhane et al., 2011, Nuber et al., 2016, Borisov et al., 2018, 2020, Liu et al., 2021]. Errors introduced by staircased surface approximations are sufficient that nominally-more-expensive FEM approaches may yield superior accuracy at equivalent computational cost [Mulder, 1996, Zhebel et al., 2014], cited by Romdhane et al. [2011], Borisov et al. [2018, 2020] as a barrier to leveraging the reduced computational footprint of FDMs versus FEMs. Implementing the elastic free surface as an immersed boundary would make oversampling unnecessary, greatly reducing computational cost, whilst

avoiding the need to transform the free-surface conditions, as in the case of coordinate-transform FD [Hestholm and Ruud, 1994, Zhang and Chen, 2006].

Immersed elastic free surfaces have been explored by Lombard et al. [2008], Almuheidib and Toksoz [2015], and Gao et al. [2015]. Solving the equations of Virieux [1986] (Equations 1.20 and 1.19), Gao et al. [2015] explores the immersed boundary approach in combination with a staggered discretisation, whilst Lombard et al. [2008] and Almuheidib and Toksoz [2015] use ADER schemes based on a single grid. Differing approaches to field continuation were taken, Almuheidib and Toksoz [2015] extending the characteristic-variable-based free surface implementation of Bayliss et al. [1986] to non-horizontal surfaces through the application of a rotation matrix, interpolating onto mirror points before projecting these values orthogonally across the boundary onto a ghost grid. Lombard et al. [2008], Gao et al. [2015] use N-dimensional Taylor-series extrapolation around boundary points, defined as the intersection of the unit normal from each ghost node to the boundary, to obtain ghost node values. Extending a circular support region, an overdetermined linear system is formed to obtain the extrapolant. A parametric representation of the boundary is used, enabling derivatives of the boundary conditions to be taken relative to the boundary surface [Gao et al., 2015], thereby accounting for local boundary curvature [Lombard et al., 2008]. In combination with ADER schemes, results for circular and inclined straight-line boundaries match exact solutions, and reasonable reflections are generated for a sinusoidal surface [Lombard et al., 2008]. [Lombard et al., 2008] used the long-time resonance of a wave inside a closed circle was used to validate the stability of their scheme. However, when a staggered scheme was used, numerical dispersion was required to suppress numerical instability, attenuating surface waves and reflections [Gao et al., 2015] whilst spatial discretisations of higher than second-order were entirely unstable. Lombard et al. [2008] reported similar issues applying their method to this discretisation, although did not attempt to rectify the problem.

The mesh-free topography implementation of Takekawa and Mikada [2019] incorporates the free-surface condition directly into the stencils rather than using a ghost grid to enforce the boundary condition. The efficacy of this approach is demonstrated in a test case featuring topography

from the SEG foothills model [Takekawa and Mikada, 2019]: a more realistic setup than those in Lombard et al. [2008], Almuheidib and Toksoz [2015], Gao et al. [2015]. This particular study made use of the second-order formulation of the elastic wave equation as in Kelly et al. [1976], although the first-order formulation can also be solved with mesh-free FD [Martin et al., 2013].

This Chapter outlines the application of immersed boundary treatments of the type outlined in Chapter 2 to isotropic, VTI, and TTI second-order elastic wave equations and isotropic first-order elastic wave equations. The derivation, behaviour, and stability of immersed elastic free surfaces are explored, and examples featuring synthetic and real topographic profiles presented.

6.1. 2nd-Order Elastic Wave Equations

Formulating the elastic wave equation in terms of particle displacement yields a coupled system of equations, given in Equation 1.4. Accurate reproduction of reflections and phase conversions from topography requires imposition of the corresponding elastic free-surface condition on the upper boundary.

In 2D, the zero-traction boundary condition which defines the free surface

$$\mathbf{n} \cdot \boldsymbol{\tau}(t, \mathbf{x}_b) = 0, \quad (6.1)$$

can be written in the form

$$\begin{aligned} n_x \tau_{xx}(t, \mathbf{x}_b) + n_y \tau_{xy}(t, \mathbf{x}_b) &= 0 \\ n_x \tau_{xy}(t, \mathbf{x}_b) + n_y \tau_{yy}(t, \mathbf{x}_b) &= 0. \end{aligned} \quad (6.2)$$

Given Equation 1.1, the relation between strain and particle displacement in Equation 1.2, and the isotropic stiffness tensor

$$\mathbf{C} = \begin{bmatrix} C_{11} & (C_{11} - 2C_{33}) & \\ & C_{11} & \\ & & C_{33} \end{bmatrix}, \quad (6.3)$$

where

$$C_{11} = \lambda + 2\mu, \quad (6.4)$$

and

$$C_{33} = \mu, \quad (6.5)$$

the first-order particle displacement boundary conditions follow:

$$\begin{aligned} n_x(\lambda + 2\mu) \frac{\partial u_x}{\partial x}(t, \mathbf{x}_b) + n_x \lambda \frac{\partial u_y}{\partial y}(t, \mathbf{x}_b) + n_y \mu \left(\frac{\partial u_x}{\partial y}(t, \mathbf{x}_b) + \frac{\partial u_y}{\partial x}(t, \mathbf{x}_b) \right) &= 0, \\ n_x \mu \left(\frac{\partial u_x}{\partial y}(t, \mathbf{x}_b) + \frac{\partial u_y}{\partial x}(t, \mathbf{x}_b) \right) + n_y(\lambda + 2\mu) \frac{\partial u_y}{\partial y}(t, \mathbf{x}_b) + n_y \lambda \frac{\partial u_x}{\partial x}(t, \mathbf{x}_b) &= 0. \end{aligned} \quad (6.6)$$

To obtain corresponding third-order derivatives, the second derivative of Equation 6.6 is taken with respect to time, and Equation 1.4 is substituted. Assuming constant material parameters, this yields

$$\begin{aligned} &n_x \lambda \mu \frac{\partial^3 u_y}{\partial y^3}(t, \mathbf{x}_b) + n_y \mu^2 \frac{\partial^3 u_x}{\partial y^3}(t, \mathbf{x}_b) + n_y(\lambda^2 + 2\lambda\mu + 2\mu^2) \frac{\partial^3 u_x}{\partial x^2 \partial y}(t, \mathbf{x}_b) \\ &+ n_x(\lambda + 2\mu)^2 \frac{\partial^3 u_x}{\partial x^3}(t, \mathbf{x}_b) + n_x \mu(2\lambda + 3\mu) \frac{\partial^3 u_x}{\partial x \partial y^2}(t, \mathbf{x}_b) \\ &+ n_x(\lambda^2 + 3\lambda\mu + \mu^2) \frac{\partial^3 u_y}{\partial x^2 \partial y}(t, \mathbf{x}_b) + n_y \mu(\lambda + 2\mu) \frac{\partial^3 u_y}{\partial x^3}(t, \mathbf{x}_b) \\ &+ n_y(\lambda^2 + 3\lambda\mu + 3\mu^2) \frac{\partial^3 u_y}{\partial x \partial y^2}(t, \mathbf{x}_b) = 0 \\ &n_y \lambda \mu \frac{\partial^3 u_x}{\partial x^3}(t, \mathbf{x}_b) + n_x \mu^2 \frac{\partial^3 u_y}{\partial x^3}(t, \mathbf{x}_b) + n_x(\lambda^2 + \lambda\mu + 3\mu^2) \frac{\partial^3 u_x}{\partial x^2 \partial y}(t, \mathbf{x}_b) \\ &+ n_y(\lambda^2 + 3\lambda\mu + \mu^2) \frac{\partial^3 u_x}{\partial x \partial y^2}(t, \mathbf{x}_b) + n_x \mu(\lambda + 2\mu) \frac{\partial^3 u_x}{\partial y^3}(t, \mathbf{x}_b) \\ &+ n_y \mu(2\lambda + 3\mu) \frac{\partial^3 u_y}{\partial x^2 \partial y}(t, \mathbf{x}_b) + n_x(\lambda^2 + 2\lambda\mu + 2\mu^2) \frac{\partial^3 u_x}{\partial x \partial y^2}(t, \mathbf{x}_b) \\ &+ n_y(\lambda + 2\mu)^2 \frac{\partial^3 u_y}{\partial y^3}(t, \mathbf{x}_b) = 0. \end{aligned} \quad (6.7)$$

For higher-order discretisations, the process is iterated to obtain additional conditions up to the spatial discretisation order.

6.1.1. Application to Simple Topography

An immersed elastic free surface, devised via the methodology detailed in Chapter 2, is demonstrated in a simple test case featuring a sinusoidal hill topography. The domain used is 1km square with 5m grid spacing, with homogeneous material properties to ensure clearly defined reflections from the surface. S-wave velocity was set to 0.65kms^{-1} , whilst P-wave velocity was set to 1kms^{-1} . The timestep was set to obtain a Courant number of 0.5. A fourth-order spatial discretisation and second-order explicit timestepping are used. Conditions detailed in Equations 6.6 and 6.7 are imposed on the immersed surface, whilst zero Dirichlet boundary conditions were applied on the remaining edges of the domain. An 8Hz Ricker source was injected into the horizontal displacement field, this being reflected in the radiation pattern (P-waves emanate horizontally in this field, whilst S-waves have greater amplitude in the vertical direction). This source was located 100m below the apex of the hill.

Figure 6.1 shows reflection of incident P and S-waves, alongside conversions between the two modes at the free surface. To highlight the phases present, the P and S wavefields were separated by taking the divergence and curl of particle displacement [Dellinger and Etgen, 1990], illustrated in Figure 6.2. Variants on this approach are widely used in elastic RTM approaches [Dellinger and Etgen, 1990, Zhang and Shi, 2019] to enable separate cross-correlation of P and S waves. Reflections are identifiable by their continuity with the incident wave, whilst conversions are disconnected from other wavefronts.

The performance of the immersed free surface is satisfactory for this particular test case, producing the expected phase conversions and yielding a smooth representation of the boundary surface, evidenced by the lack of scattering artefacts. The successful imposition of the elastic free surface conditions in the presence of a constantly changing surface normal direction associated with the sinusoid hill is promising for the application of the method to more realistic topographic profiles.

6.1.2. Application to Real-World Topography

To explore the efficacy of this immersed boundary treatment with more irregular topography, a second test case featuring the Mt St Helens topog-

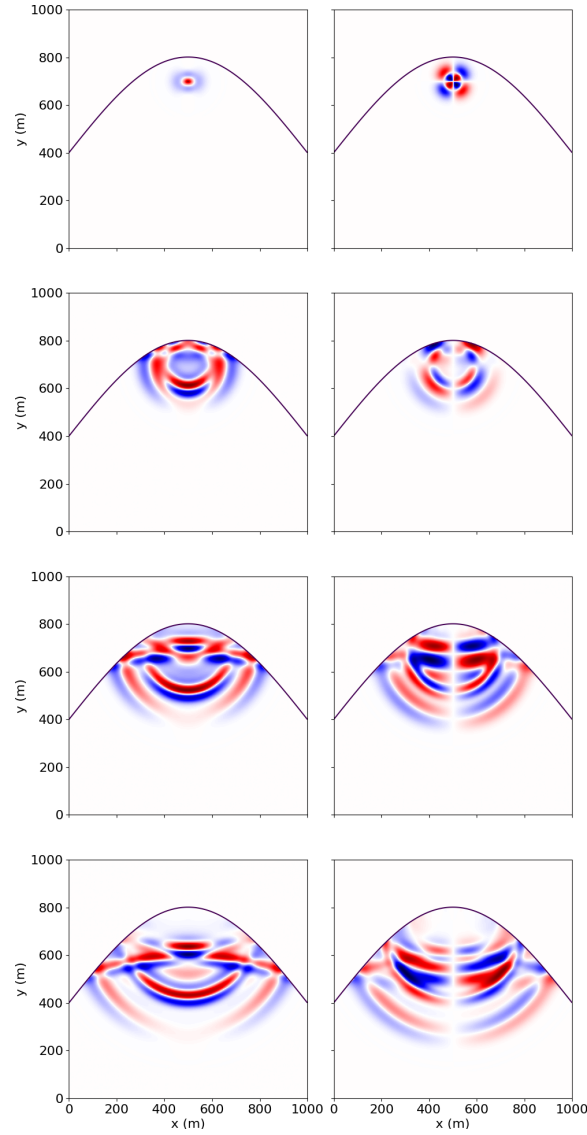


Figure 6.1.: Particle displacements u_x and u_y are shown in the left and right columns respectively. From top to bottom, snapshots are at 137.5ms, 275ms, 412.5ms, and 550ms. All fields are normalised for clarity.

raphy was developed. P-wave velocity was set to 2.5kms^{-1} , S-wave velocity was 1.5kms^{-1} , and density was set to 3gcm^{-3} . A Ricker source was injected in the same position as in previous examples, although its peak frequency was reduced to 6Hz to prevent numerical dispersion of high-wavenumber components resulting from slowly propagating shear phases. As with the previous test case, this wavelet was injected as a point source into the u_x

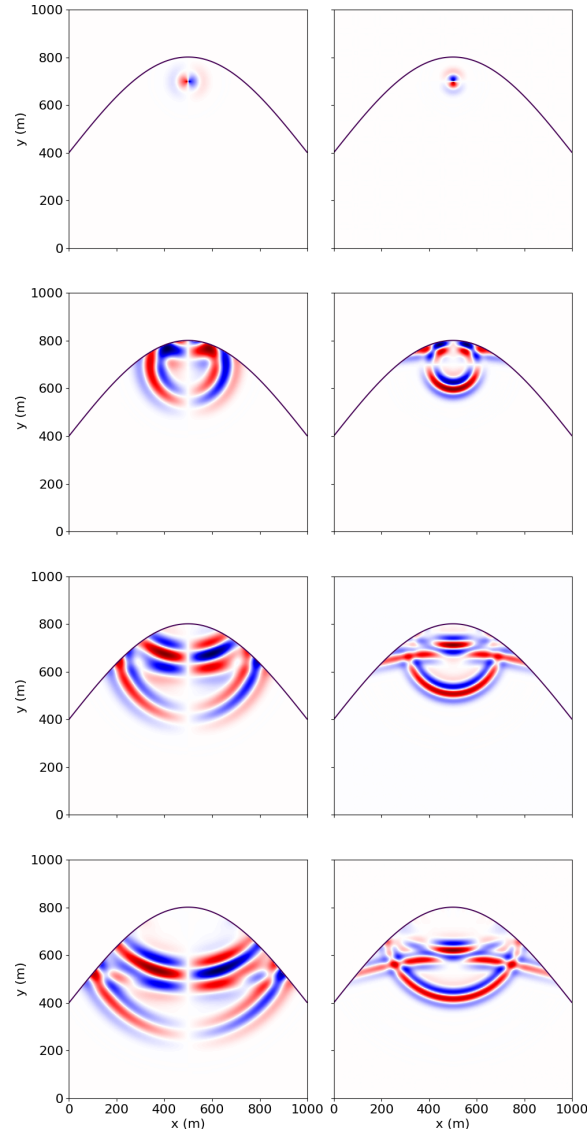


Figure 6.2.: Snapshots of $\nabla \cdot \mathbf{u}$ and the out-of-page component of $\nabla \times \mathbf{u}$ shown in Figure 6.1 at 137.5ms, 275ms, 412.5ms, and 550ms. The separated compressive (P) component is shown in the left column whilst the shear (S) component is shown on the right. All fields are normalised for clarity.

field.

Figure 6.3 exhibits smooth reflected wavefronts, without the characteristic spurious scattering associated with staircasing error. The PS conversion is visible as a near-horizontal wavefront, predominantly in the u_x field. The incidence of both P and S wavefronts on the surface and the presence of phase conversions result in a much more complex wavefield than the

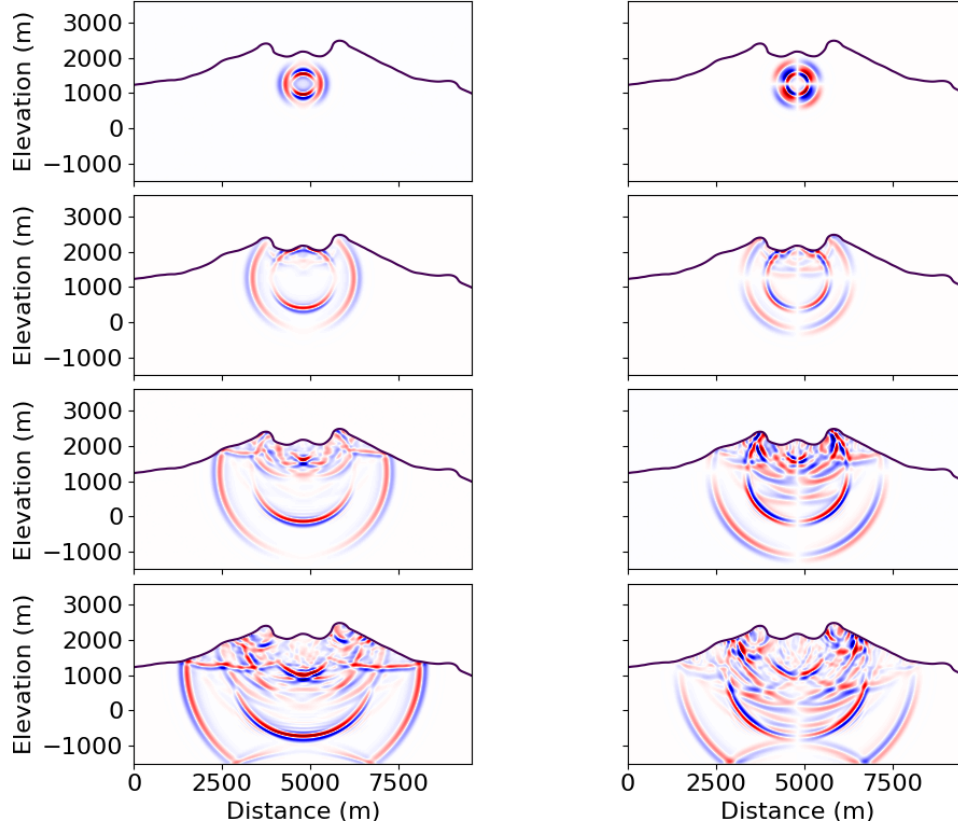


Figure 6.3.: Solving the 2nd-order elastic wave equations in the presence of the 2D Mt St Helens topographic profile. The u_x and u_y wavefields are shown in the left and right columns respectively, showing snapshots at 375ms, 750ms, 1125ms, and 1500ms. All snapshots are normalised for clarity.

acoustic case. This is particularly true around the crater rim where waves undergo multiple reflections. Note that phase conversion artefacts were observed in these areas in examples shown in Chapter 5; their presence in the elastic case confirms that these were expected.

To ensure that energy in the system is conserved, the sum of the L2 norms of the wavefields was calculated for each snapshot. No increase was observed during the simulation implying that the immersed boundary treatment does not introduce instability for this parameterisation, at least over the timescales considered in this test case.

6.2. VTI Free Surface

Elastic wave equations can naturally accommodate anisotropy in models of seismic wave propagation. As outlined in Chapter 5, the most common forms of anisotropy encountered in shallow crustal rocks are VTI and TTI.

Reduced to 2D, Equation 1.4 with $\mathbf{C}$ taken from Equation 1.6 yields the equations

$$\begin{aligned}\frac{\partial^2 u_x}{\partial t^2} &= v_{px}^2 \frac{\partial^2 u_x}{\partial x^2} + v_s^2 \frac{\partial^2 u_x}{\partial y^2} + \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \frac{\partial^2 u_y}{\partial x \partial y} \\ \frac{\partial^2 u_y}{\partial t^2} &= v_s^2 \frac{\partial^2 u_y}{\partial x^2} + v_p^2 \frac{\partial^2 u_y}{\partial y^2} + \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \frac{\partial^2 u_x}{\partial x \partial y}.\end{aligned}\quad (6.8)$$

To obtain the free-surface conditions corresponding with these equations, the derivation detailed in the preceding section is repeated, using Equations 6.1 and 1.1 alongside the VTI stiffness tensor given in Equation 1.6. In 2D, this yields the conditions

$$\begin{aligned}& n_x \left(v_{px}^2 \frac{\partial u_x}{\partial x}(t, \mathbf{x}_b) - v_s^2 \frac{\partial u_y}{\partial y}(t, \mathbf{x}_b) + \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \frac{\partial u_y}{\partial y}(t, \mathbf{x}_b) \right) \\ & + n_y v_s \left(\frac{\partial u_x}{\partial y}(t, \mathbf{x}_b) + \frac{\partial u_y}{\partial x}(t, \mathbf{x}_b) \right) = 0, \\ & n_x v_s \left(\frac{\partial u_x}{\partial y}(t, \mathbf{x}_b) + \frac{\partial u_y}{\partial x}(t, \mathbf{x}_b) \right) \\ & + n_y \left(v_p^2 \frac{\partial u_y}{\partial y}(t, \mathbf{x}_b) - v_s^2 \frac{\partial u_x}{\partial x}(t, \mathbf{x}_b) + \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \frac{\partial u_x}{\partial x}(t, \mathbf{x}_b) \right) = 0.\end{aligned}\quad (6.9)$$

Taking the second derivative of Equation 6.9 with respect to time and

substituting Equation 6.8 yields

$$\begin{aligned}
& n_x v_{pn}^2 v_{px}^2 \frac{\partial^3 u_x}{\partial x^3}(t, \mathbf{x}_b) + n_y v_s^4 \frac{\partial^3 u_x}{\partial y^3}(t, \mathbf{x}_b) + n_y v_{pn}^2 v_s^2 \frac{\partial u_y}{\partial x^3}(t, \mathbf{x}_b) \\
& + n_y \left(v_p^2 v_{pn}^2 - v_p^2 v_s^2 + v_s^4 - v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_x}{\partial x^2 \partial y}(t, \mathbf{x}_b) \\
& + n_x \left(v_{px}^2 v_s^2 + v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_x}{\partial x \partial y^2}(t, \mathbf{x}_b) \\
& + n_x \left(-v_{pn}^2 v_s^2 + v_{pn}^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} + v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_y}{\partial x^2 \partial y}(t, \mathbf{x}_b) \\
& + n_y \left(v_p^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} + v_s^4 \right) \frac{\partial^3 u_y}{\partial x \partial y^2}(t, \mathbf{x}_b) \\
& + n_x \left(-v_s^4 + v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_y}{\partial y^3}(t, \mathbf{x}_b) = 0 \\
& n_x v_p^2 v_s^2 \frac{\partial^3 u_x}{\partial y^3}(t, \mathbf{x}_b) + n_x v_s^4 \frac{\partial^3 u_y}{\partial x^3}(t, \mathbf{x}_b) + n_y v_p^4 \frac{\partial^3 u_y}{\partial y^3}(t, \mathbf{x}_b) \\
& + n_x \left(v_{px}^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} + v_s^4 \right) \frac{\partial^3 u_x}{\partial x^2 \partial y}(t, \mathbf{x}_b) \\
& + n_y \left(-v_s^4 + v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_x}{\partial x^3}(t, \mathbf{x}_b) \\
& + n_y \left(-v_p^2 v_s^2 + v_p^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} + v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_x}{\partial x \partial y^2}(t, \mathbf{x}_b) \\
& + n_y \left(v_p^2 v_s^2 + v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_y}{\partial x^2 \partial y}(t, \mathbf{x}_b) \\
& + n_x \left(v_p^2 v_{pn}^2 - v_{pn}^2 v_s^2 + v_s^4 - v_s^2 \sqrt{(v_p^2 - v_s^2)(v_{pn}^2 - v_s^2)} \right) \frac{\partial^3 u_y}{\partial x \partial y^2}(t, \mathbf{x}_b) = 0,
\end{aligned} \tag{6.10}$$

these being the 3rd-order boundary conditions required for the subsequent 4th-order-accurate examples. A symbolic algebra package such as SymPy is beneficial in deriving these boundary conditions, particularly for higher orders, due to the verbosity and complexity of the expressions.

Unlike the anisotropic pseudoacoustic cases discussed in Chapter 5, the imposition of isotropic boundary conditions upon the VTI equations was found to result in severe numerical instability.

6.2.1. Application to Simple Topography

To aid qualitative comparison to the isotropic case, a sinusoid hill test case featuring VTI was developed. The isotropic and VTI cases are expected to exhibit visibly different propagation behaviour; whilst a similar pattern of wavefronts should be produced in both cases, the positioning of these wavefronts will be altered in the presence of anisotropy. Parameterisation and geometry from the isotropic test case were reused, with the addition of constant anisotropy parameters $\delta = 0.1$ and $\epsilon = 0.25$.

The effects of anisotropy upon the interaction of the wavefield with the simple sinusoidal hill topography are shown in Figure 6.4. Comparing the wavefield snapshots shown in this figure to those in Figure 6.1, it is apparent that the vertical extent of the radiating P-wavefront matches that in the isotropic case at each snapshot. However, the presence of VTI horizontally stretches the P-wavefront into an elliptical shape, whilst the direct S-wavefront becomes squared off, altering the relative positions of the reflections and phase conversions. Separated wavefields are shown in Figure 6.5. Despite anisotropic effects, both u_x and u_y fields share features recognisable in the isotropic case.

From this test, the immersed boundary treatment appears compatible with the VTI elastic free surface conditions, producing wavefields consistent with the isotropic case. Furthermore, the boundary treatment was not found to create instability over the timescales considered in this test case.

6.2.2. Application to Real-World Topography

The Mt St Helens test case detailed in Section 6.1.2 was revisited to better establish the suitability of the method in combination with more complex surfaces. Parameterisation is as in Section 6.1.2, albeit specifying $\delta = 0.1$ and $\epsilon = 0.25$. Wavefield snapshots for this test case are shown in Figure 6.6.

The effect of anisotropy upon the incident wavefronts considerably alters the pattern of conversions and reflections versus the isotropic case, the most obvious example being the distinctive PS conversion generated by the incident P-wave as it meets the surface. This is particularly apparent in the later timesteps, where the position of this wavefront is notably

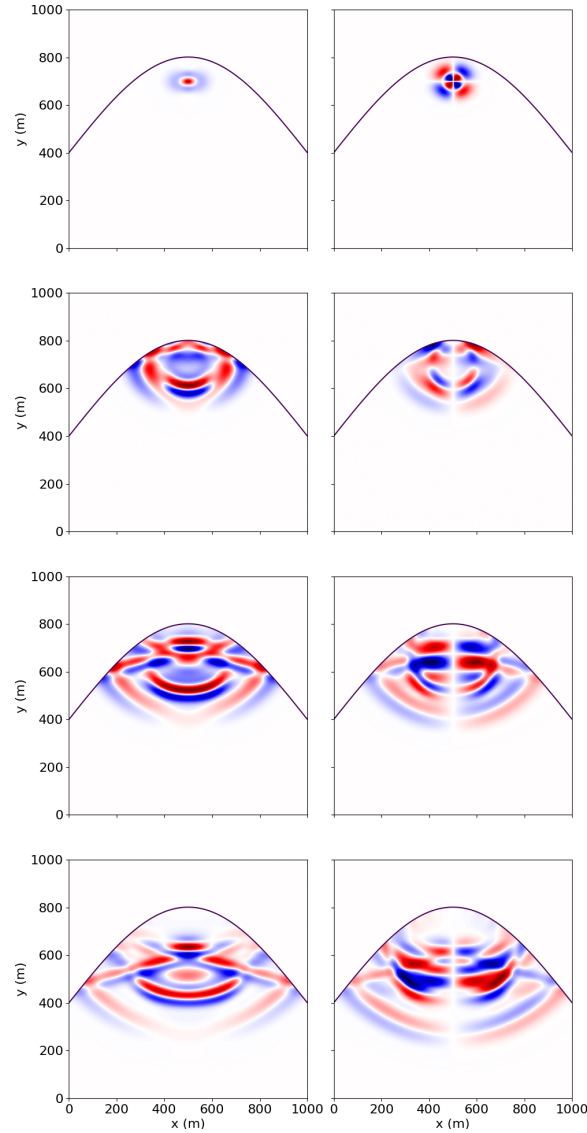


Figure 6.4.: Particle displacements u_x and u_y for a model featuring VTI are shown in the left and right columns respectively. From top to bottom, snapshots are at 137.5ms, 275ms, 412.5ms, and 550ms. All fields are normalised for clarity.

deeper than in the isotropic case shown in Figure 6.3 as a result of the earlier incidence of the P-wavefront upon the surface. Whilst the main body of reflections differs notably from those in Figure 6.3, most major wavefronts can be matched to those in the isotropic case.

As in the isotropic case, the sum of the L2 norms of the wavefields was found not to increase after source injection, implying that the boundary

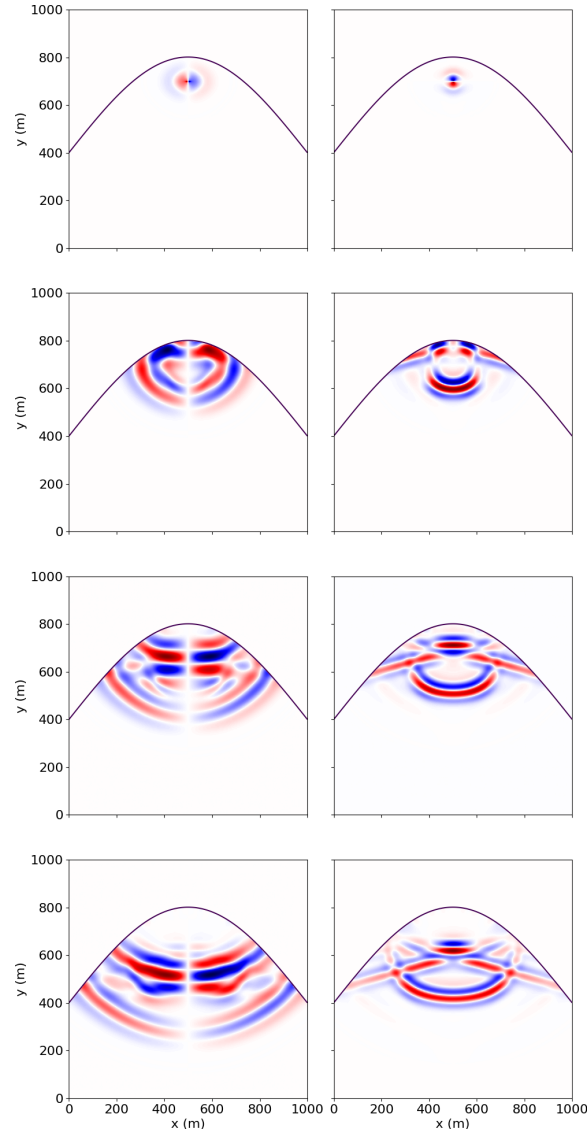


Figure 6.5.: Snapshots of $\nabla \cdot \mathbf{u}$ and the out-of-page component of $\nabla \times \mathbf{u}$ shown in Figure 6.4 at 137.5ms, 275ms, 412.5ms, and 550ms. The separated compressive (P) component is shown in the left column whilst the shear (S) component is shown on the right. All fields are normalised for clarity.

treatment did not introduce instability over the course of the simulation. Overall, performance is satisfactory, yielding expected behaviour without obvious artefacts.

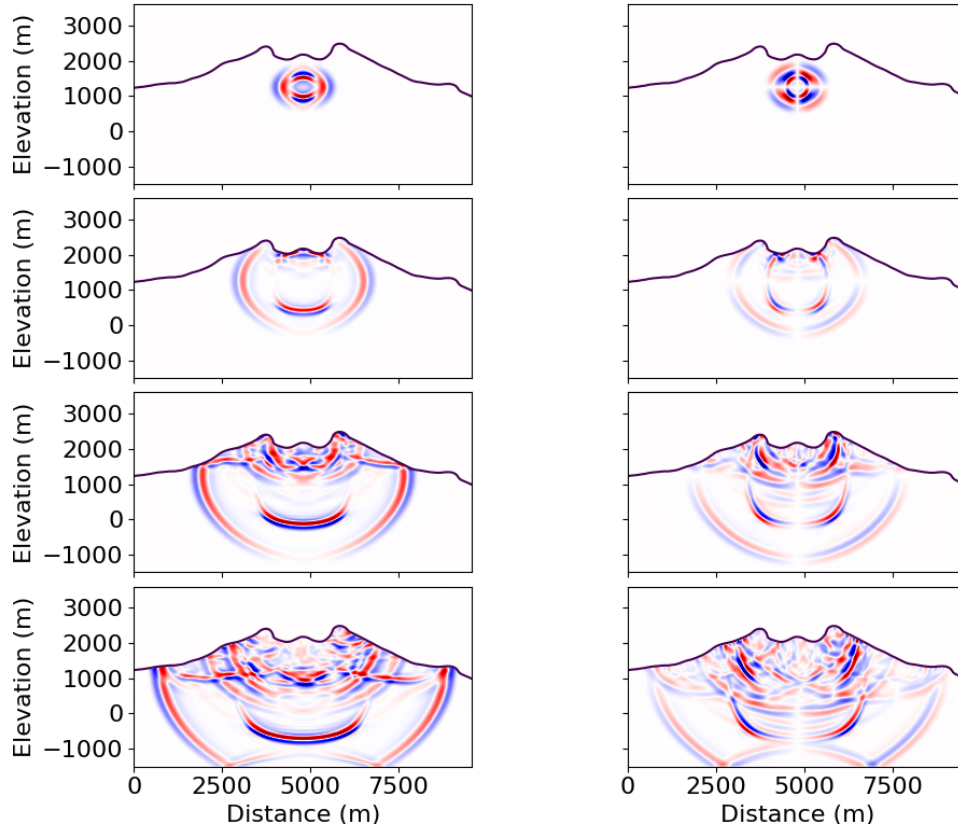


Figure 6.6.: Solving the 2nd-order VTI elastic wave equations in the presence of the 2D Mt St Helens topographic profile. The u_x and u_y wavefields are shown in the left and right columns respectively, showing snapshots at 375ms, 750ms, 1125ms, and 1500ms. All snapshots are normalised for clarity.

6.3. TTI Free Surface

As in Chapter 5, whilst in layered sedimentary rocks the symmetry axis of the anisotropy is commonly subvertical, a rotation must be applied where this is not the case. However, unlike the pseudoacoustic case, there are two avenues by which to achieve this: either the derivative operators or the stiffness tensor can be rotated. When rotating derivative operators, it is necessary to include derivatives of the trigonometric operators within the rotated FD operators to ensure stability in the presence of rapid changes in tilt or azimuth Duveneck and Bakker [2011], Hu et al. [2018], increasing computational cost and complexity. Rotating the stiffness matrix $\mathbf{C}$ as in Yang et al. [2015], Barucq et al. [2014] represents an alternative, straightforward means of accommodating the rotation of the symmetry axis.

Rotating the VTI stiffness tensor yields a dense tensor: TTI is a special case of triclinic anisotropy. Deriving free surface boundary conditions with the rotated stiffness tensor, the 1st-order boundary conditions

$$\begin{aligned}
& n_x \left(C'_{11} \frac{\partial u_x}{\partial x}(t, x_b) + C'_{12} \frac{\partial u_y}{\partial y}(t, x_b) + C'_{13} \left(\frac{\partial u_x}{\partial y}(t, x_b) + \frac{\partial u_y}{\partial x}(t, x_b) \right) \right) \\
& + n_y \left(C'_{13} \frac{\partial u_x}{\partial x}(t, x_b) + C'_{23} \frac{\partial u_y}{\partial y}(t, x_b) + C'_{33} \left(\frac{\partial u_x}{\partial y}(t, x_b) + \frac{\partial u_y}{\partial x}(t, x_b) \right) \right) = 0 \\
& n_x \left(C'_{13} \frac{\partial u_x}{\partial x}(t, x_b) + C'_{23} \frac{\partial u_y}{\partial y}(t, x_b) + C'_{33} \left(\frac{\partial u_x}{\partial y}(t, x_b) + \frac{\partial u_y}{\partial x}(t, x_b) \right) \right) \\
& + n_y \left(C'_{12} \frac{\partial u_x}{\partial x}(t, x_b) + C'_{22} \frac{\partial u_y}{\partial y}(t, x_b) + C'_{23} \left(\frac{\partial u_x}{\partial y}(t, x_b) + \frac{\partial u_y}{\partial x}(t, x_b) \right) \right) = 0,
\end{aligned} \tag{6.11}$$

are obtained. By differentiating Equation 6.11 twice with respect to time, and substituting the governing equations, a set of third-order boundary conditions can be obtained. These conditions are extremely verbose, even more so than Equation 6.10, and are not reproduced here. These higher-order conditions were however derived for use in the subsequent TTI examples, all of which use a fourth-order discretisation; their omission was found to produce serious instability.

6.3.1. Application to Simple Topography

Introducing a tilt parameter θ , set to 45° , a version of the sinusoid hill test case was developed for the TTI elastic case.

Particle-displacement fields are shown in Figure 6.7, whilst separated compressional and shear wavefields are found in Figure 6.8. The introduction of TTI causes the raw wavefields to appear much more confused as the left-right symmetry of the wavefield is broken by the tilted anisotropy axis. Note that the P wave first encounters the boundary slightly to the left of the apex of the hill. The interaction between anisotropy, topography, and the phase conversions introduced by the elastic free surface create a wavefield far more complex than the minimal setup might imply.

The separated wavefields help to clarify this particular example, particularly the S-wavefield, sharing similarities with the VTI case, albeit with the expected rotation of the direct and reflected shear wavefronts

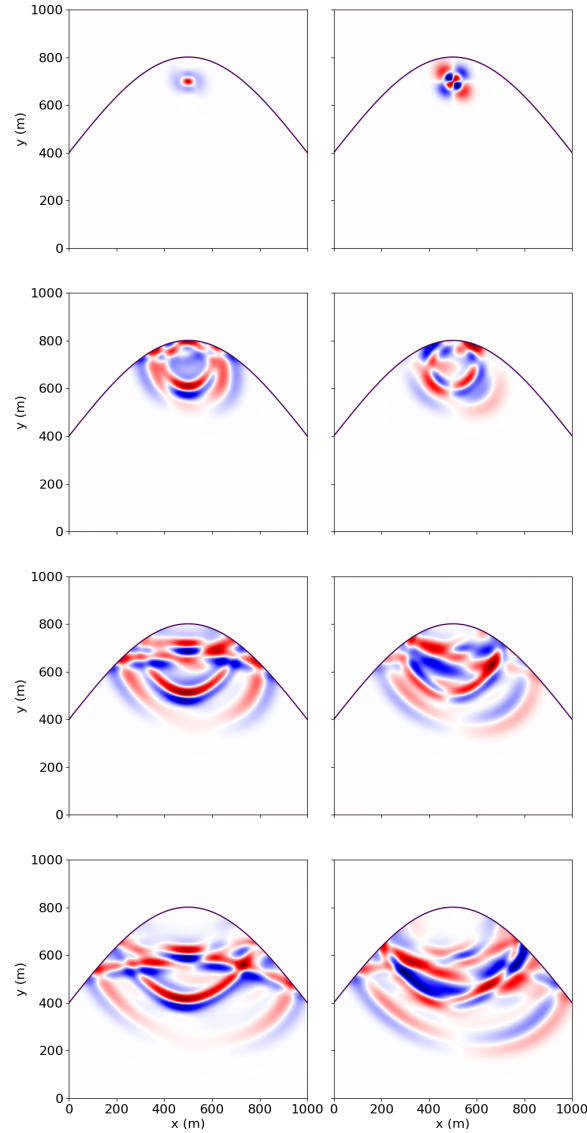


Figure 6.7.: Particle displacements u_x and u_y for a model featuring TTI are shown in the left and right columns respectively. From top to bottom, snapshots are at 137.5ms, 275ms, 412.5ms, and 550ms. All fields are normalised for clarity.

and skewing of the PS conversion due to the diagonal elongation of the P wavefronts. The combination of wavefront geometries and this particular topographic profile cause the P wavefield to be considerably more confused than in either the isotropic or VTI cases, as the reflected P wave and SP conversion overlap in a fairly complex manner. However, the wavefields share features broadly recognisable in Figures 6.2 and 6.5.

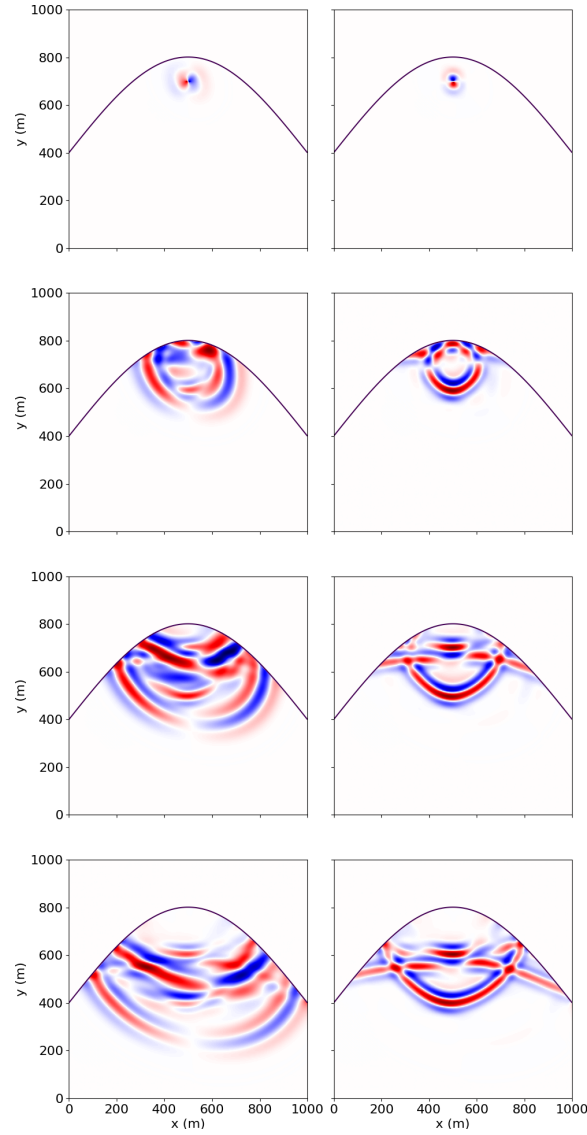


Figure 6.8.: Snapshots of $\nabla \cdot \mathbf{u}$ and the out-of-page component of $\nabla \times \mathbf{u}$ shown in Figure 6.7 at 137.5ms, 275ms, 412.5ms, and 550ms. The separated compressive (P) component is shown in the left column whilst the shear (S) component is shown on the right. All fields are normalised for clarity.

6.3.2. Application to Real-World Topography

Given the adequate performance of the immersed boundary treatment with the hill test case, application to real-world topography is the logical next step, representing the most complex physics to which the immersed boundary method proposed in Chapter 2 was applied. For com-

parison, parameterisation from the VTI example detailed in Section 6.2.2 was reused, with the addition of $\theta = 45 \text{ deg}$. The interaction between anisotropic wave propagation and the complex free-surface geometry is shown in Figure 6.9.

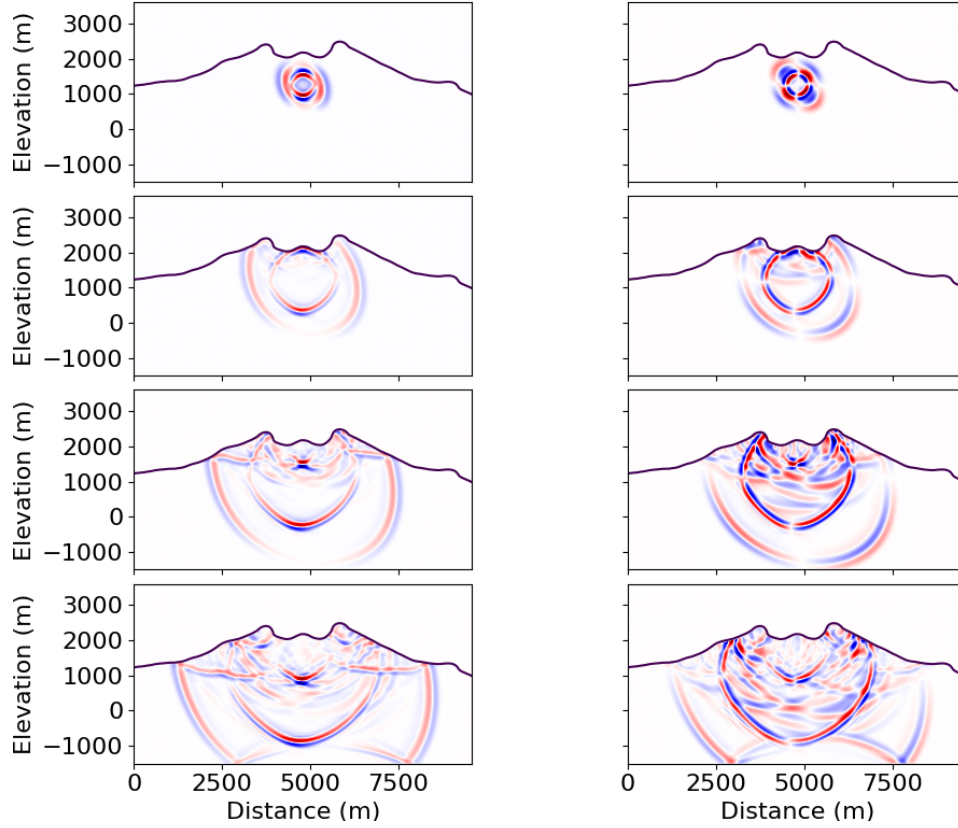


Figure 6.9.: Solving the 2nd-order TTI elastic wave equations in the presence of the 2D Mt St Helens topographic profile. The u_x and u_y wavefields are shown in the left and right columns respectively, showing snapshots at 375ms, 750ms, 1125ms, and 1500ms. All snapshots are normalised for clarity.

The immersed boundary treatment appeared to perform well for this test case - no noticeable artefacts were observed - and the L2 norm of the wavefields was not found to grow over the course of the simulation post-source-injection. As in the case of the simple topography example, whilst the fine-scale wavefield detail is considerably altered by the introduction of anisotropy, several large-scale features remain identifiable (for example the PS conversion and central cross where the reflected S wavefront is focused by the shape of the topography). However, they are noticeably distorted by the anisotropy.

That the immersed boundary treatment first detailed in Chapter 2 can be scaled in complexity from isotropic acoustic through to TTI elastic through several intermediate levels is a noteworthy result. The gulf between both the equations considered and the boundary conditions imposed is significant, especially in light of the consistency of the methodology applied.

6.4. Long-Time Stability

Whilst the examples shown are all stable for the duration of interest, it was found that a high-wavenumber instability was prone to forming along the boundary in longer model runs. In the prior examples, the time taken for this instability to arise was more than enough for the wavefront to reach the edges of the domain, in most cases emerging around two to four times this length of time. This took the form of alternating lobes, with wavelength of around two grid increments, oriented approximately perpendicular to the boundary surface. Small-scale instabilities are likely present in the results shown, although at the snapshots shown, their magnitude is much smaller than that of the wavefield.

Given the similarity of boundary conditions imposed and footprint of the boundary operators, it is of interest to see if the mesh-free elastic free surface implementation described by Takekawa and Mikada [2019] encounters similar issues; examples shown are not run to long time, leaving this somewhat ambiguous.

6.4.1. Characterising Instabilities

Whilst exhaustive characterisation of this unstable behaviour was prevented by time constraints, some aspects of the instability could be established.

The first of these was the apparent high-wavenumber content of the unstable artefact, consisting of an approximately grid-grid alternation of positive and negative values. A test case configuration found to rapidly produce this mode of instability was repeated at multiple resolutions and discretisation orders, whilst sweeping a range of material parameterisations. This investigation found the dominant wavenumber of the instabil-

ity to be broadly dependent upon grid spacing, whilst being insensitive to both material parameters and spatial discretisation.

High-wavenumber information incident on the boundary was found to excite more pronounced instability: higher-frequency sources (injecting white noise as an extreme case) resulted in noticeable instability earlier in the simulation than low-frequency source signatures. The amplitude of this excitation had no effect other than to linearly scale the amplitude of the instability; the pattern of instability, its amplitude relative to that of the wavefield, and its polarity remained unchanged.

The other main control on the emergence of instability was found to be the ratio of P to S wave velocities: instability was more severe when this ratio was high. Experiments found that the degree of instability was comparable over a range of P-wave velocities, given a consistent ratio between P and S-wave velocities. The presence of anisotropy was also found to somewhat increase the degree of instability observed, particularly in the TTI case, although there was insufficient time to investigate this interaction more thoroughly, and its cause remains poorly understood.

Boundary curvature also appears to be a control. Specific points on the sinusoidal hill boundary geometry were found to be consistent hotspots for instability across the configurations tested, with the amplitude of the instability generally being greatest here at long time. However, it is not immediately clear as to which aspects of the curvature control the instability. Tests with the sinusoid hill surface found that instability was most severe both at the apex of the hill where curvature is at its tightest, and partway down the flanks where curvature is minimal. As such, it is unlikely that tightness of curvature is the main control.

The hotspots were found to be associated with points where the condition number of the matrix A is anomalously large, in some cases almost two orders of magnitude greater than the median. A high condition number in this context corresponds to an oversensitivity of the extrapolant to one or more of the datapoints with which it is fitted. Such ill-conditioned extrapolants amplify small errors, leading to instability in the boundary treatment [Seo and Mittal, 2011]. To investigate this, a condition-number-thresholding approach was tested to determine the suitability of a matrix for inversion, rather than a simple check for rank. Mandating a condition number below a threshold of order 100 was found to stabilise several

2nd-order elastic test cases which previously exhibited moderate to severe instability, even for runs in excess of 10 times the original simulation length.

However, it was found that setting a suitable threshold proved challenging, and in some cases, simply expanding the support region of the extrapolant was insufficient to reduce the condition number of the resultant matrix. This resulted in a tendency for the stencil footprint to expand apparently indefinitely without obtaining a sufficiently well-conditioned matrix. It may be that the introduction of other constraints to extrapolant may serve to better condition these matrices. As to what these constraints may be is uncertain however; future research in this area may be fruitful. However, it is worth noting that whilst instability was associated with large condition numbers, large condition numbers did not always lead to instability.

Whilst a rigorous analysis would be desirable, this was prevented by time constraints, particularly given that the wide range of potential surface geometries makes comprehensive investigation unfeasible. Furthermore, such analyses are recognised to be challenging: Mulder and Huiskes [2017] were unable to find a closed form for the stability of their comparatively simple immersed boundary scheme, even for examples featuring small grids and low-order spatial discretisations. Rigorous investigation is often omitted in practice, even in fields such as CFD or aeroacoustics where immersed boundary methods have greater prevalence and maturity [Seo and Mittal, 2011, Dhamankar et al., 2015].

6.4.2. Proposed Rectification

Brehm and Fasel [2013], Brehm et al. [2015] propose variants on a scheme whereby a local stability constraint is used in the design of the operators. Whilst this approach was found to be effective in ensuring stability, the process of including these stability constraints in the basis-fitting process is mathematically complex. Furthermore, constructing suitable abstractions to enable information regarding the governing system of equations to be used in devising the immersed boundary treatment without violating the separation of concerns between the boundary treatment and overarching equations being solved is a non-trivial problem.

An approach more commonly found in the CFD and aeroacoustics literature is wavenumber filtering [Kennedy and Carpenter, 1994, Gaitonde et al., 1999, Seo and Mittal, 2011, Dhamankar et al., 2015]. It is recognised that high-wavenumber nonphysical components can be introduced by boundary conditions [Gaitonde et al., 1999]. These high-wavenumber artefacts have been identified as a potential instability source [Purser, 1987]. Wavenumber filters are commonly applied to suppress these components, with filters of incrementally smaller span used as one approaches the boundary to prevent filter stencils using points outside the computational domain [Gaitonde and Visbal, 2000, Seo and Mittal, 2011, Dhamankar et al., 2015]. The footprint of the filtering operator can alternatively be biased to the left or right relative to the point at which it is applied [Gaitonde and Visbal, 1998]. However, the spectral functions of asymmetric operators can have non-ideal properties: a real component greater than unity at wavenumbers just short of the roll-off for example [Gaitonde and Visbal, 2000].

To evaluate the effectiveness of wavenumber filtering in suppressing instability introduced by the immersed elastic free surface, the test case from Section 6.1.1 was parameterised such that the onset of instability occurred early in the simulation. S-wave velocity was reduced to 0.4ms^{-1} to induce instability in the unfiltered model, and the run time was increased to 800ms. To improve the separation between the various reflections and conversions for this case, the source position was moved upwards by 50m. The source frequency was also reduced to 6Hz to ensure that the filter did not encroach on wavenumbers of interest in the main body of the domain, particularly in light of the low S-wave velocity. Snapshots of an isotropic elastic model in this configuration are shown in Figure 6.10. In the later snapshots, strong instabilities are apparent along the boundary, on the flanks of the hill. The normalisation causes these to be the only visible parts of the wavefield in later snapshots.

Stencils for explicit wavenumber filters were taken from Purser [1987]. Approaching the boundary, filter order was reduced incrementally to minimise intersection with the surface. A wavenumber filter pass was applied for each dimension at the end of every timestep to attenuate the spurious frequencies associated with the instability. More performant explicit filters than those used here are likely available. In an ideal case, explicit

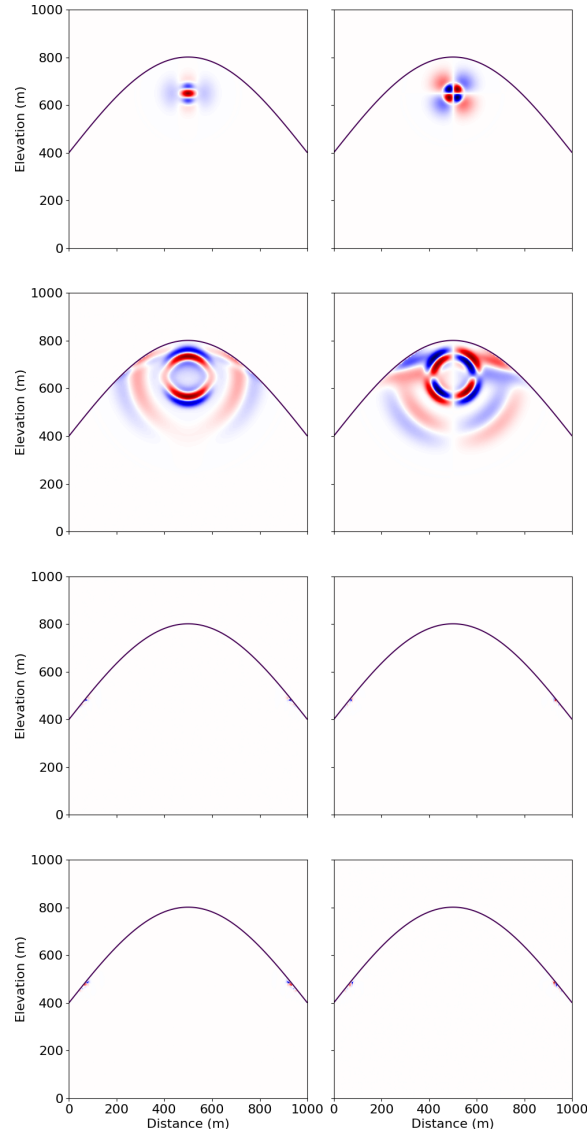


Figure 6.10.: Particle displacement snapshots for an isotropic elastic configuration which exhibits instability over short time scales. Wavefields are as in Figure 6.1, and are normalised against maximum absolute values. Snapshots are shown at 200ms, 400ms, 600ms, and 800ms from top to bottom.

wavenumber filters with a boundary-aware footprint would be designed for optimal performance. However, time constraints and the degree of engineering required to obtain a satisfactorily elegant implementation ruled this possibility out.

An eighth-order filter was applied in the interior of the domain, this being reduced incrementally to second-order for boundary-adjacent points.

To minimise nonphysical behaviour due to attenuation of higher frequencies of interest, for any given order, the filter with the steepest and highest roll-off was selected from Purser [1987]. Whilst these filters were of higher order than the discretisation within the interior, it was found that lower-order filters required adjacent to the boundary attenuated higher-wavenumber components of the propagating wave resulting in artefacts and unrealistic propagation behaviour.

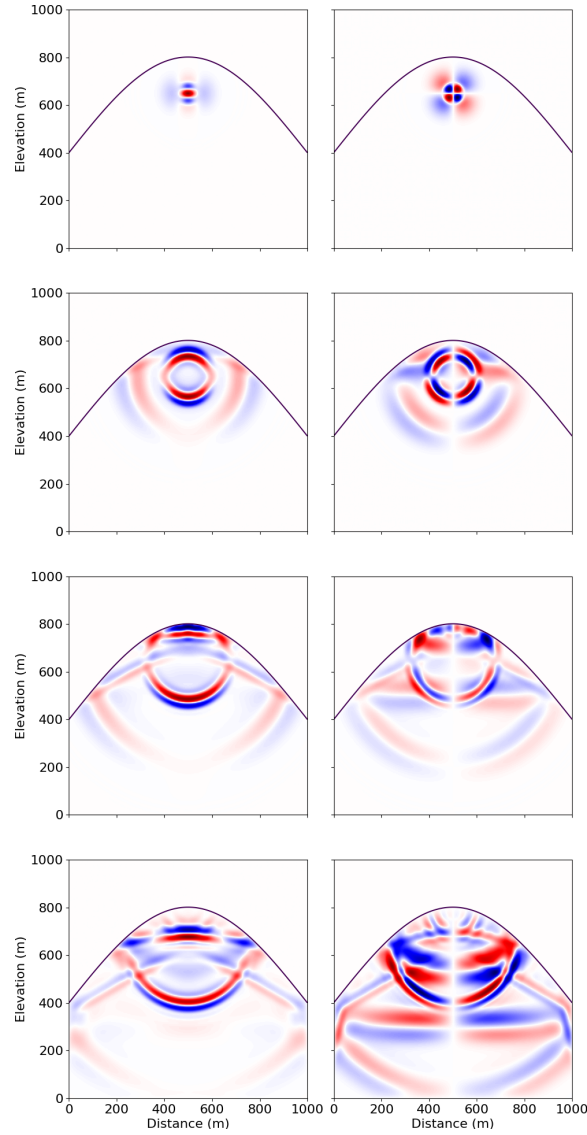


Figure 6.11.: As Figure 6.10, albeit with wavenumber filters from Purser [1987] applied after each timestep.

Results of this particular test case are shown in Figure 6.11. The wavenumber-filtering approach was found to successfully stabilise the boundary treatment; the sum of the L2 norms of the wavefields was found not to grow post-source-injection and it was possible to run the model over a time range tenfold that shown in Figure 6.1) without observable instability. However, whilst reflections and conversions were generated with expected geometries and approximate amplitudes, their polarity was reversed by the filtering routine. Comparison of wavefields from unfiltered and filtered simulations for a stable case found the L2 norms in the filtered and unfiltered cases to be broadly consistent with one another in spite of the polarity reversal. It is not immediately apparent why this polarity reversal should occur, although it was possible to reproduce similar behaviour by applying the wavenumber filtering scheme to the pressure field in a 2nd-order acoustic example featuring a rigid surface. Again the polarity of the reflected wave was inverted, although this was not the case when an acoustic free surface was imposed. For this acoustic case, imposing filters up to the boundary surface was found to induce this behaviour, but avoiding applying the filtering routine at boundary-adjacent interior points was found to prevent this. However, if this convention was followed in the elastic case, the desired stabilisation was not obtained. Furthermore, similar behaviour could be obtained by filtering the particle velocity fields in the first-order elastic case (discussed more thoroughly in Section 6.5). The boundary conditions in all these cases concern fluxes. It is postulated that the wavenumber filters as applied here adversely interact with the imposition of this flux, although further investigation is necessary on this front to understand the mechanism by which this occurs.

Beyond this issue, additional kinematic inaccuracies were apparent. Some artefacts are observable trailing the S wave reflection, particularly at the apex of the hill where surface curvature is large. There are also minor amplitude inconsistencies (a slight rippling along the length of the wavefront) in this reflection, although these may be a result of these artefacts. The other notable kinematic effect of the filtering procedure is the dilation and drag on waves in the vicinity of the boundary, becoming somewhat smeared (most obviously trailing the PS conversion). It is not entirely clear as to whether this behaviour is inherent in the filtering, the particular filter chosen, or its underlying implementation: this would benefit from future

investigation.

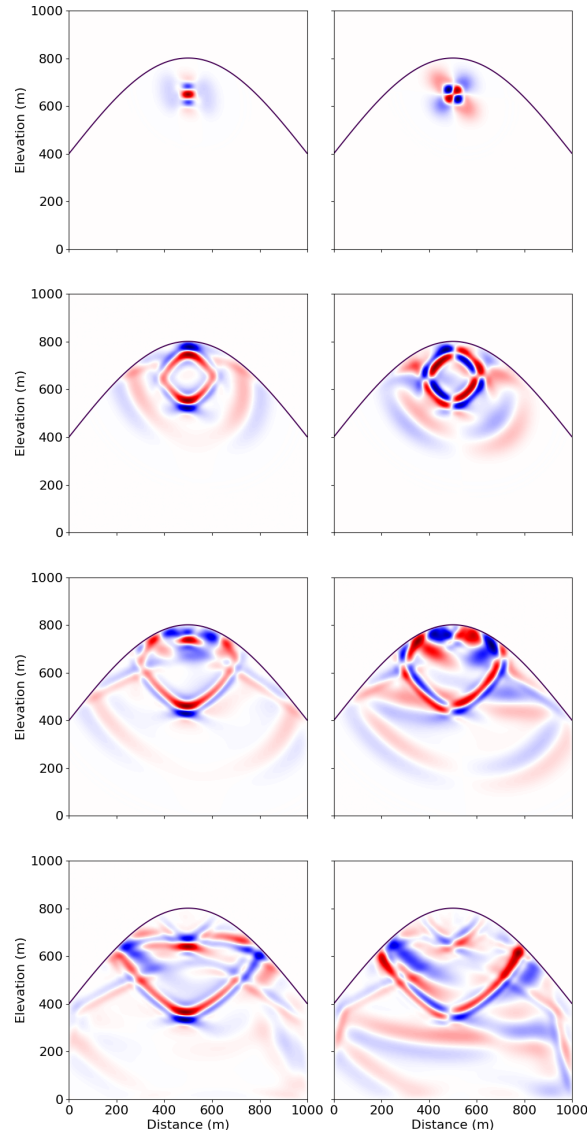


Figure 6.12.: The filtering routine demonstrated in Figure 6.11 applied to the TTI elastic immersed free surface. Anisotropy parameters are set to $\delta = 0.1$ and $\epsilon = 0.25$, whilst the tilt of the isotropy axis is 45 deg. Snapshots are shown at 200ms, 400ms, 600ms, and 800ms from top to bottom.

The test was repeated for a TTI elastic free surface with similar results, shown in Figure 6.12. Note that the artefact trailing the reflected S wave below the apex of the surface is also present in this case, appearing slightly distorted versus the isotropic case. This is most likely a result of the interaction between the boundary-adjacent filter and the short wavelength of

the incident and reflected wave, both of which are attenuated, producing kinetic inaccuracies.

Overall, wavenumber-filtering shows promise for ensuring long-time stability in second-order elastic modelling contexts. However, refinement of the approach is needed. The reliance on preexisting filters with a symmetrical footprint proved a limitation, as filter order had to be reduced towards the boundary to avoid using points located outside the computational domain. Lower-order filters of the type detailed by Purser [1987] have a roll-off that is not only more gradual but which occurs at a lower wavenumber. As such, the filters applied adjacent to the boundary attenuate some high-wavenumber information within the range of interest, hence the observed smearing. This is postulated to be the source of some of the other kinematic inaccuracies observed. This also required a lower frequency than would otherwise be used, as the maximum wavenumber specified by the filter cutoff was lower than that dictated by the resolution of the FD scheme, especially in the vicinity of the boundary.

Ideally, these wavenumber filters would be designed in tandem with the immersed boundary operators, or perhaps even integrated into them. By designing suitable filters with a boundary-aware footprint, it should hopefully be possible to retain stability benefits whilst reducing the undesirable behaviours observed in these examples. Investigation into the area over which filtering is required may also be beneficial in limiting computational overhead.

6.5. 1st-Order Elastic

The elastic wave equations can alternatively be formulated as a coupled system of stress and particle velocity, as introduced by Virieux [1986]. The second-order formulation must be used with care where rapid variation in material properties are present; strictly speaking, nested derivatives should be used in the heterogeneous formulation [Kelly et al., 1976]. Whilst this can be addressed via explicit treatment of interfaces via the imposition of boundary conditions, this is impractical for all but the most basic of subsurface models [Kelly et al., 1976]. This need can be avoided however if gradients of material parameters are sufficiently gentle. By transforming the system of equations for the heterogeneous case as de-

scribed by Virieux [1986], strong material parameter variations can be handled naturally without the risk of numerical instability. The governing coupled system of equations is given by Equations 1.20 and 1.19.

Similar to the first-order acoustic wave equation discussed in Chapter 4, Equations 1.20 and 1.19 are commonly discretised using a staggered grid in which normal stresses are located on the reference node, whilst particle velocities are staggered by half a grid increment in their respective dimensions [Virieux, 1986]. Shear stresses are offset by half a grid increment in each of their respective dimensions. Note that whilst other staggering schemes such as rotated-staggered have been devised to achieve similar ends [Yang et al., 2015, Gao and Huang, 2017], this discretisation remains the prevalent approach.

6.6. Free-Surface Boundary Conditions

The zero-traction condition given in Equation 6.1 can be naturally accommodated with these equations, as stress is explicitly featured. In combination with Equations 1.20 and 1.19, additional particle velocity and stress conditions of incrementally higher order can be derived. First-order particle velocity conditions are obtained by taking first derivatives of Equation 6.1 with respect to time and substituting stress derivatives as given by Equation 1.20. This yields

$$\begin{aligned} n_x(\lambda + 2\mu) \frac{\partial v_x}{\partial x}(t, \mathbf{x}_b) + n_x \lambda \frac{\partial v_y}{\partial y}(t, \mathbf{x}_b) + n_y \mu \left(\frac{\partial v_x}{\partial y}(t, \mathbf{x}_b) + \frac{\partial v_y}{\partial x}(t, \mathbf{x}_b) \right) &= 0 \\ n_x \mu \left(\frac{\partial v_x}{\partial y}(t, \mathbf{x}_b) + \frac{\partial v_y}{\partial x}(t, \mathbf{x}_b) \right) + n_y(\lambda + 2\mu) \frac{\partial v_y}{\partial y}(t, \mathbf{x}_b) + n_y \lambda \frac{\partial v_x}{\partial x}(t, \mathbf{x}_b) &= 0. \end{aligned} \quad (6.12)$$

This is the first derivative of Equation 6.6 with respect to time, reflecting the relationship between particle velocity and displacement.

Deriving second-order stress boundary conditions by taking derivatives with respect to time of the above conditions and substituting velocity

derivatives from Equation 1.19, the constraints

$$\begin{aligned}
& n_x \left((\lambda + 2\mu) \frac{\partial^2 \tau_{xx}}{\partial x^2}(t, \mathbf{x}_b) + (2\lambda + 2\mu) \frac{\partial^2 \tau_{xy}}{\partial x \partial y}(t, \mathbf{x}_b) + \lambda \frac{\partial^2 \tau_{yy}}{\partial y^2}(t, \mathbf{x}_b) \right) \\
& + n_y \mu \left(\frac{\partial^2 \tau_{xx}}{\partial x \partial y}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_{xy}}{\partial x^2}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_{xy}}{\partial y^2}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_{yy}}{\partial x \partial y}(t, \mathbf{x}_b) \right) = 0 \\
& n_x \mu \left(\frac{\partial^2 \tau_{xx}}{\partial x \partial y}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_{xy}}{\partial x^2}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_{xy}}{\partial y^2}(t, \mathbf{x}_b) + \frac{\partial^2 \tau_{yy}}{\partial x \partial y}(t, \mathbf{x}_b) \right) \\
& + n_y \left(\lambda \frac{\partial^2 \tau_{xx}}{\partial x^2}(t, \mathbf{x}_b) + (2\lambda + 2\mu) \frac{\partial^2 \tau_{xy}}{\partial x \partial y}(t, \mathbf{x}_b) + (\lambda + 2\mu) \frac{\partial^2 \tau_{yy}}{\partial y^2}(t, \mathbf{x}_b) \right) = 0
\end{aligned} \tag{6.13}$$

are formed, assuming locally constant material parameters. This process can be repeated to obtain third-order particle velocity and fourth-order stress boundary conditions.

6.7. Examples

To investigate the application of the immersed boundary method to the equation and discretisation of Virieux [1986], a set of simple test cases was devised. For the following test cases, a 1km by 1km model domain with grid spacing of 10m was used. P-wave velocity was set to 1kms^{-1} , S-wave velocity was set to 0.65kms^{-1} , and density was set to 1gcm^{-3} . An 8Hz Ricker wavelet point source was injected, and the model run for 550ms. Other than the immersed free surface, all other domain edges featured zero Dirichlet boundary conditions.

The first of these test cases featured a horizontal free surface. This trivial configuration was found to be unstable by Lombard et al. [2008] if the surface lay between the reference stress node (at which normal stresses are discretised) and the corresponding vertical particle velocity node. Whilst tests using both second and fourth-order discretisations appeared stable over relatively short timespans, the latter shown in Figure 6.13, propagation over longer time periods highlighted the emergence of an oscillatory instability emanating from the boundary when the surface was positioned between the reference stress node and its corresponding vertical particle velocity node. The emergence of instability was found to be more rapid

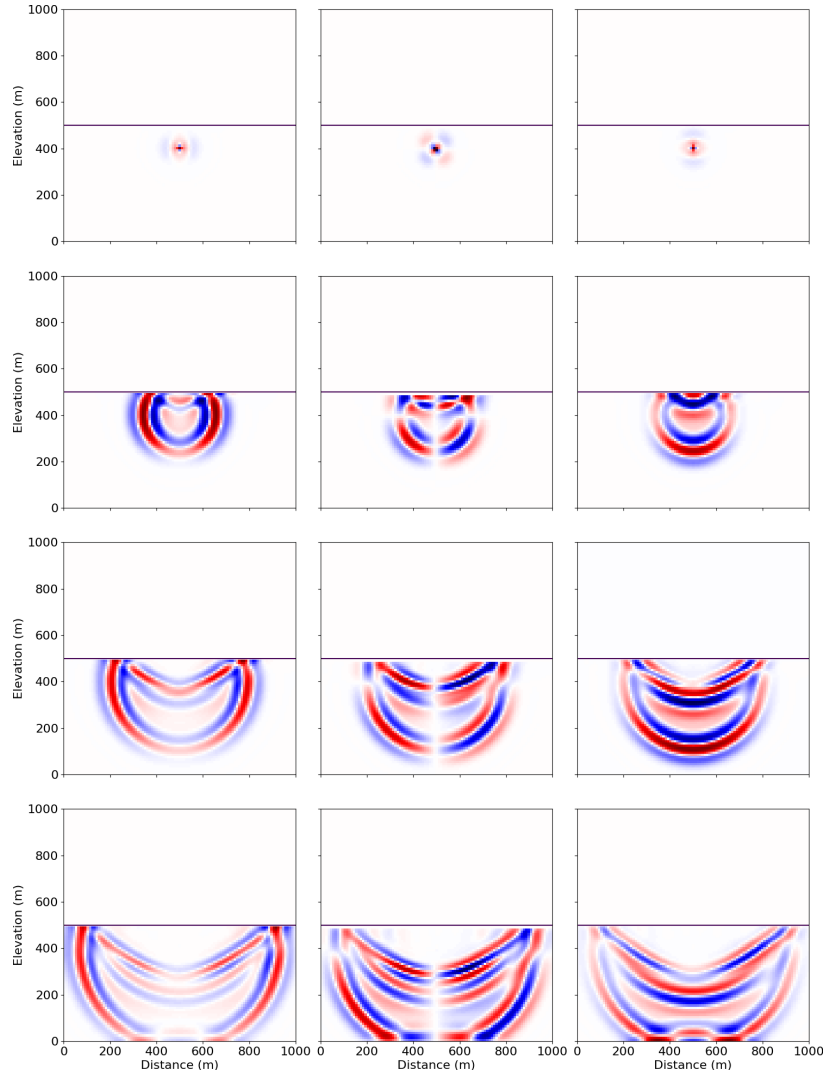


Figure 6.13: Snapshots of the normalised stress fields for a horizontal-surface test case. From left to right, the fields are τ_{xx} , τ_{xy} , and τ_{yy} . Snapshots from top to bottom are at 137.5ms, 275ms, 412.5ms, and 550ms. This particular case was found to be stable. Note the PS conversion as a result of the boundary conditions imposed.

with the wave approaching the boundary from above (the vertical particle velocity nodes being staggered upwards relative to the reference grid). This instability was observed even when absorbing boundary conditions were imposed along the edges of the domain, implying that this instability does not stem from waves reflecting back into the domain. An experiment shifting a boundary-source-receiver assembly relative to the grid found

that the phase and amplitudes of arrivals in the receiver data corresponding with waves interacting with the boundary were consistent regardless of the position of the boundary relative to the grid prior to the onset of instability. This was found to be the case on both sides of the boundary.

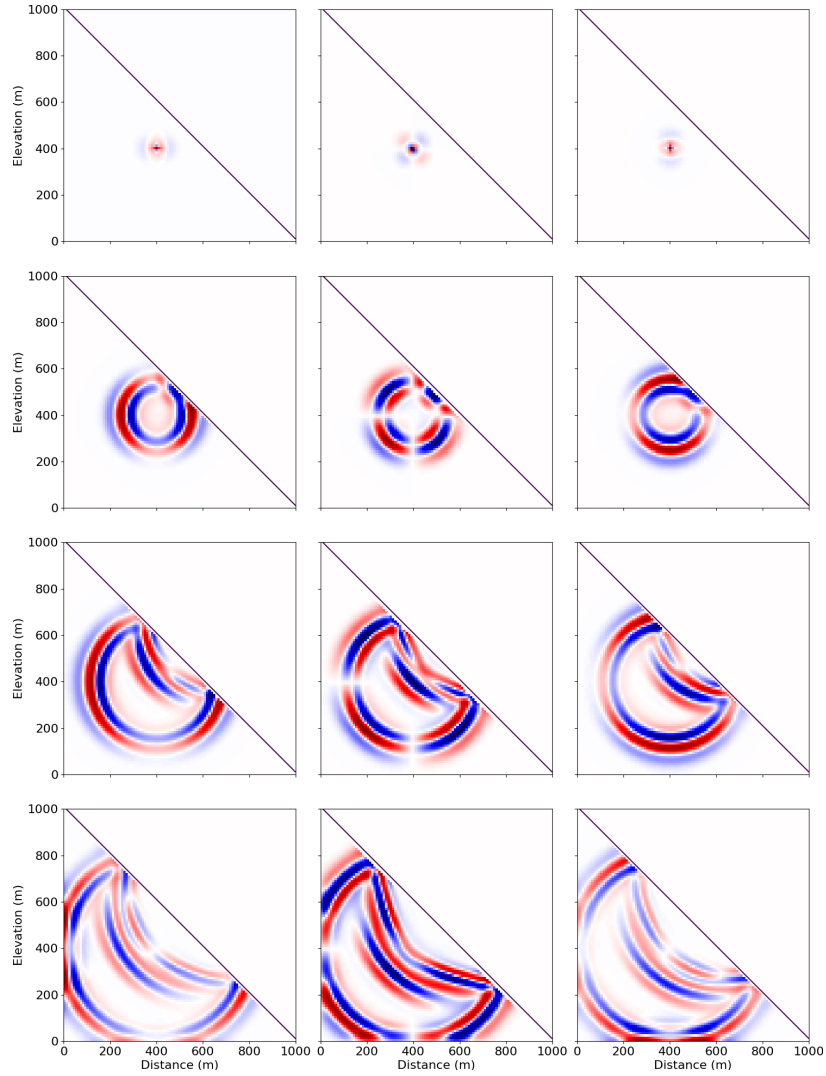


Figure 6.14: Snapshots of the normalised stress fields for a diagonal-surface test case. From left to right, the fields are τ_{xx} , τ_{xy} , and τ_{yy} . Snapshots from top to bottom are at 137.5ms, 275ms, 412.5ms, and 550ms. This particular case was found to be stable. As in the horizontal cases, the expected phase conversions are generated. Note also the symmetry of the τ_{xy} field and that the normal stress fields are mirror images of one another on the diagonal.

The second test case featured a surface tilted by 45° , such that the position of the boundary relative to the grid was the same in both x and y

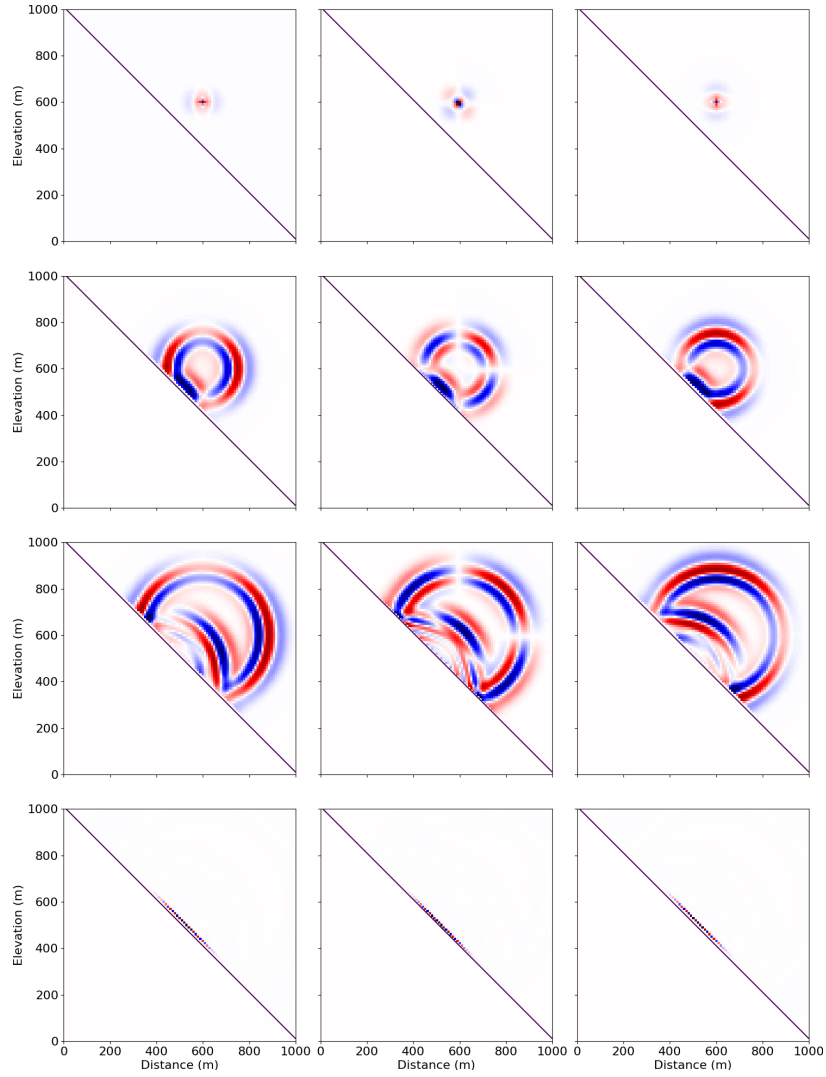


Figure 6.15.: As Figure 6.14, albeit with sectioning reversed such that the wave incident upon the boundary is travelling in the opposite direction to the staggering. The source position has been mirrored across the boundary, which remains in the same position (although with the opposite orientation). The final snapshots appear mostly blank due to normalisation of the fields in the presence of an exponential blowup along the boundary.

directions. Instability rapidly emerged with a fourth-order discretisation, whilst the second-order discretisation was found to suffer from instability at longer time. Wavefield snapshots for a second-order discretisation over a shorter interval are shown in Figure 6.14, demonstrating expected reflection geometry and kinematics early in the simulation. The onset of instability was found to be earlier when the wavefront approached the sur-

face from the positive direction (i.e. the direction in which the shear stress nodes are staggered). This can be seen in Figure 6.15 in which blowup has occurred by the 550ms mark, despite accurate behaviour in previous snapshots. Examining the τ_{xy} field in the 412.5ms snapshot, some faint artefacts can be discerned at the centre of the boundary where the wave was first incident. Instability was found to occur regardless of the position of the surface relative to the grid, although as in the horizontal case, the phase and amplitude of waves interacting with the boundary were very consistent across these scenarios prior to the onset of instability. Similar behaviour was also exhibited when the boundary was mirrored to run along the opposite diagonal.

6.7.1. Instability with Staggered discretisation

Beyond a handful of trivial cases, the immersed boundary treatment was unstable in combination with the approach of Virieux [1986]. Even a flat surface tilted by 5° from the horizontal resulted in blowup. Instability persisted with shear modulus set to zero, nominally equivalent to the 1st-order acoustic case (note that the linear system formed has a different structure, however). This was observed for all discretisation orders tested.

There is precedent for this behaviour: Gao et al. [2015] found that their immersed boundary approach was unstable without the addition of artificial diffusion, and was impossible to stabilise beyond second-order accuracy. Furthermore, Lombard et al. [2008] found that applying their approach to the staggered discretisation of Virieux [1986] was unstable in one of the trivial horizontal cases demonstrated to be unstable with the proposed scheme.

Altering the cutoff η at which points are included or excluded, as in the acoustic case, was investigated as a means of stabilising the boundary treatment. No parameterisation which provided notable suppression of the instability was found. Increasing the support region radius, either globally or selectively, to reduce the condition number of the extrapolation matrices was not found to prevent this instability, although it did produce a damping effect. Furthermore, as in the second-order case, this was insufficient to improve the condition number in some cases. Similarly, the wavenumber filtering routine explored in the second-order elastic cases

was insufficient to yield numerical stability.

Both Lombard et al. [2008] and Almuhaideb and Toksoz [2015] successfully applied their immersed boundary approaches to schemes from aeroacoustics which eschew staggering. In this vein, a similar approach was explored for this work. One approach used in CFD applications is the application of unstaggered centred differences to approximate first derivatives, suppressing the resulting even-odd decoupling, or checkerboard error, via wavenumber filtering [Kennedy and Carpenter, 1994, Gaitonde and Visbal, 1998]. This approach has been used for wave propagation problems including investigations of underwater acoustic scattering featuring immersed boundaries to represent complex geometries [Zhao et al., 2022]. As such, an unstaggered discretisation was developed based on centered derivatives, whilst reusing the filtering approach detailed for the 2nd-order elastic case to suppress checkerboard error.

6.7.2. Application of Unstaggered Scheme

To assess this approach, the sinusoid hill geometry was revisited. With a staggered discretisation, blowup occurs shortly after contact between the incident wavefront and the boundary. It was found that filtering applied to the velocity field heavily damped reflections and conversions. However, filtering only stress fields at the end of each timestep was found to be sufficient to prevent checkerboard error.

Free-surface boundary conditions were imposed on the surface, whilst zero Dirichlet boundary conditions were imposed on all other edges of each field. A constant P-wave velocity of 1kms^{-1} , S-wave velocity of 0.65kms^{-1} , and density of 1gcm^{-3} were used. A 12Hz Ricker source was injected at an elevation of 600m, centred in the x axis, and the model run for 650ms.

The result is shown in Figure 6.16. The numerical scheme appears stable (the sum of L2 norms of the stress fields over time implies gradual dispersion of energy), and the PP reflection and PS conversion at the surface are generated with the expected geometries and amplitudes. There are minor artefacts, including those surrounding the source in earlier snapshots. These mainly take the form of ripples trailing the reflected wavefronts, these being strongest in the τ_{xx} and τ_{xy} fields at the apex of the hill where surface curvature is strongest. These nonphysical ripples diminish over

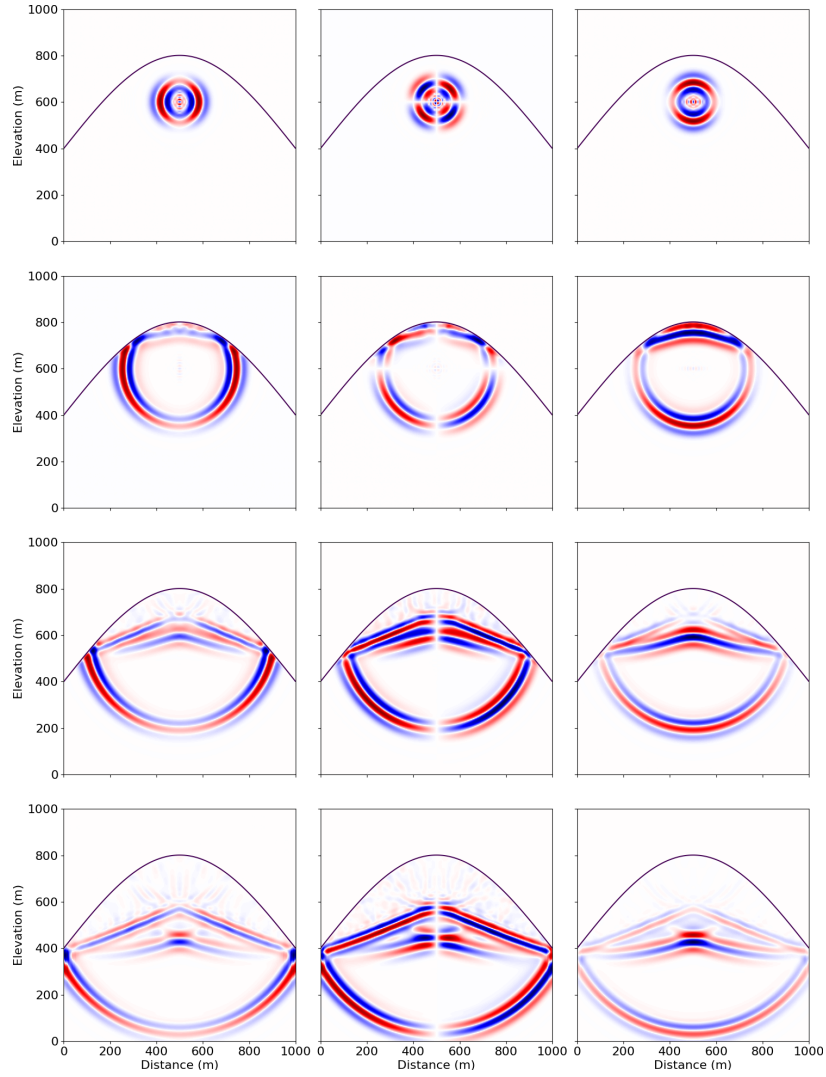


Figure 6.16: Snapshots of the normalised stress fields for the test case featuring an immersed free surface alongside an unstaggered discretisation with wavenumber filtering. From left to right, the fields are τ_{xx} , τ_{xy} , and τ_{yy} . Snapshots from top to bottom are at 162.5ms, 325ms, 487.5ms, and 650ms.

time, implying that they are not instabilities. These are not dissimilar to artefacts observed in the second-order case when filtering is applied, implying that they are likely a product of deficiencies in the filtering approach chosen. Again, due to shortcomings of the filtering scheme, it is difficult to separate potential adverse effects of the boundary treatment from those introduced by filtering. Improved filter design would enable the immersed boundary to be more comprehensively assessed in this case.

Whilst application to real-world terrain as demonstrated for previous examples would be feasible with the approach used here, refinement of the methodology is needed for this to be a useful exercise. However, the method appears broadly sound. Improved tuning of the filters to better match the resolution of the interior discretisation would achieve more accurate wave propagation throughout the domain. In the vicinity of the free surface, the design of filters with boundary-aware footprints would prevent the need to reduce their order and prevent accidental truncation of filter footprints, thereby improving the accuracy of the method and suppressing undesired kinematic effects.

Beyond being amenable to the immersed boundary treatment, the proposed underlying numerical scheme would also simplify implementation of TTI elastic, removing the need for rotated-staggered grids [Yang et al., 2015, Gao and Huang, 2017] which add complexity, or interpolation routines in the conventionally-staggered case [Gao and Huang, 2017] which add computational expense. Further investigation in this area may be beneficial.

6.8. Conclusions

The immersed boundary treatment outlined in Chapter 2 was found to be sufficiently general and flexible to accommodate free surface boundary conditions associated with 2nd-order elastic wave equations including those featuring both VTI and TTI. The resultant numerical scheme was found to convincingly handle analytic and real-world surface geometries for all variants explored. It is clear from these experiments that the general method developed is widely applicable: likely more so than has been explored in this work. However, issues with stability during longer model runs were identified. Whilst generally occurring fairly late in the simulation, particular parameterisations were found to exacerbate this phenomenon, most notably large ratios of P to S wave velocity and anisotropy. For particular practical applications where propagation over long time periods is paramount (FWI and RTM for example), this will need addressing. An approach of improving extrapolant conditioning, whilst effective in some cases, proved challenging to carry out reliably; wavenumber filtering after each timestep was proposed as an alternative

option. This strategy was found to be effective in ensuring stability, even over very long runs, although the filtering routine trialed was found to introduce undesirable kinematic effects. The most notable of these was an interaction with flux-type boundary conditions resulting in a polarity flip in reflected and converted wavefronts, the source of which remains undetermined.

Boundary conditions necessitated by the 1st-order formulation of the isotropic elastic wave equation could similarly be imposed via the method developed in Chapter 2. However, instabilities were encountered when combining the immersed boundary treatment described with a staggered discretisation. Even trivial geometries were found to exhibit this behaviour, becoming more acute at higher orders. However, the phase and amplitude of waves interacting with the boundary was found to be consistent regardless of the position of the boundary relative to the grid, before the onset of instability. An alternative numerical scheme using a single grid was investigated for use with the immersed free surface. Explicit wavenumber filters were applied to the stress fields at the end of each timestep to suppress the checkerboard error introduced by repeated application of central differences. This scheme was found to be stable in combination with the sinusoid hill benchmark surface. Whilst minor artefacts were observed, these could not be conclusively attributed to the immersed boundary treatment due to shortcomings in the filtering scheme.

6.9. Future Work

Whilst increasing the support region of the stencil can be used as a means of reducing the condition number of extrapolation matrices, it is unreliable and translates to a more expensive FD kernel. An improved strategy to construct better-conditioned linear systems would likely aid stabilisation of the method given the association between large condition numbers and instability hotspots. Additionally, an improvement of the wavenumber filtering approach trialed with both first and second-order elastic wave equations to address undesirable kinematic effects in the reflected wavefronts is warranted. Reliance on preexisting symmetrical filter stencils resulted in the need to reduce the filter order at boundary-adjacent points to minimise intersection with the boundary. A methodology to design

suitable wavenumber filters with boundary-aware footprints would likely go a long way to addressing these issues. It is anticipated that practical applications would likely want to employ some combination of these two approaches to ensure stability across a wide range of scenarios.

Given the issues in applying the immersed boundary treatment to staggered discretisations, it would be of interest to investigate its interaction with similar discretisations such as the rotated-staggered schemes used in the TTI case. It would also be worth investigating the numerical discretisation developed to anisotropic formulations of the 1st-order elastic wave equation. Since filtering is used to prevent checkerboard error, the complex staggering schemes or interpolation routines commonly used are sidestepped, potentially enabling straightforward topography inclusion.

The free surface is not the only interface of interest in the elastic case, the acoustic-elastic transition at the seafloor has also been the subject of much investigation. Misrepresentation of bathymetry due to the discretisation can produce similar staircasing error to that from topography [Vieira et al., 2018]. Multiple studies, generally using coordinate-transform approaches, have aimed to address this by representing the interface with the correct condition and location [Sun et al., 2017, Qu et al., 2020]. Data from ocean-bottom seismometers requires accurate representation of this transition, given the proximity of the receiver and interface. Furthermore, by coupling acoustic and elastic solvers at the interface, computational cost is greatly reduced, as the isotropic acoustic wave equation solved in the water column has a considerably smaller computational footprint. Extending the mathematical framework to accommodate this coupling at the seafloor would potentially greatly simplify its inclusion in models.

Finally, during the work detailed in this Chapter, an inconsistency and lack of understanding of free surface conditions and implementations within the literature became apparent. For example, Lan and Zhang [2011] compares six distinct elastic free surface implementations, concluding that many of them are unstable, produce inaccurate results, or limit the order of convergence. The zero traction condition originates from the displacement of the free surface until surface-normal stresses are zero. However, this movement of the boundary introduces nonlinearity and is rarely considered. It is anticipated that investigation into such conditions and immersed boundaries implementing them could benefit accurate simulation

of free surface behaviour.

Chapter 7

Implementation of the Mathematical Framework

This Chapter details the software implementation of Schism, a software framework based upon the mathematical method outlined in Chapter 2, used to generate the boundary treatments detailed in Chapters 3 through 6. The core design principles of this software were abstraction and generality, leveraging symbolic computing to automate the derivation of immersed boundary schemes. This framework was designed with sufficient flexibility to automatically generate the requisite boundary treatments demonstrated throughout this work from a concise, consistent specification. To the author's knowledge, no comparable open-source frameworks address immersed boundary implementation in this manner; existing immersed boundary frameworks are typically narrow in scope, and are consequently difficult to port between problems.

A cornerstone of the work, Schism underpins the broad suite of applications presented. Whilst the implementation of any non-trivial boundary treatment will feature some degree of automation, automated generation of the numerical treatment enables a level of flexibility unprecedented in published literature. A simplified overview of its operation is given in Figure 7.1, showing both the specification provided by the user and the various steps involved in devising the immersed boundary scheme and calculating the modified FD operators.

The advantages of the depth of automation were multifaceted. The rapidity with which boundary treatments could be specified, tested, and

adjusted as necessary was greatly enhanced, as all aspects of implementation below the topmost specification were handled algorithmically. This enabled extension of the method across a wide range of physics including equations and boundary conditions for which immersed boundary implementations have not previously been published. Furthermore, consistency between these treatments was ensured, low-level implementation being abstracted from the user and carried out according to a uniform algorithm.

Abstraction and automation were crucial in managing complexity introduced by the boundary treatment, particularly for examples featuring large, multi-field stencils (Chapter 4), complex or verbose boundary conditions (Chapter 5), or combinations thereof (Chapter 6). This complexity is abstracted from the user, the numerical scheme automatically devised by Schism, whilst Devito generates the underlying kernel, yielding concise top-level specification and separation of concerns. The high-level specification enabled maximum code reuse whilst minimising duplicated effort, proving beneficial in investigating the capabilities of the underlying mathematical method as a wide range of problems and parameterisations could be tested with only minimal reworking of the top-level code.

7.1. API Approach and Considerations

7.1.1. Intent of the Software Framework

The generality of the mathematical method detailed in Chapter 2 enables immersed boundary treatments to be devised in a consistent manner for a wide range of boundary conditions and governing equations. The information required to characterise each of these boundary treatments is similarly consistent, consisting of boundary geometry, interior discretization, and boundary conditions. From this specification, it is possible to algorithmically construct immersed boundary operators of this class, paralleling the wider movement towards encapsulating numerical methods in higher-level frameworks.

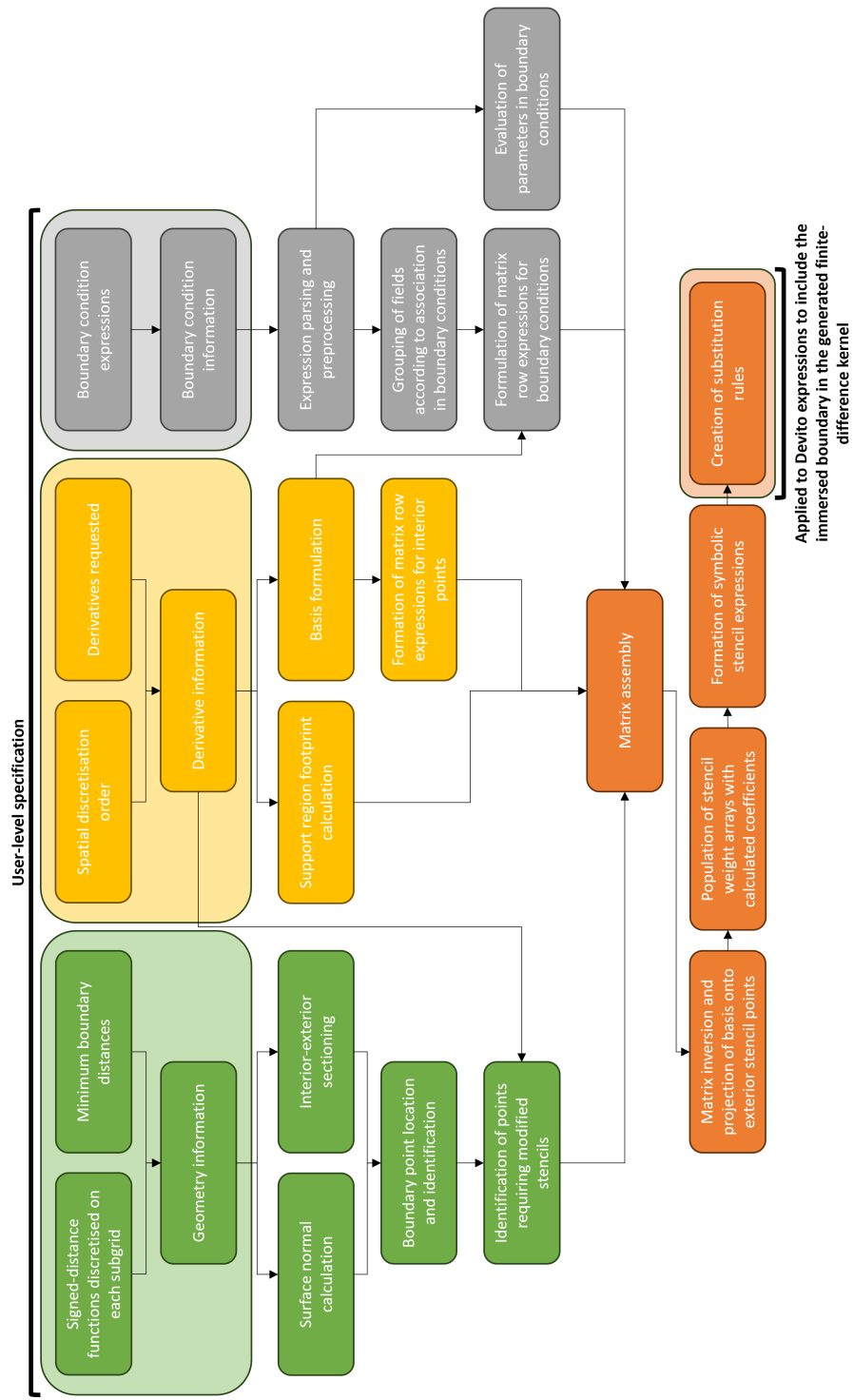


Figure 7.1.: An overview of the tasks carried out by Schism in the process of generating immersed boundary operators. From a compact user-level specification, Schism automatically generates a set of substitution rules that are supplied to Devito. Application of these substitutions replaces Devito-generated stencil expressions with the modified stencils implementing the immersed boundary.

Such efforts are exemplified by projects including Devito [Luporini et al., 2019, Louboutin et al., 2019] and Firedrake [Rathgeber et al., 2016]. By presenting the user with flexible, high-level abstractions to express their problem whilst automatically generating the highly-optimised underlying routines, a compact specification can be used to implement complex numerical schemes. These tools greatly benefit domain scientists. Engaging with the high-level abstraction enables them to focus on their field of interest, without needing to delve too deeply into the underlying code implementation. Given that the numerical methods used in such contexts are rarely the crux of such works, it follows that this separation of concerns has potential to greatly increase scientific output.

Schism aims to provide a Devito-like interface to specify immersed boundaries. From a user perspective, only a qualitative understanding of the immersed boundary treatment is required, rather than its implementation at the level of the numerical scheme. In turn, users can focus on the continuous mathematics of the problem they want to implement, enabling more straightforward implementation of immersed boundaries in overarching workflows. This reasoning is not dissimilar to that of domain-specific language (DSL) and code generation frameworks, in which the user specifies their problem in a format resembling the maths as it appears on paper, whilst the underlying numerical discretization is handled automatically.

7.1.2. API Example

Given the design of Schism as a plugin for Devito, the interface encapsulates much of the necessary information within symbols from Devito's suite of extended SymPy objects. This ensures that integration of the two packages is straightforward and intuitive from a user standpoint. It also enables Schism to interface with Devito at the topmost level.

The compactness of Devito's specification is commonly cited as a key advantage; a 2D diffusion kernel corresponds to a handful of lines. To illustrate the ability of Schism to streamline the implementation of immersed boundaries in numerical models, this simple diffusion operator is considered, highlighting the minimal additional code required, and its separability from the main body of the specification. The secondary intent

is to emphasise the correspondence between the mathematical concepts and user-level specification.

The diffusion or heat equation itself is given as

$$\frac{\partial u}{\partial t} = c \nabla^2 u \quad (7.1)$$

where u is the diffusing field and c is some constant controlling the rate of diffusion. Dirichlet boundary conditions imposed along the domain edge represent some sink; an immersed boundary approach allows for such boundaries to be non-rectilinear and smoothly curved, enabling potentially more realistic geometries.

For comparison, a basic 2D diffusion operator will be considered first, consisting of a compact block of code.

Listing 7.1: A basic diffusion operator written with Devito.

```
import devito as dv
import numpy as np

grid = dv.Grid(shape=(101, 101), extent=(1., 1.)) # FD grid

c = 1. # Diffusion constant
dt = 0.000025 # Timestep (stability limit)
u = dv.TimeFunction(name='u', grid=grid, space_order=2)
u.data[0, 40:-40, 40:-40] = 1. # Initialise field

rhs = u + dt*c*u.laplace
eq = dv.Eq(u.forward, rhs) # Update equation

# Run for 1000 timesteps
dv.Operator(eq)(t_M=1000, dt=dt)
```

The initial field specified by this code and final result after the operator has been run for 2000 timesteps are shown in Figure 7.2, demonstrating the expected softening of sharp edges caused by the diffusion process. The initial field diffuses outwards until it reaches the edges of the FD grid. The zero padding used by Devito to prevent out-of-bounds memory access enforces a zero Dirichlet boundary condition on these edges.

Restricting the computational domain to a central circle with diameter equal to half the width of the FD grid can be achieved via the use of an immersed boundary. This process is broadly separable from the initial

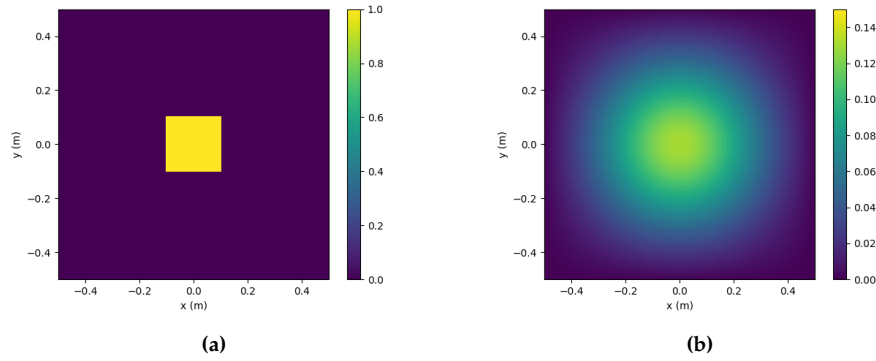


Figure 7.2.: The initial field and result after running the operator specified in Listing 7.1. Note that the colour scale in Subfigure 7.2a and Subfigure 7.2b differ from one another: the diffusion causes the maximum field value to reduce over time. In this case, the initial central concentration diffuses out to the edges of the computational grid, resulting in the field shown in Subfigure 7.2b.

diffusion operator code, beyond replacing the spatial derivative operators in the update equation. As touched upon in previous chapters, the boundary conditions themselves and thus the resultant boundary treatment is equation-agnostic, and the same substitution would be equally valid for imposing the specified boundary conditions on any other equation containing second-order spatial derivatives. In this particular case, the Dirichlet boundary condition $u(t, \mathbf{x}_b) = 0$ is imposed on the surface, which given Equation 7.1 implies $\nabla^2 u(t, \mathbf{x}_b) = 0$. Use of Schism to include such a boundary in the above operator is illustrated in the following listing, highlighting the aforementioned modularity and conciseness of the specification.

Listing 7.2: Introducing an immersed boundary to the diffusion operator

```
import devito as dv
import numpy as np

#### Schism imports ####
import sympy as sp
from schism import BoundaryConditions, BoundaryGeometry, Boundary
#####

grid = dv.Grid(shape=(101, 101), extent=(1., 1.))

#### SDF setup ####
```

```

sdf = dv.Function(name='sdf', grid=grid, space_order=2)
x_msh, y_msh = np.meshgrid(*[np.linspace(-0.5, 0.5, 101)
                               for d in grid.dimensions])
sdf.data[:] = 0.25 - np.sqrt(x_msh**2 + y_msh**2)
#####

c = 1. # Diffusion constant
dt = 0.000025 # Timestep (stability limit)
u = dv.TimeFunction(name='u', grid=grid, space_order=2)
u.data[0, 40:-40, 40:-40] = 1.

#### Boundary setup ####
zero = sp.core.numbers.Zero()
# Override default cutoff (eta=0)
bg = BoundaryGeometry(sdf, cutoff={(zero, zero): 0.})
bcs = BoundaryConditions([dv.Eq(u, 0),
                          dv.Eq(u.laplace, 0)])
boundary = Boundary(bcs, bg)
subs = boundary.substitutions((u.dx2, u.dy2))
#####

rhs = u + dt*c*u.laplace

#### Substitute stencils into RHS ####
eq = dv.Eq(u.forward, rhs.subs(subs))
#####

# Run for 1000 timesteps
dv.Operator(eq)(t_M=1000, dt=dt)

```

Setting up the immersed boundary itself requires minimal additional lines of code, and is separable from the basic model specification. Indeed, if the algebraic substitution was not applied, the operator would remain unchanged. This emphasises the concept of separation of concerns, enabling model complexity to be increased or decreased with minimal interdependence between features. This is particularly important for prototyping purposes as it enables aspects of the numerical model to be tested in isolation.

The results of the operator featuring the immersed boundary are shown in Figure 7.3.

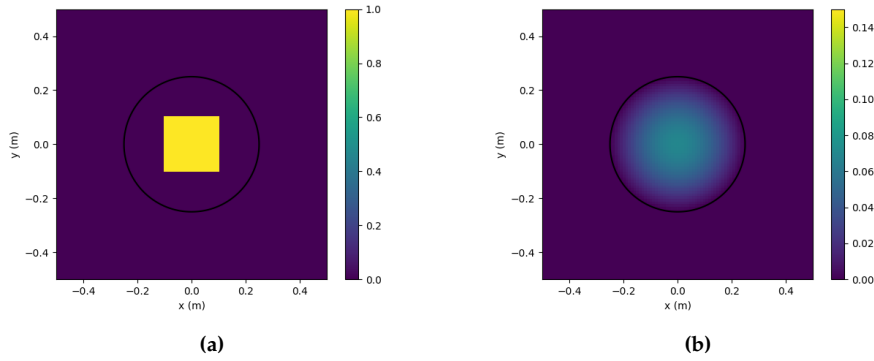


Figure 7.3.: The initial field and result after running the operator featuring an immersed boundary generated by Schism, specified in Listing 7.2. The solid black line shows the zero isosurface on the SDF, corresponding with the boundary. Colour scales are as in Subfigure 7.2a and Subfigure 7.2b. The initial central concentration diffuses outwards until it hits the immersed boundary. As the zero Dirichlet boundary condition is in closer proximity to the original concentration, the maximum value of the field at the end of the simulation is lower than in the case without the immersed boundary.

7.1.3. API Overview

Three new objects were introduced to express the immersed boundary in this diffusion operator, these being `BoundaryConditions`, `BoundaryGeometry`, and `Boundary`. These are used to represent the boundary conditions imposed, the geometry of the surface, and the immersed boundary itself respectively. The surface geometry is supplied as a `Devito Function` (a discrete, time-invariant function) containing values of the signed distance function (SDF) representing the surface. The `BoundaryGeometry` takes this and carries out various preprocessing routines, for example interior-exterior sectioning, calculation of normal directions, and location of boundary points. The additional cutoff parameter supplied in the above example simply overrides the default distance from the surface at which gridpoints are considered “too close” to be used in the discretization.

Boundary conditions are expressed symbolically as a list of `Devito Eq`, each of these representing an equality that should hold on the surface. The `BoundaryConditions` object parses and collects these, identifying links between fields specified by the boundary conditions. Both this and the `BoundaryGeometry` are used to construct a `Boundary` object, encapsulating the immersed boundary as a whole. A set of substitutions are ob-

tained using the `Boundary.substitutions()` method, returning a mapping between Devito derivative expressions and symbolic expressions for the modified stencils implementing the immersed boundary, thereby skipping the first step of Devito’s lowering¹ process. These substitutions are applied to the update stencil to include the immersed boundary in the model.

7.1.4. Numerical Scheme Specification

The information characterising a particular boundary treatment of the class described in Chapter 2 must be understood to create an algorithm to automate the process. This is important to avoid ad-hoc extensions to the framework as it is applied to new problems. Given the proposed approach, this information can be broadly summarised as follows:

Boundary Condition Information The set of equalities that should hold on the boundary surface in the specified numerical scheme.

Geometry Information Some representation of the surface upon which boundary conditions should be imposed, and its position relative to the FD grid. This must also contain some notion of interior versus exterior.

Discretization Information The discretization used in the main body of the scheme, including the grid shape, extent, and the order of the FD approximations used.

Derivative Information The derivatives within the governing PDEs which will require approximation with the modified boundary operators, and the positions at which derivatives are evaluated relative to the current index.

For some combination of these, there is a particular set of stencils which implement the specified immersed boundary. Note that this may result in identical stencils for entirely separate equations: a reflection of the separation between the immersed boundary treatment and governing equations.

¹“Lowering” is the term applied to the process of transforming a high-level specification into a low-level implementation through a series of intermediate representations. In Devito, the first of these steps is the discretisation of continuous derivative operators to obtain a symbolic representation of the FD stencils.

7.1.5. Geometry Specification and Handling

Within Schism, the boundary geometry is represented (and supplied by the user) as a signed distance function (SDF), this having several advantageous properties for geometry handling. These can be broadly separated into two categories: advantages in generation and storage, and those in handling and calculation required for the interior scheme.

SDFs can be generated from all manner of data, and routines for doing so are included in multiple widely-used packages such as VTK. This is beneficial in geophysical contexts (and indeed those more widely) as geometry data may be provided in a wide variety of ways (analytical functions, LIDAR, or DEM to name a few), and may be structured, unstructured, or even potentially some composite thereof. The precision with which this data is represented may be very high compared to the FD grid spacing, and the resolution may be much finer in some dimensions than others. Using an SDF allows the precision of the surface representation to be decoupled from the resolution at which it is stored, allowing the former to reflect the collection of the geometry data, whilst the latter reflects the interior discretization. As distance to the surface is stored at each point rather than a simple interior-exterior section, the isosurface upon which the function is zero cuts smoothly between gridpoints. This removes the need to reconcile two inconsistently represented geometries, as is the case with other commonly-used geometry specifications, polygonal meshes for example.

The other main advantage in the use of an SDF lies in its mathematical properties which greatly simplify the geometric calculations involved in boundary handling routines. The sign of function at any given point indicates its location either inside or outside of the boundary, making interior-exterior sectioning trivial. Furthermore, the value of the function is the shortest distance to the surface and its divergence is the unit normal vector to this point. As such, one can obtain the vector $\mathbf{n}$ to the closest surface point via

$$\mathbf{n} = s \nabla s, \quad (7.2)$$

where s is the SDF. Calculating the divergence with FD approximations of the same order as the interior discretization gives a boundary representation with consistent error behaviour. This procedure is essentially un-

changed in any number of dimensions: another benefit of this approach.

The SDF is supplied to Schism via the `BoundaryGeometry` object, which takes a `Devito Function` or tuple thereof whose data attribute contains the discretized SDF. An operator to calculate boundary point positions is generated and run. For discretizations featuring multiple subgrids, an SDF is supplied for each with the subgrid to which they pertain automatically determined from the specified staggering.

7.1.6. Boundary Condition Specification and Handling

`BoundaryConditions` carries out three immediate functions, these being the parsing and preprocessing of boundary condition expressions, tagging of each boundary condition with metadata, and identifying links between scalar fields specified by boundary conditions.

The parsing of boundary condition expressions, beyond checks on their validity, includes the flattening of vector and tensor expressions, and the identification of functions and derivatives contained therein, in particular those upon which the boundary condition is to be imposed (by default, these are assumed to be time-variant fields, but this can be overridden). With this, spatially-variant coefficients of these terms are substituted for placeholders for later evaluation as needed. Much of this information is stored as metadata associated with each boundary condition, alongside the dimensions with respect to which derivatives are taken, among others.

Given that the structure of the linear system required to extrapolate the fields is dependent on the linking of these fields by boundary conditions, it is worth identifying any block-diagonality prior to matrix assembly. The prior tagging of boundary conditions with the scalar fields in each can be used to determine which fields will be present in the same block via the construction of a simple network graph. The algorithm by which this graph is constructed is as follows:

1. Take a boundary condition and add the scalar fields contained therein as nodes in the graph, assuming they are not already present within the graph.
2. Insert edges connecting the nodes associated with these fields, assuming they do not already exist.

3. Iterate this process for each flattened boundary condition in turn.

Scalar fields will be connected within the graph if they are connected by boundary conditions, even if there are several links in this chain. As such, functions which will be present in a single block in the linear system will be linked within the graph, with each block corresponding to a separate set of connected nodes. An example is shown in Figure 7.4, in which the zero divergence condition inserts edges between all nodes corresponding to components of $\mathbf{v}$, whilst the pressure boundary conditions contain pressure alone and thus do not link the corresponding node to any others. This results in two disconnected groups in the graph, one containing pressure, and the other containing velocity components, implying that extrapolating pressure will require only values from the pressure fields, whilst extrapolation of any particle velocity component will require values from all of them. When a derivative is requested, these groups can be queried to obtain the fields required to construct the modified derivative. This function is carried out using the NetworkX package.

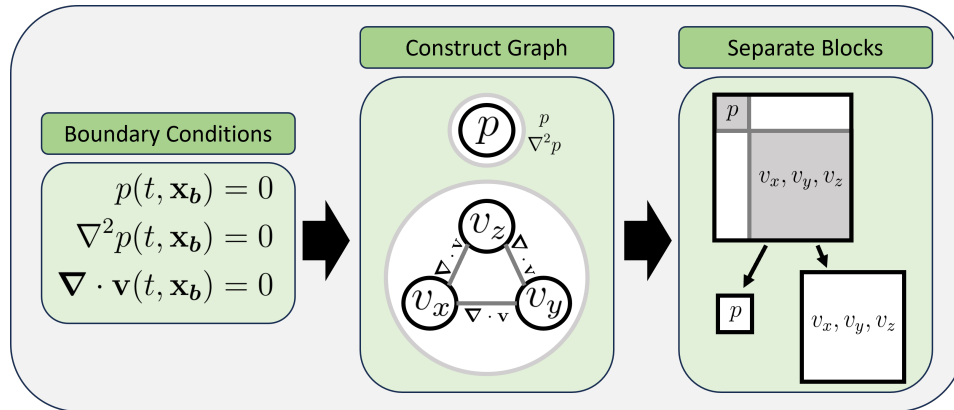


Figure 7.4.: Constructing the network for a simple set of boundary conditions, and correspondence between connected nodes on the graph and blocks in the linear system used for extrapolation.

These groupings are stored for use in constructing the linear system. When derivatives are subsequently requested, this information is used to specify the smallest linear system possible for fitting the required extrapolation.

7.2. Generation of Immersed Boundary Operators

Whilst the Boundary object collates the information defining the immersed boundary treatment, it is only when particular modified stencils are requested that the process of stencil generation is set in motion.

This process loops over the tuple of derivatives requested, with the process to obtain requisite stencils for each consisting of the following:

1. Locate indices at which stencils approximating the specified derivative will be truncated by the boundary.
2. Get the group of boundary conditions pertaining to the function of which the derivative is taken.
3. Create a symbolic representation of the generic polynomials used to approximate each of the fields present in the aforementioned group of boundary conditions.
4. Determine the stencil footprint and coefficients and create the corresponding substitutions to be applied to the equations.

A set of substitutions mapping continuous symbolic derivatives to corresponding modified stencil expressions is returned.

7.2.1. Identifying Boundary-Adjacent Points

Focusing on the first of these items, the immersed boundary treatment requires local modification of derivative operators applied at positions which would otherwise result in out-of-bounds stencil accesses. Given that this method directly modifies derivative stencils, these positions form a boundary-adjacent skin of points associated with each particular derivative. To obtain this skin, the footprint of the interior stencil for the requested derivative must be determined. This footprint is assumed to be symmetrical; staggered stencils can be considered symmetrical with one zero coefficient at the end. Given this symmetry, the following logic can be applied: an interior stencil which will require points beyond the computational domain will require at least one point located within a boundary cell. The furthest that the stencil centre could be from the boundary where this would be true is if a point on the extremities of the stencil corresponds

with one of these boundary cells. As a consequence of the symmetry, the reverse is true: the footprint of such a stencil centred on the boundary cell will cover all the positions at which applying the stencil would result in a footprint containing this boundary cell. As such, superimposing these footprints for each boundary cell and taking the intersection with the interior, one obtains the skin of points at which modified stencils will be required.

Note that the intersection must be taken with the interior mask of the subgrid onto which the derivative stencil evaluates rather than that of the subgrid upon which the function is defined. In the case of a staggered grid, it may be that the centre point of the stencil required to update some interior point is located outside the boundary. The function value at this centre point will be determined via extrapolation in the same manner as any other exterior point.

This approach errs slightly on the side of safety: for certain parameterisations this may introduce a small number of extraneous extrapolations where boundary and interior points are present in the same cell. Since these stencils require a cell containing a boundary point but consist exclusively of interior points they will be flagged as requiring extrapolations, despite the interior-exterior sectioning implying that all points are located on the interior. However, this lack of exterior points means that whilst extrapolations will be constructed for these stencil positions, they will never be used as the stencils contain no exterior points and thus remain unmodified. Examples where this is liable to occur include setups in which the cutoff parameter η is set to less than one-half.

7.2.2. Symbolic Representation of Constraints

Forming the Taylor series approximations of boundary and interior constraints required to extrapolate a function consists broadly of two steps: finding the fields which will require a series approximation and generating a symbolic representation of each.

The first of these will depend on the boundary conditions relevant to the field upon which the derivative is taken. The prior grouping of boundary conditions according to the functions which they link reduces this process to a simple lookup to find all those linked to the function to which the

derivative pertains. The formation of a polynomial to approximate each of these simply requires the order of the spatial discretization of each.

7.2.3. Bridging the Gap Between Symbolic and Numeric

To bridge the gap between the boundary conditions as expressed in the API and the corresponding rows in the extrapolation matrix, multiple levels of lowering are required. The first of these is from the continuous form of the boundary conditions as specified by the user to their Taylor series approximation. This manipulation takes place on the symbolic level, enabling the desired level of generality. This operation uses SymPy's substitution routines to replace each derivative expression with the corresponding derivative of the approximating polynomial. Note that in this polynomial, derivatives are replaced with placeholder symbols to simplify collection of like terms and factor out grid increments.

Whilst one could form and invert the matrix symbolically, the computational cost of this is considerable, and the resultant expression is quite incomprehensible and difficult to manipulate further. As such, this process is carried out numerically. Thus the next stages of lowering concern evaluation of the coefficients of the Taylor series to construct an extrapolation matrix for each point at which modified derivatives are required. To make this next step, Schism contains a routine to generate Python functions from symbolic Taylor series expressions using SymPy's `lambdify` functionality. These functions are in turn used to efficiently evaluate the corresponding matrix rows at each point as part of the matrix assembly process. The input for the resultant functions consists of 1D arrays for point locations and spatially-variable coefficients, outputting a 2D array corresponding to the matrix rows for this particular expression evaluated at these points.

Note that parameters (or assemblies thereof) needed to evaluate boundary condition rows may be spatially variant. Currently, coefficients for each derivative are considered and if recognised to be spatially variant, Devito's `SparseFunction` functionality is used to interpolate values for these expressions at the boundary points.

7.2.4. Matrix Assembly and Inversion

Fitting the polynomials used to continue the fields beyond the boundary requires solving Equation 2.24. A right-hand side where the number or relative position of entries are variable between points is liable to be a headache. Keeping track of what each value refers to introduces a layer of indirection and will make vectorised operations on the resultant set of linear systems challenging.

Given the current approach of explicitly inverting $\mathbf{A}$, zero rows in this matrix will correspond to zero columns in the inverse, thereby excluding the respective entries in the right-hand vector. A masking strategy is employed in which rows are zeroed if they correspond with points outside the boundary, such that all matrices have the same overall shape. Whilst resulting in matrices larger than they need to be, this approach is reliable, reflecting priorities of a versatile research code rather than a narrowly focused, performance-optimised one. Matrix assembly is carried out in two stages, the first of these being assembly of rows for interior constraints. As the interior footprint for any of the boundary-modified stencils will be some subset of the hyperspherical footprint when not truncated by the boundary, the rows for this footprint are calculated once and suitably masked with zeroes based on the interior-exterior sectioning.

The second stage is the assembly of rows for boundary constraints. In the interest of maintaining a consistent right-hand vector, a consistent number of rows are allocated per constraint, as in the interior case. This approach is reliable and relatively uncomplicated from a software engineering perspective. A vector of all boundary point positions relative to the centres of the modified stencils is generated alongside a mask indicating where these are located within the resultant rows. This approach ensures that boundary points located in some FD cell positioned relative to the stencil centre consistently correspond to the same row in the block of rows corresponding to some boundary condition.

For each boundary condition, the Taylor-series approximation is constructed as previously detailed and the function used to evaluate the characteristic row for this particular boundary condition is generated. This is then used to evaluate rows corresponding to the boundary condition for each of the boundary points, these being assigned to the larger matrix

using the mask. This process is repeated for all the boundary conditions, creating a list of submatrices, each associated with a particular boundary condition.

The two stacks of interior and boundary constraints are then concatenated to obtain the final stack of matrices. To avoid calculating a pseudoinverse of matrices at points where the extrapolant will not be fully constrained, the rank of each matrix in the stack is checked and a mask is formed for those which are full rank. For points where the matrix is not full-rank, the above process is repeated in due course using a support region of greater radius. The full-rank matrices are inverted using NumPy's Moore-Penrose pseudoinverse routine. Note that this is the other reason behind the maintenance of a consistent right-hand vector: this particular function can only operate on stacks of matrices of a consistent size as they are passed as a 3D array.

7.2.5. Coefficient Calculation

Obtaining the modified coefficients requires replacement of function values at exterior points with their extrapolated values. The first step of this is finding the exterior points in each boundary-intersecting stencil which will require extrapolating onto. From Equations 2.26 and 2.27, one obtains

$$\mathbf{BA}^+ \begin{bmatrix} f(\mathbf{X}) \\ g_1(\mathbf{X}_b) \\ \vdots \\ g_k(\mathbf{X}_b) \end{bmatrix} = \tilde{f}(\mathbf{X}_e). \quad (7.3)$$

As in the case of constructing $\mathbf{A}$, the variable geometry of the boundary results in differing points requiring extrapolation depending on the point at which the modified stencil is being constructed. To obtain a consistently-sized matrix such that operations can be performed on a stack and to simplify combination with the interior stencil coefficients, a generic $\mathbf{B}$ is constructed, projecting the polynomial onto all interior stencil points.

However, function values used in the interior stencil which do not require extrapolation want to be taken directly from the left-hand vector in Equation 7.3. Once $\mathbf{BA}^+$ has been calculated using the generic $\mathbf{B}$, it can be modified such that this mapping is applied. This is achieved by replacing

the corresponding rows with the direct mapping such that

$$\tilde{f}(\mathbf{X}_i) = f(\mathbf{X}_i), \quad (7.4)$$

where $\mathbf{X}_i$ are interior points in the interior stencil. Once modified in this manner, $\mathbf{BA}^+$ is henceforth referred to as $\mathbf{C}$. This modification is carried out using a mask to identify the relevant matrix rows and replace them with the appropriate direct mapping.

The final modified stencil is given by

$$\mathbf{w}^T \mathbf{C} \begin{bmatrix} f(\mathbf{X}) \\ g_1(\mathbf{X}_b) \\ \vdots \\ g_k(\mathbf{X}_b) \end{bmatrix} = d, \quad (7.5)$$

where $\mathbf{w}$ is the vector containing coefficients of the interior stencil approximating derivative d . This operation can again be applied to the stack of matrices, with $\mathbf{w}^T \mathbf{C}$ containing coefficients for the modified stencils.

7.2.6. Boundary Operator Expressions

Whilst strictly speaking the support region of the modified stencil at each point is variable according to the local boundary geometry and support region radius, in practice this would make the kernel unmanageably complex, and would be extremely challenging to implement and test. This would also certainly involve indirection to cope with the shifting, dilating stencil footprint, and it is not entirely clear how one would even express this at a high level.

As such, the stencil footprint is kept consistent, zero padding the coefficients as required to alter its extent and shape. Each stencil coefficient is a Devito Function in its own right, these being automatically tagged and paired with their corresponding stencil point.

With this, a stencil expression can be constructed and passed to Devito. This utilises the same internal class as the expressions generated when Devito discretizes a continuous derivative expression, allowing various performance optimisations applied by the Devito compiler to be leveraged (various manipulations for flop reduction for example). When forming

stencil expressions, Schism mimics the order and structure used by Devito as closely as possible.

Repeating this process for each derivative requested, a Python dictionary mapping each of the derivatives to their modified stencils is created and returned. In SymPy (and thus Devito’s extended SymPy layer), algebraic substitution can be carried out by calling a method on some given object and supplying a mapping between the symbols to be substituted and the expressions which they should be substituted for. The immersed boundary can be introduced into a preexisting operator written in Devito by calling this substitution method on the right-hand side of the update equation and supplying the modified stencils dictionary. This allows the immersed boundary to be straightforwardly added and removed for testing and debugging.

7.2.7. Support Regions

The matrix assembly process requires definition of a support region to identify the constraints needed to fit the extrapolant. Support regions are here defined by cell indices rather than absolute positions to minimise the complexity of locating and subsequently keeping track of each point. To construct a support region of some given radius, a suitable hypercube of indices is selected before applying a “cookie-cutter” to obtain indices of cells whose centres lie within the desired radius.

By default, support region radii are incremented by one at points where a full-rank extrapolation matrix cannot be formed. Due to the zero padding approach, it is necessary to keep track of the largest of these as it will dictate the form of the symbolic stencil expression. Stencils of smaller radii are padded out to match the footprint of the largest applied.

7.3. Testing, Documentation, and Availability

So that results can be reproduced and to encourage wider uptake of the method and framework, considerable effort was put into ensuring that the code is reliable, understandable, and readily available. Over 400 unit tests are present, including verification of components of the codebase and convergence testing of the stencils generated. Continuous integration

(CI) is set up using Github Actions, running the test suite with multiple Python versions on the latest Ubuntu and MacOS distributions for each commit to a pull request or the master branch. Flake8 compliance is also checked to ensure that the codebase remains readable and tidy.

Docstrings were written for all user-facing and most internal functions and classes contained within the package, and a series of tutorial notebooks were created. These demonstrate simple example implementations of various problems of interest using the package, ranging from isotropic acoustic to TTI elastic. Additional examples based on real terrain are included both for illustration purposes and to allow for replication of results presented both in this work and elsewhere. The intention was to provide an overview of the API and create a foundation from which others can build more sophisticated applications.

Schism is open-source and available at <https://github.com/devitocodes/schism>. It can be installed from this repository as a Python module using Pip, or alternatively, a Dockerfile to run the code is also included. An archived version of the codebase can be found on Zenodo at <https://zenodo.org/record/8167794>.

7.4. Future Research Directions

The most immediate future directions for this work concern optimisation of the existing framework, for example tackling various known inefficiencies present in the current codebase. Eliminating redundancy in the extrapolation-fitting process and reducing the size of the resultant linear systems would go a long way to reducing the computational cost associated with the calculation of the modified stencils.

Requested stencils may have identical or overlapping footprints. In such cases, there is considerable overlap between points at which these derivatives will require approximation via modified stencils. With the current approach, this translates to redundant extrapolations for particular points, as these are constructed on a derivative basis. To remove these redundant calculations and expedite the forming of these stencils, it would be better to construct some global set of points where extrapolants will require construction, obtaining suitable subsets as required based on combinations of stencil footprint and evaluation location. This would result in consider-

able speedup for complex boundary conditions and equations containing large numbers of derivatives, greatly reducing the computational cost of computing the modified boundary operators in such cases.

Further redundancy could be removed in the evaluation of spatially variant parameters: whilst the identification of these coefficients is carried out during boundary condition parsing, their evaluation is carried out within the loop over derivatives, identifying required boundary points for each modified stencil and evaluating the necessary values. Whilst a workable solution, this results in a great deal of redundant computation. Particular symbols may appear in multiple boundary conditions, and coefficients are reevaluated for each derivative. Alongside the proposed revision of extrapolation formation, it would make sense to perform this interpolation for all spatially-variant symbols present in boundary conditions at all boundary points prior to the loop over derivatives.

This would be complemented by a revised approach to the symbolic representation of the spatially-variant coefficients; each coefficient is currently represented by its own `Function`. In the case of large stencils or complex equations, this can lead to large numbers of objects supplied to the `Devito Operator`, causing slow compilation and messy generated code. The most straightforward way to reduce this would likely involve adding an additional dimension to the weight `Function`, this being used to store the vector of coefficients.

More substantial future use of this codebase necessitates an improved approach to linear algebra elements. The current approach is a product of circumstance, designed for generality and simplicity at the expense of performance. Whilst both are nominally achievable, substantial software engineering time and expertise would need to be invested to do so, this being precluded by the mathematical focus of this project: evaluating possibility rather than optimising the method.

Efficiencies from removing reliance on masking matrix rows with zeroes would go a long way to making this process more rapid and efficient. The use of zero padding in the linear system bloats the matrices, this being particularly egregious in the case of boundary constraints, as boundary points are more sparse than interior points and the number of constraints per point more numerous. This would require some means of handling variable right-hand vectors and thus matrices of variable size. This would

likely require some form of indirection to handle this telescoping of the right-hand vector whilst keeping track of all the entries. This will also complicate the process of assembling the matrix, as rows will correspond to different constraints depending on the points used to construct the support region. It is also likely that some kind of purpose-built data structure would be required to collate the matrices and keep track of their right-hand sides. Given that it would no longer be possible to represent all the matrices (or at least the majority of them) as a single array, operating on them at the Python level would require loops with large numbers of iterations, which tend to be fairly slow. An additional layer of lowering would likely be needed to rapidly assemble, keep track of, and invert these matrices efficiently.

Eliminating zero padding necessitates adaptive stencil footprints, corresponding to changing the update expression for each point depending upon its position. Modifying the update expression presents its own issues: a naive implementation would require a huge number of update expressions to implement depending upon the position within the domain at which the update is to be applied. One is then faced with issues of trying to keep some semblance of readability and conciseness in the kernel and avoiding large strings of conditional statements. How best to achieve this is unclear. However, it is anticipated that it would be best implemented as some automatically applied transformation of some higher-level representation to manage complexity.

It would also be worth considering whether alternative libraries such as PETSc would be more suitable for the linear algebra portions of the algorithm, particularly for scaling to larger problems. The computational expense and memory requirements of the matrix inverse were found to become problematic for larger 3D problems featuring boundary conditions with large numbers of fields, an issue possibly addressed by a more specialised library.

Further improvements to the codebase include extension to boundary conditions with nonzero right-hand sides, including those where this is spatially or temporally variable. This would allow for specification of variable fluxes across the boundary for example, potentially opening up additional applications in contexts beyond those considered in this body of work. The other immediate extension would be the introduction of

boundary-aware sparse injection and interpolation routines. In seismic applications, source and receiver terms are often located immediately below the surface. However, the objects used to implement these in Devito have no knowledge of the immersed boundary which can lead to issues when sources or receivers are placed at too shallow a depth below the immersed surface. In such cases, interpolation onto or off of the grid can use a set of grid points either partially or wholly outside the model domain, forcing sources and receivers to be locally repositioned to function properly in conjunction with the immersed boundary.

Reducing the computational cost of including immersed boundaries is desirable in practical applications; the modified operators are considerably more expensive to apply than their interior counterparts, and thus it is desirable to limit them to the smallest area of the computational domain possible. Furthermore, zero updates are still applied outside the computational domain. Rectifying this would likely involve something akin to the two-kernel approach taken by Stoltzfus et al. [2021] in their boundary implementation for computational acoustics, using a conditional mask to limit boundary operators to boundary-adjacent nodes. This could alternatively be implemented as a subdomain (a set of limits on an iteration space rather than a conditional mask). With the conditional approach, overhead is introduced by checking the conditional throughout the domain, even if these points are known to be far from the boundary, whilst in the latter, highly irregular geometries could produce degenerate sets of subdomains (i.e. large numbers of tiny domains). Some combination of these two approaches is likely to be most performant, with subdomains used on a macro scale and conditionals on a smaller scale. It is proposed that an algorithm could be devised which, given some particular geometry, would determine a pseudo-optimal set of subdomains, and initialise the conditional masks within them. The heuristics with which to evaluate optimality and the parameters which may affect this are not currently known however. This conditional idea could also be extended to apply different update expressions depending on the stencil footprint. For stencils of varying radii, this would be relatively straightforward to implement.

Reducing the additional memory requirements of the immersed boundary within the kernel itself would require the ability to define the discrete functions containing the weights only over the requisite subset of the do-

main. This would reduce the amount of memory dedicated to storing zeroes, thereby reducing the overall amount of memory which must be allocated for the stencil weights.

It is anticipated that further, more aggressive and exotic optimisations may be possible to reduce the overhead of the immersed boundary. Flop-reduction and memory-locality optimisations are common in FD contexts and their interaction with the immersed boundary is of interest.

Chapter 8

Conclusions

This work has presented an immersed boundary formulation that addresses the lack of generality and problem-agnosticism present in existing approaches. This enabled a consistent approach to be applied across a range of seismological problems, corresponding with a diversity of conditions, equations, and discretizations. The generality of the mathematical method was leveraged to automatically generate immersed boundary operators from a high-level specification, reducing specification of complex geophysical models to a compact body of code. Engaging with the problem via this abstraction layer, novel immersed boundary topography implementations, hitherto uncommon in seismic modelling, could be devised for a range of seismological problems. Separation of concerns is thereby achieved, facilitating the inclusion of immersed boundary methods in overarching workflows.

8.1. Summary

In seismic wavefield modelling, a free-surface condition must be imposed on the topographic surface. Whilst acoustic and elastic free surfaces are conceptually similar, their forms differ considerably, one acting on a scalar field, whilst the other relates components of a tensor. Higher-order conditions are derived from the governing equations, resulting in diverse conditions to be imposed when one considers the potential variation here. Whilst the governing equations dictate the form of conditions imposed, the immersed boundary treatment can be considered separately from the

equations that it is used to solve.

Leveraging concepts of abstraction and automation was key to accelerating the development of immersed boundary topographic treatments for the various problems considered. By developing means of expressing the overarching class of methods, implementations for specific problems could be rapidly specified and generated. Duplication of effort when addressing particular problems is subsequently avoided and a wide suite of problems can be addressed with minimal adaption of the top-level specification.

The immersed boundary approach was designed to accommodate this flexibility. By using N -dimensional Taylor-series basis functions and hyperspherical support regions, a wide range of boundary conditions could be naturally represented, as could the local shape of the surface. These were used to approximate a generalised boundary condition, of which all those of interest were special cases. In doing so, the need to form approximations of boundary conditions based on domain knowledge was avoided. Furthermore, the diversity of boundary conditions introduced by the physics of the various problems of interest can be straightforwardly accommodated.

Initial experiments using the method developed to impose free and rigid surfaces with isotropic acoustic wave equations gave way to investigation of the free surface for various anisotropic pseudoacoustic formulations. These efforts culminated with the development of free-surface treatments for first and second-order elastic wave equations, in the latter case stretching to those featuring both vertical and tilted transverse isotropy. Due to the degree of automation leveraged in implementing these various conditions as immersed boundaries, the gulf in complexity between each of these steps and the next was made considerably easier to cross. For example, whilst adapting an isotropic acoustic propagator featuring immersed boundaries to the TTI formulation of Fletcher et al. [2009] would represent a significant effort if implemented as low-level code, the difference between the two was minimal in the high-level immersed boundary specification enabled by the method devised.

Given the wide range of equations used across various applications according to the tradeoff between complexity and required physics, there is a corresponding range of boundary conditions which must be imposed.

Devising immersed boundary treatments for each individually would likely prove onerous, acting as a barrier to integration of immersed boundaries into higher-level workflows. The development of a relatively one-size-fits-all approach represents a significant step towards more widespread application of immersed boundary treatments in these contexts.

The development of Schism, an open-source framework to automatically generate immersed boundary operators from a Devito-like specification was instrumental in achieving this. Interfacing near-seamlessly with Devito, this enabled complex geophysical models featuring immersed boundaries to be expressed via a compact specification. The introduced separation of concerns between model specification and underlying discretization enabled the inclusion of topography within higher-level algorithms including RTM. By ensuring that focus can be kept on the top-level application without the need to delve into lower-level implementation, barriers for the integration of immersed boundary methods into overarching routines are removed; seismic wave propagators are just one component of an inversion or migration algorithm. Concise specification improves code readability whilst maximising reuse, greatly increasing accessibility of the method. Furthermore, packaging the method into an open-source codebase ensures reproducibility of results; code used to produce them is both accessible and well-tested.

Overall, the efforts presented herein represent innovation on the state-of-the-art on multiple fronts. On the surface level, novel topography implementations have been developed for multiple seismological problems. Based on a sharp immersed boundary method, these treatments enable irregular curvilinear geometries to be accurately represented on regular, structured grids whilst suppressing spurious scattering and retaining consistency with the interior discretization. It has been demonstrated that such approaches present a viable topography implementation capable of handling realistic topography profiles, sidestepping the need for unstructured meshes as in FEM and SEM or the complex transformations required by curvilinear FD.

The main innovation was automated generation of immersed boundary operators from a high-level specification. Abstracting the boundary treatment from the governing equations allowed a consistent method to be applied, regardless of overarching problem. Automating the general

mathematical method to devise immersed boundary treatments for each specific problem enabled exploration of immersed boundary treatments for an unprecedented range of boundary conditions and equations of interest to geophysical wavefield modelling.

8.2. Limitations

Whilst the specific limitations of this work have been discussed in previous chapters, it is worth highlighting some of the wider aspects that remain unaddressed.

CFD and aeroacoustics studies commonly make use of immersed boundaries. However, exploration of the suitability of immersed boundary operators generated via the proposed mathematical method was beyond the scope of this investigation. Given the abstraction of the immersed boundary treatment from the governing equations, such operators could be specified and devised, but their accuracy and stability remains uncertain.

It was necessary to exclude points from the discretization to achieve stability when imposing immersed free surface conditions on acoustic and pseudoacoustic wave equations, whilst instability was also apparent in the elastic cases. The source of these instabilities remains poorly understood, and may not originate from a common source across cases in which it was present. Rigorous stability analysis is challenging, often prohibitively so, whilst *a posteriori* checks for stability rarely aid in rectifying any instability. *A priori* checks are more useful if they can be integrated into the design of the scheme, although doing so is not necessarily straightforward. Furthermore, the principle of separation between immersed boundary operators and the overarching equations and numerical scheme makes the design of abstractions to include stability constraints in the stencil generation process challenging.

8.3. Future Research Directions

Several avenues for future research have been identified based on this work. The present method assumes that the boundary surface represents the edge of the computational domain, at least for the function to which modified derivative stencils are applied. However, in many problems,

the wider computational domain extends beyond the boundary surface at which some particular coupling is applied. The immediate example in seismic contexts is the seafloor: propagation in the water layer is acoustic, whilst elastic waves propagate in the strata. Minimising computational expense by solving separate equations in each layer is an attractive solution, although often this involves a transition region whose upper surface is in the water layer in which both methods overlap. By imposing the correct interface condition on the surface, the two propagators can be coupled as in Sun et al. [2017]. The immersed boundary approach to this would avoid previously-detailed issues with curvilinear transformation approaches, whilst avoiding the fine discretizations required to suppress spurious scattering as described by de Groot-Hedlin [2004]. It is of interest to understand how extension to accommodate such cases would be achieved for the current mathematical approach, and how this would translate to the abstractions with which it is expressed. The introduction of internal interfaces also raises the possibility of intersecting boundary surfaces, for example in seismic models of coastal or lacustrine settings.

Ignoring topographic effects is known to cause aberrations in the results of multiple wave-equation-based applications. However, the effort involved to rebuild such applications around the FEM/SEM or curvilinear FD solvers conventionally required to accurately represent topography is considerable and broader modification of the algorithm is likely to be necessary. Furthermore, whilst considerable resources can be put into meshing in a research context, practical applications necessitate fast, robust solutions. An immersed boundary approach addresses these issues as the regular structured grid is retained, as are the operators throughout the majority of the domain. The framework built for this work further facilitates such efforts, particularly as immersed boundaries can be straightforwardly included in Devito-based codes with only a handful of lines needing modification. Application of many such algorithms is limited to targets with favourable properties - marine settings or those with minimal topographic variation, or built upon SEM and suchlike from the ground up where topographic variation or the Earth's curvature are unavoidable. Whilst the ability to rebuild some examples in the latter category around an FD solver (regional-scale FWI for example) may be beneficial, of greater interest is the application of algorithms such as FWI and RTM to targets

which have to date been challenging to image, volcanoes for example.

To encourage wider uptake, a performant implementation becomes a necessity. Whilst addressing known inefficiencies in the existing codebase is a priority, performance optimisation of FD kernels featuring immersed boundaries has been little discussed. Much of the overhead introduced by the present implementation is due to the zero-padding of stencils to account for their constantly shifting footprint. This introduces a large number of floating-point operations and array accesses which do not contribute to the final result. This is particularly apparent for interior stencils which are padded out to the size of the considerably more expansive and expensive boundary operators, or exterior points where a large update stencil consisting entirely of zeroes is applied. There are two aspects to addressing this, the first being a means of applying different update expressions in the generated C code, corresponding with the protean boundary stencils. The second is separation of boundary and interior kernels in the manner described by Stoltzfus et al. [2021]. By limiting the relatively expensive boundary operators to the minimum necessary area, applying a conventional update on the interior, and passing over exterior nodes, computational cost is minimised. The irregular nature of the boundary makes this separation non-trivial however. For example, whilst one could check a conditional at every point in the domain at every timestep, this would introduce unnecessary overhead as large swaths of the computational domain are known to be well outside or inside the computational domain. A set of subdomains based on minimum and maximum iteration bounds could alternatively be used, breaking the interior domain into a set of hyperrectangles. However, it is not hard to imagine some scenario in which an irregular surface leads to the creation of a large number of small iteration spaces, or those with unhelpful aspect ratios or alignment with loop blocks. Some combination of these two approaches, using subdomains at the larger scale and conditionals at the smaller scale would likely be the ideal compromise. Automated decomposition of some complex interior region via these two approaches would be of interest as there are several unknowns on this front, including performance tradeoffs and the nature of suitable quality heuristics. The boundary represents a relatively thin and potentially highly-irregular skin of points. Applying some combination of subdomains and conditionals may still lead to either conditionals

applied over large areas or large numbers of small subdomains. Taken in combination with the need for variable update expressions, a simultaneous sparse iteration over boundary points and functions evaluating the expression right-hand-side is one proposed approach.

Memory optimisation, both in terms of memory footprint and memory pressure would further bolster kernel performance. In the former case, avoidance of zero padding where possible would significantly reduce the memory required to store stencil coefficients, particularly for larger stencils. Only storing spatially variable boundary weights over the areas where they are used would further reduce the memory footprint associated with the immersed boundary. Whilst minimising stencil footprints by avoiding zero padding would reduce memory pressure, and thus should result in improved performance, more sophisticated memory optimisations are less obvious. Structured FD kernels often leverage an array of cache optimisations designed to maximise data locality and reuse between iterations. How such optimisations interact with and can be formulated for the irregular iteration patterns introduced by the immersed surface is an open question however, and worthy of future attention.

Finally, the preprocessing stage of generating the modified stencil coefficients could be accelerated through the use of more efficient and sophisticated approaches to solving the linear systems formed. Matrix inverses are expensive to compute and thus commonly avoided where possible. This is particularly true when inverting larger linear systems, corresponding with higher-order discretizations, higher dimensionality, and larger stencil footprints. It would be preferable to repose the inverse and various matrix manipulations as some kind of linear solve which could be handled with a suitable library such as PETSc. This would greatly reduce the burden associated with this step whilst use of a more suitable backend would enable greater scalability.

Refinement of the mathematical method would be beneficial in avoiding the need for expensive operations; well-designed numerical schemes are commonly overlooked in the pursuit of FD kernel performance. Investigation of basis functions beyond N-dimensional Taylor series may prove fruitful, both in terms of scheme behaviour and computational cost. Basis functions with more desirable properties, particularly those conducive to stability when used to continue fields beyond the boundary, are of interest,

and a wide range could potentially be used. Furthermore, basis functions with fewer coefficients at some given order require smaller support regions to fit, which in turn corresponds to smaller stencils with reduced computational cost. Furthermore, improved means of conditioning the extrapolants would likely improve the stability of the resultant numerical scheme.

Finally, extension to wider use cases, particularly those in aeroacoustics and CFD, would benefit from extension of the method to accommodate features such as moving boundaries and implicitness. Such enhancements would likely require some thought regarding both the low-level implementation of the immersed boundary operators and the high-level abstractions with which they are expressed.

Appendix A

Imaging and Gradients

To demonstrate both the importance of topography handling to accurate imaging in land settings, and the utility of the mathematical and software frameworks described in this work for the purpose of introducing this topography handling to imaging algorithms, an FWI gradient calculation was carried out for a synthetic example. For FWI to converge, it is necessary to construct high-quality tomographic gradients; inaccuracies and artefacts will compound through iterations of FWI, yielding an inaccurate and potentially misleading final model. Failure to correctly account for wavefield kinematics results in artefacts in the subsurface model as the inversion attempts to accommodate features in the data produced by physics missing from the propagators [Bleibinhaus and Rondenay, 2009]. By integrating an immersed free surface into the gradient calculation, accurate imaging can be achieved, even where pronounced topography is present.

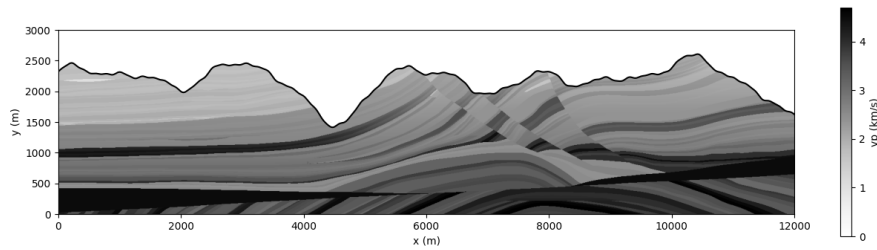


Figure A.1.: The Marmousi model modified with mountainous topography.

Taking the standard Marmousi model, sampled with 7.5m spacing,

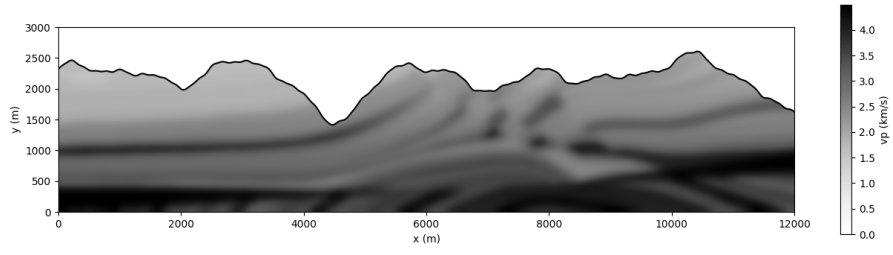


Figure A.2.: The smoothed Marmousi model featuring the same topography as in Figure A.1.

mountainous topography was introduced, shown in Figure A.1. A corresponding smoothed model was produced as an initial model for gradient calculation, shown in Figure A.2. Observed data was created by forward propagation through the sharp model; topography was implemented as an immersed free surface. Sources and receivers were equally spaced at 30m depth below the surface. Data was generated for 601 sources, with each shot recorded at 1201 receivers. An example shot gather from this data is shown in Figure A.3.

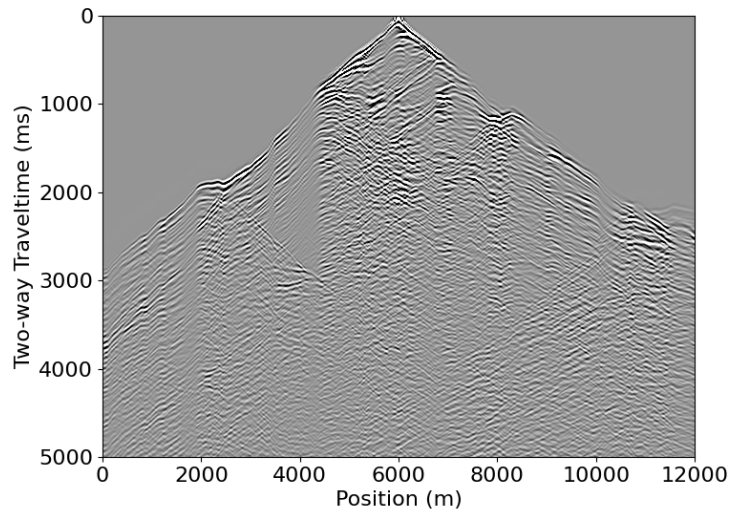


Figure A.3.: A shot gather from the dataset created for this example

Taking the smoothed model shown in A.2 as the initial model, a 20Hz FWI tomographic gradient was calculated using the observed data. Im-

mersed boundary topography was included in forward and adjoint propagators used to calculate the gradient.

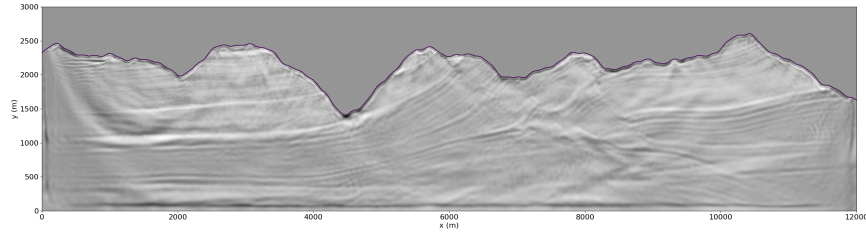


Figure A.4.: The illumination-corrected tomographic gradient calculated using propagators which incorporate topography. Note the overall good resolution of the image.

The illumination-corrected tomographic gradient is shown in Figure A.4, showing good resolution of the fine structure present in the original model. In particular, the shape and position of the major layers and lenses, fine-scale texture, and the central fault complex are clearly visible in the gradient update. Furthermore, the illumination is remarkably good as a product of the vertical spread of source and receiver positions, and the tendency for the topography and subsurface model to interact, producing reverberations. Illumination is well-balanced between shallow and deep regions of the model, as shown in Figure A.5.

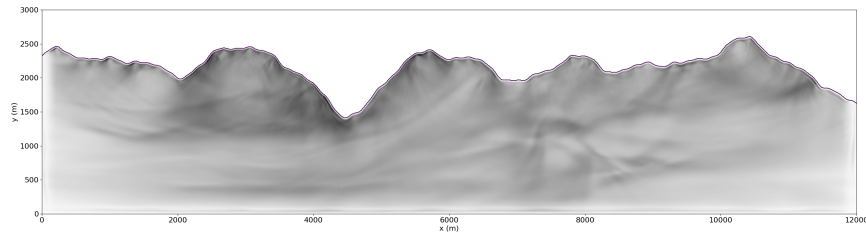


Figure A.5.: The illumination correction for the model featuring topography. Note the visibility of several horizontal reflectors in this plot, including those in deep areas.

Applying a Laplacian filter to this tomographic yields a result akin to an inverse-scattering imaging condition, highlighting sharp edges in the update. The Laplacian-filtered gradient is shown in Figure A.6.

To underscore the critical nature of accounting for topographic variation in imaging applications, an alternative gradient was calculated assuming

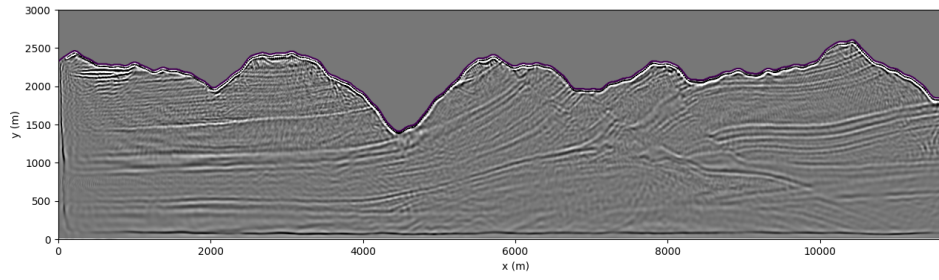


Figure A.6.: The result of applying a Laplacian filter to the calculated gradient. Note the clear texture of the original model, the outline of the fault complex, and the sharpness of several smaller features including lenses near the surface in the left and centre of the model.

a flat surface corresponding the mean topographic elevation. Sources and receivers were positioned 30m below the horizontal free surface, whilst retaining their original horizontal positions. A tomographic gradient, shown in Figure A.7, was calculated using this setup and the observed data. The quality of this gradient is extremely poor, with few discernible features, underlining the importance of accurate topography handling in imaging contexts and the consequences of its omission.

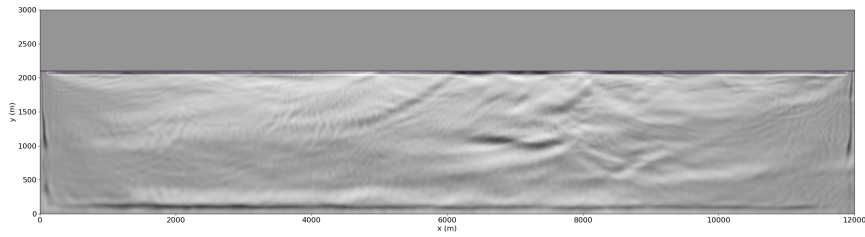


Figure A.7.: The illumination-corrected tomographic gradient calculated using propagators which omit topography. Little structure is discernible.

Bibliography

- Ò. C. Agudo, N. Vieira, M. Warner, and J. Morgan. Acoustic full-waveform inversion in an elastic world. *Geophysics*, 83(3):257–271, 2018.
- A. M. Almuheidib and M. N. Toksoz. Finite difference elastic wave modeling with an irregular free surface using ADER scheme. *Journal of Geophysics and Engineering*, 12:435–447, 2015.
- P. Anquez, J. Pellerin, M. Irakarama, P. Cupillard, B. Lévy, and G. Caumon. Automatic correction and simplification of geological maps and cross-sections for numerical simulations. *Comptes Rendus - Geoscience*, 351(1): 48–58, 2019.
- D. Asimaki and K. Mohammadi. On the complexity of seismic waves trapped in irregular topographies. *Soil Dynamics and Earthquake Engineering*, 114(May):424–437, 2018.
- G. E. Backus. Long-wave elastic anisotropy produced by horizontal layering. *Journal of Geophysical Research*, 67(11):4427–4440, 1962.
- H. Barucq, L. Boillot, H. Calandra, and J. Diaz. Absorbing Boundary Conditions for 2D Tilted Transverse Isotropic elastic media. In *ESAIM: Proceedings and Surveys*, volume 45, pages 400–409, 2014.
- A. Bayliss, K. E. Jordan, B. J. LeMesurier, and E. Turkel. A fourth-order accurate finite-difference scheme for the computation of elastic waves. *Bulletin of the Seismological Society of America*, 76(4):1115–1132, 1986.
- E. Baysal, D. D. Kosloff, and J. W. Sherwood. Reverse time migration. *Geophysics*, 48(11):1514–1524, 1983.

- S. Beller, V. Monteiller, S. Operto, G. Nolet, A. Paul, and L. Zhao. Lithospheric architecture of the South-Western Alps revealed by multiparameter teleseismic full-waveform inversion. *Geophysical Journal International*, 212(2):1369–1388, 2018.
- H. Ben-Hadj-Ali, S. Operto, J. Virieux, H. Ben, H. Ali, S. Operto, and J. V. Velocity. Velocity model building by 3D frequency-domain full-waveform inversion of wide-aperture seismic data To cite this version : HAL Id : insu-00200039. *Geophysics*, 73(5):VE101–VE117, 2008.
- S. Bilbao. Immersed boundary methods in wave-based virtual acoustics. *The Journal of the Acoustical Society of America*, 151(3):1627–1638, 2022.
- F. Bleibinhaus and S. Rondenay. Effects of surface scattering in full-waveform inversion. *Geophysics*, 74(6):69–77, 2009.
- T. Bohlen and E. H. Saenger. Accuracy of heterogeneous staggered-grid finite-difference modeling of Rayleigh waves. *Geophysics*, 71(4):109–115, 2006.
- D. M. Boore. A note on the effect of simple topography on seismic SH waves. *Bulletin of the Seismological Society of America*, 62(1):275–284, 1972.
- D. Borisov, R. Modrak, F. Gao, and J. Tromp. 3D elastic full-waveform inversion of surface waves in the presence of irregular topography using an envelope-based misfit function. *Geophysics*, 83(1):R1–R11, 2018.
- D. Borisov, F. Gao, P. Williamson, and J. Tromp. Application of 2D full-waveform inversion on exploration land data. *Geophysics*, 85(2):R75–R86, 2020.
- J. Bowling, S. Ji, D. Lin, M. Reasnor, M. Staines, and N. Burke. From isotropic to anisotropic: Puma/mad dog wide azimuth data case study. In *79th Society of Exploration Geophysicists International Exposition and Annual Meeting 2009, SEG 2009*, pages 246–250, 2009.
- C. Brehm and H. F. Fasel. A novel concept for the design of immersed interface methods. *Journal of Computational Physics*, 242:234–267, 2013.

- C. Brehm, C. Hader, and H. F. Fasel. A locally stabilized immersed boundary method for the compressible Navier – Stokes equations. *Journal of Computational Physics*, 295:475–504, 2015.
- X. H. Cai, C. J. Huang, Tao-Ran, X. P. Fan, and H. Liu. A mesh-free finite-difference scheme for frequency-domain acoustic wave simulation with topography. *Applied Geophysics*, 19(3):113, 2022.
- J. M. Carcione. Seismic modeling in viscoelastic media. *Geophysics*, 58(1):110–120, 1993.
- J. M. Carcione, D. Kosloff, and R. Kosloff. Viscoacoustic wave propagation simulation in the earth. *Geophysics*, 53(6):769–777, 1988.
- G. J. Corrêa, M. Spiegelman, S. Carbotte, and J. C. Mutter. Centered and staggered Fourier derivatives and Hilbert transforms. *Geophysics*, 67(5):1558–1563, 2002.
- E. Crase, A. Pica, M. Noble, J. McDonald, and A. Tarantola. Robust elastic nonlinear waveform inversion: application to real data. *Geophysics*, 55(5):527–538, 1990.
- C. de Groot-Hedlin. Criteria for discretization of seafloor bathymetry when using a staircase approximation: Application to computation of T-phase seismograms. *The Journal of the Acoustical Society of America*, 115(3):1103–1113, 2004.
- J. de la Puente, M. Ferrer, M. Hanzich, J. E. Castillo, and J. M. Cela. Mimetic seismic wave modeling including topography on deformed staggered grids. *Geophysics*, 79(3):125–143, 2014.
- J. Dellinger and J. Etgen. Wave-field separation in two-dimensional anisotropic media. *Geophysics*, 55(7):914–919, 1990.
- X. Deng, H. Wu, C. Sun, and R. Gao. Mesh-free least-squares reverse-time migration. *Journal of Geophysics and Engineering*, 20(2):185–195, 2023.
- N. S. Dhamankar, G. A. Blaisdell, and A. S. Lyrantzis. Implementation of a sharp immersed boundary method in a 3-D Multi-block large eddy simulation tool for jet aeroacoustics. In *53rd AIAA Aerospace Sciences Meeting*, pages 1–28, 2015.

- F. DI Michele, J. May, D. Pera, V. Kastelic, M. Carafa, C. Smerzini, I. Mazzieri, B. B. Rubino, P. F. Antonietti, A. Quarteroni, R. Aloisio, and P. Marcati. Spectral element numerical simulation of the 2009 L'Aquila earthquake on a detailed reconstructed domain. *Geophysical Journal International*, 230(1):29–49, 2022.
- Q. Dong, Y. Rao, and D. Han. Pure acoustic simulation in TTI media for irregular topography models. In *CGS/SEG International Geophysical Conference*, number April, pages 629–632, 2017.
- F. Dupros, F. D. Martin, E. Foerster, D. Komatitsch, and J. Roman. High-performance finite-element simulations of seismic wave propagation in three-dimensional nonlinear inelastic geological media. *Parallel Computing*, 36:308–325, 2010.
- E. Duveneck and P. M. Bakker. Stable P-wave modeling for reverse-time migration in tilted TI media. *Geophysics*, 76(2), 2011.
- E. Duveneck, P. Milcik, P. M. Bakker, and C. Perkins. Acoustic VTI wave equations and their application for anisotropic reverse-time migration. In *78th Society of Exploration Geophysicists International Exposition and Annual Meeting, SEG 2008*, number 1, pages 2186–2190, 2008.
- D. Fee, L. Toney, K. Kim, R. W. Sanderson, A. M. Iezzi, R. S. Matoza, S. D. Angelis, A. D. Jolly, J. J. Lyons, M. M. Haney, and D. Keir. Local Explosion Detection and Infrasound Localization by Reverse Time Migration Using 3-D Finite-Difference Wave Propagation. *Frontiers in Earth Science*, 9(February):1–14, 2021.
- Z. Feng and G. T. Schuster. Elastic least-squares reverse time migration Elastic least-squares reverse time migration. *Geophysics*, 82(2):S143–S157, 2017.
- R. P. Fletcher, X. Du, and P. J. Fowler. Reverse time migration in tilted transversely isotropic (TTI) media. *Geophysics*, 74(6):179–187, 2009.
- P. J. Fowler, X. Du, and R. P. Fletcher. Coupled equations for reverse time migration in transversely isotropic media. *Geophysics*, 75(1):S11–S22, 2010.

- A. Frankel and W. Leith. Evaluation of topographic effects on P and S-waves of explosions at the northern Novaya Zemlya test site using 3-D numerical simulations. *Geophysical Research Letters*, 19(18):1887–1890, 1992.
- D. V. Gaitonde and M. R. Visbal. High-order schemes for Navier-Stokes equations: algorithm and implementation into FDL3DI. Technical report, 1998.
- D. V. Gaitonde and M. R. Visbal. Pade-type higher-order boundary filters for the Navier-Stokes equations. *AIAA journal*, 38(11):2103–2112, 2000.
- D. V. Gaitonde, J. S. Shang, and J. L. Young. Practical aspects of higher-order numerical schemes for wave propagation phenomena. *International Journal for Numerical Methods in Engineering*, 45(12):1849–1869, 1999.
- K. Gao and L. Huang. An improved rotated staggered-grid finite-difference method with fourth-order temporal accuracy for elastic-wave modeling in anisotropic media. *Journal of Computational Physics*, 350:361–386, 2017.
- L. Gao, R. Brossier, B. Pajot, J. Tago, and J. Virieux. An immersed free-surface boundary treatment for seismic wave simulation. *Geophysics*, 80(5):193–209, 2015.
- D. Goldstein, R. Handler, and L. Sirovich. Modelling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105:354–366, 1993.
- R. W. Graves. Simulating Seismic Wave Propagation in 3D Elastic Media Using Staggered-Grid Finite Differences. *Bulletin of the Seismological Society of America*, 86(4):1091–1106, 1996.
- V. Grechka, L. Zhang, and J. W. Rector. Shear waves in acoustic transversely isotropic media. *Geophysics*, 69(2):576–582, 2004.
- D. W. Griffiths and G. A. Bollinger. The Effect of Appalachian Mountain Topography on Seismic Waves. *Bulletin of the Seismological Society of America*, 69(4):1081–1105, 1979.

- L. Guasch, O. Calderón Agudo, M. X. Tang, P. Nachev, and M. Warner. Full-waveform inversion imaging of the human brain. *npj Digital Medicine*, 28(3):1–12, 2020.
- K. Hayashi, D. R. Burns, and M. N. Tokso. Discontinuous-grid finite-difference seismic modeling including surface topography. *Bulletin of the Seismological Society of America*, 91(6):1750–1764, 2001.
- D. Heimann and R. Karle. A linearized Euler finite-difference time-domain sound propagation model with terrain-following coordinates. *The Journal of the Acoustical Society of America*, 119(6):3813–3821, 2006.
- S. Hestholm and B. Ruud. 2D finite-difference elastic wave modelling including surface topography. *Geophysical Prospecting*, 42:371–390, 1994.
- S. Hestholm and B. Ruud. 3D free-boundary conditions for coordinate-transform finite-difference seismic modelling. *Geophysical Prospecting*, 50(1):463–474, 2002.
- S. Hestholm and C. Street. 3-D surface topography boundary conditions in seismic wave modelling. In *SEG Technical Program Expanded Abstracts*, pages 2432–2435, 2000.
- S. O. Hestholm, B. O. Ruud, and E. S. Husebye. 3-D versus 2-D finite-difference seismic synthetics including real surface topography. *Physics of the Earth and Planetary Interiors*, 113:339–354, 1999.
- J. W. Hobro, C. H. Chapman, and J. O. Robertsson. A method for correcting acoustic finite-difference amplitudes for elastic effects. *Geophysics*, 79(4):T243–T255, 2014.
- S. Hu, W. Liu, and X. Wang. Improved elastic wave equations with application to stable wavefield simulating in complex TTI media. In *13th SEGJ International Symposium*, number November, pages 30–33, 2018.
- W. Hu. An improved immersed boundary finite-difference method for seismic wave propagation modeling with arbitrary surface topography. *Geophysics*, 81(6):311–322, 2016.

- M. J. Huiskes, R. Plessix, and W. A. Mulder. Acoustic VTI full-waveform inversion with 3-D free-surface topography. In *79th EAGE Conference and Exhibition 2017*, 2017.
- T. Kärnä, S. C. Kramer, L. Mitchell, D. A. Ham, M. D. Piggott, and A. M. Baptista. Thetis coastal ocean model: Discontinuous Galerkin discretization for the three-dimensional hydrostatic equations. *Geoscientific Model Development*, 11(11):4359–4382, 2018.
- K. R. Kelly, R. W. Ward, S. Treitel, and R. M. Alford. Synthetic seismograms: a finite-difference approach. *Geophysics*, 41(1):2–27, 1976.
- C. A. Kennedy and M. H. Carpenter. Several new numerical methods for compressible shear-layer simulations. *Applied Numerical Mathematics*, 14(4):397–433, 1994.
- K. Kim and J. M. Lees. Local Volcano Infrasound and Source Localization Investigated by 3D Simulation. *Seismological Research Letters*, 85(6):1177–1186, 2014.
- K. Kim, D. Fee, A. Yokoo, and J. M. Lees. Acoustic source inversion to estimate volume flux from volcanic explosions. *Geophysical Research Letters*, 42(13):5243–5249, 2015.
- D. Köhn, T. Meier, M. Fehr, D. De Nil, and M. Auras. Application of 2D elastic Rayleigh waveform inversion to ultrasonic laboratory and field data. *Near Surface Geophysics*, 14(5):461–476, 2016.
- G. Lacanna, M. Ichihara, M. Iwakuni, M. Takeo, M. Iguchi, and M. Ripepe. Influence of atmospheric structure and topography on infrasonic wave propagation. *Journal of Geophysical Research: Solid Earth*, 119(4):2988–3005, 2014.
- M.-c. Lai and C. S. Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *Journal of Computational Physics*, 719:705–719, 2000.
- H. Lan and Z. Zhang. Comparative study of the free-surface boundary condition in two-dimensional finite-difference elastic wave field simulation. *Journal of Geophysics and Engineering*, 8(2):275–286, 2011.

- S. J. Lee, Y. C. Chan, D. Komatitsch, B. S. Huang, and J. Tromp. Effects of realistic surface topography on seismic ground motion in the Yang-minshan region of Taiwan based upon the spectral-element method and LiDAR DTM. *Bulletin of the Seismological Society of America*, 99(2 A):681–693, 2009.
- B. Li, Y. Liu, M. K. Sen, and Z. Ren. Time-space-domain mesh-free finite difference based on least squares for 2D acoustic-wave modeling. *Geophysics*, 82(4):T143–T157, 2017.
- X. Li and G. Yao. An immersed boundary method with iterative symmetric interpolation for irregular surface topography in seismic wavefield modelling. *Journal of Geophysics and Engineering*, 0:1–18, 2020.
- X. Li, G. Yao, F. Niu, D. Wu, and N. Liu. Waveform inversion of seismic first arrivals acquired on irregular surface. *Geophysics*, 87(3):291–304, 2022.
- H. Liu, B. Li, H. Liu, X. Tong, Q. Liu, X. Wang, and W. Liu. The issues of prestack reverse time migration and solutions with Graphic Processing Unit implementation. *Geophysical Prospecting*, 60(5):906–918, 2012.
- S. Liu, X. Li, W. Wang, and Y. Liu. A mixed-grid finite element method with PML absorbing boundary conditions for seismic wave modelling. *Journal of Geophysics and Engineering*, 11(5):055009, 2014.
- Y. Liu and M. K. Sen. A new time – space domain high-order finite-difference method for the acoustic wave equation. *Journal of Computational Physics*, 228(23):8779–8806, 2009.
- Z. Liu, J. Hoffmann, F. J. Simons, and J. Tromp. Elastic full waveform inversion of VSP data from a complex anticline in northern Iraq. In *SEG Technical Program Expanded Abstracts*, volume 2021-Septe, pages 637–641, 2021.
- B. Lombard, J. Piriaux, C. Gelis, and J. Virieux. Free and smooth boundaries in 2-D finite-difference schemes for transient elastic waves. *Geophysical Journal International*, 172(1):252–261, 2008.

- M. Louboutin, P. Witte, and F. J. Herrmann. Effects of wrong adjoints for RTM in TTI media. *SEG Technical Program Expanded Abstracts*, (2): 331–335, 2018.
- M. Louboutin, M. Lange, F. Luporini, N. Kukreja, P. A. Witte, F. J. Herrmann, P. Velesko, and G. J. Gorman. Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration. *Geoscientific Model Development*, 12(3):1165–1187, 2019.
- F. Luporini, M. Louboutin, M. Lange, N. Kukreja, P. Witte, C. Yount, P. H. J. Kelly, F. J. Herrmann, and G. J. Gorman. Architecture and performance of Devito, a system for automated stencil computation. *CoRR*, abs/1807.0, 2019.
- B. Martin, N. Flyer, B. Fornberg, and A. St-cyr. Development of meshless computational algorithms for seismic exploration. In *SEG Annual Meeting*, pages 3543–3547, 2013.
- B. Martin, B. Fornberg, and A. St-cyr. Seismic modeling with radial-basis-function-generated finite differences. *Geophysics*, 80(4):137–146, 2015.
- V. Monteiller, D. Komatitsch, and N. Fuji. A hybrid method to compute short-period synthetic seismograms of teleseismic body waves in a 3-D regional model. *Geophysical Journal International*, 192:230–247, 2013.
- V. Monteiller, D. Komatitsch, and Y. Wang. Three-dimensional full waveform inversion of short-period teleseismic wavefields based upon the SEM-DSM hybrid method. *Geophysical Journal International*, 202:811–827, 2015.
- E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26(9):385–396, 1920.
- P. Mora. Nonlinear elastic inversion of multioffset seismic data. *1986 SEG Annual Meeting, SEG 1986*, 52(9):533–537, 1986.
- F. Muir, J. Dellinger, J. Etgen, and D. Nichols. Modeling elastic fields across irregular boundaries. *Geophysics*, 57(9):1189–1193, 1992.

- W. A. Mulder. A comparison between higher-order finite elements and finite differences for solving the wave equation. In *Proceedings of the Second ECCOMAS Conference on Numerical Methods in Engineering*, number February, pages 344–350, 1996. URL <http://geodus1.ta.tudelft.nl/PrivatePages/W.A.Mulder/Papers/Conferences/ECC096.pdf>.
- W. A. Mulder. A simple finite-difference scheme for handling topography with the second-order wave equation. *Geophysics*, 82(3):111–120, 2017.
- W. A. Mulder and M. J. Huiskes. A simple finite-difference scheme for handling topography with the first-order wave equation. *Geophysical Journal International*, 210(1):482–499, 2017.
- W. A. Mulder and R. Shamasundar. Performance of continuous mass-lumped tetrahedral elements for elastic wave propagation with and without global assembly. *Geophysical Journal International*, 207:414–421, 2016.
- A. Novoselov, F. Fuchs, and G. Bokelmann. Acoustic-to-seismic ground coupling : coupling efficiency and inferring near-surface properties. *Geophysical Journal International*, 223(1):144–160, 2020.
- A. Nuber, E. Manukyan, and H. Maurer. Ground topography effects on near-surface elastic full waveform inversion. *Geophysical Journal International*, 207:67–71, 2016.
- T. Ohminato and B. A. Chouet. A free-surface boundary condition for including 3D topography in the finite-difference method. *Bulletin of the Seismological Society of America*, 87(2):494–515, 1997.
- I. Oprsal and J. Zahradnik. Elastic finite-difference method for irregular grids. *Geophysics*, 64(1):240–250, 1999.
- Y. Park and S. Pyun. Damped wave-equation-based first-arrival traveltime tomography using the embedded boundary method. *Geophysics*, 86(4):U75–U87, 2021.
- N. Peller, A. L. Duc, F. Tremblay, and M. Manhart. High-order stable interpolations for immersed boundary methods. *International Journal for Numerical Methods in Fluids*, 52:1175–1193, 2006.

- J. Pellerin, B. Lévy, G. Caumon, and A. Botella. Automatic surface remeshing of 3D structural models at specified resolution: A method based on Voronoi diagrams. *Computers and Geosciences*, 62:103–116, 2014.
- C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.
- D. Peter, D. Komatitsch, Y. Luo, R. Martin, N. Le Goff, E. Casarotti, P. Le Loher, F. Magnoni, Q. Liu, C. Blitz, T. Nissen-Meyer, P. Basini, and J. Tromp. Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes. *Geophysical Journal International*, 186(2):721–739, 2011.
- N. A. Petersson and B. Sjögreen. Wave propagation in anisotropic elastic materials and curvilinear coordinates using a summation-by-parts finite difference method. *Journal of Computational Physics*, 299:820–841, 2015.
- A. Pica, J. P. Diet, and A. Tarantola. Nonlinear inversion of seismic reflection data in a laterally invariant medium. *Geophysics*, 55(3):284–292, 1990.
- R. E. Plessix and Q. Cao. A parametrization study for surface seismic full waveform inversion in an acoustic vertical transversely isotropic medium. *Geophysical Journal International*, 185(1):539–556, 2011.
- V. Prioux, R. Brossier, Y. Gholami, J. Virieux, O. I. Barkved, and J. H. Kommedal. On the footprint of anisotropy on isotropic full waveform inversion : the Valhall case study. *Geophysical Journal International*, 187(1):1495–1515, 2011.
- R. J. Purser. The Filtering of Meteorological Fields. *Journal of Climate and Applied Meteorology*, 26(12):1764–1769, 1987.
- Y. Qu, Z. Li, J. Huang, J. Li, and Z. Guan. Elastic full-waveform inversion for surface topography. *Geophysics*, 82(5):R269–R285, 2017.
- Y. Qu, Z. Guan, J. Li, and Z. Li. Fluid-solid coupled full-waveform inversion in the curvilinear coordinates for ocean-bottom cable data. *Geophysics*, 85(3):113–133, 2020.

- Y. Rao and Y. Wang. Seismic waveform simulation with pseudo-orthogonal grids for irregular topographic models. *Geophysical Journal International*, 194(3):1778–1788, 2013.
- F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. Mcrae, G.-t. Bercea, G. R. Markall, P. H. J. Kelly, and I. C. London. Firedrake : Automating the Finite Element Method by Composing Abstractions. *ACM Transactions on Mathematical Software*, 43(3):24, 2016.
- E. Reinoso, L. C. Wrobel, and H. Power. Three-dimensional scattering of seismic waves from topographical structures. *Soil Dynamics and Earthquake Engineering*, 16:41–61, 1996.
- J. Ripperger, H. Igel, and J. Wasserman. Seismic wave simulation in the presence of real volcano topography. *Journal of Volcanology and Geothermal Research*, 128:31–44, 2003.
- K. J. Roberts, S. Gioria, and W. J. Pringle. SeismicMesh : Triangular meshing for seismology. *The Journal of Open Source Software*, 6(57):2687, 2021a.
- K. J. Roberts, A. Olender, L. Franceschini, R. C. Kirby, R. S. Gioria, and B. S. Carmo. spyro : a Firedrake-based wave propagation and full waveform inversion finite element solver. *Geoscientific Model Development*, [preprint], 2021b.
- J. O. A. Robertsson. A numerical free-surface condition for elastic/viscoelastic finite-difference modeling in the presence of topography. *Geophysics*, 61(6):1921–1934, 1996.
- A. J. Rodgers, N. A. Petersson, and B. Sjogreen. Simulation of topographic effects on seismic waves from shallow explosions near the North Korean nuclear test site with emphasis on shear wave generation. *Journal of Geophysical Research: Solid Earth*, 115(11), 2010.
- A. Romdhane, G. Grandjean, R. Brossier, F. Réjiba, S. Operto, J. Virieux, A. Romdhane, G. Grandjean, R. Brossier, F. Réjiba, S. Operto, A. Romdhane, G. Grandjeany, R. Brossierx, F. Rejibaz, S. Operto, and J. Virieux. Shallow-structure characterization by 2D elastic full-waveform inversion. *Geophysics*, 76(3):1–13, 2011.

- G. Royle, O. Leblanc, G. Viguier, G. Lambaré, A. Sedova, S. Shutova, and D. Carotti. Acoustic Land Full-Waveform Inversion With Free-Surface Topography in Oman. In *82nd EAGE Conference and Exhibition 2021*, volume 3, pages 1837–1841, 2021.
- H. Ryan. Ricker, Ormsby; Klander, Butterworth -A Choice of wavelets. *CSEG Recorder*, 19(7):24–25, 1994.
- C. A. Schultz. A density-tapering approach for modeling the seismic response of free-surface topography. *Geophysical Research Letters*, 24(22): 2809–2812, 1997.
- J. H. Seo and R. Mittal. A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries. *Journal of Computational Physics*, 230(4):1000–1019, 2011.
- J. Shin, Y. Sim, C. Shin, and H. Calandra. Laplace-domain full waveform inversion using irregular finite elements for complex foothill environments. *Journal of Applied Geophysics*, 96:67–76, 2013.
- J. Shragge. Reverse time migration from topography. *Geophysics*, 79(4): S141–S152, 2013.
- J. Shragge. Solving the 3D acoustic wave equation on generalized structured meshes : A finite-difference time-domain approach. *Geophysics*, 79(6):363–378, 2014.
- J. Shragge. Acoustic wave propagation in tilted transversely isotropic media: Incorporating topography. *Geophysics*, 81(5):C265–C278, 2016.
- J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. CFD Vision 2030 Study : A Path to Revolutionary Computational Aerosciences. Technical Report March 2014, 2014.
- L. Stoltzfus, B. Hamilton, M. Steuwer, L. Li, and C. Dubach. Code Generation for Room Acoustics Simulations with Complex Boundary Conditions. In *IEEE International Parallel and Distributed Processing Symposium*, pages 485–496, 2021.

- X. Sun and Y. Jiang. An immersed boundary computational model for acoustic scattering problems with complex geometries. *Journal of the Acoustical Society of America*, 132(5):3190–3199, 2012.
- Y.-c. Sun, W. Zhang, J.-k. Xu, and X. Chen. Numerical simulation of 2D seismic wave propagation in the presence of a topographic fluid – solid interface at the sea bottom by the curvilinear grid finite-difference method. *Geophysical Journal International*, 210:1721–1738, 2017.
- W. W. Symes and T. Vdovina. Interface error analysis for numerical wave propagation. *Computational Geosciences*, 13:363–371, 2009.
- J. Takekawa and H. Mikada. An adaptive resolution full-waveform inversion using a mesh-free finite-difference method. In *International Symposium on Recent Advances in Exploration Geophysics*, number 2, pages 1–4, 2015.
- J. Takekawa and H. Mikada. A mesh-free finite difference method for frequency-domain visco-acoustic wave equation. In *SEG International Exposition*, number 2, pages 3841–3845, 2016a.
- J. Takekawa and H. Mikada. An absorbing boundary condition for acoustic-wave propagation using a mesh-free method. *Geophysics*, 81(4):4–13, 2016b.
- J. Takekawa and H. Mikada. Free-surface implementation in a mesh-free finite-difference method for elastic wave propagation in the frequency domain. *Geophysical Prospecting*, 67(8):2104–2114, 2019.
- J. Takekawa, H. Mikada, and N. Imamura. A mesh-free method with arbitrary-order accuracy for acoustic wave propagation. *Computers and Geosciences*, 78:15–25, 2015.
- S. Takemura, T. Furumura, and T. Maeda. Scattering of high-frequency seismic waves caused by irregular surface topography and small-scale velocity inhomogeneity. *Geophysical Journal International*, 201:459–474, 2015.
- C. K. W. Tam and J. C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of Computational Physics*, 107(1):262–281, 1993.

- A. Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 1984.
- I. Tarrass, L. Giraud, and P. Thore. New curvilinear scheme for elastic wave propagation in presence of curved topography. *Geophysical Prospecting*, 59(5):889–906, 2011.
- M. Tavelli, M. Dumbser, D. Etienne, L. Rannabauer, T. Weinzierl, and M. Bader. A simple diffuse interface approach on adaptive Cartesian grids for the linear elastic wave equations with complex topography. *Journal of Computational Physics*, 386:158–189, 2019.
- E. Tessmer, D. Kosloff, and A. Behle. Elastic wave propagation simulation in the presence of surface topography. *Geophysical Journal International*, 108(2):621–632, 1992.
- L. Thomsen. Weak elastic anisotropy. *Geophysics*, 51(10):1954–1966, 1986.
- A. Vargas, R. Mittal, and H. Dong. A computational study of the aerodynamic performance of a dragonfly wing section in gliding flight. *Bioinspiration and Biomimetics*, 3:1–13, 2008.
- N. Vieira, H. Alexander, and D. Di. Accurate seabed modeling using finite difference methods. *Computational Geoscience*, 22:469–484, 2018.
- J. Virieux. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51(4):889–901, 1986.
- J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):127–152, 2009.
- D. Wang, C. Chen, D. Zhuang, J. Mei, and P. Wang. Land Fwi: Challenges and Possibilities. In *82nd EAGE Conference and Exhibition 2021*, volume 2, pages 1342–1346, 2021. ISBN 9781713841449. doi: 10.3997/2214-4609.202011175.
- Y. Wang, S. Chevrot, V. Monteiller, D. Komatitsch, F. Mouthereau, G. Manatschal, M. Sylvander, J. Diaz, M. Ruiz, F. Grimaud, S. Benahmed, H. Pauchet, and R. Martin. The deep roots of the western Pyrenees revealed by full waveform inversion of teleseismic P waves. *Geology*, 44(6):475–478, 2016.

- M. Warner, A. Ratcliffe, T. Nangoo, J. Morgan, A. Umpleby, N. Shah, V. Vinje, Š. Ivan, L. Guasch, C. Win, G. Conroy, and A. Bertrand. Anisotropic 3D full-waveform inversion. *Geophysics*, 78(2):59–80, 2013.
- H. Wu, C. Sun, S. Li, J. Tang, and N. Xu. Mesh-free radial-basis-function-generated finite differences and their application in reverse time migration. *Geophysics*, 86(1):S91–S101, 2021.
- H. Wu, C. Sun, S. Li, X. Deng, and Z. Qiao. Mesh-free radial-basis function generated finite differences and its application in full-waveform inversion. *Exploration Geophysics*, 53(2):186–197, 2022.
- L. Yang, H. Yan, and H. Liu. Optimal rotated staggered-grid finite-difference schemes for elastic wave modeling in TTI media. *Journal of Applied Geophysics*, 122:40–52, 2015.
- J. Zahradnik, P. Moczo, and F. Hron. Testing four elastic finite-difference schemes for behaviour at discontinuities. *Bulletin of the Seismological Society of America*, 83(1):107–129, 1993.
- C. Zeng, J. Xia, R. D. Miller, and G. P. Tsoflias. An improved vacuum formulation for 2D finite-difference modeling of Rayleigh waves including surface topography and internal discontinuities. *Geophysics*, 77(1):1–9, 2012.
- D. Zhang, G. Zhan, G. T. Schuster, and K. Abdullah. Multi-source least-squares reverse time migration with topography. In *SEG Technical Program Extended Abstracts*, pages 3736–3740, 2013a.
- W. Zhang and X. Chen. Traction image method for irregular free surface boundaries in finite difference seismic wave simulation. *Geophysical Journal International*, 167:337–353, 2006.
- W. Zhang and Y. Shi. Imaging conditions for elastic reverse time migration. *Geophysics*, 84(2):S95–S111, 2019.
- W. Zhang, Z. Zhang, and X. Chen. Three-dimensional elastic wave numerical modelling in the presence of surface topography by a collocated-grid finite-difference method on curvilinear grids. *Geophysical Journal International*, 190:358–378, 2012.

- Z. Zhang, W. Zhang, H. Li, and X. Chen. Stable discontinuous finite-difference seismic wave modelling. *Geophysical Journal International*, pages 1179–1188, 2013b.
- Z. Zhang, Z. Wu, Z. Wei, J. Mei, R. Huang, and P. Wang. Enhancing salt model resolution and subsalt imaging with elastic FWI. *Leading Edge*, 42(3):207–215, 2023.
- C. Zhao, T. Zhang, and G. X. Hou. Finite-difference time-domain modeling for underwater acoustic scattering applications based on immersed boundary method. *Applied Acoustics*, 193:108764, 2022.
- E. Zhebel, S. Minisini, A. Kononov, and W. A. Mulder. A comparison of continuous mass-lumped finite elements with finite differences for 3-D wave propagation. *Geophysical Prospecting*, 63(1):1111–1125, 2014.