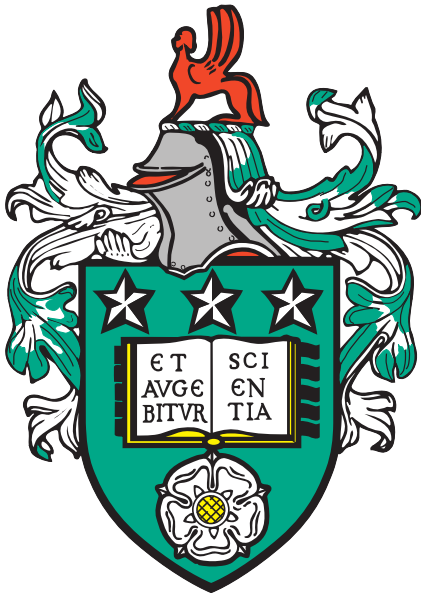
UNIVERSITY OF LEEDS

On the Modelling and Optimisation of a Novel Wave-Energy Device

Jonathan Matthew Bolton

EPSRC Centre for Doctoral Training

in Fluid Dynamics



Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

May 2026

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

- Chapter 3 contains work published in: Bolton, J.M. et al. Towards Optimization of a Wave-to-Wire Energy Device in a Breakwater Contraction. In: Greaves, D. ed. *14th European Wave and Tidal Energy Conference, 5–9 September 2021, Plymouth, UK*. [Online]. Plymouth, UK: University of Plymouth, 2021. [Accessed 25 July 2025]. Available from: URL https://submissions.ewtec.org/proc-ewtec/article/view/ewtec2021-offline-proceedings

  - The candidate carried out the mathematical derivation, prepared the initial manuscript, and presented the work.

  - Other authors contributed the execution of the simulations, supervision of the work and review of the manuscript.

- Section 5.5 and Chapter 6 contain work published in: Bokhove, O. et al. Geometric Power Optimisation of a Rogue-Wave Energy Device in a (Breakwater) Contraction. In: *2024 IEEE Conference on Control Technology and Applications (CCTA), 21–23 August 2024, Newcastle upon Tyne, UK*. [Online]. Piscataway, NJ: IEEE, 2024. [Accessed 23 February 2026]. Available from: DOI 10.1109/CCTA60707.2024.10666543

  - The candidate carried out the initial, single-coil frequency-domain analysis, created the numerical algorithm used to execute the optimisation experiments, and presented the work.

  - Other authors contributed the three-coil idea and analysis, the preparation of the manuscript, and the execution of the simulations.

  - The candidate built upon this work in this thesis, analysing the generic n-coil case and rerunning the numerical experiments for varying α_h.

For from Him and through Him and for Him are all things.

To Him be the glory forever.

Romans 11:36 (NIV)

For I can do everything through Christ, who gives me strength.

Philippians 4:13 (NLT)

Acknowledgements

All honour, glory and praise to God Almighty, without whom this thesis, and all other things, would not be possible. He has been faithful throughout these past 7 years, and will continue to be for the rest of my life and beyond.

To my supervisors, Professor Onno Bokhove, Professor Harvey Thompson and Dr Duncan Borman: Thank-you for your expertise, your input, your patience, and your understanding throughout this project. Thank-you for guiding me through the process, and for convincing me to return to finish the job!

To my colleagues in both the CDT and the School of Maths: Thank-you for sharing your time as a PhD student with me. Thank-you for countless hours spent together in a basement office with no windows, and a brand-new office with a lovely view into the city. Thank-you for sharing your work and listening to me share mine. We made it!

To Professor Rob Sturman and Dr Stephen Griffiths, my undergraduate and Master's supervisors: Thank-you for your own contributions during my 13-year stay at the University of Leeds. I wouldn't have made it to the end if you hadn't helped me navigate the first-half. Also to Dr Claire Savy, Professor Peter Jimack and all at the CDT: Thank-you for your support, and for giving me the opportunity to complete a PhD in the first place. Thank-you also to the EPSRC for funding the research.

To my new boss, Steve: thank-you for putting up with me leaving work at lunchtime every day for the past year because "I've got a book to write". You'll be glad to know the "book" is finished, and I can code for you in the afternoons too now. Fancy reading it?

To Bethan, my constant partner: Thank-you for agreeing to marry a PhD student. Thank-you for your unwavering love, prayers and support; for putting up with the late nights; for letting me quit, and letting me come back. Thank-you for making me dinner and walking our dog. Thank-you in particular for putting your life on pause this past year while I finished writing this thesis. I can't wait for the next phase of our lives.

To Helen and Andy, my loving parents: Thank-you for raising me to be inquisitive, methodical and analytical. Was anyone surprised the son of an accountant and a computer programmer would go on to do a computational maths PhD? Thank-you for letting me be a student for so long, for your continued support during my lengthy time at University. I have a job now! I know you will continue to cheer me on as I tackle more challenges in the future.

To all my friends and family: Thank-you for your prayers and encouragements. Thank-you for helping me get through this PhD: for weekends away and countryside walks; for frisbee and climbing, cycling and running; for pub trips and dinner parties. I promise, I'm now free again to spend time with you!

And finally, to Nyla: woof.

Abstract

This thesis investigates advancements to the mathematical and numerical modelling of a novel wave-energy device, currently in development. The work establishes rigorous theoretical frameworks, develops efficient computational tools, and provides insights to support device design and optimisation.

A fully nonlinear three-dimensional potential flow model is derived from first principles using a Lagrangian framework. This approach provides a rigorous and flexible framework, enabling systematic linearisation, consistent coupling of reduced-dimension hydrodynamic models to device constraints, and the incorporation of additional physical components, such as a capacitor in the electromagnetic generator.

The depth-averaged two-dimensional Benney–Luke model is critically assessed. While offering improved dispersive accuracy over the classical shallow-water approximation, the Benney–Luke formulation is shown to be incompatible with the fundamental coupling between the buoy and free surface, demonstrating that increased model fidelity does not necessarily mean practical suitability for coupled device simulations. The shallow-water model is therefore justified on both computational and physical-consistency grounds.

A flexible computational framework for solving the linearised shallow-water equations is developed, built with the Firedrake finite-element library and incorporating a novel approach for subdomain handling. The resulting algorithm significantly outperforms previous implementations, enabling fast simulations, parameter sweeps and surrogate-based optimisation studies to be carried out in practical time frames.

An analytical model of the buoy–generator subsystem is studied, yielding explicit ex-

pressions for power output which allow direct assessment of design parameters and identification of practical strategies for performance enhancement, including alternative coil configurations and control approaches.

Together, these contributions provide a coherent framework linking theoretical formulation, reduced-order modelling, numerical simulation, and device-level design insights, advancing both the understanding and practical analysis of the device, and supporting its continued development. Future work includes experimental validation, nonlinear and 3D code extensions, exploration of alternative reduced-order models, and the application of advanced control strategies to maximise power output.

Contents

List of Figures

List of Tables

List of Symbols

Latin letters

A	Forcing amplitude, area of domain
A_m	Magnet radius
a	Coil radius
C	Capacitor capacitance
D	Wire diameter
d	Contraction length (diagonal)
F	Buoy-water interface force
G	Magnetic field application function
g	Gravitational acceleration
H	Surface rest position (constrained)
H_0	Free-surface rest position (unconstrained)
H_b	Hull rest position
H_m	Magnet height (above buoy)
h	Free-surface position (unconstrained)
h_b	Hull position
$I, \dot{Q}$	Current

I_{sat}	LED saturation current
$\mathbf{K}$	FE stiffness matrix
K	Short-coil factor
L	Coil length
L_b	Waterline rest position
L_c	Contraction length (horizontal)
L_i	Coil self-inductance
L_m	Magnet length
L_x	Tank width
L_y	Tank length
l_y	Contraction wall position
$\mathbf{M}$	FE mass matrix
M	Buoy mass
m	Magnetic moment
N_{el}	Number of elements (total)
N_x	Number of elements (x-direction)
N_y	Number of elements (y-direction)
n_c	Coil turn number
n_q	LED quality factor
P	Power (instantaneous)
$\hat{P}$	Power (time-averaged)
P_g	Power output
P_l	Power losses
$\mathbf{p}, p$	Generalised momentum

Q	Charge
$\mathbf{q}, q$	Generalised position
$\dot{\mathbf{q}}, \dot{q}$	Generalised velocity
R	Wavemaker position
R_c	Circuit resistance
R_i	Coil resistance
R_l	LED linearisation coefficient
T	Total/final time
t	Time
$\mathbf{u}$	Water velocity
V	LED voltage
V_T	LED thermal voltage
$\mathbf{v}, v$	Unknown in a Lagrangian
v	Test function
$W, \dot{Z}$	Buoy velocity
$\mathbf{x}$	Position vector
x	Horizontal position (widthways)
y	Horizontal position (lengthways)
y_b	Waterline position
Z	Buoy position
Z_0	Buoy rest position
z	Vertical space

Greek letters

α	Hull angle

α_h Coil offset parameter

Γ 1D boundary of 2D domain

Γ_w Stationary tank perimeter (1D)

γ Electromagnet scale parameter

Δt Time-step

Δx Step-size (x-direction)

ΔY Waterline displacement scale parameter

Δy Step-size (y-direction)

ΔZ Buoy displacement scale parameter

$\delta(\cdot)$ Variation of $(\cdot)$

$\delta_{(\cdot)}$ Variation of a functional in $(\cdot)$

$\delta\Omega$ 2D boundary of 3D domain

$\delta\Omega_w$ Stationary tank walls & base (2D)

ϵ BL nonlinearity parameter

ε Variation magnitude

η Free-surface displacement

Θ Heaviside step function

θ Contraction angle

Λ Buoy-water interface rest pressure

λ Buoy-water interface pressure, buoy-water interface condition Lagrange multiplier

μ BL long-wavelength parameter

μ_0 Permeability of free space

ξ Tank-wall impermeability BC Lagrange multiplier

ρ, ρ_0 Water density

σ Wire conductivity

Φ 2D water velocity-potential

ϕ 3D water velocity-potential, conservation of mass Lagrange multiplier

$\boldsymbol{\chi}$ Wavemaker impermeability BC Lagrange multiplier

Ψ Free-surface kinematic BC Lagrange multiplier

ψ FE basis function

Ω 3D domain

Ω_e Element e

ω Sinusoidal forcing frequency

Calligraphic letters

$\mathcal{F}$ Equation generated by a variation

$\mathcal{G}$ Magnetic field application function evaluated at the buoy's rest position, $G(Z_0)$

$\mathcal{H}$ Hamiltonian

$\mathcal{K}$ Kinetic energy

$\mathcal{L}$ Lagrangian

$\mathcal{O}$ Order of approximation

$\mathcal{P}$ Constant density Lagrange multiplier

$\mathcal{R}$ Total effective resistance, $R_i + R_c + R_l$

$\mathcal{S}$ Action

$\mathcal{U}$ Potential energy

Operators

$(\cdot)'$ Derivative of single-variable function $(\cdot)$

$\dot{(\cdot)}$ Time-derivative of $(\cdot)$

$\widetilde{(\cdot)}$ Displacement (from rest) of $(\cdot)$

$\hat{(\cdot)}$	Nondimensional form of $(\cdot)$, FE basis coefficient of $(\cdot)$
$(\cdot)^{(e)}$	Global FE index
$(\cdot)_i^{(e)}$	Local FE index
$(\cdot)_n$	Time-step index
∇	2D or 3D vector differential operator
∇_H	2D horizontal vector differential operator
∇^2	Laplacian operator

List of Abbreviations

AC alternating current

BC boundary condition

BL Benney–Luke (equations / approximation)

DC direct current

FE finite-element

FEM finite element method

GMRES generalized minimal residual (method)

IBP integration-by-parts

LED light-emitting diode

LHS left-hand side (of an equation)

LU lower-upper (decomposition)

OBD oscillating body device

ODE ordinary differential equation

OTC overtopping converter

OWC oscillating water column

PDE partial differential equation

PETSc the Portable, Extensible Toolkit for Scientific Computation

RHS right-hand side (of an equation)

RTT Reynolds transport theorem

SE symplectic Euler (method)

SPD symmetric-positive-definite

SV Störmer–Verlet (method)

SW shallow-water (equations / approximation)

UFL Unified Form Language

WEC wave energy converter

Chapter 1

Introduction

1.1 Wave power as a form of green energy

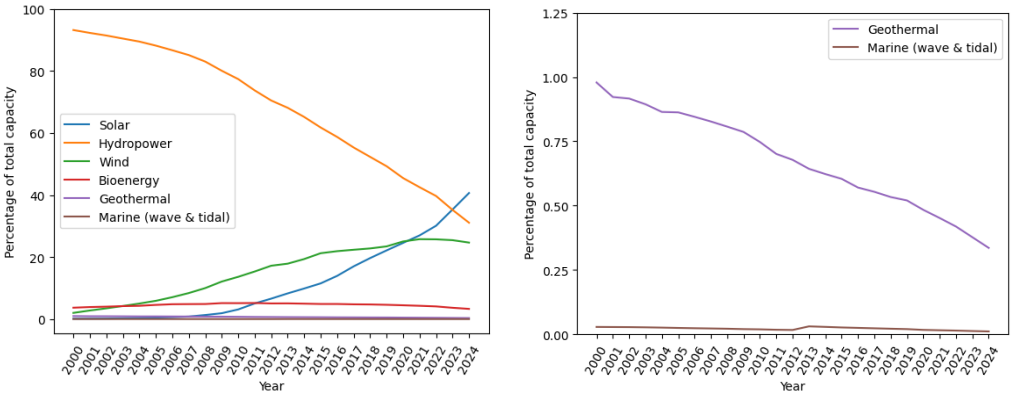
In today's climate-conscious world, the demand for sustainable, renewable energy is ever-increasing. Since 2012, the share of global energy consumption derived from renewable sources has risen each year, from 12.8 %, to 18.0 % in 2024.[1] In the European Union, the share has increased by a factor of 2.6 in the past two decades, from 9.6 % in 2004 to 25.2 % in 2024 [4]. In the USA, the total consumption of renewable energy is forecasted to increase by 3.5 % each year over the next 30 years, 1.1 % higher than the annual increase from all sources [5].

Alongside this growing demand for clean energy, the development of renewable technologies and installed capacity has accelerated. Indeed, 2020 saw the largest year-on-year increase in global renewable electricity capacity additions (that is, the net increase in the maximum amount of electrical power able to be generated from all installed renewable technologies across the world) in the previous 21 years [6], increasing from 220.0 GW in 2019 by about 33 %, to 292.4 GW.[2] The level of annual capacity additions has only continued to rise, with 294.0 GW added in 2021, 347.5 GW in 2022, 563.3 GW in 2023 and 709.0 GW in 2024.[3] The increase of 215.8 GW between 2022 and 2023 greatly exceeded the estimation of the International Energy Agency [8, p. 8], and was the largest absolute

[1] Percentages calculated from Our World in Data [3].
[2] Totals calculated from the International Energy Agency [7].
[3] See footnote 2.

increase ever; while not as big a leap, the huge addition in 2024 suggests this trend is not slowing down any time soon.

The bulk of this current growth in capacity is from solar and wind energy, which, as of 2024, made up 40.6 % and 24.7 % of the total capacity of renewable energy (which was approximately 4,593 GW). Additionally, hydropower contributed 31.1 %, and has historically been by far the most utilised form of renewable energy; as Figure 1.1 shows, 2023 was the first time any other type of renewable overtook the capacity of hydropower. Bioenergy contributed 3.29 %, while 0.34 % came in the form of geothermal power.[4]



(a) The full share.

(b) A zoomed-in version, showing only the two smallest sources.

Figure 1.1: The percentage share of the global renewable electricity capacity for different sources. Data taken from Our World in Data [9].

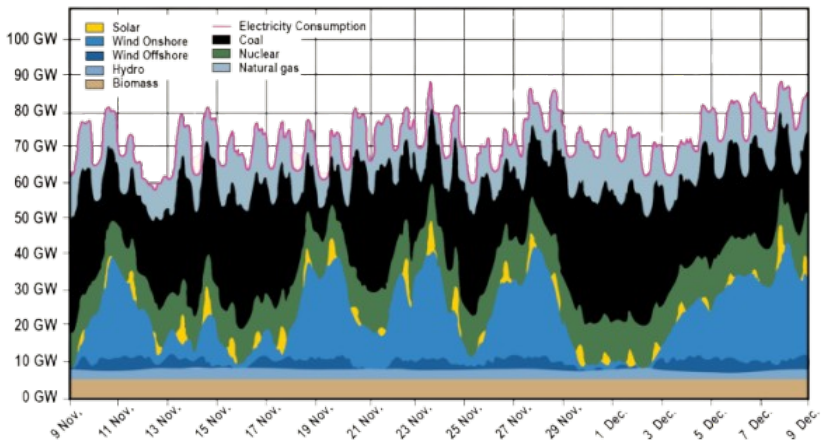
Ocean waves produced by wind action are a major potential source of energy; the theoretical potential of wave energy is estimated to be 8,000–80,000 TW h per year [10, p. 27], or 913.2–9,132 GW, which is 19.9–199 % of the current total capacity of renewables. However, currently only a tiny proportion (494 MW,[5] or about 0.011 %) of our renewable energy comes from the ocean.

Despite the significant lack of utilisation, wave energy has a number of key advantages over the other, more well-known forms of green energy. These include its high reliability and predictability (unlike the wind, which is not as easy to forecast), persistence (unlike the sun, which is only "out" for part of the day), and the fact that waves can travel long distances with very little loss [11, p. 94], meaning accumulations in energy gained over

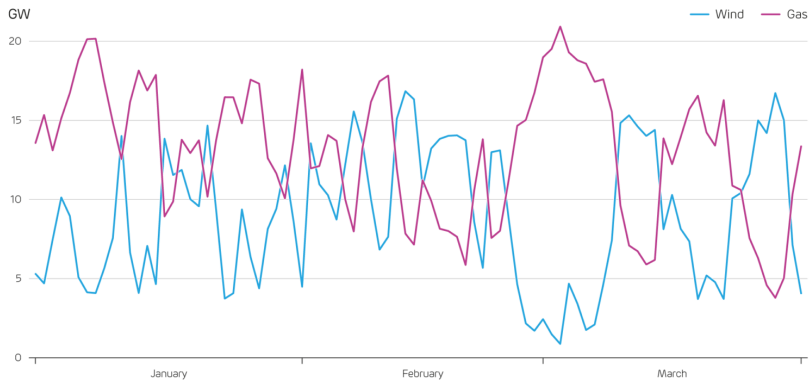
[4] Percentages calculated from the Our World in Data [9].
[5] This figure combines both wave and tidal power, and is also taken from [9].

the large ocean can be harvested on or near the coast.

The German language has a word—*Dunkelflaute* (lit. "dark lull")—which has come to be used by the renewable energy sector to refer to a weather phenomenon known as *anticyclonic gloom*: a period of time characterised by extensive cloud cover and calm conditions. During the extended winter months (October-February), countries across Northern Europe experience on average between 50 and 150 hours of Dunkelflauten per month [12]. The UK experienced a particularly notable event in November 2024, when many weather stations across England and Wales reported no sunshine at all during the first eight days of the month, and when, over a period spanning more than a quarter of the month, only 5 % of the typical November sunshine occurred [13].



(a) Germany, November & December 2017. Taken from [14].



(b) UK, January-March 2021. Taken from [15]

Figure 1.2: A selection of diagrams portraying reduced wind and solar output during Dunkelflauten, and the associated rise in fossil-fuel usage.

During a Dunkelflaute, little-to-no energy can be generated from wind or solar power,

and countries that have begun to rely on these sources for their green energy demands must either import energy from abroad, or revert to burning fossil fuels. Figure 1.2a shows several Dunkelflauten in Germany over November and December 2017 (including a particularly noticeable four-day event); in each case, coal usage rose sharply. Similarly, Figure 1.2b compares the UK's wind and gas usage from January to March 2021 (including during Britain's record-breaking 11-day Dunkelflaute between 26 February and 8 March [15]); here, too, there is a near-perfect inverse relationship between the two sources. More recently, despite wind power becoming the leading contributor to Britain's National Grid in 2024,[6] an event in mid-December saw wind fall to just 6 % of the total supply, with gas stations increasing their output to the highest ever recorded—supplying over 73 % and driving consumer costs sharply upward [16]. Instead of relying on non-renewable sources during these weather events, wave energy could provide a viable alternative for power generation.

In a similar vein, a country like the UK, with a comparatively large amount of coastline, can improve its energy security by investing in wave energy conversion within its own territorial waters. Due to the ongoing Russian invasion of Ukraine, energy security has returned to the public spotlight in recent years, as a significant proportion of Europe's energy came from Russian oil and gas exports. Russia's actions and counter-response to the West's own response to the invasion created panic and uncertainty within the energy market, which subsequently led to a huge rise in wholesale energy prices, impacting both businesses and households in many countries across Europe. Having no control over such a significant economical impact is a severe breach of a country's national security and sovereignty, and reducing the reliance on energy imports, by generating more energy locally, can mitigate these issues in the future.

Furthermore, when ranked by energy quality (that is, the relative ease at which a form of energy can be converted and used as other forms), the mechanical energy of ocean waves outranks solar and wind energy, and is comparable to the geopotential and kinetic energy of river water (the main components of hydropower) [17]. Given these (and other) benefits of wave energy, and the huge untapped potential of the ocean, the development

[6] A first for any type of green energy.

of efficient and cost-effective wave-energy-conversion technologies has an important role to play in meeting the ever-increasing demand for renewable energy.

1.2 Wave energy converters

Unfortunately, extracting energy from ocean waves is complicated by a number of factors, including complex diffraction and radiation phenomena, the need to account for variability in energy availability, and survivability in extreme conditions. The theoretical basis of wave energy conversion, including the governing hydrodynamic principles, has been extensively reviewed by Thomas [18]. A key consideration in many devices is the matching of the resonant frequency of the converter to the frequency of the incoming waves.

The practical complexities associated with wave energy conversion have led to the development of a wide variety of technologies, which rely upon a range of mechanisms for absorbing energy from the waves. Since at least 1981, numerous books and articles reviewing the then-state-of-the-art of existing wave-energy devices, commonly referred to as *wave energy converters* (WECs), have been published (see, for example, [19–29]). In this time, there have been a very large number of proposed WEC designs, which can be broadly grouped into three categories: *oscillating water columns* (OWCs), *overtopping converters* (OTCs) and *oscillating body devices* (OBDs).

1.2.1 Oscillating water columns

An OWC is a type of WEC which captures wave energy via the oscillatory movement of a column of water trapped inside a chamber. The chamber has an underwater opening through which water flows as the waves wax and wane, which subsequently causes the surface of the column to rise and fall. This motion drives airflow in the top part of the chamber, which is directed through a turbine to drive a generator. See Figure 1.3 for a diagram of a typical OWC.

Curiously, the first known use of an OWC (in the late 19th century) was not to generate electricity, but sound; navigational buoys warned boats of dangerous water by using

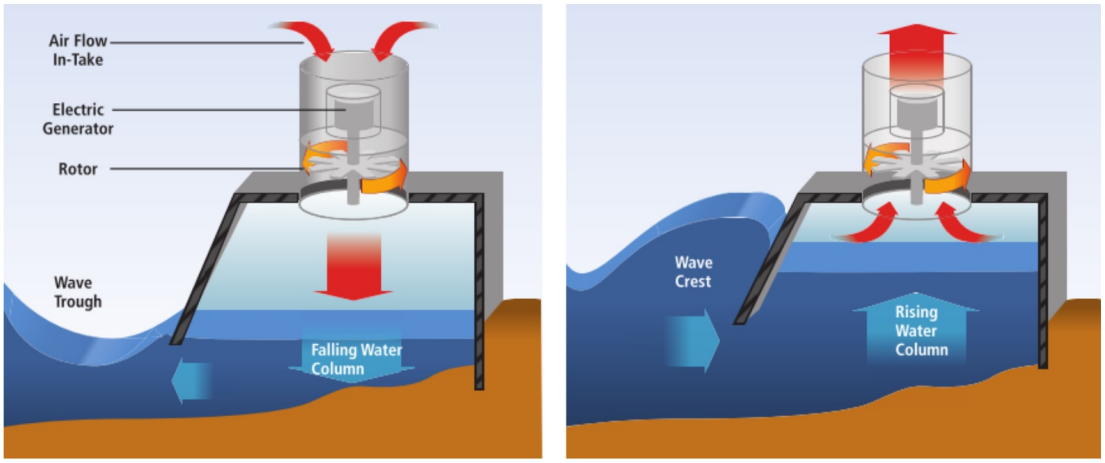


Figure 1.3: A diagram of a typical OWC. Taken from [30, p. 90].

OWCs in which the airflow was directed into a whistle or foghorn [31]. The first OWC to convert the ocean's kinetic energy into electricity (and, indeed, the first electricity-generating wave-energy device of any kind) was developed in Japan in 1947 by Yoshio Masuda, who is widely regarded as the father of modern wave energy conversion. Masuda's device was another navigational buoy, which used the generated electricity to power lights [31].

Other OWCs of note include: the Islay LIMPET [32], which was the world's first commercial WEC and was connected to the UK's national grid system between 2000 and 2012, when it was decommissioned; the Mutriku Breakwater Wave Plant [33], the first multi-turbine breakwater WEC, which has been supplying power to Spain's grid since 2011, and, as of 2020, holds the records for the most electricity produced (over 2 GW h) and the most cumulative operating hours for a wave-energy power plant [34]; and the OE Buoy [35], which is a floating OWC currently in its testing phase.

Due to the oscillatory behaviour of the airflow, the turbine used must be bidirectional; meaning its direction of rotation is the same, regardless of the direction of the airflow. The most commonly used bidirectional turbine is the Wells turbine, which was invented in the 1970s by Professor Allan Wells at Queen's University Belfast specifically for use in OWCs [36]; its blades feature a symmetrical airfoil, meaning they move in the same direction when the air travels from either side of the turbine. In recent years, a number of other turbines which offer alternative benefits and drawbacks have been developed.

A key factor of an OWC's performance is the efficiency of its turbine [25].

1.2.2 Overtopping converters

Where OWCs feature airflow driving a generator, the generators in OTCs are driven by the water itself. In these devices, overtopping waves spill into a reservoir, with the water in the reservoir stored at a higher elevation than the mean ocean surface height. The potential energy difference thus causes the water in the reservoir to flow back into the ocean; directing this flow through a turbine can generate electricity. Because the flow of water will only ever be in one direction, conventional turbines can be used in these devices, which typically offer greater efficiency then the bidirectional turbines used in OWCs.

A key feature of OTCs is a wave-enhancing contraction, often U- or V-shaped. As waves travel down these channels, their horizontal extent gradually decreases, causing their amplitude to increase, which in turn causes more water to spill into the reservoir and ultimately increases the power output. Examples of OTCs include: the TapChan [37], of which a prototype was built in 1985 but was sadly destroyed in a storm; the Wave Dragon [38], the first offshore grid-connected WEC, of which a number have been built since its first prototype in 2003; and the Seawave Slot-Cone Generator [39]. Figure 1.4 shows a diagram of the Wave Dragon OTC.

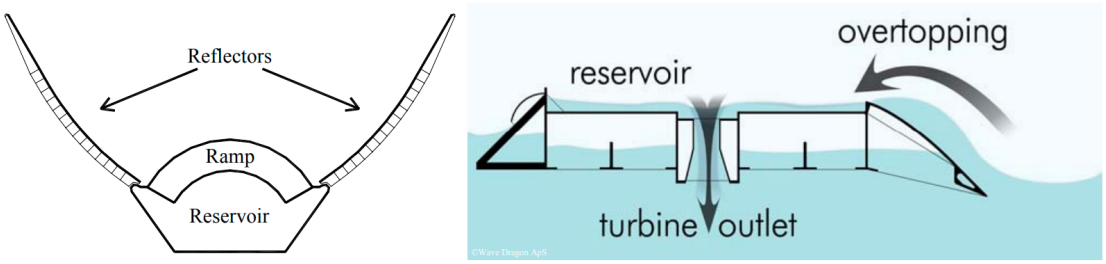


Figure 1.4: A diagram of an OTC (in particular, the Wave Dragon). Taken from [38, p. 1].

1.2.3 Oscillating body devices

While OWCs and OTCs are primarily stationary (with the movement of the air/water driving the generators), an OBD is a device which is directly displaced by the waves. An OBD could consist of a single body (whose motion relative to the water or some fixed

reference drives a generator) or multiple bodies (in which a generator is driven by the relative motion between them), and can be designed to either float on the water's surface, float beneath the surface, or be attached to the ocean floor. Through a combination of vertical forces (buoyancy, gravity and hydrodynamic pressure) and horizontal forces (drag and added mass) produced by the waves, bodies both on and under the surface tend to move along a circular path; though often OBDs are constrained to move only in one direction, be it heaving (translational) or pitching (rotational). See Figure 1.5 for examples of some OBD designs.

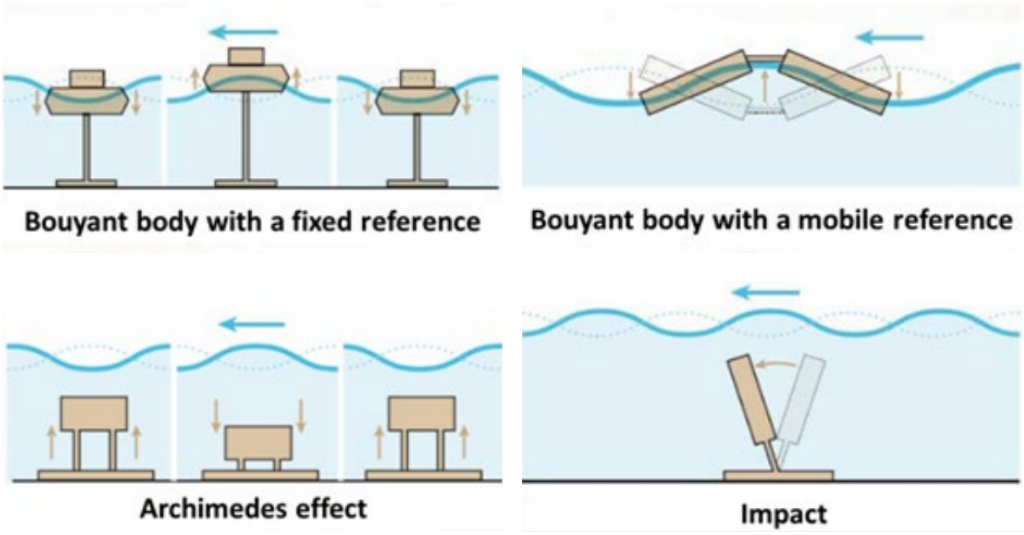


Figure 1.5: Examples of different OBD designs. Taken from [40, p. 64].

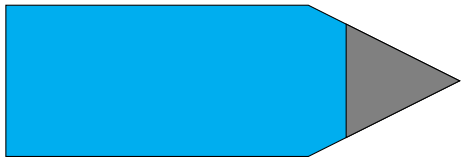
Many OBDs are used to power buoys for navigational or scientific purposes (where the conversion and use of the electricity is self-contained), but they are also viable options for grid-connected wave-farms. Examples include: the Azura [41], a device which captures energy from both the heave (vertical) and surge (horizontal) movement of the waves, and which was the first grid-connected WEC in the United States [42]; and the Wave Star [43], consisting of an array of floats attached to a central beam by arms which pivot as the floats are moved by the waves—this motion is transferred to a generator via hydraulics.

1.3 Our device: description and advantages

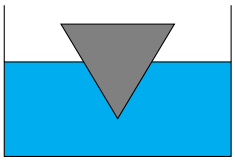
A novel WEC has recently been proposed and developed by Bokhove et al. [44, 45], following their experimental and numerical study of extreme nonlinear water-wave amplification in a contraction. The device consists of:

- an ocean-facing channel featuring a V-shaped contraction, designed to amplify the incoming waves;
- a buoy, placed inside the contraction and constrained to move along a fixed trajectory; and
- a magnet (or a set of multiple magnets) attached to the buoy, travelling through a fixed induction coil as the buoy is displaced by the waves.

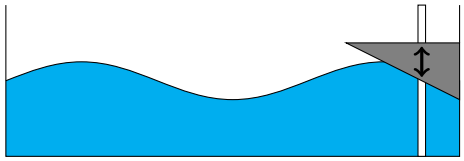
This electromagnetic induction system forms a unidirectional generator, converting the kinetic energy of the waves into electricity via the buoy's motion.



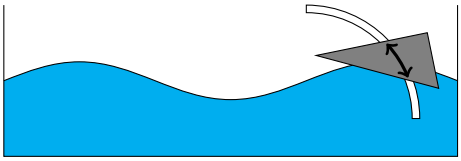
(a) Top view of the tank and buoy, showing how the buoy fits the shape of the contraction.



(b) Front view of the buoy.



(c) Side view, with the buoy constrained to move vertically.



(d) Side view, with the buoy constrained to move along an arc.

Figure 1.6: Sketches of the top (a), front (b) and side views (c, d) of the tank and buoy (the water is shaded blue and the buoy is grey).

Out of the ocean, a tank with a wavemaker is used to generate and contain the waves. Figure 1.6 shows several diagrams which outline the geometry of the tank and buoy. The buoy can be constrained to travel in the vertical (Figure 1.6c) or along an arc (Figure 1.6d), with the latter having two advantages: a rotating axle is more durable than the bearings on a vertical rod, while holding the pivot fixed with the buoy sustained above

the water means it can be taken out of action in a storm. However, the mathematical description of the rotating trajectory is more involved.

The device can be classified as an OBD and has similarities with the IPS Heaving Buoy [46] and the Berkeley Wedge [47], both of which have a comparable unidirectional generator. In addition, the contraction bears resemblance to the TapChan OTC [37], but instead of increasing the amount of water that overtops, here the increased wave-amplitude is intended to exaggerate the motion of the buoy. The device is perhaps best situated in breakwaters and docks, where its construction and maintenance costs will be reduced, and where it can contribute to the normal wave-obstructing function of a breakwater through the damping of waves via the harvesting of their energy. Alternatively, the device could be placed out at sea, as part of a multi-device array (as in [48]), where the breadth of efficient operating conditions can be increased.

In addition to demonstrating a proof-of-concept of the device (seen in the video [49]), Bokhove et al. [44, 45] developed a novel and fully-coupled nonlinear mathematical model of the combined 3D potential-flow hydrodynamics, wave-activated buoy dynamics and electromagnetic power generation. The governing equations were derived using a single, concise Lagrangian,[7] an approach new to the fields of wave and tidal energy and introduced at the European Wave and Tidal Energy Conference in 2019 [45]. By solving simplified forms of these equations, they investigated the effect of induction coils, buoy mass and wavemaker frequency upon the amount of power generated, providing the first study of the resonant behaviour and parameter dependence of the device.

1.4 Thesis outline

The focus of this thesis is on extending the previous work carried out by Bokhove et al. [44, 45], to further develop the analytical and computational tools required to investigate whether the device is feasible for the efficient capture and conversion of wave energy. The thesis outline is as follows:

- Chapter 2 builds upon the mathematical model first presented by Bokhove et al.

[7] See Section 2.1 for a discussion on deriving governing equations from a Lagrangian formulation.

[44, 45]. A full derivation of the fully nonlinear 3D potential-flow governing equations using a Lagrangian built from first-principles is presented, together with a detailed linearisation obtained both from the equations and directly from a "quadricised" Lagrangian.

- Following this, Chapter 3 uses the same Lagrangian framework to investigate a potential improvement to the reduced-dimension shallow-water approximation of the governing equations. This Benney–Luke model is coupled to the buoy and generator, and a numerical investigation is carried out.

- These simulations were undertaken using the same in-house MATLAB code as in [44, 45], employing the finite element method for the spatial discretisation and the symplectic Euler method for time. These numerical methodologies are presented in Chapter 4, which also discusses the use of the external Firedrake finite-element Python library for greater efficiency and simplicity.

- Chapter 5 details the development of a new algorithm, employing Firedrake and intended to replace the MATLAB code used previously. This algorithm is more efficient and offers greater flexibility, and is designed to be used as part of an effective overall numerical optimisation strategy for maximising the device's power output subject to practical operating constraints.

- Finally, Chapter 6 considers an analytical treatment of the buoy-generator subsystem, investigating potential improvements to the electromagnetic induction system and therefore to the performance of the energy conversation of the device.

Chapter 2

Mathematical Fundamentals

The device and its mechanics are modelled via a comprehensive system of partial and ordinary differential equations (PDE/ODEs). By solving these equations numerically, the power output can be calculated, given a prescribed wavemaker motion. The device can be divided into three constituent sub-systems, which are:

- the hydrodynamics, driven by the prescribed motion of the wavemaker;
- the motion of the buoy, which is driven by the movement of the water beneath it; and
- the power generated by the electromagnetic induction system, itself driven by the movement of the buoy.

The first two sub-systems are coupled through the buoy-water interface, which is the region of the water's surface lying beneath, and constrained by, the buoy's hull. Additionally, there is a two-way coupling between the buoy's motion and the current in the coil, with the current applying a (small) magnetic force back onto the buoy. This coupling system is represented by the schematic in Figure 2.1.

In this chapter, the general approach used to derive the device's governing equations (namely, *Lagrangian mechanics*) is first discussed (Section 2.1), before the aforementioned components are mathematically described and the Lagrangian is formed (Section 2.2). A complete derivation of the nonlinear 3D equations is then carried out

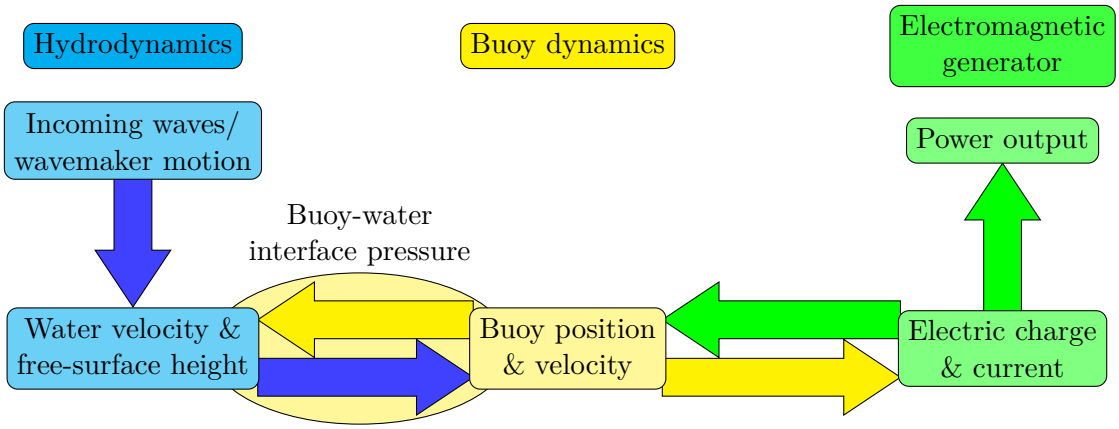


Figure 2.1: A schematic showing the three components of the device, their key quantities and the two-way coupling between them.

(Section 2.3), before the linearisation around the rest state is presented (Section 2.4).

2.1 Lagrangian mechanics

In the present context, the exact mathematical coupling between the hydrodynamics, the buoy's mechanics and the electromagnetic induction system is not immediately obvious. However, the field of *Lagrangian mechanics* allows us to obtain a fully coupled system of equations describing all components of the device. While classical Newtonian mechanics aims to describe the motion of individual components in a system under the influence of the separate forces being applied upon and between them (i.e. Newton's second law $\mathbf{F} = m\mathbf{a}$), Lagrangian mechanics considers the energies of the system as a whole, substituting motion with kinetic energy and force with potential energy. Lanczos [50] provides details on the history, concepts and mathematics of the Lagrangian formulation of mechanical theory, and is highly recommended to the interested reader.

There are two fundamental differences between Newtonian and Lagrangian mechanics:

1. While the motion and forces in Newton's equations have an associated direction (by the use of Euclidean vectors), the energy terms in the Lagrangian formulation are simpler scalar quantities.

2. In Lagrangian mechanics, a "principle" representing the system is formed, from which governing equations can be derived. This contrasts the Newtonian theory, in which the derivation of equations is carried out directly from the system's be-

haviour.[1]

These differences offer Lagrangian mechanics a number of advantages over a Newtonian approach:

- No matter its complexity, the entire system can be initially represented by just one equation (or, more correctly, one principle).
- Each new component or constraint can be introduced simply as an additional term in the principle, rather than by additions to numerous equations or even by additional equations themselves.
- No particular coordinate system is more or less suited than any others (providing the energy can be represented in said coordinates); whatever system is best for the user can be chosen without having to factor in the representation of the forces in that coordinate system. Additionally, no basis-vector transformations are required when changing between coordinate systems.

Essentially, Lagrangian mechanics provides a powerful alternative framework for deriving equations of motion in arbitrary coordinate systems, especially as system complexity grows.[2] However, while the standard formulation naturally handles conservative forces, it is not directly formulated for non-conservative forces (such as friction and electrical resistance), which do not have an associated potential energy.[3] Extensions that incorporate such forces do exist [52, 53], but often, as is the case in the present work, the impact of these forces can simply be introduced a posteriori, by including additional terms to the conservative equations.

This method has been used to derive equations coupling water-waves to ship dynamics [54], wind-turbine elasticity [55, 56], and wavetanks driven by waveflap wavemakers [57, 58]; and has been used alongside nonlinear time-dependent coordinate transformations [59–62], but it is believed this application is the first use of the method in the wave-energy community.

[1] The resulting equations are the same in both formulations, however (at least in the conservative limit, see below).

[2] An excellent example is that of the double pendulum [51].

[3] Non-conservative forces arise from microscopic interactions, and are typically neglected in macroscopic energy depictions.

2.1.1 The principle of least action

The lack of direction associated with a scalar energy presents an obvious and immediate problem: how can one ever know the direction of motion in the system? This question is answered by the *principle of least action*, which states that (as in Figure 2.2) the path $v(t)$ taken by a particle at the point v_1 at time t_1 to reach the point v_2 at time t_2 is the one for which the *action* is minimised. Richard Feynman [63], as part of his published collection of physics lectures, gives an excellent, detailed-yet-accessible description of this idea, outlining the process by which the equations governing the development of the path over time are derived.

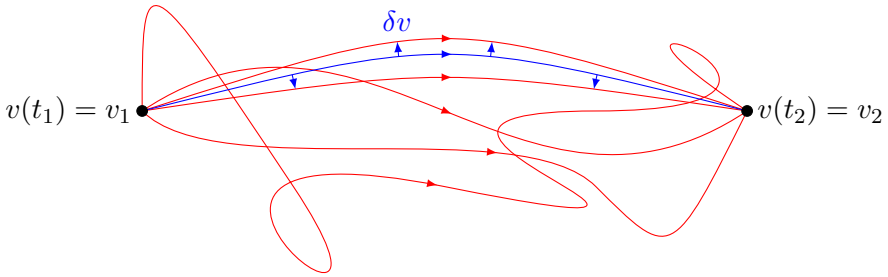


Figure 2.2: Some examples of the possible paths $v(t)$ that a particle might take to get from the point v_1 at time t_1 to the point v_2 at time t_2. The blue path is the one "chosen" by the particle, and is the one for which the action is minimised; the red paths all generate a larger action. For a given path, the deviation from the actual path is δv; the action is minimised when, as the magnitude of this deviation tends towards zero, the ratio between the change in the action and the magnitude of the deviation is also zero.

In classical calculus, minimising a function involves finding the value(s) of the function's coordinate(s) which minimise the function's value. Action, however, is not simply a function, but a *functional*—something which takes as its input a number of functions (dependent variables) rather than a number of coordinates (independent variables), and produces a single value. In the simple example in Figure 2.2, the action depends on the path of the particle, which is itself a function of time. Minimising a functional involves finding the specific function(s) which minimise the value of the functional. This branch of mathematics is known as *the calculus of variations*.

Mathematically, the minimum of some functional $\mathcal{S}[\mathbf{v}(\mathbf{x}, t)]$[4] occurs when its *variation*

[4]Here we expand the example in Figure 2.2, where there is only one unknown time-dependent function $v(t)$, to an arbitrary number of spatio-temporal unknowns $v_i(\mathbf{x}, t)$ contained within the vector $\mathbf{v}$.

$\delta\mathcal{S}$ is zero, i.e.

$$\delta\mathcal{S} \equiv \lim_{\varepsilon \to 0} \left[\frac{\mathcal{S}[\mathbf{v} + \varepsilon\delta\mathbf{v}] - \mathcal{S}[\mathbf{v}]}{\varepsilon} \right] = 0, \tag{2.1}$$

where $\delta\mathbf{v}$ denotes an arbitrary change in $\mathbf{v}$, and ε is the magnitude of such a change. Despite the name, the principle of least action does not specifically require the action to be minimised; finding a stationary point is sufficient—indeed, some authors use the term "principle of *stationary* action" instead.

Now, the action $\mathcal{S}$ is defined as the time-integral of the *Lagrangian* $\mathcal{L}[\mathbf{v}]$, where, in an unconstrained system, the Lagrangian is defined as kinetic energy $\mathcal{K}[\mathbf{v}]$ minus potential energy $\mathcal{U}[\mathbf{v}]$,[5] i.e.

$$\mathcal{S} \equiv \int_{t_1}^{t_2} \mathcal{L}\,\mathrm{d}t, \quad \text{where } \mathcal{L} \equiv \mathcal{K} - \mathcal{U}.^{6} \tag{2.2}$$

For a complex, multi-component system like ours, providing we can find expressions for the kinetic and potential energies of each sub-system, the Lagrangian for the entire device can be easily constructed by taking the sum of the individual Lagrangians.

2.1.2 Constraints and the Lagrange multiplier

The definition of the Lagrangian used in (2.2) is for an *unconstrained* problem, where the solution is free to vary in all dimensions present and has no restrictions placed upon the path it is able to take. However, constraints attached to a problem reduce the number of dimensions in which the solution can vary. For example, a mass sliding down a slope would be constrained to stay on it; the mass cannot pass through or levitate above the slope.

For a problem in p-dimensional space with q constraints, the solution's "p minus q" unknowns must satisfy the q constraint equations as well as the $(p-q)$ governing equations. In a Newtonian framework, the q constraint equations must be explicitly solved to remove q unknowns from the p-dimensional solution, leaving the remaining $(p-q)$ unknowns to be solved for "freely". For example, if a body in 3D space ($p = 3$) is constrained to move

[5] For a nice explanation as to why we take "kinetic minus potential", one could refer to the problem of throwing a ball directly upwards, as explained by Feynman et al. [63, 19.1–19.2].

[6] Note that action is simply a number; the integral removes the time-dependence of the Lagrangian. Recall that $\mathbf{v}$ may also be a function of space; typically, the expressions for the energies $\mathcal{K}$ and $\mathcal{U}$ already have these dependencies integrated out, and $\mathcal{L}$ does not generally depend on space explicitly.

along a circle of radius a and in the x-y plane only ($q = 2$), the solution must satisfy the equations $z = \text{constant}$ and $x^2 + y^2 = a^2$; fixing z and parametrising the motion with a single coordinate ($p - q = 1$)—the angle ϕ satisfying $x = a\cos\phi$ and $y = a\sin\phi$—yields a free problem whose solution automatically satisfies the constraints. In general, however, convenient coordinate transforms do not always exist, and constraint equations can be cumbersome, difficult or even impossible to solve; instead, Lagrangian mechanics offers an easier approach to handling constraints, through the use of *Lagrange multipliers*.

A detailed mathematical explanation of how the Lagrange multiplier approach works is given by Lanczos [50, 43–48]. In practice, each constraint is incorporated by simply adding a term to the unconstrained Lagrangian:

$$\mathcal{L} \equiv \mathcal{K} - \mathcal{U} + \sum_{i=1}^{q} \iiint_{\Omega} \lambda_i f_i \, \mathrm{d}V,$$

where $f_i[\mathbf{v}] = 0$ is the (possibly implicit) i^{th} constraint equation, dependent upon the unknowns $\mathbf{v}$, and $\lambda_i(\mathbf{x}, t)$ is the i^{th} Lagrange multiplier, dependent upon only the co-ordinates $\mathbf{x}$ and t. After making this addition, the augmented system can be treated as unconstrained, with the multipliers becoming additional unknowns to be found. The number of unknowns is therefore increased from $p - q$ to $p + q$, but this is a price worth paying. Since the Lagrangian is a function of time only, any spatial dependencies from the constraint equations must be integrated out.

2.1.3 The Hamiltonian and the Legendre transform

Recall that the Lagrangian is defined as the difference between kinetic and potential energy, $\mathcal{L} \equiv \mathcal{K} - \mathcal{U}$. In general, potential energy typically depends on the positions $\mathbf{q}(t)$ (or some contextual analogy such as charge), while kinetic energy depends on the velocities $\dot{\mathbf{q}}(t)$.[7] There may also be some explicit time-dependence, usually arising from the addition of a time-dependent constraint. Consequently, we have $\mathbf{v} = (\mathbf{q}(t), \dot{\mathbf{q}}(t))$, and the Lagrangian can be written as $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t)$.

There is, of course, an alternative energy measure: the *total energy*, $\mathcal{K} + \mathcal{U}$. This

[7] Here, and throughout the thesis, we use the over-dot shorthand to denote a time derivative, i.e. $\dot{(\cdot)} \equiv \mathrm{d}(\cdot)/\mathrm{d}t$.

is alternatively known as the *Hamiltonian*, and is denoted $\mathcal{H}$. The Lagrangian and Hamiltonian are closely related; one can be derived from the other through a simple manipulation known as the Legendre transform (see e.g. Lanczos [50, 161–166] for an overview):

$$\mathcal{H}(\mathbf{q}, \mathbf{p}, t) = \mathbf{p} \cdot \dot{\mathbf{q}} - \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t), \tag{2.3}$$

where the conjugate momentum $\mathbf{p}$ replaces the velocity $\dot{\mathbf{q}}$ via the identity

$$p_i \equiv \frac{\partial \mathcal{L}}{\partial \dot{q}_i} = \frac{\partial}{\partial \dot{q}_i}(\mathcal{K}(\dot{q}_i) - \mathcal{U}(q_i)) = \frac{\partial \mathcal{K}}{\partial \dot{q}_i},$$

with $\mathbf{q}$ treated as independent of $\dot{\mathbf{q}}$. For a typical kinetic energy of the form $\mathcal{K} = \sum_i m_i \dot{q}_i^2/2$ with m_i some physical constant (e.g. mass), this gives $p_i = m_i \dot{q}_i$, and the Hamiltonian becomes

$$\mathcal{H} = \sum_{i=1}^{n} \left(m_i \dot{q}_i^2 - \frac{1}{2} m_i \dot{q}_i^2 \right) + \mathcal{U} = \sum_{i=1}^{n} \frac{1}{2} m_i \dot{q}_i^2 + \mathcal{U} = \mathcal{K} + \mathcal{U},$$

as expected.

Therefore, given a Lagrangian for a system, we can obtain its total energy as a function of its unknowns by applying (2.3). We can also obtain the energy balance of the system by computing $\partial \mathcal{H}/\partial t$; Hamiltonians with no explicit time-dependence have an energy balance of 0 (energy is conserved), while explicit time-dependence arising from constraints provides an external addition/reduction to the energy balance over time.

2.1.4 Equation derivation process

Writing the principle of least action (2.1) in terms of $\mathcal{L}$, the (total) variation $\delta \mathcal{S}$ is given by

$$\delta \mathcal{S} \equiv \lim_{\varepsilon \to 0} \left[\frac{1}{\varepsilon} \int_{t_1}^{t_2} \{ \mathcal{L}[\mathbf{v} + \varepsilon \delta \mathbf{v}] - \mathcal{L}[\mathbf{v}] \} \, \mathrm{d}t \right].$$

Now, instead of allowing all unknowns to deviate simultaneously, let us change just a single unknown v_i, such that $\delta \mathbf{v}$ becomes $\delta \mathbf{v}_i \equiv (0, \ldots, \delta v_i, \ldots, 0)$. Then, trivially,

$\delta\mathbf{v} = \sum_i \delta\mathbf{v}_i$. Denoting the variation under this change as $\delta_{v_i}\mathcal{S}$, such that

$$\delta_{v_i}\mathcal{S} \equiv \lim_{\varepsilon\to 0}\left[\frac{1}{\varepsilon}\int_{t_1}^{t_2}\{\mathcal{L}[\mathbf{v}+\varepsilon\delta\mathbf{v}_i]-\mathcal{L}[\mathbf{v}]\}\,\mathrm{d}t\right], \tag{2.4}$$

it then follows from the linearity of $\delta\mathcal{S}$ in $\delta\mathbf{v}$ that $\delta\mathcal{S} = \sum_i \delta_{v_i}\mathcal{S}$; in other words, the variation when allowing all unknowns $\mathbf{v}$ to change together is equal to the sum of the variations when each unknown v_i is changed individually.[8]

Now, for each variation, if one were able to obtain a expression of the form

$$\delta_{v_i}\mathcal{S} = \int_{t_1}^{t_2}\delta v_i\mathcal{F}_i\,\mathrm{d}t \tag{2.5}$$

for some functional $\mathcal{F}_i[\mathbf{v}]$, then (2.1) could be written as

$$\int_{t_1}^{t_2}\sum_i \delta v_i\mathcal{F}_i\,\mathrm{d}t = 0. \tag{2.6}$$

Then, since each δv_i represents some arbitrary change to the unknown v_i, and (2.6) must hold for *any* δv_i, we must have

$$\mathcal{F}_i = 0 \text{ for all } i. \tag{2.7}$$

The equations given by (2.7) are precisely those that govern the system described by the Lagrangian $\mathcal{L}$.

In summary, the process for deriving the governing equations for the system is to take the variation in each of the system's unknowns in turn, manipulating the resulting integrals into the form of (2.5), and writing down the resulting $\mathcal{F}_i = 0$. In practice, obtaining this form often requires derivatives of δv_i to be transferred via integration-by-parts (IBP), which for time derivatives results in evaluations of δv_i at t_1 and t_2. To handle these, we impose the *endpoint conditions*

$$\delta v_i|_{t_1} = \delta v_i|_{t_2} = 0 \text{ for all } i. \tag{2.8}$$

These conditions arise from considering the question that the principle of least action

[8] This is the same principle that allows a multivariable derivative to be expressed as a sum of its partial derivatives.

aims to answer: among all possible paths *connecting two fixed points in time*, which path minimises the action? At intermediate times, the path is free to vary ($\delta\mathbf{v} \neq 0$); only at the endpoints is the path constrained ($\delta\mathbf{v} = 0$).

For initial-value problems, such as those considered in this thesis, it may seem confusing that the solution at a future time is "fixed" when it is not yet known. In fact, knowledge of this future solution is not required; the only requirement is that the paths considered share the same endpoints (which could, in principle, be any arbitrary times). To clarify this distinction, we use t_1 and t_2 to denote these fixed times used in the derivation of the equations, and $t = 0$ and $t = T$ to denote the start and end of the initial-value problem. Simply put: *Between any two arbitrary fixed points in time (t_1 and t_2), how does the solution evolve?* This is what Lagrangian mechanics answers, by providing the equations that govern the solution at all times; *Given an initial solution at $t = 0$, what is the solution at time $t = T$?* This is what solving the equations answers.

2.1.5 Example: throwing a ball up into the air

To illustrate the above process, consider the example used by Feynman et al. [63]: that of throwing a ball up into the air in the presence of a constant gravitational acceleration g. The ball has mass m and its unknown position is given by $z(t)$, such that its kinetic energy is $m\dot{z}^2/2$ and its potential energy is mgz, and the Lagrangian is

$$\mathcal{L}(z, \dot{z}) = \frac{1}{2}m\dot{z}^2 - mgz. \tag{2.9}$$

The conjugate momentum $p(t)$ is

$$p \equiv \frac{\partial \mathcal{L}}{\partial \dot{z}} = \frac{\partial}{\partial \dot{z}}\left(\frac{1}{2}m\dot{z}^2 - mgz\right) = m\dot{z},$$

leading to the relationship $\dot{z} = p/m$. Thus, the Hamiltonian is

$$\begin{aligned}\mathcal{H}(z, p) &= p\dot{z} - \mathcal{L}(z, \dot{z}) \\ &= \frac{p^2}{m} - \frac{1}{2}m\left(\frac{p}{m}\right)^2 + mgz \\ &= \frac{p^2}{2m} + mgz.\end{aligned}$$

Note that $\partial\mathcal{H}/\partial t = 0$, and energy is conserved. Writing the momentum p in terms of velocity w ($p \equiv mw$) gives the more familiar

$$\mathcal{H} = \frac{1}{2}mw^2 + mgz.$$

Using (2.9) in (2.4), the variation $\delta_z\mathcal{S}$ is given by

$$\begin{aligned}
\delta_z\mathcal{S} &= \lim_{\varepsilon\to 0}\left[\frac{1}{\varepsilon}\int_{t_1}^{t_2}\left\{\frac{m}{2}\left(\frac{\mathrm{d}}{\mathrm{d}t}(z+\varepsilon\delta z)\right)^2 - mg(z+\varepsilon\delta z) - \frac{1}{2}m\dot{z}^2 + mgz\right\}\mathrm{d}t\right] \\
&= \lim_{\varepsilon\to 0}\left[\frac{1}{\varepsilon}\int_{t_1}^{t_2}\left\{\cancel{\frac{1}{2}m\dot{z}^2} + m\dot{z}\varepsilon\delta\dot{z} + \frac{1}{2}m\varepsilon^2\delta\dot{z}^2 - \cancel{mgz} - mg\varepsilon\delta z - \cancel{\frac{1}{2}m\dot{z}^2} + \cancel{mgz}\right\}\mathrm{d}t\right] \\
&= \lim_{\varepsilon\to 0}\left[\int_{t_1}^{t_2}\left\{m\dot{z}\delta\dot{z} + \cancel{\frac{1}{2}m\varepsilon\delta\dot{z}^2} - mg\delta z\right\}\mathrm{d}t\right] \\
&= \int_{t_1}^{t_2}\{m\dot{z}\delta\dot{z} - mg\delta z\}\,\mathrm{d}t.
\end{aligned}$$

This results in a term containing the derivative $\delta\dot{z}$, so to obtain the desired form (2.5) we need to use IBP to transfer it. Recalling the endpoint conditions (2.8), we have

$$\int_{t_1}^{t_2} m\dot{z}\delta\dot{z}\,\mathrm{d}t = \cancel{[m\dot{z}\delta z]_{t_1}^{t_2}} - \int_{t_1}^{t_2} m\ddot{z}\delta z\,\mathrm{d}t,$$

giving

$$\delta\mathcal{S} = \int_{t_1}^{t_2}\{-m\ddot{z}\delta z - mg\delta z\}\,\mathrm{d}t = -m\int_{t_1}^{t_2}\delta z[\ddot{z}+g]\,\mathrm{d}t = 0.$$

The term in square brackets corresponds to the $\mathcal{F}_i$ in (2.6), and the only way that $\delta\mathcal{S}$ can be zero for any δz is if $\ddot{z} = -g$. That is, the acceleration of the ball $\ddot{z}$ is equal to the negative of the acceleration due to gravity, which is precisely what Newton's equation $\mathbf{F} = m\mathbf{a}$ would have given for this problem.[9]

2.2 Initial mathematical description

Let us now outline the mathematical description of the device, including its geometry and associated parameters and variables. The notation used is summarised in Table 2.1.

[9] Neglecting air resistance, of course, which is a non-conservative force.

Table 2.1: The various notations associated with the mathematical description of the device.

Notation	Name	Unit
	Spatio-temporal coordinates	
x	Horizontal position (widthways)	m
y	Horizontal position (lengthways)	m
z	Vertical position	m
t	Time	s
$\partial\Omega_w$	Stationary tank walls/base (2D)	m^2
Γ_w	Stationary tank perimeter (1D)	m
	Tank geometry	
L_x	Tank width	m
L_y	Tank length	m
L_c	Contraction length (horizontal)	m
d	Contraction length (diagonal)	m
$l_y(x)$	Contraction wall position	m
θ	Contraction angle	$^\circ$ or rad
	Hydrodynamics	
$\mathbf{u}(x,y,z,t)$	Water velocity	$\mathrm{m\,s^{-1}}$
$\phi(x,y,z,t)$	Water velocity potential[10]	$\mathrm{m^2\,s^{-1}}$
$\rho_0, \rho(x,y,z,t)$	Water density	$\mathrm{kg\,m^{-3}}$
$R(t)$	Wavemaker position	m
$h(x,y,t)$	Free-surface position	m
H_0	Free-surface rest position	m
$\eta(x,y,t)$	Free-surface displacement (from rest)	m
g	Gravitational acceleration	$\mathrm{m\,s^{-2}}$
$\mathcal{P}(x,y,z,t)$	Constant density multiplier	$\mathrm{m^2\,s^{-2}}$
$\boldsymbol{\chi}(x,y,z,t)$	Wavemaker impermeability BC multiplier	$\mathrm{kg\,m^{-1}\,s^{-1}}$

(continued on next page)

Table 2.1 (continued)

Notation	Name	Unit
$\xi(x,y,z,t)$	Tank-wall impermeability BC multiplier	$\mathrm{kg\,m^{-1}\,s^{-1}}$
$\Psi(x,y,z,t)$	Free-surface kinematic BC multiplier	$\mathrm{kg\,m^{-1}\,s^{-1}}$
	Buoy	
M	Buoy (plus mast and magnet) mass	kg
$Z(t)$	Buoy position	m
Z_0	Buoy rest position	m
$W(t)$	Buoy velocity	$\mathrm{m\,s^{-1}}$
α	Hull angle	$^\circ$ or rad
$h_b(y,t)$	Hull position	m
$H_b(y)$	Hull rest position	m
$y_b(x,t)$	Waterline position	m
L_b	Waterline rest position	m
$\Theta(y)$	Heaviside step function	-
$\lambda(x,y,t)$	Buoy-water interface multiplier	$\mathrm{m^2\,s^{-2}}$
	Electromagnetic induction system	
L_m	Magnet length	m
A_m	Magnet radius	m
H_m	Magnet height (above buoy)	m
m	Magnetic moment	$\mathrm{A\,m^2}$
L	Coil length	m
a	Coil radius	m
α_h	Offset parameter	-
D	Wire diameter	m
σ	Wire conductivity	$\Omega^{-1}\,\mathrm{m^{-1}} = \mathrm{kg^{-1}\,m^{-3}\,s^3\,A^2}$
n_c	Coil turn number	-
L_i	Coil self-inductance	$\mathrm{H} = \mathrm{kg\,m^2\,s^{-2}\,A^{-2}}$

(continued on next page)

Table 2.1 (continued)

Notation	Name	Unit
K	Short-coil factor	-
R_i	Coil resistance	$\Omega = \mathrm{kg\,m^2\,s^{-3}\,A^{-2}}$
R_c	Circuit resistance	$\Omega = \mathrm{kg\,m^2\,s^{-3}\,A^{-2}}$
γ	Electromagnet scale parameter	$\mathrm{Wb\,m^2} = \mathrm{kg\,m^4\,s^{-2}\,A^{-1}}$
$G(Z)$	Magnetic field application function	$\mathrm{m^{-3}}$
μ_0	Permeability of free space	$\mathrm{H\,m^{-1}} = \mathrm{kg\,m\,s^{-2}\,A^{-2}}$
C	Capacitor capacitance	$\mathrm{F} = \mathrm{kg^{-1}\,m^{-2}\,s^4\,A^2}$
n_q	LED quality factor	-
V_T	LED thermal voltage	$\mathrm{V} = \mathrm{kg\,m^2\,s^{-3}\,A^{-1}}$
I_{sat}	LED saturation current	A
$I(t)$	Current	A
$Q(t)$	Charge	$\mathrm{C} = \mathrm{A\,s}$
$P(t)$	Power	$\mathrm{W} = \mathrm{kg\,m^2\,s^{-3}}$

2.2.1 Tank and buoy

Figure 2.3 is a partial reproduction of Figure 1.6, with the addition of relevant mathematical notation. The tank has width L_x and length L_y. The V-shaped wall is at an angle of θ to the xz-plane, and is described by

$$l_y(x) \equiv L_y - L_c \left| \frac{2x}{L_x} - 1 \right|,$$

where L_c is the (horizontal) length of the contraction. Furthermore, the diagonal length of the contraction is d. Given L_x, any one of θ, L_c or d uniquely describes the contraction geometry, while the other two can be calculated:

$$d : L_c = \sqrt{d^2 - \frac{L_x^2}{4}}, \quad \theta = \cos^{-1}\left(\frac{L_x}{2d}\right),$$

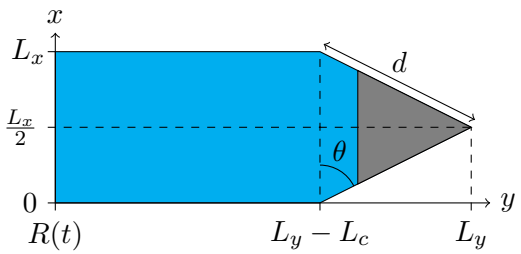
[10] This variable is introduced as the Lagrange multiplier enforcing conservation of mass; however, subsequent variation of the Lagrangian with respect to velocity $\mathbf{u}$ shows that this multiplier acts as a velocity potential.

$$L_c : d = \sqrt{L_c^2 + \frac{L_x^2}{4}}, \quad \theta = \tan^{-1}\left(\frac{2L_c}{L_x}\right),$$
$$\theta : d = \frac{L_x}{2\cos\theta}, \quad L_c = \frac{L_x}{2}\tan\theta.$$

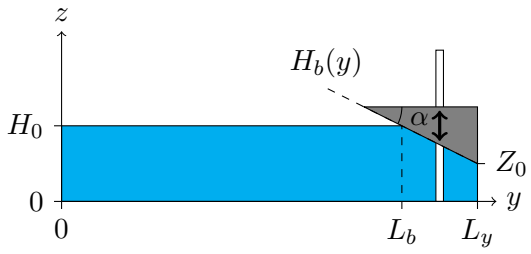
The water in the tank has velocity $\mathbf{u}(x, y, z, t)$ and constant density $\rho_0 \approx 997\,\mathrm{kg\,m^{-3}}$, and fills the region

$$\Omega = \left\{(x, y, z) \in \mathbb{R}^3 \;\middle|\; x \in [0, L_x],\;\; y \in [R(t), l_y(x)],\;\; z \in [0, h(x, y, t)]\right\},$$

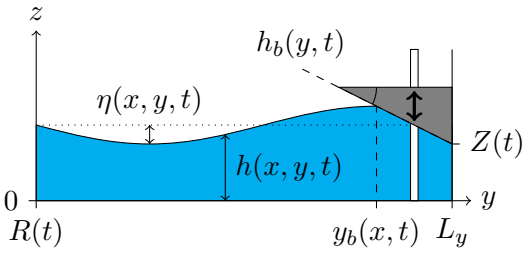
where $y = R(t)$ is the (known) position of the wavemaker and $z = h(x, y, t)$ is the variable position of the water's (single-valued)[11] free surface. The wavemaker's velocity is $\dot{R}(t)$.



(a) Top view of the tank and buoy, outlining the tank's dimensions and how the buoy fits the shape of the contraction.



(b) Side view at rest ($t = 0$), with the buoy constrained to move vertically.



(c) Side view at time t, with the buoy constrained to move vertically.

Figure 2.3: A reproduction of Figures 1.6a and 1.6c, together with a side view at rest, annotated with the relevant mathematical notation.

The buoy is a tetrahedron, with: a flat top, parallel to the horizontal xy-plane; two sides, perpendicular to the top, that fit along the walls of the contraction; and a slanted hull at an angle α to the top. Its mass, including an attached mast and magnet, is M.

[11] Note that this modelling approach requires a single value of h for any given (x, y, t), and therefore it cannot be used to model overtopping waves.

Due to added complexities with the mathematical description of the rotating trajectory, only vertical motion is considered presently. The buoy's position along this trajectory is $z = Z(t)$ and is measured to be where its bottom vertex lies (as in Figure 2.3c), with $W(t) \equiv \dot{Z}$ its velocity. The position of the hull is given by $z = h_b(y,t)$, and the waterline (defined as where the free surface meets the hull) is at $y = y_b(x,t)$. Beyond this, for $y \geq y_b$, the surface of the water is constrained to match the hull's shape, i.e

$$h = h_b \text{ for } y \geq y_b, \text{ or } (h_b - h)\Theta(y - y_b) = 0, \tag{2.10}$$

where $\Theta(y - y_b)$ is the Heaviside step function, defined to be zero for $y < y_b$ and one for $y \geq y_b$. As the buoy's motion is vertical, the angle of the hull to the horizontal remains the same at all times, so

$$h_b(y,t) \equiv h_b(y, Z(t)) \equiv Z + (L_y - y)\tan\alpha.$$

Initially (at $t = 0$), the water, wavemaker and buoy are all at rest, so $\mathbf{u}|_{t=0} = \dot{R}(0) = W(0) = 0$. Additionally, the (unconstrained) free-surface height is H_0, the wavemaker is at $R(0) = 0$, the buoy rests at $Z(0) = Z_0$ and the waterline is at $y_b(x,0) = L_b$. The at-rest position of the hull is therefore

$$h_b|_{t=0} \equiv h_b|_{Z=Z_0} \equiv H_b(y) \equiv Z_0 + (L_y - y)\tan\alpha,$$

and the at-rest position of the surface can be written as

$$h|_{t=0} \equiv H(y) \equiv \begin{cases} H_0 & y < L_b, \\ H_b(y) & y \geq L_b. \end{cases} \tag{2.11}$$

A final useful measure is the free-surface displacement $\eta(x,y,t)$, defined as the difference between the surface's position at time t and its rest-position, i.e. $\eta \equiv h - H$.

At the waterline, the free-surface height must equal the hull's position, such that we have

$$h|_{y=y_b^-} = Z + (L_y - y_b^+)\tan\alpha.$$

Taking the variation of this expression leads to

$$\begin{aligned} \delta h|_{y=y_b^-} + \left.\frac{\partial h}{\partial y}\right|_{y=y_b^-} \delta y_b &= \delta Z - \delta y_b \tan\alpha \\ \Rightarrow \delta y_b &= \frac{\delta Z - \delta h|_{y=y_b^-}}{\frac{\partial h}{\partial y}|_{y=y_b^-} + \tan\alpha}, \end{aligned} \tag{2.12}$$

implying y_b is not an independent variable in the problem. This means it is not free to vary, and we need not include it in the vector of unknowns $\mathbf{v}$ when taking variations. Similarly, since the wavemaker position R is prescribed, is it also not included in $\mathbf{v}$.

Additionally, H must be continuous at the at-rest waterline $y = L_b$, such that $H_0 = H_b(L_b)$, and L_b must satisfy

$$L_b = L_y - \frac{H_0 - Z_0}{\tan\alpha}. \tag{2.13}$$

By considering the portion of the buoy that is partially submerged, and calculating its volume at rest, the unknown at-rest position Z_0 can be written in terms of known parameters via Archimedes' principle. Referring to Appendix A.1 for details, we have

$$Z_0 = H_0 - \left(\frac{3M\tan^2\alpha\tan\theta}{\rho_0}\right)^{1/3},$$

which when used in (2.13) gives

$$L_b = L_y - \left(\frac{3M\tan\theta}{\rho_0\tan\alpha}\right)^{1/3}.$$

The Lagrangian for the hydrodynamics is

$$\mathcal{L}_h = \rho_0 \int_0^{L_x}\int_R^{l_y}\int_0^h \left\{\frac{1}{2}\|\mathbf{u}\|^2 - g(z - H_0)\right\} \mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x, \tag{2.14}$$

noting that mass can be written as an integral over the volume multiplied by the density ρ_0. The constant $g \approx 9.81\,\mathrm{m\,s^{-2}}$ is gravitational acceleration. For the buoy, the Lagrangian is

$$\mathcal{L}_b = \frac{1}{2}M\dot{Z}^2 - MgZ. \tag{2.15}$$

Note that the hydrodynamic potential energy

$$\rho_0 \int_0^{L_x} \int_R^{l_y} \int_0^h gz \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x$$

is normalised by subtracting the background hydrostatic contribution

$$\rho_0 \int_0^{L_x} \int_R^{l_y} \int_0^h gH_0 \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x,$$

leaving only the excess potential energy associated with the free-surface displacement η. This ensures that the resulting free-surface condition is expressed in terms of deviations from hydrostatic equilibrium.[12] A similar normalisation is not applied to the buoy, however, because its gravitational energy does not contain an analogous removable background contribution; its equilibrium arises entirely through interaction with the fluid.

The buoy-water interface constraint (2.10) can be included in the Lagrangian by imposing it via the Lagrange multiplier λ (multiplied by ρ_0). A boundary condition (BC) on λ at the waterline can be later derived by initially allowing the density of the water to vary—i.e. by using $\rho(x, y, z, t)$ in place of ρ_0—and then including $\rho = \rho_0$ as a further constraint imposed by the multiplier $\mathcal{P}$. Additionally, constraints applying the impermeability BC on $\mathbf{u}$ (at the wavemaker and along the walls), the kinematic BC on $\mathbf{u}$ and h at the free surface and the incompressibility of the water (relaxed to conservation of mass due to the weakly-imposed constant density) are imposed by $\boldsymbol{\chi}$, ξ, Ψ and ϕ, respectively. The Lagrangian for these constraints is

$$\begin{aligned}\mathcal{L}_c = \int_0^{L_x} \Bigg[& \int_R^{l_y} \int_0^h \left\{ \mathcal{P}(\rho_0 - \rho) + \phi \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) \right\} \mathrm{d}z \, \mathrm{d}y \\ & + \left\{ \int_0^h \boldsymbol{\chi} \cdot \left(\mathbf{u} - \dot{R}\hat{\mathbf{y}} \right) \mathrm{d}z \right\} \Bigg|_{y=R} \\ & + \int_R^{l_y} \left\{ \Psi \left(\frac{\partial h}{\partial t} + \mathbf{u}|_{z=h} \cdot (\nabla_H h, -1) \right) + \rho_0 \lambda (h_b - h) \Theta(y - y_b) \right\} \mathrm{d}y \Bigg] \, \mathrm{d}x\end{aligned}$$

[12] One may ask why we subtract H_0 and not the constrained value H; this is because H is a geometric constraint imposed by the coupling with the buoy, while H_0 is the true hydrostatic equilibrium level of the uncoupled fluid.

$$+ \iint_{\partial\Omega_w} \xi \mathbf{u} \cdot \hat{\mathbf{n}} \, \mathrm{d}S, \tag{2.16}$$

where $\partial\Omega_w = \partial\Omega \setminus (\{y = R\} \cup \{z = h\})$ denotes the five stationary boundaries of the tank (the four stationary walls and the base). Note that the units of the multipliers are given in Table 2.1; since the Lagrangian has units of energy, the units of the multipliers can be derived accordingly.

2.2.2 Magnet and coil

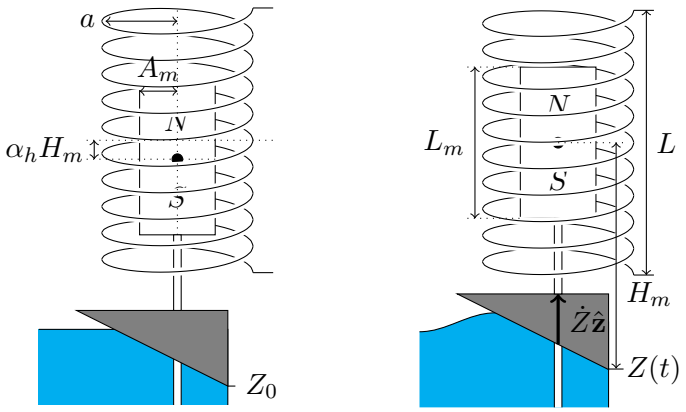


Figure 2.4: The geometry of the magnet and coil. The left figure is at rest, outlining the radii of the magnet and coil, and the offset $\alpha_h H_m$. The right figure is at time t and outlines the lengths of the magnet and coil, the height of the mast and the direction of the buoy's (and magnet's) velocity.

Figure 2.4 shows the geometry of the magnet and coil. A mast is fixed onto the top of the buoy, with a cylindrical magnet (or a set of multiple magnets) of (total) magnetic moment m, (total) length L_m and radius A_m attached to it. The magnet is aligned vertically and spans the time-dependent region

$$z \in \left[Z(t) + H_m - \frac{L_m}{2}, Z(t) + H_m + \frac{L_m}{2} \right],$$

with H_m the distance between the bottom of the buoy and the magnet's centre. The induction coil, of length L and radius $a > A_m$, is also aligned vertically and is coaxial to the magnet, but is fixed at such a height that its centre is offset from the magnet's

centre when the system is at rest. That is, the coil spans the fixed region

$$z \in \left[Z_0 + (1+\alpha_h)H_m - \frac{L}{2}, Z_0 + (1+\alpha_h)H_m + \frac{L}{2} \right],$$

with $\alpha_h H_m$ the offset and $\alpha_h \neq 0$. This offset ensures that, upon linearising, the buoy and induction system remain coupled (see Section 2.4).

The coil is made up of copper wire with diameter D and electrical conductivity $\sigma = 5.96 \times 10^7\,\Omega^{-1}\,\mathrm{m}^{-1}$ [64], and has $n_c = L/D$ turns. The length of wire used in one turn is $2\pi a$, such that the entire length of wire is $2\pi a n_c$ and the resistance of the coil is

$$R_i \equiv \frac{\text{length of wire}}{\text{conductivity} \times \text{cross-sectional area of wire}} = \frac{2\pi a n_c}{\sigma \times \frac{\pi D^2}{4}} = \frac{8 a n_c}{\sigma D^2}. \tag{2.17}$$

The coil is part of an alternating current (AC) circuit (see Figure 2.5) whose remaining components harvest or store the electrical energy and have a total resistance of R_c. The movement of the magnet through the coil causes a current $I(t)$ to flow through the circuit via electromagnetic induction, with charge $Q(t)$ satisfying $\dot{Q} \equiv I$. At rest, the magnet is stationary and there is no ambient current or charge, so $I(0) = Q(0) = 0$.

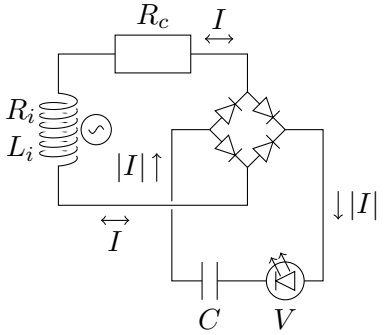


Figure 2.5: A schematic of the circuit used. The coil acts as both an AC source and an inductor when interacting with the magnet. Using an (idealised) bridge rectifier to convert the AC into DC, the current induced by the moving magnet lights up an LED.

By using Faraday's law to calculate the voltage induced by the magnet and Ohm's law to relate this to the current, the equation

$$L_i \dot{I} = \gamma G \dot{Z} - I(R_i + R_c) \tag{2.18}$$

is derived (see Appendix A.2 for details), where:

- L_i is the (constant) self-inductance of the coil, which depends on its geometry and is given by

$$L_i = K\frac{\mu_0 \pi a^2 n_c^2}{L}, \tag{2.19}$$

  where $0 < K \leq 1$ is a short-coil factor approaching 1 as $a/L \to 0$;[13]

- $\gamma G\dot{Z}$ is the induced voltage;

- γ is a constant describing the size and strength of the electromagnet, given by

$$\gamma = \frac{\mu_0 m a^2 n_c}{2L},$$

  where $\mu_0 \approx 4\pi \times 10^{-7}\,\mathrm{H\,m^{-1}}$ is a universal constant known as the *permeability of free space*; and

- $G(Z)$ is a function which acts to apply the magnetic field of the magnet to the entire coil and is given by

$$\begin{aligned} G &\equiv \frac{1}{\pi a A_m^2 L_m}\int_{-\frac{L}{2}}^{\frac{L}{2}}\int_0^{2\pi}\int_0^{A_m}(\tilde{r}\cos\tilde{\phi}-a)(\mathcal{G}^+-\mathcal{G}^-)\tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}\,\mathrm{d}\tilde{z} \\ &\quad \text{with } \mathcal{G}^{\pm} = \left(a^2 - 2a\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (Z_0 + \alpha_h H_m + \tilde{z} - Z \pm {}^{L_m}\!/_2)^2\right)^{-{}^3\!/_2} \\ &\approx \left(a^2 + (Z_0 + \alpha_h H_m - {}^{L}\!/_2 - Z)^2\right)^{-{}^3\!/_2} - \left(a^2 + (Z_0 + \alpha_h H_m + {}^{L}\!/_2 - Z)^2\right)^{-{}^3\!/_2}. \end{aligned} \tag{2.20}$$

Despite being derived using a far-field assumption, the approximate expression for G it is a very good estimator, even in close proximity to the magnet (see Figure 2.6). The offset $\alpha_h \neq 0$ ensures that $G(Z_0) \neq 0$, which is vital for the linearisation of the device (see Section 2.4).

The additional components in the circuit are a capacitor with capacitance C,[14] which stores charge on its plates, and an light-emitting diode (LED), which converts the voltage into light. An (idealised) bridge rectifier is used to convert the AC I into direct current (DC) $|I|$, ensuring that the LED remains lit regardless of the direction of the induced

[13] See Table 8.1 in [65, p. 347] for a selection of values of K for some given a/L.
[14] This has been added since the initial work of Bokhove et al. [44, 45].

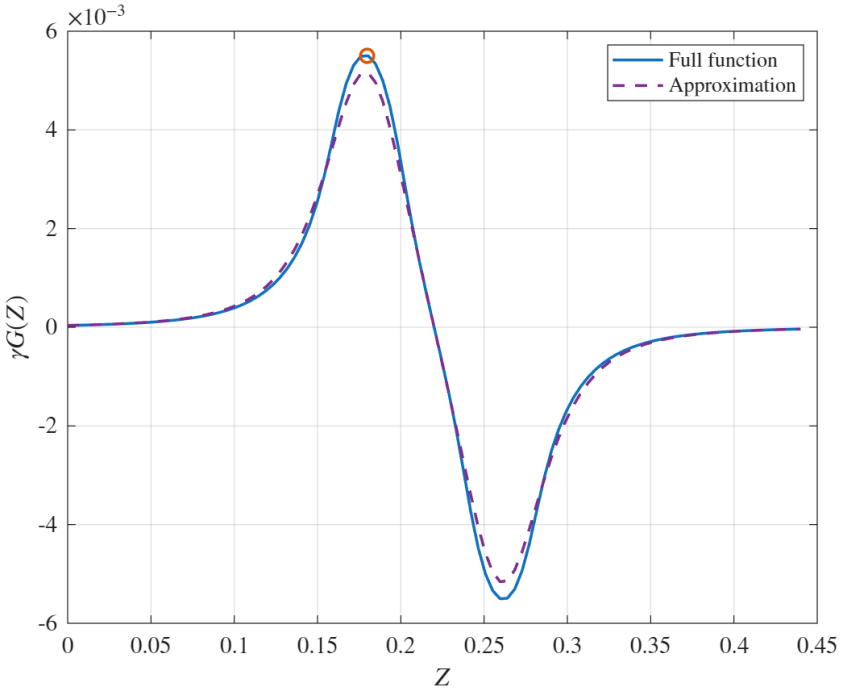


Figure 2.6: γG as a function of Z, using both the exact expression (with integrals solved numerically) and the far-field approximation. The value at $Z = Z_0$ is circled. Parameter values are as outlined in Table 3.2.

current.

The voltage in the capacitor is simply Q/C, while the voltage in the LED is modelled using an adapted form of the Shockley diode equation:

$$V(I) \equiv V(t) \equiv \text{sign}(I) n_q V_T \ln\left(1 + \frac{|I|}{I_{sat}}\right), \tag{2.21}$$

with n_q the quality factor, V_T the thermal voltage and I_{sat} the saturation current. This expression is obtained by algebraically inverting the standard Shockley current–voltage relation (e.g. [66, p. 94]) and symmetrically extending the relation to allow for bidirectional currents. Initially, $V|_{t=0} = 0$. The voltages of these components can simply be subtracted from[15] the voltage balance (2.18), giving

$$L_i \dot{I} = \gamma G \dot{Z} - \frac{Q}{C} - I(R_i + R_c) - V. \tag{2.22}$$

The power produced by the harvested voltage is $P(t) \equiv IV$, with the mean power output

[15] Since both components draw voltage *out* of the circuit (for the capacitor, when $Q > 0$), a negative sign is required to describe the voltage balance correctly.

between $t = 0$ and $t = T$ given by

$$\hat{P}_g \equiv \frac{1}{T}\int_0^T P\,\mathrm{d}t = \frac{1}{T}\int_0^T IV\,\mathrm{d}t. \tag{2.23}$$

Similarly, the power dissipated by the resistances in the circuit is

$$\hat{P}_l \equiv \frac{1}{T}\int_0^T I^2(R_i + R_c)\,\mathrm{d}t. \tag{2.24}$$

The equation (2.22) represents Kirchhoff's voltage law for the circuit, which can be viewed as analogous to Newton's second law ($\mathbf{F} = m\mathbf{a}$) for an electrical circuit:

- The rate-of-change of current $\dot{I}$, analogous to acceleration, describes how the circuit responds to the "forces" (voltages) applied.
- The inductance L_i, analogous to mass, measures the circuit's resistance to changes in current.
- The induced voltage $\gamma G\dot{Z}$ acts as the driving force that pushes current through the circuit.
- In the same way that a spring responds to changes in displacement, the capacitor responds to changes in charge, storing or releasing energy as needed.
- The resistances oppose and dissipate current, analogous to linear damping or drag.
- Similarly, the voltage harvested by the LED reduces the effective driving force.

To define the circuit's Lagrangian, first note that the kinetic energy analogue is the energy stored in the inductor, $L_i\dot{Q}^2/2$, and the potential energy is that of the capacitor, $Q^2/2C$. Also, since the dissipative elements $I(R_i + R_c)$ and V are non-conservative, we do not include them a priori. However, to capture the driving force and coupling between the buoy and the generator, we must return to our definition of the Lagrangian.

In Section 2.1.3, we stated that kinetic energy generally depends on velocity, with potential energy depending on position.[16] However, the Lagrangian approach does not neces-

[16] Typical position-only potentials include gravitational potential and spring potential, both of which we have already used.

sarily require potential energy do be dependent upon position only; "velocity-dependent" or "generalized" potentials may be used. One common example of a generalized potential is precisely the case we have here: electromagnetic interaction of moving charges. Referring to Goldstein et al. [67, 22–24] for a more detailed discussion, the potential for the buoy-generator coupling is $-\gamma G\dot{Z}Q$; we can clearly see the dependence upon the buoy's velocity $\dot{Z}$.

Therefore, the full Lagrangian for the generator is

$$\mathcal{L}_e = \frac{1}{2}L_i\dot{Q}^2 - \frac{Q^2}{2C} + \gamma G\dot{Z}Q. \tag{2.25}$$

The variation of (2.25) in Q gives the voltage balance (2.22), in the absence of the dissipative terms $I(R_i + R_c)$ and V and with the identity $\dot{I} \equiv \ddot{Q}$.

2.3 3D potential-flow equations

The Lagrangians for the three sub-systems plus the six hydrodynamic constraints are given in (2.14), (2.15), (2.25) and (2.16). These can all be combined into a single Lagrangian for the entire system, given by

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_h|_{\rho_0\to\rho} + \mathcal{L}_b + \mathcal{L}_c + \mathcal{L}_e \\
&= \int_0^{L_x}\Bigg[\int_R^{l_y}\int_0^h \left\{\frac{1}{2}\rho\|\mathbf{u}\|^2 - \rho g(z-H_0) + \mathcal{P}(\rho_0-\rho) + \phi\left(\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u})\right)\right\}\mathrm{d}z\,\mathrm{d}y \\
&\qquad + \left\{\int_0^h \boldsymbol{\chi}\cdot\left(\mathbf{u} - \dot{R}\hat{\mathbf{y}}\right)\mathrm{d}z\right\}\Bigg|_{y=R} \\
&\qquad + \int_R^{l_y}\left\{\Psi\left(\frac{\partial h}{\partial t} + \mathbf{u}|_{z=h}\cdot(\nabla_H h, -1)\right) + \rho_0\lambda(h_b-h)\Theta(y-y_b)\right\}\mathrm{d}y\Bigg]\,\mathrm{d}x \\
&\quad + \iint_{\partial\Omega_w}\xi\mathbf{u}\cdot\hat{\mathbf{n}}\,\mathrm{d}S + \frac{1}{2}M\dot{Z}^2 - MgZ + \frac{1}{2}L_i\dot{Q}^2 - \frac{Q^2}{2C} + \gamma G\dot{Z}Q.
\end{aligned} \tag{2.26}$$

Variations of this nonlinear, 3D Lagrangian will generate the nonlinear 3D potential-flow equations governing the entire system.

2.3.1 Simplification of the Lagrangian

Following the process of Cotter and Bokhove [68], before taking variations, (2.26) can be manipulated and drastically simplified, by taking the variation in $\mathbf{u}$ to eliminate a number of Lagrange multipliers. To begin, we shall transfer the divergence of $\rho\mathbf{u}$ (the red term) using the (reversed) 3D IBP (A.48), with $f = \phi$ and $\mathbf{F} = \rho\mathbf{u}$:

$$\begin{aligned}\int_0^{L_x}\int_R^{l_y}\int_0^h \phi\nabla\cdot(\rho\mathbf{u})\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x = &-\int_0^{L_x}\int_R^{l_y}\int_0^h \rho\mathbf{u}\cdot\nabla\phi\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\\ &-\int_0^{L_x}\left\{\int_0^h \phi\rho v\,\mathrm{d}z\right\}\Big|_{y=R}\mathrm{d}x\\ &+\int_0^{L_x}\int_R^{l_y}(\phi\rho\mathbf{u})|_{z=h}\cdot(-\nabla_H h,1)\,\mathrm{d}y\,\mathrm{d}x\\ &+\iint_{\partial\Omega_w}\phi\rho\mathbf{u}\cdot\hat{\mathbf{n}}\,\mathrm{d}S.\end{aligned}$$

This gives

$$\begin{aligned}\mathcal{L} = \int_0^{L_x}\Bigg[\int_R^{l_y}\int_0^h &\left\{\frac{1}{2}\rho\|\mathbf{u}\|^2-\rho g(z-H_0)+\mathcal{P}(\rho_0-\rho)+\phi\frac{\partial\rho}{\partial t}-\rho\mathbf{u}\cdot\nabla\phi\right\}\mathrm{d}z\,\mathrm{d}y\\ &+\left\{\int_0^h \boldsymbol{\chi}\cdot\left(\mathbf{u}-\dot{R}\hat{\mathbf{y}}\right)-\phi\rho v\,\mathrm{d}z\right\}\Big|_{y=R}\\ &+\int_R^{l_y}\Bigg\{\Psi\frac{\partial h}{\partial t}+(\Psi-(\phi\rho)|_{z=h})\mathbf{u}|_{z=h}\cdot(\nabla_H h,-1)\\ &\qquad+\rho_0\lambda(h_b-h)\Theta(y-y_b)\Bigg\}\mathrm{d}y\Bigg]\mathrm{d}x\\ +\iint_{\partial\Omega_w}(\xi+\phi\rho)\mathbf{u}\cdot\hat{\mathbf{n}}\,\mathrm{d}S &+\frac{1}{2}M\dot{Z}^2-MgZ+\frac{1}{2}L_i\dot{Q}^2-\frac{Q^2}{2C}+\gamma G\dot{Z}Q.\end{aligned}\tag{2.27}$$

The variation of the action $\mathcal{S}$ in $\mathbf{u}$ is then

$$\begin{aligned}\delta_{\mathbf{u}}\mathcal{S} = \int_{t_1}^{t_2}\Bigg(\int_0^{L_x}\Bigg[\int_R^{l_y}\int_0^h \rho\delta\mathbf{u}\cdot[\mathbf{u}-\nabla\phi]\,\mathrm{d}z\,\mathrm{d}y+\left\{\int_0^h[\boldsymbol{\chi}-(0,\phi\rho,0)]\cdot\delta\mathbf{u}\,\mathrm{d}z\right\}\Big|_{y=R}\\ +\int_R^{l_y}[\Psi-(\phi\rho)|_{z=h}]\delta\mathbf{u}|_{z=h}\cdot(\nabla_H h,-1)\,\mathrm{d}y\Bigg]\mathrm{d}x+\iint_{\partial\Omega_w}[\xi+\phi\rho]\delta\mathbf{u}\cdot\hat{\mathbf{n}}\,\mathrm{d}S\Bigg)\mathrm{d}t.\end{aligned}$$

The terms in square brackets correspond to the $\mathcal{F}_i$ in (2.7), giving

$$\mathbf{u} = \nabla\phi, \quad \boldsymbol{\chi} = (0, \phi\rho, 0), \quad \Psi = (\phi\rho)|_{z=h} \text{ and } \xi = -\phi\rho. \tag{2.28}$$

The first equality above shows that the Lagrange multiplier ϕ acts as a velocity potential, while the remaining relations express the Lagrange multipliers $\boldsymbol{\chi}$, Ψ and ξ as special cases of $\phi\rho$.

Note that, although no explicit assumption of irrotationality has been imposed, the variation yields a velocity potential describing irrotational flow. The Lagrangian (2.26) does not accommodate independent vorticity degrees-of-freedom, and consequently only admits solutions that are irrotational. To describe rotational flows, the formulation must be extended to include additional structure (e.g. Clebsch variables, or a Lagrangian particle-relabelling framework [69, 70]). This restriction of the hydrodynamics to irrotational flow is standard in water-wave formulations.

Using the equalities from (2.28) in (2.27) gives

$$\begin{aligned}\mathcal{L} = \int_0^{L_x} \Bigg[\int_R^{l_y} \int_0^h \left\{ -\frac{1}{2}\rho\|\nabla\phi\|^2 - \rho g(z - H_0) + \mathcal{P}(\rho_0 - \rho) + \phi\frac{\partial\rho}{\partial t} \right\} \mathrm{d}z\,\mathrm{d}y \\ - \left\{ \int_0^h \phi\rho\dot{R}\,\mathrm{d}z \right\}\Bigg|_{y=R} + \int_R^{l_y} \left\{ (\phi\rho)|_{z=h}\frac{\partial h}{\partial t} + \rho_0\lambda(h_b - h)\Theta(y - y_b) \right\} \mathrm{d}y \Bigg] \mathrm{d}x \\ + \frac{1}{2}M\dot{Z}^2 - MgZ + \frac{1}{2}L_i\dot{Q}^2 - \frac{Q^2}{2C} + \gamma G\dot{Z}Q.\end{aligned}$$

This can be simplified further, by using the 3D Reynolds transport theorem (RTT) (A.52) with $f = \phi\rho$ to combine the three blue terms into one:

$$\begin{aligned}\int_0^{L_x} \left\{ \int_R^{l_y} \int_0^h \phi\frac{\partial\rho}{\partial t}\,\mathrm{d}z\,\mathrm{d}y - \left\{ \int_0^h \phi\rho\dot{R}\,\mathrm{d}z \right\}\Bigg|_{y=R} + \int_R^{l_y} (\phi\rho)|_{z=h}\frac{\partial h}{\partial t}\,\mathrm{d}y \right\} \mathrm{d}x \\ = -\int_0^{L_x}\int_R^{l_y}\int_0^h \rho\frac{\partial\phi}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x + \frac{\mathrm{d}}{\mathrm{d}t}\cancel{\left(\int_0^{L_x}\int_R^{l_y}\int_0^h \phi\rho\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x \right)}.\end{aligned}$$

This results in

$$\mathcal{L} = \int_0^{L_x} \int_R^{l_y} \left[\int_0^h \left\{ -\rho \left(\frac{\partial \phi}{\partial t} + \frac{1}{2} \|\nabla \phi\|^2 + g(z - H_0) \right) + \mathcal{P}(\rho_0 - \rho) \right\} \mathrm{d}z \right.$$
$$\left. + \rho_0 \lambda (h_b - h) \Theta(y - y_b) \right] \mathrm{d}y \, \mathrm{d}x + \frac{1}{2} M \dot{Z}^2 - MgZ + \frac{1}{2} L_i \dot{Q}^2 - \frac{Q^2}{2C} + \gamma G \dot{Z} Q.$$

The total time derivative is discarded because, when taking variations, the time derivative and integral will cancel to leave

$$\delta \left[\int_0^{L_x} \int_R^{l_y} \int_0^h \phi \rho \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x \right]_{t_1}^{t_2},$$

which will be zero because of the endpoint conditions (2.8).

After multiplying by -1 (which does not affect the result of the variations), we have

$$\mathcal{L}[\mathbf{v}] = \int_0^{L_x} \int_R^{l_y} \left[\int_0^h \left\{ \rho \left(\frac{\partial \phi}{\partial t} + \frac{1}{2} \|\nabla \phi\|^2 + g(z - H_0) \right) + \mathcal{P}(\rho - \rho_0) \right\} \mathrm{d}z \right.$$
$$\left. + \rho_0 \lambda (h - h_b(y, Z)) \Theta(y - y_b) \right] \mathrm{d}y \, \mathrm{d}x - \frac{1}{2} M \dot{Z}^2 + MgZ - \frac{1}{2} L_i \dot{Q}^2 + \frac{Q^2}{2C} - \gamma G(Z) \dot{Z} Q, \tag{2.29}$$

with

$$\mathbf{v} = (h, \rho, \phi, \mathcal{P}, \lambda, Z, Q) \tag{2.30}$$

the vector of unknowns with which variations must be taken. This Lagrangian is equivalent to the one used by Bokhove et al. [44, 45], which can be obtained from (2.29) via the use of a Legendre transform (see Appendix A.4 for details).

2.3.2 Hamiltonian and energy balance

Following the process outlined in Section 2.1.3, we can derive the Hamiltonian for the system. First, we must reverse the negation of the Lagrangian. Then, the velocities are $\partial\phi/\partial t$, $\dot{Z}$ and $\dot{Q}$, with corresponding momenta

$$P_\phi \equiv \frac{\partial \mathcal{L}}{\partial \partial\phi/\partial t} = -\int_0^{L_x} \int_R^{l_y} \int_0^h \rho \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x;$$
$$P_Z \equiv \frac{\partial \mathcal{L}}{\partial \dot{Z}} = M\dot{Z} + \gamma G Q \Rightarrow \dot{Z} = \frac{P_Z - \gamma G Q}{M};$$

$$P_Q \equiv \frac{\partial \mathcal{L}}{\partial \dot{Q}} = L_i \dot{Q} \Rightarrow \dot{Q} = \frac{P_Q}{L_i}.$$

After enforcing the constraints $\rho = \rho_0$ and $h = h_b$ for $y \geq y_b$ strongly (that is, using the equalities directly and therefore removing the relevant Lagrange-multiplier terms), the Hamiltonian is

$$\mathcal{H} = \rho_0 \int_0^{L_x} \int_R^{l_y} \int_0^h \left\{ \frac{1}{2} \|\nabla \phi\|^2 + g(z - H_0) \right\} \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x + \frac{(P_Z - \gamma G Q)^2}{2M} + MgZ + \frac{P_Q^2}{2L_i} + \frac{Q^2}{2C}.$$

Using the identities

$$P_Z \equiv MW + \gamma G Q \text{ and } P_Q \equiv L_i I$$

to relate the velocity W and current I to the momenta P_Z and P_Q,[17] we have

$$\mathcal{H} = \rho_0 \int_0^{L_x} \int_{R(t)}^{l_y} \int_0^h \left\{ \frac{1}{2} \|\nabla \phi\|^2 + g(z - H_0) \right\} \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x + \frac{1}{2} MW^2 + MgZ + \frac{1}{2} L_i I^2 + \frac{Q^2}{2C}.$$

The energy balance is given by $\partial \mathcal{H} / \partial t$. The only explicit time-dependence comes from the wavemaker $R(t)$,[18] and so, using the 3D RTT (A.52) with ϕ and h held constant, we find

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial t} &= \rho_0 \frac{\partial}{\partial t} \left(\int_0^{L_x} \int_{R(t)}^{l_y} \int_0^h \left\{ \frac{1}{2} \|\nabla \phi\|^2 + g(z - H_0) \right\} \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x \right) \\ &= -\rho_0 \dot{R} \int_0^{L_x} \left\{ \int_0^h \left(\frac{1}{2} \|\nabla \phi\|^2 + g(z - H_0) \right) \mathrm{d}z \right\} \Bigg|_{y=R} \mathrm{d}x. \end{aligned}$$

Therefore, internal energy is not conserved, as the wavemaker motion $\dot{R}$ injects and extracts energy over time. As soon as the wavemaker is turned off $\dot{R} = 0$, we have $\partial \mathcal{H} / \partial t = 0$, and energy is then conserved.

2.3.3 Variations and equations

Using the Lagrangian (2.29), the variations of $\mathcal{S}$ (2.4) in the seven unknowns (2.30) can be evaluated. Most of the variations are simple to write down, but with h appearing as

[17] Note that: a) the buoy "momentum" P_Z is not simply the mechanical momentum MW; it also includes the coupling term $\gamma G Q$; and b) the electromagnetic "momentum" is equivalent to the magnetic flux $\Phi = L_i I$ (e.g. [71, p. 324]).

[18] Many other functions depend on t—e.g. $h(x, y, t)$, $Z(t)$, $Q(t)$—but since these are unknowns, their time-dependence is implicit.

the upper bound of the z integral and Z as a parameter in the functions h_b and G, the variation for these unknowns is not immediately obvious.

First, writing the z integrand as F, i.e.

$$\rho\left(\frac{\partial\phi}{\partial t}+\frac{1}{2}\|\nabla\phi\|^2+g(z-H_0)\right)+\mathcal{P}(\rho-\rho_0)\equiv F,$$

the z integral can be written as a function of h:

$$\int_0^h F\,\mathrm{d}z\equiv f(h).$$

Now, a Taylor expansion of $f(h+\varepsilon\delta h)$ can be carried out, giving

$$f(h+\varepsilon\delta h)-f(h)=\varepsilon f'(h)\delta h+\cancel{\mathcal{O}\left(\varepsilon^2\right)}.$$

Next, $f'(h)$ can be evaluated using Leibniz's integral rule:

$$f'(h)\equiv\frac{\mathrm{d}}{\mathrm{d}h}\left(\int_0^h F\,\mathrm{d}z\right)=F|_{z=h}\frac{\partial h}{\partial h}+\int_0^h\frac{\cancel{\partial F}}{\cancel{\partial h}}\,\mathrm{d}z=F|_{z=h},$$ [19]

giving

$$f(h+\varepsilon\delta h)-f(h)=\varepsilon F|_{z=h}\delta h.$$

For the variation in Z, note that because h_b is linear in Z, we have simply

$$h_b(y,Z+\varepsilon\delta Z)-h_b(y,Z)=\varepsilon\delta Z.$$

However, the $G\dot{Z}$ term requires further attention. The variation produces the expression

$$-\gamma\left(G(Z+\varepsilon\delta Z)(\dot{Z}+\varepsilon\delta\dot{Z})-G(Z)\dot{Z}\right)Q, \tag{2.31}$$

where again $G(Z+\varepsilon\delta Z)$ can be Taylor-expanded:

$$G(Z+\varepsilon\delta Z)=G(Z)+\varepsilon G'(Z)\delta Z+\cancel{\mathcal{O}\left(\varepsilon^2\right)},$$

[19] Note the integrand F is not a function of h.

turning (2.31) into

$$-\gamma\left(\cancel{G\dot{Z}}+\varepsilon G\delta\dot{Z}+\varepsilon G'\delta Z\dot{Z}+\cancel{\mathcal{O}\left(\varepsilon^2\right)}-\cancel{G\dot{Z}}\right)Q=-\gamma\varepsilon\left(G\delta\dot{Z}+G'\delta Z\dot{Z}\right)Q. \tag{2.32}$$

The G' term disappears when dealing with the $\delta\dot{Z}$ term, via the product rule for three factors:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(G\delta ZQ)&=G\delta Z\dot{Q}+\left(G\delta\dot{Z}+\dot{G}\delta Z\right)Q\\&=G\delta Z\dot{Q}+\left(G\delta\dot{Z}+G'\dot{Z}\delta Z\right)Q,\end{aligned}$$

turning (2.32) into

$$\gamma\varepsilon\left(G\delta Z\dot{Q}-\cancel{\frac{\mathrm{d}}{\mathrm{d}t}\{G\delta ZQ\}}\right).^{20}$$

The variations in h and Z can now be written down. After eliminating the derivatives of $\delta\phi$ via the 3D RTT (A.52) and the 3D IBP (A.48), and δZ and δQ via a simple 1D IBP:

Initial variations:

$$\begin{aligned}\delta_h\mathcal{S}&=\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\delta h\left[F|_{z=h}+\rho_0\lambda\Theta(y-y_b)\right]\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\
\delta_\rho\mathcal{S}&=\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^{h}\delta\rho\left[\frac{\partial\phi}{\partial t}+\frac{1}{2}\|\nabla\phi\|^2+g(z-H_0)+\mathcal{P}\right]\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\
\delta_\phi\mathcal{S}&=\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^{h}\left\{\rho\frac{\partial\delta\phi}{\partial t}+\rho\nabla\phi\cdot\nabla\delta\phi\right\}\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\
\delta_{\mathcal{P}}\mathcal{S}&=\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^{h}\delta\mathcal{P}\left[\rho-\rho_0\right]\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\
\delta_\lambda\mathcal{S}&=\rho_0\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\delta\lambda[h-h_b]\Theta(y-y_b)\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\
\delta_Z\mathcal{S}&=\int_{t_1}^{t_2}\left\{-M\dot{Z}\delta\dot{Z}+\delta Z\left[Mg+\gamma G\dot{Q}-\rho_0\int_0^{L_x}\int_R^{l_y}\lambda\Theta(y-y_b)\,\mathrm{d}y\,\mathrm{d}x\right]\right\}\mathrm{d}t;\\
\delta_Q\mathcal{S}&=\int_{t_1}^{t_2}\left\{-L_i\dot{Q}\delta\dot{Q}+\delta Q\left[\frac{Q}{C}-\gamma G\dot{Z}\right]\right\}\mathrm{d}t.\end{aligned}$$

[20] Yet again, the endpoint conditions (2.8) are employed here to discard the time derivative; for the remainder of the thesis, it is assumed the reader is familiar with this step, and further explanations are omitted, with $\not{x}$ meaning x has been discarded. Similarly, it is assumed that terms in square brackets multiplied by variations δv_i are known to correspond to the $\mathcal{F}_i$ in (2.7).

3D RTT (A.52) with $f = \delta\phi\rho$ (on blue term):

$$\begin{aligned}\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^h \rho\frac{\partial\delta\phi}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t = &-\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^h \delta\phi\frac{\partial\rho}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t\\ &+\int_{t_1}^{t_2}\int_0^{L_x}\left\{\int_0^h \delta\phi\rho\dot{R}\,\mathrm{d}z\right\}\Bigg|_{y=R}\mathrm{d}x\,\mathrm{d}t\\ &-\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}(\delta\phi\rho)|_{z=h}\frac{\partial h}{\partial t}\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t.\end{aligned}$$

3D IBP (A.48) with $f = \delta\phi$ and $\mathbf{F} = \rho\nabla\phi$ (on red term):

$$\begin{aligned}&\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^h \rho\nabla\phi\cdot\nabla\delta\phi\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t\\ &\qquad= -\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^h \delta\phi\nabla\cdot(\rho\nabla\phi)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t\\ &\qquad\quad-\int_{t_1}^{t_2}\int_0^{L_x}\left\{\int_0^h \delta\phi\rho\frac{\partial\phi}{\partial y}\,\mathrm{d}z\right\}\Bigg|_{y=R}\mathrm{d}x\,\mathrm{d}t\\ &\qquad\quad+\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}(\delta\phi\rho\nabla\phi)|_{z=h}\cdot(-\nabla_H h,1)\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t\\ &\qquad\quad+\int_{t_1}^{t_2}\iint_{\partial\Omega_w}\delta\phi\rho\nabla\phi\cdot\hat{\mathbf{n}}\,\mathrm{d}S\,\mathrm{d}t.\end{aligned}$$

1D IBP (on green terms):

$$\begin{aligned}-\int_{t_1}^{t_2} M\dot{Z}\delta\dot{Z}\,\mathrm{d}t &= -\cancel{\left[M\dot{Z}\delta Z\right]_{t_1}^{t_2}} + \int_{t_1}^{t_2} M\ddot{Z}\delta Z\,\mathrm{d}t,\\ -\int_{t_1}^{t_2} L_i\dot{Q}\delta\dot{Q}\,\mathrm{d}t &= -\cancel{\left[L_i\dot{Q}\delta Q\right]_{t_1}^{t_2}} + \int_{t_1}^{t_2} L_i\ddot{Q}\delta Q\,\mathrm{d}t.\end{aligned}$$

The final variations are:

$$\begin{aligned}\delta_h\mathcal{S} &= \int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\delta h\,[F|_{z=h} + \rho_0\lambda\Theta(y-y_b)]\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\ \delta_\rho\mathcal{S} &= \int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^h \delta\rho\left[\frac{\partial\phi}{\partial t} + \frac{1}{2}\|\nabla\phi\|^2 + g(z-H_0) + \mathcal{P}\right]\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;\\ \delta_\phi\mathcal{S} &= \int_{t_1}^{t_2}\Bigg(\int_0^{L_x}\Bigg[-\int_R^{l_y}\int_0^h \delta\phi\left[\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\nabla\phi)\right]\mathrm{d}z\,\mathrm{d}y\\ &\qquad\qquad+\left\{\int_0^h \delta\phi\rho\left[\dot{R} - \frac{\partial\phi}{\partial y}\right]\mathrm{d}z\right\}\Bigg|_{y=R}\\ &\qquad\qquad+\int_R^{l_y}(\delta\phi\rho)|_{z=h}\left[(\nabla\phi)|_{z=h}\cdot(-\nabla_H h,1) - \frac{\partial h}{\partial t}\right]\mathrm{d}y\Bigg]\mathrm{d}x\end{aligned}$$

$$+ \iint_{\partial\Omega_w} \delta\phi\rho\nabla\phi\cdot\hat{\mathbf{n}}\,\mathrm{d}S\Bigg)\,\mathrm{d}t;$$

$$\delta_{\mathcal{P}}\mathcal{S} = \int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\int_0^h \delta\mathcal{P}\,[\rho-\rho_0]\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;$$

$$\delta_{\lambda}\mathcal{S} = \rho_0\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y} \delta\lambda[h-h_b]\Theta(y-y_b)\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t;$$

$$\delta_{Z}\mathcal{S} = \int_{t_1}^{t_2}\delta Z\left[M\ddot{Z}+Mg+\gamma G\dot{Q}-\rho_0\int_0^{L_x}\int_R^{l_y}\lambda\Theta(y-y_b)\,\mathrm{d}y\,\mathrm{d}x\right]\mathrm{d}t;$$

$$\delta_{Q}\mathcal{S} = \int_{t_1}^{t_2}\delta Q\left[L_i\ddot{Q}+\frac{Q}{C}-\gamma G\dot{Z}\right]\mathrm{d}t.$$

The variations in h, ρ, ϕ and $\mathcal{P}$ yield the following equations:

$$\delta h : \rho\left(\frac{\partial\phi}{\partial t}+\frac{1}{2}\|\nabla\phi\|^2+g(h-H_0)\right)+\mathcal{P}(\rho-\rho_0)+\rho_0\lambda\Theta(y-y_b)=0 \ \text{ at } \ z=h, \tag{2.33a}$$

$$\delta\rho : \frac{\partial\phi}{\partial t}+\frac{1}{2}\|\nabla\phi\|^2+g(z-H_0)+\mathcal{P}=0, \tag{2.33b}$$

$$\delta\phi : \frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\nabla\phi)=0, \tag{2.33c}$$

$$\delta\mathcal{P} : \rho=\rho_0. \tag{2.33d}$$

We can use (2.33d) to simplify some of the other equations: (2.33c) becomes

$$\nabla^2\phi=0, \tag{2.34}$$

and (2.33a) becomes

$$\frac{\partial\phi}{\partial t}+\frac{1}{2}\|\nabla\phi\|^2+g(h-H_0)+\lambda\Theta(y-y_b)=0 \ \text{ at } \ z=h. \tag{2.35}$$

A link between $\mathcal{P}$ and λ can be obtained by comparing (2.35) with the evaluation of (2.33b) at $z=h$. This gives $\mathcal{P}|_{z=h}=\lambda\Theta(y-y_b)$, from which $\mathcal{P}|_{z=h}=0$ for $y<y_b$ and $\mathcal{P}|_{z=h}=\lambda$ for $y\geq y_b$. That is, $\mathcal{P}$ is the hydrostatic pressure of the water (normalised by ρ_0), which is zero at the free surface, and λ is the normalised hydrodynamic pressure at the interface of the buoy and the water, which is zero at the waterline.

The variation in Z gives

$$M\ddot{Z} = -Mg - \gamma G\dot{Q} + \rho_0 \int_0^{L_x} \int_R^{l_y} \lambda\Theta(y - y_b)\,\mathrm{d}y\,\mathrm{d}x. \tag{2.36}$$

Similarly, the variation in Q gives

$$L_i\ddot{Q} = \gamma G\dot{Z} - \frac{Q}{C}, \tag{2.37}$$

which is equivalent to the voltage balance (2.22) in the non-dispersive limit R_i, R_c, $V \to 0$. Both (2.36) and (2.37) are second-order ODEs, which can be reduced to being first order via the identities $\dot{Z} = W$ and $\dot{Q} = I$, giving:

$$M\dot{W} = -Mg - \gamma GI + \rho_0 \int_0^{L_x} \int_R^{l_y} \lambda\Theta(y - y_b)\,\mathrm{d}y\,\mathrm{d}x, \tag{2.38a}$$

$$\dot{Z} = W, \tag{2.38b}$$

$$\dot{Q} = I, \tag{2.38c}$$

$$L_i\dot{I} = \gamma GW - \frac{Q}{C}. \tag{2.38d}$$

These equations form a system of first-order ODEs that describe the buoy/generator sub-system of the device, and clearly show how γG is integral to the coupling between them. This system is the subject of Chapter 6, where the hydrodynamic pressure term in (2.38a) is approximated as a simple forcing term $F(t)$.

We can now: replace (2.33a)–(2.33d) with (2.34), (2.35) and the BC on λ; replace (2.36) and (2.37) with (2.38a)–(2.38d), with the dispersive terms included in (2.38d); and evaluate the remaining variations (in $\phi|_{y=R}$, $\phi|_{z=h}$, $\phi|_{\partial\Omega_w}$ and λ), to obtain our final system of equations:

Hydrodynamics:

$$\nabla^2\phi = 0, \tag{2.39a}$$

$$\frac{\partial\phi}{\partial t} + \frac{1}{2}\|\nabla\phi\|^2 + g(h - H_0) + \lambda\Theta(y - y_b) = 0 \ \text{ at } \ z = h, \tag{2.39b}$$

$$\frac{\partial h}{\partial t} + \nabla_H\phi \cdot \nabla_H h = \frac{\partial\phi}{\partial z} \ \text{ at } \ z = h, \tag{2.39c}$$

Buoy dynamics:

$$M\dot{W} = -Mg - \gamma GI + \rho_0 \int_0^{L_x} \int_R^{l_y} \lambda\Theta(y - y_b)\,\mathrm{d}y\,\mathrm{d}x, \tag{2.39d}$$

$$\dot{Z} = W, \tag{2.39e}$$

Electromagnetic generator:

$$\dot{Q} = I, \tag{2.39f}$$

$$L_i\dot{I} = \gamma GW - \frac{Q}{C} - I(R_i + R_c) - V, \tag{2.39g}$$

Free-surface-equals-buoy-shape constraint:

$$h = h_b \text{ for } y \geq y_b, \tag{2.39h}$$

Boundary conditions:

$$\lambda = 0 \text{ at } y = y_b, \tag{2.39i}$$

$$\frac{\partial\phi}{\partial y} = \dot{R} \text{ at } y = R, \tag{2.39j}$$

$$\nabla\phi \cdot \hat{\mathbf{n}} = 0 \text{ on } \partial\Omega_w. \tag{2.39k}$$

The system includes three pairs of equations relating to the device's three components, each governing the rate-of-change of a conjugate pair: $\{\phi, h\}$, $\{W, Z\}$ and $\{Q, I\}$.

The first pair of equations (2.39b) and (2.39c) are the Bernoulli and kinematic conditions at the free surface $z = h$. Together with the Laplace equation (2.39a), these govern the incompressible hydrodynamics via the velocity potential ϕ. Included in (2.39b) is the pressure applied to the water by the wetted part of the buoy. The second pair of equations (2.39d) and (2.39e) govern the position and velocity of the buoy. The second of these is Newton's second law, involving the gravitational force ($-Mg$), the electromagnetic force applied by the induction motor above (second term on the right-hand side (RHS)) and the force applied by the water below (final term). The third pair of equations (2.39f) and (2.39g) concern the electromagnetic power generator.

An initial condition on λ is found by considering the Bernoulli condition (2.39b) for $y \geq y_b$, evaluated at $t = 0$ and using the at-rest surface position (2.11), giving

$$\lambda|_{t=0} \equiv \Lambda(y) \equiv g(H_0 - H_b), \tag{2.40}$$

with $\Lambda(L_b) = 0$ by (2.39i). This initial pressure must generate a force on the buoy that is equal and opposite to gravity, since the net force at rest must be zero and there is no electromagnetic force. Mathematically, this means

$$\rho_0 \int_0^{L_x} \int_0^{l_y} \Lambda\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x = Mg. \tag{2.41}$$

The system (2.39) can be approximated in a number of ways. This includes the linearisation around the rest state of the system, which is outlined in Section 2.4, and the removal of the vertical z coordinate and thus a reduction to a 2D domain, which is the subject of Chapter 3. These approximations ultimately lead to an increase in efficiency when carrying out numerical solutions of the equations, which is of vital import when carrying out optimisation of the device (see Section 5.5).

2.4 Linearisation

The system (2.39) can be linearised about its rest-state by writing the functions as their "rest-value-plus-displacement" decompositions, which are as follows:

$$\begin{aligned}
\phi(x,y,z,t) &= 0 + \widetilde{\phi}(x,y,z,t), & h(x,y,t) &= H(y) + \eta(x,y,t), & R(t) &= 0 + \widetilde{R}(t),\\
\lambda(x,y,t) &= \Lambda(y) + \widetilde{\lambda}(x,y,t), & y_b(x,t) &= L_b + \widetilde{y}_b(x,t), & W(t) &= 0 + \widetilde{W}(t),\\
Z(t) &= Z_0 + \widetilde{Z}(t), & Q(t) &= 0 + \widetilde{Q}(t), & I(t) &= 0 + \widetilde{I}(t).
\end{aligned}$$

Note that, in a similar vein, the position of the buoy's hull is

$$h_b(y, Z(t)) = H_b(y) + \widetilde{Z}(t).$$

After substituting these decompositions into the equations and noting that all constant rest-state terms can be cancelled, the resulting nonlinear system governing the displacements is linearised by: first, Taylor-expanding throughout to separate terms linear in the displacements from higher-order terms; then, under the assumption that the displacements are small, neglecting said higher-order terms. In this linearised limit, the moving

boundaries at $z = h(x, y, t)$, $y = R(t)$ and $y = y_b(x, t)$ become stationary boundaries at $z = H(y)$, $y = 0$ and $y = L_b$, respectively.

Referring to Appendix A.5.1 for a detailed derivation, the linearised system is as follows (after dropping the tildes):

Hydrodynamics:

$$\nabla^2\phi = 0, \tag{2.42a}$$

$$\frac{\partial\phi}{\partial t} + g\eta + \lambda\Theta(y - L_b) = 0 \text{ at } z = H, \tag{2.42b}$$

$$\frac{\partial\eta}{\partial t} + H'\frac{\partial\phi}{\partial y} = \frac{\partial\phi}{\partial z} \text{ at } z = H, \tag{2.42c}$$

Buoy dynamics:

$$M\dot{W} = -\gamma G(Z_0)I + \rho_0 \int_0^{L_x}\int_0^{l_y} \lambda\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x, \tag{2.42d}$$

$$\dot{Z} = W, \tag{2.42e}$$

Electromagnetic generator:

$$\dot{Q} = I, \tag{2.42f}$$

$$L_i\dot{I} = \gamma G(Z_0)W - \frac{Q}{C} - I(R_i + R_c + R_l), \tag{2.42g}$$

Free-surface-equals-buoy-shape constraint:

$$\eta = Z \text{ for } y \geq L_b, \tag{2.42h}$$

Boundary conditions:

$$\lambda = g(\eta|_{y=L_b^-} - Z) \text{ at } y = L_b, \tag{2.42i}$$

$$\frac{\partial\phi}{\partial y} = \dot{R} \text{ at } y = 0, \tag{2.42j}$$

$$\nabla\phi \cdot \hat{\mathbf{n}} = 0 \text{ on } \partial\Omega_w. \tag{2.42k}$$

Comparing the linear voltage balance (2.42g) with its nonlinear counterpart (2.39g), the nonlinear LED voltage V (2.21) linearises to IR_l, where $R_l \equiv n_q V_T/I_{sat}$ is a linearisation coefficient for the voltage-current relationship in the LED.

Note in (2.42d) and (2.42g) how W and I are coupled via the coefficient $\gamma G(Z_0)$. Re-

calling the definition of G (2.20), we have

$$G(Z_0) = \left(a^2 + (\alpha_h H_m - {}^{L}/_{2})^2\right)^{-{}^{3}/_{2}} - \left(a^2 + (\alpha_h H_m + {}^{L}/_{2})^2\right)^{-{}^{3}/_{2}},$$

and one can see that $G(Z_0)$ is equal to 0 when the offset α_h is also 0. Mathematically, this would result in the equations becoming uncoupled; physically, negligible[21] current would be induced by the magnet. Hence, a nonzero offset is required.

Given the fully nonlinear Lagrangian (2.29), we obtained the linear system (2.42) by first taking variations, and then linearising the equations. An alternative approach is to linearise the Lagrangian, and then take variations to obtain the linear equations directly. To carry out this linearisation, we can follow precisely the same steps: write the functions as their "rest-value-plus-displacement" decompositions, cancel-out leading-order rest-state terms, Taylor-expand the remaining terms, and remove higher-order contributions. However, note that whatever order a particular term in the Lagrangian is, it produces a term in an equation at a level one order below (e.g. linear variations give constant equations, and constant variations contribute nothing). Therefore, after Taylor-expanding, we retain terms that are quadratic in the displacements; our linearised Lagrangian is actually "quadricised".

Referring to Appendix A.5.2 for details, this "quadricised" Lagrangian is

$$\begin{aligned}\mathcal{L} = \rho_0 \int_0^{L_x} \Bigg[&\int_0^{l_y}\int_0^{H} \frac{1}{2}\|\nabla\phi\|^2 \,\mathrm{d}z\,\mathrm{d}y + \dot{R}\int_0^{H_0} \phi|_{y=0}\,\mathrm{d}z \\ &+ \int_0^{l_y} \left\{ \eta \left.\frac{\partial\phi}{\partial t}\right|_{z=H} + \frac{1}{2}g\eta^2 + \lambda(\eta - Z)\Theta(y - L_b) \right\} \mathrm{d}y \Bigg] \mathrm{d}x \\ &- \frac{1}{2}M\dot{Z}^2 - \frac{1}{2}L_i\dot{Q}^2 + \frac{Q^2}{2C} - \gamma G(Z_0)\dot{Z}Q, \end{aligned} \tag{2.43}$$

with associated Hamiltonian

$$\mathcal{H} = \rho_0 \int_0^{L_x} \left[\int_0^{l_y} \left\{ \int_0^{H} \frac{1}{2}\|\nabla\phi\|^2 \,\mathrm{d}z + \frac{1}{2}g\eta^2 \right\} \mathrm{d}y + \dot{R}(t)\int_0^{H_0} \phi|_{y=0}\,\mathrm{d}z \right] \mathrm{d}x$$

[21] Precisely at $Z = Z_0$, there would be no current induced, and the direction of the current would reverse as the buoy passes through this value. In the linearised limit, the buoy still oscillates about Z_0, but the absolute value of the induced current would be negligible to leading-order. The offset ensures that the current's magnitude is not neglected.

$$+\frac{1}{2}MW^2+\frac{1}{2}L_iI^2+\frac{Q^2}{2C}$$

and energy balance

$$\frac{\partial\mathcal{H}}{\partial t}=\rho_0\ddot{R}\int_0^{L_x}\int_0^{H_0}\phi|_{y=0}\,\mathrm{d}z\,\mathrm{d}x.$$

As before, the nonzero energy balance is impacted by the wavemaker. However, in the linearised limit, the dependence is upon $\ddot{R}$ not $\dot{R}$; this is an artifact of the linearisation, reflecting that in the small-amplitude, leading-order truncation, only accelerations of the boundary transfer energy. Also notice that the buoy's potential energy MgZ no longer appears in the Hamiltonian; this is due to the linearisation about the rest-state, into which the energy is absorbed, and where no leading-order perturbation exists.

It can be verified that taking the variations of the Lagrangian (2.43) produces the linearised system (2.42) directly (in the conservative limit $R_i,\ R_c,\ R_l \to 0$). Note the presence of terms evaluated at $y=0$ and $z=H$. There appear to be no analogous terms in the nonlinear Lagrangian (2.29); these are in-fact "hidden" inside the moving integral boundaries by the use of the 3D RTT (A.52). Because the boundaries become stationary in the linearised limit, the RTT cannot be used and these terms remain explicit.

Numerically, solving (2.42) will be far more efficient than solving its nonlinear counterpart (2.39). However, it is only valid under the small-displacement assumption. As part of the wider goal of maximising the output of the device through wave-amplification, this assumption is some-what of a paradox. Despite this, due to their simplicity and efficiency, linear models are often still useful in the early stages of WEC development, and as a stepping stone towards full nonlinear numerics. For the remainder of this thesis, the linear assumption shall be applied, though work to develop the numerical methodology required to solve the nonlinear model has been undertaken separately [72].

A further reduction in computational cost can be achieved under a shallow-water (SW) scaling, in which the vertical length scale is small relative to the horizontal extent. In this regime, the 3D Laplace equation (2.39a) can be reduced asymptotically, and depth-averaging over the vertical coordinate z then yields a 2D system. At leading order, the

potential becomes independent of z, producing the classical SW approximation employed by Bokhove et al. [44, 45]. Retaining higher-order terms captures vertical hydrodynamic effects more accurately, while still yielding a 2D model; this extension forms the focus of the next chapter.

Chapter 3

Benney–Luke: An Improved 2D Hydrodynamic Approximation?

In the 3D system (2.39), the hydrodynamic equations are time-dependent, 2D and 3D PDEs, while the buoy motion and power generation are governed by simple 1D ODEs in time. Computationally, solving the higher-dimensional equations is considerably more expensive, due to the larger number of discretisation points required. Consequently, most of the computational cost arises from solving the 3D Laplace equation (2.39a) to obtain the 3D velocity potential $\phi(x, y, z, t)$.

For practical purposes, accurate yet efficient 2D approximations, with a reduced number of discretisation nodes, are therefore highly desirable. As introduced at the end of the previous chapter, one approach is to represent ϕ in terms of some 2D function $\Phi(x, y, t)$ via an asymptotic expansion in the vertical coordinate z, which can then be depth-averaged to produce a 2D system. Taking only the leading-order term of this expansion yields the classical *shallow-water* (SW) approximation, in which the potential becomes entirely independent of z. This approach was adopted by Bokhove et al. [44, 45].

One may immediately ask the question: "Given that the coupling between the hydrodynamics and the buoy produces entirely vertical motion, how is a horizontally formulated, depth-averaged approximation permissible?" In short, this is because the buoy is not driven by the fully-resolved vertical fluid motion, but by the integrated pressure field

exerted by the water as a whole. Although the explicit vertical structure of the hydrodynamics has been integrated out, the pressure field λ is retained, and continues to produce a net upward force on the buoy.

As an alternative to the leading-order terms of the SW approximation considered previously, higher-order terms that more-accurately capture the vertical structure of the hydrodynamics can be retained. First introduced by Benney and Luke [73], this *Benney–Luke* (BL) approximation replaces the vertical dependence with higher-order derivative terms, describing dispersive effects in weakly nonlinear, large-wavelength waves and providing increased accuracy compared to the SW approximation. While previously applied to water-wave problems [74–77], this is believed to be the first application of the BL approximation in a system coupled with a rigid body.

The derivation of the approximation proceeds as follows. First, as is standard in water-wave theory (and described by Johnson [78]), the system is non-dimensionalised. The typical scalings for the coordinates x, y, z and t and the hydrodynamic unknowns ϕ and h (given in e.g. [74, 76]) are extended here to include the non-standard wavemaker and buoy terms, which are coupled directly to the hydrodynamics. Since the electromagnetic terms do not directly impact upon the hydrodynamics, these terms can be discarded while deriving the approximation, and can be recovered afterwards. These scalings are then used in the Lagrangian (2.29), creating a non-dimensional version from which a dimensionless Laplace equation and BCs for the potential are obtained. This is then used to write the dimensionless potential ϕ as an asymptotic expansion of some z-invariant function (which will become Φ) in powers of z. This expansion is then placed back into the dimensionless Lagrangian, from which the entire, explicit z-dependence can be integrated out to form a 2D alternative. Finally, this Lagrangian is re-dimensionalised, the electromagnetic terms are recovered, and variations are taken, resulting in a system of dimensional, 2D BL equations.

This chapter details the above process. Section 3.1 describes the non-dimensionalisation, while Section 3.2 discusses the expansion of ϕ. In Section 3.3, the vertical coordinate is removed and the 2D dimensional Lagrangian is formed, with the resulting equations derived (Section 3.4) and linearised (Section 3.5). Section 3.6 concludes the chapter with

a discussion surrounding the numerical solution of the 2D BL equations.

3.1 Non-dimensionalisation

For a wave with amplitude A_0, wavelength ℓ_0 and rest-depth H_0, we define a large-amplitude nonlinearity parameter $\epsilon = A_0/H_0$ and a long-wavelength parameter $\mu = (H_0/\ell_0)^2$. Then the amplitude can be written as $A_0 = \epsilon H_0$, and the wavelength as $\ell_0 = H_0/\sqrt{\mu}$. Then, we scale x and y by the wavelength, z by the rest-depth and t by the wavelength divided by the SW wave speed $\sqrt{gH_0}$. Similarly, ϕ is scaled by the SW wave speed multiplied by the wavelength multiplied by the nonlinearity parameter. For the free surface h, we write $h = H_0 + \eta$ (where η is the displacement from the rest-depth),[1] and scale η by the amplitude. Thus, denoting dimensionless quantities with $\hat{\cdot}$, the "standard" BL scalings are

$$(x, y) = \frac{H_0}{\sqrt{\mu}}(\hat{x}, \hat{y}), \quad z = H_0\hat{z}, \quad t = \sqrt{\frac{H_0}{\mu g}}\,\hat{t}, \quad \phi = \epsilon H_0\sqrt{\frac{gH_0}{\mu}}\,\hat{\phi} \text{ and } \eta = \epsilon H_0\hat{\eta},$$

with $h = H_0(1 + \epsilon\hat{\eta})$.

To augment the typical water-only BL approximation, we now need to derive appropriate scalings for the wavemaker and buoy-water-interface terms. The wavemaker BC requires R to scale like $t\phi/y$, while the Bernoulli equation (2.39b) can be used to scale λ like ϕ/t. To scale the position of the buoy and hull, the displacement $\tilde{Z}$ (analogous to η and scaled by some unknown quantity ΔZ) from their rest values Z_0 and H_b is introduced, i.e. $Z = Z_0 + \Delta Z\hat{\tilde{Z}}$ and $h_b = H_b + \Delta Z\hat{\tilde{Z}}$. Similarly, the position of the waterline is $y_b = L_b + \widetilde{y}_b = L_b + \Delta Y\hat{\widetilde{y}}_b$, where $\widetilde{y}_b$ is scaled by ΔY. Finally, W scales like $\Delta Z/t$.

Table 3.1: The scalings used in the BL approximation.

Variable Name	Scaling
Horizontal space	$(x, y) = \frac{H_0}{\sqrt{\mu}}(\hat{x}, \hat{y})$
Vertical space	$z = H_0\hat{z}$
Time	$t = \sqrt{\frac{H_0}{\mu g}}\,\hat{t}$

(continued on next page)

[1] Note that η here is slightly different to η as used in the linearisation (in particular, under the buoy).

Table 3.1 (continued)

Variable Name	Scaling
Velocity potential	$\phi = \epsilon H_0 \sqrt{\frac{gH_0}{\mu}}\hat{\phi}$
Free-surface height	$h = H_0(1 + \epsilon\hat{\eta})$
Wavemaker position	$R = \frac{\epsilon H_0}{\sqrt{\mu}}\hat{R}$
Buoy position	$Z = Z_0 + \Delta Z\hat{\tilde{Z}}$
Buoy velocity	$\dot{Z} = \sqrt{\frac{\mu g}{H_0}}\Delta Z\hat{\dot{\tilde{Z}}}$
Buoy-water interface position	$h_b = H_b + \Delta Z\hat{\tilde{Z}}$
Waterline position	$y_b = L_b + \Delta Y\hat{\tilde{y}}_b$
Buoy-water interface pressure	$\lambda = \epsilon g H_0 \hat{\lambda}$

These scalings are summarised in Table 3.1. After substitution into the Lagrangian (2.29) (with $\rho = \rho_0$ evaluated explicitly), discarding the electromagnetic terms, removing the constant MgZ_0 (which has no impact on the final variations), and dropping the hats on the non-dimensional quantities, the dimensionless 3D Lagrangian is

$$\begin{aligned}\mathcal{L} = \frac{\epsilon\rho_0 g H_0^4}{\mu} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \Bigg[\int_0^{1+\epsilon\eta} \left\{ \frac{\partial\phi}{\partial t} + \frac{\epsilon}{2}\|\nabla_H\phi\|^2 + \frac{\epsilon}{2\mu}\left(\frac{\partial\phi}{\partial z}\right)^2 + \frac{z-1}{\epsilon} \right\} \mathrm{d}z \\ + \lambda\left(1 + \epsilon\eta - \frac{1}{H_0}(H_b + \Delta Z\widetilde{Z})\right)\Theta\left(\frac{H_0}{\sqrt{\mu}}y - L_b - \Delta Y\widetilde{y}_b\right)\Bigg] \mathrm{d}y\,\mathrm{d}x \\ - \frac{\mu M g \Delta Z^2}{2H_0}\dot{\widetilde{Z}}^2 + Mg\Delta Z\widetilde{Z}. \quad (3.1)\end{aligned}$$

Now, the only terms that depend on z are functions of ϕ, and z itself. Thus, to remove the z-dependence, only the variation in ϕ needs to be considered. As in Section 2.3, derivatives of $\delta\phi$ in this variation are eliminated using the dimensionless forms of the 3D RTT (A.52) and the 3D IBP (A.48):

Initial variation, with the coefficient outside the integral removed, divided by ϵ:

$$\delta_\Phi \mathcal{S} = \int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \int_0^{1+\epsilon\eta} \left\{ \frac{1}{\epsilon}\frac{\partial \delta\phi}{\partial t} + \nabla_H \phi \cdot \nabla_H \delta\phi + \frac{1}{\mu}\frac{\partial \phi}{\partial z}\frac{\partial \delta\phi}{\partial z} \right\} \mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t.$$

Dimensionless form of 3D RTT (A.52) with $f = \delta\phi/\epsilon$:

$$\int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \int_0^{1+\epsilon\eta} \frac{1}{\epsilon}\frac{\partial \delta\phi}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t = \int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \left\{ \int_0^{1+\epsilon\eta} \delta\phi \dot{R}\,\mathrm{d}z \right\}\Bigg|_{y=\epsilon R} \mathrm{d}x\,\mathrm{d}t - \int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \delta\phi|_{z=1+\epsilon\eta} \frac{\partial \eta}{\partial t}\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t.$$

Dimensionless form of 3D IBP (A.48) with $f = \delta\phi$ and $\mathbf{F} = (\nabla_H \phi, \frac{1}{\mu}\frac{\partial \phi}{\partial z})$:

$$\begin{aligned}
&\int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \int_0^{1+\epsilon\eta} \left\{ \nabla_H \phi \cdot \nabla_H \delta\phi + \frac{1}{\mu}\frac{\partial \phi}{\partial z}\frac{\partial \delta\phi}{\partial z} \right\} \mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t \\
&\quad = -\int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \int_0^{1+\epsilon\eta} \delta\phi \left[\nabla_H^2 \phi + \frac{1}{\mu}\frac{\partial^2 \phi}{\partial z^2} \right] \mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t \\
&\qquad - \int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \left\{ \int_0^{1+\epsilon\eta} \delta\phi \frac{\partial \phi}{\partial y}\,\mathrm{d}z \right\}\Bigg|_{y=\epsilon R} \mathrm{d}x\,\mathrm{d}t \\
&\qquad + \int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \left\{ \delta\phi \left[\frac{1}{\mu}\frac{\partial \phi}{\partial z} - \epsilon \nabla_H \phi \cdot \nabla_H \eta \right] \right\}\Bigg|_{z=1+\epsilon\eta} \mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t \\
&\qquad + \int_{t_1}^{t_2} \iint_{\partial\Omega_w} \delta\phi \left(\nabla_H \phi, \frac{1}{\mu}\frac{\partial \phi}{\partial z} \right) \cdot \hat{\mathbf{n}}\,\mathrm{d}S\,\mathrm{d}t.
\end{aligned}$$

Final variation, with $\partial\Omega_w$ split into $z = 0$ and $\Gamma_w \times [0, 1 + \epsilon\eta]$:

$$\begin{aligned}
\delta_\Phi \mathcal{S} = &-\int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \int_0^{1+\epsilon\eta} \delta\phi \left[\nabla_H^2 \phi + \frac{1}{\mu}\frac{\partial^2 \phi}{\partial z^2} \right] \mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t \\
&+ \int_{t_1}^{t_2} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \left\{ \int_0^{1+\epsilon\eta} \delta\phi \left[\dot{R} - \frac{\partial \phi}{\partial y} \right] \mathrm{d}z \right\}\Bigg|_{y=\epsilon R} \mathrm{d}x\,\mathrm{d}t
\end{aligned}$$

$$+\int_{t_1}^{t_2}\int_{0}^{\sqrt{\mu}\frac{L_x}{H_0}}\int_{\epsilon R}^{l_y}\left\{\delta\phi\left[\frac{1}{\mu}\frac{\partial\phi}{\partial z}-\epsilon\nabla_H\phi\cdot\nabla_H\eta-\frac{\partial\eta}{\partial t}\right]\right\}\Big|_{z=1+\epsilon\eta}\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$-\int_{t_1}^{t_2}\int_{0}^{\sqrt{\mu}\frac{L_x}{H_0}}\int_{\epsilon R}^{l_y}\left\{\frac{\delta\phi}{\mu}\frac{\partial\phi}{\partial z}\right\}\Big|_{z=0}\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$+\int_{t_1}^{t_2}\int_{\Gamma_w}\int_{0}^{1+\epsilon\eta}\delta\phi\nabla_H\phi\cdot\hat{\mathbf{n}}\,\mathrm{d}z\,\mathrm{d}s\,\mathrm{d}t.$$

Thus, the following equations are obtained:

$$\delta\phi:\mu\nabla_H^2\phi+\frac{\partial^2\phi}{\partial z^2}=0, \tag{3.2}$$
$$\delta\phi|_{z=1+\epsilon\eta}:\frac{\partial\eta}{\partial t}+\epsilon\nabla_H\phi\cdot\nabla_H\eta=\frac{1}{\mu}\frac{\partial\phi}{\partial z}\ \text{ at }\ z=1+\epsilon\eta$$
$$\delta\phi|_{y=\epsilon R}:\frac{\partial\phi}{\partial y}=\dot{R}\ \text{ at }\ y=\epsilon R, \tag{3.3}$$
$$\delta\phi|_{z=0}:\frac{\partial\phi}{\partial z}=0\ \text{ at }\ z=0, \tag{3.4}$$
$$\delta\phi|_{\Gamma_w}:\nabla_H\phi\cdot\hat{\mathbf{n}}=0\ \text{ on }\ \Gamma_w, \tag{3.5}$$

where the stationary boundaries $\partial\Omega_w$ have been decomposed into $z=0$ and the stationary vertical walls, which themselves have been further decomposed into the Cartesian product of their perimeter (denoted Γ_w) and the vertical dimension z, i.e. $\partial\Omega_w=\{z=0\}\cup\{\Gamma_w\times[0,1+\epsilon\eta]\}$.

The second equation above (unlabelled) is not required to remove the z-dependence from the system, and can be discarded. Following the process of Bokhove and Kalogirou [76], the remaining equations shall be used to form the 2D approximation to the 3D potential.

3.2 Asymptotic expansion of velocity potential

Defining the *bottom potential* at $z = 0$ as $\Phi(x, y, t) \equiv \phi(x, y, 0, t)$, the 3D potential ϕ can be written as an asymptotic expansion in powers of μ:

$$\phi = \Phi + \sum_{i=1}^{\infty} \mu^i \phi_i, \tag{3.6}$$

where the ϕ_i are functions to be found. Then the definition of Φ and the BC at $z = 0$ (3.4) give, respectively,

$$\sum_{i=1}^{\infty} \mu^i \phi_i|_{z=0} = 0 \text{ and } \sum_{i=1}^{\infty} \mu^i \left.\frac{\partial \phi_i}{\partial z}\right|_{z=0} = 0,$$

or, since $\mu \neq 0$,

$$\phi_i|_{z=0} = \left.\frac{\partial \phi_i}{\partial z}\right|_{z=0} = 0 \text{ for all } i. \tag{3.7}$$

Next, substituting (3.6) into the Laplace equation (3.2) gives

$$\mu \left(\nabla_H^2 \Phi + \frac{\partial^2 \phi_1}{\partial z^2} \right) + \sum_{i=2}^{\infty} \mu^i \left(\nabla_H^2 \phi_{i-1} + \frac{\partial^2 \phi_i}{\partial z^2} \right) = 0,$$

which, using the BCs (3.7), generates a single PDE at each power of μ:

$$\text{at } \mathcal{O}\left(\mu^i\right), \frac{\partial^2 \phi_i}{\partial z^2} = -\nabla_H^2 \phi_{i-1} \text{ with } \phi_i|_{z=0} = \left.\frac{\partial \phi_i}{\partial z}\right|_{z=0} = 0,$$

where, for $i = 1$, $\phi_{i-1} = \phi_0 \equiv \Phi$.

The $\mathcal{O}(\mu)$ equation can be solved easily, and the solution at $\mathcal{O}(\mu^i)$ can be used to solve the $\mathcal{O}(\mu^{i+1})$ equation, such that ϕ_i can be found in terms of Φ for all i, and (3.6) can be written as an expansion of Φ in powers of z:

$$\phi = \Phi + \sum_{i=1}^{\infty} \frac{(-\mu)^i z^{2i}}{(2i)!} \nabla_H^{2i} \Phi, \tag{3.8}$$

where

$$\nabla_H^{2i} \Phi = \sum_{k=0}^{i} \binom{i}{k} \frac{\partial^{2i} \Phi}{\partial x^{2(i-k)} \partial y^{2k}}$$

is the result of taking the Laplacian of Φ i times. Substituting (3.8) into the remaining

BCs (3.3) and (3.5) gives

$$\frac{\partial \Phi}{\partial y} - \dot{R} = -\sum_{i=1}^{\infty} \frac{(-\mu)^i z^{2i}}{(2i)!} \frac{\partial}{\partial y} \left(\nabla_H^{2i} \Phi \right)$$

at the wavemaker and

$$\nabla_H \Phi \cdot \hat{\mathbf{n}} = -\sum_{i=1}^{\infty} \frac{(-\mu)^i z^{2i}}{(2i)!} \nabla_H \left(\nabla_H^{2i} \Phi \right) \cdot \hat{\mathbf{n}}$$

along the walls. Both of these expressions have a z-invariant function on the left-hand side (LHS) equal to a polynomial of z whose (variable) coefficients are also z-invariant on the RHS. The only way this is possible for all z is if each of the z-invariant coefficient functions are equal to 0. This gives the following BCs on Φ and $\nabla_H^{2i}\Phi$:

$$\frac{\partial \Phi}{\partial y} = \dot{R} \text{ at } y = \epsilon R \text{ and } \nabla_H \Phi \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_w, \tag{3.9}$$

and

$$\nabla_H \left(\nabla_H^{2i} \Phi \right) \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma, \text{ for all } i,$$

where $\Gamma = \Gamma_w \cup \{y = R\}$ is the line around the entire tank (including both the wavemaker and the vertical walls).

The expression (3.8) is the aforementioned asymptotic expansion of ϕ. The presence of the infinite sum requires the expansion to be truncated at some power of μ; truncating at $\mathcal{O}(\mu^0)$ (i.e. writing $\phi = \Phi$) results in the SW approximation, while for BL we truncate at $\mathcal{O}(\mu)$ (hence the validity of the BL approximation only for long wavelengths, when $\mu \ll 1$, which allows this truncation to take place). Thus, the dimensionless potential using in the BL approximation is

$$\phi = \Phi - \frac{\mu z^2}{2} \nabla_H^2 \Phi. \tag{3.10}$$

Note that, when re-dimensionalised, this potential is

$$\phi = \Phi - \frac{z^2}{2} \nabla_H^2 \Phi.$$

3.3 Depth-integration

The so-called BL potential (3.10) can be substituted into the dimensionless Lagrangian (3.1), with the explicit z-dependence able to be simply integrated out. This requires the following derivatives:

$$\begin{aligned}
\frac{\partial\phi}{\partial t} &= \frac{\partial\Phi}{\partial t} - \underline{\frac{\mu z^2}{2}\frac{\partial}{\partial t}\left(\nabla_H^2\Phi\right)} \\
\frac{\epsilon}{2}\|\nabla_H\phi\|^2 &= \frac{\epsilon}{2}\|\nabla_H\Phi\|^2 - \underline{\frac{\epsilon\mu z^2}{2}\nabla_H\Phi\cdot\nabla_H\left(\nabla_H^2\Phi\right)} + \cancel{\mathcal{O}\left(\mu^2\right)} \\
\frac{\epsilon}{2\mu}\left(\frac{\partial\phi}{\partial z}\right)^2 &= \underline{\frac{\epsilon\mu z^2}{2}\left(\nabla_H^2\Phi\right)^2},
\end{aligned}$$

where the underlined terms are those arising from the additional term in (3.10); as such, neglecting these terms reduces the BL approximation to the SW approximation. After substitution and subsequent evaluation of the z-integral, (3.1) becomes

$$\begin{aligned}
\mathcal{L} = \frac{\epsilon\rho_0 g H_0^4}{\mu}\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\int_{\epsilon R}^{l_y}\Bigg\{&(1+\epsilon\eta)\left(\frac{\partial\Phi}{\partial t} + \frac{\epsilon}{2}\|\nabla\Phi\|^2\right) + \frac{\epsilon\eta^2}{2} - \cancel{\frac{1}{2\epsilon}} \\
&+ \underline{\frac{\mu(1+\epsilon\eta)^3}{6}\left(\epsilon\left(\nabla^2\Phi\right)^2 - \frac{\partial}{\partial t}\left(\nabla^2\Phi\right) - \epsilon\nabla\Phi\cdot\nabla\left(\nabla^2\Phi\right)\right)} \\
&+ \lambda\left(1+\epsilon\eta - \frac{1}{H_0}(H_b + \Delta Z\widetilde{Z})\right)\Theta\left(\frac{H_0}{\sqrt{\mu}}y - L_b - \Delta Y\widetilde{y}_b\right)\Bigg\}\,\mathrm{d}y\,\mathrm{d}x \\
&- \frac{\mu M g\Delta Z^2}{2H_0}\dot{\widetilde{Z}}^2 + Mg\Delta Z\widetilde{Z},
\end{aligned}$$

where the subscript H has been dropped, since in 2D $\nabla_H \equiv \nabla$.

Now, note the terms in blue. These are the only hydrodynamic terms without a common factor of ϵ; however, using the dimensionless form of the 2D RTT (A.53) with $f = \Phi - \underline{\mu/6\nabla^2\Phi}$ on these terms replaces them with new terms with a factor of ϵ, such that ϵ can be taken as a common factor of the entire integral:

$$\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\int_{\epsilon R}^{l_y}\frac{\partial}{\partial t}\left\{\Phi - \underline{\frac{\mu}{6}\nabla^2\Phi}\right\}\mathrm{d}y\,\mathrm{d}x = \epsilon\dot{R}\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\left\{\Phi - \underline{\frac{\mu}{6}\nabla^2\Phi}\right\}\Big|_{y=\epsilon R}\mathrm{d}x.$$

$$\mathcal{L} = \frac{\epsilon^2 \rho_0 g H_0^4}{\mu} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \left[\int_{\epsilon R}^{l_y} \left\{ \eta \frac{\partial \Phi}{\partial t} + \frac{1+\epsilon\eta}{2} \|\nabla\Phi\|^2 + \frac{\eta^2}{2} - \underline{\frac{\mu\eta(3+3\epsilon\eta+\epsilon^2\eta^2)}{6} \frac{\partial}{\partial t}\left(\nabla^2\Phi\right)} \right.\right.$$
$$+ \underline{\frac{\mu(1+\epsilon\eta)^3}{6}\left(\left(\nabla^2\Phi\right)^2 - \nabla\Phi\cdot\nabla\left(\nabla^2\Phi\right)\right)}$$
$$\left.\left. + \frac{\lambda}{\epsilon}\left(1+\epsilon\eta - \frac{1}{H_0}(H_b + \Delta Z\widetilde{Z})\right)\Theta\left(\frac{H_0}{\sqrt{\mu}}y - L_b - \Delta Y\widetilde{y}_b\right) \right\} \mathrm{d}y \right.$$
$$\left. + \dot{R}\left\{\Phi - \underline{\frac{\mu}{6}\nabla^2\Phi}\right\}\Big|_{y=\epsilon R} \right] \mathrm{d}x - \frac{\mu M g \Delta Z^2}{2H_0}\dot{\widetilde{Z}}^2 + Mg\Delta Z\widetilde{Z}.$$

Next, note we have higher powers of ϵ and μ. Under the weakly-nonlinear and long-wavelength assumptions, we have $\epsilon \ll 1$ and $\mu \ll 1$ and we can neglect these higher-order terms, retaining only those linear in ϵ and μ. This gives

$$\mathcal{L} = \frac{\epsilon^2 \rho_0 g H_0^4}{\mu} \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \left[\int_{\epsilon R}^{l_y} \left\{ \eta \frac{\partial}{\partial t}\left(\Phi - \underline{\frac{\mu}{2}\nabla^2\Phi}\right) + \frac{1+\epsilon\eta}{2}\|\nabla\Phi\|^2 + \frac{\eta^2}{2} \right.\right.$$
$$+ \underline{\frac{\mu}{6}\left(\left(\nabla^2\Phi\right)^2 - \nabla\Phi\cdot\nabla\left(\nabla^2\Phi\right)\right)}$$
$$\left.\left. + \frac{\lambda}{\epsilon}\left(1+\epsilon\eta - \frac{1}{H_0}(H_b + \Delta Z\widetilde{Z})\right)\Theta\left(\frac{H_0}{\sqrt{\mu}}y - L_b - \Delta Y\widetilde{y}_b\right) \right\} \mathrm{d}y \right.$$
$$\left. + \dot{R}\left\{\Phi - \underline{\frac{\mu}{6}\nabla^2\Phi}\right\}\Big|_{y=\epsilon R} \right] \mathrm{d}x - \frac{\mu M g \Delta Z^2}{2H_0}\dot{\widetilde{Z}}^2 + Mg\Delta Z\widetilde{Z}. \quad (3.11)$$

Finally, the red terms can be simplified through the use of the dimensionless form of Green's first identity (2D) (A.50) with $f = \nabla^2\Phi$ and $g = \Phi$, together with the BCs (3.9):

$$\int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \nabla\Phi\cdot\nabla\left(\nabla^2\Phi\right)\mathrm{d}y\,\mathrm{d}x + \int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \nabla^2\Phi|_{y=\epsilon R}\dot{R}\,\mathrm{d}x$$
$$= -\int_0^{\sqrt{\mu}\frac{L_x}{H_0}} \int_{\epsilon R}^{l_y} \left(\nabla^2\Phi\right)^2 \mathrm{d}y\,\mathrm{d}x + \cancel{\int_{P_w} \nabla^2\Phi\nabla\Phi\cdot\hat{\mathbf{n}}\,\mathrm{d}s}.$$

This results in a dimensionless 2D Lagrangian of

$$\mathcal{L} = \frac{\epsilon^2\rho_0 g H_0^4}{\mu}\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\Bigg[\int_{\epsilon R}^{l_y}\Bigg\{\eta\frac{\partial}{\partial t}\left(\Phi - \underline{\frac{\mu}{2}\nabla^2\Phi}\right) + \frac{1+\epsilon\eta}{2}\|\nabla\Phi\|^2 + \frac{\eta^2}{2} + \underline{\frac{\mu}{3}\left(\nabla^2\Phi\right)^2} + \frac{\lambda}{\epsilon}\left(1+\epsilon\eta - \frac{1}{H_0}(H_b + \Delta Z\widetilde{Z})\right)\Theta\left(\frac{H_0}{\sqrt{\mu}}y - L_b - \Delta Y\widetilde{y}_b\right)\Bigg\}\mathrm{d}y + \Phi|_{y=\epsilon R}\dot{R}\Bigg]\mathrm{d}x - \frac{\mu M g\Delta Z^2}{2H_0}\dot{\widetilde{Z}}^2 + Mg\Delta Z\widetilde{Z}.$$

To finish, let us re-dimensionalise this Lagrangian. This is made simpler by transferring the time derivatives of Φ (blue terms) onto η via the dimensionless form of the 2D RTT (A.53), with $f = \eta(\Phi - \underline{\mu/2}\nabla^2\Phi)$:

$$\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\int_{\epsilon R}^{l_y}\eta\frac{\partial}{\partial t}\left\{\Phi - \underline{\frac{\mu}{2}\nabla^2\Phi}\right\}\mathrm{d}y\,\mathrm{d}x = \int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\int_{\epsilon R}^{l_y}\left\{\underline{\frac{\mu}{2}\nabla^2\Phi} - \Phi\right\}\frac{\partial\eta}{\partial t}\mathrm{d}y\,\mathrm{d}x + \dot{R}\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\left\{\epsilon\eta\Phi - \cancel{\frac{\epsilon\mu\eta}{2}\nabla^2\Phi}\right\}\Big|_{y=\epsilon R}\mathrm{d}x.$$

$$\mathcal{L} = \frac{\epsilon^2\rho_0 g H_0^4}{\mu}\int_0^{\sqrt{\mu}\frac{L_x}{H_0}}\Bigg[\int_{\epsilon R}^{l_y}\Bigg\{\left(\underline{\frac{\mu}{2}\nabla^2\Phi} - \Phi\right)\frac{\partial\eta}{\partial t} + \frac{1+\epsilon\eta}{2}\|\nabla\Phi\|^2 + \frac{\eta^2}{2} + \underline{\frac{\mu}{3}\left(\nabla^2\Phi\right)^2} + \frac{\lambda}{\epsilon}\left(1+\epsilon\eta - \frac{1}{H_0}(H_b + \Delta Z\widetilde{Z})\right)\Theta\left(\frac{H_0}{\sqrt{\mu}}y - L_b - \Delta Y\widetilde{y}_b\right)\Bigg\}\mathrm{d}y + ((1+\epsilon\eta)\Phi)|_{y=\epsilon R}\dot{R}\Bigg]\mathrm{d}x - \frac{\mu M g\Delta Z^2}{2H_0}\dot{\widetilde{Z}}^2 + Mg\Delta Z\widetilde{Z}.$$

This is now ready to re-dimensionalise:

$$\mathcal{L} = \rho_0\int_0^{L_x}\Bigg[\int_R^{l_y}\Bigg\{\left(\underline{\frac{H_0^2}{2}\nabla^2\Phi} - \Phi\right)\frac{\partial h}{\partial t} + \frac{h}{2}\|\nabla\Phi\|^2 + \frac{g(h^2 - 2H_0h + \cancel{H_0^2})}{2} + \underline{\frac{H_0^3}{3}\left(\nabla^2\Phi\right)^2} + \lambda(h-h_b)\Theta(y-y_b)\Bigg\}\mathrm{d}y + (h\Phi)|_{y=R}\dot{R}\Bigg]\mathrm{d}x$$

$$-\frac{1}{2}M\dot{Z}^2 + Mg(Z - \cancel{Z_0}).$$

Finally, we shall transfer the time derivative back from h to Φ using the 2D RTT (A.53) with $f = h(\underline{H_0^2/2\nabla^2\Phi} - \Phi)$:

$$\int_0^{L_x}\int_R^{l_y}\left(\underline{\frac{H_0^2}{2}\nabla^2\Phi} - \Phi\right)\frac{\partial h}{\partial t}\,\mathrm{d}y\,\mathrm{d}x + \int_0^{L_x}(h\Phi)|_{y=R}\dot{R}\,\mathrm{d}x$$
$$= \int_0^{L_x}\int_R^{l_y} h\frac{\partial}{\partial t}\left(\Phi - \underline{\frac{H_0^2}{2}\nabla^2\Phi}\right)\mathrm{d}y\,\mathrm{d}x + \underline{\frac{H_0^2\dot{R}}{2}\cancel{\int_0^{L_x}(h\nabla^2\Phi)|_{y=R}\,\mathrm{d}x}}.$$

Note that, in the dimensionless form, the boundary term arising from the underlined term was neglected as it was nonlinear in ϵ and μ; as such, that same term must again be neglected in dimensional form. After this step, the final dimensional 2D BL Lagrangian, with the electromagnetic terms recovered, is

$$\mathcal{L}[\mathbf{v}] = \rho_0\int_0^{L_x}\int_R^{l_y}\left\{h\frac{\partial}{\partial t}\left(\Phi - \underline{\frac{H_0^2}{2}\nabla^2\Phi}\right) + \frac{h}{2}\|\nabla\Phi\|^2 + \underline{\frac{H_0^3}{3}\left(\nabla^2\Phi\right)^2}\right.$$
$$\left. + \frac{gh(h-2H_0)}{2} + \lambda(h-h_b)\Theta(y-y_b)\right\}\mathrm{d}y\,\mathrm{d}x$$
$$-\frac{1}{2}M\dot{Z}^2 + MgZ - \frac{1}{2}L_i\dot{Q}^2 + \frac{Q^2}{2C} - \gamma G\dot{Z}Q, \tag{3.12}$$

with unknowns

$$\mathbf{v} = (h, \Phi, \lambda, Z, Q). \tag{3.13}$$

The 2D BL Hamiltonian is

$$\mathcal{H} = \rho_0\int_0^{L_x}\int_{R(t)}^{l_y}\left\{\frac{h}{2}\|\nabla\Phi\|^2 + \underline{\frac{H_0^3}{3}\left(\nabla^2\Phi\right)^2} + \frac{gh(h-2H_0)}{2}\right\}\mathrm{d}y\,\mathrm{d}x$$
$$+\frac{1}{2}M\dot{Z}^2 + MgZ + \frac{1}{2}L_i\dot{Q}^2 + \frac{Q^2}{2C},$$

and the energy balance is

$$\frac{\partial \mathcal{H}}{\partial t} = -\rho_0 \dot{R} \int_0^{L_x} \left\{ \frac{h}{2} \|\nabla \Phi\|^2 + \underline{\frac{H_0^3}{3} \left(\nabla^2 \Phi\right)^2} + \frac{gh(h-2H_0)}{2} \right\}\Bigg|_{y=R} \mathrm{d}x.$$

As with its 3D counterpart (2.29), taking the variation of $\mathcal{S}$ using (3.12) in the unknowns (3.13) will yield a system of 2D equations, approximating the 3D potential flow system (2.39) via the BL approximation for the potential

$$\phi = \Phi - \frac{z^2}{2} \nabla_H^2 \Phi.$$

Note that removing the underlined terms in (3.12) gives the SW Lagrangian, which can be obtained directly from (2.29) by writing $\phi = \Phi$ before explicitly evaluating the z-integral.

3.4 2D equations

In the 2D case, the variations in the unknowns λ, Z and Q are identical to their 3D counterparts, while the variation in h is trivial to find. However, as we have seen many times before, the variation in Φ produces derivatives of $\delta\Phi$, which must be transferred: the time derivatives are handled using the 2D RTT (A.53), the gradient via the 2D IBP (A.49), and the Laplacian terms by applying Green's second identity (A.51):

Initial variation:

$$\delta_\Phi \mathcal{S} = \rho_0 \int_{t_1}^{t_2} \int_0^{L_x} \int_R^{l_y} \left\{ h \frac{\partial}{\partial t} \left(\delta\Phi - \underline{\frac{H_0^2}{2} \nabla^2 \delta\Phi} \right) + h \nabla\Phi \cdot \nabla \delta\Phi \right.$$
$$\left. + \underline{\frac{2H_0^3}{3} \nabla^2 \Phi \nabla^2 \delta\Phi} \right\} \mathrm{d}y \, \mathrm{d}x \, \mathrm{d}t.$$

Dimensional form of 2D RTT (A.53) with $f = h(\delta\Phi - \underline{H_0^2/2 \nabla^2 \delta\Phi})$, neglecting the underlined boundary term:

$$\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y} h\frac{\partial}{\partial t}\left\{\delta\Phi - \underline{\frac{H_0^2}{2}\nabla^2\delta\Phi}\right\}\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$= \int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\left\{\underline{\frac{H_0^2}{2}\nabla^2\delta\Phi} - \delta\Phi\right\}\frac{\partial h}{\partial t}\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$+ \int_{t_1}^{t_2}\int_0^{L_x}\left\{\delta\Phi h\dot{R} - \cancel{\frac{H_0^2\dot{R}}{2}h\nabla^2\delta\Phi}\right\}\Big|_{y=R}\mathrm{d}x\,\mathrm{d}t.$$

2D IBP (A.49) with $f = \delta\Phi$ and $\mathbf{F} = h\nabla\Phi$:

$$\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y} h\nabla\Phi\cdot\nabla\delta\Phi\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t = -\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\delta\Phi\nabla\cdot(h\nabla\Phi)\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$-\int_{t_1}^{t_2}\int_0^{L_x}\left\{\delta\Phi h\frac{\partial\Phi}{\partial y}\right\}\Big|_{y=R}\mathrm{d}x\,\mathrm{d}t$$
$$+\int_{t_1}^{t_2}\int_{\Gamma_w}\delta\Phi h\nabla\Phi\cdot\hat{\mathbf{n}}\,\mathrm{d}s\,\mathrm{d}t.$$

Intermediate variation:

$$\delta_\Phi\mathcal{S} = \rho_0\int_{t_1}^{t_2}\Bigg(\int_0^{L_x}\Bigg[\int_R^{l_y}\Bigg\{\underline{\left(\frac{H_0^2}{2}\frac{\partial h}{\partial t} + \frac{2H_0^3}{3}\nabla^2\Phi\right)\nabla^2\delta\Phi}$$
$$-\delta\Phi\left[\frac{\partial h}{\partial t} + \nabla\cdot(h\nabla\Phi)\right]\Bigg\}\mathrm{d}y$$
$$+\left\{\delta\Phi h\left[\dot{R} - \frac{\partial\Phi}{\partial y}\right]\right\}\Big|_{y=R}\Bigg]\mathrm{d}x$$
$$+\int_{\Gamma_w}\delta\Phi h\nabla\Phi\cdot\hat{\mathbf{n}}\,\mathrm{d}s\Bigg)\mathrm{d}t.$$

Green's second identity (A.51) with $g = \frac{H_0^2}{2}\frac{\partial h}{\partial t} + \frac{2H_0^3}{3}\nabla^2\Phi$ and $h = \delta\Phi$, noting that, since $\nabla\Phi$ is known on Γ, there is no change in the function there, so $\nabla\delta\Phi\cdot\hat{\mathbf{n}} = 0$:

$$\int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\left(\frac{H_0^2}{2}\frac{\partial h}{\partial t} + \frac{2H_0^3}{3}\nabla^2\Phi\right)\nabla^2\delta\Phi\,\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$= \int_{t_1}^{t_2}\int_0^{L_x}\int_R^{l_y}\delta\Phi\left[\frac{H_0^2}{2}\frac{\partial}{\partial t}\left(\nabla^2 h\right) + \frac{2H_0^3}{3}\nabla^4\Phi\right]\mathrm{d}y\,\mathrm{d}x\,\mathrm{d}t$$
$$+\int_{t_1}^{t_2}\int_\Gamma\cancel{\left[\frac{H_0^2}{2}\frac{\partial h}{\partial t} + \frac{2H_0^3}{3}\nabla^2\Phi\right]}\nabla\delta\Phi\cdot\hat{\mathbf{n}}\,\mathrm{d}s\,\mathrm{d}t$$
$$-\int_{t_1}^{t_2}\int_\Gamma\delta\Phi\left[\frac{H_0^2}{2}\frac{\partial}{\partial t}(\nabla h) + \frac{2H_0^3}{3}\nabla\left(\nabla^2\Phi\right)\right]\cdot\hat{\mathbf{n}}\,\mathrm{d}s\,\mathrm{d}t.$$

This process yields the following:

$$\delta_\Phi \mathcal{S} = -\rho_0 \int_{t_1}^{t_2} \left(\int_0^{L_x} \left[\int_R^{l_y} \left\{ \delta\Phi \left[\frac{\partial}{\partial t} \left(h - \underline{\frac{H_0^2}{2} \nabla^2 h} \right) + \nabla \cdot (h \nabla \Phi) - \underline{\frac{2H_0^3}{3} \nabla^4 \Phi} \right] \right\} \mathrm{d}y \right.\right.$$
$$\left.\left. + \left\{ \delta\Phi h \left[\frac{\partial \Phi}{\partial y} - \dot{R} \right] \right\} \Big|_{y=R} \right] \mathrm{d}x$$
$$- \int_{\Gamma_w} \delta\Phi h \nabla\Phi \cdot \hat{\mathbf{n}} \, \mathrm{d}s + \underline{\int_\Gamma \delta\Phi \left[\frac{H_0^2}{2} \frac{\partial}{\partial t} (\nabla h) + \frac{2H_0^3}{3} \nabla \left(\nabla^2 \Phi \right) \right] \cdot \hat{\mathbf{n}} \, \mathrm{d}s} \Bigg) \mathrm{d}t,$$

together with the variation in h,

$$\delta_h \mathcal{S} = \rho_0 \int_{t_1}^{t_2} \int_0^{L_x} \int_R^{l_y} \delta h \left[\frac{\partial}{\partial t} \left(\Phi - \underline{\frac{H_0^2}{2} \nabla^2 \Phi} \right) + \frac{1}{2} \|\nabla\Phi\|^2 \right.$$
$$\left. + g(h - H_0) + \lambda \Theta(y - y_b) \right] \mathrm{d}y \, \mathrm{d}x \, \mathrm{d}t.$$

Note how this process produces a fourth-order derivative on Φ; this can be reduced by introducing an auxiliary function $q = -2/3 \nabla^2 \Phi$.

Therefore, the equations derived via the 2D BL Lagrangian (3.12) are

Hydrodynamics:

$$\frac{\partial}{\partial t} \left(\Phi - \underline{\frac{H_0^2}{2} \nabla^2 \Phi} \right) + \frac{1}{2} \|\nabla\Phi\|^2 + g(h - H_0) + \lambda \Theta(y - y_b) = 0, \tag{3.14a}$$

$$\frac{\partial}{\partial t} \left(h - \underline{\frac{H_0^2}{2} \nabla^2 h} \right) + \nabla \cdot (h \nabla \Phi) + \underline{H_0^3 \nabla^2 q} = 0, \tag{3.14b}$$

$$\underline{q = -\frac{2}{3} \nabla^2 \Phi}, \tag{3.14c}$$

Buoy dynamics:

$$M\dot{W} = -Mg - \gamma G I + \rho_0 \int_0^{L_x} \int_R^{l_y} \lambda \Theta(y - y_b) \, \mathrm{d}y \, \mathrm{d}x, \tag{3.14d}$$

$$\dot{Z} = W, \tag{3.14e}$$

Electromagnetic generator:

$$\dot{Q} = I, \tag{3.14f}$$

$$L_i \dot{I} = \gamma G W - \frac{Q}{C} - I(R_i + R_c) - V, \tag{3.14g}$$

Free-surface-equals-buoy-shape constraint:

$$h = h_b \text{ for } y \geq y_b, \tag{3.14h}$$

Boundary conditions:

$$\lambda = 0 \text{ at } y = y_b, \tag{3.14i}$$

$$\frac{\partial \Phi}{\partial y} = \dot{R} \text{ at } y = R, \tag{3.14j}$$

$$\nabla \Phi \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_w, \tag{3.14k}$$

$$\underline{\nabla q \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma}, \tag{3.14l}$$

$$\underline{\frac{\partial}{\partial t}(\nabla h \cdot \hat{\mathbf{n}}) = 0, \text{ or } \nabla h \cdot \hat{\mathbf{n}} = f(x,y), \text{ on } \Gamma}, \tag{3.14m}$$

with $f(x,y) = \nabla H \cdot \hat{\mathbf{n}}$ given by the initial condition on h.

The system (3.14) approximates the 3D potential-flow system (2.39) via the BL approximation. Furthermore, removing the underlined terms in (3.14a) and (3.14b) and the underlined equations (3.14c), (3.14l) and (3.14m) reduces the system to the SW equations, whose linearisation matches (23) from [44] (after the Legendre transformation in Appendix A.4 is applied). These additional terms introduce dispersive gravity waves into the problem, resulting in a more accurate approximation of the 3D potential-flow equations than is provided by the SW approximation. This increased accuracy comes at the cost of introducing higher-order spatial derivatives to the problem, requiring the additional BCs.

3.5 Linearisation

In the same way as its 3D counterpart (2.39), the nonlinear 2D system (3.14) can be linearised using the following decompositions:

$$\begin{gathered}
\Phi(x,y,t) = 0 + \widetilde{\Phi}(x,y,t), \quad q(x,y,t) = 0 + \widetilde{q}(x,y,t), \quad h(x,y,t) = H(y) + \eta(x,y,t), \\
R(t) = 0 + \widetilde{R}(t), \quad \lambda(x,y,t) = \Lambda(y) + \widetilde{\lambda}(x,y,t), \quad y_b(x,t) = L_b + \widetilde{y}_b(x,t), \\
W(t) = 0 + \widetilde{W}(t), \quad Z(t) = Z_0 + \widetilde{Z}(t), \quad Q(t) = 0 + \widetilde{Q}(t), \\
I(t) = 0 + \widetilde{I}(t).
\end{gathered}$$

Following near-identical steps as in Appendix A.5.1, the linearised BL system is:

Hydrodynamics:

$$\frac{\partial}{\partial t}\left(\Phi - \frac{H_0^2}{2}\nabla^2\Phi\right) + g\eta + \lambda\Theta(y - L_b) = 0, \tag{3.15a}$$

$$\frac{\partial}{\partial t}\left(\eta - \frac{H_0^2}{2}\nabla^2\eta\right) + \nabla\cdot(H\nabla\Phi) + H_0^3\nabla^2 q = 0, \tag{3.15b}$$

$$q = -\frac{2}{3}\nabla^2\Phi, \tag{3.15c}$$

Buoy dynamics:

$$M\dot{W} = -\gamma G(Z_0)I + \rho_0\int_0^{L_x}\int_0^{l_y}\lambda\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x, \tag{3.15d}$$

$$\dot{Z} = W, \tag{3.15e}$$

Electromagnetic generator:

$$\dot{Q} = I, \tag{3.15f}$$

$$L_i\dot{I} = \gamma G(Z_0)W - \frac{Q}{C} - I(R_i + R_c + R_l), \tag{3.15g}$$

Free-surface-equals-buoy-shape constraint:

$$\eta = Z \text{ for } y \geq L_b, \tag{3.15h}$$

Boundary conditions:

$$\lambda = g(\eta - Z) \text{ at } y = L_b, \tag{3.15i}$$

$$\frac{\partial\Phi}{\partial y} = \dot{R} \text{ at } y = 0, \tag{3.15j}$$

$$\nabla\Phi\cdot\hat{\mathbf{n}} = 0 \text{ on } \Gamma_w, \tag{3.15k}$$

$$\nabla q\cdot\hat{\mathbf{n}} = 0 \text{ on } \Gamma, \tag{3.15l}$$

$$\nabla\eta\cdot\hat{\mathbf{n}} = 0 \text{ on } \Gamma. \tag{3.15m}$$

In the same way that the "quadricised" 3D Lagrangian (2.43) for the linear 3D system (2.42) was formed (see Appendix A.5.2), a Lagrangian for the linear 2D system (3.15) can also be obtained, by "quadricising" the nonlinear 2D Lagrangian (3.12):

$$\mathcal{L} = \rho_0\int_0^{L_x}\left[\int_0^{l_y}\left\{\eta\frac{\partial}{\partial t}\left(\Phi - \frac{H_0^2}{2}\nabla^2\Phi\right) + \frac{H}{2}\|\nabla\Phi\|^2 + \frac{1}{2}g\eta^2 + \frac{H_0^3}{3}\left(\nabla^2\Phi\right)^2\right.\right.$$

$$+ \lambda(\eta - Z)\Theta(y - L_b)\Big\} \mathrm{d}y - H_0 R \left.\frac{\partial \Phi}{\partial t}\right|_{y=0}\Bigg] \mathrm{d}x$$
$$- \frac{1}{2}M\dot{Z}^2 - \frac{1}{2}L_i\dot{Q}^2 + \frac{Q^2}{2C} - \gamma G(Z_0)\dot{Z}Q. \tag{3.16}$$

This has Hamiltonian

$$\mathcal{H} = \rho_0 \int_0^{L_x} \left\{ \int_0^{l_y} \left\{ \frac{H}{2}\|\nabla\Phi\|^2 + \frac{1}{2}g\eta^2 + \underline{\frac{H_0^3}{3}\left(\nabla^2\Phi\right)^2} \right\} \mathrm{d}y + H_0\dot{R}\Phi|_{y=0} \right\} \mathrm{d}x$$
$$+ \frac{1}{2}M\dot{Z}^2 + \frac{1}{2}L_i\dot{Q}^2 + \frac{Q^2}{2C},$$

and energy balance

$$\frac{\partial \mathcal{H}}{\partial t} = \rho_0 H_0 \ddot{R} \int_0^{L_x} \Phi|_{y=0}\, \mathrm{d}x.$$

The Lagrangian (3.16) can also be obtained by starting with the "quadricised" 3D Lagrangian (2.43) and carrying out a similar non-dimensionalisation and asymptotic expansion. Therefore, we note the commutativity between linearising and taking variations, and linearising and depth-averaging. When beginning with the nonlinear 3D Lagrangian (2.29), there are three different paths one can take to obtain the linear 2D system (3.15): 1) depth-average, then take variations, then linearise; 2) depth-average, then "quadricise", then take variations; and 3) "quadricise", then depth-average, then take variations. We note, however, that depth-averaging and taking variations do not commute, since depth-averaging requires the integrals that taking variations removes.

Now, through the manipulation of a number of equations in (3.15), a system of equations for λ and $\partial q/\partial t$ can be generated. First, consider (3.15b) for $y \geq L_b$; differentiating in time and using (3.15d), (3.15e) and (3.15h) gives

$$-\frac{\gamma G(Z_0)I}{M} + \frac{\rho_0}{M}\int_0^{L_x}\int_0^{l_y} \lambda\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x - \underline{\frac{H_0^2}{2}\frac{\partial^2}{\partial t^2}\left(\cancel{\nabla^2 Z}\right)}$$
$$+ \nabla\cdot\left(H\nabla\left(\frac{\partial\Phi}{\partial t}\right)\right) + \underline{H_0^3\nabla^2\left(\frac{\partial q}{\partial t}\right)} = 0 \quad \text{for} \quad y \geq L_b. \tag{3.17}$$

Next, differentiating (3.15c) gives

$$\underline{\frac{\partial q}{\partial t} = -\frac{2}{3}\nabla^2\left(\frac{\partial \Phi}{\partial t}\right)}, \tag{3.18}$$

or, equivalently,

$$\underline{\frac{\partial}{\partial t}\left(\frac{H_0^2}{2}\nabla^2\Phi\right) = -\frac{3H_0^2}{4}\frac{\partial q}{\partial t}};$$

then (3.15a) can be written as

$$\frac{\partial \Phi}{\partial t} = -\underline{\frac{3H_0^2}{4}\frac{\partial q}{\partial t}} - g\eta - \lambda\Theta(y - L_b). \tag{3.19}$$

Finally, substituting (3.19) into (3.17) and (3.18) gives

$$\begin{aligned}\nabla\cdot(H\nabla\lambda) - \frac{\rho_0}{M}\int_0^{L_x}\int_0^{l_y}\lambda\Theta(y-L_b)\,\mathrm{d}y\,\mathrm{d}x = \\ -\nabla\cdot(gH\nabla\eta) - \frac{\gamma G(Z_0)I}{M} - \underline{\frac{3H_0^2}{4}\nabla\cdot\left(H\nabla\left(\frac{\partial q}{\partial t}\right)\right)} + \underline{H_0^3\nabla^2\left(\frac{\partial q}{\partial t}\right)} \text{ for } y \geq L_b;\end{aligned} \tag{3.20a}$$

$$\underline{\frac{\partial}{\partial t}\left(q - \frac{H_0^2}{2}\nabla^2 q\right) = \frac{2g}{3}\nabla^2\eta + \frac{2}{3}\nabla^2\left(\lambda\Theta(y-L_b)\right)}, \tag{3.20b}$$

which can be used in-place of the constraint (3.15h) when solving (3.15). It needs to be solved as a system of equations, something that a suitable numerical method is readily able to do. Note that the system reduces to (23h) in [44] when the underlined terms are removed. As a consistency check, (3.15h) is calculated during the course of the numerical solution, noting that we would expect to see values close to machine precision (i.e. 10^{-16}).

3.6 Numerical solution, results and discussion

The system (3.15) was solved using the finite element method (FEM) and the symplectic Euler (SE) time-stepping method (for a detailed discussion of these methods, see Chapter 4). A version of the system with the underlined terms/equations dropped (i.e. the SW equations) was solved by Bokhove et al. [44, 45], using an in-house MATLAB code;

this same code was extended to incorporate the additional BL terms arising presently.

Table 3.2: The parameter values used in the numerical simulations.

Parameter	Value	Parameter	Value
L_x	$0.2\,\mathrm{m}$	L_y	$2.0\,\mathrm{m}$
d	$0.27\,\mathrm{m}$	ρ_0	$997\,\mathrm{kg\,m^{-3}}$
H_0	$0.1\,\mathrm{m}$	g	$9.81\,\mathrm{m\,s^{-2}}$
M	$0.1\,\mathrm{kg}$	α	$\approx 0.642\,\mathrm{rad}$ (see note)[2]
L_m	$0.04\,\mathrm{m}$	A_m	$0.032\,\mathrm{m}$
H_m	$0.2\,\mathrm{m}$	m	$0.1\,\mathrm{A\,m^2}$
L	$0.08\,\mathrm{m}$	a	$0.04\,\mathrm{m}$
α_h	0.2	D	$2.769 \times 10^{-4}\,\mathrm{m}$
σ	$5.96 \times 10^{7}\,\Omega^{-1}\,\mathrm{m}^{-1}$	K	0.53
μ_0	$4\pi \times 10^{-7}\,\mathrm{H\,m^{-1}}$	C	∞ (see note)[3]
n_q	1	V_T	$2.05\,\mathrm{V}$
I_{sat}	$0.02\,\mathrm{A}$		

The system was solved using parameter values outlined in Table 3.2.[4] These parameters give rise to the following calculated values:

$$L_c \approx 0.251\,\mathrm{m} \quad \theta \approx 68.3^\circ \quad Z_0 \approx 0.025{,}0\,\mathrm{m} \quad L_b \approx 1.90\,\mathrm{m}$$

$$n_c \approx 289 \quad L_i \approx 3.49 \times 10^{-3}\,\mathrm{H} \quad R_i \approx 20.2\,\Omega \quad R_c = R_i$$

$$R_l = 102.5\,\Omega \quad \gamma \approx 3.63 \times 10^{-7}\,\mathrm{kg\,m^4\,s^{-2}\,A^{-1}} \quad G(Z_0) \approx 14{,}200\,\mathrm{m^{-3}}.$$

To compare the performance of the BL approximation with the SW approximation, both

[2] In the MATLAB code, this parameter is not provided, but is instead calculated. The waterline $y = L_b$ must lie on one of the straight lateral lines of the mesh (see Figure 5.1). To ensure this, one of the lines (in this case, the one closest to $y = 1.9$) is chosen, and α is subsequently calculated by rearranging (2.13), such that the chosen value for L_b arises. This is a major limitation to the MATLAB code, in that, for a given mesh, only particular parameter combinations resulting in a valid value for L_b are permitted. To this end, an improved methodology utilising a more-flexible mesh-construction algorithm has been developed (see Section 5.2.2), allowing any parameter combination, regardless of mesh—providing, of course, that L_b remains inside the contraction. Note also that the value for α quoted in Bolton et al. [1] is incorrect.

[3] No capacitor is modelled.

[4] These are the parameter values used in Bolton et al. [1] and Bokhove et al. [44, 45]; however, a more realistic collection has since been created. Presented in Table 6.1, this was used in Bokhove et al. [2, 72].

systems were solved. The simulations were driven by the precise form of the wavemaker motion $\dot{R}(t)$, given by

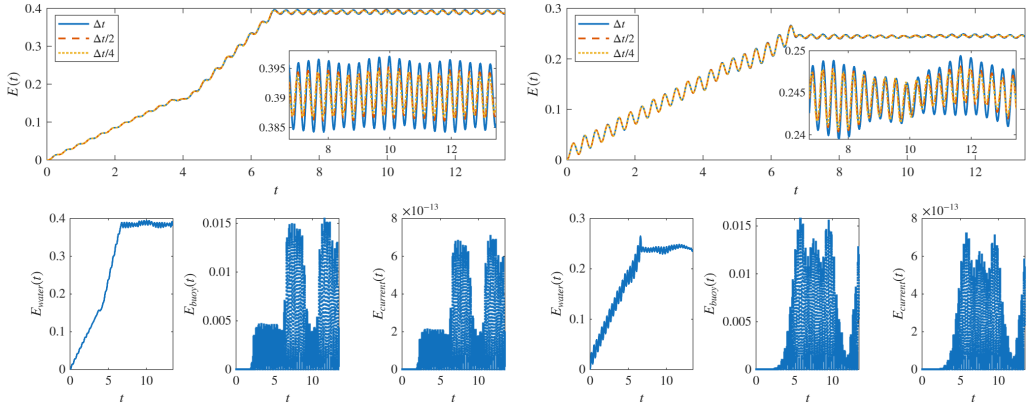
$$\dot{R} = \begin{cases} A \sin \omega t & 0 < t < 10P \\ 0 & 10P < t < T \end{cases},$$

with amplitude A, frequency ω, period $P = 2\pi/\omega$ and final time T. That is, the wavemaker was switched on for 10 periods, before being switched off and the simulation continuing.

The mesh used contained $N_x = 10$ elements in the x-direction and $N_y = 50$ elements in the y-direction, with 500 total elements in the rectangular part, and another 55 in the contraction. These values yielded step-sizes of $\Delta x = L_x/N_x = 0.02\,\text{m}$ and $\Delta y = (L_y - L_c)/N_y \approx 0.035{,}0\,\text{m}$, with a step-size in the contraction of $\Delta yc = L_c/N_x \approx 0.025{,}1\,\text{m}$. Referring to the stability analysis in Section 4.2.3, we use a time-step Δt of

$$\Delta t = 0.6 \left(\frac{2}{\pi \sqrt{g H_0} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}} \right),$$

or $\Delta t \approx 6.70 \times 10^{-3}\,\text{s}$.



(a) SW **(b)** BL

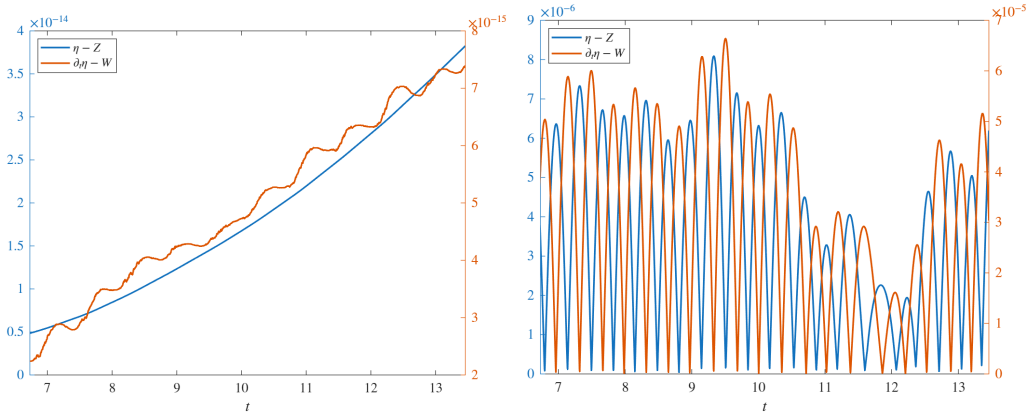
Figure 3.1: The total energy in the system over time (upper) split into the energies of the three sub-systems water (lower-left), buoy (lower-centre) and generator (lower-right), for both the SW and BL simulations. The inset panel demonstrates the temporal convergence for the three time-steps Δt, $\Delta t/2$ and $\Delta t/4$.

Initially, for both SW and BL, a single simulation with

$$\omega = \sqrt{gH_0}\frac{6\pi}{L_y} = \omega_6 \approx 9.33\,\text{Hz} \;\; \text{(see note)}^{5} \quad \text{and} \quad A = \frac{\omega_6(L_y - L_c)}{250} = A_0 \approx 0.065{,}3\,\text{m}$$

was carried out, with a final time of $T = 20P \approx 13.4\,\text{s}$. To demonstrate temporal convergence, simulations with $\Delta t \to \Delta t/2$ and $\Delta t \to \Delta t/4$ were also undertaken.

Figure 3.1 shows the energy evolution of the two simulations over time. When taking into account the influx/outflux of energy from both the wavemaker and the losses associated with the circuit resistance and harvested energy, both models are seen to be energy-conserving. In both cases, the energy increases while the wavemaker is turned on, and then stays constant after it is turned off, albeit it with small oscillations due to both numerical error and the oscillatory behaviour of the wavemaker. These oscillations can be seen to reduce as Δt is decreased.

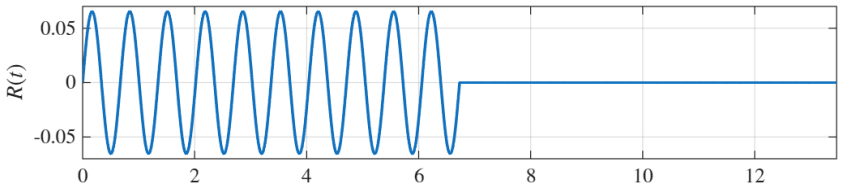


(a) SW **(b)** BL

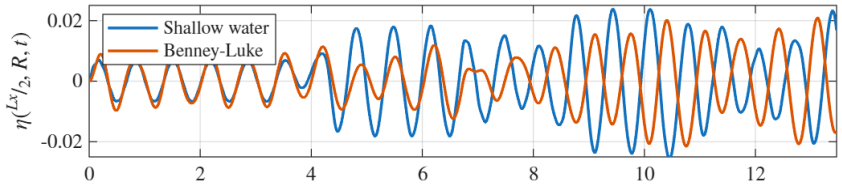
Figure 3.2: The maximum absolute value of the constraint (3.15h) and its time derivative, over time, for both simulations. In the SW case, the constraints hold up to machine precision; for the BL case, they are much less accurate, instead displaying similar wave-forms to the rest of the system.

Figure 3.2 shows the maximum absolute value of the constraint (3.15h) and its time derivative, over time. In the SW case, these constraints are obeyed up to machine precision $\mathcal{O}\left(10^{-14}, 10^{-15}\right)$; however, in the BL case they are obeyed only up-to $\mathcal{O}\left(10^{-6}, 10^{-5}\right)$, and display the same oscillatory behaviour as the rest of the system.

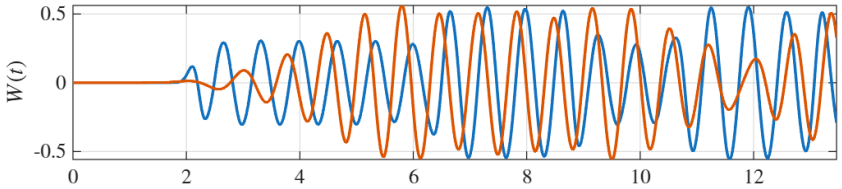
[5] I.e. such a frequency as to create a travelling wave with 6 half-wavelengths fitting into the domain $y \in [0, L_y]$. In the code, the value 6 is the mode number m_y.



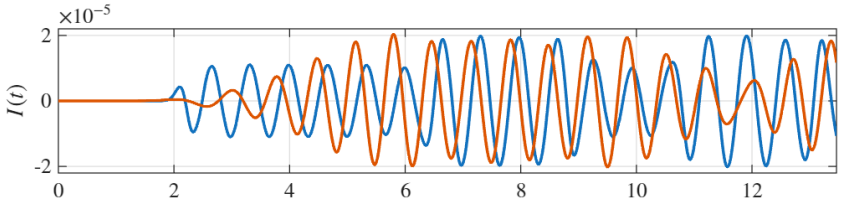
(a) Wavemaker position $R(t)$.



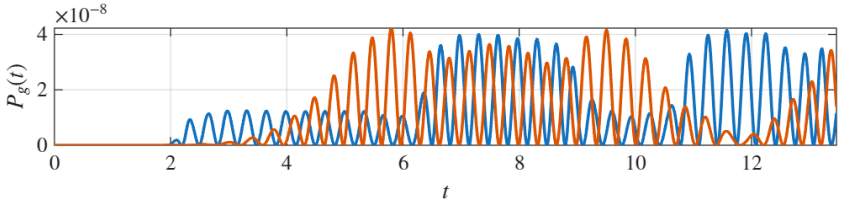
(b) Centerline free-surface displacement $\eta(L_x/2, R, t)$.



(c) Buoy velocity $W(t)$.



(d) Induced current $I(t)$.



(e) Harvested power $P_g(t)$.

Figure 3.3: The evolution of selected functions over time, for both simulations. The wavemaker was active for 10 periods before being switched off, with the simulation continuing for another 10 periods.

The evolution of various functions over time can be seen in Figure 3.3. Considering first the SW case, Figures 3.3c–3.3e show how the waves reach the buoy at around 2 s; these are then reflected back towards the wavemaker, causing a spike in the free-surface displacement shortly after 4 s (in Figure 3.3b). They are then reflected back towards the buoy again, causing a spike in the buoy's motion (and subsequent power generation) at around 6 s, when combining with the newer waves generated by the wavemaker. This

interference between the wavemaker and pre-existing waves make the wavetrain more irregular, better modelling the real scenarios seen at sea. However, in the BL results, these features are less prominent, due to the model's ability to capture dispersion of the waves. These higher-order dispersive terms explain the large difference between the behaviours of the two simulations. Finally, Figures 3.3c and 3.3d also show how the buoy and generator are directly related, with the velocity W and current I having identical profiles.

Following these initial simulations, a resonance and spatial convergence investigation was carried out, with varying values for A, ω, N_x and N_y given by:

$$A = \frac{n}{4}A_0 \text{ with } n \in [1, 2, 3],^{6}$$
$$\omega = \sqrt{gH_0}\frac{m_y\pi}{L_y} \text{ with } m_y \in [4, 5, 6, 7, 8, 9, 10],^{7}$$
$$(N_x, N_y) \in [(10, 50), (10, 100), (20, 100)].$$

Here, all simulations were stopped after $T = 10\,\mathrm{s}$. For each simulation, the average power output

$$\hat{P}_g = \frac{1}{T}\int_0^T I^2 R_l \,\mathrm{d}t$$

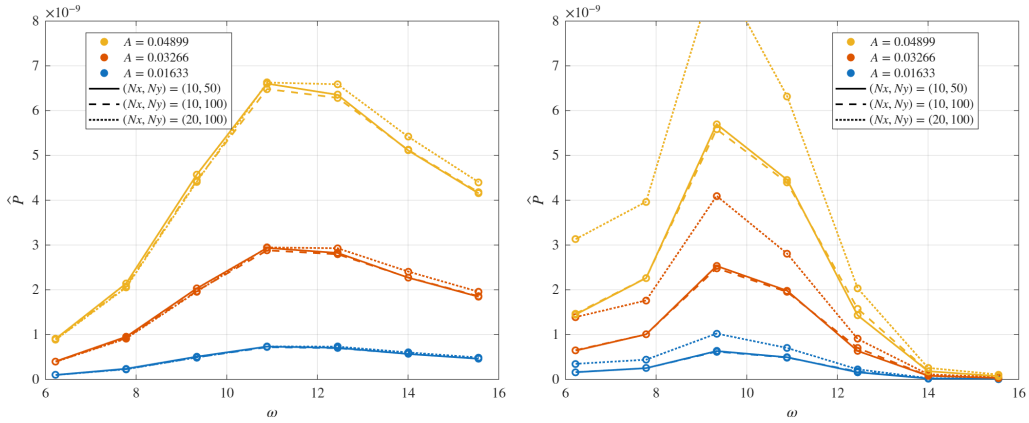
was numerically calculated. Figure 3.4 shows these power outputs as a function of frequency, for various amplitudes solves on various mesh refinements.

The power output of the BL model is generally less than in the SW model, due to the dispersive BL terms causing a reduction in the overall energy in the system (Figure 3.1). It is interesting to note that the resonance in the BL model is lower.

In the SW case, the solutions are generally convergent on the three different meshes; however, these are seen to diverge for the BL model. Note that doubling N_y but keeping N_x fixed (solid to dashed) doesn't change the results much, but subsequently doubling N_x (dashed to dotted) causes a large increase in the power output. As shown in the diagram in Chapter 5 (Figure 5.1), the mesh construction in the contraction depends

[6] $A \approx [0.0163, 0.0327, 0.0490]$.

[7] $\omega \approx [6.22, 7.78, 9.33, 10.89, 12.45, 14.00, 15.56]$: frequencies producing waves with particular numbers of half-wavelengths.



(a) SW: resonance is between 10.89 Hz and 12.45 Hz.

(b) BL: resonance is around 9.33 Hz.

Figure 3.4: Average power-output $\hat{P}$ as a function of wavemaker frequency ω, for three wavemaker amplitudes A and three spatial resolutions. Resonance for the two models occurs at different frequencies, and the BL solutions are seen to diverge as the mesh is refined.

entirely on N_x; the fact that changing N_y appears not to impact the results, but changing N_x has such a large impact, suggests this divergence is caused by the behaviour of the buoy-water coupling.

It appears that the numerical consistency of the constraints must be satisfied to machine precision, rather than merely to $\mathcal{O}\left(10^{-6}\right)$. Because the selected discretisation of the BL model does not meet this criterion, it is not numerically consistent with the key constraint requiring the free surface and buoy to remain coupled throughout the simulation. It is therefore concluded that, under the present approach in which this constraint is reformulated as a governing equation for λ, the BL model is incompatible and unsuitable for use in modelling the device. While it may be possible to design an alternative discretisation that enforces this constraint strictly, such an approach has not been investigated here.

For the remainder of this thesis, the SW approximation shall be employed. However, there are other approaches that could be explored. One could, for example, implement the *Green–Naghdi* approximation [79], which can be viewed as an extension of BL that captures fully nonlinear dispersive effects. The derivation proceeds in precisely the same manner, differing only in that all powers of ϵ in (3.11) are retained. Note, however, that upon linearisation, the Green–Naghdi system reduces to BL, and is therefore only

advantageous in the fully nonlinear regime.

Alternatively, a "full" 3D solution with a small number of vertical grid points could be crudely computed. As a flexible middle ground, a *vertical basis model* could be formed, in which the 3D potential is approximated as a sum of basis functions in the vertical dimension, with horizontally- and temporally-varying coefficients; depth integration then produces a 2D system governing these coefficients. This is similar to the FEM, but with the integrals evaluated analytically rather than numerically. A vertical basis model using a single quadratic basis function recovers the same vertical approximation as BL.

Chapter 4

Firedrake: An Improved Numerical Methodology

As discussed in the previous chapter, unfortunately the BL approximation is incompatible with the key constraint upon the position of the buoy and the water's surface underneath it, as the numerical consistency of the method is not conserved. The BL approximation is therefore not suitable for this problem; as a result, the SW approximation shall be used instead, by neglecting the underlined terms in the "quadricised" 2D Lagrangian (3.16):

$$\mathcal{L} = \rho_0 \int_0^{L_x} \left[\int_0^{l_y} \left\{ \eta \frac{\partial \Phi}{\partial t} + \frac{H}{2} \|\nabla \Phi\|^2 + \frac{1}{2} g \eta^2 + \lambda(\eta - Z)\Theta(y - L_b) \right\} \mathrm{d}y + H_0 \dot{R} \Phi|_{y=0} \right] \mathrm{d}x - \frac{1}{2} M \dot{Z}^2 - \frac{1}{2} L_i \dot{Q}^2 + \frac{Q^2}{2C} - \gamma G(Z_0) \dot{Z} Q. \tag{4.1}$$

This has Hamiltonian

$$\mathcal{H} = \rho_0 \int_0^{L_x} \left\{ \int_0^{l_y} \left\{ \frac{H}{2} \|\nabla \Phi\|^2 + \frac{1}{2} g \eta^2 \right\} \mathrm{d}y + H_0 \dot{R} \Phi|_{y=0} \right\} \mathrm{d}x + \frac{1}{2} M \dot{Z}^2 + \frac{1}{2} L_i \dot{Q}^2 + \frac{Q^2}{2C}, \tag{4.2}$$

and energy balance

$$\frac{\partial \mathcal{H}}{\partial t} = \rho_0 H_0 \ddot{R} \int_0^{L_x} \Phi|_{y=0} \, \mathrm{d}x.$$

Variations of (4.1) yield the final system of equations to be solved:

Hydrodynamics:

$$\frac{\partial \Phi}{\partial t} + g\eta + \lambda\Theta(y - L_b) = 0, \tag{4.3a}$$

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H\nabla\Phi) = 0, \tag{4.3b}$$

Buoy dynamics:

$$M\dot{W} = -\gamma G(Z_0)I + \rho_0 \int_0^{L_x} \int_0^{l_y} \lambda\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x, \tag{4.3c}$$

$$\dot{Z} = W, \tag{4.3d}$$

Electromagnetic generator:

$$\dot{Q} = I, \tag{4.3e}$$

$$L_i\dot{I} = \gamma G(Z_0)W - \frac{Q}{C} - I(R_i + R_c + R_l), \tag{4.3f}$$

Equation for λ:

$$\nabla \cdot (H\nabla\lambda) - \frac{\rho_0}{M} \int_0^{L_x} \int_0^{l_y} \lambda\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x = -\nabla \cdot (gH\nabla\eta) - \frac{\gamma G(Z_0)I}{M} \text{ for } y \geq L_b, \tag{4.3g}$$

Boundary conditions:

$$\lambda = g(\eta - Z) \text{ at } y = L_b, \tag{4.3h}$$

$$\frac{\partial \Phi}{\partial y} = \dot{R} \text{ at } y = 0, \tag{4.3i}$$

$$\nabla\Phi \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_w. \tag{4.3j}$$

The SW system (4.3) consists of two PDEs with spatial and temporal dependencies, one integro-elliptic equation, and four time-dependent ODEs. It is a complex system that is unlikely to admit analytical solutions; instead, numerical methods will be employed to compute approximate solutions. Both the spatial and temporal dimensions require separate numerical treatment; for the spatial domain, the finite element method (FEM) will be utilised, while symplectic temporal integrators will be used to preserve the physical consistency of the numerical solution over time.

This system was solved by Bokhove et al. [44, 45] using an in-house FEM code written in MATLAB. This code is inefficient and inflexible, and is not scalable to higher-order

numerical accuracy or to solving the nonlinear version of the problem. There does exist, however, a specialised external, Python-based FEM toolkit named *Firedrake*, which abstracts all of the heavy-lifting of the method and enables flexibility with regard to higher-order methods and nonlinearities; it is also far more efficient at carrying out the matrix assembly and solves required.

This chapter details the numerical methodologies required to solve (4.3), with Section 4.1 detailing the steps taken in an FEM algorithm, and Section 4.2 outlining the precise discretisation of two symplectic integrators. Firedrake is introduced in Section 4.3 in the form of a tutorial, with a specific subdomain feature explored (Section 4.4). The following chapter (Chapter 5) describes the development of a bespoke algorithm that solves the system (4.3) using Firedrake, highlighting improvements over the pre-existing MATLAB code and exploring its use for some preliminary optimisation investigations.

4.1 The finite element method

The FEM is a numerical method that partitions the domain into a finite number of smaller subregions, or *elements*, and constructs an approximate solution over each element. Unlike, for example, the finite difference method, which is most naturally applied on structured, typically rectangular or cuboidal meshes with uniform spacing, FEM was expressly developed to operate on meshes composed of elements of varying shape and size. As a result, it is particularly well suited to problems defined on complex geometries, such as those arising in fluid flow, mass transport, structural analysis and heat transfer [80–82].

4.1.1 The weak-form representation

The first step in modelling problems using FEM is to transform PDE into their *weak-form* equivalents, obtained by multiplying the equation (or *strong form*) by an arbitrary, but suitably-defined,[1] *test function*, and then integrating. In the weak form, the equation is no longer required to hold absolutely, but instead holds only with respect to the test

[1] That is, belonging to an appropriate function space such that the resulting integrals are well-defined. For example, as illustrated in (4.5), for second-order problems such as the Laplace equation, the test function needs to have first-derivatives that are square-integrable.

function.

An example of a strong-weak form relationship, which we have already encountered when discussing the calculus of variations in Section 2.1, is the equivalence between (2.6) and (2.7). In the strong form (2.7), the equation $\mathcal{F}_i = 0$ holds pointwise, whereas in the weak form (2.6) it holds only with respect to the variation δv_i. That is, δv_i serves as the test function in this context; variations in Lagrangian mechanics and test functions in weak forms both measure the extent to which the corresponding *trial function* is free to vary. Indeed, FEM and the calculus of variations are closely related: after taking a variation, the resulting expression is often identical to the weak form of the equation defined by that variation.[2]

Using the Laplace equation (2.39a) as a worked example, the strong form

$$\nabla^2 \phi = 0 \tag{4.4}$$

is equivalent to the weak form

$$\iiint_\Omega v \nabla^2 \phi \, \mathrm{d}V = 0,$$

which holds for all test functions $v(\mathbf{x})$. For reasons that will become clear in Section 4.1.5, it is desirable to transfer one derivative from ϕ to v using Green's first identity, e.g. (A.50) (reversed, and generalised to any Ω), giving

$$\iiint_\Omega \nabla v \cdot \nabla \phi \, \mathrm{d}V = \iint_{\partial\Omega} v \nabla \phi \cdot \hat{\mathbf{n}} \, \mathrm{d}S. \tag{4.5}$$

Note that this reduction in derivative order introduces a boundary integral. In FEM, the treatment of this integral depends on the type of BCs present in the problem:

- For **Dirichlet BCs**, where the value of the solution (trial function) is specified on the boundary ($\phi = f_D$ on $\partial\Omega_D$), the corresponding test function (representing an admissible variation of the fixed solution) is required to vanish. As a result, the

[2] The methods are not always exactly equivalent, particularly when material coefficients (parameters describing the material or medium, such as stiffness, density, or conductivity) vary in space.

boundary integral disappears:

$$\iiint_\Omega \nabla v \cdot \nabla \phi \, \mathrm{d}V = 0.$$

- For **Neumann BCs**, where the derivative of the solution normal to the boundary is prescribed ($\nabla\phi \cdot \hat{\mathbf{n}} = f_N$ on $\partial\Omega_N$), the solution value on the boundary remains unknown, and the test function is not constrained. However, since the known quantity $\nabla\phi \cdot \hat{\mathbf{n}}$ appears in the boundary integral, it acts as a forcing term:

$$\iiint_\Omega \nabla v \cdot \nabla \phi \, \mathrm{d}V = \iint_{\partial\Omega_N} v f_N \, \mathrm{d}S.$$

- For **Robin BCs**, where a linear combination of solution and derivative is imposed ($\alpha\phi + \beta\nabla\phi \cdot \hat{\mathbf{n}} = f_R$ on $\partial\Omega_R$), the solution is not fully known and again the test function is generally nonzero. Rearranging the BC reveals that that the boundary integral contains both known and unknown terms; these are separated into a known forcing term, and an additional contribution to the problem's unknowns:

$$\iiint_\Omega \nabla v \cdot \nabla \phi \, \mathrm{d}V + \frac{\alpha}{\beta} \iint_{\partial\Omega_R} v\phi \, \mathrm{d}S = \frac{1}{\beta} \iint_{\partial\Omega_R} v f_R \, \mathrm{d}S.$$

4.1.2 Domain partitioning

After obtaining our weak forms, we partition the domain into elements using a mesh. In 1D, this corresponds to dividing the interval into smaller subintervals, much as in finite difference methods; however, in FEM these subintervals need not be uniformly spaced. While in 1D this flexibility is of limited practical significance, in higher dimensions it becomes critical. For example, in 2D, rather than being restricted to structured rectangular grids, elements of arbitrary convex polygons—most commonly triangles and convex, non-rectangular quadrilaterals—can be readily used. Consequently, the domain need not be a simple hypercube, but may possess an essentially arbitrary geometry. Although it is possible to achieve comparable flexibility within finite difference frameworks, for example through coordinate transformations or embedded-boundary techniques, such generality arises naturally within the FEM formulation.

Now, recalling that the generalised weak form (4.5) is an integral over the entire domain Ω, it can be equivalently represented as a discrete sum of integrals over the N_{el} elements Ω_e:

$$\sum_{e=1}^{N_{el}} \iiint_{\Omega_e} \nabla v \cdot \nabla \phi \, \mathrm{d}V = \sum_{e=1}^{N_{el}} \iint_{\partial\Omega_e \cap \partial\Omega} v \nabla \phi \cdot \hat{\mathbf{n}} \, \mathrm{d}S. \tag{4.6}$$

Note the domain of integration for the boundary integral, which is the intersection between the element's boundary and the entire domain's boundary. For an element that does not lie on the edge of the domain, this intersection is empty and that element can be discounted from the summation.

There is nothing particularly clever about the weak form itself; it is still an exact reformulation of the governing equations, and (4.6) is in principle no easier to solve than the strong form (4.4). Its primary contribution is simply enabling the element-wise integral summation. The true power of numerical methods based on weak forms emerges in the next step, where a suitable approximation of the integrands is formed.

4.1.3 Basis functions

Lagrange polynomials

Besides FEM, there are other numerical methods which utilise weak forms. In most cases, the methods differ in how they approximate the integrands, which is usually by representing unknowns as linear combinations of *basis functions*.[3] Extending our stationary worked example to incorporate time-dependence, the representation of the spatio-temporal variable $\phi(\mathbf{x}, t)$ is

$$\phi(\mathbf{x}, t) \approx \sum_i \hat{\phi}_i(t) \psi_i(\mathbf{x}), \tag{4.7}$$

which consists of (known) spatially-dependent basis functions $\psi_i(\mathbf{x})$ and (unknown) time-dependent coefficients $\hat{\phi}_i(t)$.

The key feature of FEM is the particular choice of basis functions; there are a number of

[3] Simple and well-known examples of representing functions as linear combinations of basis functions are: 1) polynomials of degree n, which are linear combinations of all monomials of degree n and lower; 2) the Taylor series, which also uses monomials as a basis; and 3) the Fourier series, whose basis is formed of trigonometric functions.

bases to choose from (see e.g. [83, 4–6]), but by far the most common are the *Lagrange polynomials*. This is mainly due to their simplicity and ease of construction via an explicit formula, leading to a Kronecker delta property which simplifies the application of Dirichlet BCs and the interpolation of the resulting solution (see equation (4.8) and the associated discussion).

Consider first a 1D problem, in which the domain is a simple straight line defined on the interval $x \in [a, b]$, and the e^{th} element is the subinterval $x \in [x^{(e-1)}, x^{(e)}]$, with $x^{(0)} = a$ and $x^{(N_{el})} = b$. Then the Lagrange polynomials on element e are defined as

$$\ell_i^{(e)}(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x - x_k^{(e)}}{x_i^{(e)} - x_k^{(e)}} \quad i \in \{0, \dots, n\}. \tag{4.8}$$

This formulation defines $(n+1)$ polynomials of degree n spanning the element, based on its $(n+1)$ nodes $x_i^{(e)}$.[4]

The Lagrange polynomials have the property $\ell_i^{(e)}\left(x_k^{(e)}\right) = \delta_{ik}$, with δ representing the Kronecker delta; that is, $\ell_i^{(e)}$ equals zero at all nodes other than $x_i^{(e)}$, at which it equals one. Within the element and between the nodes, $\ell_i^{(e)}$ interpolates using an n^{th}-degree polynomial, while outside of the element $\ell_i^{(e)}$ is identically zero.

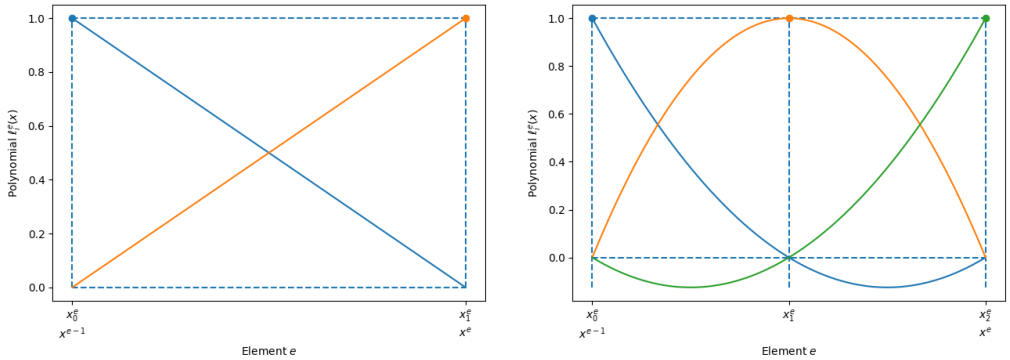
Node spacing

Figure 4.1 contains examples of these polynomials for different degrees, using equispaced nodes where

$$x_i^{(e)} = x^{(e-1)} + \left(x^{(e)} - x^{(e-1)}\right) \frac{i}{n}.$$

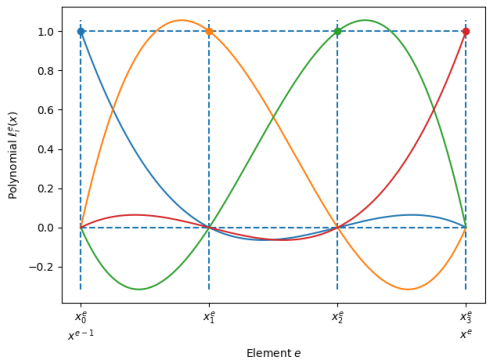
Note in particular Figure 4.1d, which features extreme behaviour towards the endpoints of the element. This is known as Runge's phenomenon [84], and is a common problem in polynomial interpolation at high degrees. This is easily overcome, however, by instead using a set of nodes that is more densely distributed towards the endpoints; a common

[4] Note the contrasting indexing: $x^{(e)}$ is an indexing of the $(N_{el}+1)$ *global* nodes defining the partitioning of the entire domain into elements, while $x_i^{(e)}$ is an indexing of the $(n+1)$ *local* nodes belonging to the e^{th} element.

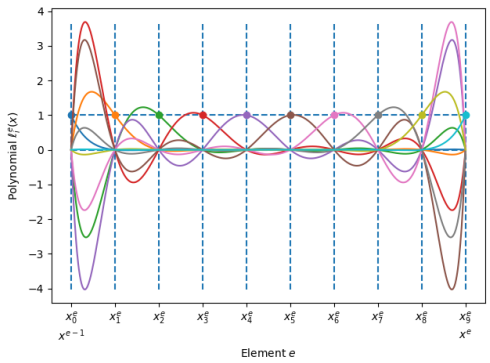


(a) Linear ($n = 1$).

(b) Quadratic ($n = 2$).



(c) Cubic ($n = 3$).



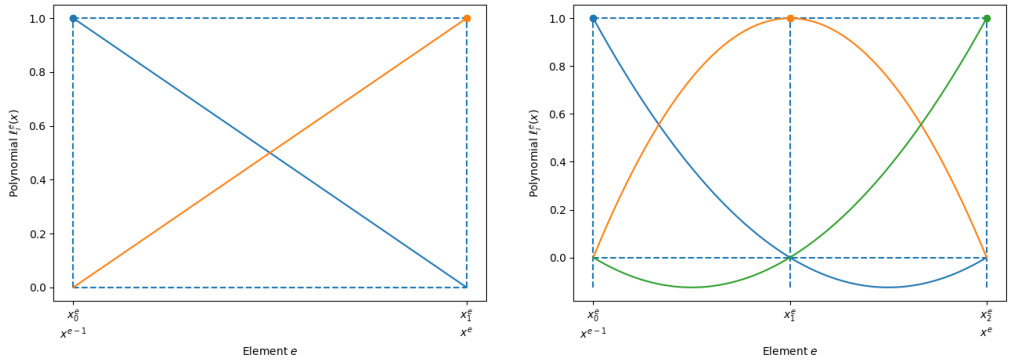
(d) Nonic ($n = 9$).

Figure 4.1: The Lagrange polynomials of degree 1, 2, 3 and 9, using equispaced nodes.

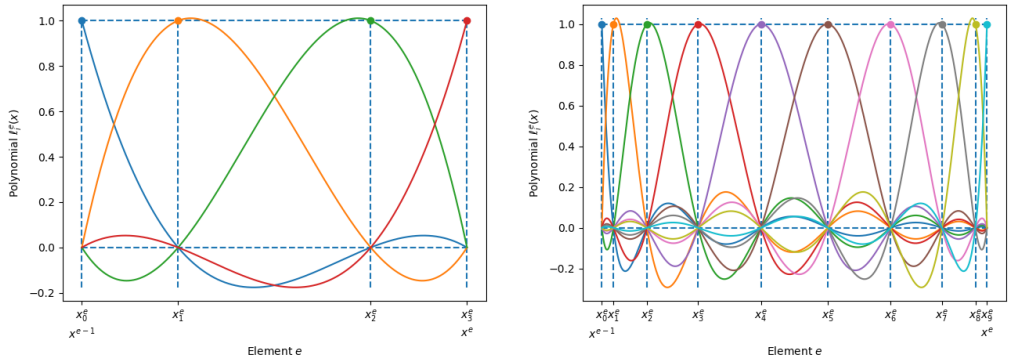
choice are the Chebyshev nodes given by

$$x_i^{(e)} = \frac{x^{(e-1)} + x^{(e)}}{2} + \frac{x^{(e-1)} - x^{(e)}}{2} \cos\left(\frac{i\pi}{n}\right).$$

Figure 4.2 contains the same-degree polynomials as in Figure 4.1, but with Chebyshev nodes. For first- and second-degree polynomials, the results are identical, while there is little change for $n = 3$; however, one can clearly see the improvement when $n = 9$. Other sets of unevenly-distributed nodes include: the Gauss–Legendre nodes, which are the roots of the $(n + 1)^{\text{th}}$-degree Legendre polynomial; the Gauss–Legendre–Lobatto nodes, which are the roots of the derivative of the n^{th} Legendre polynomial plus the element's endpoints; and those generated by the recursive algorithm presented in [85], which are a leading set of points in the literature [86, p. 5].



(a) Linear. **(b)** Quadratic.



(c) Cubic. **(d)** Nonic.

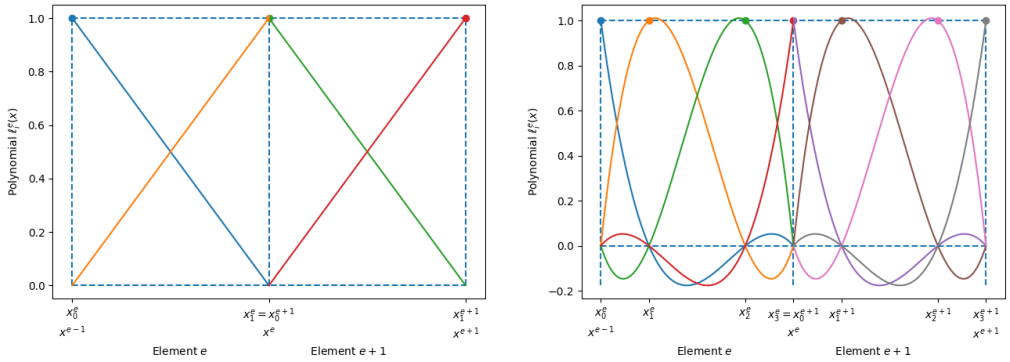
Figure 4.2: The Lagrange polynomials of degree 1, 2, 3 and 9, using Chebyshev nodes.

Continuous and discontinuous interpolations

Now, consider the domain $x \in [a, b]$ as a whole. Recall that, for each element, the first and last Lagrange polynomials are equal to one at the element's first and last nodes. Therefore, for neighbouring elements sharing a node, basis-function contributions from both elements exist at the shared node. See, for example, two neighbouring nodes using linear and cubic basis functions in Figure 4.3.

Then, consider the contribution of the coefficients in the approximation (4.7). These scale their corresponding basis function such that, as a result of the Kronecker-delta property, the resulting linear combination produces an approximation taking the value of each coefficient at its corresponding node, while for any other value of x the approximation interpolates between the nodes defining the element containing that particular value of x. This linear combination is known as a Lagrange *interpolating* polynomial.

Here, an important question arises: do the nodes common to multiple elements have just



(a) Linear. **(b)** Cubic.

Figure 4.3: The Lagrange polynomials of degree 1 and 3 on neighbouring elements.

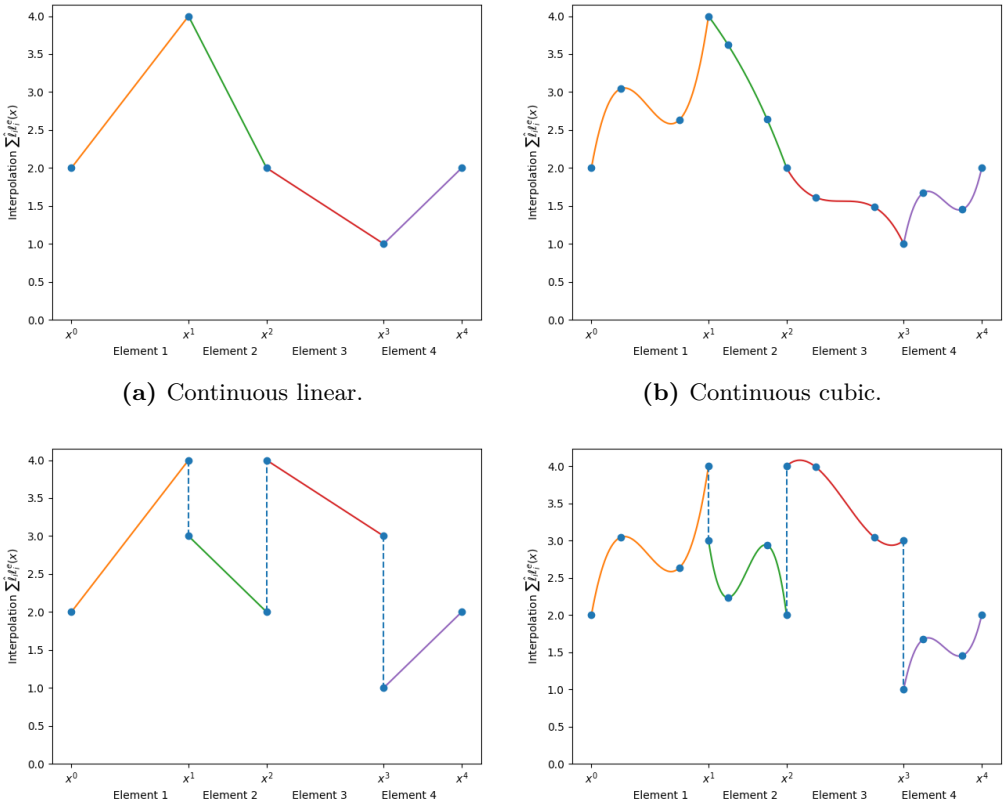
a single coefficient, or does each element contribute its own coefficient? In fact, both scenarios are possible: the second option gives rise to the notion of *discontinuous* FEM, while the first option produces the simpler *continuous* method (see Figure 4.4). The continuous Lagrange polynomials automatically satisfy the global first-derivative-square-integrability requirement of the Laplace operator, whereas the discontinuous approach only does so element-wise; additional element-interface terms (jumps, averages, and numerical fluxes) are required to account for these discontinuities. In this thesis, only the continuous FEM is utilised.

Multidimensional Lagrange polynomials

To enable discussion of the Lagrange polynomials in their most basic form, we have so-far only considered a 1D example. However, they are just as suitable in higher dimensions: consider, for example, a 2D problem defined on a quadrilateral mesh. First, note that any convex quadrilateral can be transformed into a rectangle defined on $(x, y) \in [x^{(e_x-1)}, x^{(e_x)}] \times [y^{(e_y-1)}, y^{(e_y)}]$ [87]. Since the domain is the Cartesian product of the two 1D domains $x \in [x^{(e_x-1)}, x^{(e_x)}]$ and $y \in [y^{(e_y-1)}, y^{(e_y)}]$, the $(n+1)^2$ 2D n^{th}-degree Lagrange polynomials are found by simply taking the Cartesian product of the $(n+1)$ 1D functions in x and the $(n+1)$ 1D functions in y:

$$\ell_{ij}^{(e_x,e_y)}(x, y) = \ell_i^{(e_x)}(x)\ell_j^{(e_y)}(y) \quad i, j \in \{0, \ldots, n\}.$$

Figure 4.5 gives an example, for $n = 4$, $i = 2$ and $j = 1$.



(a) Continuous linear.

(b) Continuous cubic.

(c) Discontinuous linear.

(d) Discontinuous cubic.

Figure 4.4: Lagrange interpolating polynomials of degree 1 and 3, using both continuous and discontinuous coefficients.

For 2D triangular meshes, the basis functions can be constructed in a similar manner, while for higher dimensions using N-dimensional hypercubes and simplexes, the N-dimensional basis functions can also be defined.

4.1.4 Matrix-vector assembly

Using the same basis-functions in (4.7) to approximate both the test and trial functions[5] (with the caveat that the time-independent test function v has constant coefficients $\hat{v}_i$), the LHS of the element-wise weak form (4.6) is approximated by

$$\sum_{e=1}^{N_{el}} \iiint_{\Omega_e} \nabla v \cdot \nabla \phi \, \mathrm{d}V \approx \sum_{e=1}^{N_{el}} \sum_{i=0}^{n} \sum_{j=0}^{n} \hat{v}_i^{(e)} \left(\iiint_{\Omega_e} \nabla \psi_i^{(e)} \cdot \nabla \psi_j^{(e)} \, \mathrm{d}V \right) \hat{\phi}_j^{(e)}. \tag{4.9}$$

[5]Methods in which the bases for the test and trial functions differ do exist, and are known as Petrov–Galerkin methods (e.g. [88, p. 60]); however, these will not be considered here.

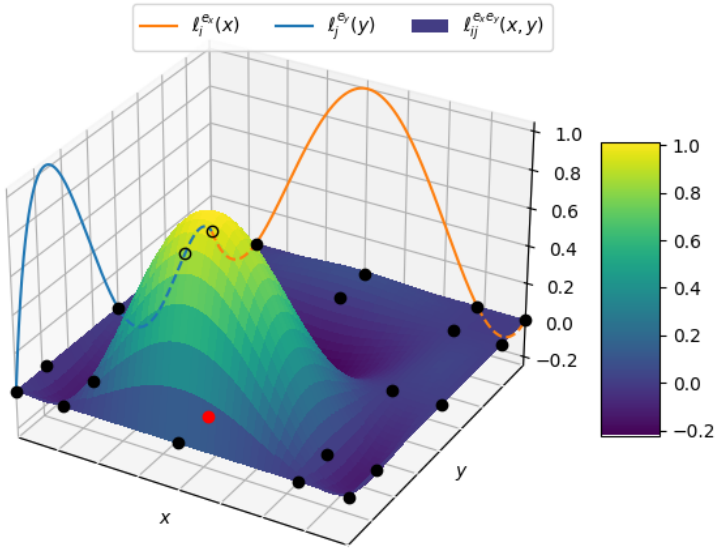


Figure 4.5: A Lagrange polynomial defined over a 2D rectangular element, derived by multiplying two independent 1D polynomials. The polynomial is zero at all black nodes, and one at the red node.

Since we are considering continuous FEM only, the triple sum produces multiple terms containing the same coefficients $\hat{v}$ and/or $\hat{\phi}$ whenever i and/or j take the values 0 and n (i.e. when they correspond to the element-boundary nodes). By redefining the indexing, one can reduce the triple summation to a double summation.

By using the property of individual basis functions being identically zero outside of their elements, they can be considered to be defined globally rather than only within their elements. Additionally, for all groups of functions sharing an element-boundary coefficient, a single function can be defined as the sum of the entire group. Then each global node has a single corresponding coefficient and a single, globally-defined basis function, and the local indexing $(\cdot)_i^{(e)}$ can be redefined as a global indexing $(\cdot)_i$ for the N_n total nodes. Then (4.9) can be equivalently written as

$$\sum_{e=1}^{N_{el}} \iiint_{\Omega_e} \nabla v \cdot \nabla \phi \, \mathrm{d}V \approx \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \hat{v}_i \left(\iiint_{\Omega} \nabla \psi_i \cdot \nabla \psi_j \, \mathrm{d}V \right) \hat{\phi}_j, \tag{4.10}$$

which in turn can be written in the matrix-vector form

$$\sum_{e=1}^{N_{el}} \iiint_{\Omega_e} \nabla v \cdot \nabla \phi \, \mathrm{d}V \approx \mathbf{v} \cdot \mathbf{K}\boldsymbol{\phi},$$

where $\mathbf{v}$ and $\boldsymbol{\phi}$ are vectors of the (unknown) coefficients $\hat{v}_i$ and $\hat{\phi}_j$, and $\mathbf{K}$ is a matrix with entries given by

$$K_{ij} = \iiint_{\Omega} \nabla\psi_i \cdot \nabla\psi_j \,\mathrm{d}V.$$

Boundary conditions

For the RHS of (4.6), recall that the form of the boundary integral is dependent upon the BCs:

- **Neumann**: After using the approximation (4.7), the forcing term becomes

$$\iint_{\partial\Omega_N} v f_N \,\mathrm{d}S \approx \sum_{i=1}^{N_n} \hat{v}_i \left(\iint_{\partial\Omega_N} \psi_i f_N \,\mathrm{d}S \right) = \mathbf{v} \cdot \mathbf{f}, \quad \text{with} \quad f_i = \iint_{\partial\Omega_N} \psi_i f_N \,\mathrm{d}S.$$

  Here we see how the boundary integral reduces to a known forcing term $\mathbf{f}$. Note that for any node not on the boundary, its (global) basis function ψ_i is zero on $\partial\Omega_N$, and $f_i = 0$.

- **Dirichlet**: Recall that the RHS reduces to 0. However, since the coefficients in (4.7) match the function values at their corresponding point in space, and since trial-function values along the boundary are known, some parts of the LHS are known, and can be moved to the RHS as forcing terms. This is easiest to see by writing the matrix-vector equation $\mathbf{v} \cdot \mathbf{K}\boldsymbol{\phi} = 0$ in block form:

$$\begin{aligned} \mathbf{v} \cdot \mathbf{K}\boldsymbol{\phi} &= \begin{pmatrix} \mathbf{v}_{internal} \\ \mathbf{v}_{boundary} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{K}_{i,i} & \mathbf{K}_{i,b} \\ \mathbf{K}_{b,i} & \mathbf{K}_{b,b} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi}_{internal} \\ \boldsymbol{\phi}_{boundary} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{v}_{internal} \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{K}_{i,i} & \mathbf{K}_{i,b} \\ \mathbf{K}_{b,i} & \mathbf{K}_{b,b} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi}_{internal} \\ \boldsymbol{f_D} \end{pmatrix} = 0, \end{aligned}$$

  where we recall that the test- and trial-function boundary coefficients are zero and $\boldsymbol{f_D}$, respectively. Performing the matrix multiplications reduces this to

$$\mathbf{v}_{internal} \cdot (\mathbf{K}_{i,i}\boldsymbol{\phi}_{internal} + \mathbf{K}_{i,b}\boldsymbol{f_D}) = 0,$$

where it is now clear to see that the term involving $\boldsymbol{f_D}$ can be moved to the RHS:

$$\mathbf{v} \cdot \mathbf{K}\boldsymbol{\phi} = \mathbf{v} \cdot \mathbf{f} \text{ with } \mathbf{f} = -\mathbf{K}_{i,b}\boldsymbol{f_D},$$

after dropping the subscripts. Therefore, despite causing the boundary integral to vanish, Dirichlet BCs still result in forcing terms appearing on the RHS of the matrix-vector equation.

- **Robin**: In a similar way to the Neumann case, using (4.7) in the boundary integrals yields

$$\mathbf{v} \cdot \mathbf{K}\boldsymbol{\phi} + \frac{\alpha}{\beta} \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \hat{v}_i \left(\iint_{\partial\Omega_R} \psi_i \psi_j \, \mathrm{d}S \right) \hat{\phi}_j = \frac{1}{\beta} \sum_{i=1}^{N_n} \hat{v}_i \left(\iint_{\partial\Omega_R} \psi_i f_R \, \mathrm{d}S \right)$$
$$\Rightarrow \mathbf{v} \cdot (\mathbf{K} + \mathbf{M})\boldsymbol{\phi} = \mathbf{v} \cdot \mathbf{f},$$

where the matrix $\mathbf{M}$ and vector $\mathbf{f}$ have entries

$$M_{ij} = \frac{\alpha}{\beta} \iint_{\partial\Omega_R} \psi_i \psi_j \, \mathrm{d}S \text{ and } f_i = \frac{1}{\beta} \iint_{\partial\Omega_R} \psi_i f_R \, \mathrm{d}S.$$

Again, we obtain a matrix-vector equation with a RHS forcing term. Interestingly, these types of BCs also produce a second matrix on the LHS; during assembly, these two matrices are summed together to produce a single matrix to be solved.

4.1.5 FEM as a system of simultaneous equations

Regardless of the types of BC present, FEM always produces a matrix-vector equation of the form

$$\mathbf{v} \cdot \mathbf{A}\boldsymbol{\phi} = \mathbf{v} \cdot \mathbf{f}. \tag{4.11}$$

Now, recall that the weak form (4.5) holds for *any* arbitrary test function v. Therefore, the matrix-vector equation (4.11) holds for any vector $\mathbf{v}$. The only way this is possible is if

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{f}, \tag{4.12}$$

which is a system of equations for the unknowns $\boldsymbol{\phi}$, and can be solved by $\boldsymbol{\phi} = \mathbf{A}^{-1}\mathbf{f}$.[6]

The matrices $\mathbf{M}$ and $\mathbf{K}$ are known as the *mass* and *stiffness* matrices, respectively; names that hark back to FEM's inception as a method for solving structural mechanics problems, where the mathematical terms describing a structure's mass and stiffness produce equivalent matrices in the finite-element (FE) formulation. Note that both $\mathbf{M}$ and $\mathbf{K}$ are symmetric; for $\mathbf{K}$ this is a property that arises from the removal of trial-function second derivatives in (4.5). Additionally, recall that ψ_i is identically zero over the majority of the domain; it is only nonzero over the elements that touch node x_i. Therefore, the products $\psi_i\psi_j$ and $\nabla\psi_i \cdot \nabla\psi_j$ are only nonzero for a small number of ij combinations: those belonging to the same or neighbouring elements.

This property is the true power of FEM, and its effect is that the matrices $\mathbf{M}$ and $\mathbf{K}$ are *sparse*, meaning the vast majority of their entries are zero. Additionally, the sparsity (defined as the proportion of matrix entries that are zero) increases asymptotically towards 1 as the number of elements and/or the basis-function degree (and thus number of nodes) increases. As an example, Figure 4.9 shows the sparsity pattern[7] of the stiffness matrix for the Laplace equation over a regular 2D mesh with 8 elements in each direction, using linear Lagrange polynomials (giving 64 elements and 81 nodes in total). The high sparsity of these matrices mean efficient algorithms can be employed to both store them and carry out operations with them, meaning problems with many millions of nodes can be efficiently solved with modern computing power.

Nonlinear equations

In this section we have so far only considered the FE formulation of a linear equation, which produces the system of linear equations (4.12). However, FEM can just as easily be applied to nonlinear equations, such as the kinematic equation (2.39c). This equation's

[6] By design, the matrix $\mathbf{A}$ is sparse, and in all but the most trivial of problems, finding its inverse is not actually practical. Instead, the problem is solved either using decomposition methods (such as LU or Cholesky), or iterative methods (such as Jacobi or Gauss–Seidel). See Chapters 3 and 4 of Quarteroni et al. [89] for a discussion on such techniques.

[7] A sparsity pattern is a pictorial representation of a sparse matrix, where each matrix entry is represented as either a coloured (nonzero) or white (zero) pixel.

weak form is

$$\int_0^{L_x}\int_R^{l_y}\int_0^h v\left\{\frac{\partial h}{\partial t}+\nabla_H\phi\cdot\nabla_H h-\frac{\partial\phi}{\partial z}\right\}\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x=0,$$

which, after substitution of the approximation (4.7) becomes

$$\sum_{i=1}^{N_n}\hat{v}_i\int_0^{L_x}\int_R^{l_y}\int_0^h\psi_i\left\{\sum_{j=1}^{N_n}\dot{\hat{h}}_j\psi_j+\sum_{j=1}^{N_n}\sum_{k=1}^{N_n}\hat{\phi}_j\nabla_H\psi_j\cdot\nabla_H\psi_k\hat{h}_k-\sum_{j=1}^{N_n}\hat{\phi}_j\frac{\partial\psi_j}{\partial z}\right\}\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x=0.$$

Written in matrix-vector form, this is equivalently $\mathbf{v}\cdot\mathbf{f}=0$, or just $\mathbf{f}=0$ when the arbitrariness of $\mathbf{v}$ is considered, with $\mathbf{f}$ representing a system of nonlinear equations where the i^{th} equation is

$$f_i=\int_0^{L_x}\int_R^{l_y}\int_0^h\psi_i\left\{\dot{\mathbf{h}}\cdot\boldsymbol{\psi}+\boldsymbol{\phi}\cdot\mathbf{K}_H\mathbf{h}-\boldsymbol{\phi}\cdot\frac{\partial\boldsymbol{\psi}}{\partial z}\right\}\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x,$$

with $\boldsymbol{\psi}$ a vector of basis functions and $\mathbf{K}_H$ a horizontal stiffness matrix with entries

$$\mathbf{K}_{Hjk}=\nabla_H\psi_j\cdot\nabla_H\psi_k.$$

That is, $\mathbf{f}=0$ represents a system of coupled, nonlinear equations with unknowns $\mathbf{h}$ and $\boldsymbol{\phi}$. Since this system is nonlinear, a simple linear matrix-vector solution method (as in the linear case) is not possible; however, the multidimensional form of Newton's method can be utilised to find an approximate solution. To do this, the Jacobian matrix $\mathbf{J}_\mathbf{f}$ needs to be found, but because $\mathbf{f}$ contains known (albeit very complicated) and differentiable algebraic terms, the calculation of this matrix is possible. As in the linear case, this matrix is also sparse.

There is currently, of course, a major issue with solving this nonlinear system; the sizes of $\mathbf{f}$, $\mathbf{h}$ and $\boldsymbol{\phi}$ are all N_n, meaning $\mathbf{f}=0$ represents a system of N_n equations but with $2N_n$ unknowns. The system either needs to be coupled to another system of N_n equations in the same $2N_n$ unknowns, to be solved simultaneously, or one of $\mathbf{h}$ or $\boldsymbol{\phi}$ needs to be known before it can be used to solve for the other. The numerical methodology for the time-dependence of the system, discussed in the following section, allows the second of

these situations to occur.

Note, also, that when solving coupled systems with FEM, an inappropriate choice of basis functions for the unknowns can lead to numerical instability [90]. However, as mentioned above, the particular temporal discretisation employed decouples the equations, thereby circumventing the inf-sup constraints associated with a fully-coupled FEM formulation. The same basis functions are therefore used for all unknowns; although no formal stability analysis has been carried out, the numerical evidence indicates stable behaviour in all cases considered.

4.2 Symplectic temporal integrators

4.2.1 Symplecticity

Since our SW system (4.3) describes a real-world mechanical and electrical system, it is important that the numerical solution preserves key physical properties. One obvious example is energy conservation: over time, the numerical solution should maintain the system's total energy. Another, less well-known property in dynamical systems derived from Lagrangian mechanics is *symplecticity*.

Recall Section 2.1.3, where we stated that the Lagrangian, typically dependent upon some generalised positions $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$, can be rewritten in Hamiltonian form via a Legendre transform, by introducing the conjugate momenta $\mathbf{p} = \partial\mathcal{L}/\partial\dot{\mathbf{q}}$. It can be shown (e.g. by Hairer et al. [91, 181–182]) that carrying out variations on the Hamiltonian leads to a first-order dynamical system in $\mathbf{p}$ and $\mathbf{q}$ of the form

$$\dot{\mathbf{p}} = -\frac{\partial\mathcal{H}}{\partial\mathbf{q}}(\mathbf{p},\mathbf{q}) \quad \text{and} \quad \dot{\mathbf{q}} = \frac{\partial\mathcal{H}}{\partial\mathbf{p}}(\mathbf{p},\mathbf{q}). \tag{4.13}$$

The flow map χ_t that advances the system from its initial state to its state at time t—i.e. $\chi_t(\mathbf{p}(0),\mathbf{q}(0)) = (\mathbf{p}(t),\mathbf{q}(t))$—is symplectic if it preserves the phase-space volume $\mathrm{d}\mathbf{p} \wedge \mathrm{d}\mathbf{q}$. In the single-component case, this reduces to the area of the parallelogram spanned by $\mathrm{d}\mathbf{p}$ and $\mathrm{d}\mathbf{q}$; in higher dimensions, it generalises to a hyper-volume in phase space. Symplecticity therefore ensures the qualitative geometric structure of the system

is maintained over time (e.g. Figure 2.1 of [91, p. 183]).

It is well known that any system derived from a Lagrangian is inherently symplectic [92, 93]. Consequently, *symplectic integrators*, which preserve this property numerically, are preferred for such systems. Two widely used methods are the first-order symplectic Euler (SE) and the second-order Störmer–Verlet (SV).[8]

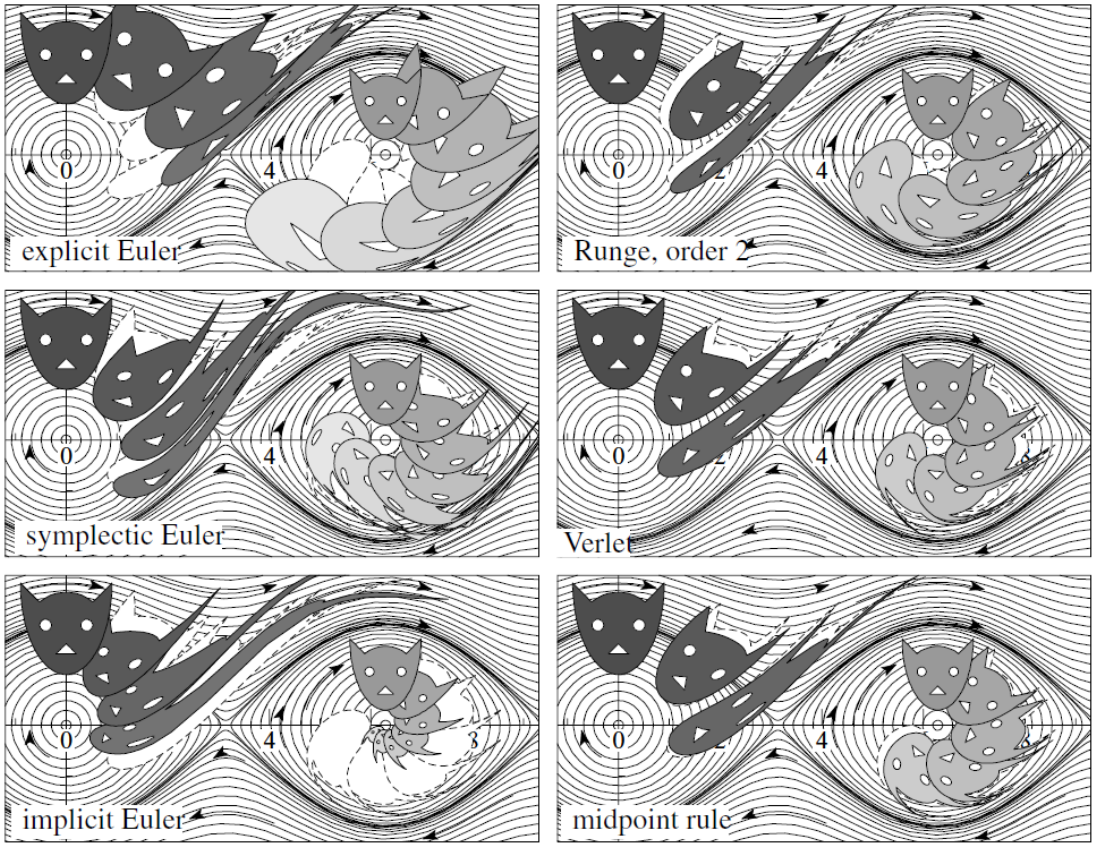


Figure 4.6: A demonstration of the *symplecticity*—preservation of phase-space volume (the area of the cat faces) over time—or lack thereof, of various numerical methods. Taken from [91, p. 188]. The methods on the left are first-order, and on the right are second-order. The explicit Euler, Runge–Kutta and implicit Euler methods are non-symplectic (the first two exaggerate the area, while the latter causes it to diminish). The symplectic Euler, Störmer–Verlet and implicit midpoint methods are symplectic, each retaining the area, despite numerical error skewing the shape.

Hairer et al. [91] demonstrated and proved the symplecticity of these methods, comparing them with non-symplectic alternatives such as explicit and implicit Euler, and second-order Runge–Kutta. For the typical pendulum problem of a mass swinging in a plane

[8] Higher-order symplectic integrators exist (see, for example, [94–97]), but they are more complex and less commonly employed.

under the influence of gravity, Figure 4.6 shows how the area of the solution evolves under the different time-stepping schemes. The SE and SV methods preserve the area (depicted as the "cat faces"), with some numerical error skewing the shape, whereas the explicit Euler and second-order Runge–Kutta methods cause the area to grow, and implicit Euler causes it to shrink towards zero. This illustrates the long-term qualitative advantage of symplectic integrators. The implicit midpoint method, included in Figure 4.6, is also symplectic, but for the reasons discussed below it is not utilised presently.

4.2.2 The symplectic Euler and Störmer–Verlet methods

Using the subscripts n and $(n+1)$ to denote values at the n^{th} and $(n+1)^{\text{th}}$ time-step Δt, the SE and SV methods are as follows (e.g. [91, p. 189]):

SE:

$$\frac{\mathbf{p}_{n+1} - \mathbf{p}_n}{\Delta t} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}(\mathbf{p}_{n+1}, \mathbf{q}_n) \tag{4.14a}$$

$$\frac{\mathbf{q}_{n+1} - \mathbf{q}_n}{\Delta t} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}(\mathbf{p}_{n+1}, \mathbf{q}_n); \tag{4.14b}$$

SV:

$$\frac{\mathbf{p}_{n+1/2} - \mathbf{p}_n}{\Delta t/2} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}(\mathbf{p}_{n+1/2}, \mathbf{q}_n) \tag{4.15a}$$

$$\frac{\mathbf{q}_{n+1} - \mathbf{q}_n}{\Delta t} = \frac{1}{2}\left(\frac{\partial \mathcal{H}}{\partial \mathbf{p}}(\mathbf{p}_{n+1/2}, \mathbf{q}_n) + \frac{\partial \mathcal{H}}{\partial \mathbf{p}}(\mathbf{p}_{n+1/2}, \mathbf{q}_{n+1})\right) \tag{4.15b}$$

$$\frac{\mathbf{p}_{n+1} - \mathbf{p}_{n+1/2}}{\Delta t/2} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}(\mathbf{p}_{n+1/2}, \mathbf{q}_{n+1}). \tag{4.15c}$$

In achieving symplecticity, the SE method combines features of both the implicit and explicit Euler methods: first, $\mathbf{p}_{n+1}$ is calculated implicitly using $\mathbf{p}_{n+1}$ and $\mathbf{q}_n$ on the RHS, and then $\mathbf{q}_{n+1}$ is computed explicitly using $\mathbf{q}_n$ and the already-obtained $\mathbf{p}_{n+1}$. Similarly, SV combines both implicit and explicit features: an implicit half-step is used to find the interim $\mathbf{p}_{n+1/2}$, which is then used in a full-step for $\mathbf{q}_{n+1}$ (itself found from the average of an explicit and an implicit calculation), before both $\mathbf{p}_{n+1/2}$ and $\mathbf{q}_{n+1}$ are used in an explicit half-step to calculate $\mathbf{p}_{n+1}$.

Note that, in both methods, despite the continuous system (4.13) being coupled, the

discretisation is such that the numerical systems become uncoupled, and equations can be solved individually, in sequence. On the other hand, consider the implicit midpoint method (e.g. [91, p. 3]):

$$\frac{\mathbf{p}_{n+1}-\mathbf{p}_n}{\Delta t} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}\left(\frac{\mathbf{p}_n+\mathbf{p}_{n+1}}{2}, \frac{\mathbf{q}_n+\mathbf{q}_{n+1}}{2}\right)$$
$$\frac{\mathbf{q}_{n+1}-\mathbf{q}_n}{\Delta t} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}\left(\frac{\mathbf{p}_n+\mathbf{p}_{n+1}}{2}, \frac{\mathbf{q}_n+\mathbf{q}_{n+1}}{2}\right).$$

In each equation, both $\mathbf{p}_{n+1}$ and $\mathbf{q}_{n+1}$ appear in the RHS, and the discrete equations cannot be uncoupled; they must be solved simultaneously. It is for this reason that we do not use the implicit midpoint method, particularly given that SV is also second-order, yet it allows a sequential, uncoupled solve to be carried out.

In (4.3) we have $\mathbf{p} = (\Phi, W, I)$ and $\mathbf{q} = (\eta, Z, Q)$. Now, both SE and SV are equally valid under the substitution $\mathbf{p} \leftrightarrow \mathbf{q}$, and we can swap components, providing we always keep the pairs $\{\Phi, \eta\}$, $\{W, Z\}$ and $\{I, Q\}$ separate. Since the equation for λ (4.3g) depends on η, Z and I, we shall swap I and Q, allowing λ to be included in $\mathbf{q}$.

4.2.3 Stability of the methods

For each method, the update formulae for $\mathbf{p}_{n+1}$ and $\mathbf{q}_{n+1}$ lead to a matrix equation of the form

$$\begin{pmatrix}\mathbf{p}_{n+1}\\ \mathbf{q}_{n+1}\end{pmatrix} = \mathbf{A}\begin{pmatrix}\mathbf{p}_n\\ \mathbf{q}_n\end{pmatrix},$$

with $\mathbf{A}$ describing how the solution advances during each time-step. Therefore, starting with $\mathbf{p}_0$ and $\mathbf{q}_0$, after $(n+1)$ steps we have

$$\begin{pmatrix}\mathbf{p}_{n+1}\\ \mathbf{q}_{n+1}\end{pmatrix} = \mathbf{A}^{n+1}\begin{pmatrix}\mathbf{p}_0\\ \mathbf{q}_0\end{pmatrix},$$

where our solutions have been obtained by applying the matrix $\mathbf{A}$ $(n+1)$ times.

For example, consider a simple dimensionless harmonic oscillator with eigenfrequency ω;

this has $\mathcal{H} = p^2/2 + (\omega q)^2/2$. Solving using SE gives

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \Delta t \omega^2 \mathbf{q}_n,$$

from which

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \Delta t \mathbf{p}_{n+1} = \Delta t \mathbf{p}_n + \left(1 - \Delta t^2 \omega^2\right) \mathbf{q}_n,$$

and we have

$$\mathbf{A}_{SE} = \begin{pmatrix} 1 & -\Delta t \omega^2 \\ \Delta t & 1 - \Delta t^2 \omega^2 \end{pmatrix}.$$

Similar manipulation of the SV equations gives

$$\mathbf{A}_{SV} = \begin{pmatrix} 1 - \frac{\Delta t^2 \omega^2}{2} & \Delta t \omega^2 \left(\frac{\Delta t^2 \omega^2}{4} - 1\right) \\ \Delta t & 1 - \frac{\Delta t^2 \omega^2}{2} \end{pmatrix}.$$

Now, $\mathbf{A}$ is diagonalisable if it can be written as $\mathbf{A} = \mathbf{P}\,\mathrm{diag}(\lambda_1, \lambda_2)\mathbf{P}^{-1}$, where λ_1 and λ_2 are the eigenvalues of $\mathbf{A}$, and $\mathbf{P}$ is the matrix formed by taking the eigenvectors of $\mathbf{A}$ as its columns. If this is the case, then $\mathbf{A}^{n+1} = \mathbf{P}\,\mathrm{diag}(\lambda_1^{n+1}, \lambda_2^{n+1})\mathbf{P}^{-1}$, and the eigenvalues act as amplification factors, with $\mathbf{p}_{n+1}$ and $\mathbf{q}_{n+1}$ both $\mathcal{O}\left(\max|\lambda_i|^{n+1}\right)$. In other words, the numerical solutions scale by the largest eigenvalue at each time-step. Therefore, $\max|\lambda_i| > 1$ leads to unstable solutions, while $\max|\lambda_i| < 1$ cause the solutions to decay; for stable solutions we require $\max|\lambda_i| = 1$.

For the matrices generated by the SE and SV methods, their eigenvalues satisfy

$$\lambda^2 - \mathrm{tr}(\mathbf{A})\lambda + \det(\mathbf{A}) = 0, \tag{4.16}$$

giving

$$\lambda = \frac{\mathrm{tr}(\mathbf{A})}{2} \pm \frac{1}{2}\sqrt{\mathrm{tr}(\mathbf{A})^2 - 4\det(\mathbf{A})}.$$

Note that both matrices have $\mathrm{tr}(\mathbf{A}) = 2 - \Delta t^2 \omega^2$ and $\det(\mathbf{A}) = 1$,[9] meaning they have the same eigenvalues, and thus the same stability conditions. Now, there are two cases to consider: when (4.16) emits either real or complex roots, which is dependent upon

[9] $\det(\mathbf{A}) = 1$ is in fact the case for any symplectic method.

the sign of $\mathrm{tr}(\mathbf{A})^2 - 4$.

Real roots occur when $\mathrm{tr}(\mathbf{A})^2 \geq 4$, which is equivalently $\Delta t\omega \geq 2$, since $\Delta t\omega$ must be positive. Since $\det(\mathbf{A}) = 1$, the two roots would be $\lambda_1 = a$ and $\lambda_2 = 1/a$ for some a; the stability condition would then require $|a| = 1$. Taking $a = 1$ leads to $\mathrm{tr}(\mathbf{A}) = 2$ and subsequently to $\Delta t\omega = 0$, which is not possible, while taking $a = -1$ leads to $\mathrm{tr}(\mathbf{A}) = -2$, giving $\Delta t\omega = 2$.

However, since in this case we have repeated eigenvalues, we are not guaranteed to have two linearly-independent eigenvectors, meaning the matrices may be defective and may not be diagonalisable. Indeed, note that with $\Delta t\omega = 2$ the matrices for each method become

$$\mathbf{A}_{SE} = \begin{pmatrix} 1 & -2\omega \\ \frac{2}{\omega} & -3 \end{pmatrix} \text{ and } \mathbf{A}_{SV} = \begin{pmatrix} -1 & 0 \\ \frac{2}{\omega} & -1 \end{pmatrix};$$

recalling that each of the eigenvalues for both matrices are $\lambda = -1$, it is straightforward to see that both matrices are defective. This means we cannot write $\mathbf{A} = \mathbf{P}\,\mathrm{diag}(\lambda_1, \lambda_2)\mathbf{P}^{-1}$. We can, however, take the Jordan canonical form $\mathbf{J}$, which for both matrices is

$$\mathbf{J} = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix};$$

this then enables us to write $\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1}$, with $\mathbf{A}^{n+1} = \mathbf{P}\mathbf{J}^{n+1}\mathbf{P}^{-1}$. However, note that

$$\mathbf{J}^{n+1} = \begin{pmatrix} (-1)^{n+1} & (n+1)(-1)^n \\ 0 & (-1)^{n+1} \end{pmatrix}.$$

The presence of the $n+1$ term in the off-diagonal entry means, despite $|\lambda| = 1$, $\mathbf{J}$ still grows with each time-step and the methods will be unstable, even for $\Delta t\omega = 2$.

The alternative is that (4.16) emits complex roots, when $\Delta t\omega < 2$. In this case, the eigenvalues λ are of the form

$$\lambda = \frac{\mathrm{tr}(\mathbf{A})}{2} \pm \frac{i}{2}\sqrt{4 - \mathrm{tr}(\mathbf{A})^2},$$

and we have

$$|\lambda| = \sqrt{\frac{\mathrm{tr}(\mathbf{A})^2}{4} + \frac{1}{4}(4 - \mathrm{tr}(\mathbf{A})^2)} = 1;$$

that is, since $\det(\mathbf{A}) = 1$, $|\lambda| = 1$ for any complex root. Additionally, since the eigenvalues are distinct, so are their eigenvectors, and $\mathbf{A}$ is diagonalisable. Therefore, both the SE and SV methods are unconditionally stable, provided the time-step criteria

$$\Delta t < \frac{2}{\omega} \tag{4.17}$$

is satisfied.

In practice, the SW system (4.3) contains multiple oscillators with different eigenfrequencies ω_i; in these cases, we require each eigenvalue associated with each oscillator to be complex, meaning (4.17) must hold for each ω_i. Providing this holds for the largest eigenfrequency, it will then hold for all eigenfrequencies; therefore, the stability criteria becomes

$$\Delta t < \frac{2}{\max \omega_i}.$$

Similarly, the spatial discretisation applied to the PDEs results in the addition of even more oscillators to the system, with eigenfrequencies dependent upon the element size. For the SW equations, the standard dispersion relation relating eigenfrequency ω with wave-number k is $\omega^2 = gH_0k^2$ (e.g. [98]). In 2D, $k^2 = k_x^2 + k_y^2$, where k_x and k_y are the x- and y-direction wave-numbers. Now, after discretising in space, the wave-number becomes inversely proportional to the step-size, with $k_x = \pi/\Delta x$ (and similarly for y). Thus, we have

$$\omega^2 = \pi^2 gH_0 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right).$$

Now, given Δx and Δy will be small for accurate numerical solutions, we can assume the largest eigenfrequency will always be the one associated with the hydrodynamics, and therefore we find a maximum time-step of

$$\Delta t < \frac{2}{\pi\sqrt{gH_0}\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}}.$$

In practice, since this is an idealised estimate of the maximum time-step, we shall scale by some number less than 1 to ensure stability.

4.2.4 Dealing with non-conservative damping terms

Recall that the equation for I (4.3f) contains a non-conservative damping term of the form $-\alpha I$, which was added to the equation after taking variations. Thus, it is not a standard term in the Lagrangian equations (4.13), and its discretisation must be dealt with independently.

Consider the equation $\dot{I} = -\alpha I$, which has an analytical solution of $I(t) = I(0)e^{-\alpha t}$. Let us solve this equation using both the explicit and implicit steps appearing in SE (4.14), and also the central symmetric step of SV (4.15). For these three methods, the discrete time-stepping equations are

$$\text{Explicit: } I_{n+1} = (1 - \alpha\Delta t)I_n \tag{4.18a}$$

$$\text{Implicit: } I_{n+1} = \frac{I_n}{(1 + \alpha\Delta t)} \tag{4.18b}$$

$$\text{Symmetric: } I_{n+1} = \frac{(1 - \frac{\alpha\Delta t}{2})}{(1 + \frac{\alpha\Delta t}{2})} I_n. \tag{4.18c}$$

Figure 4.7 presents the numerical solutions obtained using the three methods in (4.18), alongside the analytical solution, with $\alpha = 1$, $I(0) = 1$ and $\Delta t = 0.5$. Note how the explicit method under-estimates the solution, while the implicit method over-estimates. On the other hand, the symmetric method matches the analytical solution much more accurately, and should be used in all cases; while this symmetric approach is already employed by SV, the standard SE formulation will be enhanced with it.

4.3 Introducing Firedrake

As mentioned in Section 3.6, Bokhove et al. [44, 45] carried out FEM calculations using an in-house MATLAB code, in which the mass and stiffness matrices were explicitly calculated for linear ($n = 1$) basis functions, and for the linear 2D approximation only. Increasing the degree of basis polynomials or extending the problem to 3D would require

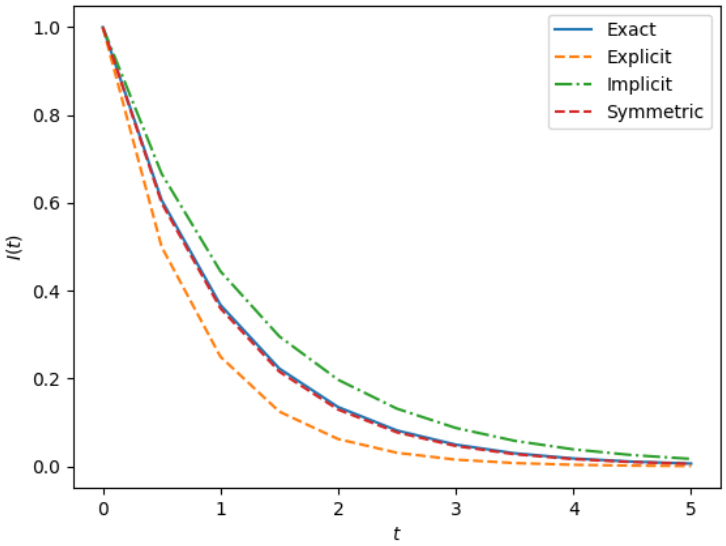


Figure 4.7: The analytical and numerical solutions to the equation $\dot{I} = -\alpha I$, using the three methods presented in (4.18).

extensive rewriting of this code, since entirely different matrices would need to be assembled. Similarly, solving the nonlinear version of the equations would require a rewrite, not only to obtain the nonlinear expressions f_i, but also to carry-out the iterations of Newton's method required to solve the nonlinear system. Thankfully, there exists an external package which can enable these extensions to be trivially carried out.

Firedrake [83] is an "automated system for the solution of partial differential equations using the finite element method".[10] Written in Python, it is an open-source FE library that uses the Unified Form Language (UFL) from the FEniCS project [99] to represent weak forms and leverages PETSc (the Portable, Extensible Toolkit for Scientific Computation) [100] to efficiently solve systems of linear and nonlinear equations. The cumbersome procedure of assembling and solving the matrix-vector equations is carried out "under-the-hood", meaning relatively little code (from the user's point-of-view) is required to produce solutions to even the most complicated of problems.

We shall explore the use of Firedrake by continuing with the example of the Laplace equation used in Section 4.1. The code used for this, presented in snippets in the text, can be found in-full at `https://github.com/jonnybolton16/waveEnergyPhDPublic`

[10] Available from `https://www.firedrakeproject.org/index.html`.

/blob/main/4_3solveLaplace/solveLaplace.py.

4.3.1 Toy problem: statement and analytical solution

Consider the Laplace equation $\nabla^2 u = 0$ in the 2D rectangular domain

$$(x, y) \in \Omega = \left[-\frac{W}{2}, \frac{W}{2}\right] \times \left[-\frac{H}{2}, \frac{H}{2}\right],$$

with the following Dirichlet BCs:

$$u\left(-\frac{W}{2}, y\right) = X_-(y) = 3\cos\left(\frac{\pi y}{H}\right) \qquad u\left(\frac{W}{2}, y\right) = X_+(y) = \cos\left(\frac{\pi y}{H}\right)$$
$$u\left(x, -\frac{H}{2}\right) = Y_-(x) = -2\cos\left(\frac{\pi x}{W}\right) \qquad u\left(x, \frac{H}{2}\right) = Y_+(x) = -2\cos\left(\frac{\pi x}{W}\right).$$

This domain is shown in Figure 4.8.

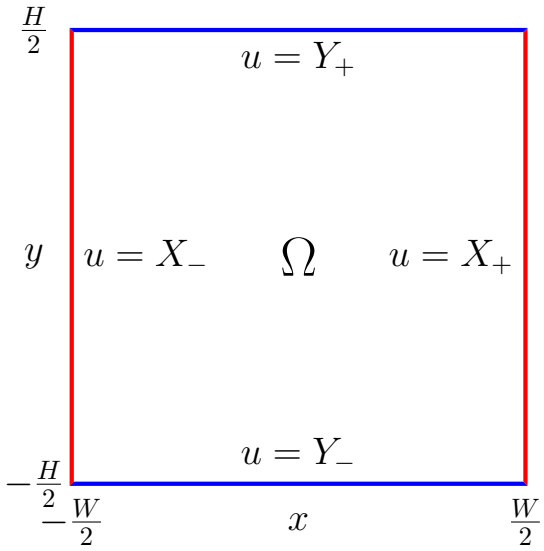


Figure 4.8: The domain Ω for the toy problem, including the Dirichlet BCs.

Due to its simplicity, the problem can be solved analytically, using separation of variables. Referring to Appendix A.6 for a detailed derivation, this analytical solution $u(x, y)$ is given by

$$u = \left\{3\sinh\left(\frac{\pi(W-2x)}{2H}\right) + \sinh\left(\frac{\pi(W+2x)}{2H}\right)\right\} \frac{\cos\left(\pi y/H\right)}{\sinh\left(\pi W/H\right)}$$
$$- 2\left\{\sinh\left(\frac{\pi(H-2y)}{2W}\right) + \sinh\left(\frac{\pi(H+2y)}{2W}\right)\right\} \frac{\cos\left(\pi x/W\right)}{\sinh\left(\pi H/W\right)}. \quad (4.19)$$

4.3.2 Firedrake solution workflow

We will now solve this equation using Firedrake. After importing `firedrake` as `fd`, we first generate the mesh[11] and obtain the x and y coordinate objects. Here, `nel` is the number of elements in both the x and y dimensions (so the mesh has a total of `nel`^{2} elements), while `W` and `H` are the domain's width and height.

```
mesh = fd.RectangleMesh(nel, nel, W/2, H/2, -W/2, -H/2,
    quadrilateral=True) # the default mesh is triangular
x, y = fd.SpatialCoordinate(mesh)
```

Next, we use `FunctionSpace` to set the family[12] and degree (equivalently, the "space") of the basis functions we would like to use, as well as `TrialFunction` and `TestFunction` to declare the trial (solution) and test functions.

```
V = fd.FunctionSpace(mesh, 'CG', degree) # 'CG' corresponds to
    continuous Lagrange polynomials
u = fd.TrialFunction(V)
v = fd.TestFunction(V)
```

Now, as we saw in Section 4.1, the weak form for the Laplace equation with only Dirichlet BCs is

$$\iiint_\Omega \nabla v \cdot \nabla u \, \mathrm{d}V = 0;$$

leveraging UFL's symbolic notation, this can simply be written in the code.[13]

```
a = fd.dot(fd.grad(v), fd.grad(u)) * fd.dx
L = 0
```

Next, we define the Dirichlet BCs. Again, the symbolic nature of the code makes this very straightforward; we use `DirichletBC`, which takes as arguments the `FunctionSpace`, a symbolic expression for the BC, and the ID(s) of the mesh's boundary(ies) on which the BC is to be applied (Firedrake's built-in meshes have pre-defined boundary IDs).

```
cos, pi = fd.cos, fd.pi
bcs = [
    fd.DirichletBC(V,  3*cos(pi*y/H),     1 ), # 1: x = -W/2
    fd.DirichletBC(V,    cos(pi*y/H),     2 ), # 2: x = W/2
    fd.DirichletBC(V, -2*cos(pi*x/W), (3, 4))  # (3, 4): y = (-H/2, H/2)
]
```

[11] Firedrake contains a number of built-in meshes (documented at https://www.firedrakeproject.org/firedrake.html#module-firedrake.utility_meshes), as well as the ability to generate custom meshes (as seen in Sections 4.4 and 5.2.2) or to read external mesh files in various formats.

[12] Each of the families outlined in [83, 4–6] has its own string that can be used here.

[13] The standard notation used to define a (linear) problem in Firedrake is $a(u, v) = L(v)$, where the LHS a is bilinear in u and v, and the RHS L is linear in v [83, p. 11].

Finally, the problem is solved using `solve`, with the solution stored in a `Function`. This object can be plotted, point-evaluated and analysed in a number of other ways.

```
u_num = fd.Function(V)
fd.solve(a==L, u_num, bcs=bcs)
```

To enable comparison with the numerical solution, the analytical solution (4.19) is stored in another `Function`, with `interpolate` used to set the value of the `Function`'s basis-function coefficients, again using simple symbolic notation.

```
sinh = fd.sinh
u_sol = fd.Function(V).interpolate(
      (3*sinh(pi*(W-2*x)/(2*H)) +
          sinh(pi*(W+2*x)/(2*H)))*cos(pi*y/H)/sinh(pi*W/H)
    - 2*(sinh(pi*(H-2*y)/(2*W)) +
        sinh(pi*(H+2*y)/(2*W)))*cos(pi*x/W)/sinh(pi*H/W)
)
```

Now, one measure of the accuracy of the numerical solution is the ℓ-2 error norm, defined as

$$|u| = \sqrt{\iiint_\Omega (\texttt{u\_sol} - \texttt{u\_num})^2 \, \mathrm{d}V} \,,$$

which can be calculated using `errornorm`.

```
norm = fd.errornorm(u_sol, u_num)
```

This short programme is all that is needed to solve this problem using Firedrake, and to compare the numerical solution to the analytical solution. Note that we did not need to calculate and build the complicated FE matrices and vectors, nor did we need to implement any matrix-solver algorithms; all of that complexity is carried out within the calls of the various Firedrake functions that we imported. As a result, changing the degree of polynomial used for the basis functions is as simple as changing the value of the `degree` parameter passed to `FunctionSpace`, while solving a 3D problem just requires a 3D mesh (e.g. `BoxMesh`) and additional `DirichletBC`s.

4.3.3 Results and discussion

In order to analyse FEM a little further, this problem was solved for a number of combinations of `nel` and `degree`, with `nel` $= 4 \cdot 2^i$ for $i \in [0, 4]$ (i.e. `nel` $= 4, 8, \ldots, 64$) and `degree` $\in [1, 4]$.

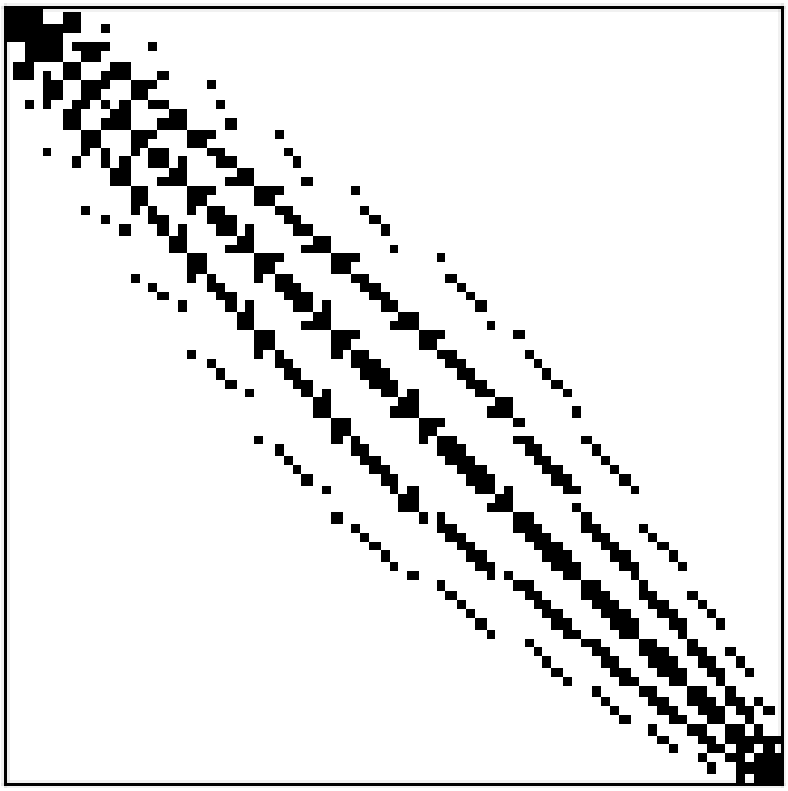


Figure 4.9: The sparsity pattern of the stiffness matrix appearing in the FE formulation of the Laplace equation, obtained directly from Firedrake via the UFL object `a`. Careful inspection reveals four rows with four entries each (these are the four corners of the domain, whose nodes span just one element each), some rows with six entries (the remaining boundary nodes, spanning two elements each), and the rest with nine entries (the internal nodes, each spanning four elements). Note that permuting the matrix can change the distribution of the non-zero entries about the main diagonal (i.e. the matrix's *bandwidth*), which can significantly impact computational performance; thankfully, Firedrake automatically orders the unknowns to optimise this measure [83, p. 1].

Figure 4.10 shows plots of the (interpolated) analytical solution `u_sol` (4.19),[14] the numerical solution `u_num` calculated by Firedrake, and the error $|\texttt{u\_sol} - \texttt{u\_num}|$, for $(\texttt{nel}, \texttt{degree}) = (4, 1)$ and $(64, 4)$. Even at the crudest level of resolution, the numerical solution already appears to be relatively accurate, with the correct saddle shape starting to be seen. At the highest resolution, the numerical solution appears to be almost perfect, with a maximum error of $\mathcal{O}\left(10^{-12}\right)$.

Table 4.1 contains the ℓ-2 error norm $|u|$ for the different combinations of `nel` and

[14] When storing analytical functions in Firedrake, an interpolated representation is provided, where the values at the nodes match the analytical solution, but over the elements the nodal values are interpolated using the same basis functions as the numerical solution.

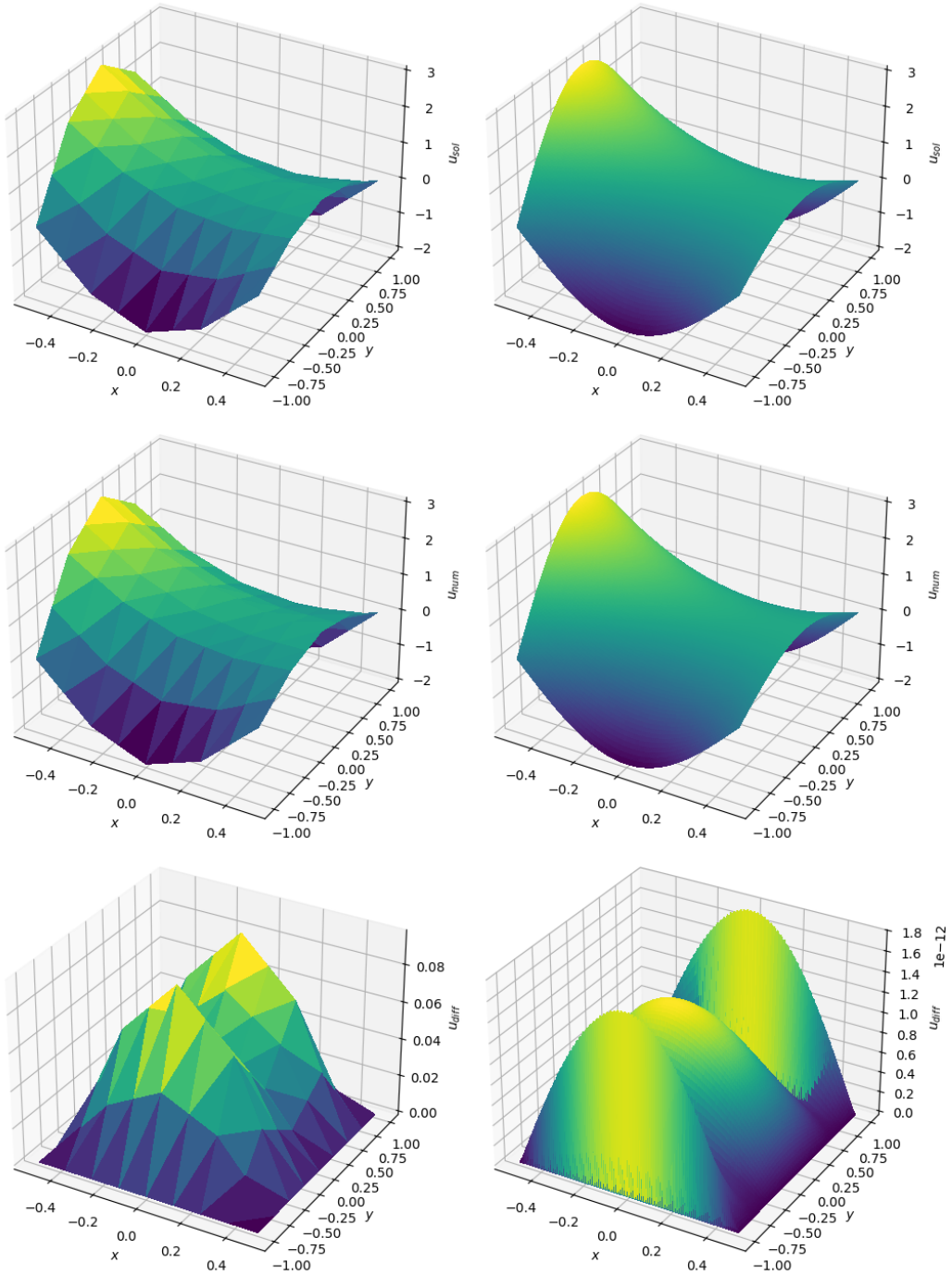


Figure 4.10: The solutions to the Laplace problem. Top row: the (interpolated) analytical solution `u_sol`. Middle row: the numerical solution `u_num` calculated by Firedrake. Bottom row: the error $|$`u_sol` $-$ `u_num`$|$. Left column: linear basis functions and `nel` $= 4$ elements in both x and y. Right column: quartic basis functions, `nel` $= 64$.

`degree`; as expected, we can clearly see that increasing either the number of elements or the degree of the polynomial reduces the error. Table 4.2 shows values of $\log_2 \frac{|u|_{\texttt{nel}}}{|u|_{\texttt{2nel}}}$, for each `degree`; this is a measure of the proportional rate-of-decrease in the error when

Table 4.1: ℓ-2 error norms for a selection of `nel` and `degree` values.

`nel`	`degree` 1	2	3	4
4	6.15e-02	1.84e-03	1.20e-04	7.08e-06
8	1.62e-02	1.26e-04	4.04e-06	1.22e-07
16	4.09e-03	8.08e-06	1.27e-07	1.96e-09
32	1.03e-03	5.08e-07	3.98e-09	3.07e-11
64	2.57e-04	3.18e-08	1.24e-10	4.82e-13

the number of elements is doubled. We can see that, for linear polynomials, the error is reduced by approximately 2^2 each time the number of elements is doubled, meaning it is a second-order method. For the higher-degree polynomials, we see a reduction of approximately $2^{\texttt{degree}+2}$; this suggests some super-convergence of the method, since it is well known that FEM using continuous degree-d Lagrange polynomials produces a method of order $d+1$.

Table 4.2: $\log_2$ of the ratio of norms for successive values of `nel`, for each value of `degree`.

`nel`	`degree` 1	2	3	4
$4 \to 8$	1.9257	3.8673	4.8959	5.8584
$8 \to 16$	1.9834	3.9658	4.9891	5.9637
$16 \to 32$	1.9962	3.9913	4.9988	5.9909
$32 \to 64$	1.9991	3.9978	4.9998	5.9946

4.4 Solving a problem on a subdomain

The previous example considers solving an equation over the entire domain. In our system, we require the solution of an equation for λ over only a subset of the domain (the portion of the surface underneath the buoy). We shall deal with such a subdomain by utilising a `DirichletBC`, explicitly setting the value of the solution outside of the subdomain to be 0. To test this methodology, we shall consider first a steady 1D problem, then a time-dependent problem, and finally expanding to 2D. The BCs for the λ equation involve both Dirichlet and Neumann BCs, so our toy problem will consider similar BCs. The codes for these problems can be found at https://github.com/jonnybolton16/waveEnergyPhDPublic/tree/main/4_4subdomain.

4.4.1 1D

In 1D, we shall solve the following equation:

$$\frac{\partial^2 u}{\partial x^2} = f \text{ with } u\left(\frac{1}{2}\right) = g \text{ and } \left.\frac{\partial u}{\partial x}\right|_{x=1} = h$$

on the subdomain $x \geq 1/2$ in the parent domain $x \in [0, 1]$.

Reduced domain

We first solve for only $x \in [1/2, 1]$. The analytical solution is given by

$$u = \frac{fx^2}{2} + (h - f)x + g - \frac{f}{8} + \frac{f - h}{2},$$

while the weak form is

$$\int_{0.5}^{1} \frac{\partial v}{\partial x}\frac{\partial u}{\partial x}\,\mathrm{d}x = v(1)h - \int_{0.5}^{1} vf\,\mathrm{d}x \text{ with Dirichlet BC } u\left(\frac{1}{2}\right) = g.$$

The solution methodology proceeds in much of the same way as in the previous section, with one additional feature: the surface integral (point evaluation in 1D) appearing in the weak form's RHS, arising from the Neumann BC. This is implemented using `ds`, with the mesh's relevant boundary ID as an argument.

```
mesh = fd.IntervalMesh(nel, 0.5, 1)
x, = fd.SpatialCoordinate(mesh)
...
L = v*h*fd.ds(2) - v*f*fd.dx # 2 corresponds to the right-hand boundary
    of IntervalMesh
...
bcs = fd.DirichletBC(V, g, 1) # 1 corresponds to the left-hand boundary
...
u_sol = fd.Function(V).interpolate(
    0.5*f*x**2 + (h-f)*x + g - 0.125*f + 0.5*(f-h)
)
```

Subdomain

Next, we extend the solution for the full domain $x \in [0, 1]$. As u is undefined for $x < 1/2$, we shall set it to be zero there, such that the analytical solution is now

$$u = \begin{cases} 0 & x < \frac{1}{2} \\ \frac{fx^2}{2} + (h - f)x + g - \frac{f}{8} + \frac{f-h}{2} & x \geq \frac{1}{2} \end{cases}.$$

In this case, the weak form is:

$$\int_0^1 \frac{\partial v}{\partial x}\frac{\partial u}{\partial x}\,\mathrm{d}x = v(1)h - \int_0^1 vf\,\mathrm{d}x \text{ with Dirichlet BCs } u\left(\frac{1}{2}\right) = g \text{ and } u\left(x < \frac{1}{2}\right) = 0.$$

Note the additional Dirichlet BCs explicitly setting $u = 0$ for $x < 1/2$.

There is clearly a discontinuity when $g \neq 0$; to ensure the consistency of the numerical solution, we require this discontinuity to be smoothed out over the first element beyond the subdomain's boundary. Thankfully, the way in which Firedrake interpolates between nodes readily allows this to happen.[15]

First, a custom mesh in which all nodes $x < 1/2$ are given a "boundary ID" is required; this is built using an amended version of the source code for `IntervalMesh`,[16] where an additional `if` condition is used to apply the new label of 0 to nodes outside of the subdomain.

```
# use 2*nel elements so the step-size is the same as the reduced problem
coords = np.linspace(0, 1, 2*nel+1, dtype=np.double).reshape(-1, 1)
# coords[i] returns the x coordinate of the ith global node
...
# Apply boundary IDs by looping over nodes
for v in range(vStart, vEnd):
    vcoord = ...
    if vcoord[0] == coords[nel]:
        # node is on subdomain boundary
        plex.setLabelValue(..., 1)
    if vcoord[0] == coords[-1]:
        # node is on right-hand boundary
        plex.setLabelValue(..., 2)
    if vcoord[0] < coords[nel]:
```

[15] One might attempt to remove this discontinuity by solving for $w = u - g$ instead, but since u is required to be 0 outside the subdomain, w would then be $-g$ and the discontinuity would merely be shifted. If one did set $w = 0$ to remove the discontinuity, when recovering u after the solve it would simply reappear.

[16] See https://www.firedrakeproject.org/_modules/firedrake/utility_meshes.html#IntervalMesh.

```
        # node is outside of subdomain
        plex.setLabelValue(..., 0)
```

The problem is then solved as before, with the additional `DirichletBC` on ID 0 included, and `conditional` used to define the piecewise analytical solution.

```
bcs = [fd.DirichletBC(V, 0, 0), fd.DirichletBC(V, g, 1)]
...
u_sol = fd.Function(V).interpolate(fd.conditional(
    x<0.5, 0,
    0.5*f*x**2 + (h-f)*x + g - 0.125*f + 0.5*(f-h)
))
```

Time-dependent

Finally, we solve an unsteady problem: let $g \to g \cos t$, such that the Dirichlet BC at $x = 1/2$ becomes time-dependent. Since there are no time derivatives present, with just a time-dependent forcing (supplied by the BCs), the solution methodology is simple: at each time-step, we solve the same steady problem, but with an updated BC applied. Again, thanks to Firedrake's use of UFL, this is incredibly straightforward.

First, rather than using the basic Python integer/float `g` as before, we create a `Constant` to store the value of `g`; at a later point, we can use `assign` to update this stored value.

```
G = fd.Constant(g*cos(0))
bcs = [fd.DirichletBC(V, 0, 0), fd.DirichletBC(V, G, 1)]
```

Next, instead of calling `solve` as before, we create a `LinearVariationalProblem` to store the problem, and a `LinearVariationalSolver` to store its solver. For any iterative solution methodology, where the same LHS is repeatedly solved for a varying RHS, declaring these objects once and repeatedly calling the `solve` method of the `Solver` is much more efficient than repeated calls to the direct `solve` function, as the associated FE matrices for the unchanging LHS are assembled just once, and are cached for later use.[17]

```
problem = fd.LinearVariationalProblem(a, L, u_num, bcs=bcs)
solver = fd.LinearVariationalSolver(problem)
```

Now, for the magic of UFL. Since the `Constant` was passed to the `DirichletBC`, which itself was passed to the `Solver` (via the `Problem`), the `Solver` can "see" the `Constant`

[17] In-fact, under-the-hood, the `solve` function generates these objects itself; the key difference is whether they are reused, or are discarded and re-created each time-step.

throughout the simulation; updates to the `Constant`'s value at each time-step are automatically carried through to the `Solver`, meaning the `Solver` can simply be solved again, with the updated `DirichletBC` automatically applied. Similarly, the `Function` that was also passed to the `Problem` is automatically updated with the new solution.

```
# time loop at time t
G.assign(g*cos(t))
solver.solve()
# u_num now contains the new solution
```

Results

This problem was solved with $f = 15$, $g = 1$ and $h = 5$, on a mesh with `nel` $= 5$ elements and using values for `degree` $= 1, 3$. Figure 4.11 contains plots of the analytical and numerical solutions, and the error, for both the reduced domain and the subdomain, and for linear and cubic basis functions. Furthermore, animations of the time-dependent results (where $t \in [0, 2\pi]$) can be found on GitHub.

When using linear basis functions (first two rows), we clearly see good correspondence between the analytical and numerical solutions, with errors of $\mathcal{O}\left(10^{-16}\right)$. More importantly, we can see how the numerical results of the subdomain problem satisfy the smoothing-out requirement: over the first element beyond the subdomain boundary ($x \in [0.4, 0.5]$ in this case), the numerical solution is interpolated between g and 0, and is identically 0 everywhere beyond this element.

Unfortunately, however, when using cubic basis functions (bottom two rows) we can see that the `DirichletBC` has not properly constrained the solution to be 0 everywhere outside of the subdomain: the solution is 0 at the element boundaries, but not at the intra-element nodes. It seems that this `DirichletBC` approach successfully fixes the solution at the mesh's nodal points, but does not fix the value at the additional nodes that arise when higher-order basis functions are used.

Recall the code used to mark the $x < 1/2$ nodes. The loop is over the x-coordinates of the mesh's nodes; that is, no intra-element positions are considered, and they are subsequently not included in the label. Therefore, it is understandable that the `DirichletBC` has not constrained their values.

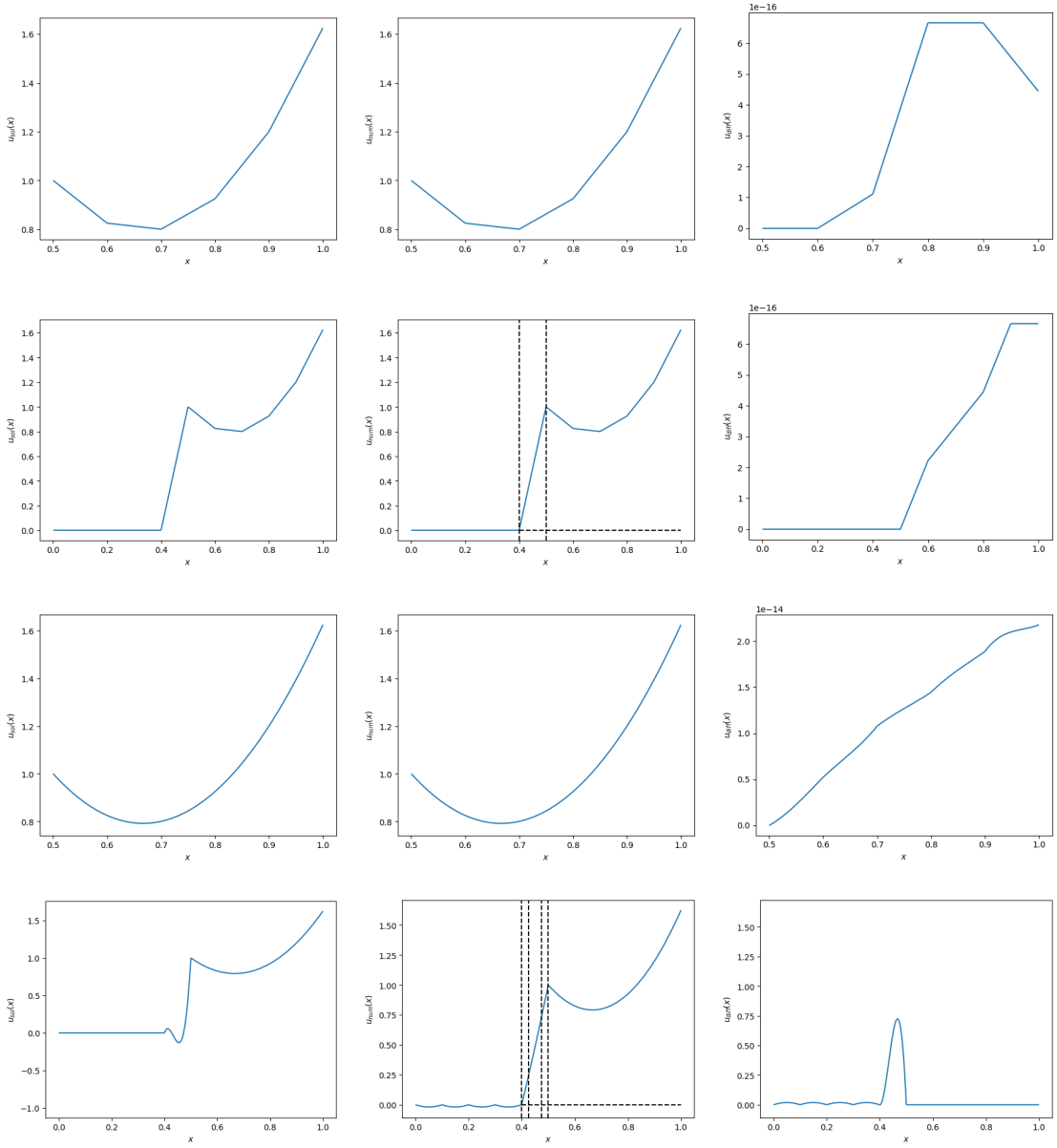


Figure 4.11: The solutions to the 1D problem. Left column: the (interpolated) analytical solution `u_sol`. Middle column: the numerical solution `u_num`. Right column: the error $|$`u_sol` $-$ `u_num`$|$. Top row: the reduced domain with linear basis functions. Second row: the subdomain with linear basis functions—the smoothing-out of the discontinuity over the first element outside of the subdomain is outlined by the dotted lines. Third row: reduced domain, cubic functions. Bottom row: subdomain, cubic functions—the inability of the method to constrain the intra-element nodal values is evident.

```
for v in range(vStart, vEnd):
    vcoord = ...
    if vcoord[0] < coords[nel]:
        plex.setLabelValue(..., 0)
```

4.4.2 2D

Next, we shall test an implementation of the `DirichletBC` method in 2D. Let $u(x, y, t)$ on the rectangular domain $(x, y) \in [-1, 1] \times [-1, 1]$ be defined as

$$u = \begin{cases} 0 & x < 0 \\ \frac{\cosh(\pi(x-1))}{\cosh \pi} \cos(\pi(y + \cos t)) & x \geq 0 \end{cases}. \tag{4.20}$$

Then, for $x \geq 0$, u satisfies $\nabla^2 u = 0$, with BCs

$$u(0, y, t) = \cos(\pi(y + \cos t)) \qquad \frac{\partial u}{\partial x}(1, y, t) = 0$$
$$\frac{\partial u}{\partial y}(x, -1, t) = \frac{\partial u}{\partial y}(x, 1, t) = \frac{\pi \sin(\pi \cos t)}{\cosh \pi} \cosh(\pi(x - 1)).$$

This has a weak form of

$$\int_0^1 \int_{-1}^1 \nabla v \cdot \nabla u \, \mathrm{d}y \, \mathrm{d}x = \frac{\pi \sin(\pi \cos t)}{\cosh \pi} \int_{-1}^1 \cosh(\pi(x - 1))(v(x, 1) - v(x, -1)) \, \mathrm{d}x,$$ [18]

with Dirichlet BCs $u(x < 0, y, t) = 0$ and $u(0, y, t) = \cos(\pi(y + \cos t))$.

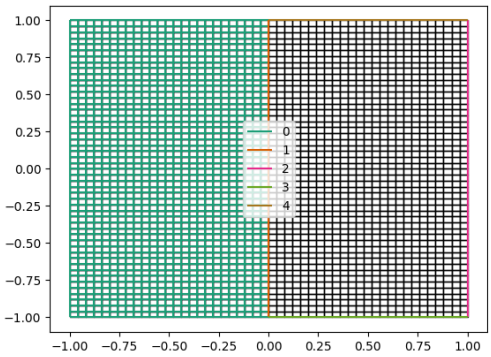


Figure 4.12: The custom mesh, with the boundary IDs highlighted. The entire left-half of the domain is marked with the ID 0.

First, we build another custom mesh (this time based on `TensorRectangleMesh`)[19]—plotted in Figure 4.12—and then follow the now-familiar process of initialising and solving the problem. Figure 4.13 contains plots of the results with `nel` $= 50$ and `degree` $= 1, 3$, and, as before, an animation of the time-dependent result can be found

[18] Noting that the outward-pointing normal at $y = -1$ causes a sign flip.

[19] https://www.firedrakeproject.org/_modules/firedrake/utility_meshes.html#TensorRectangleMesh.

on GitHub. Good correspondence is seen in the linear case, but in the cubic case we again see some discrepancy surrounding the smoothing-out of the discontinuity.

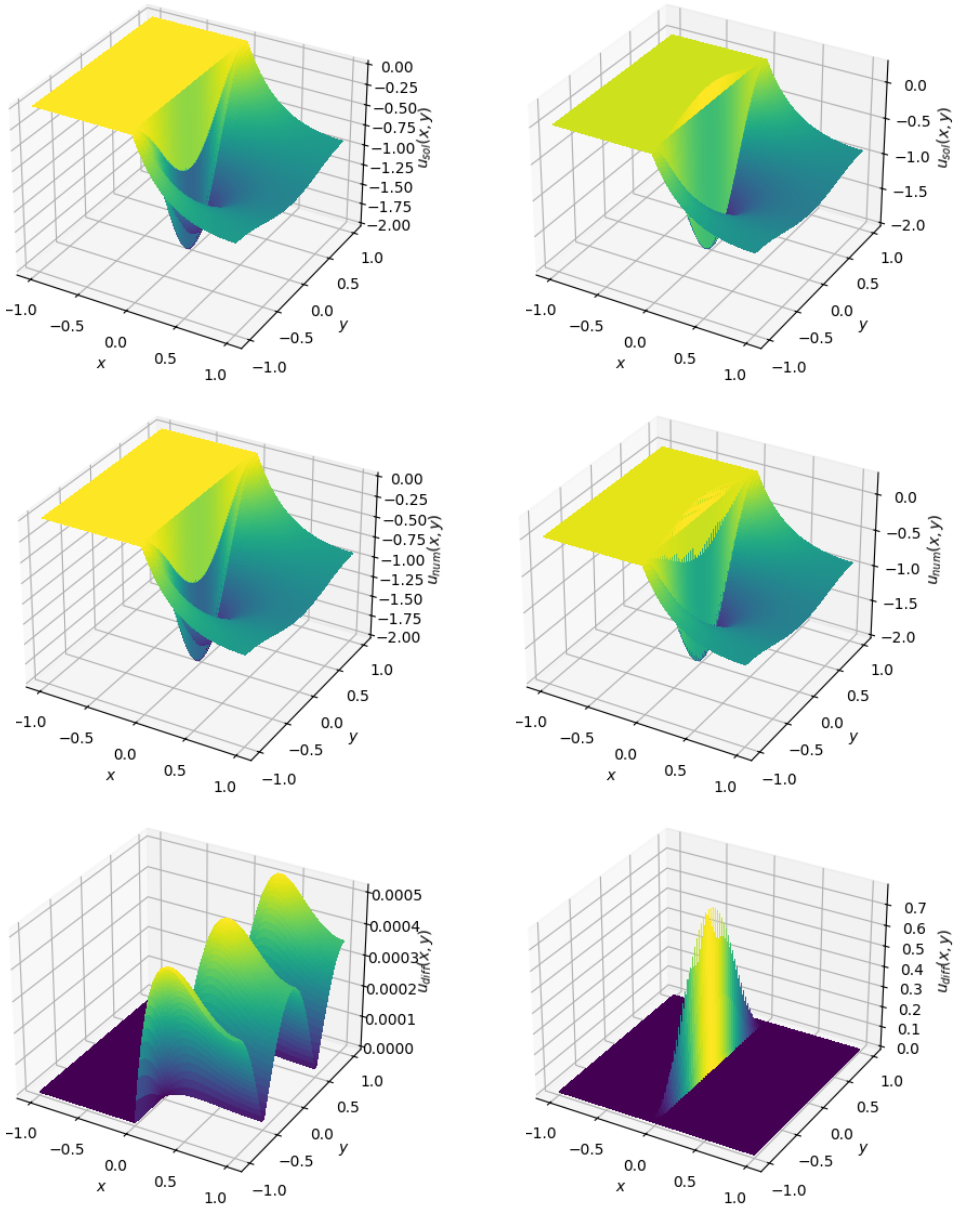


Figure 4.13: The solutions to the 2D problem. Top row: the (interpolated) analytical solution `u_sol`. Middle row: the numerical solution `u_num`. Bottom row: the error $|$`u_sol` $-$ `u_num`$|$. Left column: linear basis functions. Right column: cubic basis functions.

We can confirm this by extracting the actual basis-function coefficient arrays stored inside the `Functions`, accessed via `dat.data_ro`. We also need to map the indices of these arrays to their spatial coordinates, which we can do by interpolating the `SpatialCoordinate` object `x` onto a `Function`.

```
X = fd.Function(V).interpolate(x)
idx = fd.np.where(X.dat.data_ro < 0)
abs_u = max(abs(u_sol.dat.data_ro[idx]))
max_idx = fd.np.argmax(abs_u)
print(
    f'Largest value of u_sol for x<0: {abs_u} at
        x={X.dat.data_ro[max_idx]}'
)
abs_u = max(abs(u_num.dat.data_ro[idx]))
max_idx = fd.np.argmax(abs_u)
print(
    f'Largest value of u_sol for x<0: {abs_u} at
        x={X.dat.data_ro[max_idx]}'
)
```

Checking these arrays reveals that, whilst the linear `u_num` has all $x < 0$ coefficients equal to 0, the cubic `u_num` does indeed have nonzero coefficients; particularly at $x \approx -0.029$ and $x \approx -0.011$, which we note are the intra-element nodes for the first element beyond the subdomain boundary $x \in [-0.04, 0]$. As was the case in 1D, the extension into this element is not being applied properly, since the solution at these intra-element nodes has not been set to 0 by the `DirichletBC`. It is interesting to note, however, that whereas the 1D solution had additional nonzero coefficients beyond this first element, we do not see the same in 2D.

In conclusion, keeping in-mind the slight inaccuracy in this small region just beyond the subdomain boundary, it seems this `DirichletBC` approach can be successfully employed in 2D. An alternative approach, utilising recent Firedrake developments involving the `Submesh` functionality, may improve the smoothing-out of the discontinuity. However, this has not yet been shown to work; see Chapter 7 for a further discussion on this possible methodology.

Chapter 5

Developing a Flexible, Parametrisable, Numerical Algorithm for the Device

To aid in the testing and optimisation of the device, a flexible and parametrisable algorithm solving the SW system (4.3) has been developed. Using the methodologies outlined in the previous chapter, this algorithm takes an object of parameter values as an input, calculates the numerical solution over time, and produces data enabling the solution to be visualised and the generated power to be calculated.

The algorithm consists of:

- a flexible parameter definition structure,
- the generation of a custom mesh,
- UFL weak-form construction,
- PETSc solver control,
- time-stepping solution, and
- output data extraction and storage.

The full code for the algorithm is available at https://github.com/jonnybolton16/w

aveEnergyPhDPublic/blob/main/5algorithm/main.py.

First, the time-discrete, weak-form representation of (4.3) that Firedrake will solve, for both SE and SV, are presented (Section 5.1). Then, the various steps of the algorithm are detailed (Section 5.2), while Section 5.3 discusses the specific matrix solver options passed to PETSc, and the theory behind them. Finally, Section 5.4 discusses the improvements that this algorithm offers compared to the MATLAB code, while Section 5.5 presents examples of its use to carry out some preliminary optimisation investigations.

5.1 Time-discrete weak forms

Before we can begin with the development of the algorithm, we need to approximate (4.3) by the final time-discrete system of weak forms and ODEs that will be solved. The weak forms are derived using the process outlined in Section 4.1, while the time derivatives are discretised using either SE (4.14) or SV (4.15), with the inclusion of the symmetric damping term in the equation for I.

Notice, however, the λ-integral appearing in the strong form (4.3g); turning this equation into a weak form directly would generate a nested integral containing the unknown λ, which Firedrake is not able to handle. Recalling that this integral represents the force under the buoy, we introduce the scalar quantity $F(t)$, satisfying

$$F = \rho_0 \int_0^{L_x} \int_0^{l_y} \lambda \Theta(y - L_b) \, \mathrm{d}y \, \mathrm{d}x. \tag{5.1}$$

Rather than being evaluated directly, (5.1) is imposed weakly via

$$\int_0^{L_x} \int_0^{l_y} v \left(\frac{F}{A} - \rho_0 \lambda \Theta(y - L_b) \right) \mathrm{d}y \, \mathrm{d}x = 0, \tag{5.2}$$

where the division of F by A (the area of the domain) is required to normalise the expression, since integrating the spatially-invariant F introduces a factor of A. Provided that (5.2) holds for arbitrary v, it enforces (5.1) while avoiding the problematic nested integral. Consequently, (5.2) becomes an additional weak form to be solved, while the unknown F replaces the integral in (4.3g).

Symplectic Euler

The full time-discrete system of weak forms, using SE with symmetric damping for I in red, is as follows:

First step $\mathbf{p}_{n+1}$:

$$\int_0^{L_x}\int_0^{l_y} v\left(\frac{(\Phi_{n+1}-\Phi_n)}{\Delta t}+g\eta_n+\lambda_n\Theta(y-L_b)\right)\mathrm{d}y\,\mathrm{d}x=0, \tag{5.3a}$$

$$M\frac{(W_{n+1}-W_n)}{\Delta t}=F_n-\gamma G(Z_0)I_n,$$

$$\Rightarrow W_{n+1}=W_n+\frac{\Delta t}{M}(F_n-\gamma G(Z_0)I_n), \tag{5.3b}$$

$$\frac{(Q_{n+1}-Q_n)}{\Delta t}=I_n,$$

$$\Rightarrow Q_{n+1}=Q_n+\Delta t I_n, \tag{5.3c}$$

Second step $\mathbf{q}_{n+1}$:

$$\int_0^{L_x}\int_0^{l_y}\left\{v\frac{(\eta_{n+1}-\eta_n)}{\Delta t}-H\nabla v\cdot\nabla\Phi_{n+1}\right\}\mathrm{d}y\,\mathrm{d}x-H_0\int_0^{L_x}v|_{y=0}\dot{R}_{n+1}\,\mathrm{d}x=0, \tag{5.3d}$$

$$\frac{(Z_{n+1}-Z_n)}{\Delta t}=W_{n+1},$$

$$\Rightarrow Z_{n+1}=Z_n+\Delta t W_{n+1}, \tag{5.3e}$$

$$L_i\frac{(I_{n+1}-I_n)}{\Delta t}=\gamma G(Z_0)W_{n+1}-\frac{Q_{n+1}}{C}-\frac{(I_{n+1}+I_n)}{2}(R_i+R_c+R_l),$$

$$\Rightarrow I_{n+1}=\frac{L_iI_n+\Delta t\left(\gamma G(Z_0)W_{n+1}-\frac{Q_{n+1}}{C}-\frac{I_n}{2}(R_i+R_c+R_l)\right)}{L_i+\frac{\Delta t}{2}(R_i+R_c+R_l)}, \tag{5.3f}$$

$$\int_0^{L_x}\int_0^{l_y}\left\{H\nabla v\cdot\nabla(\lambda_{n+1}+g\eta_{n+1})+\frac{v}{M}(F_{n+1}-\gamma G(Z_0)I_{n+1})\right\}\Theta(y-L_b)\,\mathrm{d}y\,\mathrm{d}x=0, \tag{5.3g}$$

$$\int_0^{L_x}\int_0^{l_y} v\left(\frac{F_{n+1}}{A}-\rho_0\lambda_{n+1}\Theta(y-L_b)\right)\mathrm{d}y\,\mathrm{d}x=0, \tag{5.3h}$$

Dirichlet BC:

$$\lambda_{n+1}=g(\eta_{n+1}-Z_{n+1})\ \text{ at }\ y=L_b. \tag{5.3i}$$

Störmer–Verlet

Similarly, the system employing SV is:

First step $\mathbf{p}_{n+1/2}$:

$$\int_0^{L_x}\int_0^{l_y} v\left(\frac{(\Phi_{n+1/2}-\Phi_n)}{\Delta t/2}+g\eta_n+\lambda_n\Theta(y-L_b)\right)\mathrm{d}y\,\mathrm{d}x=0, \tag{5.4a}$$

$$M\frac{(W_{n+1/2}-W_n)}{\Delta t/2}=F_n-\gamma G(Z_0)I_n,$$

$$\Rightarrow W_{n+1/2}=W_n+\frac{\Delta t}{2M}(F_n-\gamma G(Z_0)I_n), \tag{5.4b}$$

$$\frac{(Q_{n+1/2}-Q_n)}{\Delta t/2}=I_n,$$

$$\Rightarrow Q_{n+1/2}=Q_n+\frac{\Delta t I_n}{2}, \tag{5.4c}$$

Second step $\mathbf{q}_{n+1}$:

$$\int_0^{L_x}\int_0^{l_y}\left\{v\frac{(\eta_{n+1}-\eta_n)}{\Delta t}-H\nabla v\cdot\nabla\Phi_{n+1/2}\right\}\mathrm{d}y\,\mathrm{d}x-H_0\int_0^{L_x}v|_{y=0}\dot{R}_{n+1/2}\,\mathrm{d}x=0, \tag{5.4d}$$

$$\frac{(Z_{n+1}-Z_n)}{\Delta t}=W_{n+1/2},$$

$$\Rightarrow Z_{n+1}=Z_n+\Delta t W_{n+1/2}, \tag{5.4e}$$

$$L_i\frac{(I_{n+1}-I_n)}{\Delta t}=\gamma G(Z_0)W_{n+1/2}-\frac{Q_{n+1/2}}{C}-\frac{(I_{n+1}+I_n)}{2}(R_i+R_c+R_l),$$

$$\Rightarrow I_{n+1}=\frac{L_iI_n+\Delta t\left(\gamma G(Z_0)W_{n+1/2}-\frac{Q_{n+1/2}}{C}-\frac{I_n}{2}(R_i+R_c+R_l)\right)}{L_i+\frac{\Delta t}{2}(R_i+R_c+R_l)}, \tag{5.4f}$$

$$\int_0^{L_x}\int_0^{l_y}\left\{H\nabla v\cdot\nabla(\lambda_{n+1}+g\eta_{n+1})+\frac{v}{M}(F_{n+1}-\gamma G(Z_0)I_{n+1})\right\}\Theta(y-L_b)\,\mathrm{d}y\,\mathrm{d}x=0, \tag{5.4g}$$

$$\int_0^{L_x}\int_0^{l_y} v\left(\frac{F_{n+1}}{A}-\rho_0\lambda_{n+1}\Theta(y-L_b)\right)\mathrm{d}y\,\mathrm{d}x=0, \tag{5.4h}$$

Dirichlet BC:

$$\lambda_{n+1}=g(\eta_{n+1}-Z_{n+1})\ \text{ at }\ y=L_b, \tag{5.4i}$$

Third step $\mathbf{p}_{n+1}$:

$$\int_0^{L_x}\int_0^{l_y} v\left(\frac{(\Phi_{n+1}-\Phi_{n+1/2})}{\Delta t/2}+g\eta_{n+1}+\lambda_{n+1}\Theta(y-L_b)\right)\mathrm{d}y\,\mathrm{d}x=0, \tag{5.4j}$$

$$M\frac{(W_{n+1}-W_{n+1/2})}{\Delta t/2}=F_{n+1}-\gamma G(Z_0)I_{n+1},$$

$$\Rightarrow W_{n+1}=W_{n+1/2}+\frac{\Delta t}{2M}(F_{n+1}-\gamma G(Z_0)I_{n+1}), \tag{5.4k}$$

$$\frac{(Q_{n+1}-Q_{n+1/2})}{\Delta t/2}=I_{n+1},$$

$$\Rightarrow Q_{n+1}=Q_{n+1/2}+\frac{\Delta t I_{n+1}}{2}. \tag{5.4l}$$

Note that when deriving the weak form for λ, boundary terms on the gradients of λ and η appear. Applying the $\nabla(\_) \cdot \hat{\mathbf{n}}$ operator to the Bernoulli equation (4.3b) and using the known BC on Φ produces the following BC on λ and η: $\nabla(\lambda + g\eta) \cdot \hat{\mathbf{n}} = 0$.[1] Thus, these boundary terms cancel.

Upon closer inspection of the time-discrete terms in (5.3) and (5.4), and assuming each of the values at time-step n are known, the time-discretisation has enabled a near-perfect decoupling of the equations, in which the majority of the $(n+1)$ values can be obtained in turn by solving the equations one-at-a-time, in the order presented. The only unknowns for which this is not the case are λ_{n+1} and F_{n+1}; neither can be obtained entirely independently, as both require knowledge of the other in order to be calculated. Therefore, these two expressions need to be solved together, as a coupled system; see Section 5.2.3 for a discussion on how coupled systems are solved in Firedrake.

5.2 The algorithm

5.2.1 Parameters

The first step is to define the various parameter values that the algorithm will use, which is made easily customisable by creating a `params` class to hold the values. This class is passed to the `main` code, which will first read the parameter values and calculate any derived quantities. Many of the parameter values are self-explanatory, while comments are included to provide additional explanations where helpful. The template uses the values presented in Table 3.2.

```
1  import main
2
3  class params:
4      # geometry and mesh
5      Lx = 0.2
6      Ly = 2
7      Nx = 10
8      Ny = 50
9      theta = 68.26 # d = 0.27 # Lc = 0.25 # only pass one of these
10
11     # water
12     g = 9.81
13     H0 = 0.1
14     rho0 = 997
15
```

[1] Here we report a correction to [44], which states incorrectly that $\nabla\lambda \cdot \hat{\mathbf{n}} = 0$ (p. 236, eqn. 25h).

```
    # buoy
    buoy = True # if False, no buoy is modeled
    M = 0.1
    alpha = 0.6418818298648808
    # calculated using atan(3*M*tan(radians(theta))/rho0/(Ly-Lb)**3)
    # with Lb = 1.8996831214658347 (from mesh coordinates)

    # generator
    m = 0.1
    a = 0.04
    D = 0.2769e-3
    K = 0.53
    L = 0.08
    sigma = 5.96e7
    nq = 1
    VT = 2.05
    Isat = 0.02
    alphah = 0.2
    Hm = 0.2
    C = float('inf') # a capacitance of inf models no capacitor

    # wavemaker
    wavenumber = 6
    smooth = False  # if wavemaker start/stop is smooth or not
    up_period = 2   # periods taken to reach max amplitude (when smooth)
    off_period = 10 # number of periods until wavemaker is turned off

    # time
    time_scheme = 'SE' # SE or SV
    T = 20
    T_in_periods = True # if True, T *= period

    # Lagrange polynomial degree
    CGN = 1

    # save parameters
    save_folder = 'outputs'
    gif_save = 10   # save .gif and/or .pvd file every n time-steps
    n_yvals = 101   # number of centerline points to sample for .gif
    pvd_save = None # if None, no file is created

main.main(params)
```

5.2.2 Mesh

After defining the parameter values used by the algorithm, and before the UFL objects holding the weak forms can be built, the next step is to define and build the custom Firedrake mesh. The script used to produce the mesh can be found at https://github.com/jonnybolton16/waveEnergyPhDPublic/blob/main/5algorithm/meshing.py.

The mesh is built entirely from quadrilaterals (which are generally more accurate and more efficient then triangular elements in structured geometries [101]), with the two distinct sections of the tank defined separately; a typical Cartesian grid is used in the

rectangular part, while a combination of parallelograms, scalene quadrilaterals and kites allow the triangular shape of the contraction to be meshed using a semi-regular pattern of quadrilaterals. See Figure 5.1 for an example of this pattern.

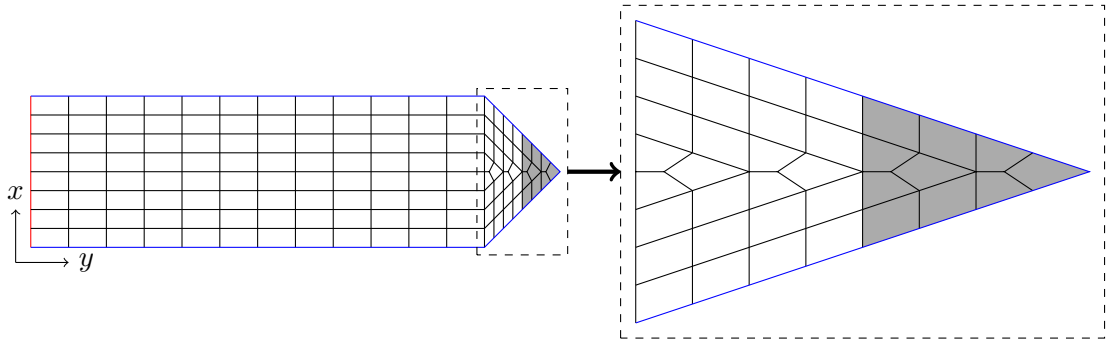


Figure 5.1: An example of the mesh used. A semi-regular structured mesh is used in the triangular contraction, making use of only quadrilateral element shapes. The linearised/rest-state waterline $y = L_b$ lies along one of the straight lateral (x-direction) grid lines. To the right of this line (gray) is where the wetted part of the buoy lies, and is the subdomain over which the λ equation is solved.

The Cartesian grid contains N_x and N_y elements in the x and y directions, respectively, with corresponding grid resolutions of $dx = L_x/N_x$ and $dy = (L_y - L_c)/N_y$. For the contraction region, the N_x rows of the Cartesian grid morph into N_x "columns", with one fewer element appearing in each column due to the kink formed between the two scalene quadrilaterals and the kite. This alternating pattern of scalene quadrilaterals and kites requires that N_x always be even.

Following the process seen in Section 4.4, the node coordinates are calculated, the wavemaker ($y = 0$) and waterline ($y = L_b$) boundaries as well as all grid lines outside of the subdomain are marked, and the Firedrake mesh object is created.

Dealing with the subdomain boundary $y = L_b$

Now, note that there is a straight line (with constant y) between every pair of columns in the contraction. To ensure that the mesh for the subdomain corresponding to the wetted part of the buoy conforms to the parent mesh, we require its boundary $y = L_b$ to lie on one of these lines. The original MATLAB code used a fixed width for these columns, meaning only parameter combinations yielding particular L_b values were able to be modelled. Instead, a more flexible mesh-creation process (in which these column-widths are able to vary) has been developed, enabling any L_b to be modelled. Essentially,

one of the constant-y grid lines is fixed at $y = L_b$, with the surrounding column-widths increased or decreased accordingly.

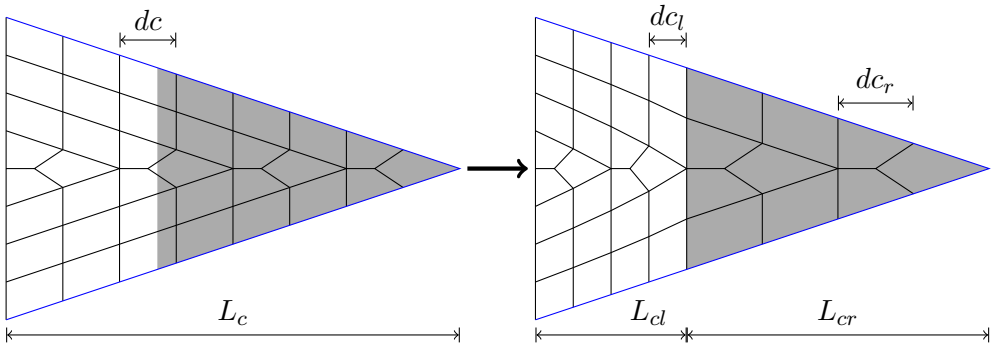


Figure 5.2: An example of how the flexible mesh allows any value of L_b to be modelled. On the left, each column is the same width dc, and the given L_b does not happen to lie on one of the constant-y grid lines. On the right, the second line has been shifted leftwards so that it lies on $y = L_b$, with the four columns to the left each having the same width dc_l, and the four on the right each having width dc_r, with $dc_l < dc_r$. This particular mesh is actually not the optimal choice: the ratio dc_r/dc_l is equal to 2, but if the first grid line was chosen (with two columns to the left and six to the right), the ratio would be 1.5.

This process, explained graphically in Figure 5.2, is described as follows. We first define left- and right-hand contraction lengths L_{cl} and L_{cr}, either side of L_b, i.e. $L_{cl} = L_b - (L_y - L_c)$ and $L_{cr} = L_y - L_b$. Now, for the given N_x columns, there are a possible $N_x/2-1$ choices for which line to place at $y = L_b$,[2] with each choice generating corresponding left- and right-hand column-counts N_l and N_r (i.e. for choice i, we have $N_l = 2i$ and $N_r = N_x - 2i$). We then ensure all columns on the same side of L_b have equal widths, producing corresponding left- and right-hand grid resolutions $dc_l = L_{cl}/N_l$ and $dc_r = L_{cr}/N_r$. Note that dc_l and dc_r are likely different; placing one of the lines at $y = L_b$ naturally requires some measure of skewness between the two resolutions. The grid line producing resolutions that are the closest (i.e. where $\max(dc_l, dc_r)/\min(dc_l, dc_r)$ is minimised) is the one that is chosen.

5.2.3 Firedrake objects

After building the mesh, we next create the Firedrake-specific objects central to the algorithm: `FunctionSpace`s on which to build the spatially-dependent `Function`s, and

[2]The line at the entrance to the contraction ($y = (L_y - L_c)$) is not considered, as it is assumed that $L_b > (L_y - L_c)$ (i.e. the waterline does not reach the edge of the contraction) and there is always some region in the contraction to the left of the waterline.

`Constant`s to store the spatially-invariant quantities.

First, the $\texttt{CGN}^{\text{th}}$-degree `FunctionSpace` and corresponding `TestFunction` (`V_CGN` and `v_CGN`, respectively) are created. This space will hold the `Function`s for Φ (`phi_old`, `phi_mid`, `phi_new`), η (`eta_old`, `eta_new`), λ (`lambda_old`, `lambda_new`) and H, recalling the latter's definition (2.11). By default, these quantities are initialised to be zero everywhere; should a nonzero initial condition be required, this can be declared using either `interpolate` (if an expression is used), or `assign` (if another `Function` holds the initial condition).

```
H = fd.Function(V_CGN).interpolate(
    fd.conditional(y < Lb, H0, Z0 + (Ly-y)*np.tan(alpha))
)
```

Next, a special `FunctionSpace` for F is defined. Despite being a spatially-invariant quantity, it needs to be treated like a `Function` to allow the solution of a UFL weak form. Thankfully, this is made possible in Firedrake by using the `'R'`, or *Real*, space. In this space, a `Function` is defined as having a constant value on every element of the mesh, and as-such the space always has a degree of 0; the `FunctionSpace` is therefore named `V_R0`. Note that this space has just a single basis function (which is simply equal to 1) and a single coefficient.

Then, recall how λ and F are to be solved as a coupled system. To facilitate this, we require a special "mixed" space, constructed by simply multiplying the relevant spaces together. Any `Function`s and `TestFunction`s defined on this space will have components corresponding to its constituent spaces, and can be accessed via indexing.

```
V_mixed = V_CGN * V_R0
v1, v2 = fd.TestFunctions(V_mixed) # v1 is the CGN TestFunction, v2 is R0

w_old = fd.Function(V_mixed)
lambda_old, F_old = fd.split(w_old)
w_new = fd.Function(V_mixed)
lambda_new, F_new = fd.split(w_new)
```

Note that `lambda_old` and `lambda_new` are defined via the mixed space `V_mixed`, rather than directly from `fd.Function(V_CGN)`, because they need to keep hold of their internal references back to the mixed system. Due to the internal indexing of `split`, they are, however, still `Function`s defined on `V_CGN`.

Finally, the remaining spatially-invariant quantities $\dot{R}$, W, Q, Z and I are declared. Since these are not solved-for in the same way as F, simple Python floats are sufficient, which are used alongside `Constant`s (e.g. `Rcont`, `Icont`) to store the relevant values in the UFL forms during time-stepping.

5.2.4 UFL weak forms

We can now write the weak forms appearing in the discrete systems (5.3) and (5.4) in UFL, as follows:

```
if time_scheme == 1:
    F_phi = v_CGN*((phi_new-phi_old)/dt + g*eta_old + lambda_old)*fd.dx
    phi_full = fd.NonlinearVariationalSolver(
        fd.NonlinearVariationalProblem(F_phi, phi_new),
        solver_parameters={'pc_type':'cholesky'}
    )

    F_eta =
    (v_CGN*(eta_new-eta_old)/dt - H*dot(grad(v_CGN),grad(phi_new)))*dx
        - H0*v_CGN*Rcont*ds(wavemaker)
    eta_full = fd.NonlinearVariationalSolver(
        fd.NonlinearVariationalProblem(F_eta, eta_new),
        solver_parameters={'pc_type':'cholesky'}
    )

    if buoy:
        F_lambda = (H*fd.dot(fd.grad(v1),fd.grad(lambda_new + g*eta_new))
                    + v1/M*(F_new - gamma*GZ0*Icont))*fd.dx
        F_F = v2*(F_new/area - rho0*lambda_new)*fd.dx

        lambda_full = fd.NonlinearVariationalSolver(
            fd.NonlinearVariationalProblem(
                F_lambda + F_F, w_new,
                bcs=[
                    fd.DirichletBC(
                        V_mixed.sub(0), 0, left_of_waterline
                    ),
                    fd.DirichletBC(
                        V_mixed.sub(0), g*(eta_new - Zcont), waterline
                    )
                ]
            ),
            solver_parameters=solver_parameters
        )

elif time_scheme == 2:
    F_phi_half =
        v_CGN*((phi_mid-phi_old)/(dt/2) + g*eta_old + lambda_old)*fd.dx
    phi_half = fd.NonlinearVariationalSolver(
        fd.NonlinearVariationalProblem(F_phi_half, phi_mid),
        solver_parameters={'pc_type':'cholesky'}
    )

    F_eta =
    (v_CGN*(eta_new-eta_old)/dt - H*dot(grad(v_CGN),grad(phi_mid)))*dx
```

```
        - H0*v_CGN*Rcont*ds(wavemaker)
    eta_full = fd.NonlinearVariationalSolver(
        fd.NonlinearVariationalProblem(F_eta, eta_new),
        solver_parameters={'pc_type':'cholesky'}
    )

    if buoy:
        F_lambda = (H*fd.dot(fd.grad(v1),fd.grad(lambda_new + g*eta_new))
                    + v1/M*(F_new - gamma*GZ0*Icont))*fd.dx
        F_F = v2*(F_new/area - rho0*lambda_new)*fd.dx

        lambda_full = fd.NonlinearVariationalSolver(
            fd.NonlinearVariationalProblem(
                F_lambda + F_F, w_new,
                bcs=[
                    fd.DirichletBC(
                        V_mixed.sub(0), 0, left_of_waterline
                    ),
                    fd.DirichletBC(
                        V_mixed.sub(0), g*(eta_new - Zcont), waterline
                    )
                ]
            ),
            solver_parameters=solver_parameters
        )

    F_phi_full =
        v_CGN*((phi_new-phi_mid)/(dt/2) + g*eta_new + lambda_new)*fd.dx
    phi_full = fd.NonlinearVariationalSolver(
        fd.NonlinearVariationalProblem(F_phi_full, phi_new),
        solver_parameters={'pc_type':'cholesky'}
    )
```

The above code makes use of some not-yet-seen objects: `wavemaker`, `left_of_waterline` and `waterline` are simply names of objects holding the boundary IDs `0`, `(0,1)` and `2`, respectively, while `V_mixed.sub(0)` extracts the first subspace of the mixed space (`V_CGN`, on which `lambda_new` is defined), ensuring the `DirichletBC` is applied to the correct unknown within the mixed system. Additionally, the nonlinear versions of `VariationalProblem/Solver` and the key-word argument `solver_parameters` have been used; see the following section discussing the former, and Section 5.3 for a detailed discussion on the latter.

Using `NonlinearVariationalProblem/Solver`

In Section 4.4, we utilised `LinearVariationalProblem/Solver` to define a problem with a bilinear LHS `a` and a linear RHS `L`, using a `TrialFunction u` in the definition of `a` and a `Function u_num` as an argument to `LinearVariationalProblem`; that is, the unknown being solved for was the `TrialFunction`, with the solution then being stored

in the `Function`. However, despite the equations still being linear, here we have utilised `NonlinearVariationalProblem/Solver`.

In matrix form, whereas linear problems are typically written as $\mathbf{Ax} = \mathbf{b}$, nonlinear problems are typically written as $\mathbf{f}(\mathbf{x}) = 0$; therefore, an equivalent "nonlinear" representation of a linear problem is simply

$$\mathbf{f} = \mathbf{Ax} - \mathbf{b} = 0,$$

with solution

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}.$$

When solving a nonlinear problem numerically, Firedrake (via PETSc) uses Newton's method, which in multidimensional form is

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J_f}(\mathbf{x}_n)^{-1}\mathbf{f}(\mathbf{x}_n), \tag{5.5}$$

where $\mathbf{x}_n$ is an initial guess (or the result of a previous iteration) and $\mathbf{J_f}$ is the Jacobian matrix, defined as

$$(J_f)_{ij} = \frac{\partial f_i}{\partial x_j}.$$

Now, in component form, $\mathbf{f}$ is

$$f_i = \sum_k A_{ik}x_k - b_i,$$

and differentiation with respect to x_j gives

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\sum_k A_{ik}x_k\right) - \frac{\partial b_i}{\partial x_j}.$$

Note that the first term on the RHS is only nonzero for $k = j$, while the second term is always zero since b_i is constant, and therefore we have simply

$$\frac{\partial f_i}{\partial x_j} = A_{ij};$$

or, in matrix form, $\mathbf{J_f} = \mathbf{A}$. Then, (5.5) gives

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{A}^{-1}\left(\mathbf{A}\mathbf{x}_n - \mathbf{b}\right) = \mathbf{A}^{-1}\mathbf{b},$$

which is to say that, for a linear problem, Newton's method finds an exact solution after just one iteration, regardless of the starting guess. Therefore, we can use the `NonlinearVariationalProblem/Solver` functions for linear problems without issue. Indeed, under-the-hood, a `LinearVariationalProblem/Solver` is actually a nonlinear equivalent with an additional flag telling PETSc not to bother checking for convergence or carrying out any further Newton iterations after the first one.

For a `NonlinearVariationalProblem`, we no longer use a `TrialFunction`, and instead use the `Function` being solved for directly in the equation definition. We can also write other (known) `Functions` directly into the equation's LHS, without having to separate these into the negative RHS, as a `LinearVariationalProblem` would require.

The main reason for setting up the problems in this way is for programmatic convenience. Most obviously, `TrialFunctions` are not required, and the UFL weak forms can be coded precisely as presented in (5.3) and (5.4), without any rearrangement. Additionally, should future research require the solution of truly nonlinear equations, the current algorithm can be expanded by simply re-writing the UFL weak forms, without the need to change all `LinearVariationalProblems/Solvers` into nonlinear ones and replace multiple `TrialFunctions` with `Functions`.

5.2.5 Data output initialisation

Before commencing time-stepping, the final preparation stage is to initialise the outputs of the simulation. First, we deal with a subtlety of how mixed spaces work in Firedrake. For non-mixed spaces, a `Function` is both a symbolic expression passed to UFL and a basis-function-coefficient storage container, meaning we can retain the `Functions` used in the UFL forms (`phi_old`, `eta_new` etc.) for use in calculating outputs. However, mixed spaces work subtly differently.

While a `Function` defined on a mixed space (`w_old` and `w_new` in our case) acts as

a container for coefficient-storage, it is not suitable for UFL and instead requires the use of `split` to extract the relevant symbolic objects. These, however, cannot themselves act as coefficient-containers for the `Function`s they represent; this is instead achieved through `subfunctions`. Now, because the UFL forms using the symbolic objects (which we named `lambda_old`, `F_old` etc.) have already been stored in the various `NonlinearVariationalProblem`s, those individual objects are no longer required, and we can re-define them to store the components of `subfunctions`. We can then use these new container objects in the same way as the non-mixed `Function`s for calculating outputs.

```
lambda_old, F_old = w_old.subfunctions
lambda_new, F_new = w_new.subfunctions
```

Then, we initialise a list of dictionaries `data`, storing relevant calculated outputs (such as energy, ODE solutions and power-output) at each time-step. Additionally, depending on the parameters `gif_save` and `pvd_save`, we optionally initialise a second dictionary to store the values of η and λ along the tank's centreline $x = L_x/2$ (which can later be used to create a `.gif` file), and/or initialise a `.pvd` file which can be used to view the spatially varying fields ϕ, η and λ in ParaView.

Finally, two fixed images are generated; a plot of $H(y)$ for the given parameters, and a drawing of the mesh.

5.2.6 Time-stepping

We now come to the execution of the time-stepping equation solutions. At each time-step, given each of the previous values `(.)_old`, we can execute the `solve` method of each of the `NonlinearVariationalSolver`s and calculate the updated ODE values, in the order presented in (5.3) and (5.4). We additionally make use of `assign` to update spatially-invariant `Constant`s, while the current value of F can be extracted from the corresponding `R`-space `Function` simply by passing the object to the built-in `float` function. Incrementing `t` by `dt` each time-step, we continue in a `while` loop so-long as `t <= T`, stopping at the first integer multiple of `dt` greater than `T`.

```
while t <= T:
    step += 1

```

```
    # solve equations
    if time_scheme == 1:
        phi_full.solve()

        if buoy:
            W = W + dt/M*(float(F_old) - gamma*GZ0*I)
            Wcont.assign(W)

            Q = Q + dt*I

        t += dt
        Rcont.assign(Rdot(t))
        eta_full.solve()

        if buoy:
            Z = Z + dt*W
            Zcont.assign(Z)

            I = (Li*I + dt*(gamma*GZ0*W - I/2*(Ri+Rc+Rl) - Q/C))/(Li +
                dt/2*(Ri+Rc+Rl))
            Icont.assign(I)

            lambda_full.solve()

    elif time_scheme == 2:
        phi_half.solve()

        if buoy:
            W = W + dt/(2*M)*(float(F_old) - gamma*GZ0*I)
            Wcont.assign(W)

            Q = Q + dt*I/2

        t += dt/2
        Rcont.assign(Rdot(t))
        eta_full.solve()

        if buoy:
            Z = Z + dt*W
            Zcont.assign(Z)

            I = (Li*I + dt*(gamma*GZ0*W - I/2*(Ri+Rc+Rl) - Q/C))/(Li +
                dt/2*(Ri+Rc+Rl))
            Icont.assign(I)

            lambda_full.solve()

        t += dt/2
        phi_full.solve()

        if buoy:
            W = W + dt/(2*M)*(float(F_new) - gamma*GZ0*I)
            Wcont.assign(W)

            Q = Q + dt*I/2
```

Then, before progressing to the next time-step, various outputs are calculated and stored in the dictionaries initialised previously. In particular, we calculate and store:

- the current time t;
- the current values for F, W, Z, Q and I;
- the energy of the hydrodynamic sub-system, given in (4.2):

$$E_w = \frac{\rho_0}{2} \int_0^{L_x} \int_0^{l_y} \left\{ H \|\nabla \phi\|^2 + g\eta^2 \right\} \mathrm{d}y \, \mathrm{d}x;$$

- the energy of the buoy, given in (4.2) and using $\dot{Z} = W$:

$$E_b = \frac{MW^2}{2};^{3}$$

- the energy of the generator, given in (4.2) and using $\dot{Q} = I$:

$$E_i = \frac{L_i I^2}{2} + \frac{Q^2}{2C};$$

- the total energy of the system:

$$E = E_w + E_b + E_i;$$

- the power generated (harvested by the LED), given by the integrand of (2.23), where $V = IR_l$ after linearising:

$$P_g = I^2 R_l; \text{ and}$$

- the power lost (dissipated by the circuit resistances), given by the integrand of (2.24)

$$P_l = I^2 (R_i + R_c).$$

Furthermore, at the intervals governed by the `gif_save` and `pvd_save` parameters, we calculate and store the centerline values for η and λ, and write the solutions for ϕ, η and λ to the `.pvd` file.

[3] Recall potential energy MgZ is not included in the linear limit.

Finally, using `assign`, the `(.)_old Functions` are updated with the `(.)_new` values, ready for the next time-step. As is the case for the `Constants`, because these `Functions` have previously been stored in the `Solvers`, these updates are automatically taken into account when the next iteration executes the `solve` methods.

5.2.7 Data extraction

After time-stepping has commenced, we plot and save the sparsity patterns for the problem (such as those presented in Figure 5.3). We also save the dictionaries storing the calculated time-dependent quantities at each time-step and (if generated) the `.gif` data. These dictionaries are saved as text files in `.json` format; this is a popular human-readable and language-independent data format, which can be intuitively understood by a viewer and can be parsed by many different applications [102].

Additionally, the main `data` object is converted into a `pandas` DataFrame, from which time-series of individual quantities can be readily accessed as individual columns. Post-processing of this data is left to the user. Finally, using the power output `'Pg'` and time `'t'` columns, an estimate of the total energy generated over the course of the simulation is obtained, using the trapezium rule.

```
return np.trapezoid(data['Pg'], data['t'])
```

5.3 PETSc solver parameters

As mentioned previously, after assembling the required matrices and vectors, Firedrake leverages PETSc to obtain the solutions. For an overview of the concepts involved in solving linear and nonlinear systems using both direct and iterative methods (on which PETSc is built), one may refer to Quarteroni et al. [89]. The user does not interact with PETSc directly, as Firedrake caries out all of the heavy lifting internally; however, through the use of the `solver_parameters` key-word argument of `NonlinearVariationalSolver`, some of the settings that Firedrake passes to PETSc can be chosen by.

A standard algorithm for solving a matrix equation using Newton's method (5.5) is:

- assemble the vector $\mathbf{f}(\mathbf{x}_n)$;
- assemble the Jacobian $\mathbf{J}_\mathbf{f}(\mathbf{x}_n)$;
- solve the linear step $\mathbf{J}_\mathbf{f}(\mathbf{x}_n)\Delta\mathbf{x} = -\mathbf{f}(\mathbf{x}_n)$, either:
  - directly (using lower-upper (LU) or Cholesky decomposition), or
  - iteratively (using one of a number of Krylov-subspace methods);
    - optionally, apply a preconditioner during each of the Krylov iterations, to ease convergence;
- update the solution $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}$;
  - optionally, include a line-search or damping parameter $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha\Delta\mathbf{x}$ with $0 < \alpha \leq 1$, to ease convergence;[4]
- continue until convergence is achieved.

Each of these steps can be controlled using `solver_parameters`. The default parameters for a `NonlinearVariationalSolver` are:

- `'mat_type'='aij'`: this tells PETSc to store the matrix in compressed sparse row format.
- `'ksp_type'='preonly'`: this says do not carry out any Krylov iterations, effectively turning the linear solve into a direct solve using the preconditioner (if given).
- `'ksp_rtol'=1e-5`: the relative tolerance for Krylov-iteration convergence; this is not actually used if `'ksp_type'='preonly'`, since no iterations are carried out.
- `'pc_type'='lu'`: use LU decomposition as the preconditioner; used alongside `'ksp_type'='preonly'`, this effectively solves the linear step directly using LU.
- `'pc_factor_mat_solver_type'='mumps'`: this parameter tells PETSc to leverage the external MUMPS package [103] to carry out the LU decomposition.
- `'pc_factor_mat_mumps_icntl_14'=200`: this is a parameter that PETSc passed to MUMPS, telling it to allocate 200 % more memory than it otherwise would;

[4] For the nonlinear representation of the linear system described previously to remain valid, α must be 1.

this is because sparse FE matrices are susceptible to significant fill-in during the decomposition process [104], which may otherwise cause out-of-memory errors.

- `'snes_type'='newtonls'`: this tells PETSc to update the solution $\mathbf{x}_{n+1}$ using a line-search method with some α.
- `'snes_linesearch_type'='basic'`: this means do not actually implement a line-search, effectively setting $\alpha = 1$.

These default settings have been chosen because they are good at handling the vast majority of problems. However, for problems meeting certain criteria, greater efficiency may be obtained by particular changes to these settings. For example, Cholesky decomposition is known to be approximately half as computationally expensive as LU, but is only applicable to symmetric-positive-definite (SPD) matrices [89, p. 82]. Thankfully, the mass matrix appearing in the ϕ and η weak forms is always SPD,[5] and we can overwrite the default parameters with `'pc_type'='cholesky'`.

5.3.1 The `'fieldsplit'` solver parameter

While the solution approach for the ϕ and η equations is quite straightforward, special attention is required for the mixed system combining λ and F. Let us write the system at any given time-step in its discretised form (i.e. (4.10) with the removal of the arbitrary $\hat{v}_i$), with the known terms η and I taken into the RHS as b_i:

$$\sum_{j=1}^{N_n} \left(\iiint_\Omega H \nabla\psi_i \cdot \nabla\psi_j \,\mathrm{d}V \right) \hat{\lambda}_j + \left(\iiint_\Omega \frac{\psi_i}{M} \,\mathrm{d}V \right) \hat{F} = b_i, \tag{5.6a}$$

$$\left(\iiint_\Omega \frac{1}{A} \,\mathrm{d}V \right) \hat{F} - \sum_{j=1}^{N_n} \left(\rho_0 \iiint_\Omega \psi_j \,\mathrm{d}V \right) \hat{\lambda}_j = 0. \tag{5.6b}$$

Note we have used the fact that the (single) basis function ψ_1 in the F equation is simply equal to 1, with $\hat{F}$ the corresponding coefficient, i.e.

$$F(t) = \sum_{i=1}^{1} \hat{F}_i(t)\psi_i(\mathbf{x}) = \hat{F}_1(t) \cdot 1 \equiv \hat{F}(t).$$

[5] The positive-definiteness of the mass matrix $M_{ij} = \iiint_\Omega \psi_i\psi_j \,\mathrm{d}V$ can be proven by considering the expression $\mathbf{v}^T\mathbf{M}\mathbf{v}$ for some vector $\mathbf{v}$, which becomes $\iiint_\Omega (v_i\psi_i)^2 \,\mathrm{d}V$; this is positive for any nonzero $\mathbf{v}$, and thus $\mathbf{M}$ is positive-definite.

Writing $\hat{\lambda}_j$ as the vector $\boldsymbol{\lambda}$ and b_i as $\mathbf{b}$, the system (5.6) can be written in the following block-matrix form:

$$\begin{pmatrix} \boldsymbol{A_{00}} & \boldsymbol{A_{01}} \\ \boldsymbol{A_{10}} & A_{11} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \hat{F} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix},$$

where

$$A_{00,ij} = \iiint_\Omega H \nabla\psi_i \cdot \nabla\psi_j \, \mathrm{d}V$$

is the standard FE stiffness matrix (augmented with the spatially-dependent H),

$$A_{01,i} = \iiint_\Omega \frac{\psi_i}{M} \, \mathrm{d}V \quad \text{and} \quad A_{10,j} = -\rho_0 \iiint_\Omega \psi_j \, \mathrm{d}V$$

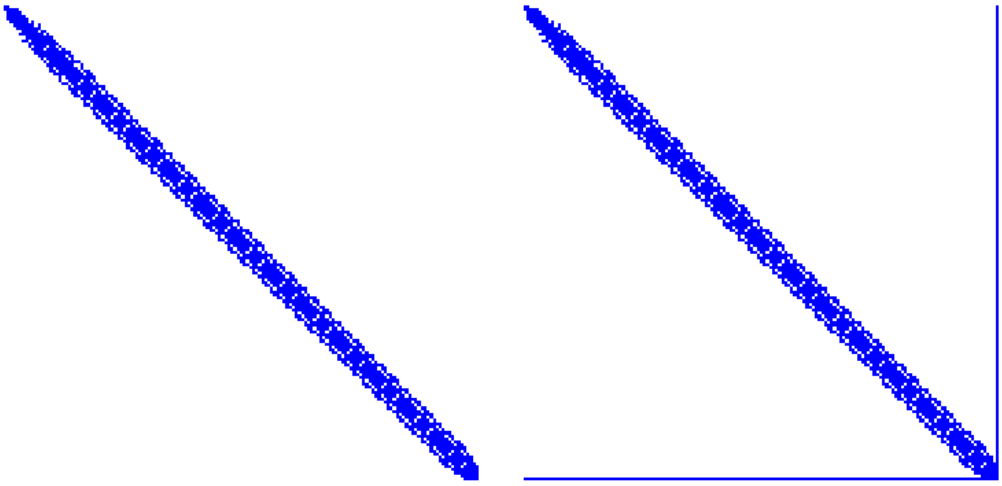
are vectors describing the coupling between λ and F, and

$$A_{11} = \iiint_\Omega \frac{1}{A} \, \mathrm{d}V = 1.$$

Note that the vectors $\boldsymbol{A_{01}}$ and $\boldsymbol{A_{10}}$ are dense, since they each combine either local basis functions or coefficients with a global coefficient or basis function; the lack of local support for the Real space causes this lack of sparsity. On the other hand, the stiffness matrix $\boldsymbol{A_{00}}$ is sparse, while A_{11} is simply equal to 1. See Figure 5.3 for the sparsity pattern of this block matrix, compared to the single mass matrix arising in the ϕ and η equations; the dense row and column created by these vectors are evident.

Now, if one were to solve for the components λ or F independently of one another, the dense off-diagonal blocks could be moved to the RHS, and standard sparse solution methods could be used (with F actually being entirely trivial to solve). However, to solve for both λ and F simultaneously, the contributions of $\boldsymbol{A_{01}}$ and $\boldsymbol{A_{10}}$ mean the combined matrix can no longer be treated as sparse, despite the significant emptiness between the stiffness matrix's diagonal and the dense vectors, and therefore any direct solution of the full coupled system will be much more expensive.

Thankfully, as `solver_parameters` can be utilised to tell PETSc to execute an iterative solve, we can approximate the solution by solving for the two fields somewhat independently, thus utilising the sparse blocks. This is achieved with `'pc_type'='fieldsplit'`,



(a) The mass matrix, used for ϕ and η. This is always SPD.

(b) The matrix used in the mixed system for λ and F; in particular, note the dense row and column arising from the use of the Real space.

Figure 5.3: Sparsity patterns for the matrices appearing in the problem.

and `'pc_fieldsplit_type'='multiplicative'`, which essentially applies the block-Gauss–Seidel method to the system. That is, (5.6) is approximated as

$$\sum_{j=1}^{N_n} \left(\iiint_\Omega H \nabla\psi_i \cdot \nabla\psi_j \,\mathrm{d}V \right) \hat{\lambda}_j^{new} + \left(\iiint_\Omega \frac{\psi_i}{M} \,\mathrm{d}V \right) \hat{F}^{old} = b_i,$$

$$\left(\iiint_\Omega \frac{1}{A} \,\mathrm{d}V \right) \hat{F}^{new} - \sum_{j=1}^{N_n} \left(\rho_0 \iiint_\Omega \psi_j \,\mathrm{d}V \right) \hat{\lambda}_j^{new} = 0,$$

which is to say that λ is first solved using the previous value of F, and then F is subsequently solved using the new value of λ. Other fieldsplit options include `'additive'`, in which both λ and F are solved using the previous iteration's value (this is essentially block-Jacobi), and `'schur'`, which applies a Schur-complement approach to effectively eliminate one of the components from the solve entirely. Of the various fieldsplit methods, `'multiplicative'` is a good compromise between computational expense and convergence speed.

Furthermore, since each component is to be solved individually, each can have its own `'ksp_type'` and `'pc_type'` parameters. Both will use `'ksp_type'='preonly'`, with the λ equation (whose stiffness matrix is not guaranteed to be SPD) employing `'pc_type'='lu'` while the trivial solve for F has `'pc_type'='none'`. These are defined

in a nested way, using the fieldsplit indices 0 and 1 corresponding to their order within the mixed system.

5.3.2 The GMRES method

The solution of any matrix equation can be though of as beginning at an estimated solution $\mathbf{x}_0$, "travelling" a certain distance in a certain direction, and hoping to stop at the true solution $\mathbf{x}$. Mathematically this can be written as

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{P}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0), \tag{5.7}$$

where $\mathbf{P}$ is a matrix representation of the particular solution method used, which in-effect calculates the distance and direction of travel, based off of the problem statement and the current position. When $\mathbf{P} = \mathbf{A}$, the true solution is found in just one step and the solve is direct; iterative methods are required when $\mathbf{P} \neq \mathbf{A}$, where the hope is each step brings the traveller closer to the true solution.

For the block-Gauss–Seidel method invoked by the multiplicative fieldsplit, we have

$$\mathbf{P} = \begin{pmatrix} \boldsymbol{A_{00}} & 0 \\ \boldsymbol{A_{10}} & A_{11} \end{pmatrix} \neq \mathbf{A} = \begin{pmatrix} \boldsymbol{A_{00}} & \boldsymbol{A_{01}} \\ \boldsymbol{A_{10}} & A_{11} \end{pmatrix},$$

meaning iterations are required; the default `'ksp_type'='preonly'`, which carries out just a single iteration of (5.7), is not sufficient. A crude implementation of the iterative method consists of simply repeating (5.7) until the solution is reached, and can be implemented with `'ksp_type'='richardson'`; however, depending on the precise form of $\mathbf{P}$, this approach can be incredibly slow.

Convergence can be accelerated by applying some scaling parameter α to (5.7), giving

$$\mathbf{x} = \mathbf{x}_0 + \alpha\mathbf{P}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0).$$

This is like starting at $\mathbf{x}_0$ and travelling in the direction that $\mathbf{P}$ dictates, but modifying the distance travelled. The `'ksp_richardson_scale'` parameter can be passed to set a constant value of α, for all iterations; however, this will usually not converge unless

α is very small, and would normally be used to fine-tune the method if it is known the starting guess is close to the true solution.

More elegant approaches exist, however, which set α dynamically at each iteration. One such method is the *generalized minimal residual* (GMRES) method, which is a good general-purpose method for asymmetric matrices.[6] Referring to Section 4.4.2 of Quarteroni et al. [89, p. 167] for a detailed discussion, in summary GMRES accelerates convergence by calculating a new, more optimal α at each iteration, using information collected at previous iterations. This is like, after calculating the direction of travel for the current iteration, looking backwards at the path previously taken, identifying the pattern and modifying the distance to be travelled in this iteration to optimise the journey towards the true solution. The combination of the block-Gauss–Seidel preconditioner finding a good search direction and the GMRES Krylov method fine-tuning the travel distance results in a significant increase in convergence speed.

Thus, the final set of `'solver_parameters'` for the mixed solver is:

```
solver_parameters = {
    'mat_type':'nest', # required for fieldsplit
    'ksp_type':'gmres',
    'ksp_gmres_restart': 100,              # these control specific
    'ksp_gmres_modifiedgramschmidt': True, # GMRES parameters
    'pc_type':'fieldsplit',
    'pc_fieldsplit_type':'multiplicative',
    'fieldsplit_0': {
        'ksp_type':'preonly',
        'pc_type':'lu'
    },
    'fieldsplit_1': {
        'ksp_type':'preonly',
        'pc_type':'none'
    }
}
```

5.4 Comparing the cost of the Firedrake and MATLAB codes

We assess the efficiency of the new algorithm, by comparing the run times of equivalent simulations using the existing MATLAB implementation and the new Firedrake code.

[6] Note that despite the symmetry of the stiffness block $\boldsymbol{A_{00}}$, the matrix for the entire mixed system is not symmetric because the off-diagonal blocks $\boldsymbol{A_{01}}$ and $\boldsymbol{A_{10}}$ differ.

All experiments were performed on a Lenovo 81N6 laptop running Microsoft Windows 11 Home (build 26200), equipped with an AMD Ryzen 5 3500U processor (4 cores, 8 threads, 2.1 GHz) and 6 GB of RAM.

Both implementations were executed using the parameter values given in Table 3.2, with a simulation time of 13.4 s. For each configuration, 30 runs were performed, of which the first five were discarded as warm-ups, leaving 25 timed executions. Each run consisted of three simulations, with grid resolutions $(N_x, N_y) = (10, 50), (10, 100)$ and $(20, 100)$. The MATLAB functions `tic` and `toc`, and the Python function `time.time()` were used to record the wall-clock time at the start and end of the simulations.

Table 5.1: Summary statistics of the run times for the two algorithms. The median and interquartile range for each grid resolution is recorded. The speed-up is the ratio between the medians.

(N_x, N_y)	MATLAB median (IQR)	Firedrake median (IQR)	Speed-up
$(10, 50)$	129.3 s (127.6–138.2 s)	26.5 s (26.3–26.9 s)	4.9x
$(10, 100)$	508.1 s (500.0–520.0 s)	39.8 s (38.9–40.5 s)	12.8x
$(20, 100)$	3,641.7 s (3,556.1–3,764.3 s)	93.1 s (90.0–100.3 s)	39.1x

The resulting run times are summarised in Table 5.1, detailing the median run time and interquartile range for the two algorithms, and the relative speed-up offered by Firedrake. The most obvious result is that Firedrake is significantly quicker: 5 times as quick for the lowest resolution, and 40 times faster for the highest resolution. Particularly eye-catching is the difference in slow-down between the two codes, presented in Table 5.2: Upon doubling N_x, MATLAB becomes almost 10 times slower, while Firedrake's slow-down is much less extreme.

Table 5.2: Slow-down of run time as the number of elements increases. Measured relative to the median of the lowest resolution.

(N_x, N_y)	MATLAB slow-down	Firedrake slow-down
$(10, 50)$	1.0x	1.0x
$(10, 100)$	3.9x	1.5x
$(20, 100)$	28.2x	3.5x

The reasons behind this large discrepancy in performance are manifold. To assemble an $N_{el} \times N_{el}$ FE matrix, the MATLAB code executes a 5-layer-nested loop over the N_{el} elements, two sets of 4 local nodes and two sets of 2 quadrature points, performing many

calculations in a linear manner. In a high-level interpreted language, an explicit loop such as this, where entries are calculated and added to the matrix one at a time, this is a huge overhead. In contrast, built on UFL and PETSc, Firedrake has been greatly optimised for matrix assembly: it constructs a symbolic UFL representation of the FE integrals, then passes them to PETSc, where they are compiled to efficient low-level C code which executes highly efficient vectorised calculations with pre-allocated memory. While MATLAB does offer functionality to optimise elements of the matrix assembly ourselves, the fact that this has already been developed for Firedrake internally is a huge upgrade.

Another bottleneck in the MATLAB code is in matrix solving. Recall that for an iterative time-stepping scheme, the LHS of the matrix equations remain unchanged at each time-step. Therefore, decompositions and preconditions can be obtained once, cached and reused with an updated forcing term in each iteration. The MATLAB implementation does not do any of this; it carries out the same manipulations to the same matrices at each time-step. In Firedrake, this is handled automatically via the `NonlinearVariationalProblem/Solver` objects; again, work could be done on the MATLAB code to optimise these solves directly, but Firedrake already does it for us.

Additionally, Firedrake makes use of iterative solves, which can obtain approximate matrix solutions far quicker than the direct solves employed in the MATLAB code. Given the FEM is already an approximation, carrying out an exact direct solve is not worth the increased computational cost.

Besides its obvious improvement in performance, using Firedrake also provides many other additional benefits. Python is free and open-source, whereas MATLAB requires a paid licence. The Firedrake team is well-funded across multiple institutions, and there is a large team producing regular updates and adding new features to the library. Among other things, Firedrake can:

- Carry out simulations in parallel.
- Perform point-evaluation at any physical point in the computational domain, without requiring the user to map the coordinate to a nodal index, or perform any

manual interpolation.

- Work with both ParaView and `matplotlib` to visualise `Funciton` objects directly, again without requiring direct interaction with the underlying data values.
- Generate flame graphs showing an interactive graphical representation of code efficiency, grouping related processes and functions and allowing the user to quickly highlight bottlenecks in their code.
- Save and load checkpointed simulation states, allowing for simulations to be paused and resumed, and for cancelled or crashed executions to be restarted without loss of progress.

Assuming these are even possible in MATLAB, they would require substantial manual effort to replicate.

The Firedrake implementation is clearly superior to the existing MATLAB code. Firedrake's optimisations make it orders-of-magnitude more efficient, significantly more scalable, and much more flexible. Its use also brings numerous additional workflow and productivity benefits, and a professional, experienced and helpful team of developers who are always on hand to help.

5.5 Using the algorithm to optimise the device

Given the significant increase in performance offered by the Firedrake algorithm, it can now be feasibly employed to carry-out surrogate-enabled optimisation studies of the device. For a set of p design parameters, lower and upper bounds for each are chosen, generating a p-dimensional hypercube defining the design space. There is then some point in this space which optimises a given objective function; in this case, maximising the total power output. The field of surrogate-enabled optimisation considers finding an estimate of such an optimal point.

First, a set of points providing good coverage of the design space are chosen. One could choose a regular grid layout in each dimension, which would cause the number of points to increase as dimension p increases; for an efficient model and small number of

dimensions, this is not necessarily a problem. For larger problems, where scaling the number of points with the dimension would be prohibitively expensive, methods such as Latin hypercube sampling exist [105], enabling good coverage of the domain using a number of sample points that does not depend upon dimension.

The algorithm is then executed at each point, yielding a collection of sample power-outputs covering the full design space. These samples are then used to inform the generation of a global approximate response surface, which estimates the power output across the entire design space, rather than a discrete set of points. Using a chosen optimisation strategy (e.g. stationary-point analysis or gradient-based methods if gradients of the response surface are available; or genetic algorithms, simulated annealing or particle-swarm optimisation if the response is multimodal), the maxima of the response surface, approximation the true maxima's position and value, can then be obtained. Such a surrogate-enabled optimisation approach is far cheaper than executing the entire algorithm for every possible design combination.

When carrying-out a full design optimisation, the approximate optimum can be refined, by focussing on the surrounding region, selecting a fresh set of points at a higher resolution, obtaining a fresh sample of simulation results and building a finer-tuned response surface. In the present context, however, in order to demonstrate the algorithm's flexibility and suitability for such a strategy, it suffices to simply carry-out initial first-stage parameter sweeps.

5.5.1 Setting up a parameter sweep

Using the `params` class provided, executing a parameter sweep is as simple as: looping over the n design-space points, updating the values of the class attributes associated with the p design parameters, running `main` with the updated class, and storing the resulting calculated energy.

```
import main

class params:
    ...
    # definitions of non-varying parameter values remain here
    ...

energies = []
```

```
for i in range(n):
    for j in range(p):
        params.parameter_j = value_ij
    energies.append(main.main(params))
```

Three parameter sweeps have been carried out: a single-dimensional sweep of contraction angle θ, a 2D sweep of θ and rest-state water depth H_0, and a second 2D sweep of θ and tank width L_x. All other parameters values remain the same as in Table 3.2

Contraction angle

One definition of the contraction angle θ is

$$\frac{\tan\theta}{2} = \frac{L_c}{L_x}.$$

We shall use this ratio to define the bounds on θ, choosing values between 0.5 (the width L_x being twice the length L_c) and 5 (the length being fives times the width) for a fixed width; these correspond to angles of 45–84.3°. The result for $n = 30$ equispaced angles between these bounds is shown in Figure 5.4.

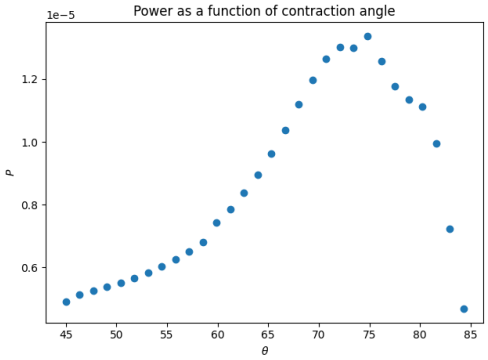


Figure 5.4: Total power output P as a function of contraction angle θ.

Contraction angle and water-depth

Using the same bounds on θ, a 2D sweep alongside H_0 was carried out, with depths of 0.075–0.15 m. A total of $n = 36$ points, using a 6×6 regular grid covering the design space, were sampled. A response surface using these samples was then generated, using Gaussian radial basis functions $\phi(r) = \exp\left(-(r\beta)^2\right)$ with $\beta = 2$ to interpolate between the points. See Figure 5.5 for these results.

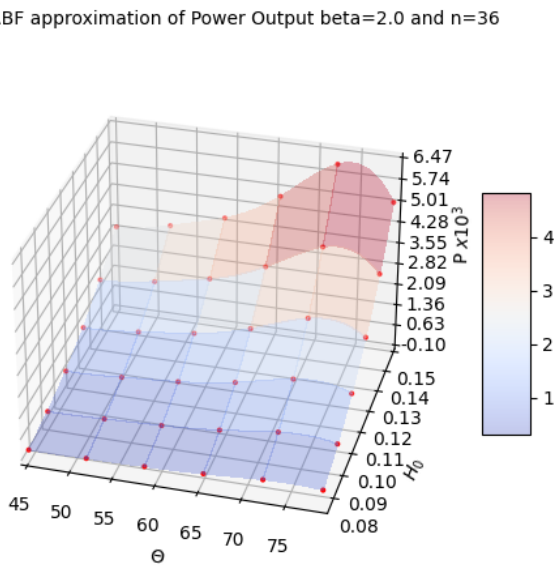


Figure 5.5: Total power output P as a function of contraction angle θ and water depth H_0.

Contraction angle and tank width

Finally, θ was combined with L_x to carry out a second 2D sweep, using widths of 0.2–0.55 m. The same basis functions were used to generate the response surface, shown in Figure 5.6.

5.5.2 Results and discussion

The theoretical dependence of the device's performance on contraction angle θ is not immediately clear; a smaller angle creates a shaper contraction, where one might expect a greater wave-amplification and therefore a greater power output, while a larger angle creates a longer contraction, giving the waves more time to be amplified and therefore also increasing performance. Therefore, one might expect an optimal angle where these two competing effects are balanced.

Such an optimal angle is evident in Figure 5.4, where, for this specific set of values used for the remaining parameters, an angle of around 75° is optimal. Figure 5.5 demonstrates this further, where for every value of H_0 the same optimal angle is obtained.

Furthermore, as the wavemaker motion was left unchanged, increasing H_0 causes a greater force to be applied to the the buoy (since a wavemaker displacing a greater

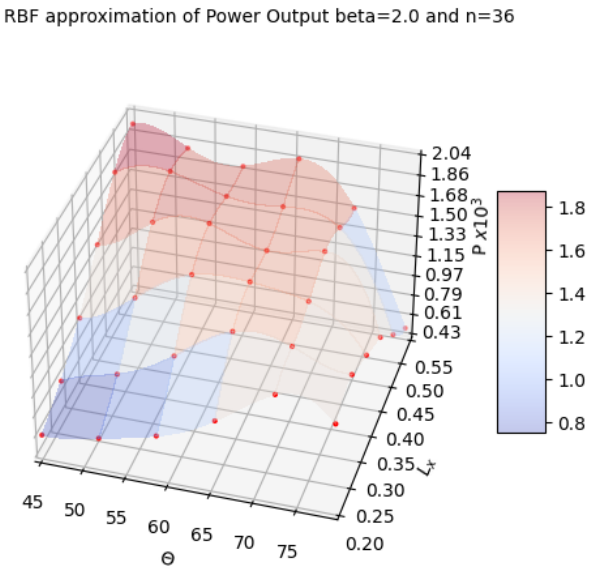


Figure 5.6: Energy generated as a function of contraction angle θ and tank width L_x.

amount of water by the same amount requires a larger force), and we would expect to see an increased performance; we see from Figure 5.5 that this is indeed the case.

Similarly, increasing L_x increases the amount of water contained in the device, and one may also expect to see the performance increase. However, changing L_x may affect how the angle impacts the performance, and for a given angle we may not see that increasing L_x guarantees an increased output. Indeed, Figure 5.6 shows that this trend is only the case close to $\theta \approx 45°$. The interaction between θ and L_x has a complex effect upon the performance, with this example suggesting an angle of 45° and a width of 0.55 m are optimal.

Clearly, further investigations covering a wider range of operating conditions are required to build a more accurate picture of the interplay between tank width and contraction angle. Similarly, combinations not considered here need to be tested, such as buoy mass, hull angle, and various electromagnetic parameters. However, through the use of the algorithm developed presently, such investigations can be much more readily carried out.

Chapter 6

Buoy-Generator Analysis

The SW system (4.3) solved by the algorithm presented in the previous chapter involves a combination of PDEs modelling the hydrodynamics and ODEs modelling the buoy and generator, with the bulk of the computational expense is spent on solving the spatially-dependent PDEs. However, the algorithm's key output of power generated during the simulation

$$P_g = I^2 R_l \tag{6.1}$$

depends entirely upon the value of I calculated from one of the ODEs. Therefore, improving the performance of the generator can go a long way to optimising the device as a whole.

By approximating the entire hydrodynamic sub-system of the device as a simple time-dependent forcing (i.e. the force $F(t)$ applied to the buoy by the water), the PDEs in the system can be discarded, and a coupled system of 4 ODEs governing the buoy and generator dynamics is obtained. That system is as follows:

$$M\dot{W} = -\gamma G(Z_0)I + F, \tag{6.2a}$$

$$\dot{Z} = W, \tag{6.2b}$$

$$\dot{Q} = I, \tag{6.2c}$$

$$L_i\dot{I} = \gamma G(Z_0)W - \frac{Q}{C} - I(R_i + R_c + R_l). \tag{6.2d}$$

This chapter considers the ODE system (6.2). Section 6.1 presents a frequency-domain analysis, where a sinusoidal forcing $F = \mathrm{Re}\{Ae^{i\omega t}\}$ and corresponding sinusoidal solutions are assumed, leading to the derivation of an explicit expression for the power generated by the device; the dependence of this expression upon the various parameters associated with the device is discussed. Section 6.2 demonstrates how the power output can be increased simply by splitting up the coil and rewiring individual sub-coils in parallel.

6.1 A frequency-domain analysis of the ODE system

Assuming a sinusoidal forcing F of the form

$$F = \mathrm{Re}\{Ae^{i\omega t}\} = A_r \cos\omega t - A_i \sin\omega t,$$

for some (complex) amplitude A and frequency ω, we assume (6.2) has corresponding solutions of the form

$$\begin{pmatrix} W \\ Z \\ Q \\ I \end{pmatrix} = \mathrm{Re}\left\{ \begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q} \\ \hat{I} \end{pmatrix} e^{i\omega t} \right\}.$$

Substituting these solutions into (6.2) and subsequently dividing the entire system by $e^{i\omega t}$ yields the following matrix representation of the system:

$$\begin{pmatrix} i\omega M & 0 & 0 & \gamma\mathcal{G} \\ -1 & i\omega & 0 & 0 \\ 0 & 0 & i\omega & -1 \\ -\gamma\mathcal{G} & 0 & \frac{1}{C} & i\omega L_i + \mathcal{R} \end{pmatrix} \begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q} \\ \hat{I} \end{pmatrix} = \begin{pmatrix} A \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{6.3}$$

with $\mathcal{G} \equiv G(Z_0)$ and $\mathcal{R} \equiv R_i + R_c + R_l$.

Now, as (6.3) contains a 4×4 matrix, solving the system by directly inverting the matrix is prohibitively difficult. However, by noting that the matrix can be written in block

form, the Schur complement can be employed to solve the system far more easily.[1] Referring to Appendix A.7.1 for details, we find

$$\begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q} \\ \hat{I} \end{pmatrix} = \begin{pmatrix} \dfrac{i(\omega^2 L_i - i\omega\mathcal{R} - 1/C)A}{\omega(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \\ \dfrac{(\omega^2 L_i - i\omega\mathcal{R} - 1/C)A}{\omega^2(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \\ \dfrac{-i\gamma\mathcal{G}A}{\omega(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \\ \dfrac{\gamma\mathcal{G}A}{\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M} \end{pmatrix}. \tag{6.4}$$

Now, the fully time-dependent solutions to the system (6.2) can be found by taking the real and imaginary parts of (6.4) separately, and multiplying by $\cos\omega t$ and $-\sin\omega t$ respectively, i.e.

$$(\cdot) = \mathrm{Re}\{\hat{(\cdot)}\}\cos\omega t - \mathrm{Im}\{\hat{(\cdot)}\}\sin\omega t.[2] \tag{6.5}$$

6.1.1 An explicit expression for the power output

An explicit expression for the power output $P_g(t)$ can be found, by combining (6.1), (6.4) and (6.5). Then, one can find the average power output $\hat{P}_g$ over a period of time $t \in [0, T]$ simply by integrating that expression and dividing by T. Assuming, then, that T is some integer multiple of the half-period π/ω, this integral becomes trivial.

First, we find that

$$P_g = \left(\mathrm{Re}\{\hat{I}\}^2\cos^2\omega t - 2\,\mathrm{Re}\{\hat{I}\}\,\mathrm{Im}\{\hat{I}\}\cos\omega t\sin\omega t + \mathrm{Im}\{\hat{I}\}^2\sin^2\omega t\right)R_l. \tag{6.6}$$

Now, for $T = n\pi/\omega$, the following identities hold:

$$\frac{1}{T}\int_0^T \cos^2\omega t\,\mathrm{d}t = \frac{1}{T}\int_0^T \sin^2\omega t\,\mathrm{d}t = \frac{1}{2} \quad \text{and} \quad \frac{1}{T}\int_0^T \cos\omega t\sin\omega t\,\mathrm{d}t = 0.$$

[1] For an explanation of how the Schur complement enables an easier calculation of the solution to block matrices, see Zhang [106]. In essence, it involves performing a block Gaussian elimination to eliminate one of the blocks.

[2] Note that $\mathrm{Re}\{\hat{(\cdot)}\}$ and $\mathrm{Im}\{\hat{(\cdot)}\}$ themselves are time-independent coefficients.

Therefore, after integrating (6.6) and recalling that $\mathrm{Re}\{\hat{(\cdot)}\}$ and $\mathrm{Im}\{\hat{(\cdot)}\}$ are constant in time, we have

$$\hat{P}_g = \frac{R_l}{2}\left(\mathrm{Re}\{\hat{I}\}^2 + \mathrm{Im}\{\hat{I}\}^2\right). \tag{6.7}$$

Next, writing $\gamma^2\mathcal{G}^2 - \omega^2 L_i M + {}^M\!/_C = \alpha$ and $\omega\mathcal{R}M = \beta$, from the ODE solutions (6.4) we have

$$\begin{aligned}\hat{I} &= \frac{\gamma\mathcal{G}A}{\alpha+i\beta} = \frac{\gamma\mathcal{G}(A_r+iA_i)(\alpha-i\beta)}{\alpha^2+\beta^2} \\ \Rightarrow \mathrm{Re}\{\hat{I}\} &= \frac{\gamma\mathcal{G}(A_r\alpha+A_i\beta)}{\alpha^2+\beta^2} \quad \text{and} \quad \mathrm{Im}\{\hat{I}\} = \frac{\gamma\mathcal{G}(A_i\alpha-A_r\beta)}{\alpha^2+\beta^2},\end{aligned}$$

such that (6.7) becomes

$$\begin{aligned}\hat{P}_g &= \frac{R_l(\gamma\mathcal{G})^2}{2\left(\alpha^2+\beta^2\right)^2}\left((A_r\alpha+A_i\beta)^2+(A_i\alpha-A_r\beta)^2\right) \\ &= \frac{R_l(\gamma\mathcal{G})^2}{2\left(\alpha^2+\beta^2\right)^2}\left((A_r\alpha)^2+(A_i\beta)^2+(A_i\alpha)^2+(A_r\beta)^2\right) \\ &= \frac{R_l(\gamma\mathcal{G})^2}{2\left(\alpha^2+\beta^2\right)^2}\left(A_r^2+A_i^2\right)\left(\alpha^2+\beta^2\right) \\ &= \frac{R_l(\gamma\mathcal{G})^2\left(A_r^2+A_i^2\right)}{2\left(\alpha^2+\beta^2\right)}.\end{aligned}$$

Finally, replacing α and β, and using the fact that $A_r^2 + A_i^2 = |A|^2$ (since $A = A_r + iA_i$), the final, explicit expression for the average power output, over any integer number of half-periods of the forcing frequency, is

$$\hat{P}_g = \frac{R_l(\gamma\mathcal{G}|A|)^2}{2\left(\left(\gamma^2\mathcal{G}^2 - \omega^2 L_i M + {}^M\!/_C\right)^2 + (\omega\mathcal{R}M)^2\right)}. \tag{6.8}$$

Clearly, the power is greater when the forcing amplitude $|A|$ is increased.[3] However, the dependence upon the remaining parameters is not as clear-cut.

6.1.2 Resonance and optimal power output

The expression (6.8) is maximised when its denominator is minimised. The precise value of ω which minimises the denominator is known as the *resonance frequency*; this

[3] In fact, a more sensible measure would be $\hat{P}_g/|A|^2$.

frequency depends heavily upon the values of the various parameters present. One particular parameter is the capacitance C, which only appears once; in the M/C term. We shall consider both the impact of including the capacitor, and the scenario in which no capacitor is modelled (mathematically, this is when $C \to \infty$).

When there is no capacitor, (6.8) becomes, after division by $|A|^2$,

$$\frac{\hat{P}_g}{|A|^2} = \frac{R_l(\gamma\mathcal{G})^2}{2\left((\gamma^2\mathcal{G}^2 - \omega^2 L_i M)^2 + (\omega\mathcal{R}M)^2\right)},$$

which, as a function of ω, is maximised when its derivative with respect to ω vanishes; or, in other words, the resonance frequency is

$$\omega = \sqrt{\frac{\gamma^2\mathcal{G}^2}{L_i M} - \frac{\mathcal{R}^2}{2L_i^2}}\,. \tag{6.9}$$

At this frequency, the power output is

$$\frac{\hat{P}_g}{|A|^2} = \frac{R_l L_i}{2\mathcal{R}^2 M\left(1 - \dfrac{\mathcal{R}^2 M}{4L_i\gamma^2\mathcal{G}^2}\right)}.$$

We can now identify how the power output responds to changes in the remaining parameters, assuming the system is operated at its resonance frequency for each parameter choice:[4]

- Buoy mass M: the power increases as the mass decreases. A lighter buoy exhibits larger-amplitude motion under a given forcing amplitude, leading to a greater induced current.
- Generator inductance L_i: the power increases as the inductance increases. At resonance, larger inductance corresponds to stronger electromagnetic coupling, leading to a greater induced voltage for the same oscillatory motion.
- LED voltage-current coefficient R_l: the dependence of the power upon the linearised voltage-current coefficient for the LED is more involved, since both the

[4]This analysis assumes that the forcing frequency can be tuned to maintain resonance. If instead the forcing frequency is fixed, variations in M or L_i will generally shift the system away from resonance, and the dependence of power on these parameters becomes more complex.

numerator and denominator increase as R_l increases; however, note that the denominator also depends upon the circuit's resistance $R_i + R_c$.

- For small R_l, such that $R_i + R_c$ dominates in the denominator, increasing R_l increases the numerator, and the power is increased.
- However, for larger R_l (when R_l dominates in the denominator), increasing R_l increases the denominator faster than it increases the numerator, decreasing the power overall.
- Therefore, there exists some optimal R_l; by differentiating $\hat{P}_g$ with respect to R_l and setting the result equal to 0, we find that R_l should be equal to $R_i + R_c$.[5] That is, we should pick an LED whose specifications result in a coefficient which matches the resistance of the wider circuit.

6.1.3 Designing a device to match a location's incident-wave frequency

Consider again the resonance frequency (6.9). This can be calculated for a particular instance of the device (since γ, $\mathcal{G}$, L_i and M would all be known). This means that, within practical operating constraints, these fixed parameters could be chosen to match any desired frequency, and, theoretically, bespoke devices could be designed for particular locations with particular incident-wave frequencies.

Now, if a capacitor is introduced, the resonance frequency becomes

$$\omega = \sqrt{\frac{\gamma^2\mathcal{G}^2 + {}^{M}\!/_{C}}{L_i M} - \frac{\mathcal{R}^2}{2L_i^2}}\,.$$

Note that C need not be fixed; it is possible to build a capacitor whose capacitance can be varied, mechanically or electrically. Rearranging for C, we find that resonance is achieved when

$$C = \left(\omega^2 L_i - \frac{\gamma^2\mathcal{G}^2}{M} + \frac{\mathcal{R}^2}{2L_i}\right)^{-1}. \tag{6.10}$$

Therefore, by utilising a variable capacitor, resonance can theoretically be maintained for a changing frequency, and not just for a fixed idealised frequency given by (6.9).

[5] This result is the well-known *maximum power transfer theorem* (e.g. [107, 148–149]).

Note that (6.10) requires

$$\omega^2 L_i > \frac{\gamma^2 \mathcal{G}^2}{M} - \frac{\mathcal{R}^2}{2L_i},$$

since a negative capacitance is not physically realisable.

Using techniques from control theory, it may be possible to design a controller that can measure the incident-wave frequency and automatically tune the capacitor capacitance in real-time to respond to changes in frequency and keep the device in resonance. Such use of a variable capacitor would have some practical implications. Depending on the typical sea-state of the location in question, the range of frequencies experienced may be practically too great for a single variable capacitor to cover. Similarly, the power levels experienced in the circuit may require a capacitor that is physically large, making rapid or precise tuning more difficult, and introducing additional non-ideal effects such as resistive losses, parasitic inductance and dielectric leakage (see e.g. Horowitz and Hill [108]). In conclusion, adaptive tuning may be beneficial for variable sea-states, but its practical implementation must balance responsiveness, efficiency, and hardware constraints.

6.2 Rewiring the generator to have n coils in parallel

Recall the description of the generator in Section 2.2.2: the magnet has length L_m, radius A_m and magnetic moment m, and the coil has length L, radius a, wire diameter D and turn number $n_c = L/D$. The coil is made up of a single wire, connected at either end to the rest of the circuit.

Now, consider replacing this single coil with multiple smaller coils, say n of them, each of length L/n and with turn number n_c/n. There are then two broad ways to connect these individual coils to each other and to the rest of the circuit. We could connect them to each other in series, and connect the first and last coils to either ends of the rest of the circuit; this would be equivalent to the single-coil case.

Alternatively, consider connecting each coil to the rest of the circuit individually, in parallel. This would create n separate induction coils, each individually experiencing the effect of the moving magnet, and each having its own induced current I_j. See

Figure 6.1 for an example with three coils. If each branch is rectified independently before being connected to the rest of the circuit, the (rectified) currents of each branch sum to give the total current passing through the rest of the circuit, such that the total current I_{tot} satisfies $|I_{tot}| = \sum_j |I_j|$.

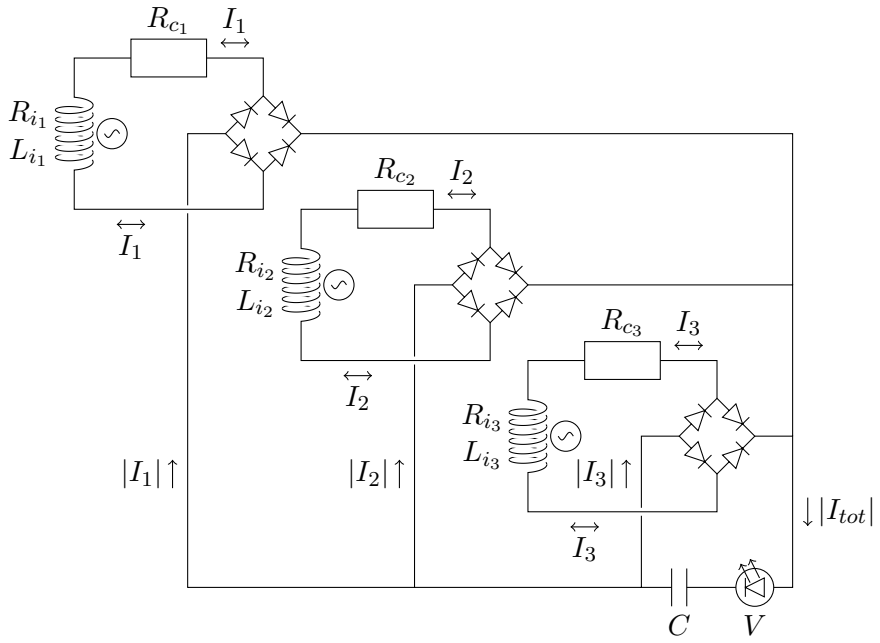


Figure 6.1: A schematic of the circuit with three individual coils, connected in parallel. Each coil has an induced current I_j, with the total rectified current satisfying $|I_{tot}| = |I_1| + |I_2| + |I_3|$.

6.2.1 The voltage in the j^{th} coil

Now, we can calculate the voltage induced in the j^{th} coil, in much the same way as we did for the single coil in Section 2.2.2. Recall that the single coil's induced voltage is given by $\gamma G\dot{Z}$, with

$$\gamma = \frac{\mu_0 m a^2 n_c}{2L}$$

and

$$G \approx \left(a^2 + (Z_0 + \alpha_h H_m - {}^{L}/_{2} - Z)^2\right)^{-{}^{3}/_{2}} - \left(a^2 + (Z_0 + \alpha_h H_m + {}^{L}/_{2} - Z)^2\right)^{-{}^{3}/_{2}}.$$

Note that each of the n coils still has radius a and wire diameter D, while the same magnet interacts with each of them; therefore γ remains the same for all coils. However,

G depends explicitly upon the length (and position relative to the magnet) of the coil, and will be different for each branch.

Recalling Appendix A.2, G was found for the single coil by integrating the (scaled) voltage-per-turn

$$\frac{3(Z_0 + \alpha_h H_m + \bar{z} - Z)}{(a^2 + (Z_0 + \alpha_h H_m + \bar{z} - Z)^2)^{5/2}} \tag{6.11}$$

along its length $\bar{z} \in [-L/2, L/2]$. Similarly, the equivalent expression for the j^{th} coil can be found by carrying out a similar integration over the range of $\bar{z}$ spanned by that coil.

First, note that the antiderivative of (6.11) is

$$-\left(a^2 + (Z_0 + \alpha_h H_m + \bar{z} - Z)^2\right)^{-3/2},$$

and the integral over $\bar{z} \in [\bar{z_0}, \bar{z_1}]$ is simply

$$\left(a^2 + (Z_0 + \alpha_h H_m + \bar{z_0} - Z)^2\right)^{-3/2} - \left(a^2 + (Z_0 + \alpha_h H_m + \bar{z_1} - Z)^2\right)^{-3/2}.$$

Then, since the single coil spans the region $\bar{z} \in [-L/2, L/2]$, the j^{th} individual coil spans

$$\bar{z} \in \left[L\left(\frac{j}{n} - \frac{1}{2} - \frac{1}{n}\right), L\left(\frac{j}{n} - \frac{1}{2}\right)\right],$$

and its voltage is $\gamma G_j \dot{Z}$, where

$$G_j \approx \left(a^2 + (Z_0 + \alpha_h H_m + L\,(j/n - 1/2 - 1/n) - Z)^2\right)^{-3/2} - \left(a^2 + (Z_0 + \alpha_h H_m + L\,(j/n - 1/2) - Z)^2\right)^{-3/2}.$$

Of course, after linearising, this becomes $G_j(Z_0) \equiv \mathcal{G}_j$, with

$$\mathcal{G}_j = \left(a^2 + (\alpha_h H_m + L\,(j/n - 1/2 - 1/n))^2\right)^{-3/2} - \left(a^2 + (\alpha_h H_m + L\,(j/n - 1/2))^2\right)^{-3/2}. \tag{6.12}$$

6.2.2 A system of $2+2n$ ODEs

Now, a system of ODEs much like (6.2) can be obtained for the n-coil generator. For the buoy equations, the electromagnetic force from the generator simply becomes

$$\gamma \mathcal{G} I \to \gamma \sum_{j=1}^{n} \mathcal{G}_j I_j.$$

Then, instead of a single pair of equations for the charge Q and current I in the single coil, we have n pairs of equations for Q_j and I_j:

$$\begin{aligned} \dot{Q}_j &= I_j, \\ L_{i_j} \dot{I}_j &= \gamma \mathcal{G}_j W - \frac{Q_j}{C} - I_j (R_{i_j} + R_{c_j} + R_{l_j}). \end{aligned}$$

Here, L_{i_j} and R_{i_j} are the inductance and resistance of the j^{th} coil; recalling their definitions for the single coil in (2.17) and (2.19), these scale linearly with the coil's length, and we have simply $L_{i_j} = L_i/n$ and $R_{i_j} = R_i/n$ for all j.[6] Similarly, we set $R_{c_j} = R_c/n$, since previously we took $R_c = R_i$.

For the combined LED harvest, we approximate its effect on each of the individual coils as being equally shared, giving $R_{l_j} = R_l/n$ also. Note this is a crude approximation: in reality, the "proportion" of the harvest felt by each coil depends on how its own induced current compares to the total current. A more accurate representation would produce a term like $V \sim \sum_i R_{l_j} I_j$, however this coupling of each current ODE to all others breaks the block structure of the system's matrix, preventing an analytic solution via the Schur complement.

Therefore, the full $(2+2n)$-equation ODE system for the buoy-driven n-coil generator reads:

$$\begin{aligned} M\dot{W} &= -\gamma \sum_{j=1}^{n} \mathcal{G}_j I_j + F, \\ \dot{Z} &= W, \end{aligned}$$

[6] Recall that L_i and R_i are the inductance and resistance of the single coil, *not* of the i^{th} individual coil.

$$\dot{Q}_j = I_j,$$
$$\frac{\mathcal{L}}{n}\dot{I}_j = \gamma \mathcal{G}_j W - \frac{Q_j}{C} - I_j \frac{\mathcal{R}}{n},$$

with $\mathcal{L} \equiv L_i$. In matrix form, this system is

$$\begin{pmatrix} i\omega M & 0 & 0 & \gamma\mathcal{G}_1 & \dots & 0 & \gamma\mathcal{G}_n \\ -1 & i\omega & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & i\omega & -1 & \dots & 0 & 0 \\ -\gamma\mathcal{G}_1 & 0 & \frac{1}{C} & \frac{i\omega\mathcal{L}+\mathcal{R}}{n} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & i\omega & -1 \\ -\gamma\mathcal{G}_n & 0 & 0 & 0 & \dots & \frac{1}{C} & \frac{i\omega\mathcal{L}+\mathcal{R}}{n} \end{pmatrix} \begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q}_1 \\ \hat{I}_1 \\ \vdots \\ \hat{Q}_n \\ \hat{I}_n \end{pmatrix} = \begin{pmatrix} A \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{6.13}$$

The matrix in (6.13) has a very similar structure to the single-coil one in (6.3), primarily due to the lack of coupling between the individual coils and the similar coupling each of them has with the buoy. Leveraging this similar block structure, the same Schur complement method (see Appendix A.7.2) can be used to obtain the solutions to (6.13), despite being generalised for arbitrary n:

$$\begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q}_j \\ \hat{I}_j \end{pmatrix} = \begin{pmatrix} \dfrac{i(\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C})A}{\omega\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C})M\right)} \\ \dfrac{(\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C})A}{\omega^2\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C})M\right)} \\ \dfrac{-in\gamma\mathcal{G}_j A}{\omega\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C})M\right)} \\ \dfrac{n\gamma\mathcal{G}_j A}{n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C})M} \end{pmatrix}. \tag{6.14}$$

As we did for the single coil generator in Section 6.1.1, we can obtain an explicit expression for the power output of the n-coil generator. Analogous to (6.1), the instantaneous

power output is

$$P_g = \sum_{j=1}^{n} I_j^2 R_l.$$

As before, using the real and imaginary parts of (6.14) and integrating over $t \in [0, n\pi/\omega]$, we eventually find that

$$\frac{\hat{P}_g}{|A|^2} = \frac{R_l (n\gamma)^2 \sum_j \mathcal{G}_j^2}{2\left(\left(n\gamma^2 \sum_j \mathcal{G}_j^2 - \omega^2 \mathcal{L} M + {}^{nM}/_{C}\right)^2 + (\omega \mathcal{R} M)^2\right)}. \tag{6.15}$$

6.2.3 The dependence of the power output on the number of coils

Using (6.15), the impact of simply increasing the number of coils—while keeping all other parameters fixed—on the power output is investigated. A different set of electromagnetic parameters to the ones used in Section 3.6 are used, reflecting recent advancements in the development of an experimental lab setup [72]. These values are given in Table 6.1.

Table 6.1: The (updated) electromagnetic parameter values used in the n coil tests.

Parameter	Value	Parameter	Value
		Prescribed	
M	$0.08\,\mathrm{kg}$	L_m	$0.025\,\mathrm{m}$
A_m	$7.5 \times 10^{-3}\,\mathrm{m}$	H_m	$0.2\,\mathrm{m}$
m	$10\,\mathrm{A\,m^2}$	L	$0.025\,\mathrm{m}$
a	$0.012\,\mathrm{m}$	α_h	0–0.5
D	$2.769 \times 10^{-4}\,\mathrm{m}$	σ	$5.96 \times 10^{7}\,\Omega^{-1}\,\mathrm{m}^{-1}$
K	0.53	μ_0	$4\pi \times 10^{-7}\,\mathrm{H\,m^{-1}}$
C	∞	n_q	1
V_T	$2.05\,\mathrm{V}$	I_{sat}	$0.02\,\mathrm{A}$
		Calculated	
n_c	903 (see note)[7]	L_i	$9.82 \times 10^{-3}\,\mathrm{H}$
R_i	$19.0\,\Omega$	R_c	$19.0\,\Omega$
R_l	$102.5\,\Omega$	γ	$3.27 \times 10^{-5}\,\mathrm{kg\,m^4\,s^{-2}\,A^{-1}}$

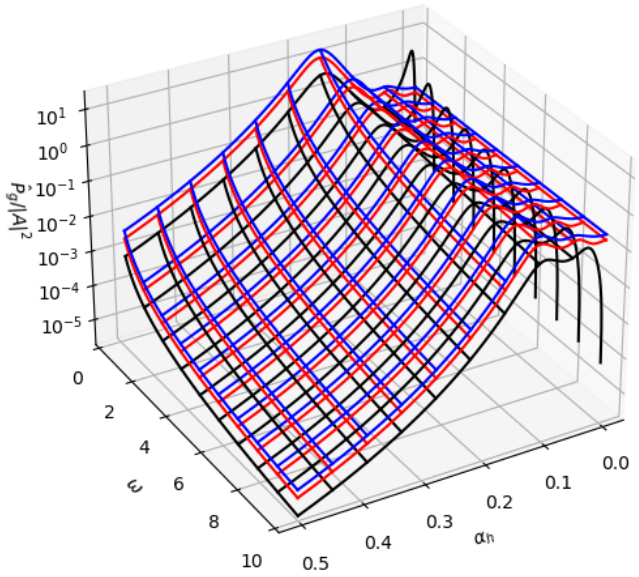


Figure 6.2: The power-output-per-unit-forcing $\hat{P}_g/|A|^2$, given by (6.15), as a function of forcing frequency ω and coil offset parameter α_h. The single coil (black), triple coil (red) and quintuple coil (blue) cases are shown.

Figure 6.2 shows the calculated power-output-per-unit-forcing, given by (6.15), as a function of forcing frequency ω and coil offset parameter α_h,[8] for a single coil (black), a triple coil (red), and a quintuple coil (blue). ω varies between 0.5–10 Hz, and α_h takes values of 0–0.5. There is a very small section, for low ω and for α_h close to 0, where the single coil performs best, but in general we see that the power is increased as the number of coils grows. This can be understood by considering Figure 2.6, the dependence of position on $\gamma G(Z)$. The offset parameter can be used to tune the position of the graph, so that the peak lies at $Z = Z_0$ (the red circle), but as soon as the position moves the induced voltage quickly drops. This is not mathematically relevant in the linearised limit, but in reality the position is oscillating about this point and the voltage is rarely at its maximum. The effect of splitting the coil is that each coil's peak voltage is at a different position, meaning a greater overall voltage is induced for a wider spread of positions.

The impact of increasing the number of coils can be quantified by calculating the root-

[7] The number of turns is increased tenfold through a multilayer winding; however, a uniform coil radius is assumed, neglecting the radial variation between each layer.

[8] Here, we remark that in [2] the calculations were carried-out using varying Z, with the full magnetic field application function $G(Z)$. This is inconsistent with the linearisation carried out, however, since there is no Z-dependence in (6.12). Instead, varying α_h allows for similar behaviour to be observed.

Table 6.2: The root-mean-square of $\hat{P}_g(\alpha_h, \omega)$, in $\mathrm{W\,N^{-2}}$, and the relative differences, for differing coil-numbers n. The middle column gives the relative difference to the base-level of $n = 1$, while the right-hand column gives the relative difference between successive n.

n	$\mathrm{RMS}\left(\hat{P}_g^{(n)}\right)$	$\frac{\mathrm{RMS}\left(\hat{P}_g^{(n)} - \hat{P}_g^{(1)}\right)}{\mathrm{RMS}\left(\hat{P}_g^{(1)}\right)}$	$\frac{\mathrm{RMS}\left(\hat{P}_g^{(n)} - \hat{P}_g^{(n-2)}\right)}{\mathrm{RMS}\left(\hat{P}_g^{(n-2)}\right)}$
1	0.338	-	-
3	0.885	1.797	1.797
5	1.469	3.505	0.661
7	2.054	5.230	0.399
9	2.640	6.959	0.285

mean-square of $\hat{P}_g(\alpha_h, \omega)$ for a given n, as well as the relative difference to the base-level of $n = 1$, and between successive n. Table 6.2 gives these values for $n \in [1, 3, 5, 7, 9]$. We see that the power output does indeed increase with each n, beginning with $0.338\,\mathrm{W\,N^{-2}}$ for the single coil, and increasing to $2.640\,\mathrm{W\,N^{-2}}$ when 9 coils are used, approximately a seven-fold increase in performance. The right-hand column shows that the relative increase reduces as n increases: the increase from 1 to 3 coils is $\approx 180\,\%$, but for 7 to 9 is only $\approx 28.5\,\%$, Thus, there are diminishing returns as the number of coils continues to increase; eventually, physical operating constraints will mean any further increase in coil-number (and therefore decrease in each coil's size) would become impractical, and there is likely some optimal number.

Chapter 7

Conclusions and Future Work

This thesis has investigated improvements to the mathematical and numerical modelling of a novel wave-energy device currently in its early development phase. Chapter 2 established the governing nonlinear 3D potential flow formulation from first principles, providing a consistent Lagrangian framework for subsequent modelling and linearisation; this framework was used in Chapter 3 to explore a depth-averaged 2D approximation based on the BL model, demonstrating its limitations in capturing the key coupling between the buoy and free surface.

Chapter 4 outlined the numerical framework used to solve the linearised SW system, including the adoption of the Firedrake finite element library and an approach for handling subdomain problems. Building on this, Chapter 5 developed a flexible and efficient computational algorithm, significantly improving performance over existing in-house MATLAB implementations and enabling surrogate-enabled optimisation studies to be carried out.

Finally, Chapter 6 analysed the buoy–generator sub-model, using derived expressions for power output to identify strategies to enhance performance through system design and control. Taken together, these contributions provide a stronger theoretical foundation and a more efficient computational framework for the analysis and optimisation of the device.

7.1 Summary of contributions

This thesis makes several contributions to the modelling and analysis of the device, spanning theoretical formulation, reduced-order modelling, computational methodology, and device-level design insights.

7.1.1 Theoretical formulation and model approximations in the Lagrangian framework

A full derivation of the nonlinear, 3D potential flow equations using Lagrangian mechanics has been provided, establishing a rigorous and flexible framework for modelling the device and its various approximations. Expanding upon the work carried out by Bokhove et al. [44, 45], the governing Lagrangian has been constructed from first principles, clarifying its relationship to the underlying physical system and its role in deriving the governing nonlinear equations. This establishes a clearer theoretical basis than previous formulations, and supports subsequent model development.

The framework also enables systematic model simplification and extension. In particular, it is shown how additional physical components, such as a capacitor in the electromagnetic generator subsystem, can be incorporated naturally through additional terms in the Lagrangian. Furthermore, the use of Lagrangian "quadricisation" provides a direct route to consistent linearisation of the governing equations.

Additionally, the framework allows reduced-order hydrodynamic models to be consistently coupled with device constraints, such as the interaction between a buoy and the free surface, within the same Lagrangian setting.

This work demonstrates the flexibility and extensibility of a Lagrangian approach to modelling wave energy converters, representing (to the author's knowledge) the first such application in this context. As such, it provides a general framework that can be extended to more complex device configurations and to other multi-component systems beyond wave energy.

7.1.2 Validity and limitations of reduced-order modelling approaches

The applicability of reduced-order hydrodynamic models to the device has been critically assessed, through the investigation of a depth-averaged BL formulation. While such models offer improved dispersive accuracy compared to SW theory, it has been shown that, using a discretisation which enforces the fundamental coupling constraint between the buoy and the free surface via an equation for the pressure at their interface, they are incompatible with said coupling.

This result highlights an important limitation: increased physical fidelity in the wave model does not necessarily translate to improved suitability for coupled device simulations. Consequently, the use of simpler SW formulations, as adopted in the improved numerical framework, is justified not only on computational grounds but also in terms of model consistency.

7.1.3 Computational methodology and implementation

A flexible finite element framework has been developed for the efficient simulation of the device in the linearised limit, leveraging the Firedrake library to enable scalable and maintainable implementations. The framework includes a novel approach to handling subdomain problems with Firedrake, allowing complex device configurations to be incorporated within a unified formulation. It also allows easy switching between different orders of accuracy, and can be readily extended to handle nonlinear solves.

The developed computational algorithm is both more-easily parameterised and significantly more efficient than previous in-house MATLAB implementations, enabling substantially faster simulations and easier handling of parameter variation. These improvements are not merely technical, but expand the practical scope of the model by making tasks such as parameter sweeps and surrogate-enabled optimisation studies computationally feasible. Such investigations are essential for the continued development and optimisation of the device.

7.1.4 Device-level modelling and design insights

An analytical model of the buoy–generator subsystem has been developed, yielding explicit power-output expressions for generalised coil configurations. This provides a direct link between electrical design parameters and device performance, enabling rapid evaluation of design changes without the need for full system simulation.

The analysis identifies practical strategies for performance enhancement, including the use of variable capacitance, and alternative coil configurations (such as parallel subcoils). These results demonstrate how relatively simple modifications to the electrical subsystem can lead to measurable improvements in power output, highlighting the importance of integrated modelling across hydrodynamic and electrical components in the design and optimisation of wave-energy devices, and providing actionable insights for improving device design and control strategies.

Collectively, these contributions establish a coherent framework linking theoretical formulation, model selection, computational implementation, and device-level design, thereby advancing both the understanding and practical modelling capability of this class of wave-energy device.

7.2 Limitations of the present work

While this thesis has made significant advances in the modelling, numerical implementation, and design analysis of the device, several limitations should be noted.

First, the hydrodynamic models considered are primarily linearised or reduced-order formulations. Although these provide computational efficiency and allow for extensive parameter studies, they neglect fully nonlinear wave interactions and three-dimensional effects that may be important under certain operational conditions; this is particularly relevant in the context of performance maximisation via motion amplification. The depth-averaged BL formulation was found to be incompatible with the device coupling constraint, highlighting the challenges in balancing model fidelity and physical consistency.

Second, while being flexible and efficient for linearised simulations, the computational

framework has not yet been fully extended to produce nonlinear or 3D simulations. Some aspects of real-world device behaviour, including extreme waves and fully coupled fluid–structure–electrical dynamics, remain outside the scope of the current implementation.

Third, the device-level analysis of the buoy–generator subsystem is based on simplified electrical models and regularised buoy motion. It also assumes idealised coil configurations. While the derived expressions provide clear design insights and guidance for optimisation, further experimental validation and inclusion of more detailed electromagnetic effects would be required to fully quantify real-world performance.

These limitations are not weaknesses of the approach per se, but natural outcomes of focusing on a tractable, analytically and computationally accessible framework. They also point directly to avenues for future work, including nonlinear hydrodynamic modelling, fully coupled simulations, and more detailed experimental characterisation of the electrical subsystem.

7.3 Future work

The work presented in this thesis opens several directions for further research and development, covering enhancements to the modelling framework as well as improvements to device performance.

First, experimental validation of the models and algorithms remains an important step. Laboratory testing with scaled prototypes would allow comparison of predicted hydrodynamic forces, buoy motion, and electrical power output with measured data, providing a crucial benchmark for the theoretical and computational frameworks developed here. Work in this area is already underway [72].

Second, the computational framework could be extended to include fully nonlinear and three-dimensional simulations, selectable by the user. Incorporating nonlinear wave interactions and three-dimensional effects would improve fidelity under realistic sea conditions and allow the study of extreme wave events. Additionally, exploration of alternative reduced-order models may yield more accurate or computationally efficient

approximations that avoid the previously-observed coupling incompatibility, or alternatively provide insight into its underlying causes.

Third, ongoing Firedrake development offers further avenues for improvements. In particular, leveraging the new `fd.Submesh` feature[1] may provide an alternative method for solving the subdomain problem, allowing the use of higher-order basis functions where the current approach cannot. This feature has been trailed alongside this thesis, but so far it has not been possible to develop a working code. Furthermore, as seen in [57–60], `fd.derivative` could be employed to derive weak forms in Firedrake directly from the Lagrangian, removing the need to carry out derivations by hand; this is especially useful when employing nonlinear coordinate transforms that produce complicated geometric terms.

Finally, the developed algorithm can be used to explore the multi-dimensional optimisation of the devices design parameters. This is a complex problem with many interlinked parameter relationships, but assessing how the performance is impacted by different designs is critical in the development of a high-performing device. Additionally, device performance could be further enhanced through advanced control strategies. Techniques such as latching control, resonance tuning, or real-time adaptive control could be applied to maximise power capture, particularly when combined with the analytical insights gained from the buoy–generator subsystem modelling.

Collectively, these directions offer a roadmap for improving the theoretical, computational, and experimental basis of the device, progressing the project from a tractable research model towards a validated, deployable wave energy converter.

[1] See https://www.firedrakeproject.org/firedrake.html#firedrake.mesh.Submesh.

References

[1] Bolton, J.M., Bokhove, O., Borman, D.J., Kalogirou, A. and Thompson, H.M. Towards Optimization of a Wave-to-Wire Energy Device in a Breakwater Contraction. In: Greaves, D. ed. *14th European Wave and Tidal Energy Conference, 5–9 September 2021, Plymouth, UK*. [Online]. Plymouth, UK: University of Plymouth, 2021. [Accessed 25 July 2025]. Available from: URL https://submissions.ewtec.org/proc-ewtec/article/view/ewtec2021-offline-proceedings (cited on pp. i, 70).

[2] Bokhove, O., Bolton, J.M. and Thompson, H.M. Geometric Power Optimisation of a Rogue-Wave Energy Device in a (Breakwater) Contraction. In: *2024 IEEE Conference on Control Technology and Applications (CCTA), 21–23 August 2024, Newcastle upon Tyne, UK*. [Online]. Piscataway, NJ: IEEE, 2024. [Accessed 23 February 2026]. Available from: DOI 10.1109/CCTA60707.2024.10666543 (cited on pp. i, 70, 159).

[3] Our World in Data. *Share of Energy Consumption by Source, World*. [Online]. 2025. [Accessed 28 March 2026]. Available from: URL https://ourworldindata.org/grapher/share-energy-source-sub?country=~OWID_WRL (cited on p. 1).

[4] Eurostat. *Renewable Energy Statistics*. [Online]. 2025. [Accessed 28 March 2026]. Available from: URL https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Renewable_energy_statistics#Share_of_renewable_energy_almost_tripled_between_2004_and_2024 (cited on p. 1).

[5] Center for Climate and Energy Solutions. *Renewable Energy*. [Online]. [no date]. [Accessed 28 March 2026]. Available from: URL https://www.c2es.org/content/renewable-energy (cited on p. 1).

[6] International Energy Agency. *Renewable Energy Market Update. Outlook for 2021 and 2022*. [Online]. Pais: International Energy Agency, 2021. [Accessed 24 October 2024]. Available from: URL https://www.iea.org/reports/renewable-energy-market-update-2021 (cited on p. 1).

[7] International Energy Agency. *Total renewable capacity additions by technology, 2019–2024*. [Online]. 2025. [Accessed 28 March 2026]. Available from: URL https://www.iea.org/data-and-statistics/charts/total-renewable-capacity-additions-by-technology-2019-2024 (cited on p. 1).

[8] International Energy Agency. *Renewable Energy Market Update. Outlook for 2023 and 2024*. [Online]. Pais: International Energy Agency, 2023. [Accessed 24 October 2024]. Available from: URL https://www.iea.org/reports/renewable-energy-market-update-june-2023 (cited on p. 1).

[9] Our World in Data. *Global installed renewable energy capacity by technology*. [Online]. 2025. [Accessed 28 March 2026]. Available from: URL https://ourworldindata.org/grapher/installed-global-renewable-energy-capacity-by-technology (cited on p. 2).

[10] Ocean Energy Systems. *IEA-OES Annual Report 2007*. [Online]. Lisbon: IEA-OES Executive Committee, 2007. [Accessed 25 June 2021]. Available from: URL https://www.ocean-energy-systems.org/publications/oes-annual-reports/document/oes-annual-report-2007/ (cited on p. 2).

[11] Barstow, S., Mørk, G., Mollison, D. and Cruz, J. The Wave Energy Resource. In: Cruz, J. ed. *Ocean Wave Energy: Current Status and Future Prespectives*. [Online]. Berlin: Springer, 2008, pp. 93–132. [Accessed 25 June 2021]. Available from: DOI 10.1007/978-3-540-74895-3_4 (cited on p. 2).

[12] Li, B., Basu, S., Watson, S.J. and Russchenberg, H.W.J. A Brief Climatology of Dunkelflaute Events over and Surrounding the North and Baltic Sea Areas. *Energies*. [Online]. 2021, **14**(20), article no: 6508 [no pagination]. [Accessed 3 September 2025]. Available from: DOI 10.3390/en14206508 (cited on p. 3).

[13] Robinson, J. *UK weather: 'Anticyclonic gloom' blamed as no sunshine recorded so far this month in many parts of country*. [Online]. 2024. [Accessed 3 September 2025]. Available from: URL https://news.sky.com/story/uk-weather-anticy

clonic-gloom-blamed-as-no-sunshine-recorded-so-far-this-month-in-many-parts-of-country-13250596 (cited on p. 3).

[14] Rudling, P. *The reasons for the high electricity prices in Sweden and Europe*. [Online]. 2022. [Accessed 3 September 2025]. Available from: URL https://energyeducation.se/the-reasons-for-the-high-electricity-prices-in-sweden-and-europe/ (cited on p. 3).

[15] Jansen, M., Staffell, I., Green, R. and Green, T. *When the wind goes, gas fills in the gap*. [Online]. [no date]. [Accessed 3 September 2025]. Available from: URL https://reports.electricinsights.co.uk/q1-2021/when-the-wind-goes-gas-fills-in-the-gap/ (cited on pp. 3, 4).

[16] Drax. *Electric Insights - A landmark year for Britain's power system, new Government AI plans and 'Dunkelflaute'*. [Online]. 2025. [Accessed 3 September 2025]. Available from: URL https://energy.drax.com/insights/electric-insights-q4-2024/ (cited on p. 4).

[17] Odum, H.T. Self-Organization, Transformity, and Information. *Science*. [Online]. 1988, **242**, pp. 1132–1139. [Accessed 26 June 2021]. Available from: DOI 10.1126/science.242.4882.1132 (cited on p. 4).

[18] Thomas, G. The Theory Behind the Conversion of Ocean Wave Energy: a Review. In: Cruz, J. ed. *Ocean Wave Energy: Current Status and Future Prespectives*. [Online]. Berlin: Springer, 2008, pp. 41–91. [Accessed 29 May 2026]. Available from: DOI 10.1007/978-3-540-74895-3_3 (cited on p. 5).

[19] McCormick, M.E. *Ocean Wave Energy Conversion*. New York: Wiley, 1981 (cited on p. 5).

[20] Salter, S.H. World Progress in Wave Energy—1988. *International Journal of Ambient Energy*. [Online]. 1989, **10**(1), pp. 3–24. [Accessed 2 April 2025]. Available from: DOI 10.1080/01430750.1989.9675119 (cited on p. 5).

[21] Thorpe, T.W. *A Brief Review of Wave Energy. A Report Produced for the UK Department of Trade and Industry*. [Online]. Didcot: AEA Technology, 1999. [Accessed 2 April 2025]. Available from: URL http://large.stanford.edu/courses/2011/ph240/yan1/docs/thorpe.pdf (cited on p. 5).

[22] Thorpe, T.W. An Overview of Wave Energy Technologies: Status, Performance and Costs. In: *Wave Power: Moving Towards Commercial Viability*. [Online]. London: Wiley, 2000. [Accessed 2 April 2025]. Available from: URL http://large.stanford.edu/courses/2011/ph240/yan1/docs/overview.pdf (cited on p. 5).

[23] Clément, A., McCullen, P., Falcão, A., Fiorentino, A., Gardner, F., Hammarlund, K., Lemonis, G., Lewis, T., Nielsen, K., Petroncini, S., Pontes, M.T., Schild, P., Sjöström, B.-O., Sørensen, H.C. and Thorpe, T. Wave Energy in Europe: Current Status and Perspectives. *Renewable and Sustainable Energy Reviews*. [Online]. 2002, **6**(5), pp. 405–431. [Accessed 2 April 2025]. Available from: DOI 10.1016/S1364-0321(02)00009-6 (cited on p. 5).

[24] Falnes, J. A Review of Wave-Energy Extraction. *Marine Structures*. [Online]. 2007, **20**(4), pp. 185–201. [Accessed 2 April 2025]. Available from: DOI 10.1016/j.marstruc.2007.09.001 (cited on p. 5).

[25] Falcão, A.F.O. Wave Energy Utilization: a Review of the Technologies. *Renewable and Sustainable Energy Reviews*. [Online]. 2010, **14**(3), pp. 899–918. [Accessed 20 April 2020]. Available from: DOI 10.1016/j.rser.2009.11.003 (cited on pp. 5, 7).

[26] López, I., Andreu, J., Ceballos, S., Martínez de Alegría, I. and Kortabarria, I. Review of Wave Energy Technologies and the Necessary Power-Equipment. *Renewable and Sustainable Energy Reviews*. [Online]. 2013, **27**, pp. 413–434. [Accessed 2 April 2025]. Available from: DOI 10.1016/j.rser.2013.07.009 (cited on p. 5).

[27] Pecher, A. and Kofoed, J.P. *Handbook of Ocean Wave Energy*. Berlin: Springer, 2017 (cited on p. 5).

[28] Rusu, E. and Onea, F. A Review of the Technologies for Wave Energy Extraction. *Clean Energy*. [Online]. 2018, **2**(1), pp. 10–19. [Accessed 2 April 2025]. Available from: DOI 10.1093/ce/zky003 (cited on p. 5).

[29] Aderinto, T. and Li, H. Review on Power Performance and Efficiency of Wave Energy Converters. *Energies*. [Online]. 2019, **12**(22), article no: 4329 [no pagina-

tion]. [Accessed 2 April 2025]. Available from: DOI 10.3390/en12224329 (cited on p. 5).

[30] Edenhofer, O., Pichs-Madruga, R., Sokona, Y., Seyboth, K., Matschoss, P., Kadner, S., Zwickel, T., Eickemeier, P., Hansen, G., Schloemer, S. and von Stechow, C. *Renewable Energy Sources and Climate Change Mitigation. Special Report of the Intergovernmental Panel on Climate Change*. [Online]. Geneva: Intergovernmental Panel on Climate Change, 2011. [Accessed 24 June 2021]. Available from: URL https://www.ipcc.ch/report/renewable-energy-sources-and-climate-change-mitigation (cited on p. 6).

[31] Heath, T.V. A Review of Oscillating Water Columns. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. [Online]. 2012, **370**, pp. 235–245. [Accessed 24 June 2021]. Available from: DOI 10.1098/rsta.2011.0164 (cited on p. 6).

[32] Heath, T., Whittaker, T.J.T. and Boake, C.B. The Design, Construction and Operation of the LIMPET Wave Energy Converter (Islay, Scotland). In: Østergaard, I. and Iversen, S. eds. *Wave Energy for the New Millennium. 4th European Wave Energy Conference, 4–6 December 2000, Aalborg, Denmark*. Tåstrup: Energicenter Danmark, 2001, pp. 49–55 (cited on p. 6).

[33] Torre-Enciso, Y., Ortubia, I., López de Aguileta, L.I. and Marqués, J. Mutriku Wave Power Plant: from the Thinking out to the Reality. In: Backer, G.D., Vantorre, M., Beels, C., Rouck, J. and Frigaard, P. eds. *8th European Wave Energy Conference, 7–11 September 2009, Uppsala, Sweden*. [Online]. Uppsala: Uppsala University, 2009, pp. 319–329. [Accessed 28 June 2021]. Available from: URL https://tethys.pnnl.gov/publications/mutriku-wave-power-plant-thinking-out-reality (cited on p. 6).

[34] Marine Energy. *Mutriku Wave Plant Hits* 2 GW h *Mark*. [Online]. 2020. [Accessed 28 June 2021]. Available from: URL https://www.offshore-energy.biz/mutriku-wave-plant-hits-2gwh-mark (cited on p. 6).

[35] Ocean Energy. *What is the OE Buoy?*. [Online]. 2024. [Accessed 28 June 2021]. Available from: URL https://oceanenergy.ie/oe-buoy (cited on p. 6).

[36] Wells, A.A. *Fluid Driven Rotary Transducer*. GB1595700A. 12 August 1981 (cited on p. 6).

[37] Mehlum, E. Tapchan. In: Evans, D.V. and Falcão, A.F.O. eds. *Hydrodynamics of Ocean Wave-Energy Utilization. IUTAM Symposium, 8–11 July 1985, Lisbon, Portugal*. [Online]. Berlin: Springer, 1986, pp. 51–55. [Accessed 30 March 2021]. Available from: DOI 10.1007/978-3-642-82666-5_3 (cited on pp. 7, 10).

[38] Kofoed, J.P., Frigaard, P., Friis-Madsen, E. and Sørensen, H.C. Prototype Testing of the Wave Energy Converter Wave Dragon. *Renewable Energy*. [Online]. 2006, **31**(2), pp. 181–189. [Accessed 26 June 2021]. Available from: DOI 10.1016/j.renene.2005.09.005 (cited on p. 7).

[39] Vicinanza, D. and Frigaard, P. Wave Pressure Acting on a Seawave Slot-Cone Generator. *Coastal Engineering*. [Online]. 2008, **55**(6), pp. 553–568. [Accessed 26 June 2021]. Available from: DOI 10.1016/j.coastaleng.2008.02.011 (cited on p. 7).

[40] Dolores Esteban, M., López-Gutiérrez, J.-S. and Negro, V. Classification of Wave Energy Converters. *Recent Advances in Petrochemical Science*. [Online]. 2017, **2**(4), article no: 555593 [no pagination]. [Accessed 26 June 2021]. Available from: DOI 10.19080/RAPSCI.2017.02.555593 (cited on p. 8).

[41] EHL Group. *How It Works*. [Online]. 2013. [Accessed 28 June 2021]. Available from: URL http://azurawave.com/technology (cited on p. 8).

[42] Dent, S. *Wave Generator Supplies US Electrical Grid for the First Time*. [Online]. 2015. [Accessed 28 June 2021]. Available from: URL https://www.engadget.com/2015-07-07-wave-generator-hawaii-power-grid.html (cited on p. 8).

[43] Wave Star. *Technology*. [Online]. [no date]. [Accessed 28 June 2021]. Available from: URL http://wavestarenergy.com/about (cited on p. 8).

[44] Bokhove, O., Kalogirou, A. and Zweers, W. From Bore-Soliton-Splash to a New Wave-to-Wire Wave-Energy Model. *Water Waves*. [Online]. 2019, **1**, pp. 217–258. [Accessed 20 April 2020]. Available from: DOI 10.1007/s42286-019-00022-9 (cited on pp. 9–11, 32, 38, 50, 51, 66, 69, 70, 78, 100, 121, 162, 195, 196).

[45] Bokhove, O., Kalogirou, A., Henry, D. and Thomas, G.P. A Novel Wave-Energy Device With Enhanced Wave Amplification and Induction Actuator. *Interna-*

tional Marine Energy Journal. [Online]. 2020, **3**(1), pp. 37–44. [Accessed 20 April 2020]. Available from: DOI 10.36688/imej.3.37-44 (cited on pp. 9–11, 32, 38, 50, 51, 69, 70, 78, 100, 162, 195, 196).

[46] Noren, S.A. *Plant for Utilizing Kinetic Energy*. US4277690A. 7 July 1981 (cited on p. 10).

[47] Madhi, F., Sinclair, M.E. and Yeung, R.W. The "Berkeley Wedge": an Asymmetrical Energy-Capturing Floating Breakwater of High Performance. *Marine Systems and Ocean Technology*. [Online]. 2014, **9**, pp. 5–16. [Accessed 30 March 2021]. Available from: DOI 10.1007/BF03449282 (cited on p. 10).

[48] Yu, T., Guo, Q., Shi, H., Li, T., Meng, X. and He, S. Experimental Investigation of a Novel OWC Wave Energy Converter. *Ocean Engineering*. [Online]. 2022, **257**, article no: 111567 [no pagination]. [Accessed 23 February 2026]. Available from: DOI 10.1016/j.oceaneng.2022.111567 (cited on p. 10).

[49] Anna Kalogirou. *WaveEnergy*. [Online]. 2016. [Accessed 20 April 2020]. Available from: URL https://www.youtube.com/watch?v=SZhe_SOxBWo. [Note: Video originally created by Bokhove, O. and Zweers, W., 2013] (cited on p. 10).

[50] Lanczos, C. *The Variational Principles of Mechanics*. 4th ed. Toronto: University of Toronto Press, 1970 (cited on pp. 14, 18, 19).

[51] González, G. *Single and Double Plane Pendulum*. [Online]. [no date]. [Accessed 20 May 2020]. Available from: URL http://www.phys.lsu.edu/faculty/gonzalez/Teaching/Phys7221/DoublePendulum.pdf (cited on p. 15).

[52] Marsden, J.E. and Morrison, P.J. Noncanonical Hamiltonian Field Theory and Reduced MHD. In: Marsden, J.E. ed. *Fluids and Plasmas: Geometry and Dynamics. Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference in the Mathematical Sciences, 17-–23 July 1983, Boulder, USA*. [Online]. 28. Providence, RI: American Mathematical Society, 1984, pp. 133–150. [Accessed 25 February 2026]. Available from: DOI 10.1090/conm/028/751979 (cited on p. 15).

[53] Galley, C.R. Classical Mechanics of Nonconservative Systems. *Physical Review Letters*. [Online]. 2013, **110**(17), article no: 174301 [no pagination]. [Accessed

25 February 2026]. Available from: DOI 10.1103/PhysRevLett.110.174301 (cited on p. 15).

[54] van Daalen, E.F.G., van Groesen, E. and Zandbergen, P.J. A Hamiltonian Formulation for Nonlinear Wave-Body Interactions. In: *8th International Workshop on Water Waves and Floating Bodies, 23–26 May 1993, St John's, Canada*. [Online]. St John's, NL: Institute for Marine Dynamics, 1993, pp. 159–163. [Accessed 2 April 2025]. Available from: URL http://www.iwwwfb.org/Abstracts/iwwwfb08/iwwwfb08_41.pdf (cited on p. 15).

[55] Salwa, T., Bokhove, O. and Kelmanson, M.A. Variational Modelling of Wave Structure Interactions with an Offshore Wind-Turbine Mast. *Journal of Engineering Mathematics*. [Online]. 2017, **107**, pp. 61–85. [Accessed 10 March 2026]. Available from: DOI 10.1007/s10665-017-9936-4 (cited on p. 15).

[56] Salwa, T.J. *On Variational Modelling of Wave Slamming by Water Waves*. Ph.D. thesis, University of Leeds, 2018 (cited on p. 15).

[57] W. Rehman, O. Bokhove and Kelmanson, M. A Systematic Approach for Developing a Numerical Wavetank to Simulate Driven Shallow- and Deep-Water Waves. In: *Proceedings of ASME 2023 42nd International Conference on Ocean, Offshore & Arctic Engineering, 11–16 June 2023, Melbourne, Australia*. [Online]. New York: The American Society of Mechanical Engineers, 2023, article no: V007T08A018 [no pagination]. [Accessed 10 March 2026]. Available from: DOI 10.1115/OMAE2023-108097 (cited on pp. 15, 166).

[58] Rehman, W. *A Novel Approach for Modelling Water Waves and Fluid-Structure Interactions*. Ph.D. thesis, University of Leeds, 2024 (cited on pp. 15, 166).

[59] Yang, L. *Numerical Wavetanks for Wave Generation, Interaction, and Dissipation: Variational and Computational Modelling*. Ph.D. thesis, University of Leeds, 2025 (cited on pp. 15, 166).

[60] Yang, L., Gidel, F., Bunnik, T., Bokhove, O. and Kelmanson, M. Variationally and Numerically Coupled Water-Wave and Surf Zone Hydrodynamics. *Journal of Engineering Mathematics*. [Online]. 2026, **157**, article no: 2 [no pagination]. [Accessed 10 March 2026]. Available from: DOI 10.1007/s10665-026-10511-9 (cited on pp. 15, 166).

[61] Bridges, T.J. and Donaldson, N.M. Variational Principles for Water Waves from the Veiwpoint of a Time-Dependent Moving Mesh. *Mathematika*. [Online]. 2011, **51**(1), pp. 147–173. [Accessed 2 April 2025]. Available from: DOI 10.1112/S0025579310001233 (cited on p. 15).

[62] Gidel, F.M.P. *Variational Water-Wave Models and Pyramidal Freak Waves*. Ph.D. thesis, University of Leeds, 2018 (cited on p. 15).

[63] Feynman, R.P., Leighton, R.B. and Sands, M. The Principle of Least Action. In: *The Feynman Lectures on Physics. Volume II: Mainly Electromagnetism and Matter*. Reading, MA: Addison-Wesley, 1964, pp. 19.1–19.14 (cited on pp. 16, 17, 21).

[64] The Engineering ToolBox. *Electrical Conductivity - Elements and Other Materials*. [Online]. 2008. [Accessed 12 May 2021]. Available from: URL https://www.engineeringtoolbox.com/conductors-d_1381.html (cited on p. 31).

[65] Lorrain, P. and Corson, D.R. *Electromagnetic Fields and Waves*. 2nd ed. San Francisco: W.H. Freeman, 1970 (cited on pp. 32, 186, 191).

[66] Sze, S.M. and Ng, K.K. *Physics of Semiconductor Devices*. 3rd ed. Hoboken, NJ: Wiley, 2007 (cited on p. 33).

[67] Goldstein, H., Poole, C.P. and Safko, J.L. *Classical Mechanics*. 3rd ed. San Francisco: Addison-Wesley, 2002 (cited on p. 35).

[68] Cotter, C. and Bokhove, O. Variational Water-Wave Model With Accurate Dispersion and Vertical Vorticity. *Journal of Engineering Mathematics*. [Online]. 2010, **67**, pp. 33–54. [Accessed 31 March 2021]. Available from: DOI 10.1007/s10665-009-9346-3 (cited on p. 36).

[69] Cendra, H. and Marsden, J.E. Lin Constraints, Clebsch Potentials and Variational Principles. *Physica D: Nonlinear Phenomena*. [Online]. 1987, **27**(1–2), pp. 63–89. [Accessed 23 May 2026]. Available from: DOI 10.1016/0167-2789(87)90005-4 (cited on p. 37).

[70] Fukagawa, H. and Fujitani, Y. Clebsch Potentials in the Variational Principle for a Perfect Fluid. *Progress of Theoretical Physics*. [Online]. 2010, **124**(3), pp. 517–531. [Accessed 23 May 2026]. Available from: DOI 10.1143/PTP.124.517 (cited on p. 37).

[71] Griffiths, D.J. *Introduction to Electrodynamics*. 4th ed. Cambridge: Cambridge University Press, 2017 (cited on pp. 39, 184–187, 190–193).

[72] Bokhove, O., Thompson, H.M., Naar, O., Grieve, P. and Kim, J. Power Optimisation of a Heaving Buoy in a Wave-Enhancing Contraction. In: Gato, L.M.C. ed. *16th European Wave and Tidal Energy Conference, 7–11 September 2025, Madeira, Portugal*. [Online]. Lisbon: University of Lisbon, 2025. [Accessed 23 February 2026]. Available from: DOI 10.36688/ewtec-2025-1248 (cited on pp. 49, 70, 158, 165).

[73] Benney, D.J. and Luke, J.C. On the Interactions of Permanent Waves of Finite Amplitude. *Journal of Mathematics and Physics*. [Online]. 1964, **43**(1–4), pp. 309–313. [Accessed 13 March 2021]. Available from: DOI 10.1002/sapm1964431309 (cited on p. 52).

[74] Pego, R.L. and Quintero, J.R. Two-Dimensional Solitary Waves for a Benney–Luke Equation. *Physica D: Nonlinear Phenomena*. [Online]. 1999, **132**(4), pp. 476–496. [Accessed 13 March 2021]. Available from: DOI 10.1016/S0167-2789(99)00058-5 (cited on p. 52).

[75] Berger, K.M., Baker, G.R. and Johnson, J.T. A Comparison of Nonlinear Water Wave Models. *International Journal of Computational Fluid Dynamics*. [Online]. 2003, **17**(3), pp. 219–224. [Accessed 9 October 2025]. Available from: DOI 10.1080/1061856021000013286 (cited on p. 52).

[76] Bokhove, O. and Kalogirou, A. Variational Water Wave Modelling: from Continuum to Experiment. In: Bridges, T.J., Groves, M.D. and Nicholls, D.P. eds. *Lectures on the Theory of Water Waves*. [Online]. Cambridge: Cambridge University Press, 2016, pp. 226–260. [Accessed 13 March 2021]. Available from: DOI 10.1017/CBO9781316411155.012 (cited on pp. 52, 56).

[77] Younas, U., Seadawy, A.R., Younis, M., Rizvi, S.T.R. and Althobaiti, S. Diverse Wave Propagation in Shallow Water Waves with the Kadomtsev–Petviashvili–Benjamin–Bona–Mahony and Benney–Luke Integrable Models. *Open Physics*. [Online]. 2021, **19**(1), pp. 808–818. [Accessed 9 October 2025]. Available from: DOI 10.1515/phys-2021-0100 (cited on p. 52).

[78] Johnson, R.S. *A Modern Introduction to the Mathematical Theory of Water Waves*. Cambridge: Cambridge University Press, 1997 (cited on p. 52).

[79] Green, A.E. and Naghdi, P.M. A Derivation of Equations for Wave Propagation in Water of Variable Depth. *Journal of Fluid Mechanics*. [Online]. 1976, **78**(2), pp. 237–246. [Accessed 2 March 2026]. Available from: DOI 10.1017/S0022112076002425 (cited on p. 75).

[80] Liu, W.K., Li, S. and Park, H. Eighty Years of the Finite Element Method: Birth, Evolution, and Future. 2021. [Accessed 9 October 2025]. Available from: DOI 10.48550/arXiv.2107.04960 (cited on p. 79).

[81] Chao, T.Y. and Chow, W.K. A Review on the Applications of Finite Element Method to Heat Fransfer and Fluid Flow. *International Journal on Architectural Science*. [Online]. 2002, **3**(1), pp. 1–19. [Accessed 9 October 2025]. Available from: URL https://hkapi.lib.cuhk.edu.hk/items/029b3b3d-e3e3-4c5d-8873-0cca30a334da (cited on p. 79).

[82] Diersch, H-J.G. *FEFLOW: Finite Element Modeling of Flow, Mass and Heat Transport in Porous and Fractured Media*. Berlin: Springer, 2014 (cited on p. 79).

[83] Ham, D.A., Kelly, P.H.J., Mitchell, L., Cotter, C.J., Kirby, R.C., Sagiyama, K., Bouziani, N., Vorderwuelbecke, S., Gregory, T.J., Betteridge, J., Shapero, D.R., Nixon-Hill, R.W., Ward, C.J., Farrell, P.E., Brubeck, P.D., Marsden, I., Gibson, T.H., Homolya, M., Sun, T., McRae, A.T.T., Luporini, F., Gregory, A., Lange, M., Funke, S.W., Rathgeber, F., Bercea, G.-T. and Markall, G.R. *Firedrake User Manual*. [Online]. London: Imperial College London et al., 2023. [Accessed 23 January 2025]. Available from: DOI 10.25561/104839 (cited on pp. 83, 101, 103, 105).

[84] Runge, C. Über Empirische Funktionen und die Interpolation Zwischen Äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*. 1901, **46**, pp. 224–243 (cited on p. 83).

[85] Isaac, T. Recursive, Parameter-Free, Explicitly Defined Interpolation Nodes for Simplices. *SIAM Journal on Scientific Computing*. [Online]. 2020, **42**(6), pp. 4046–4062. [Accessed 28 November 2024]. Available from: DOI 10.1137/20M1321802 (cited on p. 84).

[86] Brubeck, P.D., Kirby, R.C., Laakmann, F. and Mitchell, L. FIAT: Improving Performance and Accuracy for High-Order Finite Elements. *ACM Transactions on Mathematical Software*. [Online]. 2025, **51**(3), article no: 21 [no pagination]. [Accessed 7 March 2026]. Available from: DOI 10.1145/3748816 (cited on p. 84).

[87] Wren, C.R. *Perspective Transform Estimation*. [Online]. 1998. [Accessed 9 January 2025]. Available from: URL https://web.archive.org/web/20100801071311/http://alumni.media.mit.edu/%7Ecwren/interpolator/ (cited on p. 86).

[88] Reddy, J.N. *An Introduction to the Finite Element Method*. 3rd ed. New York: McGraw-Hill, 2006 (cited on p. 87).

[89] Quarteroni, A., Sacco, R. and Saleri, F. *Numerical Mathematics*. 2nd ed. Berlin: Springer, 2007 (cited on pp. 91, 133, 135, 139).

[90] Hughes, T.J.R., Franca, L.P. and Balestra, M. A New Finite Element Formulation for Computational Fluid Dynamics: V. Circumventing the Babuška-Brezzi Condition: a Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-Order Interpolations. *Computer Methods in Applied Mechanics and Engineering*. 1986, **59**(1), pp. 85–99. [Accessed 26 May 2026] (cited on p. 93).

[91] Hairer, E., Lubich, C. and Wanner, G. *Geometric Numerical Integration*. 2nd ed. Berlin: Springer, 2006 (cited on pp. 93–96).

[92] Arnold, V.I. *Mathematical Methods of Classical Mechanics*. 2nd ed. New York: Springer, 1989 (cited on p. 94).

[93] Abraham, R. and Marsden, J.E. *Foundations of Mechanics*. 2nd ed. Providence, RI: Americal Mathematical Society, 1978 (cited on p. 94).

[94] Ruth, R.D. A Canonical Integration Technique. *IEEE Transactions on Nuclear Science*. [Online]. 1983, **30**(4), pp. 2669–2671. [Accessed 16 January 2025]. Available from: DOI 10.1109/TNS.1983.4332919 (cited on p. 94).

[95] Forest, E. and Ruth, R.D. Fourth-Order Symplectic Integration. *Physica D: Nonlinear Phenomena*. [Online]. 1990, **43**(1), pp. 105–117. [Accessed 16 January 2025]. Available from: DOI 10.1016/0167-2789(90)90019-L (cited on p. 94).

[96] Yoshida, H. Construction of Higher Order Symplectic Integrators. *Physics Letters A*. [Online]. 1990, **150**(5–7), pp. 262–268. [Accessed 16 January 2025]. Available from: DOI 10.1016/0375-9601(90)90092-3 (cited on p. 94).

[97] Candy, J. and Rozmus, W. A Symplectic Integration Algorithm for Separable Hamiltonian Functions. *Journal of Computational Physics*. [Online]. 1991, **92**(1), pp. 230–256. [Accessed 16 January 2025]. Available from: DOI 10.1016/0021-9991(91)90299-Z (cited on p. 94).

[98] Stewart, R.H. *16.1: Linear Theory of Ocean Surface Waves*. [Online]. [no date]. [Accessed 27 November 2025]. Available from: URL https://geo.libretexts.org/Bookshelves/Oceanography/Introduction_to_Physical_Oceanography_%28Stewart%29/16%3A_Ocean_Waves/16.1%3A_Linear_Theory_of_Ocean_Surface_Waves (cited on p. 99).

[99] Alnaes, M.S., Logg, A., Ølgaard, K.B., Rognes, M.E. and Wells, G.N. Unified Form Language: A Domain-Specific Language for Weak Formulations of Partial Differential Equations. *ACM Transactions on Mathematical Software*. [Online]. 2014, **40**(2), article no: 9 [no pagination]. [Accessed 29 January 2025]. Available from: DOI 10.1145/2566630 (cited on p. 101).

[100] Balay, S., Abhyankar, S., Adams, M.F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W.D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M.G., Kong, F., Kruger, S., May, D.A., McInnes, L.C., Mills, R.T., Mitchell, L., Munson, T., Roman, J.E., Rupp, K., Sanan, P., Sarich, J., Smith, B.F., Suh, H., Zampini, S., Zhang, H., Zhang, H. and Zhang, J. *PETSc/TAO Users Manual*. [Online]. Argonne, IL: Argonne National Laboratory, 2025. [Accessed 2 April 2025]. Available from: DOI 10.2172/2476320 (cited on p. 101).

[101] *Solid (Continuum) Elements*. [Online]. [no date]. [Accessed 20 November 2025]. Available from: URL https://abaqus-docs.mit.edu/2017/English/SIMACAEELMRefMap/simaelm-c-solidcont.htm#simaelm-c-solidcont-t-elemchoice1 (cited on p. 122).

[102] *Introducing JSON*. [Online]. [no date]. [Accessed 8 October 2025]. Available from: URL https://www.json.org/ (cited on p. 133).

[103] *MUMPS: MUltifrontal Massively Parallel sparse direct Solver*. [Online]. [no date]. [Accessed 25 September 2025]. Available from: URL https://mumps-solver.org/ (cited on p. 134).

[104] Dongarra, J., Eijkhout, V. and Łuszczek, P. Recursive Approach in Sparse Matrix LU Factorization. *Scientific Programming*. [Online]. 2001, **9**(1), pp. 51–60. [Accessed 20 November 2025]. Available from: DOI 10.1155/2001/569670 (cited on p. 135).

[105] McKay, M.D., Beckman, R.J. and Conover, W.J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*. [Online]. 1979, **21**(2), pp. 239–245. [Accessed 10 March 2026]. Available from: DOI 10.2307/1268522 (cited on p. 143).

[106] Zhang, F. ed. *The Schur Complement and Its Applications*. New York: Springer, 2005 (cited on p. 149).

[107] Alexander, C.K. and Sadiku, M.N.O. *Fundamentals of Electric Circuits*. 6th ed. New York: McGraw-Hill, 2017 (cited on p. 152).

[108] Horowitz, P. and Hill, W. *The Art of Electronics*. 3rd ed. Cambridge: Cambridge University Press, 2015 (cited on p. 153).

[109] Griffiths, D.J. *Instructor's Solution Manual. Introduction to Electrodynamics*. [Online]. 4th ed. Cambridge: Cambridge University Press, 2017. [Accessed 20 November 2024]. Available from: URL https://koberstudios.com/docs/PHYS110.pdf (cited on p. 186).

Appendix A

Supplementary Mathematical Detail

A.1 Relating the unknown at-rest position of the buoy to known geometric parameters

Archimedes' principle states that the buoyant force applied to the buoy by the water is equal to the weight of the water displaced by the buoy. At rest, since the buoy is neutrally buoyant, this buoyant force is equal to the buoy's weight, such that the weights of the buoy and (displaced) water are equal; therefore, the masses of the buoy and water are also equal. But the mass of the water is equal to its density multiplied by the volume. Then, because the buoy is partially submerged, the volume of water displaced by the buoy is equal to the volume of the submerged section, such that the buoy's mass M is equal to the water density ρ_0 multiplied by the submerged volume V, i.e.

$$M = \rho_0 V. \tag{A.1}$$

Figure A.1 contains a sketch of the submerged portion of the buoy. This is a smaller tetrahedron—isomorphic to, and sharing a bottom vertex with, the entire buoy—with a top face at $z = H_0$. The waterline $y = L_b$ has a width equivalent to the width of

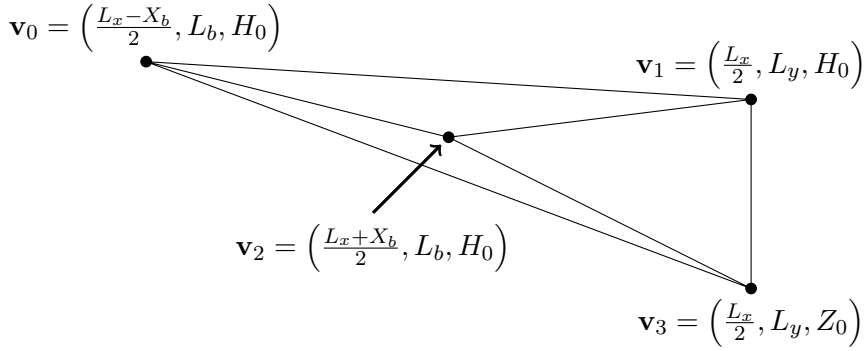


Figure A.1: The positions of the vertices $\mathbf{v}_0, \ldots, \mathbf{v}_3$ of the partially submerged portion of the buoy at rest.

the front-most part of this submerged section, which is denoted X_b. Therefore, the submerged tetrahedron has vertices

$$\mathbf{v}_0 = \left(\frac{L_x - X_b}{2}, L_b, H_0\right), \quad \mathbf{v}_1 = \left(\frac{L_x}{2}, L_y, H_0\right), \quad \mathbf{v}_2 = \left(\frac{L_x + X_b}{2}, L_b, H_0\right)$$
$$\text{and} \quad \mathbf{v}_3 = \left(\frac{L_x}{2}, L_y, Z_0\right).$$

Since a tetrahedron is a 3D simplex, its volume is given by

$$V \equiv \frac{1}{6} |\det(\mathbf{v}_1 - \mathbf{v}_0, \mathbf{v}_2 - \mathbf{v}_0, \mathbf{v}_3 - \mathbf{v}_0)|, \tag{A.2}$$

with

$$\begin{aligned} \mathbf{v}_1 - \mathbf{v}_0 = \left(\frac{X_b}{2}, L_y - L_b, 0\right), \quad \mathbf{v}_2 - \mathbf{v}_0 = (X_b, 0, 0) \\ \text{and} \quad \mathbf{v}_3 - \mathbf{v}_0 = \left(\frac{X_b}{2}, L_y - L_b, Z_0 - H_0\right). \end{aligned} \tag{A.3}$$

Now, rearranging (2.13) gives

$$L_y - L_b = \frac{H_0 - Z_0}{\tan\alpha}. \tag{A.4}$$

Then, considering one half of the top face of the tetrahedron, and given the contraction angle θ, we have

$$\frac{X_b}{2} = \frac{L_y - L_b}{\tan\theta} = \frac{H_0 - Z_0}{\tan\alpha \tan\theta}. \tag{A.5}$$

Therefore, using (A.4) and (A.5) in (A.3), we have

$$\mathbf{v}_1 - \mathbf{v}_0 = \left(\frac{H_0 - Z_0}{\tan\alpha\tan\theta}, \frac{H_0 - Z_0}{\tan\alpha}, 0\right), \quad \mathbf{v}_2 - \mathbf{v}_0 = \left(\frac{2(H_0 - Z_0)}{\tan\alpha\tan\theta}, 0, 0\right)$$
$$\text{and} \quad \mathbf{v}_3 - \mathbf{v}_0 = \left(\frac{H_0 - Z_0}{\tan\alpha\tan\theta}, \frac{H_0 - Z_0}{\tan\alpha}, Z_0 - H_0\right),$$

and (A.2) gives

$$\begin{aligned} V &= \frac{(H_0 - Z_0)^3}{6}\left|\det\begin{pmatrix} \frac{1}{\tan\alpha\tan\theta} & \frac{1}{\tan\alpha} & 0 \\ \frac{2}{\tan\alpha\tan\theta} & 0 & 0 \\ \frac{1}{\tan\alpha\tan\theta} & \frac{1}{\tan\alpha} & -1 \end{pmatrix}\right| \\ &= \frac{(H_0 - Z_0)^3}{6}\left|\det\begin{pmatrix} \frac{1}{\tan\alpha\tan\theta} & \frac{1}{\tan\alpha} \\ \frac{2}{\tan\alpha\tan\theta} & 0 \end{pmatrix}\right| \\ &= \frac{(H_0 - Z_0)^3}{3\tan^2\alpha\tan\theta}. \end{aligned}$$

Therefore, using (A.1) we find that

$$M = \frac{\rho_0(H_0 - Z_0)^3}{3\tan^2\alpha\tan\theta},$$

and rearranging for Z_0 gives

$$Z_0 = H_0 - \left(\frac{3M\tan^2\alpha\tan\theta}{\rho_0}\right)^{1/3}.$$

A.2 Deriving Kirchhoff's law for the voltage induced by the moving buoy

Relative motion between a magnet and a coil causes the strength of the magnetic field (or, more accurately, magnetic flux density) passing through the coil to change over time. This changing field produces a changing magnetic flux (defined as the total flux density passing through a given area) in the coil. Faraday's law states that this change in flux induces a voltage in the coil, which, by Ohm's law, causes an electric current to travel around the circuit. This process is well known as *electromagnetic induction*.

Simultaneously, the current travelling through the coil generates a secondary magnetic field, due to Ampère's circuital law. As the magnet continues to move, the strength of the induced current changes, causing this secondary field and its associated flux to change. This changing secondary flux produces a secondary voltage, which acts to "slow" the current (Lenz's law states that this current acts to oppose the original current; otherwise, the total current would be infinitely increasing). This phenomenon is known as *self-induction*.

This process can be described mathematically, starting with the vector form of Ohm's law:

$$\mathbf{J} = \sigma(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \text{ (e.g. [71, p. 296])}, \tag{A.6}$$

where $\mathbf{J}$ is the current density in the wire (current per cross-sectional area of wire in the direction of the flow), σ is the conductivity of the wire, $\mathbf{E}$ is the electric field generated by the current (which is related to the secondary magnetic field it generates), $\mathbf{v}$ is the velocity of the magnet and $\mathbf{B}$ is the magnetic field of the magnet (evaluated at the coil). Integrating (A.6) along the entire length of wire C and dividing by the conductivity gives

$$\frac{1}{\sigma} \oint_C \mathbf{J} \cdot \mathrm{d}\mathbf{l} = \oint_C \mathbf{E} \cdot \mathrm{d}\mathbf{l} + \oint_C (\mathbf{v} \times \mathbf{B}) \cdot \mathrm{d}\mathbf{l}. \tag{A.7}$$

We now consider (A.7) term-by-term.

A.2.1 Resistance of the coil

The term on the left of (A.7) corresponds to the voltage in the coil due to the flow of current and the resistance in the wire. Since $\mathbf{J}$ and $\mathrm{d}\mathbf{l}$ are co-linear and $\mathbf{J}$ is uniform along the wire, the integral reduces to the magnitude of $\mathbf{J}$ multiplied by the length of the wire, or

$$\frac{1}{\sigma} \oint_C \mathbf{J} \cdot \mathrm{d}\mathbf{l} = \frac{1}{\sigma} \frac{I}{\pi D^2/4} (2\pi a n_c) = I R_i, \tag{A.8}$$

with

$$R_i \equiv \frac{8 a n_c}{\sigma D^2}$$

the resistance of the coil. This expression is equivalent to the well-known formula for the resistance of a length of wire: length divided by cross-sectional area divided by conductivity.

A.2.2 Self-induced voltage in the coil

The first term on the right of (A.7) corresponds to the self-induced (secondary) voltage generated by the changing current. For a single turn of the coil, the electric field $\mathbf{E}$ and the secondary magnetic flux Φ are related by Faraday's law:

$$\oint_{1 \text{ turn}} \mathbf{E} \cdot \mathrm{d}\mathbf{l} = -\dot{\Phi} = -L_1 \dot{I} \text{ (e.g. [71, p. 313])}.$$

The second equality uses the fact that the flux is directly proportional to the current, with constant of proportionality known as the self-inductance and denoted L_1, i.e. $\Phi = L_1 I$ (e.g. [71, p. 324]). For n_c turns, the integral becomes

$$\oint_C \mathbf{E} \cdot \mathrm{d}\mathbf{l} = -n_c \dot{\Phi} = -L_i \dot{I}, \tag{A.9}$$

with $L_i = n_c L_1 = n_c \Phi / I$ (i.e. the inductance of the entire coil is just the sum of the inductances of each turn). Thus, the self-induced voltage is simply $-L_i \dot{I}$, in which the negative sign is clearly seen to counteract an increase in current. The constant L_i depends entirely on the geometry of the coil, and is calculated as follows. First, the magnetic field inside a "long" coil is

$$B = \frac{\mu_0 n_c I}{L} \text{ (e.g. [71, p. 237])}.$$

Thus, since the flux is just the field multiplied by the cross-sectional area of the coil (*not of the wire*), we have, for a "long" coil,

$$\Phi = \frac{\mu_0 n_c I}{L} (\pi a^2),$$

and therefore

$$L_i = \frac{\mu_0 \pi a^2 n_c^2}{L} \text{ (e.g. problem 7.24 of [71, p. 327], solution in [109, p. 151]).}$$

A "short" coil's inductance is reduced by a factor of $0 < K \leq 1$, with Table 8.1 of [65, p. 347] giving values of K for a range of radius-to-length ratios, with $K \to 1$ for $a/L \to 0$.

A.2.3 Voltage induced by the moving magnet

The final term in (A.7) gives the voltage induced by the moving magnet (e.g. [71, p. 305]), which depends heavily upon the geometry of the electromagnet. The integration can be carried out first for a single turn, and then for the entire n_c-turn coil. Using cylindrical coordinates (r, ϕ, z), the orientation of the coil is such that the line element of each turn is in the ϕ direction only, so $\mathrm{d}\mathbf{l} = \mathrm{d}l\hat{\boldsymbol{\phi}} = a\,\mathrm{d}\phi\hat{\boldsymbol{\phi}}$, recalling that $r = a$ along the coil. Also, since the magnet is attached to the buoy which moves with velocity $\dot{Z}\hat{\mathbf{z}}$, we have $\mathbf{v} = \dot{Z}\hat{\mathbf{z}}$ and $\mathbf{v} \times \mathbf{B} = \dot{Z}(B_r\hat{\boldsymbol{\phi}} - B_\phi\hat{\mathbf{r}})$, with B_r and B_ϕ the radial and azimuthal components of $\mathbf{B}$. Thus, the integral over one turn reduces to

$$\oint_{1\text{ turn}} (\mathbf{v} \times \mathbf{B}) \cdot \mathrm{d}\mathbf{l} = \int_0^{2\pi} \dot{Z} B_r|_{r=a} a\,\mathrm{d}\phi.$$

Then, the evaluation of the integral over the whole coil simply requires integrating the above along the coil's length (since the magnetic field depends on z) and multiplying by the turn density n_c/L:

$$\begin{aligned}\oint_C (\mathbf{v} \times \mathbf{B}) \cdot \mathrm{d}\mathbf{l} &= \frac{n_c}{L} \int_{Z_0+(1+\alpha_h)H_m-L/2}^{Z_0+(1+\alpha_h)H_m+L/2} \int_0^{2\pi} \dot{Z} B_r|_{r=a} a\,\mathrm{d}\phi\,\mathrm{d}z \\ &= \frac{n_c}{L} \int_{-\frac{L}{2}}^{\frac{L}{2}} \int_0^{2\pi} \dot{Z} B_r|_{r=a} a\,\mathrm{d}\phi\,\mathrm{d}\bar{z}, \end{aligned} \tag{A.10}$$

after making the substitution $z = Z_0 + (1 + \alpha_h)H_m + \bar{z}$. The result of this integral depends on the precise form of B_r.

A.2.4 The magnetic field of a cylindrical magnet

First, $\mathbf{B}$ is related to the field *intensity* $\mathbf{H}$ and magnetisation $\mathbf{M}$ (defined as the magnetic moment per unit volume, e.g. [71, p. 273]) by

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) \text{ (e.g. [71, p. 280])}, \tag{A.11}$$

while the field intensity is related to the current density by $\nabla \times \mathbf{H} = \mathbf{J}$ (e.g. [71, p. 280]). However, no current flows through the magnet, so $\mathbf{J} = 0$ and $\nabla \times \mathbf{H} = 0$. In other words, $\mathbf{H}$ is irrotational, such that it can be written as the gradient of some function Ψ, known as the magnetic scalar potential, i.e. $\mathbf{H} = \nabla\Psi$.

Using Gauss's law for magnetism—i.e. $\nabla \cdot \mathbf{B} = 0$ for *any* magnetic field (e.g. [71, p. 332])—taking the divergence of (A.11) gives $\nabla \cdot \mathbf{M} = -\nabla \cdot \mathbf{H} = -\nabla^2\Psi$. Defining the divergence of $\mathbf{M}$ to be the magnetic charge density $\rho_m(\mathbf{r})$ (where $\mathbf{r}$ is the global position vector), we then have the Poisson equation $\nabla^2\Psi = -\rho_m$, which has an integral solution via Green's function of

$$\Psi(\mathbf{r}) = \int_{-\infty}^{\infty}\int_{0}^{2\pi}\int_{0}^{\infty} \frac{\rho_m(\tilde{\mathbf{r}})}{4\pi\|\mathbf{r} - \tilde{\mathbf{r}}\|}\tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}\,\mathrm{d}\tilde{z}, \tag{A.12}$$

where $\tilde{\mathbf{r}}$ is the local position vector, measured from the magnet's centre. The integral is over all space, but the form of ρ_m found next will limit the integral to just the top and bottom faces of the magnet. We also need to write down $\|\mathbf{r} - \tilde{\mathbf{r}}\|$.

The magnetisation of a cylindrical magnet pointed in the z-direction is $\mathbf{M} = M(\tilde{r}, \tilde{z})\hat{\mathbf{z}}$, giving $\rho_m(\tilde{\mathbf{r}}) = \rho_m(\tilde{r}, \tilde{z}) = \partial M/\partial\tilde{z}$. Given a (constant) magnetic moment m and volume $\pi A_m^2 L_m$, we have

$$\begin{aligned} M &= \begin{cases} \frac{m}{\pi A_m^2 L_m} & \tilde{r} \in [0, A_m],\ \ \tilde{z} \in [-L_m/2, L_m/2] \text{ (inside the magnet)} \\ 0 & \text{otherwise} \end{cases} \\ &= \frac{m}{\pi A_m^2 L_m}\Big(\Theta(\tilde{r}) - \Theta(\tilde{r} - A_m)\Big)\Big(\Theta(\tilde{z} + L_m/2) - \Theta(\tilde{z} - L_m/2)\Big), \end{aligned}$$

and therefore

$$\begin{aligned}\rho_m &= \frac{m}{\pi A_m^2 L_m}\Big(\Theta(\tilde{r}) - \Theta(\tilde{r} - A_m)\Big)\frac{\partial}{\partial \tilde{z}}\Big(\Theta(\tilde{z} + {}^{L_m}\!/_2) - \Theta(\tilde{z} - {}^{L_m}\!/_2)\Big) \\ &= \frac{m}{\pi A_m^2 L_m}\Big(\Theta(\tilde{r}) - \Theta(\tilde{r} - A_m)\Big)\Big(\delta(\tilde{z} + {}^{L_m}\!/_2) - \delta(\tilde{z} - {}^{L_m}\!/_2)\Big) \\ &= \begin{cases} \frac{m}{\pi A_m^2 L_m}\Big(\delta(\tilde{z} + {}^{L_m}\!/_2) - \delta(\tilde{z} - {}^{L_m}\!/_2)\Big) & \tilde{r} \in [0, A_m] \\ 0 & \text{otherwise,} \end{cases}\end{aligned} \tag{A.13}$$

with δ the Dirac delta function. Next, $\mathbf{r}$ and $\tilde{\mathbf{r}}$ can be written in Cartesian basis vectors as

$$\begin{aligned}\mathbf{r} &= r\cos\phi\hat{\mathbf{x}} + r\sin\phi\hat{\mathbf{y}} + (Z_0 + \alpha_h H_m + \bar{z} - Z)\hat{\mathbf{z}}; \\ \tilde{\mathbf{r}} &= \tilde{r}\cos\tilde{\phi}\hat{\mathbf{x}} + \tilde{r}\sin\tilde{\phi}\hat{\mathbf{y}} + \tilde{z}\hat{\mathbf{z}},\end{aligned} \tag{A.14}$$

where, for the global position $\mathbf{r}$, the height of the centre of the magnet has been taken into account (i.e. $r_z = z - Z - H_m$, before the substitution).

For a cylindrical magnet, $\mathbf{B}$ is axisymmetric (so does not depend on ϕ). Then, since $\mathbf{M}$ does not depend on ϕ either, $\mathbf{H}$ and thus Ψ are also axisymmetric. This means ϕ is arbitrary, so we choose $\phi = 0$ for simplicity. Then, using (A.14), we have

$$\begin{aligned}\|\mathbf{r} - \tilde{\mathbf{r}}\| &= \Big((r - \tilde{r}\cos\tilde{\phi})^2 + \tilde{r}^2\sin^2\tilde{\phi} + (Z_0 + \alpha_h H_m + \bar{z} - Z - \tilde{z})^2\Big)^{1/2} \\ &= \Big(r^2 - 2r\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (Z_0 + \alpha_h H_m + \bar{z} - Z - \tilde{z})^2\Big)^{1/2},\end{aligned} \tag{A.15}$$

and, using (A.13) and (A.15), (A.12) becomes

$$\Psi(r, \bar{z}, Z) = \frac{m}{4\pi^2 A_m^2 L_m}\int_{-\infty}^{\infty}\int_0^{2\pi}\int_0^{A_m}\frac{\delta(\tilde{z} + {}^{L_m}\!/_2) - \delta(\tilde{z} - {}^{L_m}\!/_2)}{\Big(r^2 - 2r\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (\mathcal{Z} - \tilde{z})^2\Big)^{1/2}}\tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}\,\mathrm{d}\tilde{z} \tag{A.16}$$

$$= \frac{m}{4\pi^2 A_m^2 L_m}\int_0^{2\pi}\int_0^{A_m}\Big\{\Big(r^2 - 2r\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (\mathcal{Z} + {}^{L_m}\!/_2)^2\Big)^{-1/2} - \Big(r^2 - 2r\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (\mathcal{Z} - {}^{L_m}\!/_2)^2\Big)^{-1/2}\Big\}\tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}, \tag{A.17}$$

where the substitution $\mathcal{Z} = \bar{z} + Z_0 + \alpha_h H_m - Z$ has been applied to shorten the notation somewhat.

As mentioned previously, the solution for ρ_m in (A.13) limits the volume integral over all space in (A.12) to just two surface integrals over the top and bottom faces of the magnet: for $\tilde{r} \in [0, A_m]$ at $\tilde{z} = \pm L_m/2$. In (A.16), the upper bound on the radius has been correctly limited, while in (A.17) the evaluation of the integral of the delta function has isolated the required values of $\tilde{z}$.

Finally, outside of the magnet (where $\mathbf{M} = 0$), (A.11) reduces to $\mathbf{B} = \mu_0 \mathbf{H} = \mu_0 \nabla \Psi$, such that the radial component of the magnetic field is given by

$$\begin{aligned} B_r &= \mu_0 \frac{\partial \Psi}{\partial r} \\ &= \frac{\mu_0 m}{4\pi^2 A_m^2 L_m} \int_0^{2\pi} \int_0^{A_m} (\tilde{r}\cos\tilde{\phi} - r) \Big\{ \left(r^2 - 2r\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (\mathcal{Z} + {}^{L_m}\!/_2)^2 \right)^{-3/2} \\ &\qquad - \left(r^2 - 2r\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (\mathcal{Z} - {}^{L_m}\!/_2)^2 \right)^{-3/2} \Big\} \tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}. \end{aligned} \tag{A.18}$$

Far away from the magnet (where $\tilde{r} \ll r$ and $L_m \ll |\mathcal{Z}|$), this can be simplified significantly. First, the terms involving $\tilde{r}$ can be neglected, since $\tilde{r}\cos\tilde{\phi} \ll r$ and $\tilde{r}^2 \ll 2r\tilde{r}\cos\tilde{\phi} \ll r^2$. Then, we can multiply and divide by $\left(r^2 + \mathcal{Z}^2\right)^{3/2}$ and expand the quadratics in $\mathcal{Z}$ and $L_m/2$ to obtain

$$\begin{aligned} B_r \approx \frac{\mu_0 m r}{4\pi^2 A_m^2 L_m \left(r^2 + \mathcal{Z}^2\right)^{3/2}} \int_0^{2\pi} \int_0^{A_m} \Bigg\{ & \left(1 + \frac{{}^{L_m^2}\!/_4 - \mathcal{Z}L_m}{r^2 + \mathcal{Z}^2} \right)^{-3/2} \\ & - \left(1 + \frac{{}^{L_m^2}\!/_4 + \mathcal{Z}L_m}{r^2 + \mathcal{Z}^2} \right)^{-3/2} \Bigg\} \tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}. \end{aligned}$$

The L_m^2 terms can be neglected, since $L_m^2 \ll \mathcal{Z}L_m$, leaving terms of the form $(1 \pm x)^{-3/2}$ with $x = \mathcal{Z}L_m/(r^2 + \mathcal{Z}^2)$. Since $\mathcal{Z}L_m \ll \mathcal{Z}^2$, we have $x \ll 1$ and

$$(1 - x)^{-3/2} - (1 + x)^{-3/2} \approx (1 + {}^{3x}\!/_2) - (1 - {}^{3x}\!/_2) = 3x = \frac{3\mathcal{Z}L_m}{r^2 + \mathcal{Z}^2}.$$

Thus we have

$$B_r \approx \frac{\mu_0 m r}{4\pi^2 A_m^2 L_m \left(r^2 + \mathcal{Z}^2\right)^{3/2}} \int_0^{2\pi} \int_0^{A_m} \frac{3\mathcal{Z}L_m}{r^2 + \mathcal{Z}^2} \tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi},$$

which, after evaluating the integrals and replacing $\mathcal{Z}$, gives

$$B_r \approx \frac{3\mu_0 mr(Z_0 + \alpha_h H_m + \bar{z} - Z)}{4\pi\left(r^2 + (Z_0 + \alpha_h H_m + \bar{z} - Z)^2\right)^{5/2}}. \tag{A.19}$$

This relatively simple expression (at least, compared to (A.18)) can alternatively be found by using the equation for the field of a magnetic dipole,

$$\mathbf{B}(\mathbf{r}) \approx \frac{\mu_0}{4\pi\|\mathbf{r}\|^3}\{3(\mathbf{m}\cdot\hat{\mathbf{r}})\hat{\mathbf{r}} - \mathbf{m}\} \text{ (e.g. [71, p. 255] and derived in Appendix A.3),}$$

with $\mathbf{m} = m\hat{\mathbf{z}}$ and $\mathbf{r} = r\hat{\mathbf{r}} + (\bar{z} + Z_0 + \alpha_h H_m - Z)\hat{\mathbf{z}}$. This is because, far away from *any* magnetic material, the field resembles that of a magnetic dipole.

A.2.5 Kirchoff's law

Tying everything together, using (A.18) or it's approximation (A.19), the voltage induced by the moving magnet given by (A.10) can be calculated:

$$\oint_C (\mathbf{v}\times\mathbf{B})\cdot \mathrm{d}\mathbf{l} = \gamma G\dot{Z}, \tag{A.20}$$

with constant γ given by

$$\gamma = \frac{\mu_0 m a^2 n_c}{2L}$$

and function $G(Z)$ by

$$\begin{aligned} G &\equiv \frac{4\pi}{\mu_0 m a}\int_{-\frac{L}{2}}^{\frac{L}{2}} B_r|_{r=a}\,\mathrm{d}\bar{z} \\ &= \frac{1}{\pi a A_m^2 L_m}\int_{-\frac{L}{2}}^{\frac{L}{2}}\int_0^{2\pi}\int_0^{A_m}(\tilde{r}\cos\tilde{\phi} - a)(\mathcal{G}^+ - \mathcal{G}^-)\tilde{r}\,\mathrm{d}\tilde{r}\,\mathrm{d}\tilde{\phi}\,\mathrm{d}\bar{z} \qquad \text{(A.21)} \\ &\approx \int_{-\frac{L}{2}}^{\frac{L}{2}} \frac{3(Z_0 + \alpha_h H_m + \bar{z} - Z)}{\left(a^2 + (Z_0 + \alpha_h H_m + \bar{z} - Z)^2\right)^{5/2}}\,\mathrm{d}\bar{z} \\ &= \left(a^2 + (Z_0 + \alpha_h H_m - {}^L/_2 - Z)^2\right)^{-3/2} - \left(a^2 + (Z_0 + \alpha_h H_m + {}^L/_2 - Z)^2\right)^{-3/2}, \qquad \text{(A.22)} \end{aligned}$$

where

$$\mathcal{G}^{\pm} = \left(a^2 - 2a\tilde{r}\cos\tilde{\phi} + \tilde{r}^2 + (Z_0 + \alpha_h H_m + \bar{z} - Z \pm {}^{L_m}/_2)^2\right)^{-3/2}.$$

(A.21) is the result of using the exact expression for the magnetic field (A.18), while (A.22) is the far-field approximation found when using (A.19).

We are now in a position to write down (A.7) in a form which governs the rate-of-change of the current I: using (A.8), (A.9) and (A.20), we find

$$IR_i = -L_i\dot{I} + \gamma G\dot{Z}.$$

If the coil forms part of a circuit, whose remaining components have a total resistance of R_c, the LHS is simply changed to $I(R_i + R_c)$.

A.3 The field of a magnetic dipole

The constant *magnetic moment* $\mathbf{m}$ of a magnet gives a measure of its strength and orientation (which, for a cylindrical magnet, is along its axis of symmetry), and it is possible to write the magnetic field as a function of $\mathbf{m}$ and position $\mathbf{r}$.

First, we use the fact that a cylindrical magnet with length L and *magnetisation* M generates the same field as a coil of identical dimensions whose current density $(n_c I)/L$ matches M [65, p. 393]. I is the current and n_c/L denotes the *turn density* (turns-per-meter) of the coil, while M is related to $\mathbf{m}$ by

$$\mathbf{m} = m\hat{\mathbf{n}} = \iiint_{\text{magnet}} M\hat{\mathbf{n}}\,\mathrm{d}V',$$

where $\hat{\mathbf{n}}$ points along the magnet's axis of symmetry and is thus normal to the coil's turns. Thus, one can arrive at an expression relating the moment of the magnet to the current in the coil:

$$\mathbf{m} = \iiint_{\text{magnet}} M\hat{\mathbf{n}}\,\mathrm{d}V' = \iiint_{n_c \text{ turns}} \frac{n_c I}{L}\hat{\mathbf{n}}\,\mathrm{d}V' = \iiint_{1 \text{ turn}} \frac{I}{L}\hat{\mathbf{n}}\,\mathrm{d}V' = I\iint_{S'} \hat{\mathbf{n}}\,\mathrm{d}S' \quad \text{(e.g. [71, p. 253])}. \tag{A.23}$$

Next, the well-known Biot–Savart law gives the field produced by a current travelling

along a circuit centred at the origin:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0 I}{4\pi} \oint_{C'} \frac{\mathrm{d}\mathbf{l}' \times \hat{\mathbf{R}}}{\|\mathbf{R}\|^2} = -\frac{\mu_0 I}{4\pi} \oint_{C'} \frac{\hat{\mathbf{R}}}{\|\mathbf{R}\|^2} \times \mathrm{d}\mathbf{l}' \quad \text{(e.g. [71, p. 224])}, \tag{A.24}$$

where the infinitesimal length element of the wire $\mathrm{d}\mathbf{l}'$ is centred at $\mathbf{r}'$, and $\mathbf{R} = \mathbf{r} - \mathbf{r}'$ is the vector from the wire to the point $\mathbf{r}$. For clarity, primed coordinates represent the current-carrying space and un-primed coordinates represent the space in which the magnetic field is felt; in general, $f(\nabla, \mathbf{r}') = f(\nabla', \mathbf{r}) = 0$, while functions of $\mathbf{r}$ can be taken out of integrals over $\mathbf{r}'$ (and vice-versa).

Now, making use of (A.27) with $n = 1$, (A.24) can be written as

$$\mathbf{B} = \frac{\mu_0 I}{4\pi} \oint_{C'} \nabla \left(\frac{1}{\|\mathbf{r} - \mathbf{r}'\|} \right) \times \mathrm{d}\mathbf{l}',$$

which, by the product rule, becomes

$$\begin{aligned} \mathbf{B} &= \frac{\mu_0 I}{4\pi} \oint_{C'} \left\{ \nabla \times \left(\frac{\mathrm{d}\mathbf{l}'}{\|\mathbf{r} - \mathbf{r}'\|} \right) - \cancel{\frac{\nabla \times \mathrm{d}\mathbf{l}'}{\|\mathbf{r} - \mathbf{r}'\|}} \right\} \\ &= \nabla \times \left\{ \frac{\mu_0 I}{4\pi} \oint_{C'} \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} \, \mathrm{d}\mathbf{l}' \right\}. \end{aligned} \tag{A.25}$$

Then, since Gauss's law for magnetism gives $\nabla \cdot \mathbf{B} = 0$, we can define the magnetic vector potential $\mathbf{A}$ as satisfying $\mathbf{B} = \nabla \times \mathbf{A}$, and one can immediately use (A.25) to find

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0 I}{4\pi} \oint_{C'} \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} \, \mathrm{d}\mathbf{l}'.$$

Next, the denominator $\|\mathbf{r} - \mathbf{r}'\|$ can be expanded about $\mathbf{r}$, making use of (A.26) with $n = 1$, with only the leading-order nonzero term retained:

$$\begin{aligned} \mathbf{A} &= \frac{\mu_0 I}{4\pi} \oint_{C'} \left\{ \frac{1}{\|\mathbf{r}\|} - \nabla \left(\frac{1}{\|\mathbf{r}\|} \right) \cdot \mathbf{r}' + \cancel{\mathcal{O}\left(\mathbf{r}'^2\right)} \right\} \mathrm{d}\mathbf{l}' \\ &= \frac{\mu_0 I}{4\pi} \left\{ \frac{1}{\|\mathbf{r}\|} \cancel{\oint_{C'} \mathrm{d}\mathbf{l}'} + \oint_{C'} \frac{\mathbf{r} \cdot \mathbf{r}'}{\|\mathbf{r}\|^3} \, \mathrm{d}\mathbf{l}' \right\} \end{aligned}$$

$$= \frac{\mu_0 I}{4\pi\|\mathbf{r}\|^3} \oint_{C'} (\mathbf{r}\cdot\mathbf{r}')\,\mathrm{d}\mathbf{l}',$$

where the first term vanishes since C' is a closed loop. This is the *multipole expansion* (see [71, 252–253] for more details), and its result is that the magnetic field of a coil is roughly equal to the field of a magnetic dipole.

Starting with Stokes' theorem on the vector $(\mathbf{r}\cdot\mathbf{r}')\mathbf{r}$, and applying various vector algebra and calculus identities, we find

$$\begin{aligned}
\mathbf{r}\cdot\oint_{C'}(\mathbf{r}\cdot\mathbf{r}')\,\mathrm{d}\mathbf{l}' &= \oint_{C'}((\mathbf{r}\cdot\mathbf{r}')\mathbf{r})\cdot\mathrm{d}\mathbf{l}' \\
&= \iint_{S'}\{\nabla'\times((\mathbf{r}\cdot\mathbf{r}')\mathbf{r})\}\cdot\hat{\mathbf{n}}\,\mathrm{d}S' \\
&= \iint_{S'}\{(\mathbf{r}\cdot\mathbf{r}')\cancel{\nabla'\times\mathbf{r}} + \nabla'(\mathbf{r}\cdot\mathbf{r}')\times\mathbf{r}\}\cdot\hat{\mathbf{n}}\,\mathrm{d}S' \\
&= -\iint_{S'}\{\mathbf{r}\times\nabla'(\mathbf{r}\cdot\mathbf{r}')\}\cdot\hat{\mathbf{n}}\,\mathrm{d}S' \\
&= -\iint_{S'}\mathbf{r}\cdot\{\nabla'(\mathbf{r}\cdot\mathbf{r}')\times\hat{\mathbf{n}}\,\mathrm{d}S'\} \\
&= -\mathbf{r}\cdot\iint_{S'}\nabla'(\mathbf{r}\cdot\mathbf{r}')\times\hat{\mathbf{n}}\,\mathrm{d}S'; \\
\Rightarrow \oint_{C'}(\mathbf{r}\cdot\mathbf{r}')\,\mathrm{d}\mathbf{l}' &= -\iint_{S'}\nabla'(\mathbf{r}\cdot\mathbf{r}')\times\hat{\mathbf{n}}\,\mathrm{d}S',
\end{aligned}$$

and then

$$\begin{aligned}
\mathbf{A} &= \frac{\mu_0 I}{4\pi\|\mathbf{r}\|^3}\oint_{C'}(\mathbf{r}\cdot\mathbf{r}')\,\mathrm{d}\mathbf{l}' \\
&= -\frac{\mu_0 I}{4\pi\|\mathbf{r}\|^3}\iint_{S'}\nabla'(\mathbf{r}\cdot\mathbf{r}')\times\hat{\mathbf{n}}\,\mathrm{d}S' \\
&= -\frac{\mu_0 I}{4\pi\|\mathbf{r}\|^3}\iint_{S'}\left\{\mathbf{r}\cdot\cancel{\nabla'\mathbf{r}'}^{\mathbf{I}} + \mathbf{r}'\cdot\cancel{\nabla'\mathbf{r}}^{0} + \mathbf{r}\times\cancel{(\nabla'\times\mathbf{r}')}^{0} + \mathbf{r}'\times\cancel{(\nabla'\times\mathbf{r})}^{0}\right\}\times\hat{\mathbf{n}}\,\mathrm{d}S' \\
&= -\frac{\mu_0 I}{4\pi\|\mathbf{r}\|^3}\iint_{S'}\mathbf{r}\times\hat{\mathbf{n}}\,\mathrm{d}S' \\
&= \frac{\mu_0}{4\pi}\left(I\iint_{S'}\hat{\mathbf{n}}\,\mathrm{d}S'\right)\times\frac{\mathbf{r}}{\|\mathbf{r}\|^3}.
\end{aligned}$$

By (A.23), the term in brackets is $\mathbf{m}$ and we have

$$\mathbf{A} = \frac{\mu_0}{4\pi}\frac{\mathbf{m}\times\mathbf{r}}{\|\mathbf{r}\|^3}.$$

Finally, obtaining $\mathbf{B}$ simply requires some more vector calculus, and repeated use of (A.26) with $n = 3$:

$$\begin{aligned}
\mathbf{B} &= \nabla\times\left(\frac{\mu_0}{4\pi}\frac{\mathbf{m}\times\mathbf{r}}{\|\mathbf{r}\|^3}\right)\\
&= \frac{\mu_0}{4\pi}\left\{\mathbf{m}\nabla\cdot\left(\frac{\mathbf{r}}{\|\mathbf{r}\|^3}\right) - \frac{\mathbf{r}}{\|\mathbf{r}\|^3}\cancel{\nabla\cdot\mathbf{m}} + \frac{\mathbf{r}}{\|\mathbf{r}\|^3}\cdot\cancel{\nabla\mathbf{m}} - \mathbf{m}\cdot\nabla\left(\frac{\mathbf{r}}{\|\mathbf{r}\|^3}\right)\right\}\\
&= \frac{\mu_0}{4\pi}\left\{\mathbf{m}\left[\frac{\cancelto{3}{\nabla\cdot\mathbf{r}}}{\|\mathbf{r}\|^3} + \nabla\left(\frac{1}{\|\mathbf{r}\|^3}\right)\cdot\mathbf{r}\right] - \mathbf{m}\cdot\left[\frac{\cancelto{\mathbf{I}}{\nabla\mathbf{r}}}{\|\mathbf{r}\|^3} + \nabla\left(\frac{1}{\|\mathbf{r}\|^3}\right)\otimes\mathbf{r}\right]\right\}\\
&= \frac{\mu_0}{4\pi}\left\{\mathbf{m}\left(\frac{3}{\|\mathbf{r}\|^3} - \frac{3\mathbf{r}}{\|\mathbf{r}\|^5}\cdot\mathbf{r}\right) - \mathbf{m}\cdot\left(\frac{\mathbf{I}}{\|\mathbf{r}\|^3} - \frac{3\mathbf{r}}{\|\mathbf{r}\|^5}\otimes\mathbf{r}\right)\right\}\\
&= \frac{\mu_0}{4\pi\|\mathbf{r}\|^5}\{3\mathbf{m}(\cancel{\|\mathbf{r}\|^2 - \mathbf{r}\cdot\mathbf{r}}) + \mathbf{m}\cdot(3\mathbf{r}\otimes\mathbf{r} - \|\mathbf{r}\|^2\mathbf{I})\}\\
&= \frac{\mu_0}{4\pi\|\mathbf{r}\|^5}\{3(\mathbf{m}\cdot\mathbf{r})\mathbf{r} - \|\mathbf{r}\|^2\mathbf{m}\cdot\mathbf{I}\}\\
&= \frac{\mu_0}{4\pi\|\mathbf{r}\|^3}\{3(\mathbf{m}\cdot\hat{\mathbf{r}})\hat{\mathbf{r}} - \mathbf{m}\}.
\end{aligned}$$

A.3.1 Gradient of the reciprocal of the Euclidean norm taken to an arbitrary power

The identity

$$\nabla\left(\frac{1}{\|\mathbf{r}\|^n}\right) = -\frac{n\mathbf{r}}{\|\mathbf{r}\|^{n+2}} = -\frac{n\hat{\mathbf{r}}}{\|\mathbf{r}\|^{n+1}} \tag{A.26}$$

is used throughout this section, and is proved as follows:

$$\begin{aligned}
\nabla\left(\frac{1}{\|\mathbf{r}\|^n}\right) &= \nabla\left((\mathbf{r}\cdot\mathbf{r})^{-n/2}\right)\\
&= -n\frac{\nabla(\mathbf{r}\cdot\mathbf{r})}{2}(\mathbf{r}\cdot\mathbf{r})^{-(n+2)/2}\\
&= -\frac{n\mathbf{r}\cdot\cancelto{\mathbf{I}}{\nabla\mathbf{r}}}{\|\mathbf{r}\|^{n+2}}\\
&= -\frac{n\mathbf{r}}{\|\mathbf{r}\|^{n+2}}\\
&= -\frac{n\hat{\mathbf{r}}}{\|\mathbf{r}\|^{n+1}}.
\end{aligned}$$

When applied to $\mathbf{R} = \mathbf{r} - \mathbf{r}'$, the proof proceeds in the same way, and only differs upon the evaluation of $\nabla\mathbf{R}$. Since $\nabla\mathbf{r}' = \nabla'\mathbf{r} = 0$, we have $\nabla\mathbf{R} = \nabla\mathbf{r} = \mathbf{I}$ and $\nabla'\mathbf{R} = -\nabla'\mathbf{r}' = -\mathbf{I}$, giving

$$\nabla\left(\frac{1}{\|\mathbf{R}\|^n}\right) = -\frac{n\hat{\mathbf{R}}}{\|\mathbf{R}\|^{n+1}} \quad \text{and} \quad \nabla'\left(\frac{1}{\|\mathbf{R}\|^n}\right) = \frac{n\hat{\mathbf{R}}}{\|\mathbf{R}\|^{n+1}}. \tag{A.27}$$

A.4 Legendre transform

In this section, the precise Lagrangian used by Bokhove et al. [44, 45] is derived, from our formulation (2.29). The hydrodynamic terms remain the same, and we focus on the buoy and electromagnetic terms (2.15) and (2.25) only, with the capacitor potential dropped:

$$\mathcal{L}_{b\&e} = \mathcal{L}_b + \mathcal{L}_e = \frac{1}{2}M\dot{Z}^2 - MgZ + \frac{1}{2}L_i\dot{Q}^2 + \gamma G\dot{Z}Q. \tag{A.28}$$

We shall first obtain the Hamiltonian, via a Legendre transform as outlined in Section 2.1.3, and then use this to rewrite an equivalent Lagrangian.

To facilitate the transform, we rewrite the final term of (A.28) using the product rule, so that the time derivative acts on Q rather than Z. To this end, we introduce the antiderivative of γG, denoted $K(Z)$, i.e.

$$K(Z) \equiv \gamma \int^Z G(\mathcal{Z})\,\mathrm{d}\mathcal{Z},$$

such that

$$K' \equiv \frac{\mathrm{d}K}{\mathrm{d}Z} = \gamma G,$$

and therefore

$$\dot{K} \equiv \frac{\mathrm{d}K}{\mathrm{d}t} = K'\dot{Z} = \gamma G\dot{Z}.$$

Then, by the product rule, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(KQ) = \dot{K}Q + K\dot{Q} = \gamma G\dot{Z}Q + K\dot{Q},$$

which we can use to obtain

$$\mathcal{L}_{b\&e} = \frac{1}{2}M\dot{Z}^2 - MgZ + \frac{1}{2}L_i\dot{Q}^2 - K\dot{Q} + \cancel{\frac{\mathrm{d}}{\mathrm{d}t}(KQ)}.$$

As before, the total time derivative is discarded due to the endpoint conditions (2.8).

Now, $\mathcal{L}_{b\&e}(Z, \dot{Z}, Q, \dot{Q}, t)$ has conjugate momenta $P_Z = M\dot{Z}$ and $P_Q = L_i\dot{Q} - K$,[1] and the Hamiltonian can be found as follows:

$$\begin{aligned}
\mathcal{H}_{b\&e} &= P_Z\dot{Z} + P_Q\dot{Q} - \mathcal{L}_{b\&e} \\
&= M\dot{Z}^2 + (L_i\dot{Q} - K)\dot{Q} - \frac{1}{2}M\dot{Z}^2 + MgZ - \frac{1}{2}L_i\dot{Q}^2 + K\dot{Q} \\
&= \frac{1}{2}MW^2 + MgZ + \frac{1}{2}L_i\dot{Q}^2, \quad \text{using } \dot{Z} = W \\
&= \frac{1}{2}MW^2 + MgZ + \frac{(P_Q + K)^2}{2L_i}, \quad \text{using } \dot{Q} = \frac{P_Q + K}{L_i}.
\end{aligned}$$

Rearranging for the Lagrangian then gives

$$\begin{aligned}
\mathcal{L}_{b\&e} &= MW\dot{Z} + P_Q\dot{Q} - \mathcal{H}_{b\&e} \\
&= MW\dot{Z} + P_Q\dot{Q} - \frac{1}{2}MW^2 - MgZ - \frac{(P_Q + K)^2}{2L_i},
\end{aligned}$$

which, with the hydrodynamic terms recovered and after multiplication by -1, gives

$$\begin{aligned}
\mathcal{L}[\mathbf{v}] = \int_0^{L_x}\int_R^{l_y}\Bigg[\int_0^h \left\{\rho\left(\frac{\partial\phi}{\partial t} + \frac{1}{2}\|\nabla\phi\|^2 + g(z - H_0)\right) + \mathcal{P}(\rho - \rho_0)\right\}\mathrm{d}z \\
+ \rho_0\lambda(h - h_b)\Theta(y - y_b)\Bigg]\mathrm{d}y\,\mathrm{d}x \\
- MW\dot{Z} - P_Q\dot{Q} + \frac{1}{2}MW^2 + MgZ + \frac{(P_Q + K)^2}{2L_i},
\end{aligned}$$

with

$$\mathbf{v} = (h, \rho, \phi, \mathcal{P}, \lambda, Z, W, P_Q, Q).$$

This matches the Lagrangian used in [44] (equation 12b) after replacing ρ with $\rho_0 D$ and $\mathcal{P}$ with p, and in [45] (equation 15b) after explicitly setting $\rho = \rho_0$. The variations in Z,

[1] These are different to the momenta obtained in Section 2.3.2, due to the use of the product rule transferring a time derivative from Z to Q.

W, P_Q and Q together give the ODE system (2.38), without the capacitor term Q/C, after the substitution

$$I = \frac{P_Q + K}{L_i}.$$

A.5 Linearisation

A.5.1 Equations

To linearise the system (2.39) about its rest-state, we first write each function in terms of its rest value and displacement, then Taylor-expand and neglect higher-order terms in the displacements. These decompositions are as follows:

$$\begin{aligned}
\phi(x,y,z,t) &= 0 + \widetilde{\phi}(x,y,z,t), & h(x,y,t) &= H(y) + \eta(x,y,t), & R(t) &= 0 + \widetilde{R}(t),\\
\lambda(x,y,t) &= \Lambda(y) + \widetilde{\lambda}(x,y,t), & y_b(x,t) &= L_b + \widetilde{y}_b(x,t), & W(t) &= 0 + \widetilde{W}(t),\\
Z(t) &= Z_0 + \widetilde{Z}(t), & Q(t) &= 0 + \widetilde{Q}(t), & I(t) &= 0 + \widetilde{I}(t).
\end{aligned}$$

Recall that, by definition,

$$H(y) \equiv \begin{cases} H_0 & y < L_b, \\ H_b(y) & y \geq L_b, \end{cases} \tag{A.29}$$

and

$$\Lambda(y) \equiv g(H_0 - H_b), \tag{A.30}$$

with $\Lambda(L_b) = 0$. Also recall the expression for the harvested LED voltage V:

$$V(I) \equiv \operatorname{sign}(I) n_q V_T \ln\left(1 + \frac{|I|}{I_{sat}}\right) \tag{A.31}$$

We linearise each equation in turn.

Waterline and associated λ BC

We first consider the dependence of y_b on Z and h (2.12). In this context, the variations $\delta\cdot$ are equivalent to the displacements $\widetilde{\cdot}$, giving

$$\widetilde{y}_b = \frac{\widetilde{Z} - \eta|_{y=y_b^-}}{\frac{\partial H}{\partial y}|_{y=y_b^-} + \frac{\partial \eta}{\partial y}|_{y=y_b^-} + \tan\alpha}.$$

Then, the evaluations at $y = y_b^-$ can be Taylor-expanded:

$$\begin{aligned}\widetilde{y}_b &= \frac{\widetilde{Z} - \eta|_{y=L_b^-} + \cancel{\mathcal{O}(\widetilde{y}_b\eta)}}{\cancel{\frac{\partial H}{\partial y}|_{y=L_b^-}} + \widetilde{y}_b\cancel{\frac{\partial^2 H}{\partial y^2}|_{y=L_b^-}} + \cancel{\mathcal{O}(\widetilde{y}_b^2)} + \frac{\partial \eta}{\partial y}|_{y=L_b^-} + \cancel{\mathcal{O}(\widetilde{y}_b\eta)} + \tan\alpha} \\ &= \frac{\widetilde{Z} - \eta|_{y=L_b^-}}{\frac{\partial \eta}{\partial y}|_{y=L_b^-} + \tan\alpha},\end{aligned}$$

where we note that the derivatives of H at $y = L_b^-$ are zero by (A.29).

Now, the assumption that the displacements are small means $\frac{\partial \eta}{\partial y}|_{y=L_b^-} \ll 1$. Meanwhile, for small α, $\tan\alpha$ is also $\ll 1$, so the derivative term may not necessarily be neglected. However, for such a small hull angle, even the smallest deviation in the free-surface height would cause a large deviation in the position of the waterline, and therefore the small-displacement assumption of the linearisation would break down. Geometrically, we have $\widetilde{y}_b = \mathcal{O}(\eta/\tan\alpha)$, and it is therefore a requirement of the linearisation that the angle α is large enough for the approximation $\widetilde{y}_b \ll 1$ to hold. Thus, we have $\frac{\partial \eta}{\partial y}|_{y=L_b^-} \ll \tan\alpha$ and the derivative can be neglected, such that the explicit value of the waterline displacement is

$$\widetilde{y}_b = \frac{\widetilde{Z} - \eta|_{y=L_b^-}}{\tan\alpha}. \tag{A.32}$$

Now, consider the waterline BC (2.39i); after a Taylor expansion and use of (A.30) and (A.32), we have

$$\begin{aligned}&\Lambda(L_b + \widetilde{y}_b) + \widetilde{\lambda}|_{y=L_b+\widetilde{y}_b} = 0 \\ \Rightarrow\ &\cancel{\Lambda(L_b)} + \widetilde{y}_b\Lambda'(L_b) + \cancel{\mathcal{O}\left(\widetilde{y}_b^2\right)} + \widetilde{\lambda}|_{y=L_b} + \cancel{\mathcal{O}\left(\widetilde{y}_b\widetilde{\lambda}\right)} = 0 \\ \Rightarrow\ &\widetilde{\lambda}|_{y=L_b} = -g\widetilde{y}_b\tan\alpha = \boxed{g(\eta|_{y=L_b^-} - \widetilde{Z})}.\end{aligned}$$

Bernoulli equation

Next, we consider the Bernoulli equation (2.39b); in particular, the linearisation of both the Heaviside function and the evaluation at the free surface. After substitution of the decompositions, and making use of (A.30), we have

$$\frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\cancel{\|\nabla\widetilde{\phi}\|^2} + g(H+\eta-H_0) + (g(H_0-H_b)+\widetilde{\lambda})\Theta(y-L_b-\widetilde{y}_b) = 0 \ \text{ at } \ z = H+\eta.$$

First, we remove the Heaviside function by splitting the equation in two; for $y \leq L_b + \widetilde{y}_b$ and $y \geq L_b + \widetilde{y}_b$:

$$\begin{aligned} \frac{\partial\widetilde{\phi}}{\partial t} + g\eta + g(H-H_0) = 0 \ \text{ for } \ y \leq L_b + \widetilde{y}_b; \\ \frac{\partial\widetilde{\phi}}{\partial t} + g\eta + g(H-H_b) + \widetilde{\lambda} = 0 \ \text{ for } \ y \geq L_b + \widetilde{y}_b. \end{aligned}$$

Then, we evaluate each of the above at $y = L_b + \widetilde{y}_b$ and Taylor-expand, using (A.29) and taking the appropriate limits on H:

$$\begin{aligned} \left(\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + g(H-H_0)\right)\Big|_{y=L_b+\widetilde{y}_b^-} &= \left(\frac{\partial\widetilde{\phi}}{\partial t} + g\eta\right)\Big|_{y=L_b^-} + \cancel{\mathcal{O}\left(\widetilde{y}_b\widetilde{\phi}, \widetilde{y}_b\eta\right)}; \\ \left(\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + g(H-H_b) + \widetilde{\lambda}\right)\Big|_{y=L_b+\widetilde{y}_b^+} &= \left(\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + \widetilde{\lambda}\right)\Big|_{y=L_b^+} + \cancel{\mathcal{O}\left(\widetilde{y}_b\widetilde{\phi}, \widetilde{y}_b\eta, \widetilde{y}_b\widetilde{\lambda}\right)}. \end{aligned}$$

These resulting linear expressions can be re-written on either side of L_b:

$$\begin{aligned} \frac{\partial\widetilde{\phi}}{\partial t} + g\eta = 0 \ \text{ for } \ y \leq L_b; \\ \frac{\partial\widetilde{\phi}}{\partial t} + g\eta + \widetilde{\lambda} = 0 \ \text{ for } \ y \geq L_b, \end{aligned}$$

which can then be recombined using a Heaviside function as

$$\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + \widetilde{\lambda}\Theta(y-L_b) = 0 \ \text{ at } \ z = H + \eta(x,y,t).$$

Finally, the evaluation at $z = H + \eta$ is carried out in a similar way:

$$\left(\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + \widetilde{\lambda}\Theta(y - L_b)\right)\Big|_{z=H+\eta} = \left(\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + \widetilde{\lambda}\Theta(y - L_b)\right)\Big|_{z=H} + \cancel{\mathcal{O}\left(\eta\widetilde{\phi}, \eta^2, \eta\widetilde{\lambda}\right)},$$

giving

$$\boxed{\frac{\partial\widetilde{\phi}}{\partial t} + g\eta + \widetilde{\lambda}\Theta(y - L_b) = 0 \text{ at } z = H(y)}.$$

Kinematic condition

The treatment of the kinematic condition (2.39c) is similar to the final step for the Bernoulli equation above:

$$\begin{aligned}
&\frac{\partial\eta}{\partial t} + \nabla_H\widetilde{\phi}\cdot\nabla_H H + \cancel{\mathcal{O}\left(\widetilde{\phi}\eta\right)} = \frac{\partial\widetilde{\phi}}{\partial z} \text{ at } z = H(y) + \eta(x, y, t)\\
\Rightarrow &\left(\frac{\partial\eta}{\partial t} + H'\frac{\partial\widetilde{\phi}}{\partial y}\right)\Big|_{z=H} + \cancel{\mathcal{O}\left(\eta^2, \eta\widetilde{\phi}\right)} = \frac{\partial\widetilde{\phi}}{\partial z}\Big|_{z=H} + \cancel{\mathcal{O}\left(\eta\widetilde{\phi}\right)}\\
&\Rightarrow \boxed{\frac{\partial\eta}{\partial t} + H'\frac{\partial\widetilde{\phi}}{\partial y} = \frac{\partial\widetilde{\phi}}{\partial z} \text{ at } z = H(y)}.
\end{aligned}$$

Wavemaker BC

Similarly, the BC at the wavemaker $y = R$ (2.39j) is linearised as follows:

$$\frac{\partial\widetilde{\phi}}{\partial y}\Big|_{y=0+\widetilde{R}} = \dot{\widetilde{R}} \Rightarrow \frac{\partial\widetilde{\phi}}{\partial y}\Big|_{y=0} + \cancel{\mathcal{O}\left(\widetilde{R}\widetilde{\phi}\right)} = \dot{\widetilde{R}} \Rightarrow \boxed{\frac{\partial\widetilde{\phi}}{\partial y} = \dot{\widetilde{R}} \text{ at } y = 0}.$$

Buoy acceleration ODE

Considering next the equation for the buoy acceleration W (2.39d), note in particular the presence of the integral, which needs to be linearised. After substituting in the decompositions and trivially Taylor-expanding G, we have

$$M\dot{\widetilde{W}} = -Mg - \gamma G(Z_0)\widetilde{I} + \cancel{\mathcal{O}\left(\widetilde{Z}\widetilde{I}\right)} + \rho_0\int_0^{L_x}\int_{\widetilde{R}}^{l_y}(\Lambda + \widetilde{\lambda})\Theta(y - L_b - \widetilde{y}_b)\,\mathrm{d}y\,\mathrm{d}x.$$

The integral can be manipulated in a number of ways.

First, note that the presence of the Heaviside function means the lower bound of the

y-integral can be modified without affecting the result, since $\widetilde{R}$ will always be less than $L_b + \widetilde{y}_b$ and providing the new bound is also less than $L_b + \widetilde{y}_b$. We can therefore choose 0 for this lower bound, giving

$$M\dot{\widetilde{W}} = -Mg - \gamma G(Z_0)\widetilde{I} + \rho_0 \int_0^{L_x} \int_0^{l_y} (\Lambda + \widetilde{\lambda})\Theta(y - L_b - \widetilde{y}_b)\,\mathrm{d}y\,\mathrm{d}x. \tag{A.33}$$

Now, note that while the y-dependence is integrated out, the value of the integral depends on the position of the waterline; thus, it can be written as a function of y_b, and can be Taylor-expanded as such. Writing

$$\rho_0 \int_0^{L_x} \int_0^{l_y} (\Lambda + \widetilde{\lambda})\Theta(y - y_b)\,\mathrm{d}y\,\mathrm{d}x = \mathcal{I}(y_b),$$

we linearise $\mathcal{I}$ as follows:

$$\begin{aligned}
&\mathcal{I}(L_b+\widetilde{y}_b) \\
&\quad = \mathcal{I}(L_b) + \widetilde{y}_b \left.\frac{\partial \mathcal{I}}{\partial y_b}\right|_{y_b=L_b} + \cancel{\mathcal{O}\left(\widetilde{y}_b^2\right)} \\
&\quad = \rho_0 \int_0^{L_x} \int_0^{l_y} \left\{ (\Lambda + \widetilde{\lambda})\Theta(y - L_b) + \widetilde{y}_b \Lambda(y) \left.\frac{\partial \Theta(y - y_b)}{\partial y_b}\right|_{y_b=L_b} \right\} \mathrm{d}y\,\mathrm{d}x + \cancel{\mathcal{O}\left(\widetilde{y}_b\widetilde{\lambda}\right)} \\
&\quad = \rho_0 \int_0^{L_x} \int_0^{l_y} \left\{ (\Lambda + \widetilde{\lambda})\Theta(y - L_b) - \widetilde{y}_b \Lambda(y)\delta(y - L_b) \right\} \mathrm{d}y\,\mathrm{d}x \\
&\quad = \rho_0 \int_0^{L_x} \int_0^{l_y} \left\{ (\Lambda + \widetilde{\lambda})\Theta(y - L_b) - \widetilde{y}_b \cancel{\Lambda(L_b)} \right\} \mathrm{d}y\,\mathrm{d}x \\
&\quad = Mg + \rho_0 \int_0^{L_x} \int_0^{l_y} \widetilde{\lambda}\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x,
\end{aligned}$$

where, in the last step, we have used (2.41).

Then, returning to (A.33), we have

$$\boxed{M\dot{\widetilde{W}} = -\gamma G(Z_0)\widetilde{I} + \rho_0 \int_0^{L_x} \int_0^{l_y} \widetilde{\lambda}\Theta(y - L_b)\,\mathrm{d}y\,\mathrm{d}x}.$$

Current ODE

For the current I (2.39g) we have

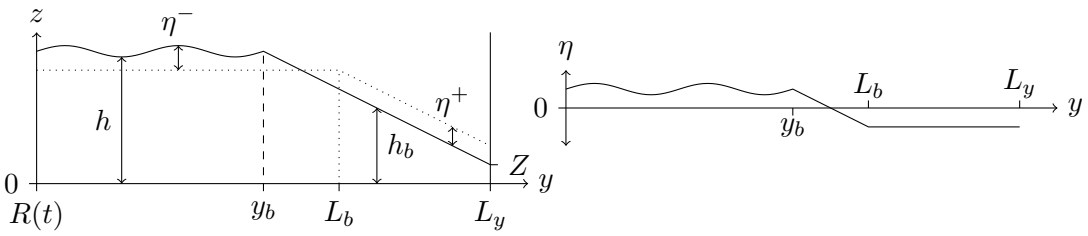
$$L_i\dot{\widetilde{I}} = \gamma G(Z_0 + \widetilde{Z})\widetilde{W} - \frac{\widetilde{Q}}{C} - \widetilde{I}(R_i + R_c) - V(\widetilde{I}).$$

Using (A.31) and Taylor-expanding the ln function, we obtain

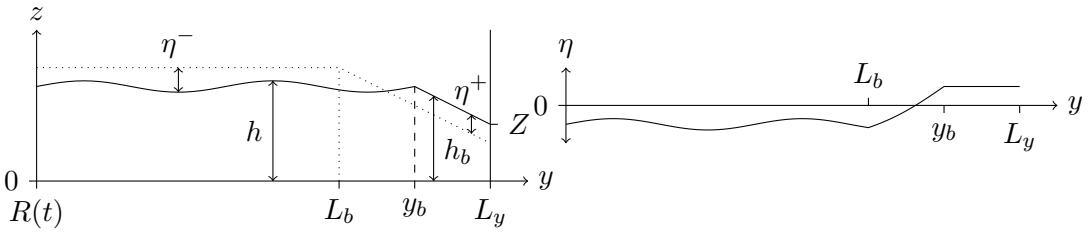
$$\begin{aligned} L_i\dot{\widetilde{I}} &= \gamma G(Z_0)\widetilde{W} + \cancel{\mathcal{O}\left(\widetilde{Z}\widetilde{W}\right)} - \frac{\widetilde{Q}}{C} - \widetilde{I}(R_i + R_c) - \operatorname{sign}(\widetilde{I})n_q V_T \left(\cancel{\ln(1)} + \frac{|\widetilde{I}|}{I_{sat}} + \cancel{\mathcal{O}\left(\widetilde{I}^2\right)}\right) \\ &= \boxed{\gamma G(Z_0)\widetilde{W} - \frac{\widetilde{Q}}{C} - \widetilde{I}(R_i + R_c + R_l)}, \end{aligned}$$

where we have used the identity $|\widetilde{I}|\operatorname{sign}(\widetilde{I}) = \widetilde{I}$, and defined $R_l \equiv n_q V_T / I_{sat}$.

Buoy-water continuity



(a) At $t = t_1$, where the buoy has moved downwards and the waterline to the left.



(b) At $t = t_2$, where the buoy has moved upwards and the waterline to the right.

Figure A.2: h (left) and η (right) at two different times, in the $x = L_x/2$ plane.

Figure A.2 shows h and η along the centreline of the tank for two cases, one where the buoy has moved downwards and the waterline to the left ($\widetilde{Z} < 0, \widetilde{y}_b < 0$) and the other where the buoy has moved upwards and the waterline to the right ($\widetilde{Z} > 0, \widetilde{y}_b > 0$).[2] The moving waterline means that, in general for $t > 0$, there is a region between y_b and L_b

[2]There will be certain combinations of wavelength and amplitude that generate other cases, for example where the buoy has moved downwards yet the waterline has moved to the right, but these are not considered here. The key fact is in which direction the waterline has moved.

where the surface is constrained but used to be free (when the waterline moves to the left), or is free but used to be constrained (when the waterline moves to the right). In general, nothing a priori can be said about η in this region, but to the right of it—i.e. for $y \geq \max(y_b, L_b)$—the buoy-water interface condition (2.39h) requires $\eta = \widetilde{Z}$. Taylor-expanding η at $L_b + \widetilde{y}_b$ gives $\eta|_{L_b+\widetilde{y}_b} = \eta|_{L_b} + \cancel{\mathcal{O}(\widetilde{y}_b\eta)}$, or simply that η at $L_b + \widetilde{y}_b$ equals η at L_b. Therefore, regardless of what $\max(L_b + \widetilde{y}_b, L_b)$ is,

$$\boxed{\eta = \widetilde{Z} \text{ for } y \geq L_b}.$$

Laplace equation and remaining ODEs and BC

Finally, the treatment of the Laplace equation (2.39a), the remaining ODEs for Z (2.39e) and Q (2.39f), and the remaining BC (2.39k) are trivial:

$$\boxed{\nabla^2\widetilde{\phi} = 0}, \qquad \boxed{\dot{\widetilde{Z}} = \widetilde{W}}, \qquad \boxed{\dot{\widetilde{Q}} = \widetilde{I}}, \qquad \boxed{\nabla\widetilde{\phi} \cdot \hat{\mathbf{n}} = 0 \text{ on } \partial\Omega_w}.$$

A.5.2 Lagrangian

To linearise the Lagrangian (2.29) about its rest-state, we again write each function as its "rest-value-plus-displacement" decomposition and Taylor-expand, noting that here we keep terms quadratic in the displacements, as quadratic variations produce linear equations. After substitution of the decompositions, the Lagrangian (2.29) (with the constant density constraint $\rho = \rho_0$ applied strongly) becomes

$$\begin{aligned}\mathcal{L} = \rho_0 \int_0^{L_x} \int_{0+\widetilde{R}}^{l_y} \Bigg[\int_0^{H+\eta} \left\{ \frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 + g(z - H_0) \right\} \mathrm{d}z \\ + (\Lambda + \widetilde{\lambda})(H + \eta - H_b - \widetilde{Z})\Theta(y - L_b - \widetilde{y}_b) \Bigg] \mathrm{d}y\,\mathrm{d}x \\ - \frac{1}{2}M\dot{\widetilde{Z}}^2 + Mg(\cancel{Z_0} + \widetilde{Z}) - \frac{1}{2}L_i\dot{\widetilde{Q}}^2 + \frac{\widetilde{Q}^2}{2C} - \gamma G(Z_0)\dot{\widetilde{Z}}\widetilde{Q} + \cancel{\mathcal{O}\left(\widetilde{Z}^2\widetilde{Q}\right)}, \end{aligned} \tag{A.34}$$

where the constant MgZ_0 and the cubic (and higher-order) terms from the expansion of $G(Z_0 + \widetilde{Z})$ are discarded. Particular attention is needed with regard to the "quadricisation" of the y- and z-integral bounds, and the Heaviside function present in the

water-to-buoy coupling.

The water-to-buoy coupling

We begin by dealing with the water-to-buoy coupling term. First, Taylor-expanding the Heaviside function applied to some generic function $f(y)$ gives

$$\int_R^{l_y} f\Theta(y - L_b - \widetilde{y}_b)\,\mathrm{d}y \\ = \int_R^{l_y} \left\{ f\Theta(y - L_b) - \widetilde{y}_b f\delta(y - L_b) + \frac{\widetilde{y}_b^2}{2} f\delta'(y - L_b) \right\} \mathrm{d}y + \cancel{\mathcal{O}\left(\widetilde{y}_b^3\right)},$$

where we note that the delta functions transform the y-integrals into evaluations of f and its derivative at $y = L_b$:

$$\int_R^{l_y} f\Theta(y - L_b - \widetilde{y}_b)\,\mathrm{d}y = \int_R^{l_y} f\Theta(y - L_b)\,\mathrm{d}y - \widetilde{y}_b f(L_b) - \frac{\widetilde{y}_b^2}{2} f'(L_b).$$

Now, we shall introduce the function f in two stages, considering both $H - H_b$ and $\eta - \widetilde{Z}$ individually.

First, let $f = (\Lambda + \widetilde{\lambda})(H - H_b)$. Then

$$\widetilde{y}_b f(L_b) = \widetilde{y}_b \{\Lambda(L_b) + \widetilde{\lambda}(L_b, y, t)\}\{H(L_b) - H_b(L_b)\}.$$

But, by the definition of H (2.11), $H(L_b) = H_b(L_b)$, so $f(L_b)$ vanishes. Similarly, for the leading term multiplying the Heaviside function, note that $H - H_b$ is zero whenever $\Theta(y - L_b)$ is nonzero. Now, before evaluating the derivative $f'(L_b)$, consider that H is piecewise-defined, and its derivative at $y = L_b$ is not well-defined. However, we must recall the range over which the Heaviside function is "active"; for $y \geq L_b$. Thus, $\delta'(y - L_b)$ distributes from above $y = L_b$, and we can take the derivative from that direction, which is well-defined:

$$\frac{\widetilde{y}_b^2}{2} f'(L_b^+) = \frac{\widetilde{y}_b^2}{2} \left(\frac{\mathrm{d}}{\mathrm{d}y}\{\Lambda(H - H_b)\} \right)\Big|_{y = L_b^+} + \cancel{\mathcal{O}\left(\widetilde{y}_b^2 \widetilde{\lambda}\right)} \\ = \frac{\widetilde{y}_b^2}{2} \Lambda'(L_b^+) \cancel{\{H(L_b^+) - H_b(L_b^+)\}} + \frac{\widetilde{y}_b^2}{2} \cancel{\Lambda(L_b^+)} \{H'(L_b^+) - H_b'(L_b^+)\}$$

$$= 0;$$

note that neither derivative actually needs to be evaluated, as both terms of the product rule have a function-evaluation at $y = L_b$ that vanishes. Thus, we see that all terms of the Taylor-expansion involving $H - H_b$ vanish.

Now, let $f = (\Lambda + \widetilde{\lambda})(\eta - \widetilde{Z})$. Then

$$\widetilde{y}_b f(L_b) = \widetilde{y}_b \cancel{\Lambda(L_b)}\{\eta(x, L_b, t) - \widetilde{Z}(t)\} + \cancel{\mathcal{O}\left(\widetilde{y}_b\widetilde{\lambda}\eta, \widetilde{y}_b\widetilde{\lambda}\widetilde{Z}\right)} = 0, \text{ and}$$
$$\frac{\widetilde{y}_b^2}{2} f'(L_b^+) = \cancel{\mathcal{O}\left(\widetilde{y}_b^2\eta, \widetilde{y}_b^2\widetilde{\lambda}\widetilde{Z}\right)} = 0,$$

where again the evaluation of $\Lambda(L_b)$ vanishes. Thus, after "quadricising" the Heaviside function, we are left with

$$\int_R^{l_y} (\Lambda + \widetilde{\lambda})(H + \eta - H_b - \widetilde{Z})\Theta(y - L_b - \widetilde{y}_b)\,\mathrm{d}y = \int_R^{l_y} (\Lambda + \widetilde{\lambda})(\eta - \widetilde{Z})\Theta(y - L_b)\,\mathrm{d}y. \quad \text{(A.35)}$$

The z-integral

Next, we must "quadricise" the upper bound of the z-integral $h = H + \eta$. We have already seen how to handle perturbations to integral bounds, when taking the variation in h (in Section 2.3): we write the integral as a function of h, then Taylor-expand it as usual. Starting with

$$\int_0^{H+\eta} \left\{ \frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 + g(z - H_0) \right\} \mathrm{d}z,$$

we first write the integrand as F for convenience, and then write the integral as $f(h)$:

$$\frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 + g(z - H_0) \equiv F;$$
$$\int_0^{H+\eta} F\,\mathrm{d}z \equiv f(H + \eta).$$

Then we can simply Taylor-expand f:

$$f(H + \eta) = f(H) + \eta f'(H) + \frac{\eta^2}{2} f''(H) + \cancel{\mathcal{O}\left(\eta^3\right)};$$

$$f(H) = \int_0^H \left\{ \frac{\partial \widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 + \cancel{g(z - H_0)} \right\} \mathrm{d}z;$$

$$\eta f'(H) = \eta F|_{z=H} = \eta \left.\frac{\partial \widetilde{\phi}}{\partial t}\right|_{z=H} + \eta g(H - H_0) + \cancel{\mathcal{O}\left(\eta\widetilde{\phi}^2\right)};$$

$$\frac{\eta^2}{2} f''(H) = \frac{\eta^2}{2} \left.\frac{\partial F}{\partial z}\right|_{z=H} = \frac{1}{2} g\eta^2 + \cancel{\mathcal{O}\left(\eta^2\widetilde{\phi}\right)},$$

where in the second step we have discarded $g(z - H_0)$, because the integral's fixed upper bound means the term evaluates to a constant, and therefore contributes nothing to the variations. Therefore, we find that

$$\int_0^{H+\eta} \left\{ \frac{\partial \widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 + g(z - H_0) \right\} \mathrm{d}z$$
$$= \int_0^H \left\{ \frac{\partial \widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 \right\} \mathrm{d}z + \eta \left.\frac{\partial \widetilde{\phi}}{\partial t}\right|_{z=H} + \eta g(H - H_0) + \frac{1}{2} g\eta^2. \quad \text{(A.36)}$$

Cancelling linear terms

Now, before progressing onto the "quadricisation" of the y-integral, let us consider the intermediate Lagrangian: after using (A.35) and (A.36) in (A.34), we have

$$\mathcal{L} = \rho_0 \int_0^{L_x} \int_R^{l_y} \left[\int_0^H \left\{ \frac{\partial \widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 \right\} \mathrm{d}z + \eta \left.\frac{\partial \widetilde{\phi}}{\partial t}\right|_{z=H} + \eta g(H - H_0) + \frac{1}{2} g\eta^2 \right.$$
$$\left. + (\Lambda + \widetilde{\lambda})(\eta - \widetilde{Z})\Theta(y - L_b) \right] \mathrm{d}y\, \mathrm{d}x - \frac{1}{2} M \dot{\widetilde{Z}}^2 + Mg\widetilde{Z} - \frac{1}{2} L_i \dot{\widetilde{Q}}^2 + \frac{\widetilde{Q}^2}{2C} - \gamma G(Z_0)\dot{\widetilde{Z}}\widetilde{Q}.$$

Note the blue terms; these are all linear in the displacements. Other than the time derivative of $\widetilde{\phi}$, each term has a constant coefficient representing some rest-state of the system; since varying these linear terms will produce constant rest-state terms in the equations, and we were able to cancel out such terms, we should be able to do the same here, cancelling out these linear terms.

Consider the terms multiplying $\widetilde{Z}$:

$$\widetilde{Z}\left(Mg - \rho_0 \int_0^{L_x} \int_R^{l_y} \Lambda\Theta(y - L_b)\, \mathrm{d}y\, \mathrm{d}x \right).$$

Recalling the equilibrium of forces acting on the buoy at rest (2.41), we can see that

these terms cancel.[3] For the terms multiplying η, we can use the piecewise definition of H (2.11) to rewrite $g(H - H_0)$ using the Heaviside function; by the definition of the at-rest pressure Λ (2.40), we then see that these terms also cancel:

$$\begin{aligned}\eta g(H - H_0) + \Lambda\eta\Theta(y - L_b) &= \eta g(H_b - H_0)\Theta(y - L_b) + \Lambda\eta\Theta(y - L_b) \\ &= \eta\{g(H_b - H_0) + g(H_0 - H_b)\}\Theta(y - L_b) \\ &= 0.\end{aligned}$$

Note that the time derivative of $\widetilde{\phi}$ has no constant coefficient which can be used to cancel it, and so, for now, it remains. It shall be seen when taking variations that this term alone does not contribute a term to the equations; however, its presence at this stage remains important because its impact on the Taylor-expansion of the y-integral (carried out next) is vital for the generation of the BC on ϕ at $y = 0$.

The y-integral

After removing the relevant linear terms, the final step is to "quadricise" the lower bound of the y-integral $R = 0 + \widetilde{R}$. We do this in precisely the same manner as the z-integral, eventually obtaining:

$$\begin{aligned}\int_{0+\widetilde{R}}^{l_y} \left\{ \int_0^H \left\{ \frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 \right\} \mathrm{d}z + \eta \left.\frac{\partial\widetilde{\phi}}{\partial t}\right|_{z=H} + \frac{1}{2}g\eta^2 + \widetilde{\lambda}(\eta - \widetilde{Z})\Theta(y - L_b) \right\} \mathrm{d}y \\ = \int_0^{l_y}\int_0^H \left\{ \frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 \right\} \mathrm{d}z\,\mathrm{d}y - \widetilde{R}\int_0^{H_0} \left.\frac{\partial\widetilde{\phi}}{\partial t}\right|_{y=0} \mathrm{d}z \\ + \int_0^{l_y} \left\{ \eta \left.\frac{\partial\widetilde{\phi}}{\partial t}\right|_{z=H} + \frac{1}{2}g\eta^2 + \widetilde{\lambda}(\eta - \widetilde{Z})\Theta(y - L_b) \right\} \mathrm{d}y.\end{aligned}$$

Therefore, our final "quadricised" Lagrangian is

$$\begin{aligned}\mathcal{L} = \rho_0 \int_0^{L_x} \Bigg[\int_0^{l_y}\int_0^H \left\{ \frac{\partial\widetilde{\phi}}{\partial t} + \frac{1}{2}\|\nabla\widetilde{\phi}\|^2 \right\} \mathrm{d}z\,\mathrm{d}y + \dot{\widetilde{R}}\int_0^{H_0} \left.\widetilde{\phi}\right|_{y=0} \mathrm{d}z \\ + \int_0^{l_y} \left\{ \eta \left.\frac{\partial\widetilde{\phi}}{\partial t}\right|_{z=H} + \frac{1}{2}g\eta^2 + \widetilde{\lambda}(\eta - \widetilde{Z})\Theta(y - L_b) \right\} \mathrm{d}y \Bigg] \mathrm{d}x\end{aligned}$$

[3] Technically, the lower bound on y in (2.41) is different to the lower bound here. However, because of the Heaviside function, and providing the wavemaker R and waterline L_b never meet, this difference is irrelevant.

$$-\frac{1}{2}M\dot{\widetilde{Z}}^2-\frac{1}{2}L_i\dot{\widetilde{Q}}^2+\frac{\widetilde{Q}^2}{2C}-\gamma G(Z_0)\dot{\widetilde{Z}}\widetilde{Q},$$

where, as a final manipulation, we have transferred the time derivative of ϕ onto $\widetilde{R}$, avoiding the need to do this after taking variations.

Verification that this Lagrangian generates the linearised system (2.42) is left as an exercise. Note that, because the limits of integration are now fixed in time, time derivatives and spatial integrals commute, without requiring the use of the 3D RTT (A.52). A time-independent form of the 3D IBP (A.48) is needed, however, to deal with the boundary terms at $y=0$ and $z=H$.

A.6 Laplace equation solution

The solution $u(x,y)$ to the equation $\nabla^2 u=0$, defined on the 2D domain

$$(x,y)\in\left[-\frac{W}{2},\frac{W}{2}\right]\times\left[-\frac{H}{2},\frac{H}{2}\right]$$

and with BCs

$$\begin{aligned} u\left(-\frac{W}{2},y\right)&=X_-(y)=3\cos\left(\frac{\pi y}{H}\right) & u\left(\frac{W}{2},y\right)&=X_+(y)=\cos\left(\frac{\pi y}{H}\right)\\ u\left(x,-\frac{H}{2}\right)&=Y_-(x)=-2\cos\left(\frac{\pi x}{W}\right) & u\left(x,\frac{H}{2}\right)&=Y_+(x)=-2\cos\left(\frac{\pi x}{W}\right), \end{aligned}$$

can be found using of separation of variables. We start by assuming a separable solution of the form $u=XY$, such that $\nabla^2 u=0$ gives

$$X''Y+XY''=0,\quad\text{or}\quad\frac{X''}{X}=-\frac{Y''}{Y}.$$

Now, we have a function of x equal to a function of y, which is only possible if they are both equal to some constant k^2, which (depending on the BCs) may be negative, zero or positive; i.e. $X''=k^2X$ and $Y''=-k^2Y$, for either $k^2>0$, $k^2=0$ or $k^2<0$.

The separation of variables does not work when there exists more than one pair of inhomogeneous BCs in the problem's dimensions. Since all our BCs are inhomogeneous,

the problem needs to be split into two sub-problems, each with homogeneous BCs on either the x or y boundaries; the solution to the original problem is then the sum of the solutions to the two sub-problems. Similarly, while it is not necessarily required, we shall split these sub-problems in half again, so that we are solving four separate problems each with a single inhomogeneous BC. This partitioning is shown in Figure A.3.

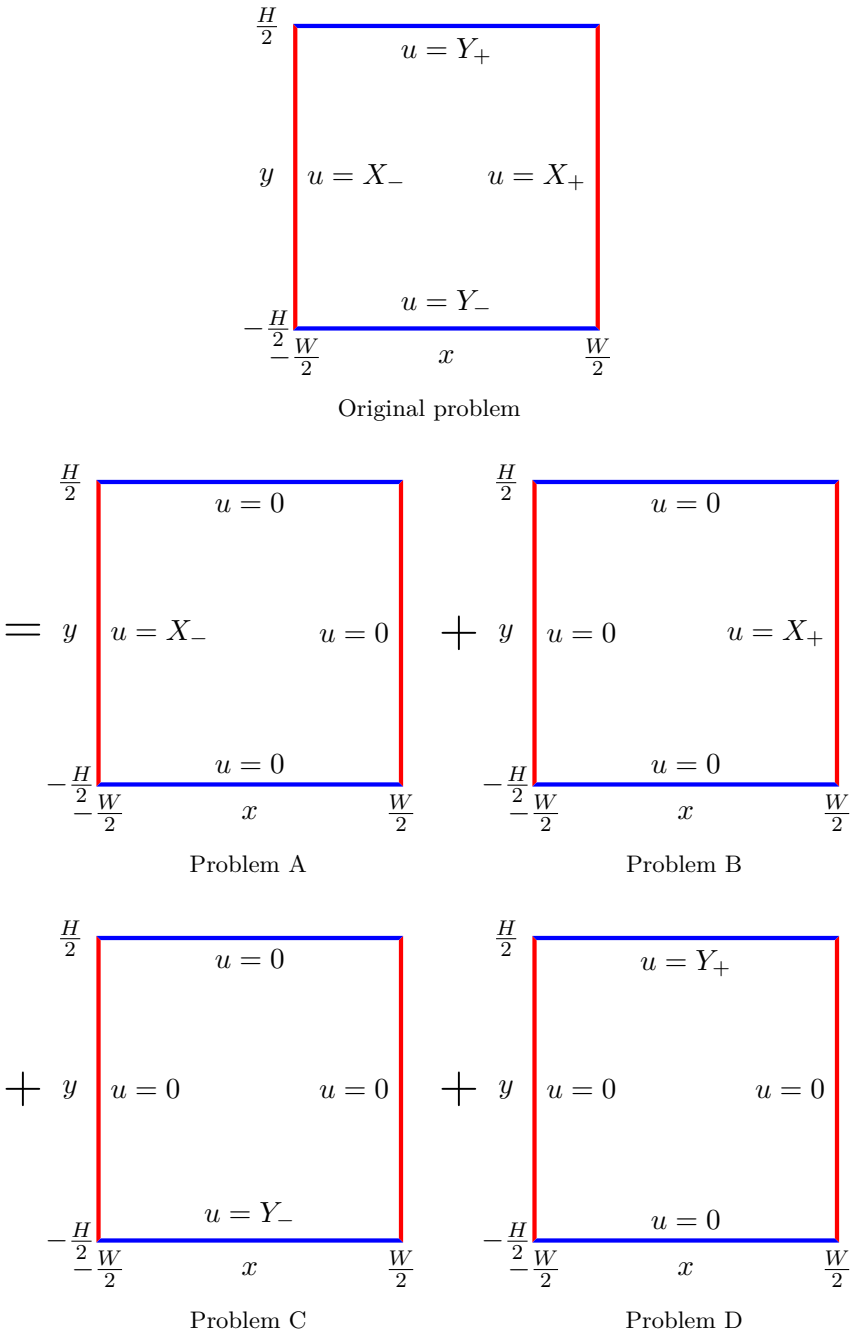


Figure A.3: The problem with four inhomogeneous BCs, split into four sub-problems A-D, each with just one inhomogeneous BC.

First, consider problem A, with $X_+ = Y_- = Y_+ = 0$. These BCs on u transfer onto X and Y as $X(W/2) = Y(-H/2) = Y(H/2) = 0$. Then the only case which permits non-trivial solutions to $Y'' = -k^2 Y$ satisfying the two BCs on Y is when k^2 is positive, with $k_n = \pi(2n+1)/H$ for $n \in \mathbb{Z}$, giving solutions of the form $Y_n = A_n \cos(k_n y)$. Then, $X'' = k^2 X$ with $X(W/2) = 0$ admits solutions of the form

$$X_n = B_n \left(\sinh(k_n x) - \tanh\left(\frac{k_n W}{2}\right) \cosh(k_n x) \right).$$

That is, there are an infinite number of solutions $u_n = X_n Y_n$ which satisfy the three homogeneous BCs; therefore, their sum is also a solution:

$$u_A = \sum_{n=-\infty}^{\infty} C_n \left(\sinh(k_n x) - \tanh\left(\frac{k_n W}{2}\right) \cosh(k_n x) \right) \cos(k_n y). \tag{A.37}$$

Applying the inhomogeneous BC $u(-W/2, y) = X_-$ gives

$$\sum_{n=-\infty}^{\infty} C_n \sinh\left(\frac{k_n W}{2}\right) \cos(k_n y) = -\frac{X_-}{2}, \tag{A.38}$$

where the coefficients C_n are those that allow (A.38) to be satisfied. These coefficients can be found by applying Fourier's method: multiplying (A.38) by $\cos(k_m y)$ for some arbitrary number $m \in \mathbb{Z}$, and integrating between $\pm H/2$. Noting that the sinh term is simply a constant, we have

$$\sum_{n=-\infty}^{\infty} C_n \sinh\left(\frac{k_n W}{2}\right) \int_{-H/2}^{H/2} \cos(k_n y) \cos(k_m y)\, \mathrm{d}y = -\frac{1}{2} \int_{-H/2}^{H/2} X_- \cos(k_m y)\, \mathrm{d}y \tag{A.39}$$

Then, by the orthogonality of cosines, we have

$$\int_{-H/2}^{H/2} \cos(k_n y) \cos(k_m y)\, \mathrm{d}y = \begin{cases} 0 & m \neq n \\ \frac{H}{2} & m = n \end{cases},$$

and (A.39) gives

$$C_n = -\frac{1}{H} \operatorname{csch}\left(\frac{k_n W}{2}\right) \int_{-H/2}^{H/2} X_- \cos(k_n y)\, \mathrm{d}y. \tag{A.40}$$

For an arbitrary X_-, (A.40) is in general nonzero for all n, meaning a solution (A.37) satisfying the inhomogeneous BC exactly would require the entire infinite sum—although, in practice, a good approximation can be found by computing a sufficiently large number of coefficients. However, for simple trigonometric functions such as $X_- = 3\cos(\pi y/H)$, again the orthogonality of cosines means (A.40) is nonzero for only certain n; in this case, for only $n = 0$. That is, C_0 is given by

$$C_0 = -\frac{3}{2}\operatorname{csch}\left(\frac{\pi W}{2H}\right),$$

and, after some rearrangement, the solution to problem A is

$$u_A = 3\sinh\left(\frac{\pi(W-2x)}{2H}\right)\operatorname{csch}\left(\frac{\pi W}{H}\right)\cos\left(\frac{\pi y}{H}\right). \tag{A.41}$$

For problem B, with $X_- = Y_- = Y_+ = 0$ and $X_+ = \cos(\pi y/H)$, the BCs on Y remain the same, while the homogeneous BC on X becomes $X(-W/2) = 0$. Therefore, the solution for X is modified to

$$X_n = B_n\left(\sinh(k_n x) + \tanh\left(\frac{k_n W}{2}\right)\cosh(k_n x)\right),$$

the coefficient C_0 becomes

$$C_0 = \frac{1}{2}\operatorname{csch}\left(\frac{\pi W}{2H}\right),$$

and the solution is

$$u_B = \sinh\left(\frac{\pi(W+2x)}{2H}\right)\operatorname{csch}\left(\frac{\pi W}{H}\right)\cos\left(\frac{\pi y}{H}\right). \tag{A.42}$$

For problems C and D, the process is precisely the same, with just the sign of k^2 switched so that both $X_- = 0$ and $X_+ = 0$ can be satisfied. The result is that the trigonometric and hyperbolic functions are swapped, with cosines of x and hyperbolic functions of y, and with W in the denominators and H in the numerators. For problem C, with

$Y_- = -2\cos(\pi x/W)$ and $Y_+ = 0$, we have

$$u_C = -2\sinh\left(\frac{\pi(H-2y)}{2W}\right)\operatorname{csch}\left(\frac{\pi H}{W}\right)\cos\left(\frac{\pi x}{W}\right), \tag{A.43}$$

while for $Y_- = 0$ and $Y_+ = -2\cos(\pi x/W)$, problem D's solution is

$$u_D = -2\sinh\left(\frac{\pi(H+2y)}{2W}\right)\operatorname{csch}\left(\frac{\pi H}{W}\right)\cos\left(\frac{\pi x}{W}\right). \tag{A.44}$$

Finally, the solution to the full problem with four inhomogeneous BCs is given by the sum of (A.41)–(A.44):

$$\begin{aligned} u = &\left\{3\sinh\left(\frac{\pi(W-2x)}{2H}\right) + \sinh\left(\frac{\pi(W+2x)}{2H}\right)\right\}\operatorname{csch}\left(\frac{\pi W}{H}\right)\cos\left(\frac{\pi y}{H}\right) \\ &- 2\left\{\sinh\left(\frac{\pi(H-2y)}{2W}\right) + \sinh\left(\frac{\pi(H+2y)}{2W}\right)\right\}\operatorname{csch}\left(\frac{\pi H}{W}\right)\cos\left(\frac{\pi x}{W}\right). \end{aligned}$$

A.7 The Schur complement

A.7.1 The 4×4 matrix

The system (6.3) contains a 4×4 matrix, for which a direct analytical inverse is not feasibly obtained. Rather, the system can be written in the following block form:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}\begin{pmatrix} (\hat{W},\hat{Z})^T \\ (\hat{Q},\hat{I})^T \end{pmatrix} = \begin{pmatrix} (A,0)^T \\ (0,0)^T \end{pmatrix}, \tag{A.45}$$

with

$$\mathbf{A} = \begin{pmatrix} i\omega M & 0 \\ -1 & i\omega \end{pmatrix} \qquad \mathbf{B} = \gamma\mathcal{G}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{C} = -\gamma\mathcal{G}\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \qquad \mathbf{D} = \begin{pmatrix} i\omega & -1 \\ \frac{1}{C} & i\omega L_i + \mathcal{R} \end{pmatrix}.$$

The block matrix equation (A.45) can be solved using the Schur complement. This method essentially involves performing block Gaussian elimination on the system; elim-

inating one of the blocks to enable the other to be more easily computed, before using back-substitution to solve for the eliminated block. Because the RHS of the second block is zero, we will find that eliminating the second block rather than the first requires fewer matrix calculations.

Note that the second block is

$$\mathbf{C}\begin{pmatrix}\hat{W}\\ \hat{Z}\end{pmatrix}+\mathbf{D}\begin{pmatrix}\hat{Q}\\ \hat{I}\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix},$$

from which

$$\begin{pmatrix}\hat{Q}\\ \hat{I}\end{pmatrix}=-\mathbf{D}^{-1}\mathbf{C}\begin{pmatrix}\hat{W}\\ \hat{Z}\end{pmatrix}. \tag{A.46}$$

Using this in the fist block gives

$$(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})\begin{pmatrix}\hat{W}\\ \hat{Z}\end{pmatrix}=\begin{pmatrix}A\\ 0\end{pmatrix}. \tag{A.47}$$

The matrix $\mathbf{S}=\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ is the Schur complement, and its use enables the much simpler (A.46) and (A.47) to be solved, in place of (A.45).

Let us now calculate $\mathbf{S}$, also keeping track of $\mathbf{D}^{-1}\mathbf{C}$ for the back-substitution.

$$\mathbf{D}^{-1}\mathbf{C}=\frac{\gamma\mathcal{G}}{\omega^2L_i-i\omega\mathcal{R}-1/C}\begin{pmatrix}1 & 0\\ i\omega & 0\end{pmatrix},$$

$$\mathbf{S}=\begin{pmatrix}i\omega M & 0\\ -1 & i\omega\end{pmatrix}-\frac{\gamma^2\mathcal{G}^2}{\omega^2L_i-i\omega\mathcal{R}-1/C}\begin{pmatrix}i\omega & 0\\ 0 & 0\end{pmatrix}=\begin{pmatrix}i\omega\left(M-\frac{\gamma^2\mathcal{G}^2}{\omega^2L_i-i\omega\mathcal{R}-1/C}\right) & 0\\ -1 & i\omega\end{pmatrix}.$$

Since $\mathbf{S}$ happens to be lower-triangular, solving (A.47) is trivial:

$$\hat{W}=\frac{A}{i\omega\left(M-\frac{\gamma^2\mathcal{G}^2}{\omega^2L_i-i\omega\mathcal{R}-1/C}\right)}=\frac{i(\omega^2L_i-i\omega\mathcal{R}-1/C)A}{\omega(\gamma^2\mathcal{G}^2-(\omega^2L_i-i\omega\mathcal{R}-1/C)M)};$$

$$\hat{Z}=\frac{\hat{W}}{i\omega}=\frac{(\omega^2L_i-i\omega\mathcal{R}-1/C)A}{\omega^2(\gamma^2\mathcal{G}^2-(\omega^2L_i-i\omega\mathcal{R}-1/C)M)},$$

with (A.46) giving

$$\begin{pmatrix} \hat{Q} \\ \hat{I} \end{pmatrix} = \frac{-\gamma\mathcal{G}}{\omega^2 L_i - i\omega\mathcal{R} - 1/C} \begin{pmatrix} 1 & 0 \\ i\omega & 0 \end{pmatrix} \frac{i(\omega^2 L_i - i\omega\mathcal{R} - 1/C)A}{\omega(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \begin{pmatrix} 1 \\ \frac{1}{i\omega} \end{pmatrix}$$
$$= \frac{\gamma\mathcal{G}A}{\omega(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \begin{pmatrix} -i \\ \omega \end{pmatrix}.$$

Therefore, in summary, the solutions to (6.3) via the block form (A.45) and Schur complement $\mathbf{S}$ are

$$\begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q} \\ \hat{I} \end{pmatrix} = \begin{pmatrix} \frac{i(\omega^2 L_i - i\omega\mathcal{R} - 1/C)A}{\omega(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \\ \frac{(\omega^2 L_i - i\omega\mathcal{R} - 1/C)A}{\omega^2(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \\ \frac{-i\gamma\mathcal{G}A}{\omega(\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M)} \\ \frac{\gamma\mathcal{G}A}{\gamma^2\mathcal{G}^2 - (\omega^2 L_i - i\omega\mathcal{R} - 1/C)M} \end{pmatrix}.$$

A.7.2 The $(2+2n) \times (2+2n)$ matrix

The extended n-coil system (6.13) can also be solved using the Schur complement. Here, the block representation is

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} (\hat{W}, \hat{Z})^T \\ \left((\hat{Q}_1, \hat{I}_1)^T, \ldots, (\hat{Q}_n, \hat{I}_n)^T\right)^T \end{pmatrix} = \begin{pmatrix} (A, 0)^T \\ \left((0,0)^T, \ldots, (0,0)^T\right)^T \end{pmatrix},$$

where

$$\mathbf{A} = \begin{pmatrix} i\omega M & 0 \\ -1 & i\omega \end{pmatrix},$$

$$\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_n) \text{ with } \mathbf{B}_j = \gamma\mathcal{G}_j \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

$$\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_n)^T \text{ with } \mathbf{C}_j = -\gamma \mathcal{G}_j \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

$$\mathbf{D} = \mathrm{diag}(\mathbf{D}_1, \ldots, \mathbf{D}_n) \text{ with } \mathbf{D}_j = \begin{pmatrix} i\omega & -1 \\ \frac{1}{C} & \frac{i\omega\mathcal{L}+\mathcal{R}}{n} \end{pmatrix} \text{ for all } j.$$

Recall the definition of the Schur complement: $\mathbf{S} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$. Thankfully, because of the precise forms of the block matrices, $\mathbf{S}$ is not too difficult to obtain, even for generic n.

First, because $\mathbf{D}$ is block-diagonal, its inverse is straightforward:

$$\mathbf{D}^{-1} = \mathrm{diag}(\mathbf{D}_1^{-1}, \ldots, \mathbf{D}_n^{-1}) \text{ with } \mathbf{D}_j^{-1} = \frac{n}{-\omega^2\mathcal{L} + i\omega\mathcal{R} + {}^{n}/_{C}} \begin{pmatrix} \frac{i\omega\mathcal{L}+\mathcal{R}}{n} & 1 \\ -\frac{1}{C} & i\omega \end{pmatrix}.$$

Now, consider $\mathbf{D}^{-1}\mathbf{C}$. Since $\mathbf{D}^{-1}$ is also block-diagonal, we simply have

$$\mathbf{D}^{-1}\mathbf{C} = (\mathbf{D}_1^{-1}\mathbf{C}_1, \ldots, \mathbf{D}_n^{-1}\mathbf{C}_n)^T \text{ with } \mathbf{D}_j^{-1}\mathbf{C}_j = \frac{n\gamma\mathcal{G}_j}{\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C}} \begin{pmatrix} 1 & 0 \\ i\omega & 0 \end{pmatrix}.$$

Then, $\mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ is a 2×2 matrix whose ik^{th} entry is given by

$$(\mathbf{B}\mathbf{D}^{-1}\mathbf{C})_{ik} = \sum_{j=1}^{2n} \mathbf{B}_{ij}(\mathbf{D}^{-1}\mathbf{C})_{jk}.$$

Note that, for all j, $\mathbf{B}_{ij}$ is zero for $i = 2$ (row 2), while $(\mathbf{D}^{-1}\mathbf{C})_{jk}$ is zero for $k = 2$ (column 2). This means, regardless of n, the only nonzero entry of $\mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ is for $i = k = 1$. Similarly, $\mathbf{B}_{1j}$ is zero for odd j (odd-numbered columns), leaving

$$\mathbf{B}\mathbf{D}^{-1}\mathbf{C} = \begin{pmatrix} \sum_{j=1}^{n} \mathbf{B}_{1(2j)}(\mathbf{D}^{-1}\mathbf{C})_{(2j)1} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{i\omega n\gamma^2}{\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}/_{C}} \sum_{j=1}^{n} \mathcal{G}_j^2 & 0 \\ 0 & 0 \end{pmatrix}.$$

Therefore, the Schur complement is

$$\mathbf{S} = \begin{pmatrix} i\omega\left(M - \frac{n\gamma^2}{\omega^2\mathcal{L} - i\omega\mathcal{R} - n/C}\sum_j \mathcal{G}_j^2\right) & 0 \\ -1 & i\omega \end{pmatrix},$$

and again we can solve

$$\mathbf{S}\begin{pmatrix} \hat{W} \\ \hat{Z} \end{pmatrix} = \begin{pmatrix} A \\ 0 \end{pmatrix}$$

to find

$$\begin{aligned} \hat{W} &= \frac{i(\omega^2\mathcal{L} - i\omega\mathcal{R} - n/C)A}{\omega\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - n/C)M\right)}; \\ \hat{Z} &= \frac{(\omega^2\mathcal{L} - i\omega\mathcal{R} - n/C)A}{\omega^2\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - n/C)M\right)}. \end{aligned}$$

Finally, consider the vector of unknown charges and currents $(\hat{Q}_1, \hat{I}_1, \ldots, \hat{Q}_n, \hat{I}_n)^T$; this can be calculated using

$$\left((\hat{Q}_1, \hat{I}_1)^T, \ldots, (\hat{Q}_n, \hat{I}_n)^T\right)^T = -\mathbf{D}^{-1}\mathbf{C}\begin{pmatrix} \hat{W} \\ \hat{Z} \end{pmatrix} = -(\mathbf{D}_1^{-1}\mathbf{C}_1, \ldots, \mathbf{D}_n^{-1}\mathbf{C}_n)^T\begin{pmatrix} \hat{W} \\ \hat{Z} \end{pmatrix}.$$

Because of the structure of the blocks due to the lack of coupling between the individual coils, we obtain a separate matrix equation for each:

$$\begin{aligned} \begin{pmatrix} \hat{Q}_j \\ \hat{I}_j \end{pmatrix} &= -\mathbf{D}_j^{-1}\mathbf{C}_j\begin{pmatrix} \hat{W} \\ \hat{Z} \end{pmatrix} \\ &= \frac{n\gamma\mathcal{G}_j A}{\omega\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - n/C)M\right)}\begin{pmatrix} -i \\ \omega \end{pmatrix}, \end{aligned}$$

from which we obtain the solutions to (6.13):

$$\begin{pmatrix} \hat{W} \\ \hat{Z} \\ \hat{Q}_j \\ \hat{I}_j \end{pmatrix} = \begin{pmatrix} \dfrac{i(\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}\!/_{C})A}{\omega\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}\!/_{C})M\right)} \\ \dfrac{(\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}\!/_{C})A}{\omega^2\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}\!/_{C})M\right)} \\ \dfrac{-in\gamma\mathcal{G}_j A}{\omega\left(n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}\!/_{C})M\right)} \\ \dfrac{n\gamma\mathcal{G}_j A}{n\gamma^2\sum_j \mathcal{G}_j^2 - (\omega^2\mathcal{L} - i\omega\mathcal{R} - {}^{n}\!/_{C})M} \end{pmatrix}.$$

A.8 Integral rules

A.8.1 3D integration-by-parts

For a scalar f and a vector $\mathbf{F} = (F_x, F_y, F_z)$, a 3D IBP transferring the gradient of f into the divergence of $\mathbf{F}$ can be derived by combining the divergence theorem and the product rule on $\nabla \cdot (f\mathbf{F})$:

$$\int_0^{L_x}\int_R^{l_y}\int_0^{h} \{\mathbf{F}\cdot\nabla f + f\nabla\cdot\mathbf{F}\}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x = \iint_{\partial\Omega} f\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S,$$

where $\partial\Omega$ is the boundary of the domain, and $\hat{\mathbf{n}}$ and $\mathrm{d}S$ are, respectively, the outward-pointing unit-normal and surface-element of each boundary.

The surface integral can be split into three parts: the moving wavemaker (at $y = R$), denoted $\partial\Omega_R$; the free surface (at $z = h$), denoted $\partial\Omega_h$; and the remaining solid and stationary boundaries (the four stationary walls and the base), denoted $\partial\Omega_w$. We then have

$$\begin{aligned}\int_0^{L_x}\int_R^{l_y}\int_0^{h} \{\mathbf{F}\cdot\nabla f + f\nabla\cdot\mathbf{F}\}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x \\ = \iint_{\partial\Omega_R} f\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S + \iint_{\partial\Omega_h} f\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S + \iint_{\partial\Omega_w} f\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S.\end{aligned}$$

At the wavemaker, we have simply $\hat{\mathbf{n}}\,\mathrm{d}S = (0,-1,0)\,\mathrm{d}z\,\mathrm{d}x$. However, as the free surface

is uneven in x and y, the integral over $\partial\Omega_h$ can instead be written on the flat x-y plane by using the following: the free surface can be parameterised by the position vector $\mathbf{r} = (x, y, h(x, y, t))$, which has a unit-normal and surface-element given by

$$\hat{\mathbf{n}}|_{z=h} = \frac{\frac{\partial \mathbf{r}}{\partial x} \times \frac{\partial \mathbf{r}}{\partial y}}{\left\| \frac{\partial \mathbf{r}}{\partial x} \times \frac{\partial \mathbf{r}}{\partial y} \right\|} \text{ and } \mathrm{d}S = \left\| \frac{\partial \mathbf{r}}{\partial x} \times \frac{\partial \mathbf{r}}{\partial y} \right\| \mathrm{d}y\,\mathrm{d}x,$$

giving

$$\begin{aligned} \hat{\mathbf{n}}\,\mathrm{d}S &= \left(\frac{\partial \mathbf{r}}{\partial x} \times \frac{\partial \mathbf{r}}{\partial y} \right) \mathrm{d}y\,\mathrm{d}x \\ &= \left(1, 0, \frac{\partial h}{\partial x} \right) \times \left(0, 1, \frac{\partial h}{\partial y} \right) \mathrm{d}y\,\mathrm{d}x \\ &= \left(-\frac{\partial h}{\partial x}, -\frac{\partial h}{\partial y}, 1 \right) \mathrm{d}y\,\mathrm{d}x \\ &= (-\nabla_H h, 1)\,\mathrm{d}y\,\mathrm{d}x. \end{aligned}$$

Using these expressions for the two moving components of $\hat{\mathbf{n}}\,\mathrm{d}S$, we can transfer the gradient of f into the divergence of $\mathbf{F}$ as follows:

$$\begin{aligned} &\int_0^{L_x} \int_R^{l_y} \int_0^h \mathbf{F} \cdot \nabla f \,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x \\ &\qquad = -\int_0^{L_x} \int_R^{l_y} \int_0^h f \nabla \cdot \mathbf{F} \,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x - \int_0^{L_x} \left\{ \int_0^h f F_y \,\mathrm{d}z \right\} \Bigg|_{y=R} \mathrm{d}x \\ &\qquad\qquad + \int_0^{L_x} \int_R^{l_y} (f\mathbf{F})|_{z=h} \cdot (-\nabla_H h, 1)\,\mathrm{d}y\,\mathrm{d}x + \iint_{\partial\Omega_w} f\mathbf{F} \cdot \hat{\mathbf{n}}\,\mathrm{d}S. \end{aligned} \quad \text{(A.48)}$$

A.8.2 2D integration-by-parts

In the 2D domain, $\mathbf{F}$ reduces to $\mathbf{F} = (F_x, F_y)$ and the combination of the product rule and divergence theorem gives

$$\int_0^{L_x} \int_R^{l_y} \{\mathbf{F} \cdot \nabla f + f \nabla \cdot \mathbf{F}\}\,\mathrm{d}y\,\mathrm{d}x = \int_\Gamma f\mathbf{F} \cdot \hat{\mathbf{n}}\,\mathrm{d}s,$$

where Γ is the horizontal line around the tank, and $\mathrm{d}s$ is its line-element. After splitting Γ into the wavemaker $y = R$—where $\hat{\mathbf{n}}\,\mathrm{d}s = (0, -1)\,\mathrm{d}x$—and the remaining four lines

Γ_w, we have

$$\int_0^{L_x}\int_R^{l_y} \mathbf{F}\cdot\nabla f \,\mathrm{d}y\,\mathrm{d}x = -\int_0^{L_x}\int_R^{l_y} f\nabla\cdot\mathbf{F}\,\mathrm{d}y\,\mathrm{d}x - \int_0^{L_x}(fF_y)|_{y=R}\,\mathrm{d}x + \int_{\Gamma_w} f\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}s. \quad \text{(A.49)}$$

A.8.3 Green's identities (2D)

For the special case of $\mathbf{F} = \nabla g$, (A.49) reduces to Green's first identity (in 2D):

$$\int_0^{L_x}\int_R^{l_y} \nabla g\cdot\nabla f \,\mathrm{d}y\,\mathrm{d}x = -\int_0^{L_x}\int_R^{l_y} f\nabla^2 g\,\mathrm{d}y\,\mathrm{d}x - \int_0^{L_x}\left\{f\frac{\partial g}{\partial y}\right\}\Big|_{y=R}\mathrm{d}x + \int_{\Gamma_w} f\nabla g\cdot\hat{\mathbf{n}}\,\mathrm{d}s, \quad \text{(A.50)}$$

where the gradient of f has been transferred into the Laplacian of g.

Now, consider the special case $f = 1$ and $\mathbf{F} = g\nabla h - h\nabla g$. By the product rule we have

$$\nabla\cdot(f\mathbf{F}) = \nabla\cdot(g\nabla h - h\nabla g) = \cancel{\nabla g\cdot\nabla h} + g\nabla^2 h - \cancel{\nabla h\cdot\nabla g} - h\nabla^2 g,$$

and (A.49) reduces to Green's second identity in 2D:

$$\int_0^{L_x}\int_R^{l_y} g\nabla^2 h\,\mathrm{d}y\,\mathrm{d}x = \int_0^{L_x}\int_R^{l_y} h\nabla^2 g\,\mathrm{d}y\,\mathrm{d}x + \int_\Gamma (g\nabla h - h\nabla g)\cdot\hat{\mathbf{n}}\,\mathrm{d}s, \quad \text{(A.51)}$$

where the Laplacian of h has been transferred into the Laplacian of g, and the two boundaries $y = R$ and Γ_w have been recombined into Γ.

A.8.4 3D Reynolds transport theorem

Due to the time-dependence of the limits of integration, time derivatives and spatial integrals do not commute. Instead, via the RTT, surface integrals appear when a time derivative is taken inside a spatial integral. The theorem reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_0^{L_x}\int_R^{l_y}\int_0^{h} f\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\right) = \int_0^{L_x}\int_R^{l_y}\int_0^{h}\frac{\partial f}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x + \iint_{\partial\Omega} f\mathbf{U}\cdot\hat{\mathbf{n}}\,\mathrm{d}S,$$

where $\mathbf{U}$ is the velocity of the surface $\partial\Omega$. Since $\partial\Omega_w$ is stationary, the surface integral is now only split in two:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_0^{L_x}\int_R^{l_y}\int_0^h f\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\right) = \int_0^{L_x}\int_R^{l_y}\int_0^h \frac{\partial f}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x + \iint_{\partial\Omega_R} f\mathbf{U}_R\cdot\hat{\mathbf{n}}\,\mathrm{d}S + \iint_{\partial\Omega_h} f\mathbf{U}_h\cdot\hat{\mathbf{n}}\,\mathrm{d}S.$$

Now, the surfaces have velocities $\mathbf{U}_R = (0,\dot{R},0)$ and $\mathbf{U}_h = (0,0,{}^{\partial h}/_{\partial t})$, so that

$$\mathbf{U}_R\cdot\hat{\mathbf{n}}\,\mathrm{d}S = -\dot{R}\,\mathrm{d}z\,\mathrm{d}x \;\text{ and }\; \mathbf{U}_h\cdot\hat{\mathbf{n}}\,\mathrm{d}S = \frac{\partial h}{\partial t}\,\mathrm{d}y\,\mathrm{d}x.$$

These expressions for $\mathbf{U}\cdot\hat{\mathbf{n}}\,\mathrm{d}S$ then give

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_0^{L_x}\int_R^{l_y}\int_0^h f\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\right) = \int_0^{L_x}\int_R^{l_y}\int_0^h \frac{\partial f}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x - \int_0^{L_x}\left\{\int_0^h f\dot{R}\,\mathrm{d}z\right\}\Big|_{y=R}\mathrm{d}x + \int_0^{L_x}\int_R^{l_y} f|_{z=h}\frac{\partial h}{\partial t}\,\mathrm{d}y\,\mathrm{d}x.$$

Since in a Lagrangian a total time derivative can be discarded, the RTT results in the transferral of the time derivative of f into time-dependent boundary terms:

$$\int_0^{L_x}\int_R^{l_y}\int_0^h \frac{\partial f}{\partial t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x = \int_0^{L_x}\left\{\int_0^h f\dot{R}\,\mathrm{d}z\right\}\Big|_{y=R}\mathrm{d}x - \int_0^{L_x}\int_R^{l_y} f|_{z=h}\frac{\partial h}{\partial t}\,\mathrm{d}y\,\mathrm{d}x. \quad (A.52)$$

A.8.5 2D Reynolds transport theorem

In the 2D domain, the RTT reduces to

$$\int_0^{L_x}\int_R^{l_y}\frac{\partial f}{\partial t}\,\mathrm{d}y\,\mathrm{d}x = -\int_\Gamma f\mathbf{U}\cdot\hat{\mathbf{n}}\,\mathrm{d}s.$$

The only moving line is $y = R$, where $\mathbf{U}_R\cdot\hat{\mathbf{n}}\,\mathrm{d}s = -\dot{R}\,\mathrm{d}x$, such that

$$\int_0^{L_x}\int_R^{l_y}\frac{\partial f}{\partial t}\,\mathrm{d}y\,\mathrm{d}x = \dot{R}\int_0^{L_x} f|_{y=R}\,\mathrm{d}x. \quad (A.53)$$