Components of Nonlinear Oscillation and Optimal Averaging for Stiff PDEs

ADAM GEOFFREY PEDDLE

I MUST GO DOWN TO THE SEAS AGAIN, TO THE LONELY SEA AND SKY,

AND ALL I ASK IS A TALL SHIP AND A STAR TO STEER HER BY;

AND THE WHEEL'S KICK AND THE WIND'S SONG AND THE WHITE SAIL'S SHAKING,

AND A GREY MIST ON THE SEA'S FACE, AND A GREY DAWN BREAKING.

I MUST GO DOWN TO THE SEAS AGAIN, FOR THE CALL OF THE RUNNING TIDE

IS A WILD CALL AND A CLEAR CALL THAT MAY NOT BE DENIED;

AND ALL I ASK IS A WINDY DAY WITH THE WHITE CLOUDS FLYING,

AND THE FLUNG SPRAY AND THE BLOWN SPUME, AND THE SEAGULLS CRYING.

I MUST GO DOWN TO THE SEAS AGAIN, TO THE VAGRANT GYPSY LIFE,

TO THE GULL'S WAY AND THE WHALE'S WAY AND THE WIND LIKE A WHETTED KNIFE;

AND ALL I ASK IS A MERRY YARN FROM A LAUGHING FELLOW ROVER,

AND A QUIET SLEEP AND A SWEET DREAM WHEN THE LONG TRICK'S OVER.

JOHN MASEFIELD

Components of Nonlinear Oscillation and Optimal Averaging for Stiff PDEs

Submitted by Adam Peddle to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Mathematics, January 2018.

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(Signature) .....

Copyright © 2018 Adam Geoffrey Peddle
Cover photo by the author

PUBLISHED BY THE UNIVERSITY OF EXETER

First printing, April 2018

Abstract

A novel solver which uses finite wave averaging to mitigate oscillatory stiffness is proposed and analysed. We have found that triad resonances contribute to the oscillatory stiffness of the problem and that they provide a natural way of understanding stability limits and the role averaging has on reducing stiffness. In particular, an explicit formulation of the nonlinearity gives rise to a *stiffness regulator function* which allows for analysis of the wave averaging.

A practical application of such a solver is also presented. As this method provides large timesteps at comparable computational cost but with some additional error when compared to a full solution, it is a natural choice for the coarse solver in a Parareal-style parallel-in-time method.

Acknowledgements

I owe a tremendous debt of gratitude to my supervisors, Beth Wingate and Peter Ashwin, for their continued support and guidance in preparing this work. Their suggestions and ideas were vital to the progress of the research, and their patience and enthusiasm was boundless. In addition to them, I would like to thank my academic mentor, Andrew Gilbert, and my assessors, Joanne Mason and John Thuburn. I am grateful for their support and advice.

In addition to the official supervisory staff, there are several people who provided far more assistance than they could reasonably have been expected to. In particular, I would like to thank Terry Haut for countless early morning Skype sessions and a very productive few weeks in New Mexico. At Exeter, Pedro Peixoto's instruction on geophysics will not be soon forgotten. Finally, thanks go to Peter Challenor for his guidance on statistics, and for making the topic so accessible.

I would like to thank my office colleagues and friends, Jamie Penn and Alex Owen, not only for their detailed readings of my thesis, but also for the enjoyable and enlightening discussions, mathematical and otherwise, and for making Room 319 such a pleasant place to spend three years.

My parents, Geoff and Kathy, have been endlessly loving and supportive, not just throughout this folly but throughout my life. How wonderful to write a thesis so I can finally thank them in print. I would be remiss not to mention my brother, Ben, whose sense of humour knows no living parallel. I am endlessly grateful to my partner, Magda, for her love, support, and vepřoknedlozelo throughout the research and writing that went into this thesis. Truly, her patience knows no bounds.

Looking farther back, I would like to thank the Bruntons: Mark, Liz, Conor, Peter, and Darren, for harbouring me at their home in Ireland while I waited for a visa, and for far longer than we all initially expected. Long overdue thanks go to Johan Bosschers, with whom I had the pleasure of working for two years. His expert guidance on numerical programming has served me well ever since, and is, I hope, well reflected in the software which was written for this project.

Finally, I would like to acknowledge the support of the College of Engineering, Mathematics, and Physical Sciences at the University of Exeter for providing financial support for this work to take place.

Contents

List of Figures

List of Symbols

α	Denotes branches of the dispersion relation
β	Lipschitz constant
ΔT	The coarse timestep
Δt	General or fine timestep
η	The length of a finite averaging window
η^*	The perturbation height of fluid
$\hat{\mathbf{u}}_k$	The k-th Fourier mode of $\mathbf{u}$
$\mathbf{k}$	A vector wavenumber
λ	The eigenvalue of an operator or matrix
$\Lambda(\eta)$	Stiffness regulator function
$\mathbb{C}$	The space of complex numbers
$\mathbb{N}$	The space of natural numbers
$\mathbb{R}$	The space of real numbers
$\mathbb{Z}$	The space of integers
$\mathbf{f}$	The Coriolis frequency
$\mathbf{f}(\cdot)$	A general vector-valued function
$\mathbf{I}$	An identity matrix
$\mathbf{r}_k$	k-th right eigenvector of $\mathcal{L}$
$\mathcal{D}$	Dissipative operator
$\mathcal{F}$	The Fourier transform

$\mathcal{F}^{-1}$	The inverse Fourier transform
$\mathcal{G}$	Gravitational linear operator
$\mathcal{L}$	Linear operator
$\mathcal{N}$	Nonlinear operator
$\mathcal{R}$	Rotational linear operator
$\mathcal{S}_{\mathbf{k},\alpha}$	The resonant set
$\mathcal{S}_{\mathbf{k},\alpha}^{\epsilon_\beta}$	The ϵ_β-near resonant set
B	The Burger number
Fr	The Froude number
Ro	The Rossby number
Ω	The triad sum
ω	The dispersion relation
$\overline{\mathcal{D}}$	Averaged Dissipative operator
$\overline{\mathcal{N}}$	Averaged Nonlinear operator
∂_x	Partial derivative with respect to x. Similarly y and t
ϕ	The geopotential height
Φ_0	The mean geopotential height
$\rho(s)$	A smooth kernel of integration
$\sigma_{\mathbf{k}}^{\alpha}$	Fourier coefficient in the eigenbasis of $\mathcal{L}$
τ	Denotes the fast timescale
ε	A sometimes-small quantity. Stands in for Rossby number.
$\mathbf{u}$	A vector unknown quantity
$C_{(\cdot)}$	An arbitrary constant
$C_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2}^{\alpha,\alpha_1,\alpha_2}$	Interaction coefficient
D	A Lipschitz-continuous subspace of $\mathbb{R}^n$

e	Euler's number
F	The inverse of the Burger number
$f(\cdot)$	A general scalar-valued function
g	Acceleration due to gravity
h	The total height of fluid
H_0	The mean water depth
i	The imaginary number
k	A scalar wavenumber
L	The side length of the spatial domain
M	The L_∞-norm over the right-hand side of some nonlinear hyperbolic system
N_x	The number of points in a spatial discretisation
p	Denotes the order of convergence of a numerical method
$P(n, \eta)$	Prokopian function
s	A dummy time variable
u	A scalar unknown quantity

1 Introduction

> 'Begin at the beginning,' the King said, very gravely,
> 'and go on till you come to the end: then stop.'
>
> C. S. Lewis, Alice in Wonderland

Numerical simulations are and continue to become more important for prediction, such as forecasting of the weather, and for investigation and research, as in their role in scientific simulations of climate. As these simulations become ever more complex, the role of high-performance computing (HPC) becomes more relevant. HPC is the design and implementation of algorithms to efficiently perform computations, increasingly in a parallel fashion.

Problems of the type which arise in geophysical fluid dynamics often exhibit *oscillatory stiffness* (q.v. Section 1.1 below), which refers to a restriction of the timestep size due to the presence of rapid oscillations in the solution. When solving problems of this type, it is necessary to either provide sufficient computational power to resolve the necessary scales or to model the effect of them. Specifically considering geophysical problems which are modelled with PDEs of hyperbolic or parabolic type, where we are interested in modelling the time evolution of some quantities which are defined on a spatio-temporal domain (Vallis, 2006) a common approach is to sub-divide the spatial domain into continuously smaller blocks which may be handled on separate processors, and which communicate at their boundaries – so-called *domain decomposition* (Gropp, 1992; Minkoff, 2002).

Increasing the number of processors across which the problem is distributed increases the speed, but does so subject to a law of diminishing returns due to various scaling restrictions arising from communications bottlenecks, non-uniform problem sizes, serial portions, etc. (McCool et al., 2012). The restriction on parallel speedup due to serial portions of the algorithm in particular is known as *Amdahl's Law* (Rodgers, 1985) and spectral methods, such as those which we will use here, suffer particularly from this (Temperton, 2000).

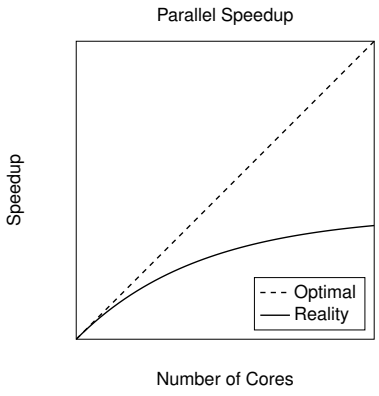

Figure 1.1: The limitations of the parallel model. Ideally, doubling the number of processors would cause the program to run twice as fast (2× the processors, 2× the speed) as with the dashed line. In practice, parallel algorithms follow a law of diminishing returns, depicted by the solid line.

1.1 Numerical Stiffness

Stiffness is a property of differential equations which makes their numerical solution by standard methods difficult, first discovered by Curtiss and Hirschfelder (1952) who coined the term 'stiffness'. They also developed the first backward differentiation formulas to handle it. In the years since, there have been many advances, but the problem of stiffness is still with us. This is due to the difficulties in handling nonlinearity in stiff PDEs, a limitation which this thesis addresses.

There exist myriad definitions for stiffness, many of which are particular to a given method or system. In order to describe what stiffness is and to develop a more general understanding, consider an ODE of the form

$$\frac{\mathrm{d}u}{\mathrm{d}t} = f(t,u), \tag{1.1}$$

as well as an explicit timestepping algorithm,

$$u_n = u_{n-1} + \Delta t f(t_{n-1}, u_{n-1}), \quad n = 1, 2, \ldots, N. \tag{1.2}$$

If we assume that the solution is Lipschitz continuous, i.e.

$$|f(t,y) - f(t,x)| \leq \beta |y - x|, \tag{1.3}$$

for some finite constant, β, then a stiff problem is one for which $\beta \Delta t \gg 1$ (Spijker, 1996). While this definition is useful numerically, in the interest of developing intuition it is worth considering some symptoms of stiff problems, following Trefethen (1996). With stiff problems:

1. there is a large variation in timescales;
2. stability is a greater constraint on the timestep than accuracy;
3. implicit methods perform significantly better than explicit methods.

It is important to understand that stiffness as used throughout this work is a *purely numerical* phenomenon which does not arise in the analytical solution of differential equations. According to Higham and Trefethen (1993),

> Instability and stiffness are transient phenomena, involving finite time intervals $[t_0, t_1]$. They cannot be characterised by considering only the limits $t \to \infty$ or $t \to t_0$.

This relationship to finite time intervals is an important aspect of numerical stiffness, and will feature significantly in our attempts to understand and mitigate numerical stiffness.

The third symptom in the list above is used as the definition of a stiff problem by Hochbruck and Ostermann (2010). For the case of *dissipatively stiff* problems (*cf.* Figure 1.2), this definition holds due to the stability requirements imposed by the CFL condition. In a dissipatively stiff problem, the right hand side of equation (1.1) has large, negative, real eigenvalues such that the gradient on the right-hand side is large. This leads directly to a requirement of small timesteps in order to resolve the rapidly-varying flow. However, we are in this work primarily concerned with problems displaying *oscillatory stiffness*, for which this definition is inadequate.

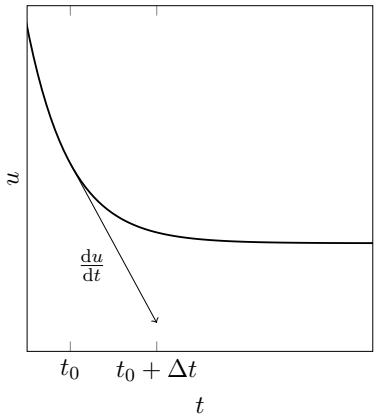

Figure 1.2: Dissipative stiffness. The derivative evaluated at some point in time, t_0, provides a very poor approximation to the solution, $u(t_1)$, that is over a timestep of Δt. This requires the use of a smaller timestep to resolve the gradient.

We may say that a problem exhibits oscillatory stiffness when the linearisation of equation (1.1) has purely imaginary eigenvalues of large modulus. Rather than giving rise to a steep gradient like dissipative stiffness does, this instead induces rapid oscillations which require tiny timesteps in order to resolve. Indeed, the allowable timestep for explicit Euler methods is reduced for problems which are stiff in the oscillatory sense due to stability limitations. However, the implicit Euler method is also an inefficient choice here. This is not due to stability requirements – implicit methods are universally stable (Trefethen, 1996) – but rather accuracy requirements.

Durran (2010) showed that implicit methods will poorly resolve the most rapid waves in the solution. If these poorly-resolved waves are not physically significant, implicit timestepping methods are a viable way of solving oscillatory stiff equations. However, as we will show in this work (*q.v.* Chapter 4) these rapid waves are relevant to the accuracy of the solution through their interaction with other waves. It is therefore important from the standpoint of accuracy to resolve both the fast and slow waves, leading us to look beyond implicit solvers for the problem at hand.

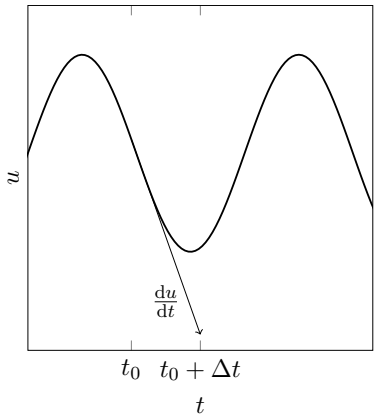

Figure 1.3: Oscillatory stiffness. As with Figure 1.2, the gradient is rapidly-varying and therefore poorly approximated over a timestep Δt as shown here, leading to a requirement of a smaller timestep to numerically approximate $u(t)$.

As oscillatory stiffness imposes a limit on the timestep and the practicalities of computing impose their own restrictions on the effectiveness of spatial parallelism, development of algorithms which are able to model the system under stiff conditions are necessary. We may propose an incomplete taxonomy of methods for handling oscillatory stiffness: one approach is to develop a model which permits the cost imposed by the oscillatory stiffness to be mitigated directly by novel parallelism. Another approach is to develop an algorithm which models the system in a non-stiff or less stiff fashion and to a sufficient degree of accuracy.

In this work, we describe a novel *coarse* solver which is based on the idea of fast-wave averaging, and which permits very long timesteps to be taken in a numerically stable fashion. As we will show in Chapters 3 and 4, the oscillatory stiffness in problems of the type given below arises through the fast nonlinear oscillations which are natural components of the solution. The fast-wave aver-

aging procedure which we introduce in Section 1.4 below is capable of reducing stiffness while maintaining accuracy because of its treatment of the *nonlinear oscillatory components*, as opposed to a more common linear wavespace filtering (e.g. Pope (2011); Sagaut (2011)).

This method is numerically tractable, in particular as the average itself is *embarrassingly parallel* in time and is particularly applicable to heterogeneous computing architectures. The averaged method carries with it a higher approximation error than a standard method would, but it directly enables the time-parallel simulation of oscillatory stiff PDEs.

This fast-wave averaged method allows the extension of the *Asymptotic Parallel in Time* (APinT) method (Haut and Wingate, 2014) to the case of finite scale separation. APinT uses the fast-wave averaged integrator developed and analysed in this work to implement a *Parareal* method (Lions et al., 2001) which is suitable for oscillatory stiff problems. The Parareal algorithm extends parallelism to the temporal domain and thus improves scalability beyond what is available spatially. The APinT method can be thought of as being the first member of the taxonomy given above, where the oscillatory stiffness is mitigated through a parallel computing model (*q.v.* Chapter 5).

In the interest of developing a general framework, we will consider an equation of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\varepsilon}\mathcal{L}\mathbf{u} + \mathcal{N}(\mathbf{u},\mathbf{u}) = \mathcal{D}\mathbf{u}, \tag{1.4}$$

where $\mathcal{L}$ is a linear operator which has purely imaginary eigenvalues, $\mathcal{N}(\mathbf{u},\mathbf{u})$ is a nonlinear operator of quadratic type, and $\mathcal{D}$ encodes the dissipation in the problem. The solution vector field, $\mathbf{u}$ is defined on the domain $\Omega \in \mathbf{x} \times [t_0, t_N]$. We assume that the dissipation is not sufficient to induce stiffness limitations, and restrict ourselves instead to this being an oscillatory-stiff problem. This is sufficient to describe many of the common equations of fluid dynamics, such as the Boussinesq equations and the Rotating Shallow Water Equations (RSWE) in many flow regimes.

1.2 *Some Historical Context*

Early numerical models of weather were restricted by several considerations, one of which was the timestep limitations imposed by the CFL condition (Lynch, 2008). In order to perform the first successful numerical weather prediction (NWP), a mathematically filtered model was used which completely eliminated the fast

oscillations from the solution.

This model led to the physical notion of 'slow' dynamics through the so-called *Quasi-Geostrophic* (QG) equations, due to Charney (1948). Based on scale analysis, they were a major advancement in the modelling of weather. These reduced equations allowed the fast waves, which were the source of the oscillatory stiffness, to be filtered while still resolving the large-scale motions of the fluid. The work was expanded upon in Charney (1949) and Charney and Phillips (1953).

The reduced equations have been rigorously shown to hold asymptotically in the limit of $\varepsilon \to 0$ (Embid and Majda, 1996). The QG equations, however, are not accurate enough for modern weather prediction. Instead, NWP relies on numerical approximations of the full equations of motion (Davies et al., 2003). The implication here is that if the QG equations are not accurate enough for prediction then at least some of the fast oscillations matter, even at large scales. This leaves us with the problem of finding some way to resolve the fast waves in order to capture the full dynamics.

Embid and Majda (1996), following Klainerman and Majda (1981), proposed a framework for fast-wave averaging in which the slow dynamics evolve independently of the fast, but the fast waves are not entirely eliminated from the system. This work is based on earlier work in averaging methods for nonlinear systems by Krylov and Bogoliubov (1935) and Bogoliubov and Mitropolsky (1961). While these methods were developed in the asymptotic regime as $\varepsilon \to 0$, an important advance in this work is the proof that such methods provide a basis for more general wave averaging outside of the QG limit.

Let us now introduce the Rotating Shallow Water Equations (RSWE), which are used for the practical computations in this work. The QG equations are derived directly from these, but will not be discussed in more detail here. For a more detailed discussion of the history of NWP, the reader is referred to Lynch (2008).

1.3 *The Rotating Shallow Water Equations*

In the practical examples throughout this work, we will use variations on the RSWE. Note that in the dimensional case, the parameter ε is absent, as it is effectively absorbed into the linear operator. In the interest of clarity, we will develop mathematical theory using the form given in equation (1.4), and perform some numerical experiments with the dimensional case. In the dimensional case, the theory still holds in common parameter regimes for

NWP, albeit with some loss of notational elegance.

Much of the theory which we will develop in this work holds for any equation in the form of equation (1.4). However, in the interest of performing numerical simulations and further illustrating certain specifics, we will consider just the RSWE. The RSWE are a hyperbolic system of partial differential equations, becoming parabolic if viscous effects are considered. We will discuss linear dissipation in theory, and limit ourselves to hyperviscosity for stability in numerical practice (Passot and Pouquet, 1988; Duchon and Robert, 1999). The RSWE are derived from the Navier-Stokes equations under the assumptions of constant fluid density and hydrostatic balance, and a horizontal scale which is much greater than the fluid depth.[1]

[1] Hence 'shallow'.

An important feature of these equations is that they exhibit *quadratic nonlinearity*[2], which gives rise to triadic interactions. We will return to these in more detail in Chapter 3. For now it is sufficent to understand that such interactions provide the discrete components of oscillation which serve as a natural atomic unit of solution. We will rely heavily on these in investigating the effects of averaging on the RSWE.

[2] The author recalls being told once by Dr. Alan Hegarty of the University of Limerick that fluid mechanics is simply the study of quadratic nonlinearity.

In less simple models of geophysical flows, the equation of state may give rise to nonlinearities which are not of quadratic – or even polynomial – type. This in turn requires the consideration of a different interaction class or approximation thereof. As long as such a model gives rise to a partially ordered and countable set of interactions, the results of this thesis related to averaging and stiffness mitigation in Chapter 4 hold.

A useful property of equations exhibiting quadratic nonlinearity is that we may order the interactions by their relevance to the long-term behaviour of the flow (Embid and Majda, 1996) and that this ordering is the same which allows us to describe the effects of averaging. We shall return to this in Section 1.4 below and in more detail in Chapter 3. In order for our analysis to apply to equations which are not quadratically nonlinear, it would be necessary that they have the same property. In the interest of making analytical progress, we limit ourselves to the RSWE and its assumption of quadratic nonlinearity from this point on.

We shall restrict ourselves to the single-layer case, where there is one layer of fluid, bounded below and with a fluid of negligible inertia above. According to Vallis (2006), this is one of the simplest useful models in geophysical fluid dynamics, as it allows for the study of the effects of rotation without the complications of stratification.

With reference to equation (1.4), the vector of unknowns for the

RSWE is

$$\mathbf{u} = (u(\mathbf{x},t), v(\mathbf{x},t), h(\mathbf{x},t))^T, \tag{1.5}$$

where the first two components are the fluid velocity fields in the x- and y-components, respectively, and the third component is the height of the fluid layer. Following Vallis (2006), we may write the RSWE in their dimensional form, neglecting dissipation, as the *momentum* equation

$$\frac{D\mathbf{v}}{Dt} + \mathbf{f} \times \mathbf{v} = -g\nabla h, \tag{1.6}$$

and the *mass* equation

$$\frac{\partial h}{\partial t} + \nabla \cdot (\mathbf{v}h) = 0, \tag{1.7}$$

where $\mathbf{v}(\mathbf{x},t) = (u(\mathbf{x},t), v(\mathbf{v}(\mathbf{x},t))$, $\mathbf{f} = f\hat{\mathbf{k}}$ is the Coriolis coefficient, g is the gravitational constant, and $\dfrac{D}{Dt}$ is the material derivative, defined as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla. \tag{1.8}$$

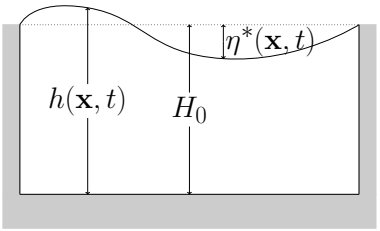

Figure 1.4: Schematic showing the domain of the rotating shallow water equations as well as the height field, broken into a constant and a perturbation component.

1.3.1 *Nondimensionalisation*

Let us now nondimensionalise the mass and momentum equations of the RSWE so as to obtain the system in the form given by equation (1.4). The total thickness of the fluid layer may be considered as the sum of the average water depth, H_0 and a perturbation height, η, as shown in Figure 1.4:

$$h(x,y,t) = H_0 + \eta(x,y,t). \tag{1.9}$$

Following Embid and Majda (1996), we now define characteristic scales for the length, time, depth, perturbation height, and velocity, such that:

$$\mathbf{v} = U\mathbf{v}^*; \; t = Tt^*; \; l = Ll^*, \tag{1.10}$$

and:

$$h = H(H_0^* + \theta\eta^*), \tag{1.11}$$

where the asterisk superscript denotes a dimensionless quantity. Note that the total water depth, h, is written in terms of a perturbation depth, η and a mean water depth, H_0. With these scalings, we may write equation (1.6) as

$$\frac{U}{T}\frac{\partial \mathbf{v}^*}{\partial t^*} + \frac{U^2}{L}(\mathbf{v}^* \cdot \nabla)\mathbf{v}^* + U\mathbf{f} \times \mathbf{v}^* = \frac{-gH\theta}{L}\nabla\eta^*. \tag{1.12}$$

We then define the timescale, T, to be $T = L/U$, and further define the *Rossby number*, Ro, as the ratio of convective to rotational forces:

$$Ro = \frac{U}{fL}. \tag{1.13}$$

This allows us to write

$$Ro\left[\frac{\partial \mathbf{v}^*}{\partial t^*} + (\mathbf{v}^* \cdot \nabla)\mathbf{v}^*\right] + \hat{\mathbf{k}} \times \mathbf{v}^* = -\frac{gH\theta}{fUL}\nabla\eta^*. \tag{1.14}$$

We now define the Froude and Burger numbers:

$$Fr = \frac{U}{\sqrt{gH}}, \tag{1.15}$$

and

$$B = \frac{Ro^2}{Fr^2}, \tag{1.16}$$

and so we let

$$\theta = Ro^{-1}Fr^2, \tag{1.17}$$

be the scaling on the perturbation height. Dropping the asterisks as is customary[3], the nondimensional formulation of the momentum equation (1.6) becomes

$$Ro\left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}\right] + \hat{\mathbf{k}} \times \mathbf{v} = -\nabla\eta^*. \tag{1.18}$$

[3] We will retain the asterisk on η^*. This is to prevent confusion when using η to represent the length of the averaging window in later chapters, in the interest of consistency with the HMM literature.

We now move onto the mass equation, equation (1.7). We apply the same scales as in the above derivation for time and perturbation height, and write the water depth in its expanded form. Thus, we may say that

$$\frac{HRoB^{-1}}{T}\frac{\partial \eta^*}{\partial t} + \frac{UH}{L}\nabla \cdot (H_0\mathbf{v}^*) + \frac{UHRoB^{-1}}{L}\nabla \cdot (\mathbf{v}^*\eta^*) = 0. \tag{1.19}$$

Dividing across by $\frac{UH}{L}$ and applying the same time scaling as before, we may write the dimensionless formulation of the mass equation, again dropping the asterisks

$$RoB^{-1}\left[\frac{\partial \eta^*}{\partial t} + \nabla \cdot (\mathbf{v}\eta^*)\right] + \nabla \cdot (H_0\mathbf{v}) = 0. \tag{1.20}$$

Then applying the scaling for θ yields

$$RoB^{-1}\left[\frac{\partial \eta^*}{\partial t} + \nabla \cdot (\mathbf{v}\eta^*)\right] + \nabla \cdot \mathbf{v} = 0. \tag{1.21}$$

All that is left for us to do is to let $\varepsilon = Ro$ and to define $F = B^{-1} = \mathcal{O}(1)$ to obtain the desired form (equation (1.4)). In the limit as $\varepsilon \to 0$, these equations are called the *rapidly rotating shallow water equations*. The linear operator, $\mathcal{L}$, takes the form:

$$\mathcal{L}\mathbf{u} = \begin{bmatrix} 0 & -1 & F^{-1/2}\partial_x \\ 1 & 0 & F^{-1/2}\partial_y \\ F^{-1/2}\partial_x & F^{-1/2}\partial_y & 0 \end{bmatrix} \begin{bmatrix} u(\mathbf{x},t) \\ v(\mathbf{x},t) \\ \eta^*(\mathbf{x},t) \end{bmatrix}, \tag{1.22}$$

and the nonlinear operator is:

$$\mathcal{N}(\mathbf{u},\mathbf{u}) = \begin{bmatrix} \mathbf{v} \cdot \nabla\mathbf{v} \\ \nabla \cdot (\eta^*\mathbf{v}) \end{bmatrix}. \tag{1.23}$$

1.3.2 The 1-D Equations

The RSWE are normally solved on a 2-dimensional spatial grid, $\mathbf{x} = x_n \times y_n$, $n = 0, 1 \ldots, N$. For a pseudospectral method (*cf.* Appendix A) on a mesh of size N^2, this has a complexity in the FFTs of $\mathcal{O}(N^2 \log(N))$ and of $\mathcal{O}(N^2)$ for the nonlinear evaluation in real space. In the interest of performing parameter studies quickly (as well as applying some computationally intense algorithms such as the brute resonance filter of Section 3.5) it is useful to have a computationally cheaper model.

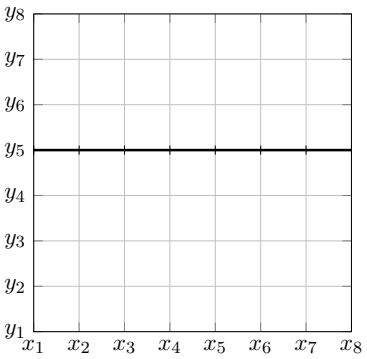

Figure 1.5: The computational domain for the RSWE. The 2-D grid is shown for $N = 8$ with grey gridlines. The 1-D reduction of this domain is depicted with a thick black line.

We may then consider the RSWE defined on a 1-dimensional mesh, $\mathbf{x} = x_n$, $n = 0, 1, \ldots, N$, but retaining the velocities in both directions and therefore retaining rotational effects. This reduces the FFT time complexity to $\mathcal{O}(N \log(N))$ and the time complexity of the nonlinear multiplication in space to $\mathcal{O}(N)$. The 1-D RSWE can be thought of as a single spatial slice out of the full 2-D RSWE. These reduced equations are sometimes called the *1-D St Venant equations*, although we retain rotational effects here which the St Venant equations generally do not. The modified linear and nonlinear operators become as follows, while retaining all important properties (skew-Hermicity, quadraticity, etc.) of the full 2-D RSWE

$$\mathcal{L} = \begin{bmatrix} 0 & -1 & F^{-1/2}\partial_x \\ 1 & 0 & 0 \\ F^{-1/2}\partial_x & 0 & 0 \end{bmatrix}, \tag{1.24}$$

and

$$\mathcal{N}(\mathbf{v}, \mathbf{v}) = \begin{bmatrix} u\dfrac{\partial u}{\partial x} \\ u\dfrac{\partial v}{\partial x} \\ \dfrac{\partial(\eta u)}{\partial x} \end{bmatrix}. \tag{1.25}$$

In effect, the derivatives in the y-direction have been lost. The component of the flow velocity in the y-direction has been retained and the unknown vector, $\mathbf{u}$ still has three components. The distinction is that these components are each of length N in the 1-D case, and of size $N \times N$ in the 2-D.

1.4 Fast Wave Averaging

At the core of the work done in this thesis is fast-wave averaging. Following Embid and Majda (1996) and speaking asymptotically we may derive a reduced system of equations which models

the RSWE but with less oscillatory stiffness. The form of this solution permits wide-ranging conclusions to be drawn about the dynamical nature of the flow and in particular the elements of its nonlinear oscillation. Outside of the asymptotic limit, where scale separation is finite, it is also possible to apply a modified version of the average derived below and to develop a fast and accurate approximate solver based on this theory.

Consider the rapidly-rotating shallow water equations, neglecting dissipation:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\varepsilon}\mathcal{L}\mathbf{u} + \mathcal{N}(\mathbf{u}, \mathbf{u}) = 0. \tag{1.26}$$

In practice, since $\mathcal{L}$ is a wave operator, as the parameter ε gets small it induces increasingly rapid oscillations and leads to more oscillatory stiffness, i.e. a restriction on the numerical step size. As discussed in Section 1.2, models of fluid flow in the atmosphere and oceans as $\varepsilon \to 0$ have been developed which have permitted advances in the analytical and numerical modelling of fluid flow. Following Embid and Majda (1996), we proceed by applying the method of multiple scales (Hinch, 1991) to equation (1.26). (See also Klainerman and Majda (1981), Majda (2002), and Haut and Wingate (2014).)

Let the temporal derivative be the sum of two separated timescales: a slow timescale, t', and a fast timescale, τ. In this context, ε is a measure of the scale separation in the system. The temporal derivative then takes the form

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial t'} + \frac{1}{\varepsilon}\frac{\partial}{\partial \tau}. \tag{1.27}$$

We expand our solution according to the following asymptotic expansion

$$\mathbf{u} = \mathbf{u}^0(t', \tau) + \varepsilon \mathbf{u}^1(t', \tau) + \dots \tag{1.28}$$

By grouping like powers of ε, we find at leading order $(1/\varepsilon)$

$$\frac{\partial \mathbf{u}^0}{\partial \tau} + \mathcal{L}\mathbf{u}^0 = 0. \tag{1.29}$$

The leading order solution is then

$$\mathbf{u}^0(\mathbf{x}, t', \tau) = e^{-\tau\mathcal{L}}\mathbf{u}^0(\mathbf{x}, t', \tau). \tag{1.30}$$

Note here that $\mathbf{u}^0$ is a function of both the slow and fast timescales. The effect of the averaging process is to filter the fast waves and therefore remove the functional dependency on the fast timescale. In light of this fact and following Embid and Majda (1996) we may define the *averaged solution* as that solution which depends only on the slow timescale and from which the full solution may be regained by the application of the exponential operator. The leading order solution then takes the form

$$\mathbf{u}^0(\mathbf{x}, t', \tau) = e^{-\tau\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t'), \tag{1.31}$$

where $\overline{\mathbf{u}}$ denotes the averaged $\mathbf{u}$. Importantly, $\overline{\mathbf{u}}$ is a function of the *only* the spatial coordinate and the slow timescale, t'. The effects on the fast timescale are completely described by the exponential operator, $e^{\tau\mathcal{L}}$. As will be discussed in more detail in Chapter 2, the application of the matrix exponential is an analytical operation and therefore does not suffer from any timestep restrictions which would otherwise be imposed by the CFL condition.

Now consider the terms of $\mathcal{O}(\varepsilon)$, which results in the following differential equation

$$\frac{\partial \mathbf{u}^1}{\partial \tau} + \mathcal{L}\mathbf{u}^1 = -\left(\frac{\partial \mathbf{u}^0}{\partial t'} + \mathcal{N}(\mathbf{u}^0, \mathbf{u}^0)\right). \tag{1.32}$$

Applying an integrating factor method where $e^{\tau\mathcal{L}}$ is the integrating factor we obtain

$$\frac{\mathrm{d}}{\mathrm{d}\tau}(e^{\tau\mathcal{L}}\mathbf{u}^1) = -e^{\tau\mathcal{L}}\left(\frac{\partial \mathbf{u}^0}{\partial t'} + \mathcal{N}(\mathbf{u}^0, \mathbf{u}^0)\right), \tag{1.33}$$

which admits the following solution

$$e^{\tau\mathcal{L}}\mathbf{u}^1 = \mathbf{u}^1 - \tau\frac{\partial \overline{\mathbf{u}}(\mathbf{x}, t')}{\partial t'} - \int_0^\tau e^{s\mathcal{L}}\mathcal{N}(e^{-s\mathcal{L}}\overline{\mathbf{u}}, e^{-s\mathcal{L}}\overline{\mathbf{u}})\,\mathrm{d}s. \tag{1.34}$$

Note that we have used the definition of the leading order solution in terms of the averaged solution from equation (1.31). In order to prevent the emergence of secularity, the second term in the asymptotic expansion (equation (1.28)) must be weaker than the first. This is expressed by the *sublinear growth condition*:

$$|\mathbf{u}^1(\mathbf{x}, t', \tau)| = o(\tau) \text{ uniformly for } 0 \leq \tau \leq T/\varepsilon. \tag{1.35}$$

The operator $e^{\tau\mathcal{L}}$ is norm-preserving, and so $e^{\tau\mathcal{L}}u^1$ satisfies the sublinear growth condition (equation (1.35)) if and only if $\mathbf{u}^1$ does. From this fact and equation (1.34) comes the following condition, which the averaged equations must satisfy

$$\begin{gathered}\frac{\partial \overline{\mathbf{u}}(\mathbf{x}, t')}{\partial t'} + \lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau e^{s\mathcal{L}}\mathcal{N}(e^{-s\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t'), e^{-s\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t'))\,\mathrm{d}s = 0, \\ \overline{\mathbf{u}}(\mathbf{x}, t')\big|_{t'=0} = \mathbf{u}^0(\mathbf{x}, \mathbf{0}, \mathbf{0}).\end{gathered} \tag{1.36}$$

Recall that the application of the matrix exponential is an analytical operation. It therefore requires no timestepping. Rather, timestepping is performed over the averaged variable $\overline{\mathbf{u}} = \overline{\mathbf{u}}(\mathbf{x}, t')$ in equation (1.36), and the fast oscillations are reintroduced by the application of the matrix exponential. Comparing it to equation (1.4), it has lost the factor of $1/\varepsilon$ and so is less stiff[4]. This fact combined with the ease of applying the matrix exponential permits very large timesteps with this averaged system.

[4] Where we use 'less stiff' to mean 'exhibits a smaller variation in timescales'. The implication is that larger timesteps may be taken for less stiff problems.

Conceptually, the solution consists of both slow and fast components as shown in Figure 1.6. There is a slow trend which modulates the fast waves and which may be determined from the full solution by a properly-formulated moving average in time over the fast waves. The improved stability and accuracy when numerically solving equation (1.36) as compared to the full rapidly-rotating equations (1.26) is due to the timestepping being performed along this underlying slower solution.

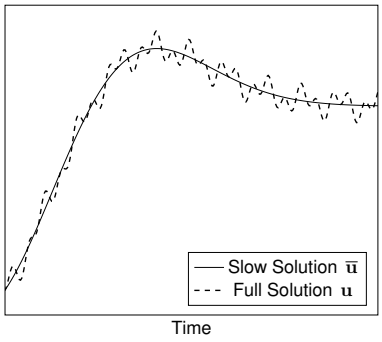

Figure 1.6: The slow trend which lacks rapid oscillations and the full solution, which follows it but features these fast waves is depicted conceptually. In practice, we timestep along the solid curve, as the oscillatory information necessary to regain the solution shown by the dotted line is contained entirely in the matrix exponential $e^{\tau\mathcal{L}/\varepsilon}$.

The wave averaged equation, (1.36), has been studied in detail since at least 1961 by Bogoliubov and Mitropolsky (1961) and more recently by Schochet (1994), who described its effects on the system in terms of *cancellation of oscillations*. The idea, which will be treated more rigorously in the coming chapters, is that the averaging integral filters out the fastest parts of the flow, while retaining the slowest, which are the most relevant for resolving the long-time dynamics. The novelty of this work lies primarily in the discovery and proof that the 'fastest parts of the flow' are not particular waves, but rather, when speaking numerically, *discrete components of nonlinear oscillation*.

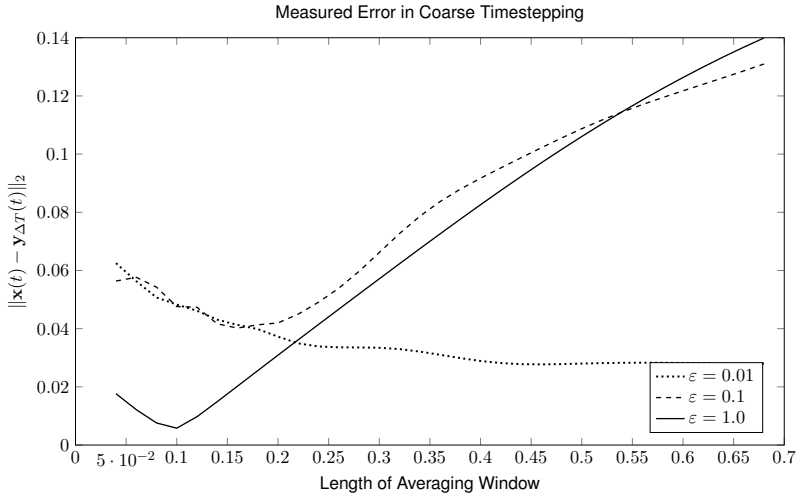

Figure 1.7: The error between a solution based on a fast-wave averaged solver ($\mathbf{y}_{\Delta T}(t)$) and one from a standard method integrating the unaveraged equations ($\mathbf{x}(t)$) is shown for three values of scale separation. A clear optimal exists as ε moves farther out of the small-ε limit. Here, the timestep with the averaged solver was 0.1, and for the full solver it was 0.002. A pseudospectral method was applied with 64 modes for the 1D RSWE.

,

We shall discuss the numerical particularities of using this averaged equation and its projection back into real-space by the matrix exponential in Chapter 4. To provide some idea of the direction in which this thesis is going, we must confront the problem that for practical geophysical flows, the scale separation is finite (i.e. ε in (1.36) is a finite quantity). This means that when modelling realistic flows, the integral in (1.36) becomes finite. In particular, we are focussed on explaining the behaviour seen in Figure 1.7, which measures the error when using a solver based on fast-wave averaging as compared to a reference 'full' solution. The behaviour observed for finite ε contradicts the predictions of the asymptotic theory as $\varepsilon \to 0$, which predicts increasing accuracy with an increasing averaging integral length. In the finite case, there is clearly

an optimal choice which varies with ε. How does one explain this? Perhaps more importantly, how does one use it to develop useful algorithms?

Before numerical analysis of this averaging is possible we will need to discuss triad interactions, which arise naturally from the quadratic nonlinearity and the dispersion relation. Triad interactions, and in particular direct and near resonances, provide a natural way to understand the effect of averaging on our system. Through making the nonlinear interaction explicit, they are those discrete components of nonlinear oscillation discussed earlier. From a wave averaging point of view, they provide a more natural atomic unit of solution than a single Fourier mode.

Key Points

- *Stiffness* is a property of differential equations which makes their numerical solution difficult.
- In the limit of infinite scale separation, a wave-averaged solution to the rotating shallow water equations exists with an infinitely long averaging window.
- For finite scale separation, there is an optimal choice for the finite averaging window length.

2 *Exponential Integration*

> I lost my job as a cricket commentator for saying 'I don't want to bore you with the details'.
>
> Milton Jones

Exponential integrators are a class of methods for the numerical solution of differential equations which are otherwise difficult to handle numerically due to stiffness (Section 1.1). Stiff systems of ODEs arise organically when solving PDEs with spatially-periodic boundary conditions via representation as Fourier series. The associated spectral and pseudo-spectral methods, which have been quite successful in many applications (*cf.* Canuto et al. (1988), Boyd (2000)) lend themselves naturally to exponential integrators. We note as well that analysing our system in terms of the matrix exponential and the associated eigendecomposition exposes the nature of the wave resonances, as we shall see in Chapter 3.

Later in this chapter we shall discuss the specific case of developing an exponential integrator for the rotating shallow water equations, as exponential integrators form the basis (in subtly different fashions) of both the coarse and fine timesteps in the APinT method introduced in Chapter 5. It should be noted that the linear operator in that case is skew-Hermitian (Definition 2.6) and so the exponential integrator, $e^{t\mathcal{L}}$, has several favourable properties such as uniform boundedness independent of timestep, unlike the propagator of the explicit Euler method. In the case of oscillatory problems, this exponential also contains all of the information about the linear oscillations, unlike the propagator of the implicit Euler method (Hochbruck and Ostermann, 2010).

We shall consider initial value problems of the following form in this work:

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= \mathcal{L}u(x,t) + \mathcal{N}(u,t), \\ u(x,0) &= u_0, \end{aligned} \tag{2.1}$$

where $\mathcal{L}$ is a linear operator with purely imaginary eigenvalues of large modulus and $\mathcal{N}$ is a nonlinear operator which is not necessarily of quadratic type. Note that we have modified our notation somewhat to reflect this broader set of governing equations. As we proceed further into this thesis, we will place some further restrictions on this equation so as to restrict ourselves to the equations of fluid mechanics, but in the interest of providing a general overview of exponential integration methods we will prefer this more general form for the time being. We shall assume that the nonlinear operator is non-stiff in that it may be reliably approximated by an appropriate explicit method. The stiffness in this problem then arises from the linear operator.

We shall define some numerical timestepping method by $\{S_{\Delta t}\}$ for some timestep, Δt, where the timestep is a suitably small portion of time, t. We then state the following three definitions:

Definition 2.1 (Order of Accuracy)**.** $\{S_{\Delta t}\}$ has **order of accuracy** p if

$$\|u(t+\Delta t) - S_{\Delta t}u(t)\| = \mathcal{O}(\Delta t^{p+1}) \text{ as } \Delta t \to 0, \tag{2.2}$$

for any $t \in [0, T]$, with $u(t)$ sufficiently smooth. The method is **consistent** for $p > 0$. ▲

According to Batkai et al. (2009), consistency roughly means that the approximating difference equations converge in the same sense to the original abstract initial value problem.

Definition 2.2 (Convergence)**.** $\{S_k\}$ is **convergent** if:

$$\lim_{\Delta t \to 0} \|S^n_{\Delta t}u(0) - u(n\Delta t)\| = 0, \tag{2.3}$$

for any $t \in [0, T]$. Here $S^n_{\Delta t}u(0)$ denotes the numerical solution at $t = n\Delta t$ with initial condition $u(0)$. ▲

Definition 2.3 (Stability)**.** $\{S_{\Delta t}\}$ is **stable** if there exists some $C > 0$ such that:

$$\|S^n_{\Delta t}u(0)\| \leq C, \tag{2.4}$$

$\forall\, n$ and Δt such that $0 \leq n\Delta t \leq T$. Here, the norm is taken over the solution after n timesteps. ▲

2.1 *Formulation of the Exponential Integrator*

RECALL THAT IN THE ABSTRACT EQUATION (2.1), the stiffness was due to the linear term, $\mathcal{L}$. In practice, this means that a very fine

timestep will have to be applied to the linear operator in order to achieve stability in the case of an explicit method or accuracy in the case of an implicit method. However, the non-linear term, $\mathcal{N}$, does not impose the same restriction. We shall further assume that the nonlinear term is significantly more expensive to compute than the linear one as is the case for pseudospectral methods. In the interest of computational expense, we aim to limit the number of expensive computations required via an increase in the step size.

The first step towards exponential integration methods as we will use them comes from the fact that an analytical solution exists to the linear form of equation (2.1), where the nonlinear term has been neglected, i.e.:

$$u(t) = e^{-t\mathcal{L}}u_0. \tag{2.5}$$

The aim of exponential integrator methods is then to solve the linear term *exactly* and to employ some numerical method for the nonlinear term. This permits larger timesteps to be taken for a given level of accuracy, as the analytical solution for the linear term is trivially stable and convergent, and the nonlinear term is not the source of stiffness. An exponential integrator has the following properties (Berland, 2005):

1. If $\mathcal{L} = 0$ the scheme reduces to a standard general linear method, which we shall term *the underlying scheme*.

2. If $\mathcal{N}(u,t) = 0 \; \forall \; u$ and t, the scheme reduces to the exact solution of the linear equation.

Consider the full equation (2.1), from which the exponential integrator form of the solution is derived. We begin by multiplying both sides by the *integrating factor*, $e^{-t\mathcal{L}}$, a process which is familiar from the solution of ordinary differential equations (Stewart, 2007)

$$e^{-t\mathcal{L}}\left(\frac{\partial u}{\partial t} - \mathcal{L}u\right) = e^{-t\mathcal{L}}\mathcal{N}(u,t). \tag{2.6}$$

Then, group the linear terms and combine them using the chain rule, i.e.:

$$\frac{\partial}{\partial t}(e^{-t\mathcal{L}}u) = e^{-t\mathcal{L}}\mathcal{N}(u,t). \tag{2.7}$$

We are now presented with the option of solving equation (2.7) as it is currently formulated through some numerical method (termed the *Integrating Factor Method*) or proceeding as below to obtain an *Exponential Time Differencing Method*.

Integrating both sides from 0 to t yields:

$$u(x,t) = e^{t\mathcal{L}}u_0 + e^{t\mathcal{L}}\int_0^t e^{-s\mathcal{L}}\mathcal{N}(u,s)\,\mathrm{d}s, \tag{2.8}$$

which, as the integral is taken over the dummy variable in time s

and not t, may be simplified to

$$u(x,t) = e^{t\mathcal{L}}u_0 + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{N}(u,s)\,\mathrm{d}s, \tag{2.9}$$

which may be recognisable as the variation of constants formula for an ODE. The integration over a dummy variable and re-projection with the full time arose in the derivation of the averaged equations in Chapter 1. This is an important idea which will arise again in the efficient numerical computation of the average in Chapter 4.

2.2 *Solution Methods*

We are then interested in some numerical scheme which approximates the integral and permits the solution to be timestepped forwards. There exist a myriad of numerical schemes which are beyond the scope of this work. Exponential Rosenbrock methods are among them, but the interested reader is referred to Pope (1963), Tokman (2006), and Hochbruck and Ostermann (2006).

The simplest possible numerical method for (2.9) involves interpolating the nonlinearity at the value $\mathcal{N}(u_0,t_0)$ only, leading to the so-called *exponential Euler* approximation

$$u_{n+1} = e^{\Delta t\mathcal{L}}u_n + \Delta t\phi_1(-\Delta t\mathcal{L})\mathcal{N}(u_n,t_n), \tag{2.10}$$

where

$$\phi_1(z) = \frac{e^z - 1}{z}. \tag{2.11}$$

The *integrating factor method* is obtained by solving equation (2.7) using some appropriate time-stepping scheme. For example, Cox and Matthews (2002) give a second-order Adams-Bashforth scheme. It is notable, however, that integrating factor methods have different fixed points than the original ODE. They also tend to have large error constants[1] and so *Exponential Time Differencing* (ETD) methods are generally preferred.

[1] With respect to Definition 2.1, this means that the error scales like: $\|u(t+\Delta t) - S_{\Delta t}u(t)\| \leq C\Delta t^{p+1}$, where C is a large constant. The performance is good asymptotically, but not for finite timesteps.

2.2.1 *Exponential Time Differencing Methods*

Exponential Time Differencing methods arise from equation (2.9) where the linear term is computed exactly and the integral is approximated in some fashion, with different ETD methods being distinguished by the method used to approximate this integral. The simplest approximation is that $\mathcal{N}(u,t)$ is constant on the interval $[t_n,t_{n+1}]$, which leads to the ETD1 scheme

$$u_{n+1} = e^{\Delta t\mathcal{L}}u_n + (e^{\Delta t\mathcal{L}} - I)\mathcal{L}^{-1}\mathcal{N}(u_n,t_n). \tag{2.12}$$

where I is the identity. Similar ETD methods using higher order polynomials may also be derived. For a truncation error of $\mathcal{O}(\Delta t^{s+1})$ it is necessary to use a polynomial of degree $s-1$. The general form of such a scheme (Cox and Matthews, 2002) is

$$u_{n+1} = e^{\Delta t \mathcal{L}} u_n + \Delta t \sum_{m=0}^{s-1} \sum_{k=0}^{m} (-1)^k g_m \binom{m}{k} \mathcal{N}(u_{n-k}, t_{n-k}), \qquad (2.13)$$

where

$$g_m = (-1)^m \int_0^1 e^{k(1-\lambda)\mathcal{L}} \binom{-\lambda}{m} \mathrm{d}\lambda \qquad (2.14)$$

where the binomial coefficient is defined in the usual way and $\lambda = \tau/\Delta t$ for $0 < \tau < \Delta t$. In practice, g_m may be computed in a reasonable fashion through the use of an appropriate generating function. It should be noted here that the ETD methods discussed above are of multistep type, which require s previous evaluations of the nonlinear term. This is a problem particularly in initialising the computation, as only the initial condition is available.

For this reason, as well as larger stability regions and smaller error constants, Runge-Kutta (RK) methods are often applied. The construction of an RK scheme of arbitrary order is straightforward, and may be found in Cox and Matthews (2002) or Iserles (2008), for example. In particular, the 4-th order ETDRK4 scheme has been applied to the Kuramoto-Sivashinsky equation in the context of sequential data assimilation by Jardak et al. (2010) and to the rotating shallow water equations in the context of parallel-in-time integration by Haut and Wingate (2014).

It should also be noted that several modified ETDRK schemes have been proposed, for example in Krogstad (2005), and Kassam and Trefethen (2005). A detailed analysis of convergence and stability of ETD schemes is provided by Hochbruck and Ostermann (2010).

2.3 *Strang Splitting*

We have also employed splitting methods within the context of the exponential integrator. A particularly readable and far-reaching review of splitting methods is given by McLachlan and Quispel (2002). They give the following three steps for a splitting method for some vector field, X:

1. Choose a set of vector fields, X_i such that $X = \sum X_i$;
2. Integrate each X_i;
3. Combine these solutions to yield an integrator for X,

with the caveat that each sub-field, X_i, should be in some sense simpler than X. In this way, they can be thought of as a class of *divide and conquer* algorithms.

The right-hand side of equation (2.1) is written as the sum of two terms. Splitting proceeds by assuming that each of these is individually integrable, i.e.

$$\frac{\partial u}{\partial t} = \mathcal{L}u, \tag{2.15}$$

and

$$\frac{\partial u}{\partial t} = \mathcal{N}(u,t). \tag{2.16}$$

At any point in phase space, we may break up the vector field (which is tangent to the solution vector) into the two components above. We first step along one, then the other curve.

The simplest possible splitting method involves taking the full timestep along one, and then the full timestep along the second curve, i.e. we first solve equation (2.15) subject to the initial condition $u(0) = u_n$ over a time Δt to find $u^* = u_1(x, \Delta t)$, i.e. the solution of the differential equation on the first vector field. Next, we solve equation (2.16) subject to the initial condition $u(0) = u^*$ for a time Δt. This yields our solution based on a first-order splitting.

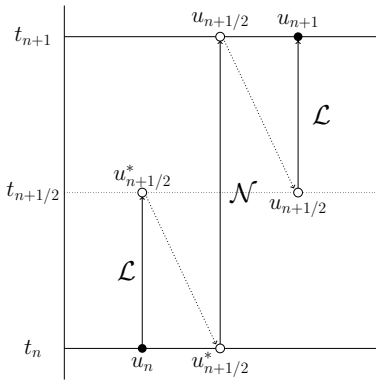

Figure 2.1: Strang Splitting Schematic

Splitting allows us to take advantage of the fact that we have an analytical solution for one of our terms in equation (2.1) via the matrix exponential as long as this term is considered separately. A simple but powerful improvement on the first-order splitting method discussed above is available, called *Strang splitting* (Strang, 1968). Strang splitting is a second-order method, and relies on the use of half-timesteps. The Strang splitting algorithm is to solve a half-step of the linear term, followed by a full-step of the nonlinear term, and then another half-step of the linear term. That is, if we denote the solution operator to the nonlinear term over a time interval Δt as $\mathcal{S}_{\mathcal{N}}(\Delta t)$, we write the three-step Strang splitting as (Chertock and Kurganov, 2009)

$$u(x, t + \Delta t) = e^{\Delta t \mathcal{L}/2} \mathcal{S}_{\mathcal{N}}(\Delta t) e^{\Delta t \mathcal{L}/2} u(x,t), \tag{2.17}$$

where we solve the nonlinear term in practice by midpoint quadrature. This is shown in Figure 2.1 and described in detail in Algorithm 2.1.

$v \leftarrow e^{(\Delta t/2)\mathcal{L}} u_0$ ▷ Half-step with exponential operator
$v \leftarrow \mathcal{S}^{\Delta t}_{\mathcal{N}}(v)$ ▷ Nonlinear step by midpoint quadrature
$v \leftarrow \mathcal{S}^{\Delta t}_{\mathcal{N}}\left(u_0 + \frac{\Delta t}{2} v\right)$
$u_1 \leftarrow e^{(\Delta t/2)\mathcal{L}} v$ ▷ Final half-step with exponential operator
return u_1

Algorithm 2.1: Strang Splitting Algorithm

2.4 Matrix Exponential Formulation

THE METHODS DISCUSSED ABOVE are numerically straightforward if $\mathcal{L}$ is a scalar. In that case, $e^{t\mathcal{L}}$ would be the scalar exponential. However, we are concerned here with the more general case which arises for partial differential equations with vector unknowns, as is the case for the Rotating Shallow Water Equations as well as other equations of fluid mechanics and mathematical physics (e.g. the Navier-Stokes equations, the Boussinesq equations, etc.) In this case we must compute the matrix exponential.

We introduce the matrix exponential by analogy with the scalar exponential. While the scalar exponential takes the Maclaurin series expansion

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, \tag{2.18}$$

the matrix exponential may be expanded in series as a matrix analogue of the scalar case:

$$e^{\mathbf{x}} = \mathbf{I} + \mathbf{x} + \frac{\mathbf{x}^2}{2!} + \frac{\mathbf{x}^3}{3!} + \dots \tag{2.19}$$

There exist many different ways to compute the matrix exponential, whether analytically or through numerical approximation. Moler and van Loan (2003) give a review of 19 of these methods and note some of the particular computational difficulties. We shall consider in this work a linear operator whose representation in Fourier space is endowed with certain properties which enable us to compute its matrix exponential analytically.

The method outlined in this section only holds under the restrictions placed on the computational domain in Chapter 1, i.e. constant mean water depth and a spatially-periodic domain. These restrictions have been made so as to bring the current work in line with existing results in the literature in which the matrix exponential is used (*cf.* Embid and Majda (1996), Majda (2002), and Haut and Wingate (2014)). A more generally-applicable method of computing $e^{t\mathcal{L}}$ for similar, i.e. oscillatory-stiff, problems is presented by Haut et al. (2015) and Schreiber et al. (2017).

2.5 Some Operator Preliminaries

AS WE ARE INTERESTED in both solving our systems via a Fourier spectral method and describing these solutions in Fourier space, we will work extensively with complex vectors and operators. It is useful then to first define our inner product on a complex vector

space. Note that for complex numbers with zero imaginary part, this reduces to the familiar inner product on a real vector space.

Definition 2.4 (Complex Inner Product)**.** The inner product on a complex vector space is defined

$$\langle \mathbf{u}, \mathbf{v} \rangle = u_i \overline{v}_i. \tag{2.20}$$

where the over-bar denotes the complex conjugate and Einstein's summation convention is applied. This product is bilinear and antilinear in the second slot. It has the following properties:

$$\langle \mathbf{u} + \mathbf{w}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{w}, \mathbf{v} \rangle \tag{2.21a}$$
$$\langle \mathbf{u}, \mathbf{v} + \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle \tag{2.21b}$$
$$\langle \alpha \mathbf{u}, \mathbf{v} \rangle = \alpha \langle \mathbf{u}, \mathbf{v} \rangle \tag{2.21c}$$
$$\langle \mathbf{u}, \alpha \mathbf{v} \rangle = \overline{\alpha} \langle \mathbf{u}, \mathbf{v} \rangle \tag{2.21d}$$
$$\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle} \tag{2.21e}$$
$$\langle \mathbf{u}, \mathbf{u} \rangle \geq 0; \text{ with equality iff } \mathbf{u} = \mathbf{0}. \tag{2.21f}$$

▲

We have already restricted our linear operator, $\mathcal{L}$, to having a full set of purely imaginary eigenvalues, which implies that it is skew-Hermitian. In this section we shall give some useful results for the computation of the matrix exponential with this type of operator, following largely from Horn and Johnson (1986). Firstly, we give the formal definition of this class of operators.

Definition 2.5 (Linear Operator)**.** Let V be a finite-dimensional Hilbert space over $\mathbb{C}$ with an inner product denoted $\langle \cdot, \cdot \rangle$. A linear operator $T \in \mathcal{L}(V)$ is uniquely determined by the values of

$$\langle Tv, w \rangle \ \forall v, w \in V. \tag{2.22}$$

▲

We also require that our linear operator (and by extension the complex-valued matrix which represents it in the Fourier domain) is Skew-Hermitian.

Definition 2.6 (Skew-Hermitian)**.** Given $T \in \mathcal{L}(V)$, the *skew-adjoint* of T is the operator T^* such that:

$$\langle Tv, w \rangle = -\langle v, T^* w \rangle \ \forall v, w \in V. \tag{2.23}$$

T is called a skew-adjoint operator if $T = -T^*$. ▲

One system of recurring interest throughout this work is the one-dimensional Rotating Shallow Water Equations (1D RSWE). The

matrix representation of the linear operator associated with the 1D RSWE is, in Fourier space:

$$\mathcal{L} = \begin{bmatrix} 0 & -1 & F^{-1/2}ik \\ 1 & 0 & 0 \\ F^{-1/2}ik & 0 & 0 \end{bmatrix}. \tag{2.24}$$

Taking the adjoint to denote complex conjugation, we find that $\mathcal{L}$ is indeed a skew-Hermitian operator, as $M(\mathcal{L}) = -M(\mathcal{L})^*$, where $M(\mathcal{L})$ denotes the matrix representation of the linear operator:

$$\begin{bmatrix} 0 & -1 & F^{-1/2}ik \\ 1 & 0 & 0 \\ F^{-1/2}ik & 0 & 0 \end{bmatrix} = - \begin{bmatrix} 0 & 1 & -F^{-1/2}ik \\ -1 & 0 & 0 \\ -F^{-1/2}ik & 0 & 0 \end{bmatrix}. \tag{2.25}$$

All skew-Hermitian operators are also *normal* operators:

Definition 2.7 (Normal Operators)**.** We call $T \in \mathcal{L}(V)$ a *normal* operator iff it commutes with its adjoint, i.e.:

$$TT^* = T^*T. \tag{2.26}$$

This implies that T is normal iff:

$$\|Tv\| = \|T^*v\| \; \forall v \in V. \tag{2.27}$$

▲

In order to construct the matrix exponential, we require a set of orthogonal eigenvectors. The *spectral decomposition theorem* ensures that the eigenvalues of our linear operator, i.e. the eigenbasis of the problem, are orthogonal.

Theorem 2.1 (Spectral Theorem)**.** *Let V be a finite-dimensional Hilbert space over $\mathbb{C}$ and $T \in \mathcal{L}(V)$. T is normal if and only if there exists an orthonormal basis for V consisting of eigenvectors of T.*

♦

Proof.
$\implies$ Let T be a normal matrix. For any operator T on a complex Hilbert space V of dimension n, there exists an orthonormal basis $e = [e_1, e_2, \ldots, e_n]^T$ for which the matrix representation of T, $M(T)$, is upper-triangular

$$M(T) = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn} \end{bmatrix}. \tag{2.28}$$

We may then say that $M(T) = a_{ij}$ with $a_{ij} = 0$ for $i > j$. This means that $Te_1 = a_{11}e_1$ and $T^*e_1 = \sum_{k=1}^{n} \overline{a}_{1k}e_k$. By the Pythagorean Theorem and Definition 2.7

$$|a_{11}|^2 = \|a_{11}e_1\|^2 = \|Te_1\|^2 = \|T^*e_1\|^2 = \left\| \sum_{k=1}^{n} \overline{a}_{1k}e_k \right\|^2 = \sum_{k=1}^{n} |a_{1k}|^2. \tag{2.29}$$

Thus, $|a_{12}| = \cdots = |a_{1n}| = 0$. This argument can be repeated to eliminate all non-diagonal entries in $M(T)$. Thus, T is diagonal with respect to the basis e and so $e_1 \dots e_n$ are eigenvectors of T.

$\Longleftarrow$ If there exists some orthonormal basis of V consisting of eigenvectors of T, then $M(T)$ with respect to this basis must be diagonal. Also, $M(T^*) = M(T)^*$ must be diagonal with respect to this basis as well. Any two diagonal matrices commute, so it follows that

$$M(TT^*) = M(T)M(T^*) = M(T^*)M(T) = M(T^*T). \tag{2.30}$$

□

Theorem 2.1 means that a matrix containing the eigenvectors of $\mathcal{L}$ as columns will be a *unitary* matrix by Theorem 2.2 below. Unitary matrices have useful properties with respect to their inverse which will simplify the implementation of the matrix exponential.

Definition 2.8 (Unitary Matrix)**.** A unitary matrix is a complex square matrix whose conjugate transpose is also its inverse, i.e.

$$U^*U = UU^* = I. \tag{2.31}$$

▲

Theorem 2.2 (Orthogonality of Columns)**.** *The columns of a unitary matrix are mutually orthogonal.*

♦

Proof. We may write the requirement of Definition 2.8 in index notation

$$\overline{u}_{ki}u_{kj} = \delta_{ij}, \tag{2.32}$$

where δ_{ij} is the Kronecker delta. Thus, the complex inner product of a column with itself must equal one, and with any other column must equal zero. By the properties of the inner product, the columns must be orthogonal.

□

The linear operator associated with the 1D RSWE then yields purely imaginary eigenvalues and an orthogonal basis of eigenvectors which may be written as a unitary matrix. This is sufficient to compute the matrix exponential in a computationally efficient fashion.

2.5.1 *Construction of the Matrix Exponential*

We consider some matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. If $\mathbf{A}$ has a complete set of linearly independent eigenvectors such that

$$\mathbf{A}\mathbf{v}_k = \lambda \mathbf{v}_k \text{ for } k = 1, \dots, n, \tag{2.33}$$

then there exists a matrix $\mathbf{T}$, such that the columns of $\mathbf{T}$ are the eigenvectors of $\mathbf{A}$. Then

$$\mathbf{A}\mathbf{T} = [\mathbf{A}\mathbf{v}_1, \mathbf{A}\mathbf{v}_2, \dots, \mathbf{A}\mathbf{v}_n] = [\lambda_1\mathbf{v}_1, \lambda_2\mathbf{v}_2, \dots, \lambda_n\mathbf{v}_n] = \mathbf{T}\mathbf{\Lambda}. \tag{2.34}$$

We may then simplify this to the form

$$\mathbf{A}\mathbf{T}\mathbf{T}^{-1} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}, \tag{2.35}$$

$$\implies \mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}. \tag{2.36}$$

Consider now the case of the exponential, keeping in mind that $\mathbf{A}$ may be taken to be an operator and thus the exponential may as well.

$$e^{\mathbf{A}} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k = \sum_{k=0}^{\infty} \frac{1}{k!}\left(\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\right)^k. \tag{2.37}$$

As $\mathbf{T}\mathbf{T}^{-1}$ is the identity, this generalises for any arbitrary number of multiplications to yield:

$$e^{\mathbf{A}} = \mathbf{T}\left(\sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{\Lambda}^k\right)\mathbf{T}^{-1}, \tag{2.38}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues and so the term in brackets, recognisable as the Maclaurin series, reduces to yield:

$$e^{\mathbf{A}} = \mathbf{T}\begin{bmatrix} e^{\lambda_1} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n} \end{bmatrix}\mathbf{T}^{-1}. \tag{2.39}$$

The matrix of eigenvectors, $\mathbf{T}$, may be precomputed, either numerically or analytically (*q.v.* Section 2.5.2). Because of its unitary property (Definition 2.8), there is no requirement to numerically invert and store this matrix: it is sufficient only to know $\mathbf{T}$, as its inverse is readily available as its conjugate transpose.

2.5.2 *Eigenbasis of the RSWE*

NOW CONSIDER THE LINEAR PROBLEM ALONE, i.e. neglecting the nonlinear term

$$\frac{\partial \mathbf{u}}{\partial t} + \mathcal{L}\mathbf{u} = 0. \tag{2.40}$$

Assume that $\mathbf{u}$ is spatially-periodic and consider its Fourier representation

$$\mathbf{u}(\mathbf{x}, t) = \sum_{\mathbf{k}\in\mathbb{C}} \hat{\mathbf{u}}_{\mathbf{k}}(t) e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)}. \tag{2.41}$$

Substituting equation (2.41) into (2.40) gives

$$\begin{pmatrix} -i\omega\hat{v}_1 \\ -i\omega\hat{v}_2 \\ -i\omega\hat{h} \end{pmatrix} + \begin{pmatrix} 0 & -1 & F^{-1/2}ik \\ 1 & 0 & 0 \\ F^{-1/2}ik & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{h} \end{pmatrix} = 0, \tag{2.42}$$

which gives the eigenvalue problem

$$\begin{pmatrix} -i\omega & -1 & F^{-1/2}ik \\ 1 & -i\omega & 0 \\ F^{-1/2}ik & 0 & -i\omega \end{pmatrix} = 0. \tag{2.43}$$

Solving this problem yields:

$$\omega = 0, \qquad \omega^2 = F^{-1}k^2 + 1. \tag{2.44}$$

In a physical context, these eigenvalues are the dispersion relation and represent one slow mode and two branches of dispersive waves moving in opposite directions. For convenience, we introduce the index α, such that:

$$\omega_k^\alpha = \alpha\sqrt{1 + F^{-1}k^2}, \tag{2.45}$$

where $\alpha = -1, 0, 1$. We then find the eigenfunctions corresponding to each of these eigenvalues for $|k| \neq 0$ to be

$$\mathbf{r}_{\mathbf{k}}^{-1} = \begin{pmatrix} \frac{i\omega}{k} \\ \frac{-iF^{1/2}}{k} \\ 1 \end{pmatrix} \hat{h}; \; \mathbf{r}_{\mathbf{k}}^{0} = \begin{pmatrix} 0 \\ F^{-1/2}ik \\ 1 \end{pmatrix} \hat{h}; \; \mathbf{r}_{\mathbf{k}}^{1} = \begin{pmatrix} \frac{-i\omega}{k} \\ \frac{-iF^{1/2}}{k} \\ 1 \end{pmatrix} \hat{h}. \tag{2.46}$$

2.5.3 *Orthonormalisation*

THESE EIGENFUNCTIONS ARE ORTHONORMAL to one another by the spectral decomposition theorem (Theorem 2.1). While the eigenfunctions are guaranteed to be orthogonal, they are not guaranteed to be orthonormal, i.e. they may have lengths which are not equal to one. The procedure for orthonormalisation is to divide each vector by its own length, where we define the norm of a vector in the

usual way using the inner product

$$\mathbf{r}^{\alpha}_{k,\,\text{normed}} = \frac{\mathbf{r}^{\alpha}_{k}}{\|\mathbf{r}^{\alpha}_{k}\|} = \frac{\mathbf{r}^{\alpha}_{k}}{\sqrt{\mathbf{r}^{\alpha}_{k}\cdot\bar{\mathbf{r}}^{\alpha}_{k}}}. \tag{2.47}$$

We then compute and report the norms of the various eigenvectors here. The orthonormalised eigenfunctions are

$$\mathbf{r}^{-1}_{\mathbf{k}} = \begin{pmatrix} \frac{i\omega}{k\sqrt{2+2F/k^2}} \\ \frac{-iF^{1/2}}{k\sqrt{2+2F/k^2}} \\ \frac{1}{\sqrt{2+2F/k^2}} \end{pmatrix}\hat{h};\ \mathbf{r}^{0}_{\mathbf{k}} = \begin{pmatrix} 0 \\ \frac{ik}{F\omega} \\ \frac{1}{\omega} \end{pmatrix}\hat{h};\ \mathbf{r}^{1}_{\mathbf{k}} = \begin{pmatrix} \frac{-i\omega}{k\sqrt{2+2F/k^2}} \\ \frac{-iF^{1/2}}{k\sqrt{2+2F/k^2}} \\ 1 \end{pmatrix}\hat{h}. \tag{2.48}$$

For the special case when $|k| = 0$, the orthonormalised eigenfunctions are

$$\mathbf{r}^{-1}_{\mathbf{k}} = \begin{pmatrix} \frac{-i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}\hat{h};\quad \mathbf{r}^{0}_{\mathbf{k}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\hat{h};\quad \mathbf{r}^{1}_{\mathbf{k}} = \begin{pmatrix} \frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}\hat{h}. \tag{2.49}$$

2.6 *Symmetrisation of the Full RSWE*

THE 1D RSWE ARE A SIMPLIFICATION of the more general Rotating Shallow Water Equations. Practical considerations require that this system be solvable in its dimensional form as well. The issue arises immediately that the linear operator for this system is not skew-Hermitian, and so does not permit work with the framework described above. This is may be remedied easily upon rescaling. Recall the dimensional RSWE (equations (1.6) and (1.7)) given in Chapter 1:

$$\frac{D\mathbf{u}}{Dt} + \mathbf{f}\times\mathbf{u} = -g\nabla\eta, \tag{2.50}$$

$$\frac{\partial h}{\partial t} + \nabla\cdot(\mathbf{u}h) = 0, \tag{2.51}$$

where the unknown vector, $\mathbf{u} = [u, v, h]^T$; and where $u(\mathbf{x}, t)$ and $v(\mathbf{x}, t)$ are the velocities in the x- and y-directions respectively and $h(\mathbf{x}, t)$ is the perturbation height of the fluid. In a similar fashion to their nondimensionalisation, we begin by rewriting the above equations, written in terms of a total water depth, h, in terms of a perturbation about some mean water depth, denoted η and H_0 respectively, such that

$$h = H_0 + \eta. \tag{2.52}$$

Upon substitution of equation (2.52) into equations (1.6) and (1.7), the following equations are obtained, keeping in mind that H_0 is a constant,

$$\frac{D\mathbf{u}}{Dt} + \mathbf{f}\times\mathbf{u} + g\nabla\eta = 0, \tag{2.53}$$

$$\frac{\partial \eta}{\partial t} + H_0 \nabla \cdot \eta + \nabla \cdot (\mathbf{u}\eta) = 0. \tag{2.54}$$

In order to achieve a skew-Hermitian operator, we rewrite the equations in terms of the *geopotential height*, ϕ, defined as:

$$\phi \equiv g\eta, \tag{2.55}$$

which, along with multiplying the mass equation by g everywhere, yields

$$\frac{D\mathbf{u}}{Dt} + \mathbf{f} \times \mathbf{u} + \nabla \phi = 0, \tag{2.56}$$

and

$$\frac{\partial \phi}{\partial t} + \Phi_0 \nabla \cdot \phi + \nabla \cdot (\mathbf{u}\phi) = 0, \tag{2.57}$$

where $\Phi_0 = gH_0$. Multiplying equation (2.57) by a factor of $\Phi_0^{-1/2}$ gives

$$\frac{1}{\sqrt{\Phi_0}} \frac{\partial \phi}{\partial t} + \sqrt{\Phi_0} \nabla \cdot \phi + \frac{1}{\sqrt{\Phi_0}} \nabla \cdot (\mathbf{u}\phi) = 0. \tag{2.58}$$

Finally, we rewrite both equations in terms of a modified vector of unknowns involving a scaling on the geopotential height

$$\left[u, v, \phi'\right]^T = \left[u, v, \frac{1}{\sqrt{\Phi_0}}\phi\right]^T, \tag{2.59}$$

which yields

$$\frac{D\mathbf{u}}{Dt} + \mathbf{f} \times \mathbf{u} + \sqrt{\Phi_0} \nabla \phi' = 0, \tag{2.60}$$

$$\frac{\partial \phi'}{\partial t} + \sqrt{\Phi_0} \nabla \cdot \phi' + \nabla \cdot (\mathbf{u}\phi') = 0. \tag{2.61}$$

In these modified variables, the linear and nonlinear operators take the following form

$$\mathcal{L}\mathbf{u} = \begin{bmatrix} 0 & -f_0 & \sqrt{\Phi_0}\partial_x \\ f_0 & 0 & \sqrt{\Phi_0}\partial_y \\ \sqrt{\Phi_0}\partial_x & \sqrt{\Phi_0}\partial_y & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ \phi' \end{bmatrix}, \tag{2.62}$$

$$\mathcal{N}(\mathbf{u}, \mathbf{u}) = \begin{bmatrix} \mathbf{u} \cdot \nabla \mathbf{u} \\ \nabla \cdot (\phi' \mathbf{u}) \end{bmatrix}. \tag{2.63}$$

Subject to this linear rescaling and when working in Fourier space, the linear operator is skew-Hermitian and so the theory described in Section 2.4 is directly applicable.

2.7 *Numerical Results*

IN ORDER TO QUANTIFY THE ERROR of two of the main methods we have discussed here, the ETDRK4 and Strang Splitting methods, we have performed some numerical studies. We compare these to a common non-exponential method, the 3rd-order Adams-Bashforth

(AB3) method. For all three integrators, we have used a Fourier spectral method with 64^2 grid points and 2/3 dealiasing, and are solving the 2-D non-dimensional RSWE with an initial Gaussian height field and a doubly-periodic spatial domain (*cf.* appendix A). As no analytical solution is available, we compare the solution to a numerical one found by a 4th-order Runge Kutta method with 256^2 grid points and a timestep of 10^{-6}.

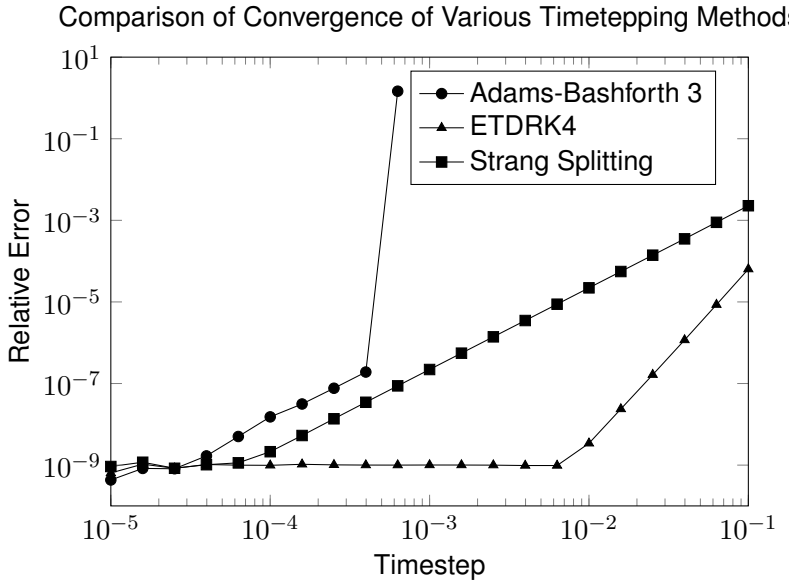

Figure 2.2: Exponential Integrator Convergence Studies, $\epsilon = 1$. The error relative to a reference solution is shown as a function of timestep. Note the improved accuracy of the two exponential integrator methods compared to the explicit 3rd-order Adams-Bashforth method, as well as stability for larger timesteps.

Figure 2.2 shows the 'non-stiff' case, where $\varepsilon = 1.0$. We see here that the exponential methods do not suffer from the same timestep limits as the AB3 method does, and were able to stably integrate with timesteps of up to 0.1. This is because the oscillations, which are primarily contained in the linear operator, are handled analytically through the matrix exponential. The AB3 method, on the other hand, was timestep-limited by the CFL limit (Trefethen, 1996) arising from the need to properly resolve all oscillations present in the solution. For larger timesteps than the largest shown in Figure 2.2 the AB3 method was numerically unstable.

The ETDRK4 method converged to single precision more quickly than the other two, achieving this degree of accuracy with a timestep of up to slightly less than 0.01. However, it does so at noticeably higher computational cost than the Strang splitting method.

In the more classically oscillatory-stiff case where $\varepsilon = 0.01$, shown in Figure 2.3, we see that the timestep limit imposed by the CFL condition on the AB3 method is even smaller than in the $\varepsilon = 1.0$ case, as would be expected, with the maximum timestep being roughly an order of magnitude smaller. The exponential integrators both continued to permit a timestep up to the order of $\Delta t = 0.1$. In this case, convergence was not achieved as quickly for the ETDRK4 method, which performed similarly to the Strang split-

ting integrator. This is explained by the fact that, while the linear term is resolved exactly by the matrix exponential, the nonlinear term is not and so is subject to numerical error. The quality of the solution depends on the interaction of the oscillations through the nonlinearity, as we shall see in more detail in chapters 3 and 4.

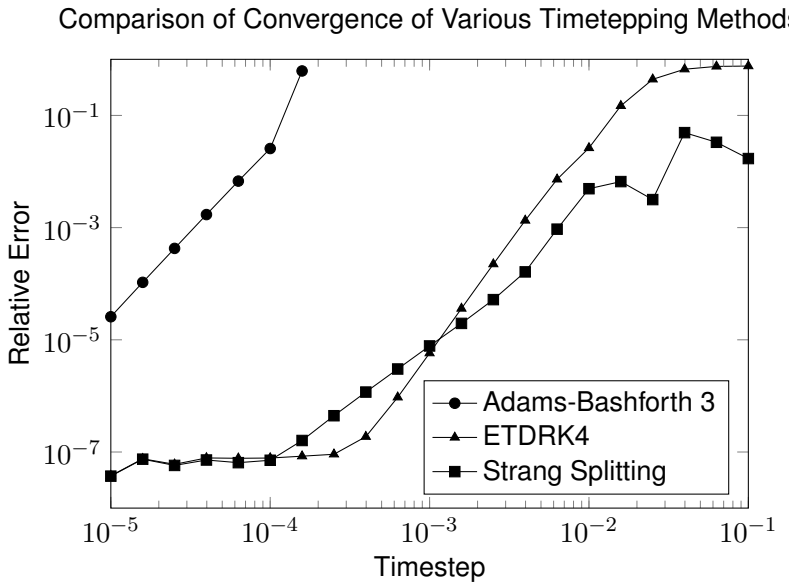

Figure 2.3: Exponential Integrator Convergence Studies, $\epsilon = 0.01$. The accuracy compared to a reference solution is shown as a function of the timestep. Note both the improved accuracy of the exponential integrator methods and the fact that stable computations were possible for timesteps approximately three orders of magnitude longer.

In this stiff case, it is fair to say that while ETDRK4 slightly outperforms Strang splitting for intermediate timesteps, both methods are similarly matched. This is due to the fact that at this degree of scale separation, the majority of the timestepping error arises from the fast waves, which are being similarly treated analytically through the exponential integrator in both cases.

Because of the implementational simplicity of the Strang splitting and the single nonlinear function call involved, we shall use exponential integrator-based Strang splitting for the remainder of this work. In particular, as we move on to performing numerical wave averages (*q.v.* Chapter 4) the simplicity of the method will prove useful both from the perspective of implementation and numerical performance. In the interest of making fair comparisons between both averaged and non-averaged computations, we will restrict ourselves to using the same numerical method in both cases.

This chapter has provided an introduction to both to the idea of using exponential integrators for numerical computation and the practical matter of constructing them for our RSWE system and pseudospectral method. In the next chapter, we will consider matrix exponentials in the analytical sense as they provide an elegant way of dealing with the particularities of solutions to the RSWE. We shall return to using matrix exponentials in the context of exponential integrators for computational purposes thereafter.

Key Points

- Exponential integrator methods aim to mitigate stiffness by analytically computing the linear term.
- When the linear operator of the system of differential equations is skew-Hermitian, we may efficiently compute the matrix exponential in its eigenvector basis.

3 *Wave Averaging and Triad Resonances*

> First Witch: Thrice the brinded cat hath mew'd.
> Second Witch: Thrice and once the hedgepig whined.
> Third Witch: Harpier cries 'Tis time, 'tis time.
>
> William Shakespeare, Macbeth, Act IV, Scene I.

It is no accident that we have been so explicit in earlier chapters in stating the quadratic nature of the nonlinear term in our general system under study. The study of this quadratic nonlinearity is fundamental to fluid mechanics and understanding of turbulence. As a direct result of this quadratic nonlinearity the transfer of energy between scales involves a set of three spatial modes called a *triad* (Kraichnan, 1958; Newell, 1969; Kramer et al., 2002; Kadri and Akylas, 2016).

Definition 3.1 (Triad Interaction)**.** A *triad interaction* or *triad* is a set of three wavevectors, $(\mathbf{k}, \mathbf{k}_1, \mathbf{k}_2)$ such that:

$$\mathbf{k} = \mathbf{k}_1 + \mathbf{k}_2. \tag{3.1}$$

▲

The concept of triadic interactions was first introduced by Kraichnan (1958), who showed that triadic interactions are conservative. Breakthroughs in the study of triadic interactions came as a result of high-performance computing in the early 1990s when Domaradzki and Rogallo (1990) analysed triads in direct numerical simulations of turbulent flow, finding that triads involving modes with very different wave numbers, so-called *nonlocal* triads, may exist and have very large amplitudes, although energy transfer is dominated by more local triads.

Another important result was due to Waleffe (1992), who numerically identified which triads contributed to forward and which to inverse energy transfers. This work was extended by Biferale et al. (2013), who used the same helical decomposition as Waleffe (1992) to build a 'decimated' version of the Navier-Stokes equa-

tions, where different classes of triadic interaction (*q.v.* Section 3.3) could be switched on and off. Decimation models of this type have been employed for rotating stratified turbulence by, e.g. Remmel et al. (2014) and Hernandez-Duenas et al. (2014).

We shall use the term *triadic* to refer to any interactions which satisfy the condition of equation (3.1). In dispersive flows, there is a second condition which may be considered in addition to that of triadicity, which is the *resonance* of the interaction. We then define a *directly-resonant* triad.[1]

[1] Some authors refer to this simply as a *resonant* triad, but the distinction between direct- and near-resonance is important to this work.

Definition 3.2 (Direct Resonance)**.** A *directly resonant triad*, often referred to as a *direct resonance*, is a triad (Definition 3.1) such that:

$$\omega(\mathbf{k}) = \omega(\mathbf{k}_1) + \omega(\mathbf{k}_2), \tag{3.2}$$

where $\omega(\mathbf{k})$ is the dispersion relation evaluated at wavevector $\mathbf{k}$.

▲

We will show in Section 3.2 that direct resonant interactions arise quite naturally and come to dominate the flow in the averaged equations in the asymptotic limit which were derived in Section 1.4. As noted by Clark di Leoni and Mininni (2016), "if a flow is dominated by rapidly varying waves, non-resonant interactions should, in principle, die out in front of resonant ones, thus leaving the bulk of the nonlinear energy transfer to the resonant triads". In fact, in the asymptotic limit of quasi-geostrophy, only the direct resonances remain to first order (Embid and Majda, 1996; Babin et al., 2000).

The restriction to direct resonant triads in the asymptotic limit leads to a decoupling between the slow manifold and the fast waves. This is sometimes thought of as the slow dynamics evolving independently of the fast, while the fast are 'swept' by the slow. Ward and Dewar (2010) use the term 'scattering' for this phenomenon, which is evocative of the way in which the fast inertia-gravity waves 'bounce off of' the slow PV waves. The concept of direct resonance has enjoyed widespread recognition in the geophysical community and has been applied in many geophysical wave models, e.g. Phillips (1968), Lelong and Riley (1991), and Embid and Majda (1998). The concept of direct resonance has also been applied in the context of weak turbulence by, e.g. Nazarenko (2011) and Newell and Rumpf (2011).

While much has been made of direct resonances, it has been known since as early as 1969 that near-resonant interactions play a role as well over particular timescales and for finite degrees of scale separation. Newell (1969) extended the direct resonance condition of Definition 3.2, and defined near-resonant interactions as those triads for which the sum of dispersion relations does not equal

zero, but rather some small finite value. That is to say,

$$\omega(\mathbf{k}) - \omega(\mathbf{k}_1) - \omega(\mathbf{k}_2) = \mathcal{O}(\varepsilon). \tag{3.3}$$

According to Newell (1969), in an asymptotic sense, these interactions are relevant on a timescale of $\mathcal{O}(1/\varepsilon)$. Similar findings were made by Chen et al. (2005), who concluded that for the rotating Navier-Stokes equations, direct resonant triads become more dominant as rotation is increased (which corresponds to the case as $\varepsilon \to 0$ for our problem). A particularly illustrative study was performed by Smith and Lee (2005), who numerically simulated decimated systems like those of Waleffe (1992), finding that for intermediate Rossby numbers consideration of purely direct resonances was insufficient to reproduce the characteristics of the flow, while the consideration of near-resonances up to $\mathcal{O}(\varepsilon)$ provided much better agreement with the full flow.

Analysis of a numerical data set by Alexakis (2015) showed that the quasi-two dimensional components of the rotating Navier-Stokes equations can only be modelled correctly if both direct- and near-resonances are considered. Work by Gallet (2015) suggested limitations of considering solely direct resonances even in the limit as $\varepsilon \to 0$ in the context of energy transfer.

In order to better understand near resonance and its relationship to wave averaging, it is instructive to begin with the somewhat stricter case of direct resonance. We will see how the quadratic nonlinearity gives rise to triad interactions, and how the averaging procedure (in the limit as $\varepsilon \to 0$) leads to only direct resonances. We will go on to show, numerically in Section 3.5 and later rigorously in Chapter 4, that near-resonant interactions are crucial to the evolution of the solution to differential equations of the type introduced in Chapter 1.

3.1 *Projecting to Different Bases*

The basis induced by the eigenvectors of the linear operator provides a more natural basis in which to discuss the effects of the wave averaging. We then need to take some time to discuss projections between different bases. Consider some quantity which may be represented in two orthonormal bases, which we shall call $\phi(x)$ and $\psi(x)$. Then we may represent the solution as either

$$u(x) = \sum_k \hat{u}_k^{\phi} \phi_k(x), \tag{3.4}$$

or in the second basis as

$$u(x) = \sum_k \hat{u}_k^{\psi} \psi_k(x). \tag{3.5}$$

If we know the coefficients for the first equation but not the second, we may solve for them as follows

$$\sum_k \hat{u}_k^\phi \phi_k(x) = \sum_k \hat{u}_k^\psi \psi_k(x). \tag{3.6}$$

We then multiply by the new basis and integrate over the domain:

$$\sum_k \hat{u}_k^\phi \int \phi_k(x)\psi_m(x)\,\mathrm{d}x = \sum_k \hat{u}_k^\psi \int \psi_k(x)\psi_m(x)\,\mathrm{d}x. \tag{3.7}$$

As we are here concerned only with orthogonal bases, the integrals are non-zero if and only if $m = k$, which yields:

$$\sum_k \hat{u}_k^\phi \int \phi_k(x)\psi_m(x)\,\mathrm{d}x = \hat{u}_m^\psi. \tag{3.8}$$

We may then rewrite the above in terms of the standard inner product and over all wavenumbers:

$$u(x) = \sum_m \sum_k \hat{u}_k^\phi \langle \phi_k(x), \psi_m(x)\rangle \psi_m(x). \tag{3.9}$$

3.2 *Resonance in Time*

We proceed by writing the unknowns of our system in an eigenvalue basis. As our linear operator is skew-Hermitian, the basis vectors in this basis are orthogonal. Without loss of generality, we will consider a vector wavenumber, $\mathbf{k}$, which may be a 1-vector in the case of the 1D RSWE. In the interest of generality, we use the two-dimensional form of the RSWE, as the one-dimensional form follows directly from that. In the interest of notational simplicity,[2] we will use the nondimensional form of the RSWE, noting that the results of this section hold in the dimensional case as well. Recall from Chapter 2 that the linear operator has three eigenvalues

[2] 'Simplicity' being here a very relative term.

$$i\omega_{\mathbf{k}}^\alpha = i\alpha\sqrt{1 + F^{-1}\mathbf{k}^2}, \quad \alpha = -1, 0, +1. \tag{3.10}$$

These eigenvalues, familiar to geophysicists as the dispersion relation, are written in terms of α for convenience. Here, $\alpha = 0$ corresponds to the 'slow' or 'potential vorticity' (PV) mode of the system, while $\alpha = \pm 1$ corresponds to fast, or 'inertia-gravity' waves travelling in opposite directions. The solution may be written in a Fourier basis as

$$\mathbf{u}(\mathbf{x}, t) = \sum_{\mathbf{k}\in\mathbb{Z}^2} \sum_{\alpha=-1}^{1} e^{i\mathbf{k}\cdot\mathbf{x}} \sigma_{\mathbf{k}}^\alpha(t) \mathbf{r}_{\mathbf{k}}^\alpha, \tag{3.11}$$

where $\sigma_{\mathbf{k}}^\alpha$ is the Fourier coefficient associated with wavenumber $\mathbf{k}$ and mode α and $\mathbf{r}_{\mathbf{k}}^\alpha$ is the right eigenvector of the linear operator. We then expand the vector $\mathbf{r}_{\mathbf{k}}^\alpha$ in the following form

$$\mathbf{u}(\mathbf{x}, t) = \sum_{\mathbf{k}} \sum_{\alpha} e^{i\mathbf{k}\cdot\mathbf{x}} \sigma_{\mathbf{k}}^\alpha(t) \begin{bmatrix} ru \\ rv \\ rh \end{bmatrix}_{\mathbf{k}}^{\alpha}, \tag{3.12}$$

where $\mathbf{r}_{\mathbf{k}}^{\alpha} = ([ru, rv, rh]_{\mathbf{k}}^{\alpha})^T$ exposes the three components of the right eigenvector. In this basis, applying the exponential integrator may be performed directly

$$e^{-\tau\mathcal{L}}\mathbf{u} = \sum_{\mathbf{k}}\sum_{\alpha} e^{i(\mathbf{k}\cdot\mathbf{x}-\omega_{\mathbf{k}}^{\alpha}\tau)}\sigma_{\mathbf{k}}^{\alpha}(t)\mathbf{r}_{\mathbf{k}}^{\alpha}. \tag{3.13}$$

Using this expression, we will compute the nonlinear term, $\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u})$. Beginning with the first component

$$\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u})_1 = u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}, \tag{3.14}$$

$$\begin{aligned}
&\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u})_1 = \\
&\left[\sum_{\mathbf{k}_1}\sum_{\alpha_1}\sigma_{\mathbf{k}_1} ru_{\mathbf{k}_1}^{\alpha_1} e^{i(\mathbf{k}_1\cdot\mathbf{x}-\omega_{\mathbf{k}_1}^{\alpha_1}\tau)}\right]\cdot\left[\sum_{\mathbf{k}_2}\sum_{\alpha_2} ik_2^1\sigma_{\mathbf{k}_2}^{\alpha_2} ru_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_2\cdot\mathbf{x}-\omega_{\mathbf{k}_2}^{\alpha_2}\tau)}\right] + \\
&\left[\sum_{\mathbf{k}_1}\sum_{\alpha_1}\sigma_{\mathbf{k}_1} rv_{\mathbf{k}_1}^{\alpha_1} e^{i(\mathbf{k}_1\cdot\mathbf{x}-\omega_{\mathbf{k}_1}^{\alpha_1}\tau)}\right]\cdot\left[\sum_{\mathbf{k}_2}\sum_{\alpha_2} ik_2^2\sigma_{\mathbf{k}_2}^{\alpha_2} ru_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_2\cdot\mathbf{x}-\omega_{\mathbf{k}_2}^{\alpha_2}\tau)}\right],
\end{aligned} \tag{3.15}$$

which we may tidy up to form

$$= \sum_{\mathbf{k}_1,\mathbf{k}_2}\sum_{\alpha_1,\alpha_2} i\sigma_{\mathbf{k}_1}^{\alpha_1}\sigma_{\mathbf{k}_2}^{\alpha_2} ru_{\mathbf{k}_2}^{\alpha_2}(k_2^1 ru_{\mathbf{k}_1}^{\alpha_1} + k_2^2 rv_{\mathbf{k}_1}^{\alpha_1}) e^{i(\mathbf{k}_1+\mathbf{k}_2)\cdot\mathbf{x} - i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau}. \tag{3.16}$$

The procedure for the second term is similar and is thus omitted here, but the result is:

$$\begin{aligned}
&\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u})_2 = \\
&\sum_{\mathbf{k}_1,\mathbf{k}_2}\sum_{\alpha_1,\alpha_2} i\sigma_{\mathbf{k}_1}^{\alpha_1}\sigma_{\mathbf{k}_2}^{\alpha_2} rv_{\mathbf{k}_2}^{\alpha_2}(k_2^1 ru_{\mathbf{k}_1}^{\alpha_1} + k_2^2 rv_{\mathbf{k}_1}^{\alpha_1}) e^{i(\mathbf{k}_1+\mathbf{k}_2)\cdot\mathbf{x} - i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau}.
\end{aligned} \tag{3.17}$$

Finally, we consider the third nonlinear term, which takes the form

$$\mathcal{N}(\mathbf{u}, \mathbf{u})_3 = \nabla\cdot(h\mathbf{v}) = h\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + u\frac{\partial h}{\partial x} + v\frac{\partial h}{\partial y}. \tag{3.18}$$

In terms of the Fourier basis

$$\begin{aligned}
&\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u})_3 = \\
&\qquad\left(\sum_{\mathbf{k}_1}\sum_{\alpha_1} rh_{\mathbf{k}_1}^{\alpha_1} e^{i(\mathbf{k}_1\cdot\mathbf{x}-\omega_{\mathbf{k}_1}^{\alpha_1}\tau)}\right)\cdot \\
&\left[\left(\sum_{\mathbf{k}_2}\sum_{\alpha_2}\sigma_{\mathbf{k}_2}^{\alpha_2} ik_2^1 ru_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_2\cdot\mathbf{x}-\omega_{\mathbf{k}_2}^{\alpha_2}\tau)}\right) + \left(\sum_{\mathbf{k}_2}\sum_{\alpha_2}\sigma_{\mathbf{k}_2}^{\alpha_2} ik_2^2 rv_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_2\cdot\mathbf{x}-\omega_{\mathbf{k}_2}^{\alpha_2}\tau)}\right)\right] \\
&\quad + \left[\sum_{\mathbf{k}_1}\sum_{\alpha_1}\sigma_{\mathbf{k}_1}^{\alpha_1} ru_{\mathbf{k}_1}^{\alpha_1} e^{i(\mathbf{k}_1\cdot\mathbf{x}-\omega_{\mathbf{k}_1}^{\alpha_1}\tau)}\right]\cdot\left[\sum_{\mathbf{k}_2}\sum_{\alpha_2}\sigma_{\mathbf{k}_2}^{\alpha_2} ik_2^1 rh_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_2\cdot\mathbf{x}-\omega_{\mathbf{k}_2}^{\alpha_2}\tau)}\right] \\
&\quad + \left[\sum_{\mathbf{k}_1}\sum_{\alpha_1}\sigma_{\mathbf{k}_1}^{\alpha_1} rv_{\mathbf{k}_1}^{\alpha_1} e^{i(\mathbf{k}_1\cdot\mathbf{x}-\omega_{\mathbf{k}_1}^{\alpha_1}\tau)}\right]\cdot\left[\sum_{\mathbf{k}_2}\sum_{\alpha_2}\sigma_{\mathbf{k}_2}^{\alpha_2} ik_2^2 rh_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_2\cdot\mathbf{x}-\omega_{\mathbf{k}_2}^{\alpha_2}\tau)}\right].
\end{aligned} \tag{3.19}$$

This simplifies to

$$\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u})_3 = \sum_{\mathbf{k}_1,\mathbf{k}_2}\sum_{\alpha_1,\alpha_2} i\sigma_{\mathbf{k}_1}^{\alpha_1}\sigma_{\mathbf{k}_2}^{\alpha_2}[rh_{\mathbf{k}_1}^{\alpha_1}(k_2^1 ru_{\mathbf{k}_2}^{\alpha_2} + k_2^2 rv_{\mathbf{k}_2}^{\alpha_2}) + rh_{\mathbf{k}_2}^{\alpha_2}(k_2^1 ru_{\mathbf{k}_1}^{\alpha_1} + k_2^2 rv_{\mathbf{k}_1}^{\alpha_1})]e^{i[(\mathbf{k}_1+\mathbf{k}_2)\cdot\mathbf{x} - \left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau]}. \quad (3.20)$$

Keeping in mind that the ru, rv, and rh terms denote vector components, we write

$$\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u}) = \sum_{\mathbf{k}_1}\sum_{\mathbf{k}_2}\sum_{\alpha_1}\sum_{\alpha_2} i\sigma_{\mathbf{k}_1}^{\alpha_1}\sigma_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_1+\mathbf{k}_2)\cdot\mathbf{x} - i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau}\Big[ru_{\mathbf{k}_2}^{\alpha_2}\hat{\mathbf{i}}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1}) + rv_{\mathbf{k}_2}^{\alpha_2}\hat{\mathbf{j}}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1}) + rh_{\mathbf{k}_1}^{\alpha_1}\hat{\mathbf{k}}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2}) + rh_{\mathbf{k}_2}^{\alpha_2}\hat{\mathbf{k}}(\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_1})\Big], \quad (3.21)$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ denote unit vectors such that

$$\mathbf{r}_{\mathbf{k}}^{\alpha} = ru_{\mathbf{k}}^{\alpha}\hat{\mathbf{i}} + rv_{\mathbf{k}}^{\alpha}\hat{\mathbf{j}} + rh_{\mathbf{k}}^{\alpha}\hat{\mathbf{k}}. \quad (3.22)$$

$\mathbf{rv}$ denotes a 2-vector comprised of the ru and rv components, i.e. $\mathbf{rv}_{\mathbf{k}}^{\alpha} = ru_{\mathbf{k}}^{\alpha}\hat{\mathbf{i}} + rv_{\mathbf{k}}^{\alpha}\hat{\mathbf{j}}$. As the wavenumbers and branches of the dispersion relation appear as dummy variables in the summations, we consider the nonlinear operator as $\mathcal{N} = 1/2 f(\mathbf{k}_1, \alpha_1; \mathbf{k}_2, \alpha_2) + 1/2 f(\mathbf{k}_2, \alpha_2; \mathbf{k}_1, \alpha_1)$, which gives

$$\begin{aligned}\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}, e^{-\tau\mathcal{L}}\mathbf{u}) = \sum_{\mathbf{k}_1}\sum_{\mathbf{k}_2}\sum_{\alpha_1}\sum_{\alpha_2} \frac{i}{2}\sigma_{\mathbf{k}_1}^{\alpha_1}\sigma_{\mathbf{k}_2}^{\alpha_2} e^{i(\mathbf{k}_1+\mathbf{k}_2)\cdot\mathbf{x} - i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau}\Big[&ru_{\mathbf{k}_2}^{\alpha_2}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})\,\hat{\mathbf{i}} + rv_{\mathbf{k}_2}^{\alpha_2}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})\hat{\mathbf{j}} + rh_{\mathbf{k}_1}^{\alpha_1}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})\hat{\mathbf{k}} + rh_{\mathbf{k}_2}^{\alpha_2}(\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})\hat{\mathbf{k}} + \\ &ru_{\mathbf{k}_1}^{\alpha_1}(\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})\hat{\mathbf{i}} + rv_{\mathbf{k}_1}^{\alpha_1}(\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})\hat{\mathbf{j}} + \\ &rh_{\mathbf{k}_2}^{\alpha_2}(\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})\hat{\mathbf{k}} + rh_{\mathbf{k}_1}^{\alpha_1}(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})\hat{\mathbf{k}}\Big]. \quad (3.23)\end{aligned}$$

This expression simplifies quite neatly in terms of our basis vector

$$\begin{aligned}\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}(\mathbf{x},t), e^{-\tau\mathcal{L}}\mathbf{u}(\mathbf{x},t)) = \sum_{\mathbf{k}_1}\sum_{\mathbf{k}_2}\sum_{\alpha_1}\sum_{\alpha_2} \frac{i}{2}\sigma_{\mathbf{k}_1}^{\alpha_1}(t)\sigma_{\mathbf{k}_2}^{\alpha_2}(t) e^{i(\mathbf{k}_1+\mathbf{k}_2)\cdot\mathbf{x} - i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau}\Big[&(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})\mathbf{r}_{\mathbf{k}_1}^{\alpha_1} + \\ &(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})\mathbf{r}_{\mathbf{k}_2}^{\alpha_2} + (\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})rh_{\mathbf{k}_1}^{\alpha_1}\hat{\mathbf{k}} + (\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})rh_{\mathbf{k}_2}^{\alpha_2}\hat{\mathbf{k}}\Big]. \quad (3.24)\end{aligned}$$

We are finally interested in projecting the solution onto the basis $\sum_{\mathbf{k}}\sum_{\alpha=-1}^{1}\sigma_{\mathbf{k}}^{\alpha}(t)e^{i\mathbf{k}\cdot\mathbf{x}}\mathbf{r}_{\mathbf{k}}^{\alpha}$. This is done in the same fashion as Section 3.1, noting that the rh terms are purely real and so identical to their

complex conjugates. Applying this projection yields equation (3.25) below, where the exponentials are written independently of the interaction coefficient (3.26) to obtain the three-wave interaction condition in the form given by Majda (2002)

$$\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}(\mathbf{x},t), e^{-\tau\mathcal{L}}\mathbf{u}(\mathbf{x},t)) = \sum_{\mathbf{k}\in\mathbb{Z}^2}\sum_{\alpha=-1}^{1}\left[\sum_{\mathbf{k}=\mathbf{k}_1+\mathbf{k}_2}\sum_{\alpha_1,\alpha_2}\sigma_{\mathbf{k}_1}^{\alpha_1}(t)\sigma_{\mathbf{k}_2}^{\alpha_2}(t)C_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2}^{\alpha,\alpha_1\alpha_2}e^{i(\mathbf{k}\cdot\mathbf{x})-i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}\right)\tau}\right]\mathbf{r}_{\mathbf{k}}^{\alpha}, \tag{3.25}$$

where the interaction coefficient is

$$C_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2}^{\alpha,\alpha_1,\alpha_2} = \frac{i}{2}[(\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})\langle\mathbf{r}_{\mathbf{k}_2}^{\alpha_2},\mathbf{r}_{\mathbf{k}}^{\alpha}\rangle + (\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})\langle\mathbf{r}_{\mathbf{k}_1}^{\alpha_1},\mathbf{r}_{\mathbf{k}}^{\alpha}\rangle + (\mathbf{k}_1\cdot\mathbf{rv}_{\mathbf{k}_1}^{\alpha_1})rh_{\mathbf{k}_2}^{\alpha_2}rh_{\mathbf{k}}^{\alpha} + (\mathbf{k}_2\cdot\mathbf{rv}_{\mathbf{k}_2}^{\alpha_2})rh_{\mathbf{k}_1}^{\alpha_1}rh_{\mathbf{k}}^{\alpha}]. \tag{3.26}$$

The interaction coefficient governs the interaction between two waves, denoted by subscripts 1 and 2, passing into the nonlinearity and the outgoing wave, denoted without subscript. We are now finally in a position to consider the full right-hand side term of interest, by including the remaining matrix exponential from equation (1.36)

$$e^{\tau\mathcal{L}}\mathcal{N}(e^{-\tau\mathcal{L}}\mathbf{u}(\mathbf{x},t), e^{-\tau\mathcal{L}}\mathbf{u}(\mathbf{x},t)) = \sum_{\mathbf{k}\in\mathbb{Z}^2}\sum_{\alpha=-1}^{1}\left[\sum_{\mathbf{k}=\mathbf{k}_1+\mathbf{k}_2}\sum_{\alpha_1,\alpha_2}\sigma_{\mathbf{k}_1}^{\alpha_1}(t)\sigma_{\mathbf{k}_2}^{\alpha_2}(t)C_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2}^{\alpha,\alpha_1\alpha_2}e^{i(\mathbf{k}\cdot\mathbf{x})-i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}-\omega_{\mathbf{k}}^{\alpha}\right)\tau}\right]\mathbf{r}_{\mathbf{k}}^{\alpha}. \tag{3.27}$$

Considering the averaging integral and the limit gives

$$\frac{\partial\overline{\mathbf{u}}}{\partial t} = -\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^{\tau}e^{s\mathcal{L}}\mathcal{N}(e^{-s\mathcal{L}}\mathbf{u}(\mathbf{x},t), e^{-s\mathcal{L}}\mathbf{u}(\mathbf{x},t))\,\mathrm{d}s \tag{3.28}$$

$$= -\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^{\tau}\sum_{\mathbf{k}\in\mathbb{Z}^2}\sum_{\alpha=-1}^{1}\left[\sum_{\mathbf{k}=\mathbf{k}_1+\mathbf{k}_2}\sum_{\alpha_1,\alpha_2}\sigma_{\mathbf{k}_1}^{\alpha_1}(t)\sigma_{\mathbf{k}_2}^{\alpha_2}(t)C_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2}^{\alpha,\alpha_1\alpha_2}e^{i(\mathbf{k}\cdot\mathbf{x})-i\left(\omega_{\mathbf{k}_1}^{\alpha_1}+\omega_{\mathbf{k}_2}^{\alpha_2}-\omega_{\mathbf{k}}^{\alpha}\right)s}\right]\mathbf{r}_{\mathbf{k}}^{\alpha}\,\mathrm{d}s, \tag{3.29}$$

Equation (3.29) is conceptually equivalent to equation (1.36), but is expressed in the eigenbasis of the linear operator and with the nonlinearity made more explicit. While complicated, it makes clear the nature of the nonlinear interactions in this equation. For our purposes, the interaction coefficient is not the most important term, but rather the presence of the sum of dispersion relations in the exponent, which should look familiar from the definition of direct resonances, Definition 3.2.

3.3 *Direct Resonances*

THE INTEGRAL in equation (3.29) is over the variable s, which denotes the fast time in the problem. This variable appears only in the exponential, i.e. $\sigma^{\alpha_1}_{\mathbf{k}_1} = \sigma^{\alpha_1}_{\mathbf{k}_1}(t)$ and $\sigma^{\alpha_2}_{\mathbf{k}_2} = \sigma^{\alpha_2}_{\mathbf{k}_2}(t)$. Consider then the integral of the complex exponential

$$I^{\alpha,\alpha_1,\alpha_2}_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2} = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau e^{i(\mathbf{k}\cdot\mathbf{x}) - i\left(\omega^{\alpha}_{\mathbf{k}} - \omega^{\alpha_1}_{\mathbf{k}_1} - \omega^{\alpha_2}_{\mathbf{k}_2}\right)s} \, \mathrm{d}s, \tag{3.30}$$

where $\mathbf{k} = \mathbf{k}_1 + \mathbf{k}_2$. When $\omega^{\alpha}_{\mathbf{k}} \neq \omega^{\alpha_1}_{\mathbf{k}_1} + \omega^{\alpha_2}_{\mathbf{k}_2}$, the integral is over an oscillatory quantity. As the length of the domain of integration[3] tends to infinity, these oscillations integrate to zero by the Riemann-Lebesgue lemma (Tolstov, 1962). The implication is that the only interactions which do not integrate to zero, which is to say that they pass through the averaging procedure, are the direct three-wave resonances from Definition 3.2

[3] The 'length of the averaging window' from now on.

$$\mathbf{k} = \mathbf{k}_1 + \mathbf{k}_2, \qquad \omega^{\alpha}_{\mathbf{k}} = \omega^{\alpha_1}_{\mathbf{k}_1} + \omega^{\alpha_2}_{\mathbf{k}_2}. \tag{3.31}$$

When this directly resonant condition is satisfied the s-term in the complex exponential is zero and no oscillations occur to be cancelled out on this timescale.[4] Subject to the simplification induced by the infinite limit, we may then finally say that the infinitely-wave-averaged solution (*cf.* equations (1.36) and (3.29)) follows

[4] This is closely related to the thinking behind the *method of stationary phase* (Temme, 2013).

$$\frac{\partial \sigma^{\alpha}_{\mathbf{k}}}{\partial t} + \sum_{\mathcal{S}_{\mathbf{k},\alpha}} \sigma^{\alpha_1}_{\mathbf{k}_1} \sigma^{\alpha_2}_{\mathbf{k}_2} C^{\alpha,\alpha_1\alpha_2}_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2} = 0, \tag{3.32}$$

where the sum is taken over the *resonant set*, $\mathcal{S}_{\mathbf{k},\alpha}$, which is defined as

$$\mathcal{S}_{\mathbf{k},\alpha} \equiv \{(\mathbf{k}_1, \mathbf{k}_2, \alpha_1, \alpha_2) : \mathbf{k} = \mathbf{k}_1 + \mathbf{k}_2, \quad \omega^{\alpha}_{\mathbf{k}} = \omega^{\alpha_1}_{\mathbf{k}_1} + \omega^{\alpha_2}_{\mathbf{k}_2}\}. \tag{3.33}$$

In the limit as $\varepsilon \to 0$, we may consider the solution as being comprised solely of these interacting direct resonances. In the interest of visualising this, Figure 3.1 shows the direct *resonant trace* for a given wavevector, in this case $\mathbf{k} = (4, 8)$.

Recall the triad and resonance conditions given in equation (3.31). For any three waves to form a triad, their vector sum must be zero. Then in a two-dimensional wavenumber space, any closed triangle created by three modes is a triad. As discussed earlier in this chapter, triads may be, but are not necessarily, resonant, but due to the quadratic nonlinearity the solution is comprised entirely of triads.[5]

[5] Some authors, e.g. Hammack and Henderson (1993), consider higher-order interactions such as quartets. While there are interesting asymptotic ramifications of that, due to our interest in numerical modelling we will consider these interactions to be expressible as sums of triads.

Not all interaction types satisfy the resonance condition of equation (3.31), for all wavenumbers. Considering the case where $\mathbf{k}$ and one of the other waves are an inertia-gravity mode and the third mode is a PV mode (which we will show in the next section to be

the only nontrivial direct resonance for the RSWE), we can visualise the resonant set in terms of the locus of points along which the resonant condition is satisfied. The curves shown in Figure 3.1 are those along which

$$\omega_{\mathbf{k}}^{\pm 1} = \omega_{\mathbf{k}_1}^{\pm 1} + \omega_{\mathbf{k}_2}^{0}, \tag{3.34}$$

and its complementary resonance in the other direction. A different set of loci could be produced for any different fixed wavevector, $\mathbf{k}$. In the numerical case of restriction to integer wavenumbers, only locations where the locus intersects the grid lead to a resonant triad.

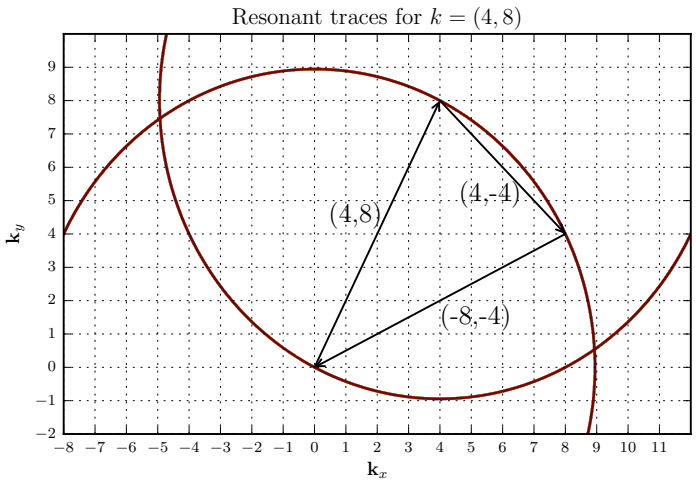

Figure 3.1: The loci of the direct resonances for a triad where $\mathbf{k} = (4, 8)$. The grid corresponds to discrete, integer wavenumbers. As we are considering vector sums, any closed triangle forms a triad. Only those triads where all three vertices sit on the resonant trace are directly resonant. In this case, there is only one more direct resonance, which trivially corresponds to taking the sum in the other direction due to symmetry in the system.

The traces which we have shown here are specifically for the RSWE. If another dispersive system with quadratic nonlinearity were to be studied, such as the Navier-Stokes or Boussinesq equations, the shape of these loci would change. Smith and Waleffe (1999) give equivalent loci for the rotating Navier-Stokes equations, for example. The theory which we discuss in the next chapter does not rely on the specific shape of these loci, rather only on the existence of triadic interactions and the consideration of the RSWE is for expository and numerical purposes.[6]

[6] As an interesting aside, for directly resonant interactions to occur for the Korteweg-de Vries (KdV) equation would require that $k^3 - k_1^3 - k_2^3 = 0$, which should be familiar to the reader as *Fermat's Last Theorem*. This equation was shown by Wiles (1995) to admit no non-trivial solutions and so the KdV equation admits directly resonant interactions only for a very restricted set of modes. However, it is worth pointing out that arbitrarily close 'near-miss' solutions exist, such as that found by Homer Simpson in *Treehouse of Horror VI* (Singh, 1997).

3.3.1 *Allowable Interactions for the RSWE*

Embid and Majda (1996) and Majda (2002) give some restrictions on which types of waves may interact to create resonances. There are only two interaction types which are allowable in the small-ε limit. We will proceed by showing that all other interaction types are not possible in lemmas 3.1 through 3.4. In terms of interactions between three fast waves, we have the following lemma.

Lemma 3.1 (Gravity Wave Interactions)**.** *No directly resonant interactions occur solely among the fast modes in the averaged equations.* ♦

Proof. Assume that the wavevector condition is satisfied for three fast waves, i.e.

$$\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}. \tag{3.35}$$

The resonance condition then requires that

$$\omega(\mathbf{k}_1) \pm \omega(\mathbf{k}_2) = \omega(\mathbf{k}), \tag{3.36}$$

where we may reduce the plus-or-minus to addition by interchanging $\mathbf{k}$ and $\mathbf{k}_2$ by the fact that $\omega(\mathbf{k})$ is a radial function. This equality is never satisfied, as

$$\omega(\mathbf{k}_1 + \mathbf{k}_2) < \omega(\mathbf{k}_1) + \omega(\mathbf{k}_2). \tag{3.37}$$

Since $\omega(\mathbf{k}) = \sqrt{1 + F^{-1}|\mathbf{k}|^2}$, this result follows directly from

$$\sqrt{1 + (x+y)^2} < \sqrt{1 + x^2} + \sqrt{1 + y^2}. \tag{3.38}$$

□

Lemma 3.2. *There are no interactions between two inertia-gravity modes and a PV mode.* ♦

Proof. Since the dispersion relation is a radial function for the RSWE, the wavenumbers must have equivalent magnitudes, i.e.

$$\sqrt{1 + F^{-1}|\mathbf{k}_1|^2} - \sqrt{1 + F^{-1}|\mathbf{k}_2|^2} = 0 \tag{3.39}$$

$$\implies |\mathbf{k}_1| = |\mathbf{k}_2|. \tag{3.40}$$

It may then be directly computed from equation (3.26) that

$$C^{0,+1,-1}_{\mathbf{k}_1+\mathbf{k}_2,\mathbf{k}_1,\mathbf{k}_2} = C^{0,-1,+1}_{\mathbf{k}_1+\mathbf{k}_2,\mathbf{k}_1,\mathbf{k}_2} = 0 \qquad \text{for } |\mathbf{k}_1| = |\mathbf{k}_2|. \tag{3.41}$$

□

Lemma 3.3. *Two incoming PV waves can not interact with an outgoing inertia-gravity wave.* ♦

Proof. Two slow modes resonating with a fast mode would require

$$\begin{aligned} \omega^{\alpha=\pm1}_{\mathbf{k}} &= \omega^0_{\mathbf{k}_1} + \omega^0_{\mathbf{k}_2}, \\ &= 0, \end{aligned} \tag{3.42}$$

which is a contradiction for

$$\omega^{\pm1}_{\mathbf{k}} = \pm\sqrt{1 + F^{-1}|\mathbf{k}|} \neq 0. \tag{3.43}$$

□

Lemma 3.4. *A fast and a slow mode may not resonate with an outgoing slow mode.* ♦

The proof of this follows the same logic as that of Lemma 3.3 and is omitted. Combining these lemmas allows us to describe exactly which resonances exist in the limit as $\varepsilon \to 0$.

Theorem 3.1 (Allowable Interactions)**.** *Consider the averaged rapidly-rotating shallow water equations* (3.29)*. In the limit as* $\varepsilon \to 0$*, only two direct resonant interactions are possible:*

1. *Interaction between three slow (PV) waves.*
2. *Interaction between an incoming gravity and an incoming PV mode with an outgoing gravity mode.*

♦

Proof. 1. All triadic interactions between three PV modes are trivially directly resonant, as

$$\omega^0_{\mathbf{k}} + \omega^0_{\mathbf{k}_1} + \omega^0_{\mathbf{k}_2} = 0 + 0 + 0 = 0, \quad \forall\, \mathbf{k}, \mathbf{k}_1, \mathbf{k}_2. \tag{3.44}$$

2. With reference to Figure 3.1, it is shown by example that at least one direct resonance exists between an incoming PV and inertia-gravity wave and an outgoing inertia-gravity wave. Consider

$$\mathbf{k} = (4, 8); \quad \mathbf{k}_1 = (4, -4); \quad \mathbf{k}_2 = (-8, -4). \tag{3.45}$$

Clearly, the triad condition is satisfied as $\mathbf{k} + \mathbf{k}_1 + \mathbf{k}_2 = 0$. Considering the case where $\mathbf{k}_1$ is the PV mode, we have

$$\begin{aligned} \omega^+_{\mathbf{k}} &= \omega^0_{\mathbf{k}_1} + \omega^+_{\mathbf{k}_2}, \\ \sqrt{1 + F^{-1}|(4,8)|} &= 0 + \sqrt{1 + F^{-1}|(-8,-4)|}. \end{aligned} \tag{3.46}$$

By the properties of the magnitude of a vector, this equality is satisfied and so this triad is directly resonant.

All other interaction types which would otherwise occur have been excluded by Lemmas 3.1 through 3.4. □

The implication of Theorem 3.1 is that in the limit, PV modes interact only amongst themselves. Combined with the second part of the theorem, this gives the concept of a *fast singular limit*, where the slow dynamics evolve independently of the fast and the fast dynamics are 'swept' by the slow. This holds in the mathematically convenient but nonphysical case as $\varepsilon \to 0$, which corresponds to

an upper limit on the averaging integral of $\tau \to \infty$. For realistic geophysical flows, ε is finite and may be of order one or greater. In order to engage with this in the framework in which we have been working, we return in earnest to the concept of *near-resonance*.

3.4 Near Resonant Interactions

We have shown how triads arise as a consequence of the quadratic nonlinearity, and that the averaging procedure permits only directly-resonant triads in the limit as $\varepsilon \to 0$. For finite ε we instead consider a finite average. In this case scale separation is no longer infinite and so complete cancellation of oscillations does not occur.

When a finite average is taken, the solution set is larger than the direct resonant set and this has an important effect on the quality of numerical methods based on finite wave averaging. When ε is finite, we define concentric shells of near-resonances, i.e. we rewrite the directly resonant triad-based form (3.32) as

$$\begin{aligned} e^{s\mathcal{L}/\varepsilon}\mathcal{N}(e^{-s\mathcal{L}/\varepsilon}\overline{\mathbf{u}}(t), e^{-s\mathcal{L}/\varepsilon}\overline{\mathbf{u}}(t)) &= \sum_{\lambda_n} e^{i\lambda_n s}\mathcal{N}_n(\overline{\mathbf{u}}(t)) \\ &= \sum_{\mathcal{S}_{\mathbf{k},\alpha}} \mathcal{N}_n(\overline{\mathbf{u}}(t)) + \\ &\quad \sum_{\mathcal{S}^{\epsilon_1}_{\mathbf{k},\alpha}} e^{i\lambda_n s}\mathcal{N}_n(\overline{\mathbf{u}}(t)) + \\ &\quad \sum_{\mathcal{S}^{\epsilon_2}_{\mathbf{k},\alpha}} e^{i\lambda_n s}\mathcal{N}_n(\overline{\mathbf{u}}(t)) + \dots, \end{aligned}$$

which we collapse to form

$$\begin{aligned} &e^{s\mathcal{L}/\varepsilon}\mathcal{N}(e^{-s\mathcal{L}/\varepsilon}\overline{\mathbf{u}}(t), e^{-s\mathcal{L}/\varepsilon}\overline{\mathbf{u}}(t)) = \\ &\qquad \sum_{\mathcal{S}_{\mathbf{k},\alpha}} \mathcal{N}_n(\overline{\mathbf{u}}(t)) + \sum_{\beta=1}^{\infty} \left(\sum_{\mathcal{S}^{\epsilon_\beta}_{\mathbf{k},\alpha}} e^{i\lambda_n s}\mathcal{N}_n(\overline{\mathbf{u}}(t)) \right), \end{aligned} \tag{3.47}$$

where one of the sums is over the direct resonant set given by equation (3.33) and the other is over progressively more distant near-resonant sets, $\mathcal{S}^{\epsilon_\beta}_{\mathbf{k},\alpha}$.

Definition 3.3 (Near Resonant Set)**.** $\mathcal{S}^{\epsilon_\beta}_{\mathbf{k},\alpha}$, $\beta = 1, 2, \dots$ is a near-resonant set, which is the set of all triads such that:

$$\mathcal{S}^{\epsilon_\beta}_{\mathbf{k},\alpha} = \left\{ (\mathbf{k}_1, \mathbf{k}_2, \alpha_1, \alpha_2) : \mathbf{k} = \mathbf{k}_1 + \mathbf{k}_2, \quad \epsilon_{\beta-1} < |\omega^{\alpha}_{\mathbf{k}} - \omega^{\alpha_1}_{\mathbf{k}_1} + \omega^{\alpha_2}_{\mathbf{k}_2}| \leq \epsilon_\beta \right\}, \tag{3.48}$$

where $\epsilon_0 = 0$ by definition. ▲

The direct-resonant set results in a solution consisting of only the slow dynamics of the system. As we will see in Chapter 4, the effect of the averaging procedure is to variably retain and reject triads of various degrees of near-resonance, and the extent to which it does this is fundamental to the quality of the averaged numerical approximation.

Figure 3.2 shows the near-resonant traces up to a nearness of 0.1 for the RSWE, and for the same wavevector ($\mathbf{k} = (4, 8)$) as in Figure 3.1. Here, any closed triad which sits on or inside the curves is admissible as a *fast-slow-fast* near resonance. While in the direct case we saw that there were only two admissible triads in this range of wavenumbers, there are significantly more possibilities in this case.

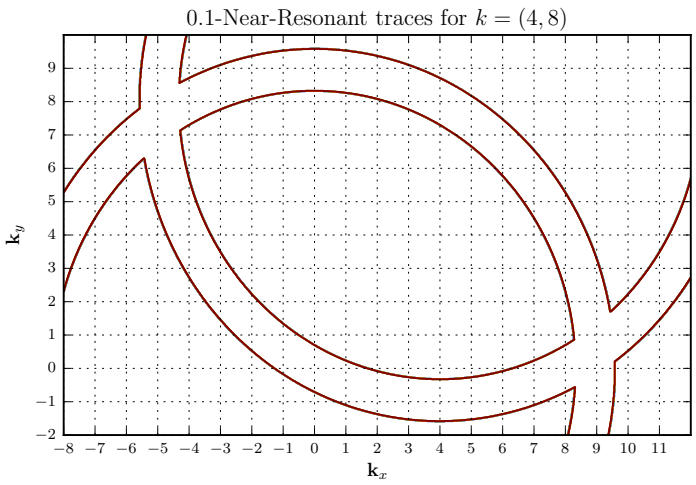

Figure 3.2: The near-resonant traces for $\mathbf{k} = (4, 8)$ which considers *fast-slow-fast* resonances of nearness up to $\epsilon_\beta = 0.1$. By analogy with Figure 3.1 which admitted only two resonant triads, it is apparent here that significantly more near-resonances are admissible. Near-resonant triads, visually speaking, are any closed triangles where one side is the vector from the origin to (4,8), and where the free vertex is constrained by the curves.

As we permit progressively farther-resonant sets, the total number of triads which are retained increases, as a comparison between Figures 3.1 and 3.2 should make clear. In addition to the inclusion of progressively more *slow-fast-slow* triads, additional types of interaction also become possible which did not satisfy the stringent conditions of a direct resonance (*cf.* Section 3.3.1).

Figure 3.3 shows the number of allowably-near-resonant triads whose outgoing wave is $\mathbf{k} = (4, 8)$ which are retained as $\max \epsilon_\beta$ increases, on a 32^2 wavenumber space. As discussed above, all *slow-slow-slow* interactions are trivially resonant, and so we see no change in the total number of these. Note that even a slight increase in the upper bounds on near-resonance immediately leads to the inclusion of new resonances while some of the more pathological cases (such as two gravity waves in the same direction interacting with a gravity wave travelling in the opposite direction) require a much looser definition of near resonance.

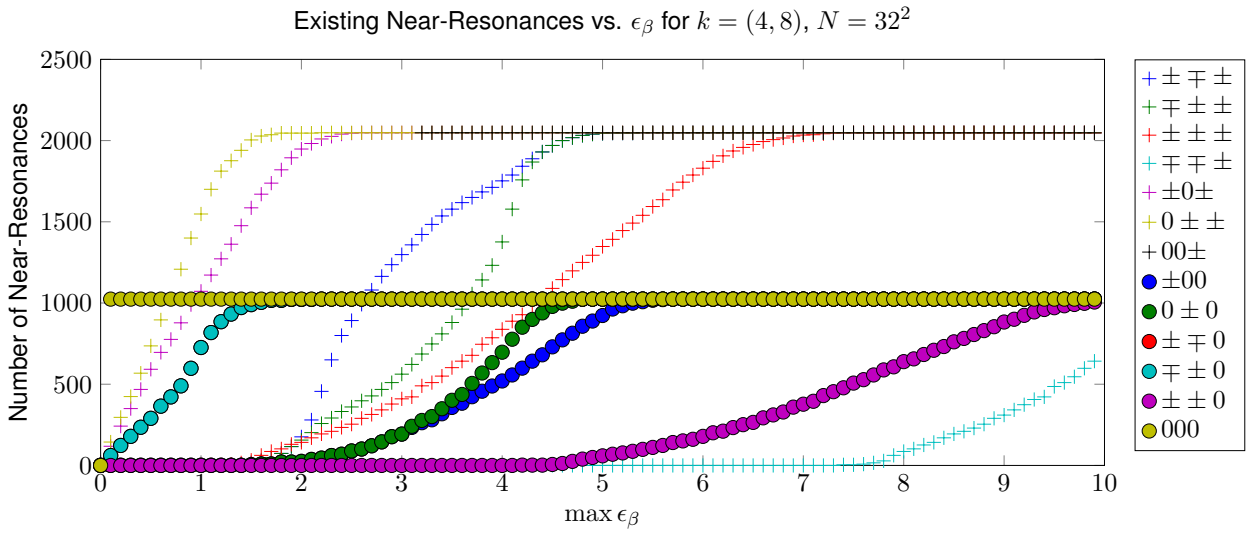

Figure 3.3: The increasing number of admissible triads in shown as the maximum distance, $\max \epsilon_\beta = \max \omega^\alpha_{\mathbf{k}} - \omega^{\alpha_1}_{\mathbf{k}_1} - \omega^{\alpha_2}_{\mathbf{k}_2}$ increases. The *slow-slow-slow* interaction is trivially resonant and so all triads of this type are admitted everywhere. For certain modes which more easily lend themselves to near-resonant interaction, saturation is observed for small values of ϵ_β, while some more pathological triads do not even begin to appear until much later. $\pm$ denotes a 'fast' or 'inertia-gravity' mode, while 0 denotes a 'slow' or 'PV' mode. The legend follows the same convention as the superscripts in the interaction coefficient, i.e. $\alpha, \alpha_1, \alpha_2$.

3.5 *A Near-Resonant Solver*

We have implemented a numerical solver which solves the rotating shallow water equations on near-resonant sets only. This amounts to numerically solving the explicit form of the averaged RSWE, equation (3.25), over the near-resonant sets in equation (3.48), such that the sum over all $\mathbf{k} = \mathbf{k}_1 + \mathbf{k}_2$ is further restricted to those triads which lie in the near-resonant set described in Definition 3.3, i.e.

$$\frac{\partial \overline{\mathbf{u}}(\mathbf{x}, t, \epsilon_\beta)}{\partial t} = - \sum_{\mathbf{k}\in\mathbb{Z}^2} \sum_{\alpha=-1}^{1} \sum_{\substack{\mathbf{k}=\mathbf{k}_1+\mathbf{k}_2 \\ |\omega^\alpha_{\mathbf{k}} - \omega^{\alpha_1}_{\mathbf{k}_1} - \omega^{\alpha_2}_{\mathbf{k}_2}| \le \epsilon_\beta}} \sum_{\alpha_1,\alpha_2} \sigma^{\alpha_1}_{\mathbf{k}_1}(t)\sigma^{\alpha_2}_{\mathbf{k}_2}(t) C^{\alpha,\alpha_1\alpha_2}_{\mathbf{k},\mathbf{k}_1,\mathbf{k}_2} e^{i(\mathbf{k}\cdot\mathbf{x}) - i\left(\omega^{\alpha_1}_{\mathbf{k}_1} + \omega^{\alpha_2}_{\mathbf{k}_2} - \omega^\alpha_{\mathbf{k}}\right)s} \mathbf{r}^\alpha_{\mathbf{k}} \, \mathrm{d}s, \tag{3.49}$$

with the interaction coefficient as in equation (3.26) and the eigenbasis is unchanged from Section 3.2. The transformation from the averaged to the full solution via the exponential linear operator is as before. In computing the sum, triads of a sufficiently near resonance are retained and all others are rejected. This process of retaining and rejecting triads is boolean in the sense that triads of sufficent nearness are *entirely* retainined in the solution, while all triads whose resonance lies outside the cutoff are completely rejected. Note that here there is no averaging integral which is evaluated either analytically or numerically. Rather, this process is intended to model the effect of such an integral.

Solving the RSWE by this method is computationally unfeasible for practical simulations, but serves to better investigate the role of near resonance in the quality of the average. Two experiments were carried out: one with $\varepsilon = 0.01$ which approximates the limit where only direct resonances should contribute, and one with $\varepsilon = 0.1$ which is in the finite-ε regime. In both runs, the timestep used for the decimated equations was $\Delta T = 0.2$ and for the full equations it was $\Delta t = 0.0002$ over a full simulation time of $T_f = 5$. The much longer timestep in the coarse case is because we are, in practice, interested in this as a method of handling oscillatory stiffness and performing much longer timesteps than the non-averaged equations allow. An initial Gaussian height field was used, and the 1-D RSWE was solved on a spatial grid of size $N_x = 32$.

It is expected that in the limit as $\varepsilon \to 0$, approximated here with $\varepsilon = 0.01$, optimal performance should be seen for a solver which retains only the direct resonances. Optimal performance is meant in the sense of minimal error, where the error is defined as the norm of the difference between the true and numerical solutions:

$$\text{error} = \|h_{\text{true}} - h_{\text{num}}\|. \tag{3.50}$$

As shown in Figure 3.4, this is indeed the case. As progressively farther resonant sets are included, the error increases as compared to the full solution. The minimum error was achieved by including only direct resonances, i.e. where $\max(|\omega_k^\alpha - \omega_{k_1}^{\alpha_1} - \omega_{k_2}^{\alpha_2}|) = 0$, which corresponds to an infinitely-long averaging window.

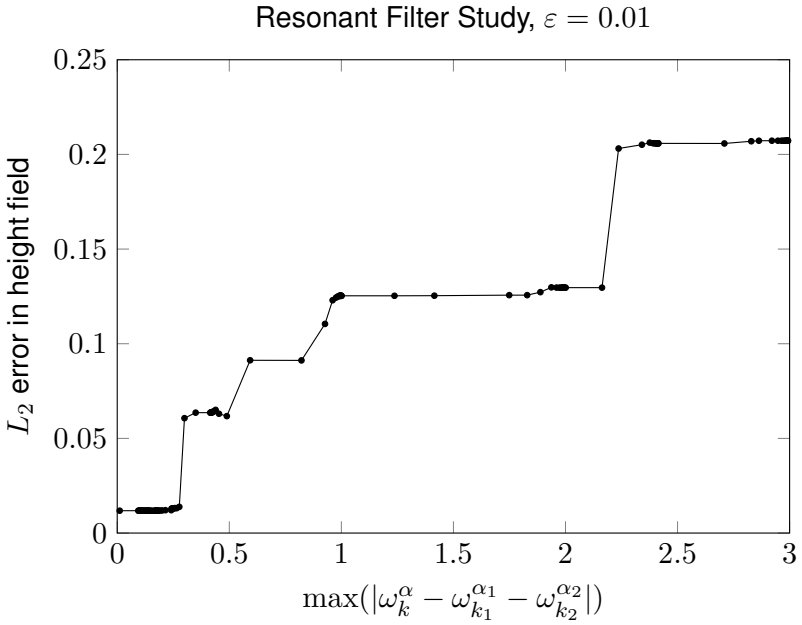

Figure 3.4: The measured error in the decimated RSWE as a function of the nearness of resonance required for a triad to be retained. This case closely approximates the asymptotic limit, with optimal performance occurring when only the direct resonances are retained.

In both Figures 3.4 and 3.5 the circles correspond to a unique model run. They are not evenly-spaced due to the fact that the inclusion of new triads doesn't occur linearly as $\max \epsilon_\beta$ increases,

as seen in Figure 3.3. Only those locations when at least one new triad was found were included in these experiments.

The small-ε case behaved exactly as the asymptotic theory predicted.[7] However, in the case of finite scale separation, i.e. intermediate ε, there is no asymptotic theory to predict the behaviour. Recalling the numerical results from Chapter 1, it was apparent that some optimal averaging window exists for finite scale separation. If that effect is related to the inclusion and rejection of near-resonant triads, we should expect to see something similar here.

[7] Our experiments have shown that numerical simulations with $\varepsilon \leq 0.01$ behave as though $\varepsilon \to 0$.

Figure 3.5 shows the error bounds for an intermediate value of $\varepsilon = 0.1$. Error was minimised by retaining triads up to a certain degree of near-resonance. This is a markedly different result from the simulation with $\varepsilon = 0.01$, where the optimal limit on triad distance was to retain only the direct resonances. It also agrees with the work of, for example, Smith and Lee (2005) or Chen et al. (2005), both of whom found that near-resonances play an important role at intermediate degrees of scale separation.

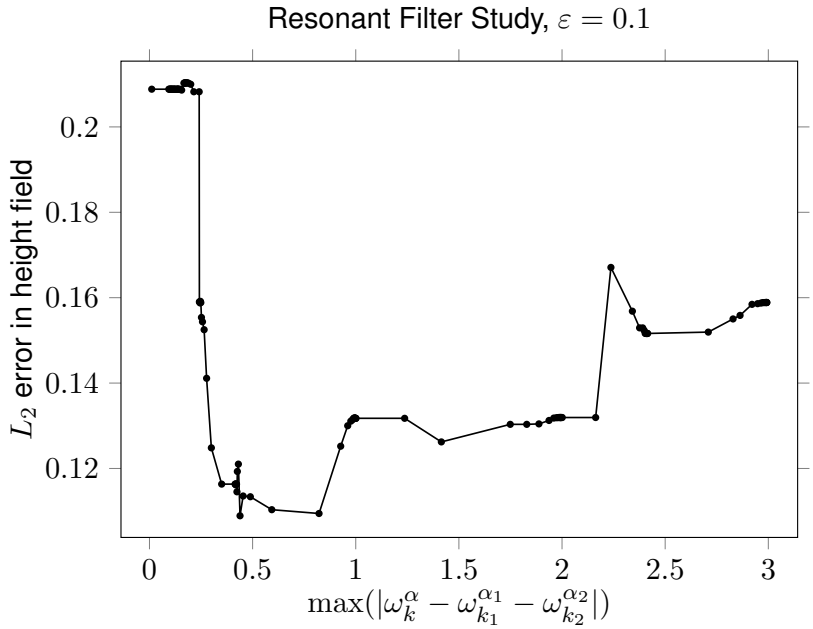

Figure 3.5: The measured error in the decimated RSWE as a function of the nearness required for a triad to be retained. $\varepsilon = 0.1$ is an intermediate value and leads to the existence of an optimal degree of nearness, similar to the averaging window width optimal.

This result supports the theory that near-resonant triads, which are effectively individual units of nonlinear oscillation, play a role in the quality of the solution found through numerical averaging. In the next chapter we will explore in detail how the nearness of resonance impacts numerical stiffness. We will also explain the existence of the optimal averaging window. For the remainder of this work, we will compute the averaging integral in equation (1.36) directly in the interests of both computational efficiency and accuracy.[8] As will be seen, the explicit consideration of triad resonances is not necessary for their existence or effect on the solution.

[8] This also allows us to apply the averaging method to models which do not use a Fourier spectral method.

Key Points

- Triadic interactions arise naturally in the RSWE as a consequence of the quadratic nonlinearity.
- In the limit of infinite scale separation, only *directly-resonant* interactions contribute to the solution.
- For finite parameter regimes, *near-resonant* interactions occur.
- Near resonant triads of order ε are relevant on $\mathcal{O}(1/\varepsilon)$ timescales.
- Consideration of near-resonant interactions is necessary for optimal averaging in the decimated equations.

4 Numerical Wave Averaging

> Just then, Goldilocks woke up and saw the three bears. She screamed, "Help!" And she jumped up and ran out of the room. Goldilocks ran down the stairs, opened the door, and ran away into the forest. And she never returned to the home of the three bears.
>
> Goldilocks and the Three Bears, Robert Southey

In the previous chapters we have seen how a system of PDEs based on fast-wave averaging is derived in the asymptotic case and that this has interesting ramifications for the discrete components of nonlinear oscillation, called *triads*. We have further seen that for finite scale separation, which is the problem which arises in practical geophysical simulations, an infinitely-long averaging window is not necessary. In fact, taking an infinite averaging window in the finite-ε case provides a suboptimal solution. What we are lacking so far is a way of computing this average and an understanding of its effect on the solution.

Enter the Heterogeneous Multiscale Method (HMM). Based on earlier solution techniques such as *split explicit methods* (Klemp and Wilhelmson, 1978; Tripoli and Cotton, 1982; Wicker and Wilhelmson, 1995), HMM provides a mathematical formalisation of these ideas (E and Engquist, 2003; E, 2003; Engquist and Tsai, 2005). The intention of a numerical method for the averaged equations is to resolve the long-time, macroscale behaviour of the solution. In order to do this it relies on both an incomplete macroscale model and supplementary information from the microscale. HMM provides a framework, rather than a direct method, for multiscale problems and so this requires some deeper understanding in order to design the algorithm for a particular problem. That understanding in our case comes from the triadic interactions and near-resonance.

The name of the framework is meant to make clear the nature of the problems to which the framework is intended to be applied: problems where there are different models at different scales. The main difference between HMM and traditional multiscale

methods is that, at their core, multiscale methods are microscale solvers. Their computational cost is comparable to that of their associated microscale solver as their purpose is to resolve the microscale details. On the other hand, HMM uses information from the microscale to resolve the macroscale, and therefore has the computational cost of a macroscale solver. As noted by E et al. (2007),

> '...multiscale problems are commonly recognised for their complexity, yet the main challenge in multiscale modelling is to recognise their simplicity, and make use of it. This has been a common theme in modern multiscale modelling. The disparity of time scales, for example, has long been a major obstacle in atomistic simulations such as molecular dynamics. But in methods such as HMM, it is used as an asset.'

HMM is related to operator splitting (*cf.* Chapter 2) which has geophysical and meteorological applications (Browning and Kriess, 1994). It is also related to multirate methods (Gear and Wells, 1984; Leimkuhler and Reich, 2001). An important distinction for our purposes is that all of these methods resolve the large eigenvalues which give rise to stiffness over time intervals independent of ε and so have a computational cost of $\mathcal{O}(\varepsilon^{-1})$, while HMM schemes have a complexity of $\mathcal{O}(1)$.

Recall equation (1.36), which we derived in Chapter 1, and which is written

$$\begin{aligned} \frac{\partial \overline{\mathbf{u}}(\mathbf{x},t')}{\partial t'} &= -\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau e^{s\mathcal{L}}\mathcal{N}(e^{-s\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x},t'),e^{-s\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x},t'))\,\mathrm{d}s \\ \overline{\mathbf{u}}(\mathbf{x},t')\big|_{t'=0} &= \mathbf{u}^0(\mathbf{x}). \end{aligned} \tag{4.1}$$

Here, there are two timescales which are separated in a notational sense: the matrix exponential and its inverse are the only locations where the fast time, τ, appears while the averaged solution, $\overline{\mathbf{u}}(\mathbf{x},t')$, is a function of the slow time, t', only. Timestepping this equation requires information from the microscale, i.e. the fast timescale, to be projected to the slow time. This projection is performed by the averaging integral which is a finite analogue of (4.1).

An important point to be aware of here is that when using the HMM the coupling between the fast and slow timescales is *data-based* rather than *solution-based* and so this problem can be handled numerically in a straightforward fashion. That is not to say that an understanding of the solution will not permit further optimisation – it does – but beyond the point of algorithm design the problem is a purely numerical one.

Our approach is then to evaluate the integral on the right-hand side of (4.1) over a finite interval and use this information to evolve the slow solution. Some authors, e.g. Engquist and Tsai (2005), refer to this as 'force estimation'. Due to the finite nature of the problem, we rewrite equation (4.1) as a finite integral, discarding the limit as

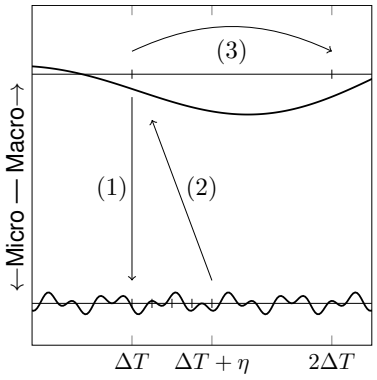

Figure 4.1: The thinking behind the HMM.
(1): Information from the macroscale is used to initialise the microscale model.
(2): The rapidly-oscillating microscale data are integrated over a window of length η and the result is projected back to macroscale.
(3): A large coarse timestep, ΔT, is taken with the macroscale model.

$\tau \to \infty$ of Section 1.4.

$$\frac{\partial \overline{\mathbf{u}}(\mathbf{x}, t')}{\partial t'} = -\frac{1}{\eta} \int_0^\eta \rho\left(\frac{s}{\eta}\right) e^{s\mathcal{L}} \mathcal{N}(e^{-s\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t'), e^{-s\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t'))\, \mathrm{d}s, \quad (4.2)$$

$$:= \overline{\mathcal{N}}(\overline{\mathbf{u}}), \quad (4.3)$$

where η is the finite length of the averaging window. In order to solve this numerically, we replace the finite integral with a finite sum,

$$\frac{\partial \overline{\mathbf{u}}(\mathbf{x}, t')}{\partial t'} \approx -\frac{1}{\overline{M}} \sum_{m=0}^{\overline{M}-1} \rho\left(\frac{s_m}{\eta}\right) e^{s_m\mathcal{L}} \mathcal{N}(e^{-s_m\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t'), e^{-s_m\mathcal{L}}\overline{\mathbf{u}}(\mathbf{x}, t')), \quad (4.4)$$

where $\overline{M}$ is the number of points in time used to discretise the averaging window and $\rho(\cdot)$ is a smooth kernel of integration which permits a shorter approximation for very fast oscillations (i.e. as $\varepsilon \to 0$) which would otherwise render the method impractical. The sum we perform, while containing macroscale information from $\overline{\mathbf{u}}$, is solely over the fast time coordinate, and the matrix exponential is then applied to project the solution back into the fast time coordinate, which is a cheap and stable operation. Algorithm 4.2 provides a pseudocode implementation.

Algorithm 4.2: Evaluating the Time Average

parfor $m = 1, \ldots, \overline{M} - 1$ **do** ▷ Time-Parallel Average
  $s_m = \eta m / \overline{M}$
  $\mathbf{u}_m \leftarrow \rho(s_m / T_0) e^{s_m\mathcal{L}} \mathcal{N}(e^{-s_m\mathcal{L}}\overline{\mathbf{u}}_0, e^{-s_m\mathcal{L}}\overline{\mathbf{u}}_0)$
end parfor
return $\mathtt{Sum}(\mathbf{u}_1, \ldots, \mathbf{u}_{\overline{M}})$

We shall refer to a numerical method for solving this system as a *coarse* solver. The terminology has its roots in the Parareal method (*q.v.* Chapter 5) but applies in this more general sense as well.

Algorithm 4.3: Coarse Solver

$\mathbf{v} \leftarrow e^{(\Delta T/2)\mathcal{D}}\mathbf{u}_0$ ▷ $\Delta T/2$ timestep for linear dissipation.
$\mathbf{v} \leftarrow \overline{\mathcal{N}}(\mathbf{v}, \mathbf{v})$ ▷ ΔT timestep for the averaged nonlinearity.
$\mathbf{v} \leftarrow \overline{\mathcal{N}}\left(\mathbf{u}_0 + \frac{\Delta T}{2}\mathbf{v}\right)$
$\mathbf{v} \leftarrow e^{(\Delta T/2)\mathcal{D}}\mathbf{u}_0$ ▷ $\Delta T/2$ timestep for linear dissipation.
$\mathbf{u}_1 \leftarrow e^{(\Delta T/\varepsilon)\mathcal{L}}\mathbf{v}$ ▷ Transform back to the fast time coordinate.
return $\mathbf{u}_1$

In order to solve the RSWE in practice, we will employ mild hyperviscosity for stability purposes. Considering the definition of the averaged nonlinear operator (4.3) we write our governing HMM-style equation as

$$\frac{\partial \overline{\mathbf{u}}(\mathbf{x}, t')}{\partial t'} = \overline{\mathcal{N}}(\overline{\mathbf{u}}) + \overline{\mathcal{D}}\mathbf{u}, \quad (4.5)$$

where following Haut and Wingate (2014) the averaged dissipation operator is

$$\overline{\mathcal{D}}\mathbf{u} = \frac{1}{\tau} \int_0^\tau \left(e^{s\mathcal{L}} \mathcal{D} e^{-s\mathcal{L}}\right) \mathbf{u}(t')\, \mathrm{d}s. \quad (4.6)$$

The method which shall be referred to in the remainder of this work as a *coarse solver* consists of the application of Strang splitting to equation (4.5). This algorithm is provided in Algorithm 4.3. As we are restricting ourselves to very minor hyperviscosity and studying the effects of the oscillatory nonlinearity on this method, we shall generally neglect dissipation in the mathematical analyses from here on, except where otherwise noted.

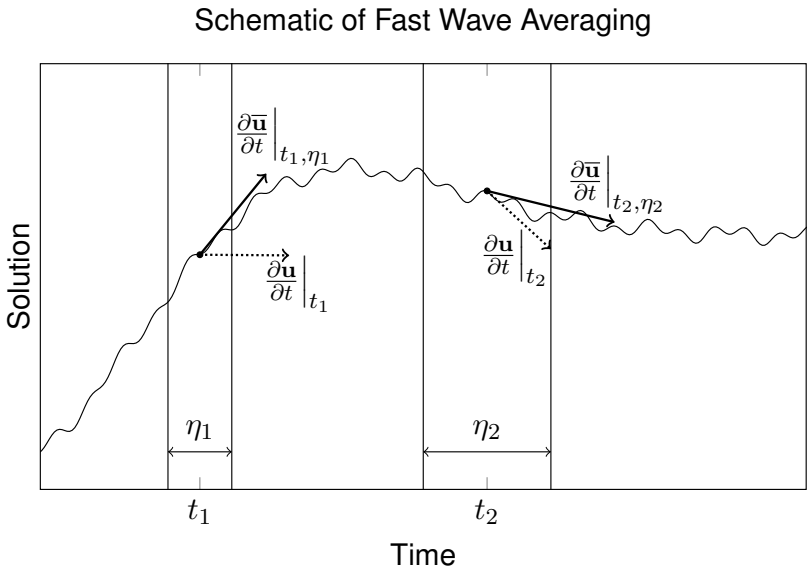

Figure 4.2: Two choices of averaging window. For a slow solution modulated by a fast, a comparison between the tendencies in time of the full solution and the averaged solution is shown. Note how the presence of the rapid oscillations leads to a time derivative which is highly inaccurate in terms of the long-term trend, while an appropriate choice of the averaging window resolves the long-term behaviour comparatively better. When the 'length of the averaging window' is referred to in this work, it is the choices of η, two of which are shown here as η_1 and η_2, which are being referred to.

In practice, the length of the averaging window, η, is a free choice and so may be optimised at run-time. A schematic of the averaging window and its relationship to the fast and slow solutions is shown in Figure 4.2 for an ODE. When solving PDEs or systems of ODEs, i.e. for higher-dimensional problems, the principle does not change: the averaging is still performed only in time.

4.1 A Smooth Kernel of Integration

THERE ARE THREE IMPORANT DIFFERENCES in equation (4.4) when compared to the infinitely-averaged equation (4.1):

- the integral is approximated by a finite sum,
- the previously infinite upper bound of integration is now finite,
- an integrating kernel as been introduced, following Engquist and Tsai (2005).

The solver we are developing must be applicable to a range of problems from where $\varepsilon = \mathcal{O}(1)$ to the rapidly oscillating situation as $\varepsilon \to 0$. In the small-ε asymptotic limit, the length of the averaging

window is necessarily very long (*cf.* Section 1.4) leading to a large computational cost. The purpose of the integrating kernel is to improve the accuracy of the averaging for fast oscillations when using a shorter averaging window (Haut and Wingate, 2014).

Definition 4.1 (Integrating Kernel)**.** Let $\rho(s) : \mathbb{R} \to \mathbb{R}$ be any function such that:

1. $\int_0^1 \rho(s)\,\mathrm{d}s = 1$;
2. $\rho(s) \in C^\infty$;
3. $\operatorname{supp}\rho(s) = [0,1]$.

It is directly implied by 3. that $\rho(0) = \rho(1) = 0$. Furthermore, conditions 2. and 3. must be satisfied for $\dfrac{\mathrm{d}^\xi \rho}{\mathrm{d}s^\xi}$, $\forall\, \xi \in \mathbb{N}$. ▲

The canonical example of such a function would be the *bump function* shown in Figure 4.3 which is defined as

$$\rho(s) = \begin{cases} Ke^{\dfrac{-1}{(1-(2s-1)^2)}} & \text{for } s \in (0,1), \\ 0 & \text{otherwise,} \end{cases} \tag{4.7}$$

where K is a constant which ensures that condition 1 in Definition 4.1 is satisfied. For more detail on the integral of a bump function, the reader is referred to Johnson (2007).

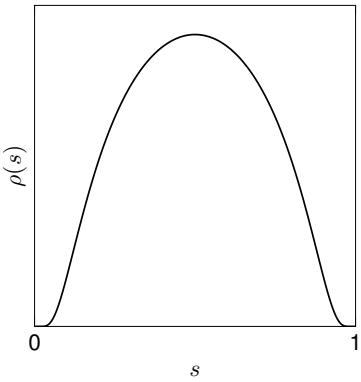

Figure 4.3: A bump function scaled to be supported on $[0,1]$. This is an example of an integrating kernel in line with Definition 4.1

Consider the behaviour of the averaged equations (4.1) in the limit as $\varepsilon \to 0$. Recall from Chapter 3 that in this limit, only direct resonances survive the averaging procedure (*cf.* equation (3.30)). Recalling that $\tau = t/\varepsilon$,

$$\frac{1}{\eta}\int_0^\eta e^{i\frac{\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}}{\varepsilon} s} = \begin{cases} 1 & \text{if } \omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

First consider the case where no integrating kernel is used and the integral of equation (4.1) is replaced with a finite sum. In this case, the integral evaluates to

$$\frac{1}{\eta}\int_0^\eta e^{i\frac{\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}}{\epsilon} s}\,\mathrm{d}s = \frac{\epsilon}{i\omega\eta}\left(e^{i\frac{\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}}{\epsilon}\eta} - 1\right). \tag{4.9}$$

For numerical purposes, we wish to replicate the effect of an infinitely-long averaging window (i.e. the limit as $\eta \to \infty$) through the use of a sufficiently large but finite window. As shown in equation (4.9), such an integral converges linearly to the solution corresponding to $\varepsilon \to 0$ as $\eta \to \infty$.[1]

[1] That is to say, the solution which consists only of direct resonances.

Now consider the integral when the smooth kernel is considered. We are ultimately interested in using this kernel to improve the numerical approximation to our integral for finite η and therefore must use a discrete approximation to the integral. Disregarding discretisation error, we are able to describe the convergence of an oscillatory integral using a smooth kernel as $\eta \to \infty$ through Theorem 4.1.

Theorem 4.1 (Convergence with a Smooth Kernel)**.** *Let $\rho(s)$ be a smooth kernel of integration as in Definition 4.1. Consider a numerical approximation to an oscillatory integral of the type:*

$$I_{\text{osc}} = \frac{1}{\eta}\int_0^\eta \rho(s) e^{i\frac{\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}}{\varepsilon}s}\,\mathrm{d}s.$$

A numerical approximation to this integral where $\rho(s)$ is approximated with $m+1$ collocation points converges at $\mathcal{O}\left(\frac{1}{\eta^m}\right)$. ♦

Proof. When $\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2} = 0$, I_{osc} is trivially equal to one, which is its analytical solution, and so this case does not need to be considered with respect to the convergence of the numerical integral. Consider the case where $\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2} \neq 0$ and $\varepsilon \to 0$, where the analytical solution for the integral is zero (*cf.* Section 3.3). For convenience, we will rescale the limits of the integral to be

$$\frac{1}{\eta}\int_0^\eta e^{i\frac{\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}}{\epsilon}s}\rho\left(\frac{s}{\eta}\right)\,\mathrm{d}s = \int_0^1 e^{i\frac{\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}}{\epsilon}\eta s}\rho(s)\,\mathrm{d}s. \tag{4.10}$$

Let us then rewrite the integral in a form more amenable to integration by parts, using Ω to denote the triad sum, $\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}$, and denoting $\frac{\mathrm{d}\rho(s)}{\mathrm{d}s}$ as $\rho'(s)$. Then upon integrating the rescaled form in (4.10) once by parts we find

$$\frac{\epsilon}{i\Omega}\int_0^1 \frac{\mathrm{d}}{\mathrm{d}s}(e^{i\frac{\Omega\eta}{\epsilon}s})\rho(s)\,\mathrm{d}s = \frac{\epsilon}{i\Omega\eta}\rho(s)e^{i\frac{\Omega\eta}{\epsilon}s}\Big|_0^1 - \frac{\epsilon}{i\frac{\Omega\eta}{\epsilon}}\int_0^1 e^{i\Omega\eta s}\rho'(s)\,\mathrm{d}s. \tag{4.11}$$

By Definition 4.1, $\rho(0) = \rho(1) = 0$, along with all its derivatives. This causes the first term on the right-hand side to vanish. Recalling that the approximation to $\rho(s)$ is performed with $m+1$ collocation points, we may repeat the same procedure m times, arriving after m integrations-by-parts with,

$$\int_0^1 e^{i\frac{\Omega\eta}{\epsilon}s}\rho(s)\,\mathrm{d}s = \left[\frac{\epsilon}{i\Omega\eta}\right]^m \int_0^1 e^{i\frac{\Omega}{\epsilon}\eta s}\rho^{(m)}(s)\,\mathrm{d}s \;\forall m \in \mathbb{Z}^+. \tag{4.12}$$

□

In the limit as $\varepsilon \to 0$, the smooth kernel prevents all non-resonant triads from surviving the averaging procedure when approximating

the averaging integral with a shorter window than would otherwise be necessary, reducing the computational cost. We see in equation (4.12) that indeed convergence is more rapid. As shown in Theorem 4.1, for some fixed m it goes to zero like $\frac{1}{\eta^m}$ rather than linearly.

We were able to use a known result here for the limit as $\varepsilon \to 0$ to show the improvement in convergence given by the kernel. This does not mean that when working with finite ε the purpose of the integrating kernel is to remove all non-resonant triads. It is to reduce the length of the averaging window which is required to approximate the integral in (4.1). For finite ε, the window length will need to be chosen to provide optimal convergence. The role of the integrating kernel is not to eliminate this choice, but to reduce the optimal length when it is chosen.

4.2 *Finite Averaging Window*

The existence and relevance of near-resonant triadic interactions outside of the asymptotic limit was discussed in Chapter 3. The relevance of these interactions for finite problems motivates our finite averaging. As we have discussed, we may consider the time derivative of the solution to be comprised of the sum of triadic interactions. As we will show here, the degree of nearness of these interactions affects both the numerical stiffness they induce and their effect on the accuracy of the macroscale model of the flow.

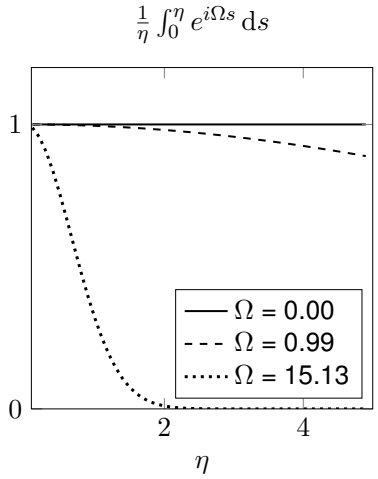

Figure 4.4: The effect of varying the averaging window length on individual triads. $\Omega = \omega^{\alpha}_{\mathbf{k}} - \omega^{\alpha_1}_{\mathbf{k}_1} - \omega^{\alpha_2}_{\mathbf{k}_2}$. The directly resonant triad, $\Omega = 0$, is unaffected by the averaging procedure. The near- and far-resonances are attenuated proportionally to both their magnitude and the averaging window length, η. The particular values of Ω arise in an actual triad of the 1-D RSWE such that $(\mathbf{k}, \mathbf{k}_1, \mathbf{k}_2) = (8, 1, 7)$ with modes (i.e. values of α) chosen to achieve direct, near- and far-resonance from the same triad.

Figure 4.4 shows the effect of numerical averaging on three particular triads. As would be expected, the direct resonances are unaffected by the averaging procedure, as $e^{i\Omega s} = 1$ when $\Omega = 0$, leading to a trivial and non-oscillatory integral.

All other triads are attenuated in proportion both to their magnitude, $|\omega^{\alpha}_{\mathbf{k}} - \omega^{\alpha_1}_{\mathbf{k}_1} - \omega^{\alpha_2}_{\mathbf{k}_2}|$, and to the length of the averaging window. In this case we see that an averaging window of $\eta = 2$ has almost completely annihilated the far-resonant triad ($\Omega = 15.13$) while it has only begun to affect the near-resonant triad ($\Omega = 0.99$). Choosing the length of this window then provides a method to smoothly reduce stiffness by targeting the fastest components of the flow more strongly than the slower ones. We also expect that the fastest (i.e. farthest-resonant) triads contribute the least to the long time behaviour of the solution (Clark di Leoni and Mininni, 2016).

Selectively but completely retaining or rejecting triads was shown in Chapter 3 to improve the performance of a timestepping method over long timesteps outside of the normal explicit timestep limit. We see here that filtering triads through an averaging win-

dow accomplishes the same end. Using a finite averaging window not only has the advantage of being more numerically tractable in that it does not require expensive convolutions in wavespace but also gives improved performance.

Just as with selecting the degree of resonance to retain, we must choose the length of the averaging window carefully to yield optimal convergence. The optimal averaging window is that which retains sufficient triadic information to be accurate, while rejecting the fastest components and therefore reducing numerical stiffness. Figures 4.5 and 4.6 show the HMM averaging procedure given by equation (4.4). Numerically, $\varepsilon = 0.01$ is close enough to approximate the behaviour of the asymptotic limit and so it is not surprising that this case, shown in Figure 4.5, is well-modelled.

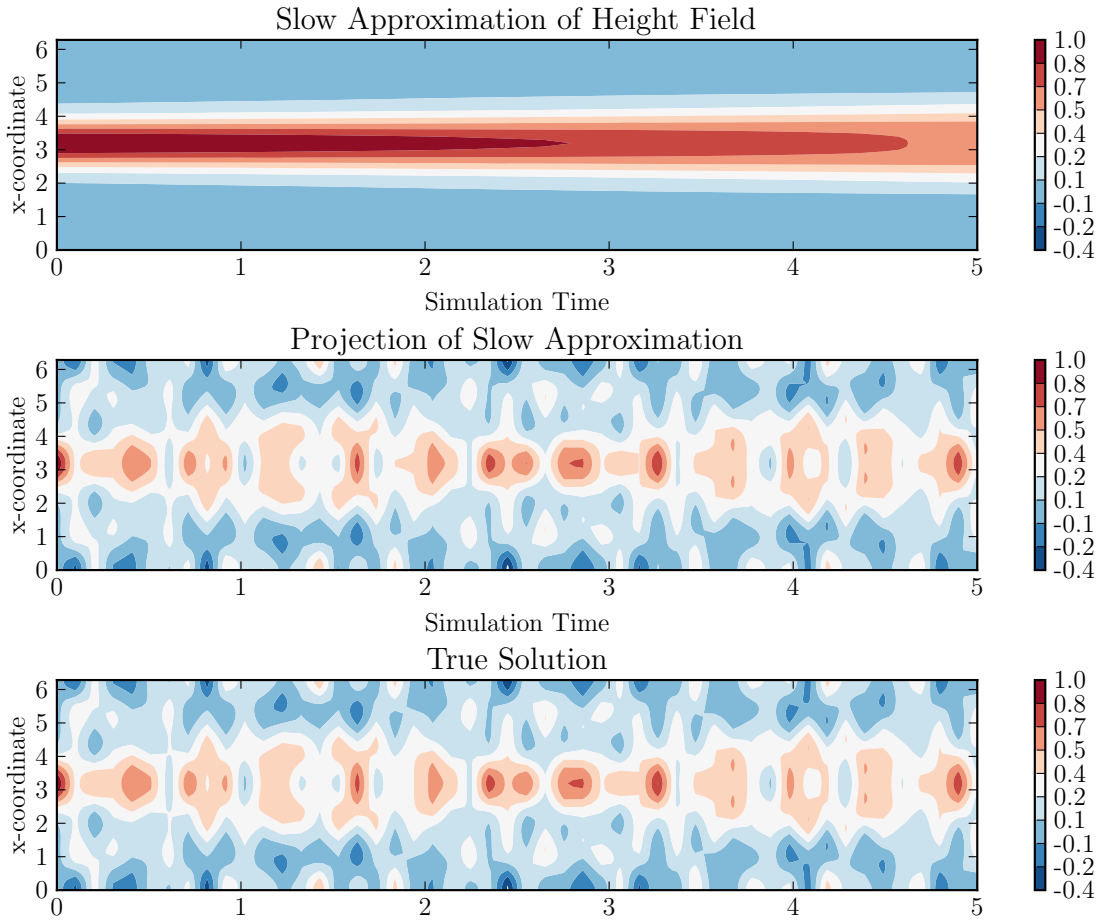

Figure 4.5: Comparison of slow, projected, and full solutions for the stiff case, $\varepsilon = 0.01$. Here the averaging is optimal, with $\eta = 10\Delta T$. This is a classically stiff problem for which averaging methods are necessary to enable such technologies as we will discuss later.

The regime with $\varepsilon = 1$, on the other hand, is well outside the range where the asymptotic description as in Section 1.4 should hold. Using an asymptotic result and taking a very long averaging window here would indeed yield a poor-quality coarse approximation. However, results of comparable quality where obtained here as well through an appropriate choice of the averaging window length.

In both figures, spatio-temporal oscillations for the 1-D RSWE with an initially stationary height field are depicted, with the only difference being the degree of scale separation, ε. The bottom-most plot shows the reference solution, computed by a fine timestepping method with no averaging. The topmost plot shows the averaged variable, $\overline{\mathbf{u}}$, over which the timestepping is performed. It is particularly clear in the stiffer of the two shown in Figure 4.5 and to a lesser extent in Figure 4.6 that timestepping over this solution will suffer less from the timestep restriction imposed by the oscillatory stiffness than the full solution, $\mathbf{u}$, does.

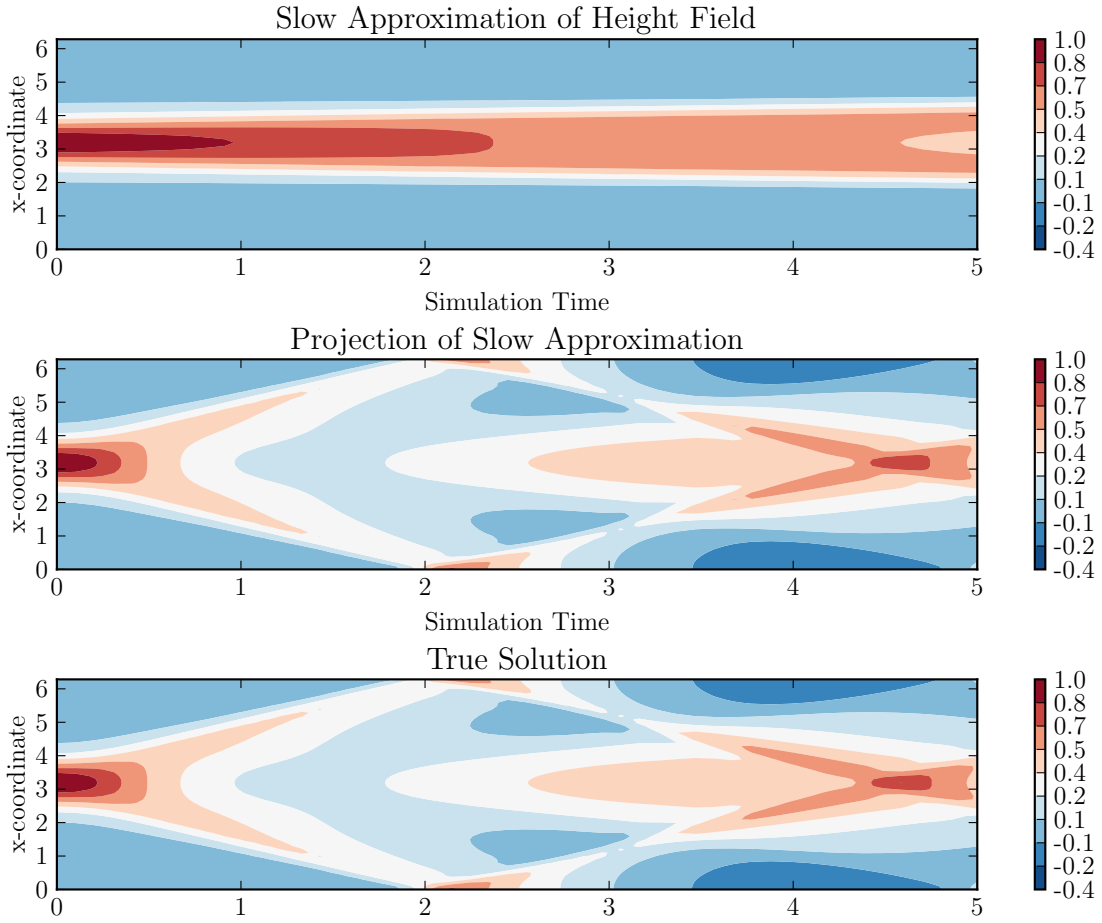

Figure 4.6: Comparison of slow, projected, and full solutions for the case with no scale separation, $\varepsilon = 1$. Here the averaging is optimal, with $\eta = \Delta T$. This is not a classically stiff problem, but it is novel and important in the context of practical problems with time-varying scale separation that a fast-wave averaging-based solution is accurate outside of the limit as $\varepsilon \to 0$.

The central plot in both of these figures is the projection of the slow solution back into the space of the full solution, i.e. $\mathbf{u} = e^{-\tau\mathcal{L}}\overline{\mathbf{u}}$ which is an operation which does not have an associated timestep restriction. The fidelity of this solution to the full solution, subject to optimal averaging, is what makes the HMM-style method viable for approximating the solution to problems of the type studied here.

With the practicalities of the algorithm in place, we are in a position to discuss convergence for the case when ε is finite. We shall neglect numerical error arising from truncation, spatial discreti-

sation, etc. and consider the two sources of error over which the averaging procedure has an effect: the averaging error and the timestepping error.

4.3 *Error Analysis*

ERROR BOUNDS ON THIS SOLVER have been proven by Haut and Wingate (2014) in the context of Parareal convergence in the asymptotic limit as $\varepsilon \to 0$. In doing so, they relied on the direct resonances which we have discussed. In extending their result to finite ε, we will necessarily consider near-resonances as well. Recalling Chapter 1 and in particular Figure 1.7, the choice of the averaging window width, η, has a effect on the convergence of the method (i.e. a four-fold change in η corresponds to an order of magnitude change in accuracy). While the choice of η is well-understood for the limit of small ε Haut and Wingate (2014), we show here that η may be chosen to provide convergence for ε up to $\mathcal{O}(1)$ for an appropriate coarse timestep. In order to use existing results for the averaging error (Sanders et al., 2007), we need to first reduce the governing equation (1.4) to a standard form for ODEs. Following Section 3.2, we write

$$\mathbf{v}_t(t) = e^{t\mathcal{L}/\varepsilon} N\left(e^{-t\mathcal{L}/\varepsilon}\mathbf{v}(t), e^{-t\mathcal{L}/\varepsilon}\mathbf{v}(t)\right), \quad t \in [0, \Delta T], \tag{4.13}$$

where the subscript t denotes the time derivative. In doing so, we make clear that we are interested in the solution over a ΔT timescale.[2] Let $\tau = t/(\varepsilon\Delta T)$, and so $\tilde{\mathbf{v}}(\tau)$, defined on the interval $[0, 1/\varepsilon]$,

$$\tilde{\mathbf{v}}(\tau) = \mathbf{v}(t). \tag{4.14}$$

[2] Recall from Section 1.1 that numerical stiffness and *finite* time intervals go hand in hand.

Then differentiation gives,

$$\partial_t \mathbf{v}(t) = \partial_t \tilde{\mathbf{v}}(t/(\varepsilon\Delta T)) = \frac{1}{\varepsilon\Delta T}\partial_\tau \tilde{\mathbf{v}}(t/(\varepsilon\Delta T)) = \frac{1}{\varepsilon\Delta T}\partial_\tau \tilde{\mathbf{v}}(\tau). \tag{4.15}$$

Upon this substitution over the discrete time interval in (4.13), we arrive at the desired form which permits us to use the framework given in Sanders et al. (2007) where they have derived bounds for averaging methods. This framework is a general form of a nonlinear system subject to wave averaging. The aim of this is to modify and reapply their result for the error bound due to averaging. We then write the governing equation in the form:

$$\partial_\tau \tilde{\mathbf{v}}(\tau) = \varepsilon\Delta T e^{\tau\Delta T\mathcal{L}} N\left(e^{-\tau\Delta T\mathcal{L}}\tilde{\mathbf{v}}(\tau), e^{-\tau\Delta T\mathcal{L}}\tilde{\mathbf{v}}(\tau)\right). \tag{4.16}$$

While our interest is in solving PDEs describing physical systems, in practice we employ a Fourier spectral method which has the effect of treating the PDE as a finite-dimensional system of ODEs. This gives us access to the machinery of the numerical analysis of

ODEs and averaging methods, following Sanders et al. (2007). Let $\mathbf{x}$ solve the governing equations for the full (i.e. unaveraged) system when they are written as a finite system of ODEs, i.e. in the form shown in (4.16). Then we may write:

$$\mathbf{x}_t = \varepsilon \mathbf{f}(\mathbf{x}, t). \tag{4.17}$$

Here, $\mathbf{x}$ denotes the ODE solution, and not the spatial variable as it has before. This is in the interest of consistency with the ODE literature. Similarly, we consider the coarse solver (4.16) written as a system of ODEs. Let $\mathbf{y}$ solve this averaged form of equation (4.16), i.e.

$$\mathbf{y}_t = \varepsilon \bar{\mathbf{f}}(\mathbf{y}, t), \tag{4.18}$$

where the averaging follows directly from the averaged equation (4.1) and is written

$$\bar{\mathbf{f}}(\mathbf{y}, t) = \frac{1}{\eta} \int_0^{\eta} \rho\left(\frac{s}{\eta}\right) \mathbf{f}(\mathbf{y}, t+s)\, ds, \tag{4.19}$$

where η denotes the finite length of the averaging window.

4.4 *Averaging Error*

Let us first consider the *averaging error*, which is the error committed by approximating the governing equations with an averaged analogue. This section does not consider any numerical effects.

Definition 4.2 (KBM-vector field)**.** Consider the vector field $\mathbf{f}(\mathbf{x}, t)$, $\mathbf{f} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ which is Lipschitz continuous in $\mathbf{x}$ on $D \subset \mathbb{R}^n$ for positive t, and where $\mathbf{f}$ is continuous in t and $\mathbf{x}$ on $\mathbb{R}^+ \times D$. If the average

$$\bar{\mathbf{f}} = \lim_{\eta \to \infty} \frac{1}{\eta} \int_0^{\eta} \mathbf{f}(\mathbf{x}, s)\, \mathrm{d}s \tag{4.20}$$

exists and has a uniform limit on compact sets $K \subset D$, then $\mathbf{f}$ is a *KBM-vector field*[3] (Sanders et al., 2007). It is assumed that any parameters in $\mathbf{f}(\mathbf{x}, t)$, as well as the initial conditions, are ε-independent. ▲

[3] KBM stands for Krylov, Bogoliubov, and Mitropolsky.

In the remainder of this work, we shall assume that $\mathbf{f}$ is a KBM-vector field.

Lemma 4.1. *Let $\phi(t)$ be a Lipschitz-continuous function with Lipschitz constant β. Further define $\phi_\eta(t)$ where the subscript η denotes time averaged $\phi(t)$ with integrating kernel $\rho(s)$ and averaging window length η. Then,*

$$\left|\phi(t) - \phi_\eta(t)\right| \leq C_0 \beta \eta, \tag{4.21}$$

where

$$C_0 = \int_0^1 \rho(s)\, s\, \mathrm{d}s. \tag{4.22}$$

♦

Proof. Using that

$$\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\, \mathrm{d}s = \int_0^1 \rho(s)\, \mathrm{d}s = 1, \tag{4.23}$$

we have that

$$\begin{aligned} \left|\phi(t) - \phi_\eta(t)\right| &= \left|\phi(t) - \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\phi(s+t)\, \mathrm{d}s\right| \\ &= \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\left|\phi(t) - \phi(s+t)\right|\, \mathrm{d}s \\ &\leq \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right) s\beta\, \mathrm{d}s \end{aligned}$$

by Lipschitz continuity. Then,

$$\begin{aligned} \left|\phi(t) - \phi_\eta(t)\right| &\leq \frac{1}{\eta}\beta\int_0^\eta \rho\left(\frac{s}{\eta}\right) s\, \mathrm{d}s \\ &= \eta\beta\int_0^1 \rho(s)\, s\, \mathrm{d}s. \end{aligned}$$

□

Lemma 4.2. *Consider*

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}\tau}(t) = \Delta T\varepsilon\mathbf{f}(\Delta T t, \mathbf{v}(t)), \quad 0 \leq t \leq \varepsilon^{-1}, \tag{4.24}$$

with $\mathbf{f}$ *continuous in each argument and* $t < \mathcal{O}(1)$*. Also assume that*

$$\|\mathbf{f}(\Delta T t, \mathbf{u}) - \mathbf{f}(\Delta T t, \mathbf{w})\| \leq \beta\|\mathbf{u} - \mathbf{w}\|, \tag{4.25}$$

and

$$M = \sup_{x\in D}\sup_{0\leq t\leq\varepsilon^{-1}} \|\mathbf{f}(\Delta T t, \mathbf{w})\| < \infty. \tag{4.26}$$

Then defining

$$\phi(t) = \int_0^t \mathbf{f}(\Delta T\tau, \mathbf{v}(\tau))\, \mathrm{d}\tau, \tag{4.27}$$

we have that

$$\left|\phi_\eta(t) - \int_0^t \mathbf{f}_\eta(\Delta T\tau, \mathbf{v}(\tau))\, \mathrm{d}\tau\right| \leq C_0(1 + \beta\Delta T\varepsilon)M\eta. \tag{4.28}$$

♦

Proof. We calculate that

$$\begin{aligned}\phi_\eta(t) &= \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\phi(s+t)\,\mathrm{d}s \\ &= \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\left(\int_0^{t+s}\mathbf{f}(\Delta T\tau,\mathbf{v}(\tau))\,\mathrm{d}\tau\right)\mathrm{d}s \\ &= \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\left(\int_s^{t+s}\mathbf{f}(\Delta T\tau,\mathbf{v}(\tau))\,\mathrm{d}\tau\right)\mathrm{d}s + R_1\end{aligned}$$

where we have changed the lower bound of integration, introducing the residual R_1 in the process. Then,

$$\begin{aligned}\phi_\eta(t) &= \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\left(\int_0^{t}\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau+s))\,\mathrm{d}\tau\right)\mathrm{d}s + R_1 \\ &= \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\left(\int_0^{t}\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau))\,\mathrm{d}\tau\right)\mathrm{d}s + R_1 + R_2\end{aligned}$$

where the dependence of the integral on the variation of $\mathbf{v}$ with respect to s is captured by the second residual, R_2.

$$\begin{aligned}\phi_\eta(t) &= \int_0^t\left(\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau))\,\mathrm{d}s\right)\mathrm{d}\tau + R_1 + R_2 \\ &= \int_0^t\int_0^t \mathbf{f}_\eta(\Delta T\tau,\mathbf{v}(\tau))\,\mathrm{d}\tau + R_1 + R_2,\end{aligned}$$

The norm of the first residual is

$$\begin{aligned}\|R_1\| &\le \left\|\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\left(\int_0^s\mathbf{f}(\Delta T\tau,\mathbf{v}(\tau))\,\mathrm{d}\tau\right)\mathrm{d}s\right\| \\ &\le \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^s\|\mathbf{f}(\Delta T\tau,\mathbf{v}(\tau))\|\,\mathrm{d}\tau\,\mathrm{d}s \\ &\le \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^s M\,\mathrm{d}\tau\,\mathrm{d}s,\end{aligned}$$

by the definition of M. Then we may write:

$$\begin{aligned}\|R_1\| &= M\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)s\,\mathrm{d}s \\ &= M\eta\int_0^1\rho(s)\,s\,\mathrm{d}s \\ &= C_0M\eta,\end{aligned}$$

with C_0 defined as in Lemma 4.1. The norm of the second residual may be found to be

$$\begin{aligned}\|R_2\| &= \left\|\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t\left(\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau+s))-\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau))\right)\mathrm{d}\tau\,\mathrm{d}s\right\| \\ &\le \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t\|\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau+s))-\mathbf{f}(\Delta T(\tau+s),\mathbf{v}(\tau))\|\,\mathrm{d}\tau\,\mathrm{d}s \\ &\le \frac{1}{\eta}\beta\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t\|\mathbf{v}(\tau+s)-\mathbf{v}(\tau)\|\,\mathrm{d}\tau\,\mathrm{d}s\end{aligned}$$

because of Lipschitz continuity. Continuing, we find that

$$\begin{aligned}
\|R_2\| &\leq \frac{1}{\eta}\beta\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t \left\|\int_\tau^{s+\tau}\frac{d\mathbf{v}}{\mathrm{d}\sigma}(\sigma)\,\mathrm{d}\sigma\right\|\mathrm{d}\tau\,\mathrm{d}s \\
&= \frac{1}{\eta}\beta\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t \left\|\int_\tau^{s+\tau}\Delta T\varepsilon\mathbf{f}\left(\Delta T\sigma,\mathbf{v}(\sigma)\right)\mathrm{d}\sigma\right\|\mathrm{d}\tau\,\mathrm{d}s \\
&\leq \frac{1}{\eta}\Delta T\varepsilon\beta\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t\int_\tau^{s+\tau}\left\|\mathbf{f}\left(\Delta T\sigma,\mathbf{v}(\sigma)\right)\right\|\mathrm{d}\sigma\,\mathrm{d}\tau\,\mathrm{d}s,
\end{aligned}$$

from the definition of M. Then

$$\begin{aligned}
\|R_2\| &\leq \frac{1}{\eta}\Delta T\varepsilon\beta M\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t\int_\tau^{s+\tau}\mathrm{d}\sigma\,\mathrm{d}\tau\,\mathrm{d}s \\
&= \frac{1}{\eta}\Delta T\varepsilon\beta M\int_0^\eta \rho\left(\frac{s}{\eta}\right)\int_0^t s\,\mathrm{d}\tau\,\mathrm{d}s \\
&= \frac{1}{\eta}\Delta T\varepsilon\beta Mt\int_0^\eta \rho\left(\frac{s}{\eta}\right)s\,\mathrm{d}s \\
&= C_0\eta\Delta T\beta M\varepsilon t \\
&\leq C_0\Delta T\eta\beta M\varepsilon.
\end{aligned}$$

In the last inequality, we have applied the assumption that $t = \mathcal{O}(1)$ and used the definition of C_0 from Lemma 4.1. □

With Lemmas 4.1 and 4.2 in hand, we are able to proceed to prove the bounds on the averaged equation.

Theorem 4.2 (Averaging Error)**.** *Considering the initial value problems in* $\mathbf{x}$ *and* $\mathbf{y}$ *as stated above where* $\mathbf{f}$ *is* $\mathbb{R}^n \times \mathbb{R}$ *Lipschitz continuous with constant* β *in* $\mathbf{x}$ *on* $D \subset \mathbb{R}^n$ *and* t *on an* $\mathcal{O}(1)$ *timescale, i.e. for all* $\mathbf{x}_1, \mathbf{x}_2 \in D$, β *is such that:*

$$\left\|\mathbf{f}\left(\mathbf{x}_1,t\right)-\bar{\mathbf{f}}\left(\mathbf{x}_2,t\right)\right\| \leq \beta\left\|\mathbf{x}_1-\mathbf{x}_2\right\|. \tag{4.29}$$

Let:

$$M = \sup_{\mathbf{x}\in D}\sup_{0\leq t\leq L}\left\|\mathbf{f}\left(\mathbf{x},t\right)\right\|. \tag{4.30}$$

Then we can bound the difference between the exact solution $\mathbf{x}$ *and the averaged solution* $\mathbf{y}$ *as:*

$$\|\mathbf{x}-\mathbf{y}\| \leq M\left(1+\frac{1}{2}\beta\varepsilon\right)\varepsilon\Delta T\eta, \tag{4.31}$$

♦

Proof. Note that

$$\mathbf{x}(t) = \mathbf{x}(0) + \Delta T\varepsilon\int_0^t \mathbf{f}\left(\Delta T\tau,\mathbf{x}(\tau)\right)\mathrm{d}\tau. \tag{4.32}$$

Let E_0 be the difference between the time integrals of the full and averaged functions, i.e.

$$E_0 = \phi_\eta(t) - \int_0^t \mathbf{f}_\eta\left(\Delta T\tau,\mathbf{x}(\tau)\right)\mathrm{d}\tau. \tag{4.33}$$

Then we may write

$$\int_0^t \mathbf{f}\left(\Delta T\tau, \mathbf{x}\left(\tau\right)\right) d\tau = \int_0^t \mathbf{f}_\eta\left(\Delta T\tau, \mathbf{x}\left(\tau\right)\right) \, \mathrm{d}\tau + E_0, \tag{4.34}$$

where by Lemma 4.2,

$$\left\|E_0\right\| \leq C_0\left(1 + \beta\varepsilon\Delta T\right) M\eta. \tag{4.35}$$

Therefore,

$$\mathbf{x}\left(t\right) = \mathbf{x}\left(0\right) + \Delta T\varepsilon \int_0^t \mathbf{f}_\eta\left(\Delta T\tau, \mathbf{x}\left(\tau\right)\right) \, \mathrm{d}\tau + E_1, \tag{4.36}$$

where

$$E_1 = \Delta T\varepsilon E_0 \tag{4.37}$$

and so

$$\left\|E_1\right\| = \left\|\Delta T\varepsilon E_0\right\| \leq C_0\left(1 + \beta\varepsilon\Delta T\right) M\eta\Delta T\varepsilon. \tag{4.38}$$

Also, since

$$\mathbf{y}\left(t\right) = \mathbf{x}\left(0\right) + \Delta T\varepsilon \int_0^t \mathbf{f}_\eta\left(\Delta T\tau, \mathbf{y}\left(t\right)\right) \, \mathrm{d}\tau, \tag{4.39}$$

we have that

$$\begin{aligned}
\left\|\mathbf{x}\left(t\right) - \mathbf{y}\left(t\right)\right\| &\leq \Delta T\varepsilon \int_0^t \left\|\mathbf{f}_\eta\left(\Delta T\tau, \mathbf{x}\left(\tau\right)\right) - \mathbf{f}_\eta\left(\Delta T\tau, \mathbf{y}\left(t\right)\right)\right\| \, \mathrm{d}\tau + \\
&\quad C_0\left(1 + \beta\varepsilon\Delta T\right) M\eta\Delta T\varepsilon \\
&\leq \Delta T\varepsilon\beta\varepsilon \int_0^t \left\|\mathbf{x}\left(\tau\right) - \mathbf{y}\left(t\right)\right\| \, \mathrm{d}\tau + \\
&\quad C_0\left(1 + \beta\varepsilon\Delta T\right) M\eta\Delta T\varepsilon.
\end{aligned}$$

Finally, by Grönwall's inequality,[4]

$$\left\|\mathbf{x}\left(t\right) - \mathbf{y}\left(t\right)\right\| \leq C_0\left(1 + \beta\varepsilon\Delta T\right) M\eta\Delta T\varepsilon e^{\Delta T\varepsilon\beta t}. \tag{4.40}$$

□

[4] Grönwall's inequality is commonly used to estimate the growth of functions that satisfy an integral inequality (Grossmann et al., 2007).

Theorem 4.2 follows from a modification of results given by Sanders et al. (2007) in order to include the kernel of integration.[5] We have here bounded the error over an $\mathcal{O}(1)$ time interval instead of $\mathcal{O}(1/\varepsilon)$ so that the rate of convergence at different degrees of scale separation may be more easily compared, as in practice we are interested in simulations over fixed timescales. Taking the unmodified lemma provides a slightly different result as it gives the averaging error over a simulation time which scales with ε. Due to the numerical nature of the proof here, this is not the appropriate timescale.

[5] The necessary modifications to Lemmas 4.1 and 4.2 and Theorem 4.2 are not the original work of the author, but are primarily the work of Terry Haut extending the work of Sanders et al. (2007).

Theorem 4.2 places a bound on the error committed by averaging over the fast waves, independent of the numerical methods used for spatial or temporal discretisation. It is important to see that it is proportional to both the averaging window length, η, and the

coarse timestep, ΔT. Such a result is intuitively understandable if we consider that as η increases, more averaging is being applied and so there is more of a difference between the true and averaged equations. Conversely, as $\eta \to 0$, no averaging is being performed and so $\bar{\mathbf{f}} \to \mathbf{f}$, i.e. the difference between the solutions becomes trivially zero. With this source of error understood, we now move on to considering the error arising from the numerical approximation of equation (4.18).

4.5 *Timestepping Error*

THE OTHER PRIMARY SOURCE OF ERROR which is controllable by the averaging window is that which arises from the timestepping. Understanding such a source of error is a central theme in numerical analysis (see for example Trefethen (1996)), and we shall apply many concepts familiar to the numericist here. There is a particular novelty in this section in the consideration of the timestepping error in terms of the near-resonant triads and a rigorous understanding of the effect averaging has on them. We must first define a suitable space on which the differential equation is defined.

Recalling the definition of a KBM-vector field (Definition 4.2), we will choose a Lipschitz-continuous subspace, D, where $\mathbf{y}$ is well-defined and further assume that:

$$\left\| \frac{\partial \mathbf{f}(\mathbf{y}, t)}{\partial \mathbf{y}} \right\| \leq M_1, \quad \mathbf{y}(t) \in D \subset \mathbb{R}^n, \tag{4.41}$$

where M_1 is a finite constant. We further assume that such a bound exists for higher derivatives of $\mathbf{f}$, such that

$$\max_j \left\| \frac{\partial^j \mathbf{f}}{\partial y_k^j} \right\| \leq M, \quad 1 \leq k \leq n, \quad 0 \leq j \leq p. \tag{4.42}$$

where M is defined as in Theorem 4.2. It is helpful to make one more definition in order to consider the triad interactions in an appropriate form for the upcoming proof. To the best of the author's knowledge, this is a novel technique in numerical methods.

Definition 4.3 (Ordered Triads)**.** In a similar fashion to Definition 3.3, let the set $\{\lambda_n : n \in \mathbb{N}\}$ be the ordered set of all triadic interactions, $\lambda_n = \omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}\Big|_n$ such that

$$|\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}|\Big|_{n+1} \geq |\omega_{\mathbf{k}}^{\alpha} - \omega_{\mathbf{k}_1}^{\alpha_1} - \omega_{\mathbf{k}_2}^{\alpha_2}|\Big|_n \tag{4.43}$$

▲

Theorem 4.3 (Timestepping Error)**.** *Denote the numerical approximation to the averaged solution* $\mathbf{y}(t)$ *with timestep* ΔT *and order* 2 *as* $\mathbf{y}_{\Delta T}(t)$*. Assume that* $\mathbf{y}_t(t) = \varepsilon\bar{\mathbf{f}}(\mathbf{y}, t)$ *and that* $\bar{\mathbf{f}} \in D \subset \mathbb{R}^n$ *exhibits quadratic nonlinearity. Assume that integration is performed with respect to a smooth kernel,* $\rho(\cdot)$*, and let* λ_n *denote the n-th near resonant triad (Definition 4.3). Then the local time-stepping error of a second order time-stepping scheme applied to equation (4.18) satisfies:*

$$\|\mathbf{y}(t) - \mathbf{y}_{\Delta T}(t)\| \leq CM\varepsilon\Delta T^3 \max_{n\in\mathbb{N}} \left(\lambda_n^2 \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right) e^{i\lambda_n s} ds\right), \quad (4.44)$$

for some constant, $C \in \mathbb{R} < \infty$ *and where* M *is the bound over the nonlinear operator as given in 4.2.* ♦

Proof. The timestepping error is bounded by

$$\|\mathbf{y}(t) - \mathbf{y}_{\Delta T}(t)\| \leq C_t (\Delta T)^{p+1} \max_t \left\| \frac{\mathrm{d}^{p+1}\mathbf{y}}{\mathrm{d}t^{p+1}}(t) \right\|_2 \quad (4.45)$$

where p is the order of convergence of the method and C_t is a constant (Kincaid and Cheney, 1991). In order to proceed with a constant C_t we assume that there exists some η_0 such that

$$C_t < C_0 + \frac{C_1}{\eta} \quad \forall \eta > \eta_0 \quad (4.46)$$

and that $\eta > \eta_0$. We first decompose $\mathbf{f}$ in terms of its basis of eigenvectors as discussed in Section 3.2. As with equation (3.27) we may write the solution as a sum of ODEs, each for a specific and ordered resonant nearness, λ_n (see Definition 4.3). Then for the j-th component of $\mathbf{y}$, we write

$$\frac{\mathrm{d}y_j}{\mathrm{d}t} = \varepsilon\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right) \sum_n \Delta T e^{i\Delta T\lambda_n(t+s)} \mathbf{N}_{n,j}(\mathbf{y})\, \mathrm{d}s, \quad (4.47)$$

where the nearness of the resonances in any particular ODE is exposed through the eigenvalue sum, λ_n, in the exponent and where the subscript $,j$ denotes the j-th component and not a derivative, as it would with Einstein's notation. Note that $\mathbf{N}_{n,j}$ is a fully nonlinear term and so the system of ODEs has not been linearised in any way. We then seek the third time derivative, which is found to be

$$\begin{aligned}\frac{\mathrm{d}^3 y_j(t)}{\mathrm{d}t^3} = \varepsilon\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right) \Bigg(& i^2\Delta T^3 \sum_n \lambda_n^2 e^{i\Delta T\lambda_n(t+s)} \mathbf{N}_{n,j} + \\ & 2i\Delta T^2 \sum_n \sum_k \lambda_n e^{i\Delta T\lambda_n(t+s)} \frac{\partial \mathbf{N}_{n,j}(\mathbf{y})}{\partial y_k} \frac{\mathrm{d}y_k(t)}{\mathrm{d}t} + \\ & \Delta T \sum_n \sum_{k,l} e^{i\Delta T\lambda_n(t+s)} \frac{\partial^2 \mathbf{N}_{n,j}(\mathbf{y})}{\partial y_k \partial y_l} \frac{\mathrm{d}y_k(t)}{\mathrm{d}t} \frac{\mathrm{d}y_l(t)}{\mathrm{d}t} + \\ & \Delta T \sum_n \sum_k e^{i\Delta T\lambda_n(t+s)} \frac{\partial \mathbf{N}_{n,j}(\mathbf{y})}{\partial y_k} \frac{\mathrm{d}^2 y_k(t)}{\mathrm{d}t^2} \Bigg) \mathrm{d}s. \end{aligned} \quad (4.48)$$

This is then the right-hand side which is integrated with respect to the smooth kernel. The magnitude of the near-resonant triad,

λ_n, now presents itself as a multiplier on the complex exponential. Recalling the definition of stiffness in Section 1.1, it is then clear that it is this value, which is zero for direct resonances but becomes large in general, which is the source of numerical stiffness. For notational simplicity, we introduce the function $P(\eta)$[6], which is defined to be

$$P(n,\eta) = \frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right) e^{i\lambda_n \Delta T s}\,\mathrm{d}s, \tag{4.49}$$

then

$$\begin{aligned}\frac{\mathrm{d}^3 y_j(t)}{\mathrm{d}t^3} = \varepsilon\Delta T^3 \sum_n P(n,\eta)\Big[&-\lambda_n^2 e^{i\Delta T\lambda_n t}\mathbf{N}_{n,j} + \\ &\left(2i\varepsilon\sum_k \lambda_n e^{i\Delta T\lambda_n t}\frac{\partial \mathbf{N}_{n,j}}{\partial y_k}\right)\left(\sum_{n'} e^{i\Delta T\lambda_{n'} t}\mathbf{N}_{n',j}\right) + \\ \varepsilon^2\sum_{k,l} e^{i\Delta T\lambda_n t}\frac{\partial^2 \mathbf{N}_{n,j}}{\partial y_k \partial y_l}&\left(\sum_{n'} e^{i\Delta T\lambda_{n'} t}\mathbf{N}_{n',j}\right)\left(\sum_{n''} e^{i\Delta T\lambda_{n''} t}\mathbf{N}_{n'',j}\right) + \\ &\sum_{k'} e^{i\Delta T\lambda_n t}\frac{\partial \mathbf{N}_{n,j}}{\partial y_{k'}}\left(\varepsilon\sum_{n'}\lambda_{n'} e^{i\Delta T\lambda_{n'} t}\mathbf{N}_{n',j} + \right. \\ &\left.\varepsilon^2\sum_{n''} e^{i\Delta T\lambda_{n''} t}\mathbf{N}_{n'',j}\sum_{n'''}\sum_{l'} e^{i\Delta T\lambda_{n'''} t}\frac{\partial \mathbf{N}_{n''',j}}{\partial y_{l'}}\right)\Big],\end{aligned} \tag{4.50}$$

In bounding the timestepping error, we are interested in the norm of this quantity. Recalling that we are working with a finite-dimensional system of ODEs and applying the triangle and Cauchy-Schwarz inequalities we find that

$$\begin{aligned}\left\|\frac{\mathrm{d}^3 y_j(t)}{\mathrm{d}t^3}\right\| \le \varepsilon\Delta T^3\sum_n \|P(n,\eta)\||\mathbf{N}_{n,j}|&\left(\|\lambda_n^2\| + \|2\lambda_n\varepsilon\|\left|\sum_k \frac{\partial \mathbf{N}_{n,j}}{\partial y_k}\right| + \right. \\ &\left.\varepsilon^2|\mathbf{N}_{n,j}|\left|\sum_{k,l}\frac{\partial^2 \mathbf{N}_{n,j}}{\partial y_k \partial y_l}\right| \|\varepsilon\lambda_n\|\left|\sum_k \frac{\partial \mathbf{N}_{n,j}}{\partial y_k}\right| + \varepsilon^2\left|\frac{\partial \mathbf{N}_{n,j}}{\partial y_k}\right|^2\right).\end{aligned} \tag{4.51}$$

Now, as $\mathbf{N}$ and all of its spatial derivatives up to and including $p = 2$ are bounded by M by (4.41), we write

$$\begin{aligned}\left\|\frac{\mathrm{d}^3 y(t)}{\mathrm{d}t^3}\right\| &\le \varepsilon\Delta T^3 M\max_{n\in\mathbb{N}} P(n,\eta)\|\lambda_n^2 + 3\lambda_n\varepsilon M + \varepsilon^2 M^2\| &(4.52)\\ &\le \varepsilon\Delta T^3 M\max_{n\in\mathbb{N}} P(n,\eta)\|\lambda_n + C_f\varepsilon M\|^2, &(4.53)\end{aligned}$$

where C_f is a positive constant. We will now assume that $|\lambda_n| \neq 0$ as we are interested in the sup-norm of these values, which is nonzero when near-resonances are included. The directly resonant case has been treated by Haut and Wingate (2014). Then we must consider two possibilities. Firstly, if $|\lambda_n| \le 1$, then we define some constant, K_1,

$$K_1 = (1 + C_f\varepsilon M)^2. \tag{4.54}$$

[6] This may be read as 'capital-Rho' or, if the reader is given to a poetic mood, as a Latin P in honour of Saint Prokop of Sázava. In one of the oldest Czech legends, Saint Prokop harnessed a devil to his plow and thus tilled his land. Such wise application of chaotic, otherwise harmful forces may be familiar to students of economics as a metaphor for the invisible hand of the market, directing greed into productivity (Heller, 2005; Sedláček, 2011). Such a metaphor seems particularly apt here as well.

If $|\lambda_n| > 1$, the binomial theorem yields:

$$\begin{aligned}(|\lambda_n| + C_f \varepsilon M)^p &= \sum_{j=0}^{p} \binom{p}{j} (|\lambda_n|)^{p-j} (C_f \varepsilon M)^j, \\ &\leq \sum_{j=0}^{p} \binom{p}{j} (|\lambda_n|)^{p} (C_f \varepsilon M)^j, \\ &= |\lambda_n|^p \sum_{j=0}^{p} \binom{p}{j} (C_f \varepsilon M)^j, \\ &= |\lambda_n|^p K_2.\end{aligned}$$

And then we may write

$$(|\lambda_n| + \varepsilon \Delta T M)^2 \leq \max(K_1, |\lambda_n|^2 K_2). \tag{4.55}$$

As for the Rotating Shallow Water Equations there must always be a value of λ_n which is strictly greater than one, we shall assume that it is the second value which is the maximum. We now let $C = C_t K$. Finally, we bound the nonlinear term in the same fashion as 4.2, where the fact that:

$$\begin{aligned}M &= \sup_{\mathbf{y} \in D} \sup_{0 \leq t \leq L} \|\mathbf{f}(\mathbf{y}, t)\| \\ &= \sup_{\mathbf{y} \in D} \sup_{0 \leq t \leq L} \left\| \sum_n \Delta T e^{i \Delta T \lambda_n t} \mathbf{N}_n(\mathbf{y}) \right\| \\ &\leq \sup_{\mathbf{y} \in D} \sup_{0 \leq t \leq L} \left(\sum_n \|\mathbf{N}_n(\mathbf{y})\| \right) < \infty,\end{aligned}$$

completes the proof by providing an upper bound for the nonlinear operator as in 4.2. This provides a bound for the error due to timestepping which does not depend directly on the solution, but rather on the general properties of the nonlinearity, in particular the triadic interactions. □

In Section 4.4 we found that the averaging error increases proportional to the length of the averaging window. In this section we have found that the timestepping error follows the opposite trend: an increase in the averaging window corresponds to a decrease in the timestepping error. Recall that with the timestepping error we are measuring the fidelity of the numerical approximation to the averaged solution to the *averaged* solution. Thus concerns of fidelity to the full solution do not apply – they are dealt with by the bound on the averaging error instead. Rather, an increase in the averaging window decreases the stiffness inherent in the equations and with it the error.

4.5.1 *The Stiffness Regulator Function*

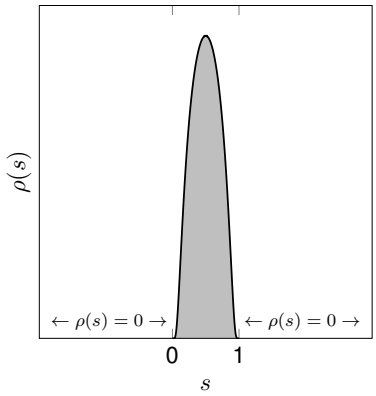

Figure 4.7: Extension of the limits of the integrating kernel to $\pm\infty$. Since $\rho(s)$ is identically zero outside of its support, these regions do not contribute to the integral and may be freely integrated over.

To clarify this somewhat, we define the *stiffness regulator function*, $\Lambda(\eta)$, which describes the filtering independent of the gain due to the scale separation and the coarse timestep, and which appears in the bound on the timestepping error. This provides novel understanding of the role of quadratic nonlinear interaction in timestepping under averaging.

$$\Lambda(\eta) \equiv \max_{n\in\mathbb{N}} |\lambda_n|^2 \frac{1}{\eta} \int_0^1 \rho(s) e^{i|\lambda_n|\eta\Delta T s} \, \mathrm{d}s. \tag{4.56}$$

$\Lambda(\eta)$ provides a measure of the extent to which the averaging integral mitigates the numerical stiffness. Recall from the discussion of stiffness in Section 1.1 that when the maximum $|\lambda_n|$ is large, as it is for highly oscillatory problems, it causes the right-hand side of the ODE to be very large, which is to say it induces steep gradients requiring a small numerical timestep. In contrast, the integral term in (4.56) tends to zero as $|\lambda_n|$ gets large, and does so superlinearly because of the integrating kernel, $\rho(s)$ (Haut and Wingate, 2014).

Consider the P-function (4.49), from which $\Lambda(\eta)$ is composed. Note that since the kernel is finitely-supported on $s = [0, 1]$ we may trivially extend the limits of integration to $\pm\infty$. Then,

$$\frac{1}{\eta} \int_0^1 \rho\left(s\right) e^{i|\lambda_n|\Delta T\eta s} \, \mathrm{d}s = \frac{1}{\eta} \int_{-\infty}^{\infty} \rho\left(s\right) e^{i|\lambda_n|\Delta T\eta s} \, \mathrm{d}s. \tag{4.57}$$

This is immediately recognisable as the Fourier transform of $\rho(s)$ evaluated at $\omega = |\lambda_n|\Delta T\eta$, up to a constant of normalisation. If we assume a bump function as a kernel, we find (following Johnson (2007)) that in the asymptotic limit of large $|\lambda_n|$,

$$\int_0^1 \rho\left(s\right) e^{i|\lambda_n|\Delta T\eta s} \, \mathrm{d}s \sim C_0 e^{-C_1\sqrt{|\lambda_n|\Delta T\eta}}. \tag{4.58}$$

If we slightly relax the restrictions of Definition 4.1 and assume that a sufficiently sharply-peaked Gaussian kernel satisfies them to within a numerical approximation we may consider the asymptotic behaviour of the kernel as $|\lambda_n|$ is large, from which we find

$$\int_0^1 \rho\left(s\right) e^{i|\lambda_n|\Delta T\eta s} \, \mathrm{d}s \sim C_0 e^{-C_1(|\lambda_n|\Delta T\eta)^2}. \tag{4.59}$$

In this latter case, we would write for a second-order timestepping method such as Strang splitting that the stiffness regulator function takes the form

$$\Lambda(\eta) \approx \max_{n\in\mathbb{N}} |\lambda_n|^2 \frac{1}{\eta} C_0 e^{-C_1(|\lambda_n|\Delta T\eta)^2}. \tag{4.60}$$

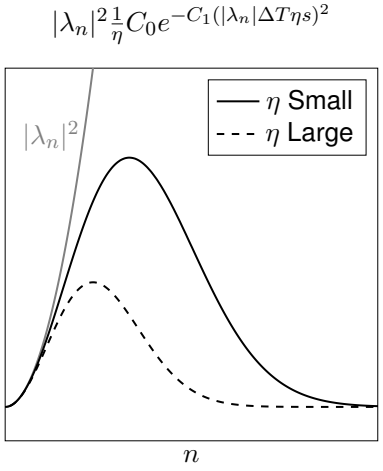

Figure 4.8: The contents of the stiffness regulator function, i.e. $\Lambda(\eta)$ except the $\max$, is shown for two values of averaging window length, η and versus n. Resonant distance increases along the horizontal axis. If not for the averaging, numerical stiffness would increase proportionally to n. The effect of the averaging window is to filter out the stiffest terms for large n, while leaving the lower ones unaffected. The reduction in the maximum of these curves is visible as η is increased. This is precisely the mechanism of stiffness reduction which makes the algorithm viable. $|\lambda_n|^2$ with no averaging applied is shown in grey for comparison.

Figure 4.8 shows this function for various values of η. $\Lambda(\eta)$ is exactly the maximum of the curves shown, which we can see are bounded. The right-hand side multiplier without averaging, on the other hand, is proportional to $|\lambda_n|^p$ and so does not achieve a maximum as $n \to \infty$. This means that the timestep is limited by the fastest scale in the system, which is itself a function of the spatial resolution of the model.

We see in studying $\Lambda(\eta)$ precisely how the averaging procedure filters the fast oscillations which arise due to the quadratic nonlinearity. For any amount of averaging applied, the stiffness regulator function achieves a lower magnitude than $\max_{n\in\mathbb{N}} |\lambda_n|^p$ – which replaces $\Lambda(\eta)$ in the unaveraged analogue of this system – would on its own. This reduction in the multiplier on the right-hand side reduces the numerical stiffness. As η is increased the system is more aggressively averaged leading to further reductions in stiffness at the cost of some fidelity to the unaveraged equations.

Crucially, the nature of the averaging is such that the slow components of the solution, which both contribute the most to the long-time dynamics of the system and the least to its numerical stiffness are affected much less than the fast components, which permits accuracy in the solution. This is readily apparent in Figure 4.8.

Recall from Section 1.1 (*cf.* Durran (2010)) that implicit methods, while providing the desired increase in stable timestep, commit nontrivial errors in the *linear* waves. This section has shown that it is the nonlinear combination of linear waves which is relevant to the quality of the solution, and so it is important to resolve the linear waves at all scales.

As a final point, $\Lambda(\eta)$ is bounded for all p and tends rapidly to zero as $\eta \to \infty$. This will prove important to use in proving convergence of APinT in the next chapter.

It was claimed at the end of Chapter 1 that triad interactions are in fact discrete components of nonlinear oscillation and that they provide a natural way to consider wave averaging and solution quality subject to oscillatory stiffness. Recalling that the set of all λ_n is simply an ordering of the set of all triad interactions based on their resonant nearness provides the justification for this claim.

4.6 *The Full Bound and Results*

THEOREMS 4.2 AND 4.3 describe the primary sources of error in the coarse solver. Based on these bounds, we seek a bound on the error in the coarse solver. Figure 4.9 sketches this bound conceptually.

Theorem 4.4. *Let ΔT denote the coarse timestep for a second order numerical method. We assume a finite scale separation on the order of ε. For an averaging window of length η, the total error in the coarse timestepping for the APinT algorithm is bounded by:*

$$\|\mathbf{x}(t) - \mathbf{y}_{\Delta T}(t)\| \leq M\varepsilon\Delta T\left((C_0 + C_1\varepsilon)\eta + D_1(\Delta T)^3\Lambda(\eta)\right), \quad (4.61)$$

where M is the sup-norm over the nonlinear operator as in Theorems 4.2 and 4.3 and C_0, C_1, and D_1 are finite constants.

♦

Proof. By the triangle inequality, we may write

$$\begin{aligned}\|\mathbf{x}(t) - \mathbf{y}_{\Delta T}(t)\| &= \|\mathbf{x}(t) - \mathbf{y}(t) + \mathbf{y}(t) - \mathbf{y}_{\Delta T}(t)\|, \\ &\leq \|\mathbf{x}(t) - \mathbf{y}(t)\| + \|\mathbf{y}(t) - \mathbf{y}_{\Delta T}(t)\|.\end{aligned}$$

Theorem 4.2 is used to bound the first term, i.e.

$$\|\mathbf{x}(t) - \mathbf{y}(t)\| \leq M(C_0 + C_1\varepsilon)\varepsilon\Delta T\eta. \quad (4.62)$$

Applying Theorem 4.3 and equation (4.56) to the second term yields

$$\begin{aligned}\|\mathbf{y}(t) - \mathbf{y}_{\Delta T}(t)\| &\leq MCC_1(\Delta T)^3\varepsilon\Lambda(\eta), \quad (4.63)\\ &\leq MD_1(\Delta T)^3\varepsilon\Lambda(\eta), \quad (4.64)\end{aligned}$$

where $\Lambda(\eta)$ is bounded independently of λ_n for any averaging window length, η. Combining the bounds in equations (4.62) and (4.64) gives the theorem as desired.

□

This proof directly explains the optimisation problem which was alluded to at the end of Chapter 1 in Figure 1.7. While the errors arising from either averaging and timestepping may be individually minimised with $\eta \to 0$ and $\eta \to \infty$ respectively, doing either of these things would cause the other source of error to be very large – prohibitively large for practical numerical use. The optimal averaging window is that which minimises the sum of these two sources of error.

It was also claimed in Section 1.1, following Higham and Trefethen (1993), that stiffness depends on a finite time interval. Again, we see here that this claim is substantiated. The time interval with which we are concerned is the coarse timestep, ΔT, which we have shown has an effect on both the magnitude of the error and on the location of the minimum point in its appearance in the D_1 term which describes the stiffness.

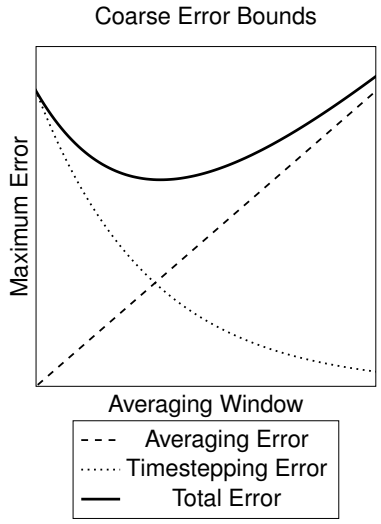

Figure 4.9: The primary sources of error in the coarse solver. The averaging error increases as more averaging is applied to the system, while the timestepping error follows the opposite trend and decreases. The worst-case total error is the sum of these two sources of error. The minimum as a function of η is then understandable as the 'Goldilocks point' where the sum of the two error sources achieves a minimum.

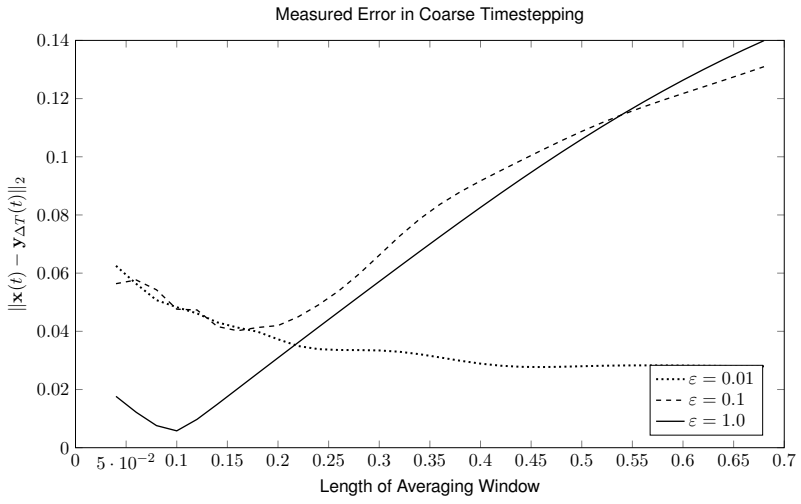

Figure 4.10: The measured error in the coarse timestepping for the RSWE as a function of the averaging window length, η, with $\Delta T = 0.1$. The existence of an optimal averaging window and the increasing relevance of it as ε gets large are both visible, in line with the results of Theorem 4.4.

The effect of varying the averaging window length for various values of ε is shown in Figures 4.10 and 4.11. The difference in the two figures is in the coarse timestep, ΔT, which is doubled in Figure 4.11. As predicted by Theorem 4.4, the optimisation problem for the averaging window length depends on both ε and ΔT. For a numerical solver which does not rely on wave averaging, a timestep of $\Delta T = 0.1$ is quite large for a problem of this type, particularly for the very stiff problem where $\varepsilon = 0.01$.

Compared to Figure 2.3, which showed the error for an exponential integrator with $\varepsilon = 0.01$, taking an optimally-averaged coarse solver with $\Delta T = 0.1$ provides an improvement in accuracy of approximately an order of magnitude (*cf.* Figure 4.10) when compared to the state-of-the-art for oscillatory stiff problems with such a large timestep. As mentioned in Section 1.1, accuracy can be as much of a consideration as stability in the context of oscillatory stiffness. Figure 4.11 provides results of very close to this quality, but with double the timestep which is possible without using the fast-wave averaged, HMM-style method which we have discussed in this chapter.

4.6.1 Timestep Extension Results

It is possible to make a more direct comparison as well. We consider the 1-D RSWE, solved spectrally with $N_x = 32$ and no hyperviscosity so as to more clearly consider the effects of oscillatory stiffness. We have already shown in Chapter 2 that exponential integrators with Strang splitting provide a method of increasing the explicit timestep limit for oscillatory stiff problems. As an illustration, we compare the timestep limits of the Strang splitting solver with the HMM-based 'coarse' solver. As this is a numerical exper-

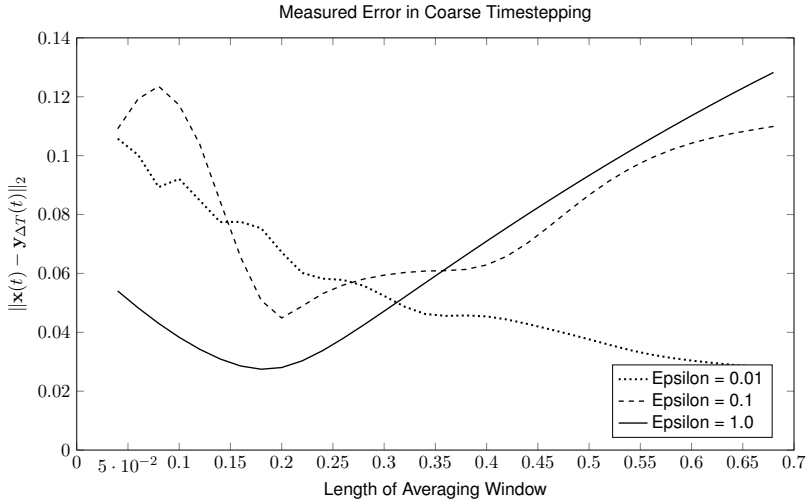

Figure 4.11: The numerically measured error in the coarse timestepping for the RSWE as a function of the averaging window length. Here, the timestep is $\Delta T = 0.2$, which is double that of Figure 4.10. It is apparent that the different time interval is relevant to the optimal averaging. In particular, the optimal η when $\varepsilon = 1$ is approximately equal to the coarse timestep in both cases.

iment, we have taken an 'unstable' timestep to be one which led to single-precision overflow when solved on a temporal domain of $t = [0, 50]$.[7]

In both cases, an optimal averaging window was used, which means that for the $\varepsilon = 0.01$ case the averaging applied was much stronger than in the $\varepsilon = 1.0$ case. We see (*cf.* Figure 4.12) that solving the optimally-averaged HMM equations provided a significant increase in the maximum stability limit for the highly-oscillatory situation. Again, this is compared to an exponential integrator which already handles the rapid oscillations much better than most explicit methods.

In the less-stiff cases, we do not see as much of an improvement in the timestep limit. This is because, especially when $\varepsilon = 1.0$, the oscillations are already quite slow and so there is less of a timestep restriction leading to very little improvement to be made by averaging. In reality, this does not cause a problem, as standard methods are quite capable of handling this 'easy' problem.

There is also a physical consideration since the coarse solver is computing *optimally-averaged* solutions, rather than *fully-averaged*. To put it succinctly, *optimal* averaging minimises error for given conditions of timestep and scale separation, while *full* averaging filters everything but direct resonances and therefore maximises timestep with complete disregard for accuracy. As $\varepsilon \to 0$ (*cf.* Section 3.3) both methods tend towards one another. For larger ε, however, it is necessary to retain more near-resonances in order to retain accuracy, as has been shown in this chapter. This necessarily means progressively less stiffness reduction as ε increases.

It is practically convenient to have a numerical method which may be tuned in real-time through the choice of the averaging

[7] This method is undoubtedly imperfect, but gives a good impression of relative stability bounds of one method to another on a fixed problem size.

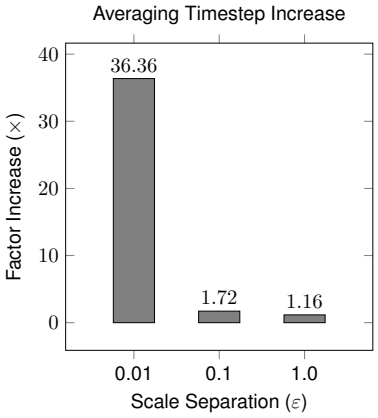

Figure 4.12: The improvement in the maximum timestep is shown for the fast-wave averaged method as compared to an unaveraged exponential integrator. We see that in the rapidly-oscillating situation where $\varepsilon = 0.01$ a significant increase is possible. In the less stiff case, less of an improvement is made, as there is less room to improve.

window to function across a wide range of ε. The coarse solver proposed and analysed in this chapter provides such a method, as opposed to one which *only* works in the stiff case described by the asymptotics of Section 1.4.

Now that we have developed and explained the 'coarse' solver, we will look at a practical application of it. The original use of this solver was in Parallel-in-Time simulation (Haut and Wingate, 2014) and the results of this chapter allow us to extend their proof of convergence in the limit of $\varepsilon \to 0$ to the finite case. We shall also discuss the particularities of choosing an optimal averaging window in computational practice in the coming chapter.

Key Points

- We are able to numerically compute the average with the Heterogeneous Multiscale Method.
- The error committed by the averaging method is dominated by the averaging error and the timestepping error, giving rise to the 'Goldilocks Point' seen in parameter studies.
- An increase in the averaging window length corresponds to an increase in averaging error and a decrease in timestepping error, and vice versa.
- The ordered set of triadic interactions provides a natural atomic unit of solution for quadratically-nonlinear systems – not the linear waves.

5 *From Parareal to APinT*

> The way the processor industry is going, is to add more and more cores, but nobody knows how to program those things. I mean, two, yeah; four, not really; eight, forget it.
>
> Steve Jobs, Apple

As discussed in Chapter 1, there is a limit on the extent to which simulations may be sped up through spatial parallelism. *Parallel in Time* methods aim to increase the speedup available on a massively parallel machine by extending parallelism to the temporal domain. There are several different methods of time parallelism, but we shall consider only the *Parareal* method here.

The Parareal method, proposed by Lions et al. (2001) and further expanded upon by Maday and Turinici (2003), first approximates the solution to an initial-value problem via a coarse timestepping method, which is then iteratively refined parallel-in-time via fine timesteps, such that the solution converges to the fine solution. To illustrate this, consider some solution to an initial value problem, $\mathbf{U}$, and an approximation to that solution, $\overline{\mathbf{U}}$. It is then an identity that

$$\mathbf{U} = \overline{\mathbf{U}} + (\mathbf{U} - \overline{\mathbf{U}}). \tag{5.1}$$

Let $\varphi_{\Delta T}(\mathbf{u}_0)$ denote the evolution operator associated with the numerical solution of the differential equation we wish to solve in a Parareal fashion such that $\mathbf{u}(t) = \varphi_{\Delta T}(\mathbf{u}_{n-1})$ numerically solves the full equation over an interval of ΔT. Similarly $\overline{\varphi}_{\Delta T}(\mathbf{u}_{n-1})$ numerically solves some suitable approximation thereof.

We then divide the time domain into N finite subintervals, $[n\Delta T, (n+1)\Delta T]$, where $n = 0, \ldots N-1$. Writing $\mathbf{U}_n = \mathbf{u}(n\Delta T)$ and neglecting truncation and discretisation errors the identity in equation (5.1) takes the form

$$\mathbf{U}_n = \overline{\varphi}_{\Delta T}(\mathbf{U}_{n-1}) + [\varphi_{\Delta T}(\mathbf{U}_{n-1}) - \overline{\varphi}_{\Delta T}(\mathbf{U}_{n-1})]. \tag{5.2}$$

Parareal methods proceed by computing approximations to the solution, $\mathbf{U}_n^k$, in an iterative fashion where k denotes the iteration

level. Formally, we write

$$\mathbf{U}_n^k = \overline{\varphi}_{\Delta T}(\mathbf{U}_{n-1}^k) + (\varphi_{\Delta T}(\mathbf{U}_{n-1}^{k-1}) - \overline{\varphi}_{\Delta T}(\mathbf{U}_{n-1}^{k-1})), \quad k = 1, 2, \ldots \quad (5.3)$$

Here, since the right-hand side quantities $\mathbf{U}_{n-1}^{k-1}$ in the difference $[\varphi_{\Delta T}(\mathbf{U}_{n-1}^{k-1}) - \overline{\varphi}_{\Delta T}(\mathbf{U}_{n-1}^{k-1})]$ are already computed at iteration level k, the difference can be computed in parallel for all n. Since the computation of $\overline{\varphi}_{\Delta T}(\mathbf{U}_{n-1}^k)$ is cheap, the overall computation is fast in a parallel sense if the iterates converge quickly.

In order to do this in practice, we start with an initial approximation computed by the so-called *coarse* solver, corresponding to $\overline{\varphi}_{\Delta T}(\cdot)$. We then solve the differential equation using both the coarse solver and the *fine* solver, $\varphi_{\Delta T}(\cdot)$, over intervals of ΔT starting at the initial conditions arising from the first approximation. This is done in a time-parallel fashion, i.e. each coarse time interval is computed simultaneously. In this way, Parareal may be thought of as an extension of domain decomposition methods to the temporal domain (Maday and Turinici, 2003).

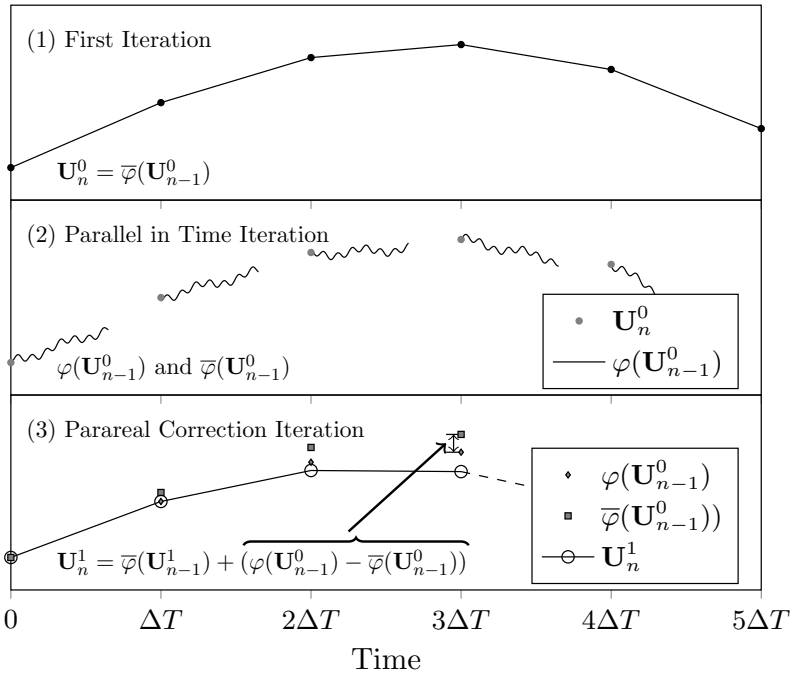

Figure 5.1: A schematic of the first iteration of the Parareal algorithm. (1): the solution is found at intervals of $j\Delta T$ by the coarse solver in serial. This permits the parallel-in-time iteration to proceed. (2): the parallel-in-time iteration. The coarse (not shown) and fine solvers integrate over intervals of ΔT, starting from the initial conditions found in the first iteration. Each interval is integrated simultaneously, hence the time-parallelism. (3): the Parareal correction iteration. Using the results from the previous iteration level, a serial sweep is made across to correct the solution at times $j\Delta T$. From here, steps (2) and (3) are repeated with the solution found in (3) providing the initial conditions for the parallel-in-time iteration in (2) until sufficient convergence has been achieved. The difference $\varphi(\mathbf{U}_4^0) - \overline{\varphi}(\mathbf{U}_4^0)$ is indicated in the third subplot.

Finally, equation (5.3) is the *Parareal correction iteration* which is performed as a serial sweep through time. This process is repeated until the solution has converged. This procedure for the initial coarse solve (iteration 0) and the first Parareal iteration cycle is shown in Figure 5.1. Algorithm 5.4 provides a pseudocode implementation of the Parareal algorithm for a general differential equation without specifying the details of the coarse and fine solvers.

The solution obtained from Parareal will converge to that of the fine solver if it converges (Gander and Vandewalle, 2005). It

Algorithm 5.4: The Parareal Algorithm

$\mathbf{U}_0^{\text{old}} \leftarrow \mathbf{u}_0$ ▷ Initial Condition
for $n = 1, \ldots, N-1$ **do** ▷ Initial guess with Slow Solver
  $\mathbf{U}_n^{\text{old}} \leftarrow$ `Coarse_Solver`$\left(\mathbf{U}_{n-1}^{\text{old}}, \Delta T\right)$
end for

$\mathbf{U}_o^{\text{new}} \leftarrow \mathbf{u}_0$
▷ Iterative refinement to convergence
while $\max_n \|\mathbf{U}_n^{\text{new}} - \mathbf{U}_n^{\text{old}}\| / \|\mathbf{U}_n^{\text{new}}\| >$ `tol` **do**
  parfor $n = 1, \ldots, N-1$ **do** ▷ Parallel-in-Time Step
    $\mathbf{U}_n^{\text{old}} \leftarrow \mathbf{U}_n^{\text{new}}$
    $\mathbf{V}_n \leftarrow$ `Fine_Solver`$\left(\mathbf{U}_{n-1}^{\text{old}}, \Delta t, \Delta T\right)$
    $\mathbf{V}_n \leftarrow \mathbf{V}_n -$ `Coarse_Solver`$\left(\mathbf{U}_{n-1}^{\text{old}}, \Delta T\right)$
  end parfor

  for $n = 1, \ldots, N-1$ **do** ▷ Parareal Correction Iteration
    $\mathbf{U}_n^{\text{new}} \leftarrow$ `Coarse_Solver`$\left(\mathbf{U}_{n-1}^{\text{old}}, \Delta T\right) + \mathbf{V}_{n-1}$
  end for
end while
return $\mathbf{U}_1^{\text{new}}, \ldots, \mathbf{U}_N^{\text{new}}$

then follows that the choice for which fine solver to use is that which is desired in practice, but which requires parallel acceleration through the Parareal framework. The same timestep size as would otherwise be used, denoted by Δt and called the *fine timestep*, is taken.

The choice of the coarse solver is more difficult and is the major advancement in this work, extending that of Haut and Wingate (2014). In practice, there are three requirements on the coarse solver which must be satisfied for the Parareal method to be of practical use:

1. The coarse timestepping method must permit large timesteps, i.e. $\Delta T \gg \Delta t$.

2. The coarse timesteps much be computationally inexpensive.

3. The method must converge quickly, i.e. $\mathbf{U}_n^k \to \mathbf{U}_n$ rapidly as $k \to \infty$.

Should the differential equation being solved be sufficiently well-behaved, i.e. lacking in stiffness and without prohibitively expensive nonlinearities, these conditions are easily satisfied. In fact, the simplest possible Parareal algorithm takes the coarse timestepping method to be the same as the fine timestepping in every regard except for a longer timestep (Lions et al., 2001). Other possibilities include a coarser space discretisation (Fischer et al., 2005), and/or a modified physical model (Maday and Turinici, 2003). It is the last of

these which shall prove particularly interesting for our work here.

Examples of applications of the Parareal algorithm being applied to parabolic PDEs include simulations of financial markets (i.e. the Black-Scholes equation for an American put (Bal and Maday, 2002)), the Navier-Stokes equations (Fischer et al., 2005; Trindade and Pereira, 2004), fluid/structure interaction (Farhat and Chandesris, 2003), a nonlinear parabolic evolutionary equation via the finite element method (He, 2010), skin transport problems (Kreienbuehl et al., 2015) and the p-Laplacian (Falgout et al., 2016).

Hyperbolic systems, on the other hand, are known to be an issue for the Parareal method. Bal (2005) showed that while a sufficiently damped coarse solver is unconditionally stable for parabolic systems, but not hyperbolic ones. Staff and Rønquist (2005) showed that Parareal is unstable for purely imaginary eigenvalues in the solution operator. Finally, Gander and Vandewalle (2005) showed that for advective problems vanilla Parareal is either unstable or inefficient. Solving hyperbolic problems with the Parareal method relies on the application of techniques to stabilise the coarse solve (Haut and Wingate, 2014; Ariel et al., 2016).

There have been several modifications to the Parareal method which are suitable for highly oscillatory systems which assume that the system may be separated into fast and slow variables. In terms of ODEs, Legoll et al. (2013) have proposed a multiscale method for singularly perturbed ODEs where the fast dynamics are dissipative. Ariel et al. (2016) propose a method for highly oscillatory ODEs which is multiscale in nature but does not require explicit knowledge of the fast and slow variables. Gander and Hairer (2014) suggest Parareal methods for Hamiltonian dynamics.[1] Approaches using symplectic integrators with applications to molecular dynamics are presented in, for example, Audouze et al. (2009) and Bal and Wu (2008). Finally, Haut and Wingate (2014) proposed a method which is motivated by asymptotic solutions for fast singular limits of nonlinear evolutionary PDEs. It is an extension of this method which we study here and which we refer to as Asymptotic Parallel-in-Time (APinT). It takes its name from the modified coarse solver which is inspired by methods used in the asymptotic analysis of PDEs.[2]

[1] According to Tuynman (2014), the symbol H, commonly used to denote the Hamiltonian, was chosen by Joseph-Louis Lagrange in honour of the Dutch scientist Christiaan Huygens.

[2] It cannot be overstated that the APinT method does *not* rely on an asymptotic solution. Rather, it is the asymptotic derivation of fast wave averaging of Section 1.4 which inspired the method. Perhaps *Asymptotically-Inspired Parallel-in-Time* is a more accurate name.

Consider once again the general PDE (1.4). It is problems of this type which we are interested in solving with a Parareal method and we shall assume without loss of generality that ε may become arbitrarily small, leading to oscillatory stiffness and necessitating a well-designed coarse propagator.

In general, the maximum timestep for this type of system is $\mathcal{O}(\varepsilon)$, so in the case of $\varepsilon = \mathcal{O}(1)$, i.e. the less-stiff case, Parareal may be applied without any modifications. However, as $\varepsilon \to 0$ this

would require too small of a coarse timestep. The insight of Haut and Wingate (2014) was that a slow solution based on a coordinate transformation and a time average over the fast waves in the nonlinear operator provides a convergent and efficiently-computable coarse approximation. In fact, they showed that under suitable assumptions of smoothness, superlinear convergence is obtained as $\varepsilon \to 0$.

The APinT algorithm as we shall apply it here consists of a Strang splitting-based exponential integrator for the fine timestepping (*cf.* Section 2.3) and the HMM-style averaged integrator for the coarse timestepping (*cf.* Chapter 4).

5.1 Complexity Bounds

Directly following Haut and Wingate (2014) we may discuss the time complexity of the APinT algorithm, which we reprint here for clarity with only minor additions. Assume without loss of generality a time interval $[0, 1]$ which is sub-divided into N sub-intervals $[T_{n-1}, T_n]$, each of length $\Delta T = 1/N$. Let M denote the number of fine timesteps used in each interval such that $M = \Delta T/\Delta t$. Finally, let τ_c denote the wall-clock time required to compute the coarse solution, $\overline{\varphi}_{\Delta T}(\mathbf{u}_0)$, over a coarse timestep and τ_f denote the wall-clock time required to compute the fine solution, $\varphi_{\Delta T}(\mathbf{u}_0)$, over a fine timestep.

The initial guess (iteration 0 in Figure 5.1) which must be computed serially requires a wall-clock time of $N\tau_c$. In order to move from iteration level k to iteration level $k+1$ we must compute the difference

$$\mathbf{V}_n^k = \varphi_{\Delta T}(\mathbf{U}_n^k) - \overline{\varphi}_{\Delta T}(\mathbf{U}_n^k). \tag{5.4}$$

This step may be performed in a parallel fashion and so requires a wall-clock time of $\tau_c + M\tau_f$. Finally, we must carry out the Parareal correction iteration

$$\mathbf{U}_n^{k+1} = \overline{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^{k+1}) + \mathbf{V}_n^k, \tag{5.5}$$

which runs in serial and requires a wall-clock time of $N\tau_c$, as $\mathbf{V}_n^k$ is already known and its addition is therefore an $\mathcal{O}(1)$ operation. The total wall-clock time after ν iterations is then

$$T_{\text{parareal}} = \nu(M\tau_f + N\tau_c + \tau_c) + N\tau_c. \tag{5.6}$$

This is compared to the serial cost of solving the equations which is

$$T_{\text{serial}} = NM\tau_f. \tag{5.7}$$

The estimated Parareal speedup is then

$$\frac{NM\tau_f}{\nu(M\tau_f + N\tau_c + \tau_c) + N\tau_c} \leq \min\left(\frac{\tau_f}{\tau_c}\frac{M}{\nu+1}, \frac{N}{\nu}\right). \tag{5.8}$$

This gives an upper bound which is proportional to $M = \Delta T/\Delta t$, which is the ratio of coarse to fine timesteps. That this value must be large was claimed earlier in this chapter to be the first requirement for a practical Parareal implementation and this provides the justification for that fact. Now consider

$$\frac{NM\tau_f}{\nu(M\tau_f + N\tau_c + \tau_c) + N\tau_c} \approx \frac{1}{\nu\tau_c}\frac{(N\tau_c)(M\tau_f)}{M\tau_f + N\tau_c}. \tag{5.9}$$

We then minimise the wall-clock time with a choice of $N\tau_c = M\tau_f$. Both τ_c and τ_f are fixed constants, leaving N and M to be chosen. In practice, the choice of N will generally be informed by the practicalities of the parallel architecture, e.g. how many nodes are available. In order to converge, the fine solver requires timesteps which are some fraction of ε, $\Delta t = a\varepsilon$, where $0 < a < 1$. This gives

$$N = \sqrt{\frac{\tau_f}{\tau_c}}\sqrt{\frac{1}{a\varepsilon}}, \quad M = \sqrt{\frac{\tau_c}{\tau_f}}\sqrt{\frac{1}{a\varepsilon}}, \tag{5.10}$$

which leads to an estimated parallel speedup of

$$\frac{T_{\text{serial}}}{T_{\text{parareal}}} = \frac{N\left(\frac{\tau_c}{\tau_f}N\right)}{\nu\left(\frac{\tau_c}{\tau_f}N + \frac{\tau_c}{\tau_f}N\right) + \frac{\tau_c}{\tau_f}N} \tag{5.11}$$

$$= \frac{1}{2\nu+1}\sqrt{\frac{\tau_f}{\tau_c}}\sqrt{\frac{1}{a\varepsilon}}. \tag{5.12}$$

The APinT method then provides an arbitrarily large speedup compared to serial numerical integrators as $\varepsilon \to 0$.

Recalling Algorithm 4.2, it is possible to compute the time average in an embarrassingly parallel manner. Doing so reduces the time complexity of the coarse solver from $\tau_c = \mathcal{O}(\overline{M})$ to $\tau_c = \mathcal{O}(\log \overline{M})$. In practice, this is necessary to satisfy the second requirement for a coarse timestepping method, which is that it be computationally inexpensive.

The time complexity is of interest in practice and provides some additional justification for the lengths we have gone to in developing the coarse solver. The primary focus of this work, however, is on the error analysis. We then turn our attention back to a consideration of the role that the errors committed by the coarse timestepping method play in the convergence of APinT. Understanding these will ultimately allow us to average optimally and therefore achieve the fastest possible convergence of the APinT method independent of scale separation.

As a final point on time complexity, improving the convergence means reducing the number of iterations, ν, which feeds directly back into the analysis of this section (*cf.* equation (5.12)). This then allows us to satisfy the third requirement on a useful coarse timestepper for the Parareal method – rapid convergence.

5.2 *Convergence of APinT*

The original authors of the APinT method showed superlinear convergence of the method in the asymptotic limit as $\varepsilon \to 0$ (Haut and Wingate, 2014). Recalling Chapter 3 this yields a solution which is comprised solely of directly resonant triads as all other interactions are filtered by the wave averaging procedure. By considering only this limit they were able to provide Theorem 5.1, which describes the convergence of the APinT method to the fine solution.

Theorem 5.1. *Consider a scale of Banach spaces $B_0 \supseteq B_1 \supseteq B_2 \supseteq \ldots$, such that functions in B_{j+1} are smoother than functions in B_j. Assuming that $\mathbf{u}_0 = \mathbf{u}(T_0) \in B_{j+k+1}$, the error, $\mathbf{u}(T_n) - \mathbf{U}_n^k$ after k Parareal iterations is bounded by*

$$\|\mathbf{u}(T_n) - \mathbf{U}_n^k\|_{B_j} \leq C_{k,j}(\Delta T^p + \varepsilon)\left(\Delta T^p + \frac{\varepsilon}{\Delta T}\right)^k)\|\mathbf{u}_0\|_{B_{j+k+1}}, \quad (5.13)$$

where p is the order of the coarse timestepping method, the norm is always taken in the Banach space denoted B_j or B_{j+k+1}, and $C_{k,j}$ is a constant that depends only on the Banach space-dependent constants $C_m, m = 0, 1, \ldots, k + j$ (recalling that the scale of Banach spaces extends from B_0 to B_{j+k+1}). ♦

By assuming that the Banach spaces coincide, which may not hold in the infinite-dimensional setting, they were able to show that the constants decrease with increasing k, yielding superlinear convergence. The reader is referred to Haut and Wingate (2014) for the proof of this theorem and necessary assumptions made. The central focus of this chapter is the extension of the convergence proof to the more physical case outside of the small-ε limit which necessarily requires the consideration of near-resonant interactions which led to the bounds on averaging and timestepping errors in Chapter 4.

We may now derive error bounds for the Parareal iteration on finite systems of ODEs. Using our improved error bound for the coarse solver which holds for finite ε, we modify the proof given in Haut and Wingate (2014), which held only as $\varepsilon \to 0$. For consistency we define several operators following Haut and Wingate

(2014). Let $\tilde{\varphi}_{\Delta T}(\cdot)$ be the evolution operator associated with numerically solving the slow equation using a second order method, such that $\tilde{\varphi}_{\Delta T}(\cdot)$ is a numerical approximation of $\overline{\varphi}_{\Delta T}(\cdot)$. Furthermore, let $\varphi_{\Delta T}(\cdot)$ denote the evolution operator for the fine solution. We then define:

$$\mathcal{E}_{\varphi,\overline{\varphi}}(\cdot) = \varphi_{\Delta T}(\cdot) - \overline{\varphi}_{\Delta T}(\cdot), \tag{5.14}$$

and

$$\mathcal{E}_{\overline{\varphi},\tilde{\varphi}}(\cdot) = \overline{\varphi}_{\Delta T}(\cdot) - \tilde{\varphi}_{\Delta T}(\cdot). \tag{5.15}$$

Then, as in Bal (2005), Haut and Wingate (2014), and based on Chapter 4 we make the following assumptions, where η is the length of the averaging window, M is the sup-norm over the nonlinear term, and $\Lambda(\eta)$ is the stiffness regulator function.

1. The operators $\varphi(\cdot)$ and $\overline{\varphi}(\cdot)$ are uniformly bounded for $0 \leq t \leq 1$

$$\|\varphi_t(\mathbf{u}_0)\| \leq C\|\mathbf{u}_0\|, \quad \|\overline{\varphi}_t(\mathbf{u}_0)\| \leq C\|\mathbf{u}_0\|. \tag{5.16}$$

2. The averaging method is accurate in the sense that

$$\|\varphi_t(\mathbf{u}_0) - \overline{\varphi}_t(\mathbf{u}_0)\| \leq \varepsilon\Delta T\eta M(C_1 + C_2\varepsilon)\|\mathbf{u}_0\|. \tag{5.17}$$

3. The averaged evolution operator satisfies

$$\|\overline{\varphi}_{\Delta T}(\mathbf{u}_1) - \overline{\varphi}_{\Delta T}(\mathbf{u}_2)\| \leq (1 + C\Delta T)\|\mathbf{u}_1 - \mathbf{u}_2\|, \tag{5.18}$$

   and the numerical approximation to the evolution equation satisfies

$$\|\tilde{\varphi}_{\Delta T}(\mathbf{u}_1) - \tilde{\varphi}_{\Delta T}(\mathbf{u}_2)\| \leq (1 + C\Delta T)\|\mathbf{u}_1 - \mathbf{u}_2\|. \tag{5.19}$$

4. Following Theorems 4.2 and 4.3 and equation (5.16), the error operators satisfy

$$\|\mathcal{E}_{\varphi,\overline{\varphi}}(\mathbf{u}_1) - \mathcal{E}_{\varphi,\overline{\varphi}}(\mathbf{u}_2)\| \leq \varepsilon\Delta T\eta M(C_1 + C_2\varepsilon)\|\mathbf{u}_1 - \mathbf{u}_2\|, \tag{5.20}$$

   and

$$\|\mathcal{E}_{\overline{\varphi},\tilde{\varphi}}(\mathbf{u}_1) - \mathcal{E}_{\overline{\varphi},\tilde{\varphi}}(\mathbf{u}_2)\| \leq \Delta T^{p+2}\varepsilon\Lambda(\eta)MC\|\mathbf{u}_1 - \mathbf{u}_2\|, \quad p \geq 1. \tag{5.21}$$

Having quantified the major sources of error in the coarse timestepping which will affect the convergence of Parareal, the proof of the convergence follows directly from these bounds.

Theorem 5.2. *Subject to the above assumptions, the error, $\mathbf{u}(T_n) - \mathbf{U}_n^k$, after the k-th Parareal iteration is bounded by*

$$\|\mathbf{u}(T_n) - \mathbf{U}_n^k\| \leq MC_g\left(C_1\Delta T^3\varepsilon\Lambda(\eta) + (C_2 + C_3\varepsilon)\varepsilon\eta\right)^{k+1}\|\mathbf{u}_0\|. \tag{5.22}$$

♦

Proof. The proof is by induction on k. When $k = 0$

$$\begin{aligned}\|\mathbf{u}(T_n) - \mathbf{U}_n^0\| &= \|\varphi_{\Delta T}(\mathbf{u}_0) - \tilde{\varphi}_{\Delta T}(\mathbf{u}_0)\| \\ &\leq \|\varphi_{\Delta T}(\mathbf{u}_0) - \bar{\varphi}_{\Delta T}(\mathbf{u}_0)\| + \|\bar{\varphi}_{\Delta T}(\mathbf{u}_0) - \tilde{\varphi}_{\Delta T}(\mathbf{u}_0)\| \\ &\leq M((C_1 + C_2\varepsilon)\varepsilon\Delta T\eta + C_3\Delta T^2)\|\mathbf{u}_0\|,\end{aligned}$$

where we have used (5.17), which bounds the error induced by the averaging procedure, to bound the first term and Lemma 4.3, which governs the timestepping error, for the second. Now assume that

$$\|\mathbf{u}(T_n) - \mathbf{U}_n^{k-1}\| \leq (\Delta T + \varepsilon)\left(C_1\Delta T^3\varepsilon\Lambda(\eta) + (C_2 + C_3\varepsilon)\varepsilon\eta\right)^k \|\mathbf{u}_0\|. \tag{5.23}$$

We may then write the Parareal iteration, (5.3) in the following form, using (5.20) and (5.21)

$$\begin{aligned}&\mathbf{u}(T_n) - \mathbf{U}_n^k = (\tilde{\varphi}_{\Delta T}(\mathbf{u}(T_{n-1})) - \tilde{\varphi}_{\Delta T}(\mathbf{U}_{n-1}^k)) + \\ &(\mathcal{E}_{\varphi,\bar{\varphi}}(\mathbf{u}(T_{n-1})) - \mathcal{E}_{\varphi,\bar{\varphi}}(\mathbf{U}_{n-1}^{k-1})) + (\mathcal{E}_{\bar{\varphi},\tilde{\varphi}}(\mathbf{u}(T_{n-1})) - \mathcal{E}_{\bar{\varphi},\tilde{\varphi}}(\mathbf{U}_{n-1}^{k-1}))\end{aligned} \tag{5.24}$$

By directly substituting equations (5.19), (5.20), and (5.21), we have

$$\begin{aligned}\|\mathbf{u}(T_n) - \mathbf{U}_n^k\| &\leq (1 + C\Delta T)\|\mathbf{u}(T_{n-1}) - \mathbf{U}_{n-1}^k\| + \\ &\quad M\left(C_1\Delta T^3\varepsilon\Lambda(\eta) + (C_2 + C_3\varepsilon)\varepsilon\Delta T\eta\right)\|\mathbf{u}(T_{n-1}) - \mathbf{U}_{n-1}^{k-1}\| \\ &\leq (1 + C\Delta T)\|\mathbf{u}(T_{n-1}) - \mathbf{U}_{n-1}^k\| + \\ &\quad M\Delta T\left(C_1\Delta T^2\varepsilon\Lambda(\eta) + (C_2 + C_3\varepsilon)\varepsilon\eta\right)^{k+1}\|\mathbf{u}_0\|.\end{aligned}$$

Finally, application of the discrete Grönwall inequality gives

$$\begin{aligned}\|\mathbf{u}(T_n) - \mathbf{U}_n^k\| &\leq \left(e^{C(T_n - T_0)} - 1\right) M\left(C_1\Delta T^2\varepsilon\Lambda(\eta) + (C_2 + C_3\varepsilon)\varepsilon\eta\right)^{k+1}\|\mathbf{u}_0\| \\ &= MC_g\left(C_1\Delta T^2\varepsilon\Lambda(\eta) + (C_2 + C_3\varepsilon)\varepsilon\eta\right)^{k+1}\|\mathbf{u}_0\|,\end{aligned}$$

where

$$C_g = e^{C(T_n - T_0)} - 1. \tag{5.25}$$

□

This proof generalises that of Haut and Wingate (2014), where they showed convergence for the asymptotic limit as $\varepsilon \to 0$, to finite ε as well. This is a significant improvement as for many physical applications such as weather and climate modelling ε remains finite.

5.2.1 *Convergence Independent of Scale Separation*

FOR THE APINT ALGORITHM TO CONVERGE, we require (following Theorem 5.2) that

$$C_1 \Delta T^3 \varepsilon \Lambda(\eta) + C_2 \varepsilon \eta + C_3 \varepsilon^2 \eta < 1. \tag{5.26}$$

We are then left with the problem of choosing an appropriate averaging window length, η, depending on the degree of scale separation, ε, and the filtered contribution of the triads as described by the stiffness regulator function, $\Lambda(\eta)$. In the interest of demonstrating that one exists, we assume the scaling (for example)

$$\eta = \frac{\Delta T}{\varepsilon^s}, \quad 0 < s < 1. \tag{5.27}$$

We then have:

$$C_1 \Delta T^3 \varepsilon \Lambda\left(\frac{\Delta T}{\varepsilon^s}\right) + C_2 \varepsilon^{1-s} \Delta T + C_3 \varepsilon^{2-s} \Delta T < 1, \tag{5.28}$$

as $\varepsilon \to 0$, our error also decreases for any value of the power s. $\Lambda\left(\frac{\Delta T}{\varepsilon^s}\right)$ is bounded, so as $\varepsilon \to 1$, all terms remain bounded and we may choose our coarse timestep accordingly to ensure convergence. This means that the method proposed here may be applied across the full range of $\varepsilon \in (0, 1]$ with only a change of averaging window length, which allows convergence for physical problems where the time scale separation may change throughout the computation. This is in contrast to the proof in the limit (Haut and Wingate, 2014) which proved convergence only for $\varepsilon \to 0$.

5.3 *Optimal Averaging for APinT*

IT WAS SHOWN in Section 5.2.1 that it is possible to choose the averaging window in such a way as to ensure convergence. We may go one step farther and choose the averaging window to obtain the fastest possible convergence. This involves a novel optimisation problem which does not discard the inherent nonlinearity of the underlying differential equations. Figure 5.2 shows the iterative error in APinT after three iterations for a range of values of scale separation.

As with the coarse error, there is a clear minimum with respect to the averaging window length for larger values of ε which leads to an optimisation problem to be solved. With reference to Theorem 5.2 above and recalling that k is the iteration level, we write this as

$$\min_{\eta \in \mathbb{R}^+} \left(C_1 \Delta T^3 \varepsilon \Lambda(\eta) + (C_2 + C_3 \varepsilon) \varepsilon \eta \right), \tag{5.29}$$

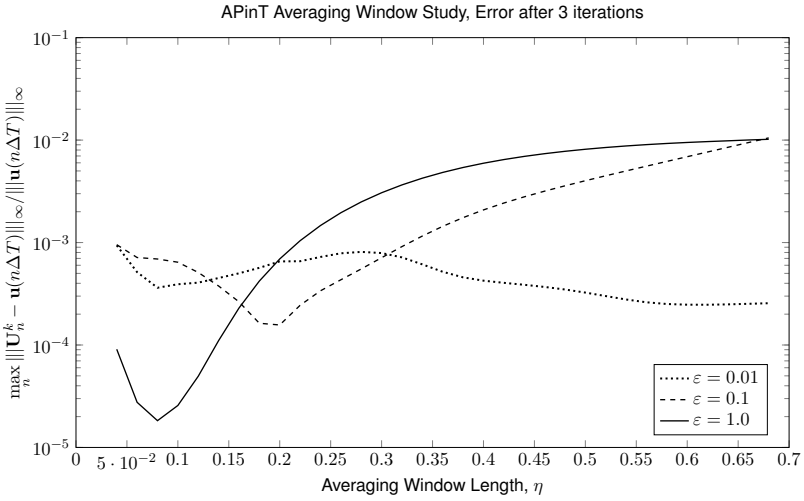

Figure 5.2: The iterative error for the APinT algorithm is shown after three iterations for a range of averaging window lengths and values of scale separation. We see the same behaviour as for the coarse solver, with an optimal choice of averaging window for the values of ε outside of the limit. As everywhere else in this work, a pseudospectral method was used on a spatially-periodic domain with $N_x = 64$, $\Delta T = 0.1$, $\Delta t = 10^{-4}$.

for some as-yet unknown constants C_1, C_2, and C_3. Seeking stationary points with respect to η and explicitly considering the stiffness regulator function (4.56), this then requires us to find η such that

$$\frac{\mathrm{d}}{\mathrm{d}\eta} \max_{\epsilon_\beta} \max_{\mathcal{S}_{k,\alpha}^{\epsilon_\beta}} \frac{\left|\omega_k^\alpha - \omega_{k_1}^{\alpha_1} - \omega_{k_2}^{\alpha_2}\right|^2}{\varepsilon} \int_0^1 \rho(s) e^{\frac{i\left|\omega_k^\alpha - \omega_{k_1}^{\alpha_1} - \omega_{k_2}^{\alpha_2}\right| \eta \Delta T}{\varepsilon} s} \,\mathrm{d}s + \frac{C_2 + C_3\varepsilon}{C_1 \Delta T^3} = 0. \quad (5.30)$$

This expression relies on several unknown constants. If these constants C_n were known, the optimal averaging window could be determined computationally. Equation (5.30) would then provide an approximation to the optimal window. Given some initial data of the type shown in Figure 5.2, C_n could be fit by, for example, least-squares. Doing so would require some initial experimental data for a given timestepping method but would permit the optimal averaging window to be recomputed 'on the fly' in a computation.

Certain practical issues arise in the computation of η. Firstly, the computation of $\frac{\mathrm{d}\Lambda}{\mathrm{d}\eta}$ requires all triads to be investigated, i.e. the maximum is taken over the set of all near-resonant sets. Doing so is computationally expensive, although if this computation were to be performed infrequently the cost could be negligible compared to the simulation cost. Additionally, finding η requires solving a transcendental equation in at least two variables (η, ε), both for the initial fitting of constants, and for the optimisation on the fly. We therefore propose a simpler model based on the behaviour of $\Lambda(s)$.

Restricting ourselves for this example to a Gaussian kernel, we may consider the asymptotic behaviour of the kernel as λ_n is large

as in Section 4.5.1. This gives

$$\frac{1}{\eta}\int_0^\eta \rho\left(\frac{s}{\eta}\right) e^{i\lambda_n \Delta T s} ds \sim C_0 e^{-C_1(|\lambda_n|\Delta T\eta)^2}. \tag{5.31}$$

We then multiply our approximation by $1 = \eta^2/\eta^2$ to obtain

$$\frac{\eta^2}{\eta^2} C_1 \Delta T^3 \varepsilon \Lambda(\eta) \approx \frac{D_1 \Delta T \varepsilon}{\eta^2}, \tag{5.32}$$

for some constant, D_1, since $x^2 e^{-x^2}$ is bounded independently of x. We then replace our first term in (5.29) and seek fixed points corresponding to the minimum error. This yields

$$\eta_{\text{optimal}} = \sqrt{\frac{D_1 \Delta T}{C_2 + C_3 \varepsilon}}. \tag{5.33}$$

This equation provides an estimate for the optimal averaging window length, η_{opt}, in terms of the computational parameters and the empirically-fit constants. This result is consistent with Theorem 5.2, as it exhibits a clear minimum for $\mathcal{O}(1)$ values of ε, with the optimal averaging window increasing as $\varepsilon \to 0$, as the asymptotic theory predicts. Both approximations are shown in Figure 5.3 for a set of minima extracted from a series of runs of the algorithm.

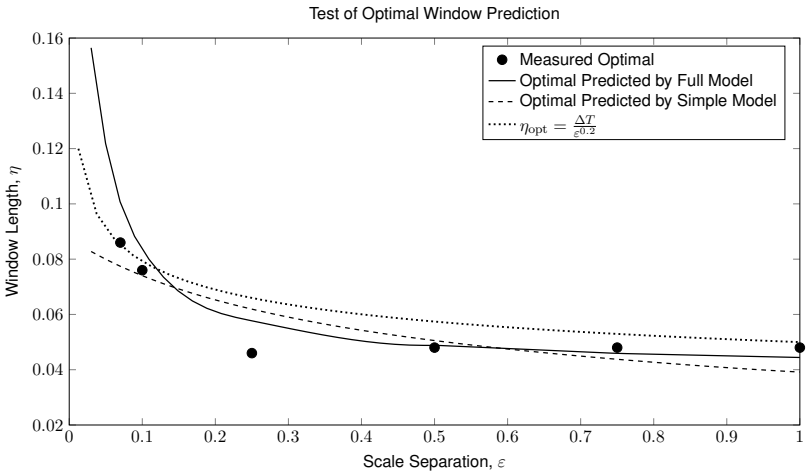

Figure 5.3: The optimal averaging window for the APinT solver is predicted in three different ways as a function of the scale separation. The measured optimal values for several runs with a coarse timestep of $\Delta T = 0.05$ are shown with the filled circles. The so-called 'full' or expensive model from (5.30) is shown with the solid curve. This model shows good agreement throughout the range and handles the long averaging windows needed as $\varepsilon \to 0$ as well. The simple model derived from asymptotic analysis on a Gaussian kernel is shown with a dashed line, and provides similar accuracy as the full model outside of the small-ε region, but at a dramatically reduced computational cost. Finally, the dotted line indicates the assumed scaling on the averaging window given in (5.27) with s taken empirically as 0.2. The trend towards a longer time averaging window being necessary for smaller ε is captured, while this scaling somewhat overestimates the window for larger values of scale separation, although it may be computed very cheaply.

The full model given in (5.30) provides a much closer approximation both to the behaviour for $\varepsilon = 1$ and as $\varepsilon \to 0$, and as an actual fit to the points. It does this, however, at the cost of several orders of magnitude more computational difficulty. The simple model of (5.33), on the other hand, provides a reasonable approximation to the error as a function of ε, but has the disadvantage of poorly resolving the trend in the limit as $\varepsilon \to 0$. While the simple prediction underestimates the optimal as $\varepsilon \to 0$, the behaviour in this range is well-understood (*cf.* Haut and Wingate (2014), Ariel et al. (2016)) and so a hybrid model may easily be applied in practice.

5.4 *Parameter Studies in One Dimension*

In order to validate the convergence results, a series of parameter studies have been performed with APinT on the 1-D RSWE. In all cases in this section, a Fourier pseudospectral method was used. Hyperviscosity was used for stability, but no other dissipation was present in the solution of the problem. The boundary conditions were periodic and the domain was of size $L = 2\pi$. The initial condition was an initially-stationary Gaussian height field.

Except where otherwise specified, all of the runs in this section used $\Delta t = 10^{-4}$ and $N_x = 64$. The smooth kernel of integration was taken to be Gaussian everywhere.

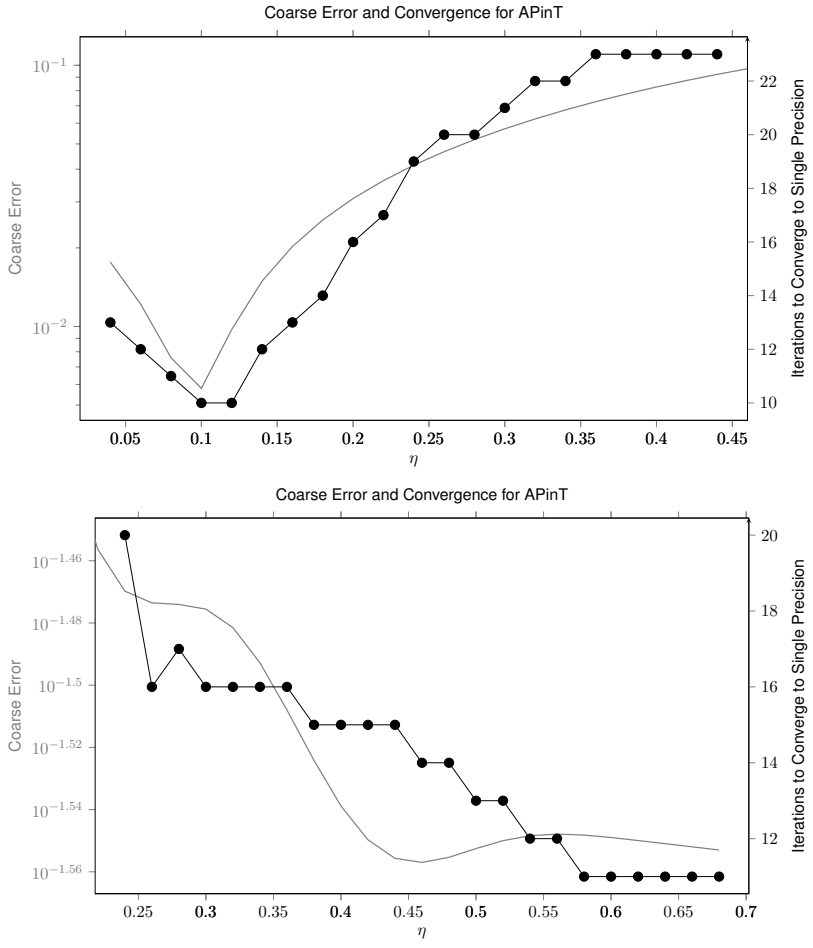

Figure 5.4: The relationship between coarse error and the number of iterations necessary for convergence is shown. The rapidly-oscillating case of $\varepsilon = 0.01$ is shown on the bottom, with the non-stiff case of $\varepsilon = 1.0$ on the top. The L_2 coarse error is shown in grey corresponding to the logarithmic scale on the left y-axis. The total number of iterations for convergence to single precision is shown with the black circles, corresponding to the linear scale on the right. This latter quantity is necessarily discrete, while the iterative error is a continuous quantity measured at discrete points.

Figure 5.4 shows the relationship between the error and the total number of iterations for convergence as a function of the length of the averaging window, η, and when $\Delta T = 0.1$. The total number of coarse timesteps taken parallel-in-time was 50. Both the non-stiff case where $\varepsilon = 1.0$ and the stiff case where $\varepsilon = 0.01$ are shown, with the L_2 coarse error before any Parareal iterations and the number of iterations for convergence to single precision shown on the same

plot. The coarse error is the L_2-norm of the difference between the coarse and fine solutions after the initial coarse sweep as in the numerical experiments in Chapter 4.

There are two main points here. Firstly, the coarse error is a good predictor of the convergence rate. This holds true both when the APinT algorithm converges very quickly and one would expect that the numerical behaviour is well-approximated, but also for highly non-optimal averaging windows.

The second important point which follows on the first is that the convergence properties of the APinT algorithm follow the same pattern as the coarse propagator. Specifically, we note that in the small-ε calculation, optimal convergence was obtained for longer averaging windows which filter all but the direct resonances. Once this level of convergence has been obtained, increasing the length of the averaging window does not provide further gains.

When $\varepsilon = 1.0$ the expected behaviour is seen with the clear presence of the 'Goldilocks point' in both the computed error and the required number of iterations. The averaging window which minimises the iteration count corresponds to where the coarse error is minimised (Theorem 4.4).

To further justify the improvement in simulation provided by Parareal, we present three comparisons of optimally-averaged APinT with vanilla Parareal, at $\varepsilon = 0.01$, 0.1, and 1.0. By 'vanilla Parareal', we mean a Parareal simulation where the coarse timestepping method is exactly the same as the fine, save for a longer timestep – no averaging is applied.

In Figures 5.5, 5.6, and 5.7, we show the iterative error of both methods as a function of iteration level, k. The convergence of APinT is bracketed by that of two vanilla Parareal runs so that the improvement in timestep may be directly compared to the existing state of the art.

In all cases, the total simulation time was $T_{\text{final}} = 5.0$. The intention is to compare the algorithms when acting on problems of similar spatio-temporal size irrespective of scale separation, as would often be the case in practice. A similar study which considers final times on the order of ε is provided by Haut and Wingate (2014).

Figure 5.5 shows the highly stiff regime with $\varepsilon = 0.01$. This is the type of problem which APinT was developed to handle, and so it is not surprising that in terms of accuracy, APinT performs as well as vanilla Parareal with between five and ten times the coarse timestep. In real terms, this corresponds to a parallel speedup of just slightly below five to ten times.

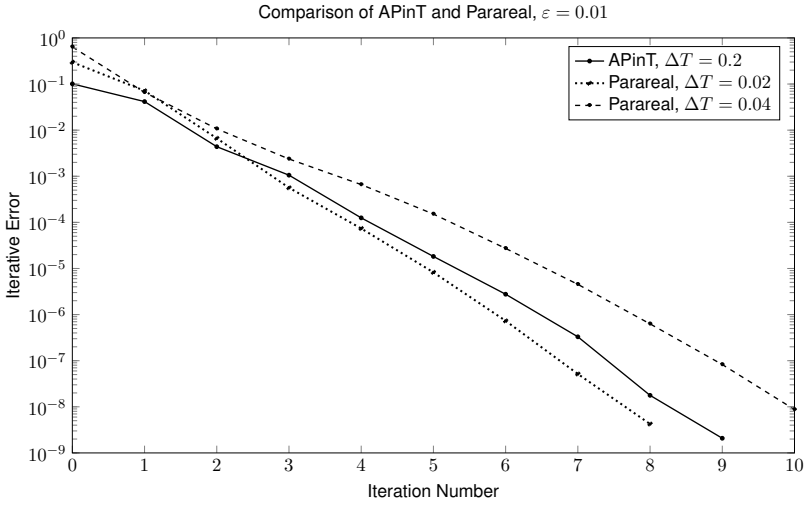

Figure 5.5: A comparison of the convergence of APinT with vanilla Parareal as a function of iteration number for the very stiff case. The convergence rates of two Parareal runs are shown which bracket a given APinT run. Note that APinT provides similar performance to the Parareal algorithms while taking a timestep between five and ten times longer.

The major novelty of this section is that the averaging window may be optimised to make APinT an effective method for larger values of ε. Recalling Chapter 3, the infinitely long averaging window only applies in the case of infinite scale separation. To a numerical approximation, Figure 5.5 showed this case.

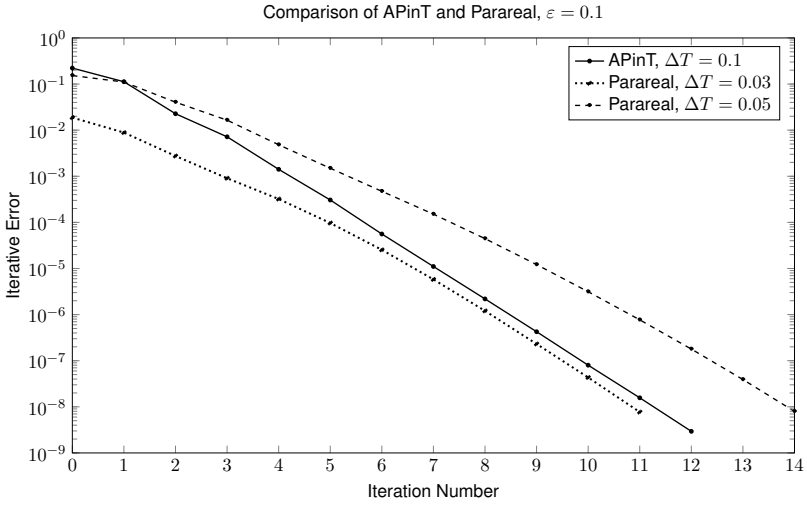

Figure 5.6: A comparison of the convergence of APinT with vanilla Parareal as a function of iteration number for the moderately stiff case. As with Figure 5.5 the convergence of APinT is bracketed by two Parareal runs. An improvement is still visible here with APinT allowing a coarse timestep of between two and three times larger for similar performance.

By choosing an optimal averaging window as discussed above in this chapter, we are able to obtain an improvement on vanilla Parareal in cases of small or no scale separation. As $\varepsilon \to 1$ the APinT method ceases to provide a significant advantage over vanilla Parareal. Importantly, however, the APinT algorithm easily adjusts to changes in scale separation and so there is no requirement to change algorithms when outside the QG limit. Figures 5.6 and 5.7 show the performance of APinT relative to Parareal for $\varepsilon = 0.1$ and 1.0, respectively.

We see that APinT provides an improvement over vanilla Parareal for intermediate scale separation with $\varepsilon = 0.1$, where there is still

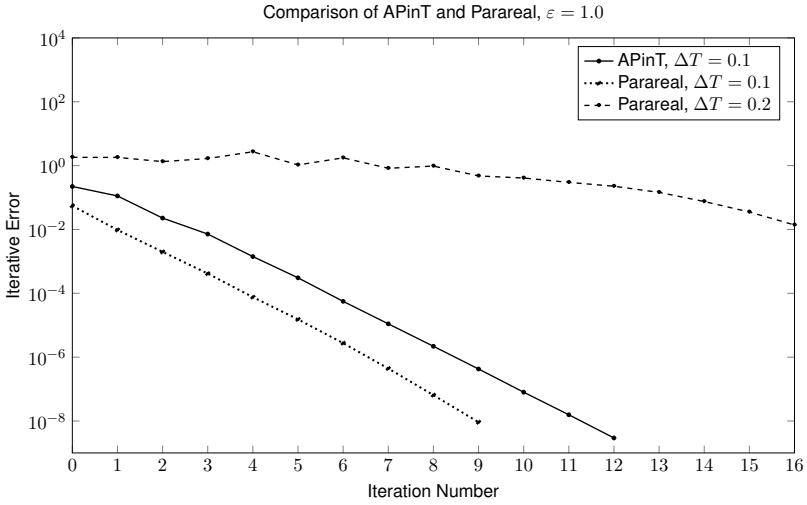

Figure 5.7: A comparison of the convergence of APinT with vanilla Parareal as a function of iteration number for the non-stiff case. As with Figures 5.5 and 5.6 the convergence of APinT is bracketed by two Parareal runs. We see that APinT remains viable outside of the limit of small ε, thus obviating any need for multiple timesteppers for different flow regimes.

some degree of oscillatory stiffness present. For $\varepsilon = 1.0$, the APinT method performs approximately as well the vanilla Parareal. While it may appear that this is a problematic result, it should be noted that this is the non-stiff case. For this degree of scale separation wave averaging is not necessary as Parareal methods already provide good speedup over time-serial methods.

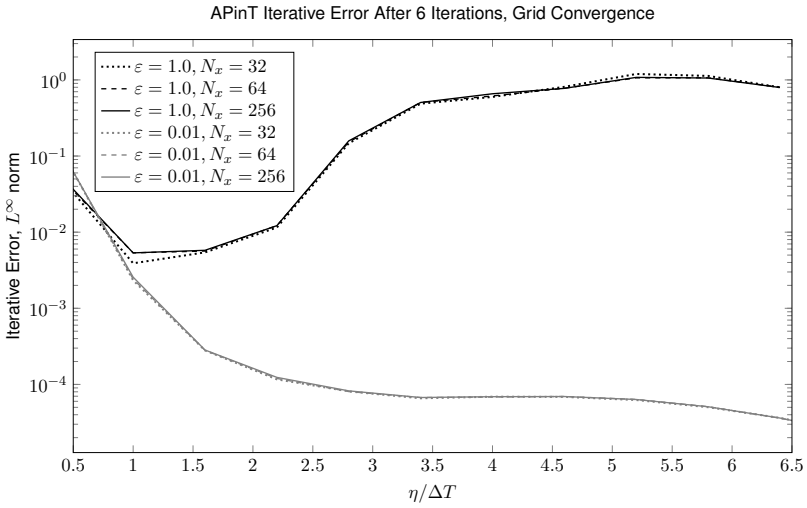

Figure 5.8: The iterative error after six iterations is shown versus the averaging window for two degrees of scale separation ($\varepsilon = 0.01$ and $\varepsilon = 1.0$) and for three grid resolutions ($N_x = 32$, 64, and 256). All other parameters are identical across all runs. As expected by Theorems 5.1 and 5.2 the convergence to the fine solution in independent of grid resolution.

The effect of grid refinement on APinT convergence is shown in Figure 5.8. Considering three different grid resolutions we find that there is no effect of grid refinement on APinT convergence. There is a mild caveat here, which is that Parareal methods converge to the fine solution and it is this convergence which we are measuring. There is therefore a different discretisation error in each run, one which we are implicitly discarding.

It is not terribly surprising that the grid resolution should have no effect on the convergence. Considering Theorem 5.2, the effects of grid refinement present themselves only through the stiffness

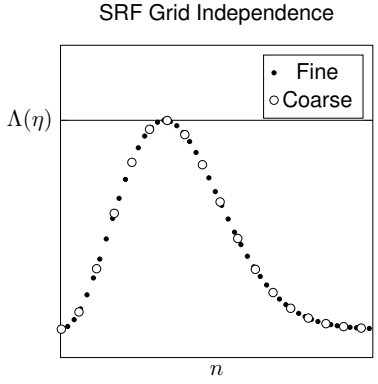

Figure 5.9: A schematic of the stiffness regulator function for a coarse and fine grid. The underlying curve is better resolved for a finer grid but $\Lambda(\eta)$ is the maximum value, which is not affected by grid refinement.

regulator function, $\Lambda(\eta)$, and they do so there only through the larger set of triads which is available at higher resolutions. Recall the analysis of Section 4.5.1 and in particular the shape of the maximand of $\Lambda(\eta)$, shown schematically in Figure 5.9.

The error in timestepping is bounded by the maximum achieved after averaging over all triads. When working with a finite grid, the discrete set of triads leads to a discrete approximation of the maximand of the stiffness regulator function. Refining the grid leads to a consideration of more triads, an ordered set of which generate the curve in Figure 5.9, thereby improving its resolution but not affecting the magnitude of its maximum. It is this maximum which is our stiffness regulator function and which bounds the coarse error and by extension the Parareal error.

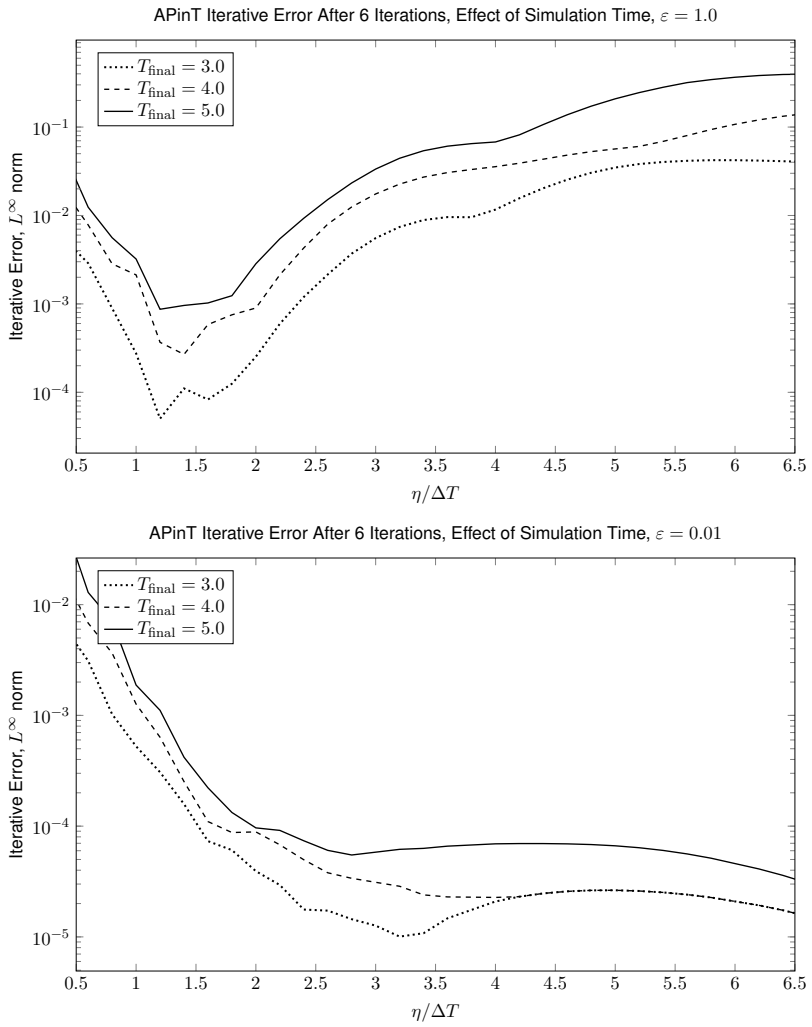

Figure 5.10: The effect that the total simulation time has on APinT is shown for a non-stiff (top) and stiff (bottom) case. As in Figure 5.8 the iterative error is shown after six iterations as a function of the averaging window length. The familiar trends are present across a range of simulation times. Note that increasing the simulation time hinders convergence, but does not modify the optimal averaging.

The final parameter study which is interesting in the one-dimensional case is that of the total simulation time (i.e. the endpoint of the simulation, T_{final}). Figure 5.10 presents the effect of averaging on APinT convergence for two values of the scale separation, ε, and three total simulation times. The coarse and fine timesteps were kept the same in all simulations. We see that the behaviour of the averaging is un-

affected by the length of the simulation in terms of the location of the optimal.

A longer simulation time leads to higher error after six iterations, and therefore more iterations required for convergence. This is caused by the global coarse error increasing as the total simulation time increases (Kincaid and Cheney, 1991; Trefethen, 1996). Recalling that the convergence is towards the fine solution, there is an interesting implication here. Specifically, the global coarse error increases with T_{final} more rapidly than the global fine error does.

Again, this result is hardly surprising. The global error committed by the coarse solver is subject to both timestepping and averaging effects. That averaging error increases with both the degree of simulation time and the degree of averaging was shown in Lemma 4.2, although we have bundled the former into the constant for clarity in our study of triadic interaction. What we see in Figure 5.10 is the total effect of the increase in global timestepping error and averaging error.

5.5 *Decaying Shallow Water Turbulence*

The 1-D RSWE provide a useful test bed for parameter studies on the APinT algorithm due to their low computational cost. In order to have an algorithm of practical use, however, we must be able to handle more physical problems. We then turn to the 2-D equations and consider a more challenging problem.

Polvani et al. (1994), extending the work of Farge and Sadourny (1989) and Spall and McWilliams (1992), numerically investigated the dynamics of initially balanced decaying turbulence in the rotating shallow water equations. They found that, as with incompressible 2-D rotating turbulence, coherent vortex structures develop from the initially random flow field. Following their balanced initialisation, we have performed an APinT simulation over 1000 eddy turnover times of the flow.[3]

[3] The numerical initial conditions were provided by Beth Wingate and Mark Taylor and are not the work of the author.

The case we have used corresponds to case E in Polvani et al. (1994), which has a Rossby number of 0.25 and a Froude number of 0.2. In the nondimensional framework, this is equivalent to $\varepsilon = 0.25$ and $F^{1/2} = 0.8$.

If APinT is to be viable in practice, it is necessary that it be able to successfully model the physical characteristics of the time dynamics of the flow. In particular, it is necessary that the error induced by APinT should not prevent the emergence of coherent vortex structures. It is known (Gander, 2015) that Parareal methods

always converge to the fine solution if they converge. This section is then a numerical study of that fact.

Simulations were performed on a regular grid of size $N_x = 128^2$ with periodic boundary conditions. Hyperviscosity was applied on both the coarse and fine timesteps, as was 3/2 dealiasing by padding.[4] A coarse timestep of $\Delta T = t_e/10$ was taken, where t_e is an eddy turnover time. The fine timestep was $\Delta t = \Delta T/1000$. Only an optimal averaging window was considered, and convergence to single precision in the L_∞ norm was obtained in 6 iterations.

[4] Some authors would write this as being the 2/3-rule on a 192^2 grid.

It is not feasible to perform a computation of this size over the entire time domain simultaneously. Instead, for reasons of both storage and accuracy (*cf.* Figure 5.10) APinT was applied over blocks of size $t_{\text{block}} = 200t_e$ which were computed back-to-back as in Figure 5.11. The full simulation then consists of a series of APinT computations computed sequentially.

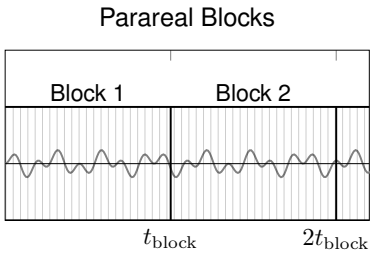

Figure 5.11: A Parareal method implemented in blocks of time. Block 1 runs parallel in time until it has converged. Then, its solution at its right endpoint is used as the initial condition for Block 2. The entire time domain is covered with blocks which are computed in series until the final time is reached.

Figure 5.12 shows three snapshots of the vorticity field for an optimally-averaged APinT simulation with convergence to single precision. We see that the initially-random flow field has given rise to coherent vortex structures. From this we deduce that the coarse timestepping method proposed is sufficiently capable of resolving the physics of the problem for convergence to the true solution.

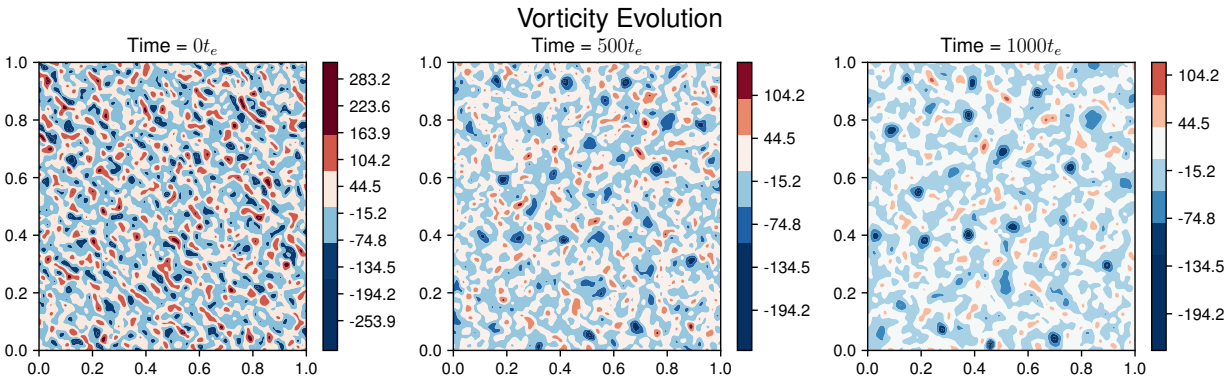

Figure 5.12: The vorticity field at three points in time is shown for decaying shallow water turbulence as computed by APinT.

It is worth comparing the APinT-simulated solution to the true solution. Figure 5.13 shows the absolute error in the vorticity field for both a timestep halfway across a block and at the end, and for three iterations. Iteration 0 is the first coarse solve, before any Parareal corrections have been applied. We see here that the averaged solution is not qualitatively different from the fine solution, obtained by Strang splitting in a separate computation.

This also makes clear another useful fact about Parareal algorithms in general: convergence is more rapid closer to the initial condition. Noting the exponents on the colourbars, we see that the error after 100 coarse timesteps at any given iteration level varies

from being between $2\times$ lower to up to almost an order of magnitude lower. Knowledge of this fact helps in designing more optimal blocking algorithms than the naive approach taken in this work.

Key Points

- The Parareal algorithm extends parallelism to the temporal domain, but suffers from limitations with oscillatory problems.
- The APinT method applies the coarse solver of Chapter 4 to enable Parareal convergence for oscillatory problems.
- The optimal averaging window for the coarse solver may be chosen numerically.
- APinT convergence is largely independent of spatial resolution.
- Both the coarse timestep size and the simulation length affect the convergence of Parareal methods.

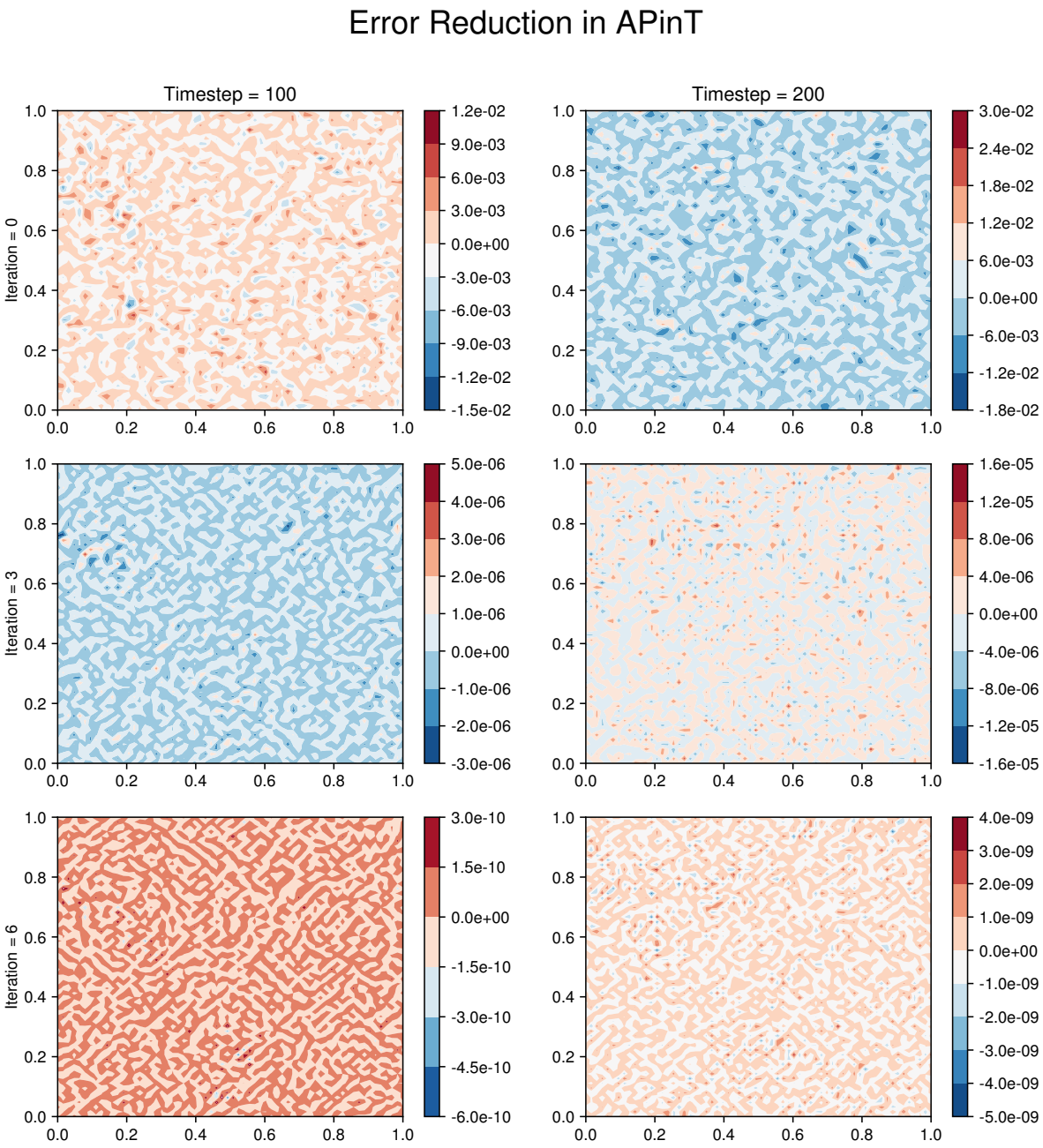

Figure 5.13: The convergence of APinT to the fine solution is shown in terms of absolute error. The first column shows the error in the vorticity field at the halfway point of the computation, while the second shows it at the final timestep. The three rows show the error at iteration 0 (i.e. the first coarse approximation) and two other iterations. Pay close attention to the exponents on the colourmap scale.

6 Conclusion and Future Work

> Youre tale anoyeth al this compaignye.
> Swich talkyng is nat worth a boterflye.
>
> Geoffrey Chaucer, The Canterbury Tales

Taking the asymptotic understanding of highly oscillatory PDEs as a starting point, it is possible to construct a numerical method based on fast-wave averaging. This provides sufficient accuracy to enable more advanced numerical methods for finite scale separation while permitting a longer timestep. This method permits accurate and stable numerical solution of highly-oscillatory PDEs with $\mathcal{O}(1)$ timesteps as $\varepsilon \to 0$, which stands in stark contrast to methods which are not designed for this purpose. As an extension to the work of Haut and Wingate (2014), we have shown that the averaging may be chosen to ensure Parareal convergence outside of the small-ε limit.

A particular application of this solver has been presented here in the form of the Asymptotic Parallel in Time method. The increased timestep[1] and therefore reduced computational time is more important than the increased coarse error when compared to a fine timestepping method which does not employ fast-wave averaging. This is because the Parareal method expects that a coarse error will be committed and then refines it to the fine solution in a time-parallel fashion, leading to a speedup in achieving a comparable solution.

[1] In fact, the allowable timestep is significantly longer than the nonlinear timestep limit as $\varepsilon \to 0$.

While it is this method which is of interest for practical simulation of geophysical flows and therefore operational weather forecasting and climate modelling, the result of the greatest mathematical significance shown here is the existence of an optimal choice of averaging window. By describing the solution in terms of the discrete components of nonlinear oscillation, called *triads*, we have shown that the error incurred by using such a solver is a trade-off between averaging and timestepping errors and that the averaging procedure provides a method to control the most rapid oscillations in the flow while allowing the expression of the low-

frequency dynamics. In doing so, we have implicitly provided new numerical analytic evidence in support of previous results (Newell, 1969; Smith and Lee, 2005) which suggested that near-resonant triads play a vital role in the time evolution of the solution.

The error bounds found in this work improve our understanding of the numerical simulation of nonlinear dynamics for systems of the type studied here. It further leads to a practical and easily-implemented optimisation of the coarse timestepping method which is applicable to both of the practical algorithms studied.

6.1 Extension to Three Scales

In this work we have assumed that the gravitational and rotational effects were roughly on the same scale, i.e. $F = \mathcal{O}(1)$, or $Ro \approx Fr$, where Ro is the Rossby number and Fr is the Froude number. For realistic geophysical flows, this may not be the case. Consider the coarse error versus the length of the averaging window for three values of F, shown in Figure 6.1. Recall that $\varepsilon = Ro = F^{-1/2}Fr$.

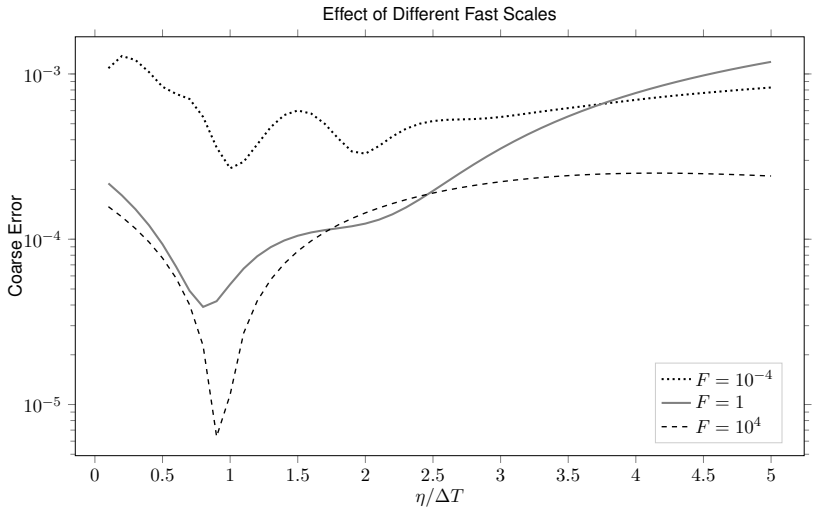

Figure 6.1: The effect of two fast time scales on the coarse error. $\varepsilon = 1.0$ in all three runs. Note that where $F = 10^{-4}$, two distinct optima are visible, corresponding to two relevant timescales.

The case where $F = 1$ is the familiar case which we have studied throughout this work. Taking $F = 10^{-4}$, however, leads to the situation where gravitational effects are two orders of magnitude larger than rotational effects. In this case we see the presence of two much shallower minima in the error curve. It is suspected that one each of these corresponds to the gravitational and the rotational waves being respectively optimally averaged.

Taking the other direction and considering $F = 10^4$ gives rotational effects which are two orders of magnitude larger than the gravitational effects. Here we see that the coarse error improves by

an order of magnitude. Surprisingly, we do not see two minima. In fact, the slight trend towards a second one which is visible in the $F = 1$ curve has vanished here. This phenomenon is unexplained, although it is fair to say that in both of these situations, there are three separated scales, two of which are to some extent fast, as opposed to the fast-slow separation which we have considered so far.

Whitehead et al. (2014), considering the rotating stratified Boussinesq equations in the same operator formulation as we have in this work, wrote their system in terms of two linear operators. Following their work we rewrite our governing equation (1.4) in the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{Ro}\mathcal{R}\mathbf{u} + \frac{1}{Fr}\mathcal{G}\mathbf{u} + \mathcal{N}(\mathbf{u},\mathbf{u}) = \mathcal{D}\mathbf{u}, \tag{6.1}$$

where the nondimensional linear rotational operator is

$$\mathcal{R}\mathbf{u} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u(\mathbf{x},t) \\ v(\mathbf{x},t) \\ \eta^*(\mathbf{x},t) \end{bmatrix}, \tag{6.2}$$

and the nondimensional linear gravitational operator is

$$\mathcal{G}\mathbf{u} = \begin{bmatrix} 0 & 0 & \partial_x \\ 0 & 0 & \partial_y \\ \partial_x & \partial_y & 0 \end{bmatrix} \begin{bmatrix} u(\mathbf{x},t) \\ v(\mathbf{x},t) \\ \eta^*(\mathbf{x},t) \end{bmatrix}. \tag{6.3}$$

We may immediately regain the linear operator used previously in this work as

$$\frac{1}{\varepsilon}\mathcal{L} = \frac{1}{Ro}\mathcal{R} + \frac{1}{Fr}\mathcal{G} \tag{6.4}$$

by using the fact that $\varepsilon = Ro = F^{-1/2}Fr$. Using the expanded form (6.1) makes the presence of three scales explicit and shows the need to take an average over both fast timescales.

Based on the intuition developed in this work and the numerical study shown in Figure 6.1, we expect that there is a method of optimally averaging in this case as well. Whitehead et al. (2014) developed an asymptotic slow solution in the limit where both of the fast scales are infinitely fast and separated from one another but it did not yield convergent numerical results when both scales were finite.

If a method of fast-wave averaging which is suited to three scales may be found, it will enable the APinT algorithm presented in Chapter 5 for more realistic flow conditions. We then leave the reader with the following conjecture.

Conjecture 6.1. *Let* $\mathbf{u} \in L^2$ *be the solution to the following initial-boundary value problem:*

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{Ro}\mathcal{R}\mathbf{u} + \frac{1}{Fr}\mathcal{G}\mathbf{u} + \mathcal{N}(\mathbf{u},\mathbf{u}) = \mathcal{D}\mathbf{u}, \tag{6.5}$$

where $\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0$ *and where the solution on the boundary is known. There exists some averaged numerical solution,* $\overline{\mathbf{u}}(\mathbf{x}, t) = \overline{\varphi}_{\Delta T, \eta_{\mathcal{R}}, \eta_{\mathcal{G}}}(\mathbf{u}(\mathbf{x}, t))$ *where the averaging method is not specified and where* $\overline{\mathbf{u}}$ *varies slowly in time.*

For $Ro \neq Fr$ *and for all* $Ro, Fr \in (0, 1]$*, there exists an optimal choice of the averaging windows,* $\eta_{\mathcal{R}}$ *and* $\eta_{\mathcal{G}}$ *such that* $\min_{\eta_{\mathcal{R}}, \eta_{\mathcal{G}}} \|\overline{\mathbf{u}} - \mathbf{u}\|$ *exists and for any given coarse timestep,* ΔT*,*

$$0 < \eta^*_{\mathcal{R}}(Ro), \eta^*_{\mathcal{G}}(Fr) < \infty, \tag{6.6}$$

where the asterisk denotes the optimal choice of the averaging window.

▲

A Pseudospectral Methods

> There are three things, young gentlemen, which you are constantly to bear in mind. Firstly, you must always implicitly obey orders, without attempting to form any opinion of your own respecting their propriety. Secondly, you must consider every man your enemy who speaks ill of your king; and thirdly, you must hate a Frenchman, as you do the devil.
>
> Vice Admiral Horatio Nelson

When solving PDEs numerically it is necessary to develop an appropriate discretisation of the system. Spectral methods use global representations of functions, as opposed to finite difference or finite element methods, which employ local representations. This allows them to significantly outperform these methods as long as the circumstances of the problem are suited to the particular spectral approximation. Let us assume that the unknown solution to our PDE of interest, $u(x)$, can be approximated by a series of $N+1$ *basis functions*, $\phi_k(x)$,

$$u(x) \approx u_N(x) = \sum_{k=0}^{N} a_k \phi_k(x). \tag{A.1}$$

Following Boyd (2000), we substitute this into the equation

$$Lu = f(x) \tag{A.2}$$

where L is the operator of the differential equation. The result is the *residual function*, defined as

$$R(x; a_0, a_1, \ldots, a_N) = Lu - f, \tag{A.3}$$

which equals zero for the exact solution and measures the error induced by the series approximation. Spectral methods take the basis functions, $\phi_k(x)$, to be global functions of high order which are non-zero except for at isolated points. This is distinct to, for example, finite element methods which take the basis functions to

be local polynomials of fixed degree. While there are several valid series expansions, we will choose the Fourier series here based on its widespread use in fluid dynamics (Canuto et al., 1988; Boyd, 2000). The truncated Fourier series with N coefficients also yields exponential convergence on a periodic spatial domain, which is to say that the residual goes to zero faster than any power of $1/N$ as $N \to \infty$ (Tadmor, 1986).

There is a distinction to be made between 'interpolating' (also known as 'collocation' or 'pseudospectral') and 'non-interpolating' (including Galerkin and tau-style) implementations. Due to its widespread use in fluid mechanics and its implementational simplicity, we shall restrict ourselves here to pseudospectral methods. In a pseudospectral method, the unknown is considered in physical space on a discrete grid of points. The coefficients of the series expansion are found by requiring that it agree with the known function at all points on the grid. In practice, this requirement is enforced through the *Fourier transform* (*q.v.* Definition A.1).

A.1 The Fourier Series

WITHOUT FURTHER ADO, let us assume that our basis functions are trignonometric, leading to the *Fourier basis*, i.e.

$$\phi_k(x) = e^{ikx}, \tag{A.4}$$

where x is the spatial coordinate and $k \in \mathbb{Z}$ are the wavenumbers familiar from Chapter 3. Using Fourier basis functions as we are doing here requires that the domain be either periodic or infinite. In the case of numerical experiments as well as some practical problems (such as flow around the Earth) it is not problematic to assume spatial periodicity. Without loss of generality, we will assume the spatial domain to be 2π-periodic, i.e. $u(x) = u(x + 2\pi)$, since it is straightforward to rescale the spatial domain such that this is true. The use of Fourier basis functions with the series approximation given in equation (A.1) leads to the *Fourier Series*.

Definition A.1 (Fourier Series)**.** Let $u(x) \in L^2$ be some function $u : [0, 2\pi] \to \mathbb{R}$. The *Fourier series* of u is

$$u(x) = \sum_{k=-\infty}^{\infty} \hat{u}_k e^{ikx}, \tag{A.5}$$

where $n \in \mathbb{Z}$ and where the Fourier coefficients are

$$\hat{u}_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ikx} \, \mathrm{d}x. \tag{A.6}$$

We will refer to the function $\mathcal{F} : L^2[0, 2\pi] \to \mathbb{C}, \; u \mapsto \hat{u}$ as the *Fourier transform*. It has a well-defined inverse, written $\mathcal{F}^{-1}$. ▲

An important property of the Fourier series and one which we have applied several times in this work is that it has an orthogonal basis.

Definition A.2 (Orthogonality)**.** A set of basis functions, $\phi_k(x)$, is said to be orthogonal with respect to a given inner product if

$$\langle \phi_m, \phi_n \rangle = \delta_{nm} \nu_n^2, \tag{A.7}$$

where δ_{nm} is the Kronecker delta function and ν_n is some constant. ▲

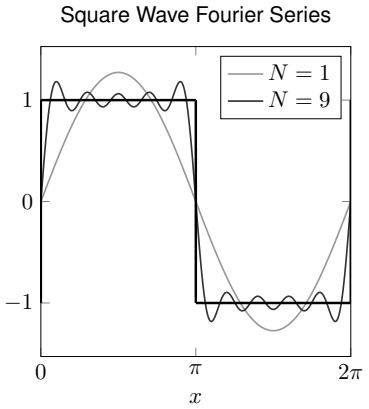

Figure A.1: Two truncated Fourier series for the square wave. Approximations to the depicted square wave are shown for two resolutions of the Fourier series. In the $N = 1$ case, the wave is poorly approximated. Including more basis functions (in some sense only four more, as even-numbered modes are identically zero in this particular case) improves the approximation.

Theorem A.1 (Orthogonality of Fourier Basis Functions)**.** *The Fourier basis functions, $\phi_{n,}(x)$, defined in Definition A.1 are orthogonal.* ♦

Proof. Consider the case where $m = n$. Then we must evaluate the complex inner product:

$$\begin{aligned}
\langle \phi_m(x), \phi_n(x) \rangle &= \int_0^{2\pi} e^{imx} \overline{e^{imx}} \, \mathrm{d}x, \\
&= \int_0^{2\pi} e^0 \, \mathrm{d}x, \\
&= 2\pi.
\end{aligned}$$

Now consider the case where $m \neq n$. Then,

$$\begin{aligned}
\langle \phi_m(x), \phi_n(x) \rangle &= \int_0^{2\pi} e^{imx} \overline{e^{inx}} \, \mathrm{d}x, \\
&= \int_0^{2\pi} e^{i(m-n)x} \, \mathrm{d}x, \quad m - n \neq 0 \\
&= 0,
\end{aligned}$$

where the last line comes about because the integral of any trigonometric function over an integer number of periods is zero. □

In applying a Fourier spectral method, we assume that our *spatial* domain is periodic. For initial-value problems, it is not generally reasonable to assume that the time domain is periodic and apply basis functions in time. Rather, we will allow the Fourier coefficients themselves to vary in time, leading to a Fourier series which takes the form:

$$u(x,t) = \sum_{k=-\infty}^{\infty} \hat{u}_n(t) e^{ikx}. \tag{A.8}$$

A.2 The Discrete Fourier Transform

In practice, we employ the *discrete Fourier transform*, or *DFT* (Canuto et al., 1988). For any integer $N > 0$, consider the following set of grid points

$$x_j = \frac{2\pi j}{N}, \quad j = 0, 1, \ldots, N-1. \tag{A.9}$$

We may then define the discrete Fourier coefficients, denoted $\tilde{u}_k$, of some function, $u(x)$ as

$$\tilde{u}_k = \frac{1}{N} \sum_{j=0}^{N-1} u(x_j) e^{-ikx_j}, \quad -N/2 \leq k \leq N/2 - 1. \tag{A.10}$$

Note that the various Fourier modes are orthogonal to one another, as in the case of the continuous Fourier transform, and so the *inverse discrete Fourier transform* is

$$u(x_j) = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k e^{ikx_j}, \quad j = 0, 1, \ldots, N-1. \tag{A.11}$$

Equation (A.11) is the truncated discrete Fourier representation which we use in practice. Following Canuto et al. (1988) we may consider the polynomial

$$I_N u(x) = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k e^{ikx} \tag{A.12}$$

to be the $N/2$-degree trigonometric interpolant of $u(x)$ at the nodes given by equation (A.9). This interpolation operator may be regarded as an orthogonal projection upon the space of trigonometric polynomials of degree $N/2$ with respect to a discrete approximation of the complex inner product.

From this point on, we will drop the tildes to denote a discrete Fourier representation, and use the more familiar hat notation. We will also use the discrete, truncated Fourier series for the remainder of this section.

A.3 Spectral Differentiation

In the preceding algorithm, we made reference to *spectral differentiation*. Consider the first spatial partial derivative of the Fourier series for a given function,

$$\frac{\partial}{\partial x} u(x,t) = \sum_{k=-\infty}^{\infty} \frac{\partial}{\partial x} \hat{u}_k(t) e^{ikx}. \tag{A.13}$$

Since the complex exponentials provide an orthogonal basis and the Fourier coefficients depend only on time, we may write this as

$$\frac{\partial}{\partial x}u(x,t) = \sum_{k=-\infty}^{\infty} ik\hat{u}_k(t)e^{ikx}. \tag{A.14}$$

Computing spatial derivatives is then both numerically straightforward and analytical for the Fourier series. In general, the j-th partial derivative in space for the discrete Fourier representation is

$$D_N^j = \sum_{k=-N/2}^{N/2-1} (ik)^j \hat{u}_k(t)e^{ikx}. \tag{A.15}$$

A.4 Spectrally Solving Linear PDEs

In practice, we apply the Fourier transform to the initial conditions to determine the Fourier coefficients, $\hat{u}_k(0)$. Timestepping is then a matter of integrating in time to predict the Fourier coefficients for a given basis. Consider some PDE,

$$\frac{\partial u}{\partial t} = Du, \quad u(x, t=0) = u_0, \tag{A.16}$$

where D encodes some arbitrary combination of partial derivatives in space only. Let us assume a discrete timestep of Δt, and denote a particular timestep with n, such that

$$\hat{u}_k(t) = \hat{u}_k(n\Delta t) \equiv \hat{u}_k^n. \tag{A.17}$$

The spectral algorithm for explicit Euler (with other timestepping methods generalising from this) is then given in Algorithm A.5. Note that in practice we must use a truncated Fourier series which has finite limits.

Algorithm A.5: The pseudospectral implementation of the explicit Euler method.

$\hat{u}_k^0 \leftarrow \mathcal{F}(u(x, t=0))$ ▷ Fourier transform by (A.6)
for $n < n_{\text{final}}$ **do**
  $\hat{u}_k'^n \leftarrow D_N \hat{u}_k^n$ ▷ Spectral Differentiation
  $\hat{u}_k^{n+1} \leftarrow \hat{u}_k^n + \Delta t \hat{u}_k'^n$
end for

A.5 Nonlinear Terms

Computing nonlinear terms requires some more careful attention than nonlinear terms in a pseudospectral framework. Consider some nonlinear term, which will shall write here without loss of generality as a general quadratic non-linearity,

$$w(x) = u(x)v(x). \tag{A.18}$$

When solving linear equations we will generally perform a single DFT to set up the initial condition and remain in Fourier space from there on. To perform a nonlinear operation as in equation (A.18) entirely in the Fourier domain requires the use of a convolution sum, which takes the form

$$\hat{w}_k = \sum_{\substack{m+n=k \\ |m|,|n|\leq N/2}} \hat{u}_m \hat{v}_n. \tag{A.19}$$

This is prohibitively expensive, however, requiring $\mathcal{O}(N^2)$ operations. We may reduce this complexity to $\mathcal{O}(N \log N)$ by computing the nonlinearity in a pseudospectral fashion. This method proceeds by performing three DFTs – two inverse and one forward – and one multiplication in real space. Applying equations (A.10) and (A.11), we may write

$$U_j = \sum_{k=-N/2}^{N/2-1} \hat{u}_k e^{ikx_j}, \quad j = 0, 1, \ldots, N-1, \tag{A.20}$$

and

$$V_j = \sum_{k=-N/2}^{N/2-1} \hat{v}_k e^{ikx_j}, \quad j = 0, 1, \ldots, N-1, \tag{A.21}$$

where x_j are the grid points. We then perform the multiplication in real space, i.e.

$$W_j = U_j V_j, \quad j = 0, 1, \ldots, N-1. \tag{A.22}$$

We may then compute the desired quantity, $\hat{W}_k$, by the DFT

$$\hat{W}_k = \frac{1}{N} \sum_{j=0}^{N-1} W_j e^{-ikx_j}, \quad -N/2 \leq k < N/2. \tag{A.23}$$

By orthogonality we find

$$\hat{W}_k = \sum_{m+n=k} \hat{u}_m \hat{v}_n + \sum_{m+n=k\pm N} \hat{u}_m \hat{v}_n, \tag{A.24}$$

which implies

$$\hat{W}_k = \hat{w}_k + \sum_{m+n=k\pm N} \hat{u}_m \hat{v}_n. \tag{A.25}$$

The second term here commits an error, called the *aliasing error*.

A.5.1 Aliasing

ALIASING WAS FIRST NOTICED by Phillips (1956), working on a model of general atmospheric circulation. In the case of this model, it led to supersonic winds and a subsequent blowing-up of the model. Phillips (1959) provided the explanation that the root of this instability was in the appearance of so-called $2h$-waves.

The hydrodynamic equations being solved, such as the rotating shallow water equations, are quadratically nonlinear. This means that the nonlinear interaction of two waves with wavenumbers $|k| > N/2$ will generate a new wave with wavenumber $|k| > N$. Aliasing arises because the grid is incapable of resolving these waves, and instead 'aliases' these high frequency waves to lower frequencies, resulting in a spurious transfer of energy to lower frequencies which breaks conservation of energy and leads to numerical instability.

Figure A.2 shows this aliasing effect for a simple case, whereby three different waves all have the same $k = -2$ interpolation on an 8-point grid despite only one of them having wavenumber $k = -2$.

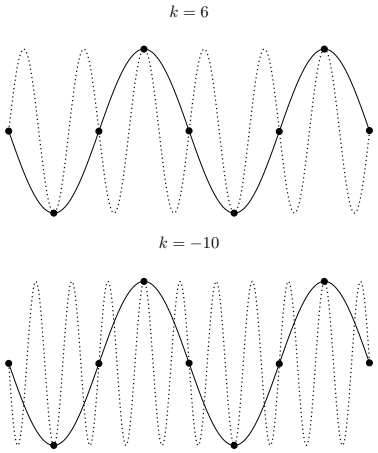

Figure A.2: Demonstration of two higher-frequency modes aliasing onto $y = \sin(-2x)$.

To illustrate this effect more rigorously, we may write the discrete Fourier coefficients in terms of the exact Fourier coefficients. If Lu converges to u at every node (A.9) then by the definition of the DFT (A.10)

$$\tilde{u}_k = \hat{u}_k + \sum_{\substack{m=-\infty \\ m\neq 0}}^{+\infty} \hat{u}_{k+Nm}, \quad -N/2 \leq k \leq N/2 - 1. \tag{A.26}$$

Thus, the k-th mode of the trigonometric interpolant of u depends not only on the k-th mode of u, but in fact on all $(k + Nm)$-th modes. These modes are said to alias the k-th mode on a discrete grid. This happens because for some periodic basis function, ϕ

$$\phi_{k+Nm}(x_j) = \phi_k(x_j). \tag{A.27}$$

Equation (A.27) then directly implies

$$I_N u = P_N u + R_n u, \tag{A.28}$$

where we write

$$R_N u = \sum_{k=-N/2}^{N/2-1} \left(\sum_{\substack{m=-\infty \\ m\neq 0}}^{+\infty} \hat{u}_{k+Nm} \right) \phi_k. \tag{A.29}$$

This error, R_N, between the interpolation polynomial and the truncated Fourier series is the *aliasing error*. It is guaranteed orthogonal to the truncation error, $u - P_N u$, such that

$$\|u - I_N u\|^2 = \|u - P_N u\|^2 + \|R_N u\|^2. \tag{A.30}$$

Thus, the error due to interpolation is always larger than the truncation error would suggest due to aliasing. It has been shown that the aliasing error is asymptotically of the same order as the truncation error by Kreiss and Oliger (1979).

A.5.2 Dealiasing by Padding or Truncation

Recall that the instability induced in the GCM model studied by Phillips (1956) arose in the form of $2h$-waves, where $2h$ denotes the wavelength. For this reason, (Phillips, 1959) originally suggested filtering the spectrum every few timesteps by setting the upper half of the wavenumbers to zero at regular time intervals in the computation, i.e. they defined some filter such that

$$F : \hat{u}_k \mapsto \begin{cases} \hat{u}_k & |k| \leq |k|_{max}/2, \\ 0 & \text{otherwise.} \end{cases} \tag{A.31}$$

According to Orszag (1971), it is not necessary to remove half the spectrum, which suppresses all waves with wavelengths $2h < k < 4h$. Rather, if only the highest third of the wavenumbers are filtered out (those for which $|k| > 2N/3$) aliasing will still occur but only in a region which is itself purged by the filtering.

We may thus *dealias* equation (A.25) by using a DFT which uses M rather than N points, where $M \geq 3N/2$. To see how this works, consider a new set of grid points,

$$y_j = \frac{2\pi j}{M}, \tag{A.32}$$

and a modified pseudospectral method based on the transforms

$$U_j = \sum_{k=-M/2}^{M/2-1} \tilde{u}_k e^{iky_j}, \quad j = 0, 1, \ldots, M-1, \tag{A.33}$$

$$V_j = \sum_{k=-M/2}^{M/2-1} \tilde{v}_k e^{iky_j}, \quad j = 0, 1, \ldots, M-1, \tag{A.34}$$

$$W_j = U_j V_j, \quad j = 0, 1, \ldots, M-1, \tag{A.35}$$

where our spectrum in this space should satisfy[1]

$$\tilde{u}_k = \begin{cases} \hat{u}_k, & |k| \leq N/2, \\ 0 & \text{otherwise.} \end{cases} \tag{A.36}$$

[1] Note that we are using the tilde here to denote the modified spectrum and that both spectra are finite and discrete.

If we rewrite our DFT similarly in terms of M and y_j, we find that our solution in Fourier space is

$$\tilde{W}_k = \sum_{m+n=k} \tilde{u}_m \tilde{v}_n + \sum_{m+n=k\pm M} \tilde{u}_m \tilde{v}_n. \tag{A.37}$$

As we are only interested in $\tilde{W}_k$ for values of $|k| \leq N/2$, we simply choose M such that the second term vanishes for these k. Since $\tilde{u}_m$ and $\tilde{v}_m$ are zero for $|m| > N/2$, the worst-case is

$$-\frac{N}{2} - \frac{N}{2} \leq \frac{N}{2} - 1 - M, \tag{A.38}$$

$$\implies M \geq \frac{3N}{2} - 1. \tag{A.39}$$

This procedure, a pseudocode implementation of which is given in Algorithm A.6, is often called the 3/2 rule. It is also possible to satisfy the above condition by truncating an M-sized spectrum, yielding the so-called 2/3 rule. According to Boyd (2000), a total of M basis functions are used at intermediate stages of the computation, while only the lowest N wavenumbers are retained at each timestep. According to Bardos and Tadmor (2015), both the spectral and the 2/3-dealiased pseudospectral methods converge spectrally for several smooth partial differential equations, but both must admit spurious oscillations for non-smooth solutions.

Algorithm A.6: Computation of nonlinear terms with dealiasing by padding.

function NONLINEAR($\tilde{u}, \tilde{v}$) ▷ $\tilde{u}$, $\tilde{v}$: size-N Fourier coeff. arrays
  $\tilde{U}, \tilde{V} \leftarrow$ <zeros>$[2N]$ ▷ Create zero arrays of size $2N$
  $\tilde{U}[N/2 : 3N/2], \tilde{V}[N/2 : 3N/2] \leftarrow \tilde{u}[:], \tilde{v}[:]$ ▷ Padding
  $U, V \leftarrow \mathcal{F}^{-1}(\tilde{U}), \mathcal{F}^{-1}(\tilde{V})$
  $W \leftarrow U * V$ ▷ Multiplication in realspace
  $\tilde{W} \leftarrow \mathcal{F}(W)$ ▷ Fourier transform to wavespace
  $\tilde{w}[:] \leftarrow \tilde{W}[N/2 : 3N/2]$ ▷ Truncate back to size N

  return $\tilde{w}$
end function

Bibliography

A. Alexakis. Rotating Taylor-Green flow. *Journal of Fluid Mechanics*, 769:46–78, 2015.

G. Ariel, S. J. Kim, and R. Tsai. Parareal methods for highly oscillatory dynamical systems. *SIAM Journal on Scientific Computing*, 2016.

Christophe Audouze, Marc Massot, and Sebastian Volz. Symplectic multi-time step parareal algorithms applied to molecular dynamics. Submitted to SIAM Journal of Scientific Computing, 2009.

A. Babin, A. Mahalov, and B. Nicolaenko. Fast singular oscillating limits and global regularity for the 3d primitive equations of geophysics. *ESAIM: M2AN*, 34(2):201–222, 2000.

G. Bal and Y. Maday. A parareal time discretization for non-linear PDEs with application to the pricing of an American put. In L. F. Pavarino and A. Toselli, editors, *Recent Developments in Domain Decomposition Methods*, volume 23 of *Lecture Notes in Computational Science and Engineering*, pages 189–202. Springer Berlin Heidelberg, 2002.

G. Bal and Q. Wu. *Symplectic Parareal*, pages 401–408. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Guillaume Bal. *On the Convergence and the Stability of the Parareal Algorithm to Solve Partial Differential Equations*, pages 425–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

C. Bardos and E. Tadmor. Stability and spectral convergence of Fourier method for nonlinear problems: on the shortcomings of the 2/3 de-aliasing method. *Numerische Mathematik*, 129(4): 749–782, April 2015.

A. Batkai, P. Csomos, and G. Nickel. Operator splittings and spatial approximations for evolution equations. *Journal of Evolution Equations*, 9(3):613–636, 2009.

H. Berland. Exponential integrators. Given at University of Central Florida, Orlando, FL, USA, 2005.

L. Biferale, S. Musacchio, and F. Toschi. Split energy-helicity cas-

cades in three-dimensional homogeneous and isotropic turbulence. *Journal of Fluid Mechanics*, 730:309–327, 2013.

N. Bogoliubov and Y. Mitropolsky. *Asymptotic Methods in the Theory of Nonlinear Oscillations*. Gordon and Breach, New York, 1961.

J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Mineola, New York, 2000.

G. Browning and H. Kriess. Splitting methods for problems with different timescales. *Monthly Weather Review*, 122(11):2614–2622, 1994.

C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, Berlin, 1988.

J. G. Charney. On the scale of atmospheric motions. *Geophysiske Publikasjoner*, 17(2):3–17, 1948.

J. G. Charney. On a physical basis for numerical prediction of large-scale motions in the atmosphere. *Journal of Meteorology*, 6:371–385, 1949.

J. G. Charney and N. Phillips. A numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows. *Journal of Meteorology*, 10:71–99, 1953.

Q. Chen, S. Chen, G.L. Eyink, and D. D. Holm. Resonant interactions in rotating homogeneous three-dimensional turbulence. *Journal of Fluid Mechanics*, 542:139–164, 2005.

A. Chertock and A. Kurganov. On splitting-based numerical methods for convection-diffusion equations. *Quaderni di Mathematica, Dept. Math. Seconda Univ. Napoli*, 24:303–343, 2009.

P. Clark di Leoni and P. D. Mininni. Quantifying resonant and near-resonant interactions in rotating turbulence. *Journal of Fluid Mechanics*, 809:821 – 842, 2016.

S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *J. Comput. Phys.*, 176(2):430–455, 2002.

C. F. Curtiss and J. O. Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America*, 38(3):235–243, 1952.

T. Davies, A. Staniforth, N. Wood, and J. Thuburn. Validity of anelastic and other equation sets as inferred from normal-mode analysis. *Quarterly Journal of the Royal Meteorological Society*, 129 (593):2761–2775, 2003.

J. A. Domaradzki and R. S. Rogallo. Local energy transfer and nonlocal interactions in homogeneous, isotropic turbulence. *Physics of Fluids A: Fluid Dynamics*, 2(3):413–426, 1990.

J. Duchon and R. Robert. Dissipation d'énergie pour des solutions faibles des équations d'Euler et Navier-Stokes incompressibles. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 329(3):243 – 248, 1999.

D. R. Durran. *Numerical Methods for Fluid Dynamics: With Applications to Geophysics*. Springer, 2010.

W. E. Analysis of the heterogeneous multiscale method for ordinary differential equations. *Commun. Math. Sci.*, 1(3):423–436, 2003.

W. E and B. Engquist. The heterognous multiscale methods. *Commun. Math. Sci.*, 1(1):87–132, 03 2003.

W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden. The heterogeneous multiscale method: A review. *Commun. Comput. Phys.*, page 2007, 2007.

P. F. Embid and A. J. Majda. Averaging over fast gravity waves for geophysical flows with arbitrary potential vorticity. *Communications in Partial Differential Equations*, 21(3–4):619–658, 1996.

P. F. Embid and A. J. Majda. Averaging over fast gravity waves for geophysical flows with unbalanced initial data. *Theoretical and Computational Fluid Dynamics*, 11:155–169, 1998.

B. Engquist and Y.-H. Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Mathematics of Computation*, 74(252):1707–1742, 2005.

R. D. Falgout, T. A. Manteuffel, B. OâĂŹNeill, and J. B. Schroder. Multigrid reduction in time for nonlinear parabolic problems. 2016.

M. Farge and R. Sadourny. Wave-vortex dynamics in rotating shallow water. *Journal of Fluid Mechanics*, 206:433âĂŞ–462, 1989.

C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. *International Journal for Numerical Methods in Engineering*, 58(9):1397–1434, 2003.

P. F. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 433–440. Springer Berlin Heidelberg, 2005.

B. Gallet. Exact two-dimensionalization of rapidly rotating large-reynolds-number flows. *Journal of Fluid Mechanics*, 783:412–447, 2015.

M. J. Gander. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition*. Springer-Verlag, 2015.

M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. Technical report, SIAM J. Sci. Comput, 2005.

Martin J. Gander and Ernst Hairer. Analysis for parareal algorithms applied to hamiltonian differential equations. *J. Comput. Appl. Math.*, 259:2–13, March 2014.

C. W. Gear and D. R. Wells. Multirate linear multistep methods. *BIT Numerical Mathematics*, 24(4):484–502, 1984.

W. Gropp. *Parallel computing and domain decomposition*, pages 349–361. Publ by Soc for Industrial & Applied Mathematics Publ, 1992.

C. Grossmann, H.-G. Roos, and M. Stynes. *Numerical Treatment of Partial Differential Equations*. Springer-Verlag Berlin Heidelberg, 1 edition, 2007.

J. L. Hammack and D. M. Henderson. Resonant interactions among surface water waves. *Annual Review of Fluid Mechanics*, 25:55–97, 1993.

T. S. Haut and B. A. Wingate. An asymptotic parallel-in-time method for highly oscillatory pdes. *SIAM Journal on Scientific Computing*, 36(2):A693–A713, 2014.

T. S. Haut, T. Babb, P. G. Martinsson, and B. A. Wingate. A high-order time-parallel scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator. *IMA Journal of Numerical Analysis*, 2015.

L. He. The reduced basis technique as a coarse solver for parareal in time simulations. *J. Comput. Math.*, 28:676 – 692, 2010.

J. Heller. *Jak Orat s Čertem: Kázání*. Kalich, 2005.

G. Hernandez-Duenas, L. M. Smith, and S. N. Stechmann. Investigation of Boussinesq dynamics using intermediate models based on wave–vortical interactions. *Journal of Fluid Mechanics*, 747: 247–287, 2014.

D. J. Higham and L. N. Trefethen. Stiffness of ODEs. *BIT Numerical Mathematics*, 33(2):285–303, 1993.

E. J. Hinch. *Perturbation methods*. Cambridge University Press, 1991.

H. Hochbruck and A. Ostermann. Explicit integrators of Rosenbrock type. *Oberwolfach Reports*, 3:1107–1110, 2006.

M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 5 2010.

R. A. Horn and C. R. Johnson, editors. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 1986.

A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, New York, NY, USA, 2nd edition, 2008.

M. Jardak, I. M. Navon, , and M. Zupanski. Comparison of sequential data assimilation methods for the Kuramoto-Sivashinsky equation. *Int. J. Numer. Meth. Fluids*, 62:374–402, 2010.

S. G. Johnson. Saddle-point integration of C^∞ "Bump" functions. *manuscript*, 2007. URL `http://math.mit.edu/~stevenj/bump-saddle.pdf`.

U. Kadri and T. R. Akylas. On resonant triad interactions of acoustic-gravity waves. *J. Fluid Mech.*, 788, 2016.

A. K. Kassam and L. N. Trefethen. Fourth-order time stepping for stiff PDEs. *SIAM J. Sci. Comput*, 26:1214–1233, 2005.

D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 1991.

S. Klainerman and A. J. Majda. Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids. *Communications in Pure and Applied Mathematics*, 34(4):481–524, 1981.

J. B. Klemp and R. B. Wilhelmson. The simulation of three-dimensional convective storm dynamics. *Journal of the Atmospheric Sciences*, 35(6):1070–1096, 1978.

R. H. Kraichnan. Irreversible statistical mechanics of incompressible hydromagnetic turbulence. *Phys. Rev.*, 109:1407–1422, Mar 1958.

P. R. Kramer, J. A. Biello, and Y. Lvov. Application of weak turbulence theory to FPU model. In *The Fourth International Conference on Dynamical Systems and Differential Equations*, 2002.

A. Kreienbuehl, A. Naegel, D. Ruprecht, R. Speck, G. Wittum, and R. Krause. Numerical simulation of skin transport using parareal. *Computing and Visualization in Science*, 17, 2015.

H-O. Kreiss and J. Oliger. Stability of the Fourier method. *SIAM J. Numer. Anal.*, 16:421–433, 1979.

S. Krogstad. Generalized integrating factor methods for stiff PDEs. *Journal of Computational Physics*, 203(1):72–88, 2005.

N. M. Krylov and N. N. Bogoliubov. Methods approchees de la mecanique non-lineaire dans leurs appication a l'Aeetude de la perturbation des mouvements periodiques de divers phenomenes de resonance s'y rapportant. *Kiev: Academie des Sciences d'Ukraine*, 1935.

F. Legoll, T. Lelièvre, and G. Samaey. A micro-macro parareal algo-

rithm: Application to singularly perturbed ordinary differential equations. *SIAM J. Scientific Computing*, 35(4), 2013.

B. Leimkuhler and S. Reich. A reversible averaging integrator for multiple time-scale dynamics. *Journal of Computational Physics*, 171 (1):95 – 114, 2001.

M. P. Lelong and J. J. Riley. Internal wave-vortical mode interactions in strongly stratified flows. *Journal of Fluid Mechanics*, 232:1–19, 1991.

J. Lions, Y. Maday, and G. Turinici. A "parareal" in time discretization of pde's. *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, 332(7):661–668, 2001.

P. Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431 – 3444, 2008.

Y. Maday and G. Turinici. Parallel in time algorithms for quantum control: Parareal time discretization scheme. *International journal of quantum chemistry*, 93(3):223–228, 2003.

A. Majda. *Introduction to PDEs and Waves for the Atmosphere and Ocean: Courant Lecture Notes Vol. 9*. American Mathematical Society and Courant Institute of Mathematical Sciences, 2002.

M. McCool, J. Reinders, and A. Robison. *Structured Parallel Programming: Patterns for Efficient Computation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012.

R. I. McLachlan and G. R. W. Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.

S. E. Minkoff. Spatial parallelism of a 3d finite difference velocity-stress elastic wave propagation code. *SIAM Journal on Scientific Computing*, 24(1):1–19, 2002.

C. Moler and C. van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45 (1):3–49, 2003.

S. Nazarenko. *Wave Turbulence*. Springer, 2011.

A. C. Newell. Rossby wave packet interactions. *Journal of Fluid Mechanics*, 35(2):255–271, 1969.

A. C. Newell and Benno Rumpf. Wave turbulence. *Annual Review of Fluid Mechanics*, 43:59–78, 1 2011.

S. A. Orszag. On the elimination of aliasing in finite difference schemes by filtering high-wavenumber components. *Journal of the Atmospheric Sciences*, 28:1074, 1971.

T. Passot and A. Pouquet. Hyperviscosity for compressible flows

using spectral methods. *Journal of Computational Physics*, 75(2):300 – 313, 1988.

N. J. Phillips. The general circulation of the atmosphere: a numerical experiment. *Quart. J. Roy. Met. Soc.*, 82:123–164, 1956.

N. J. Phillips. An example of nonlinear computational instability. *The atmosphere and the sea in motion*, pages 501–504, 1959.

O. M. Phillips. The interaction trapping of internal gravity waves. *Journal of Fluid Mechanics*, 34(2):407–416, 1968.

L. M. Polvani, J. C. McWilliams, M. A. Spall, and R. Ford. The coherent structures of shallow water turbulencel deformation radius effects, cyclone/anticyclone antisymmetry and gravity wave generation. *Chaos*, 4(177), 1994.

D. A. Pope. An exponential method of numerical integration of ordinary differential equations. *Commun. ACM*, 6(8):491–493, 1963.

S. B Pope. *Turbulent flows*. Cambridge Univ. Press, Cambridge, 2011.

M. Remmel, J. Sukhatme, and L. M. Smith. Nonlinear gravity-wave interactions in stratified turbulence. *Theoretical and Computational Fluid Dynamics*, 28(2):131–145, 2014.

D. P. Rodgers. Improvements in multiprocessor system design. *SIGARCH Comput. Archit. News*, 13(3):225–231, 1985.

P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer-Verlag Berlin Heidelberg, 3 edition, 2011.

J. A. Sanders, F. Verhulst, and J. Murdock. *Averaging methods in nonlinear dynamical systems*. Applied mathematical sciences. Springer, New York, Berlin, Heidelberg, 2 edition, 2007.

S. Schochet. Fast singular limits of hyperbolic pdes. *Journal of Differential Equations*, 114:476–512, 1994.

M. Schreiber, P. S. Peixoto, T. Haut, and B. Wingate. Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. *The International Journal of High Performance Computing Applications*, 0(0), 2017.

T. Sedláček. *The Economics of Good and Evil*. Oxford University Press, 1 edition, 2011.

S. Singh. *Fermat's Last Theorem: the story of a riddle that confounded the world's great minds for 358 years.* Fourth Estate Limited, 1997.

L. M. Smith and Y. Lee. On near resonances and symmetry breaking in forced rotating flows at moderate rossby number. *Journal of Fluid Mechanics*, 535:111–142, 2005.

L. M. Smith and F. Waleffe. Transfer of energy to two-dimensional

large scales in forced, rotating three-dimensional turbulence. *Physics of Fluids*, 11(6):1608–1622, 1999.

M. A. Spall and J. C. McWilliams. Rotational and gravitational influences on the degree of balance in the shallow water equations. *Geophys. Astrophys. Fluid Dyn.*, 64:1–29, 1992.

M.N. Spijker. Stiffness in numerical initial-value problems. *Journal of Computational and Applied Mathematics*, 72(2):393 – 406, 1996.

E. M. Staff and G. A. Rønquist. *Stability of the Parareal algorithm*, pages 425–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

J. Stewart. *Calculus*. Cengage Learning, 2007.

G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

E. Tadmor. The exponential accuracy of Fourier and Chebyshev differencing methods. *SIAM Journal on Numerical Analysis*, 23(1): 1–10, 1986.

N. M. Temme. Uniform asymptotic methods for integrals. *Indagationes Mathematicae*, 24(4):739 – 765, 2013.

C. Temperton. Can spectral methods on the sphere live forever? In *Workshop on Developments in Numerical Methods for Very High resolution global models, 5-7 June 2000*, pages 161–166, Shinfield Park, Reading, 2000. ECMWF, ECMWF.

M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (epi) methods. *Journal of Computational Physics*, 213(2):748 – 776, 2006.

G. P. Tolstov. *Fourier Series, translated from the Russian by R. A. Silverman*. Dover Publications, Inc., New York, New York, USA, 1962.

L. N. Trefethen. *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations, unpublished text*. unpublished, 1996.

J. M. F. Trindade and J. C. F. Pereira. Parallel-in-time simulation of the unsteady Navier-Stokes equations for incompressible flow. *International Journal for Numerical Methods in Fluids*, 45(10), 2004.

G. J. Tripoli and W. R. Cotton. The colorado state university three-dimensional cloud/mesoscale model. part i: General theoretical framework and sensitivity experiments. *J. Rech. Atmos.*, 16, 1982.

G. M. Tuynman. The Hamiltonian? In *Geometric Methods in Physics: XXXII Workshop, Białowieża, Poland, June 30-July 6, 2013*, pages 287–290. Springer International Publishing, 2014.

G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics*. Cambridge University Press, Cambridge, U.K., 2006.

F. Waleffe. The nature of triad interactions in homogeneous turbulence. *Physics of Fluids A: Fluid Dynamics*, 4(2):350–363, 1992.

M. L. Ward and W. K. Dewar. Scattering of gravity waves by potential vorticity in a shallow-water fluid. *Journal of Fluid Mechanics*, 663:478–506, 11 2010.

J. P. Whitehead, T. Haut, and B. A. Wingate. The separation of three distinct time scales in the rotating, stratified, Boussinesq equations: Variations from Quasi-Geostrophy. *Submitted to Nonlinearity*, 2014.

L. J. Wicker and R. B. Wilhelmson. Simulation and analysis of tornado development and decay within a three-dimensional supercell thunderstorm. *J. Atmos. Sci.*, 1995.

A. Wiles. Modular elliptic curves and Fermat's last theorem. *Annals of Mathematics*, 141(3):443–551, 1995.