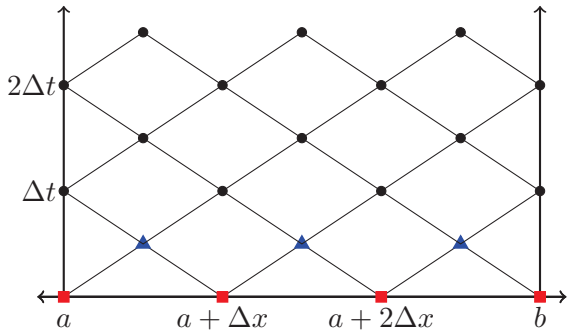
# SYMPLECTIC INTEGRATORS FOR VAKONOMIC EQUATIONS AND FOR MULTI-HAMILTONIAN EQUATIONS




A thesis presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Mathematics

at Massey University, Palmerston North,

New Zealand.

Matthew Colin Wilkins

2016

# Contents

# List of Tables

# List of Figures

# Abstract

This thesis contains research contributions to two areas of geometric integration, one concerning ordinary differential equations (ODEs) and one concerning partial differential equations (PDEs).

The ODE contribution is a symplectic integrator for variational nonholonomic equations, also known as vakanomic equations. These systems include the Hamiltonian version of variational problems subject to position and velocity constraints that are nondegenerate in the velocities. Such constraints have the form $g_i(q) \cdot \dot{q} = 0$, and arise in sub-Riemannian geometry and control theory. An equivalence between variational equations with these constraints and generalized Hamiltonian systems of the form $J\dot{z} = \nabla H(z)$ ($J$ not necessarily invertible) with such index 1 constraints is shown. This is extended to constraints of the form $g_i(q, \dot{q}) = 0$. It is shown that symplectic Runge–Kutta methods are also symplectic on generalized Hamiltonian systems and provide effective symplectic integrators for Hamiltonian systems with index 1 constraints. Two sample applications are given: finding the motion of a two-wheeled idealized vehicle, and finding the geodesics of the Heisenberg group.

The PDE contribution is a new multisymplectic integrator. A class of general purpose linear multisymplectic integrators for multi-Hamiltonian equations based on a diamond-shaped mesh is introduced. The class of integrators consists of a *simple diamond scheme*, and a *diamond scheme* parameterized by order $r$. On each diamond, the former discretizes the PDE on the corners of the diamond, and the latter discretizes the PDE with $r$-stage symplectic Runge–Kutta methods. Both schemes advance in time by filling in each diamond locally, leading to greater efficiency and parallelization, and easier treatment of boundary conditions compared to methods based on rectangular meshes. The simple diamond scheme is order two in both space and time, has solvable equations for the one-dimensional wave equation, satisfies a discrete conservation law, and can be extended to allow Dirichlet and Neumann boundary conditions for the wave equation. The diamond scheme is defined on all multi-Hamiltonian PDEs, has solvable nonlinear equations for the one-dimensional wave equation, satisfies a discrete symplectic conservation law, and scales exceedingly well with the number of processors

on which it is run. A method for handling Dirichlet and Neumann boundary conditions is presented. Discrete dispersion relations, and associated stability results, are found for the wave equation and the cubic nonlinear Schrödinger equation. Numerical experiments, for both the simple diamond and $r$ diamond schemes, are presented to support an estimate of the order of the schemes. Difficulties and complications, including boundary treatment and the multi-step nature of the diamond scheme, are discussed.

# Acknowledgements

This section was the most pleasurable to write: thinking of all the help I have received reminds me of how fortunate I am. I would not have been able to start this thesis without considerable support, let alone finish it.

First, of course, I thank my family. My parents for their support and upbringing. My brother, Andy, who: was always interested in knowing the details, asked insightful questions that made me think from a different perspective, and helped me with isoparametric coordinate transformations between triangles and diamonds. My kids who suffered having an absentee dad, especially when I was close to submitting. My wife, Heather, who not only had to put up with me doing less about the house, but also: proofread the entire thesis; checked all my articles and documents; listened to all my practice talks; put up with me grumbling about proofs that didn't work or code that was buggy; and made suggestions when I was stumped.

I started my PhD on a part-time basis while working at ITS Massey University. I am thankful to ITS, in particular Clive Martis, the CTO at the time, for allowing me to do this. ITS partially paid my fees, and allowed me time to meet with my supervisors.

In the first year of study my supervisor had two visitors: Klas Modin and Olivier Verdier. Both are very clever Mathematicians, but equally important to me, was that they were kind, clear, and willing to explain things to a slow, inexperienced student such as myself. Around that time we had the 'pre-magic conference' at Waikanae; that, and these two gentlemen really brought some life to the beginning of my PhD.

New Zealand eScience Infrastructure (NeSI) provided the parallel computing resources for me to test my code. The process, from requesting access to actually using a cluster, was straightforward and very smooth. NeSI clusters have an enormous amount of software (I didn't have to request a single package be installed!) and are staffed by knowledgeable and helpful people.

I was very thankful to be awarded a Massey Vice-Chancellor's Doctoral Scholarship. I could not have switched to full-time study without that ongoing money. Thank you to Simon Hall, head of IFS, for fostering a generous attitude within the institute towards PhD students, and authorizing the paying of my fees. Bruce van Brunt arranged part-time lecturing for me, not only did I appreciate the extra money, but greatly value the

experience this gave me.

In the course of my studies I got cancer. From my hospital bed, I emailed Kathryn Stowell asking her to be my champion and sort out all the paperwork and bureaucracy in changing from full-time back to part-time, and filing for a suspension. I have never been much for form filling at the best of times, so I very much appreciated what she did.

My co-supervisor, Stephen Marsland, is a busy person, but was always very generous with his time. When I was hopelessly lost, he took the time to explain things: I really appreciated learning what a tangent space really was. I enjoyed telling Stephen where I had got to in the project: explaining what I had been doing really helped in understanding where to go next. I am grateful for the Massey thesis LaTeX style and skeleton files that he created. Stephen read through the entire thesis in very short order and provided valuable feedback and improvements to the flow of this document.

Finally, I thank my supervisor Robert McLachlan. I could not have wished for a better supervisor. He took me on with no reservations, even though I had been out of academia for 15 years, and I quit half-way through my last attempt at a PhD! Robert has always been able to provide money for fees, conferences, and travel. He has been very generous with his encouragement and time, especially as I reached submission time. Robert was always happy to see me, and if necessary, provide commiserations and suggestions on what to try next. He always had an idea on how to rescue us from whatever mathematical hole I had managed to get us into!

# Chapter 1

# Introduction

Symplectic integrators are examples of geometric numerical integrators: methods that preserve certain geometric properties of the differential equation that is being solved numerically. Typical geometric properties include: symplectic structure (the focus of this thesis), time-reversal symmetry, phase space volume, first integrals such as energy, momentum, and angular momentum. Because geometric numerical integrators are typically run in floating point arithmetic they cannot *exactly* preserve these properties, what is meant by 'preserved' is preserved *up to round-off error*.

Numerical methods for solving ordinary differential equations (ODEs) have matured to the point that there are many robust mature numerical codes available. MATLAB includes a number of ODE solvers, such as `ode45`, that can be readily used 'out of the box'. If undergraduate textbooks on numerical methods, and such software packages as MATLAB and MATHEMATICA are anything to go by, Runge–Kutta and multi-step methods are widely taught, understood and used. Methods of any order, both explicit and implicit, can be constructed, thus giving any desired level of accuracy for a wide range of problems. Of course, this accuracy may come at increased, or prohibitive, cost. Also, very high accuracy may not be desired *per se*; instead one may desire good 'qualitative' behaviour of the modelled system. Geometric numerical integrators are concerned not only with accuracy of, for example, a particular solution at a particular time, as these traditional methods are, but also other qualitative, or geometric, properties.

As discussed at the end of this Chapter, preserving geometric properties of the ODE not only improves the qualitative nature of the computed solution, but also leads to better long time behaviour of the integrator. This is especially important for large time scale calculations such as those that arise in celestial mechanics. To calculate, to some specified accuracy, the positions and momenta of all the heavenly bodies of some reasonable size at some far future time may require a method with such high order or so many small time steps, that it is impractical. However certain symplectic integrators can, with much larger time steps and/or less work, produce a solution with

better qualitative properties and end up being just as accurate.

A constrained Hamiltonian ODE is a type of ODE that incorporates constraints. The motion of a car can be modelled with these systems: one of the constraints is that the car can only go in the direction the front wheels are pointing (assuming it isn't skidding). Symplectic methods do exist for numerically solving constrained Hamiltonian systems when the constraints are simple, such as positional constraints. This thesis will explore symplectic integrators for the so-called vakonomic equations, these are equations with other constraints, such as the example of the motion of a car.

So far this introduction has focussed on ODEs. The study of so-called multisymplectic integrators for partial differential equations (PDEs) is younger and less well understood. Added to the geometric properties discussed above for the ODEs are so-called conservation laws. Just as in the ODE case, it is desirable for the numerical method to preserve these laws, or discrete forms of them. One approach for numerically solving a PDE is *semi-discretization* in space, whereby the PDE is converted to a system of ODEs. These can then be numerically solved using a symplectic integrator. However in this thesis semi-discretization will not be used; space and time will be treated equally, and multisymplectic integrators that numerically solve the multi-Hamiltonian system directly, while preserving certain conservation laws, will be discussed. Current multisymplectic integrators will be seen to have some detracting qualities: they are often low in order, and sometimes the numerical equations are not solvable, or the integrator is restricted to a very specific PDE. In this thesis a new method, the 'diamond scheme', will be introduced that overcomes some of these problems.

The remainder of this chapter proceeds as follows. Section 1.1 starts by introducing and giving examples of Hamiltonian ODEs and PDEs. In Section 1.2, conservation laws and symmetry for ODEs and PDEs are introduced. Broadly there are two types of conservation law, those that apply to the differential equation, and those that apply to its variation. Both types are discussed for both ODEs and PDEs, and examples of numerical methods that preserve these laws are given. Section 1.3 introduces Hamiltonian ODEs with 'holonomic' (in other words positional) constraints by giving an example of such a system and later, in Section 1.3.1, numerical methods for such systems are discussed. Finally the Chapter closes with a discussion of why symplectic methods might be used.

## 1.1 Hamiltonian ODEs and PDEs

A Hamiltonian system is a mathematical description of a physical system. From the Hamiltonian, which is the total energy of the system, the equations of motion of the system can be derived [51, pg. 2–3]. The phase space for a system with $d$ degrees of freedom, is a subset of $\mathbb{R}^{2d}$, with the state variables $(q, p) \in \mathbb{R}^{2d}$, where both $q \in \mathbb{R}^d$

and $p \in \mathbb{R}^d$. The variable $q$ is often referred to as the generalized coordinates, because $q$ need not be the usual Cartesian coordinates, but could be angles or arc lengths along a curve for instance. Similarly, $p$ is often referred to as the generalized momenta because it may not be the usual physical linear momentum; it may be angular momentum for instance.

The *Hamiltonian*, $H\colon \mathbb{R}^{2d+1} \to \mathbb{R}$, is the map

$$(t, q, p) \mapsto H(t, q, p).$$

Hamilton's equations are given by

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i}, \quad i = 1, \ldots, d, \tag{1.1}$$

and define the time evolution of a point $(q, p)$ in phase space. Often the solution *trajectory*, or *orbit*, is denoted by $(q(t), p(t))$, where $q\colon \mathbb{R} \to \mathbb{R}^d$ and $p\colon \mathbb{R} \to \mathbb{R}^d$ are now thought of as functions of time. If the Hamiltonian does not explicitly depend on time, it is said to be an *autonomous* Hamiltonian. Only autonomous Hamiltonian systems will be considered in this thesis.

An example of an autonomous Hamiltonian system is the simple harmonic oscillator [39, pg. 27]: a particle with mass $m$, with position $q \in \mathbb{R}$, and moving in a potential well $V(q) = \frac{1}{2}\omega^2 q^2$. The Hamiltonian is $H(q, p) = \frac{1}{2m}p^2 + V(q)$, and thus the equations of motion are

$$m\dot{q} = p, \quad \dot{p} = -\omega^2 q. \tag{1.2}$$

Eliminating $p$ leads to $m\ddot{q} = -\omega^2 q$, which is Newton's second law of motion for a particle of unit mass acted on by the force $F = -V'(q) = -\omega^2 q$.

In this thesis, a generalization of Hamiltonian's equations (1.1) is used, which will allow us to handle a wider class of dynamics, including constraints. By introducing the dependent variable $z = (q, p) \in \mathbb{R}^n$ where $n = 2d$, the following definition can be made:

**Definition 1.1.1.** *The* generalized autonomous Hamiltonian ODE *is*

$$J\dot{z} = \nabla H(z), \tag{1.3}$$

*where $J$ is a skew-symmetric* symplectic structure matrix, *$z\colon \mathbb{R} \to \mathbb{R}^n$, and the* Hamiltonian *$H\colon \mathbb{R}^n \to \mathbb{R}$.*

This definition extends the usual *autonomous Hamiltonian ODE* definition [39, pg. 39] by allowing $J$ to be non-invertible; this case will be important in Chapter 2. By allowing rows of zeros in $J$, differential algebraic equations can be expressed in this form. If $J$ is the so-called *canonical* matrix $\begin{pmatrix} 0 & -\mathbb{I}_{d\times d} \\ \mathbb{I}_{d\times d} & 0 \end{pmatrix}$ then Hamiltonian's equations

in (1.1) are recovered.

A multi-Hamiltonian PDE is an extension of the ODE to the situation where $z$ is a function of two or more, in this thesis two, variables: $x$ a space-like variable, and $t$ a time-like variable. Space and time are on an equal footing in this PDE. The energy of the system is not represented by one function, the Hamiltonian, rather there are two functions: $E(z)$ the local energy density, and $F(z)$ the local energy flux. For an autonomous multi-Hamiltonian the total energy is conserved, but energy is not necessarily conserved within an arbitrary region in space; the two functions $E(z)$ and $F(z)$ describe how the energy distribution varies over time and space [9].

**Definition 1.1.2.** *The multi-Hamiltonian PDE is*

$$Kz_t + Lz_x = \nabla S(z), \tag{1.4}$$

*where $K$ and $L$ are constant $n \times n$ skew-symmetric matrices, $z \in \mathbb{R}^n$, and $S\colon \mathbb{R}^n \to \mathbb{R}$ is a smooth function.*

Note that $K$ and $L$ may be singular. If there is no $x$-dependence, $L = 0$, and the PDE reduces to the generalized ODE case. This definition can be extended to allow non-constant $K$ or $L$, or a non-autonomous $S$, however in this thesis the matrices are constant and the right hand side does not depend on $x$ or $t$.

The conservation laws relating the derivatives of $E(z)$ and $F(z)$ will be derived in Section 1.2.2. The canonical example of a multi-Hamiltonian system is a first-order formulation of the one-dimensional nonlinear wave equation, $u_{tt} - u_{xx} = f'(u)$, derived by introducing $v = u_t$ and $w = u_x$. The result is

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_t \\ v_t \\ w_t \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_x \\ v_x \\ w_x \end{pmatrix} = \begin{pmatrix} -f'(u) \\ v \\ -w \end{pmatrix} = \nabla \left( -f(u) + \tfrac{1}{2}v^2 - \tfrac{1}{2}w^2 \right). \tag{1.5}$$

This example appears widely in the literature, and will be used extensively in Chapter 3. Many PDEs, such as Schrödinger, Korteweg–de Vries, and Maxwell's equations can be written in multi-Hamiltonian form (see [31], [4], [74, pg. 2078] respectively); however, there is no systematic unique way of converting a PDE to this formulation.

## 1.2 Conservation laws and symmetry

A Hamiltonian ODE may have certain quantities that do not change along solutions. Or some quantity, depending on solutions to the variational equation (1.8), might be conserved on solutions to the system. While a PDE, or the first variation of the PDE, may have conserved quantities, they have something extra: conservation *laws*. These

laws show how a quantity changing in space is balanced by another quantity changing in time.

One of the key ideas in symplectic integration is that it is desirable for a numerical method applied to a system with certain conservation, or symmetry, laws to preserve discrete analogues of those laws. This will be explored in depth throughout the thesis.

### 1.2.1 ODE conservation laws and symmetry

**Definition 1.2.1.** *A function $G\colon \mathbb{R}^n \to \mathbb{R}$ is a* first integral*, or* constant of motion*, if it is constant along all solution curves $z(t) \in \mathbb{R}^n$ of the Hamiltonian ODE system. In other words*

$$G(z(t)) = G(z(0)) \quad \forall z(0) \in \mathbb{R}^n.$$

A first integral is a 'geometric' property of a system because a first integral implies that the solution of the system is, at least partially, described by the geometry of the lower dimensional manifolds $\{z \in \mathbb{R}^n\colon G(z) = \text{constant}\}$.

It is well known (and easy to check) that total energy, $H$, is a first integral of the canonical Hamiltonian equations (1.1). However, more is true: $H$ is constant for any $z(t)$ that satisfies the generalized autonomous Hamiltonian ODE (1.3).

**Theorem 1.2.2.** *An autonomous Hamiltonian is constant for any $z(t)$ that satisfies* (1.3)*.*

*Proof.* Using the skew-symmetry of $J$,

$$\frac{d}{dt}H(z(t)) = \nabla H(z) \cdot \dot{z} = (J\dot{z})^T \dot{z} = \dot{z}^T J^T \dot{z} = 0.$$

□

For Hamiltonian ODEs arising from physical systems, momentum, angular momentum, and energy are often first integrals.

If a Hamiltonian ODE has *time reversal symmetry* then for every solution trajectory there is another trajectory going in exactly the reverse direction with momentum of opposite sign. This is formalized in the following definition, and illustrated in Figure 1.1.

**Definition 1.2.3.** *A Hamiltonian ODE has* time reversal symmetry *if for every solution $z(t) = (q(t), p(t))$, $\tilde{z}(t) = (\tilde{q}(t), \tilde{p}(t)) = (q(-t), -p(-t))$ is also a solution.*

**Theorem 1.2.4.** *If a Hamiltonian is even in $p$, so $H(q,p) = H(q,-p)$, then the Hamiltonian ODE with canonical $J$ has time reversal symmetry.*

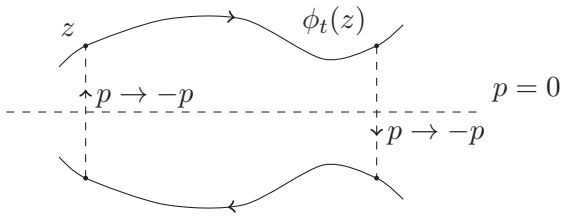



Figure 1.1: The point $z$ is mapped forward to $\phi_t(z)$, reflected about the line $p = 0$, mapped back in time by $t$, and finally reflected again in the $p = 0$ axis, arriving back at $z$.

*Proof.* Suppose that $(q(t), p(t))$ is a solution. Using the fact that if $h(x, y)$ is an even function in $y$, then $h_x$ is also even in $y$, and $h_y$ is odd in $y$,

$$\frac{d\tilde{q}(t)}{dt} = -\dot{q}(-t) = -\nabla_p H(q(-t), p(-t)) = \nabla_p H(\tilde{q}(t), \tilde{p}(t))$$
$$\frac{d\tilde{p}(t)}{dt} = \dot{p}(-t) = -\nabla_q H(q(-t), p(-t)) = -\nabla_q H(\tilde{q}(t), \tilde{p}(t)),$$

thus $(\tilde{q}(t), \tilde{p}(t))$ is also a solution. □

It is desirable for a numerical integrator to preserve whatever first integrals the ODE system has. As an example, the kinetic energy of a rigid body whose mass is at the origin is a quadratic first integral. Any symplectic (see Section 1.2.3) Runge–Kutta method preserves quadratic integrals [17, pg. 236], so the numerical solution produced by the implicit midpoint, a symplectic Runge–Kutta method, will have constant kinetic energy.

### 1.2.2 PDE conservation laws and symmetry

The definition of a Hamiltonian ODE first integral, a quantity that is constant along a solution trajectory, can be applied to Hamiltonian PDEs. The quantity must have no space dependency, an example might be the total energy $\int E(x, t)dx$. However, the PDE setting gives rise to the possibility of local *conservation laws*. These relate how a quantity changing in space is balanced by another quantity changing in time. For the next two results—Theorem 1.2.5 and Proposition 1.2.6—four quantities need to be defined:

$$\begin{aligned} \text{energy density:} \quad & E = S(z) + \tfrac{1}{2} z_x^T L z \\ \text{energy flux:} \quad & F = -\tfrac{1}{2} z_t^T L z \\ \text{momentum density:} \quad & M = -\tfrac{1}{2} z_x^T K z \\ \text{momentum flux:} \quad & I = S(z) + \tfrac{1}{2} z_t^T K z \end{aligned}$$

**Theorem 1.2.5.** *The energy conservation law*

$$E_t + F_x = 0$$

*is satisfied by the solution to* (1.4).

*Proof.* Pre-multiplying (1.4) by $z_t^T$

$$\begin{aligned} z_t^T K z_t + z_t^T L z_x &= z_t^T \nabla S(z) \\ \Rightarrow z_t^T L z_x &= z_t^T \nabla S(z). \qquad \text{using } K^T = -K \end{aligned} \tag{1.6}$$

Now

$$\begin{aligned} E_t + F_x &= \partial_t S(z) + \tfrac{1}{2}\partial_t(z_x^T L z) - \tfrac{1}{2}\partial_x(z_t^T L z) \\ &= \partial_t S(z) + \tfrac{1}{2}\left(z_{xt}^T L z + z_x^T L z_t - z_{tx}^T L z - z_t^T L z_x\right) \\ &= \partial_t S(z) - z_t^T L z_x \qquad \text{using } L^T = -L \\ &= 0 \qquad \text{using (1.6).} \end{aligned}$$

□

**Proposition 1.2.6.** *The momentum conservation law*

$$M_t + I_x = 0$$

*is satisfied by the solution to* (1.4).

*Proof.* The proof is very similar to the energy conservation law. Pre-multiplying (1.4) by $z_x^T$ gives

$$z_x^T K z_t = \partial_x S(z).$$

Then

$$\begin{aligned} M_t + I_x &= -\partial_t(z_x^T K z) + \partial_x S(z) + \partial_x(z_t^T K z) \\ &= \partial_x S(z) - z_x^T K z_t \\ &= 0. \end{aligned}$$

□

For the non-linear wave equation given in (1.5), the energy conservation law given in Theorem 1.2.5 is,

$$\left(\tfrac{1}{2}v^2 - \tfrac{1}{2}w^2 - f(u) + \tfrac{1}{2}(wu_x - uw_x)\right)_t + \left(-\tfrac{1}{2}(wu_t - uw_t)\right)_x = 0. \tag{1.7}$$

Even by using the PDE (1.5), the energy density and energy flux cannot be written as functions of $z$ only. Contrast this to the more usual energy conservation law for the non-linear wave equation, where the energy density and energy flux can be written solely in terms of $z$[1]:

$$\left(\tfrac{1}{2}v^2 + \tfrac{1}{2}w^2 - f(u)\right)_t + (-vw)_x = 0.$$

Similarly for the momentum equation.

With suitable boundary conditions a local conservation law may be integrated over space (or time) to produce a conserved quantity. For example, if the net flux across the domain boundary is zero (energy entering the domain is equal to energy leaving the domain), then integrating the energy conservation law over the entire domain, $\Omega$, yields

$$\int_\Omega (E_t + F_x)dx = 0 \Rightarrow \frac{d}{dt}\int_\Omega E dx = 0,$$

thus the total energy is constant.

### 1.2.3 ODE symplectic conservation law

Recall that an ODE conservation law involves solutions of the ODE; in contrast, the symplectic conservation law involves solutions to the variational equation. Let the solutions of the ODE live in the manifold $\mathscr{M}$, and the local coordinates about a particular $z_0 \in \mathscr{M}$ be $x^i$. An element $\mathbf{u} \in T_{z_0}\mathscr{M}$ (the tangent space of $\mathscr{M}$ at $z_0$), can be expanded in the basis $\{\frac{\partial}{\partial x^1}, \ldots, \frac{\partial}{\partial x^n}\}$

$$\sum_{i=1}^{n} u^i \left.\frac{\partial}{\partial x^i}\right|_{z_0},$$

where $u^i \colon \mathscr{M} \to \mathbb{R}$. Let $u(z) = (u^1(z), \ldots, u^n(z)) \in \mathbb{R}^n$ (note the potential confusion between the element of the tangent space $\mathbf{u} \in T_{z_0}\mathscr{M}$ and the vector in local coordinates $u(z_0)$).

**Theorem 1.2.7.** *An element* $\mathbf{u} \in T_z\mathscr{M}$ *of the tangent space satisfies the* variational

[1] This is derived like so:

$$\begin{aligned}
u_{tt} - u_{xx} &= f'(u) \\
u_{tt}u_t - u_{xx}u_t &= f'(u)u_t \\
u_{tt}u_t - f'(u)u_t &= u_{xx}u_t \\
u_{tt}u_t - f'(u)u_t &= (u_t u_x)_x - u_{tx}u_x \\
u_{tt}u_t + u_{tx}u_x - f'(u)u_t &= (u_t u_x)_x \\
\left(\tfrac{1}{2}(u_t^2 + u_x^2) - f(u)\right)_t &= (u_t u_x)_x
\end{aligned}$$

equation

$$J\frac{d}{dt}u = H''(z)u, \tag{1.8}$$

*where* $u(z) = (u^1(z), \ldots, u^n(z)) \in \mathbb{R}^n$*, and* $H''(z)$ *is the Hessian of* $H$ *at* $z$*.*

*Proof.* Let $\mathbf{u} \in T_z\mathscr{M}$, then $z + \epsilon u \in \mathscr{M}$ is a nearby solution to the ODE, so

$$J(\dot{z} + \epsilon\dot{u}) = \nabla H(z + \epsilon u).$$

Take the derivative with respect to $\epsilon$ and let $\epsilon = 0$,

$$\begin{aligned}\left.\frac{d}{d\epsilon}\right|_{\epsilon=0} J(\dot{z} + \epsilon\dot{u}) &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \nabla H(z + \epsilon u) \\ \Rightarrow J\dot{u} &= H''(z)u.\end{aligned}$$

□

The *one-form*, $dz_i$, is defined by $dz_i(\mathbf{u}) = u^i$. Vector notation for wedge products will be adopted, so

$$\begin{aligned}(\tfrac{1}{2}dz \wedge Jdz)(\mathbf{u}, \mathbf{v}) &= \tfrac{1}{2}\sum_{ij}(J_{ij}dz_i \wedge dz_j)(\mathbf{u}, \mathbf{v}) \\ &= \tfrac{1}{2}\sum_{ij} J_{ij}(dz_i(\mathbf{u})dz_j(\mathbf{v}) - dz_i(\mathbf{v})dz_j(\mathbf{u})) \\ &= \tfrac{1}{2}\sum_{ij} J_{ij}(u^i v^j - v^i u^j).\end{aligned} \tag{1.9}$$

**Theorem 1.2.8.** *Along solutions, a Hamiltonian ODE satisfies the symplectic conservation law*

$$\frac{d\omega}{dt} = 0,$$

*where the two-form* $\omega = \frac{1}{2}(dz \wedge Jdz)$*.*

*Proof.* [2]

$$\begin{aligned}
&\frac{d}{dt}(dz \wedge Jdz)(\mathbf{u}, \mathbf{v}) \\
&\quad = \frac{d}{dt}\left(\sum_{ij} J_{ij}(u^i v^j - v^i u^j)\right) \\
&\quad = \sum_{ij} J_{ij}(\dot{u}^i v^j + u^i \dot{v}^j - \dot{v}^i u^j - v^i \dot{u}^j) \\
&\quad = \sum_{ij} J_{ij}\dot{u}^i v^j + \sum_{ij} J_{ij} u^i \dot{v}^j - \sum_{ij} J_{ij}\dot{v}^i u^j - \sum_{ij} J_{ij} v^i \dot{u}^j \\
&\quad = \sum_{ij} J_{ij}\dot{u}^i v^j - \sum_{ij} J_{ij} u^j \dot{v}^i - \sum_{ij} J_{ij}\dot{v}^i u^j + \sum_{ij} J_{ij} v^j \dot{u}^i \qquad (\text{as } J_{ij} = -J_{ji}) \\
&\quad = 2\sum_{ij} J_{ij}\dot{u}^i v^j - 2\sum_{ij} J_{ij} u^j \dot{v}^i.
\end{aligned}$$

From the skew symmetry of $J$ and Theorem 1.2.7, $\sum_i J_{ij}\dot{u}^i = -\sum_i H''_{ji} u^i$ and $\sum_i J_{ij}\dot{v}^i = -\sum_i H''_{ji} v^i$, so continuing

$$\begin{aligned}
\frac{d}{dt}\omega(\mathbf{u}, \mathbf{v}) &= \sum_j \sum_i J_{ij}\dot{u}^i v^j - \sum_j \sum_i J_{ij}\dot{v}^i u^j \\
&= -\sum_j \sum_i H''_{ji} u^i v^j + \sum_j \sum_i H''_{ji} v^i u^j.
\end{aligned}$$

Swapping the indices $i$ and $j$ in the second sum and using the symmetry of $H$ gives the result. □

**Definition 1.2.9.** *Let $z(t)$ be a solution to the Hamiltonian ODE with $z(0) = z_0$, then the flow map, $\phi_t \colon \mathscr{M} \to \mathscr{M}$ is*

$$\phi_t(z_0) = z(t).$$

*The flow map takes initial points $z_0$ to final points $z(t)$ along trajectories.*

Let $\mathbf{u} \in T_{z_0}\mathscr{M}$ and $\phi'_t(z_0) \equiv \frac{d}{dz}\phi_t\big|_{z_0}$ be the Jacobian of the flow map at $z_0 \in \mathscr{M}$. Then

$$\begin{aligned}
J\frac{d}{dt}(\phi'_t(z_0)u) &= \frac{d}{dz} J \frac{d}{dt}\,\phi_t|_{z_0}\, u \\
&= \frac{d}{dz}\nabla H(\phi_t(z_0))u \\
&= H''(\phi_t(z_0))\phi'_t(z_0)u,
\end{aligned}$$

[2] A much shorter proof is given in Lemma 2.2.1, but it uses some basic rules on wedge products and differentials that have not been presented here. This proof only needs (1.8) and (1.9).

so $\phi_t'(z_0)u \in T_{\phi_t(z_0)}\mathscr{M}$. In other words, the flow maps $\mathbf{u} \in T_{z_0}\mathscr{M}$ forwards to $\phi_t'(z_0)u \in T_{\phi(z_0)}\mathscr{M}$. So two equivalent formulations of the symplectic conservation law are, for any solution $z$,

$$\omega(u,v) = \omega(\phi'(z)u, \phi'(z)v)$$
$$\omega(u,v) = \tilde{\omega}(u,v),$$

where $u, v \in T_z\mathscr{M}$, and $d\tilde{z} = \phi'(z)dz$, $\tilde{\omega} = \frac{1}{2}(d\tilde{z} \wedge Jd\tilde{z})$.

**Proposition 1.2.10.** *When $J$ is the canonical structure matrix, the two-form $\omega = \frac{1}{2}(dz \wedge Jdz)$ can be written in the equivalent forms:*

$$\omega(u,v) = (dp \wedge dq)(u,v) = u^T Jv.$$

*Proof.* Recall $z = (q_1, \ldots, q_d, p_1, \ldots, p_d)$, so

$$\begin{aligned} dz \wedge Jdz &= \sum_{i,j=1}^{n} J_{ij} dz_i \wedge dz_j = \sum_{i=1}^{d} (-1) dq_i \wedge dp_i + \sum_{i=1}^{d} dp_i \wedge dq_i \\ &= 2\sum_{i=1}^{d} dp_i \wedge dq_i = 2dp \wedge dq. \end{aligned}$$

Also

$$\begin{aligned} (dp \wedge dq)(u,v) &= \sum_{i=1}^{d} (dp_i \wedge dq_i)(u,v) = \sum_{i=1}^{d} dp_i(u)dq_i(v) - dq_i(u)dp_i(v) \\ &= \sum_{i=1}^{d} u_{d+i}v_i - u_i v_{d+i} = u^T Jv. \end{aligned}$$

□

Using this equivalent form for the two-form $\omega$, it is straightforward to check that along solutions, a Hamiltonian ODE satisfies the symplectic conservation law:

$$\begin{aligned} \frac{d}{dt}\omega(u,v) &= \dot{u}^T Jv + u^T J\dot{v} \\ &= -(J\dot{u})^T v + u^T H'' v && \text{using } J^T = -J \text{ and (1.8)} \\ &= -(H''u)^T v + u^T H'' v && \text{using (1.8)} \\ &= 0. \end{aligned}$$

When $n = 2$ and $J$ is the canonical structure matrix, $\omega(u,v) = u^2v^1 - u^1v^2$, which is the oriented area of the parallelogram spanned by $u$ and $v$. This is illustrated in

Figure 1.2. Under a symplectic map this area is preserved. It can be shown using integration with a change of variables and the fact that $\det(\phi') = 1$, that a symplectic map, $\phi$, preserves the area of any bounded region of phase space. When $n > 2$, the

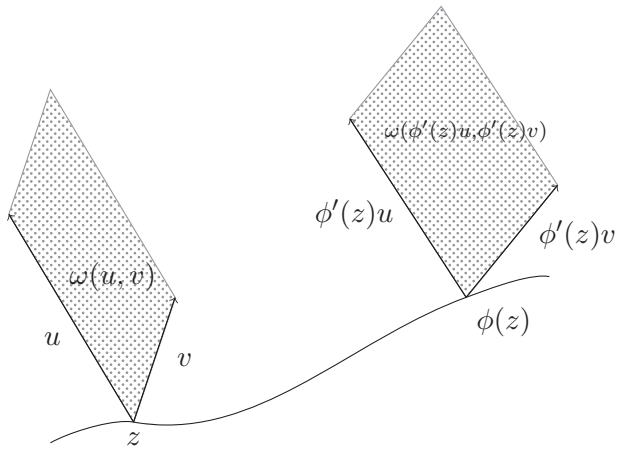



Figure 1.2: $u, v \in T_z\mathscr{M}$, and their images under the flow $\phi$. If the map, $\phi$, is symplectic then the areas of the parallelograms are equal.

two-form $\omega$ is the sum of the oriented areas of the projections of $u$ and $v$ onto $(p_i, q_i)$-coordinates.

Although these geometric results rely on $J$ being canonical, by using Darboux's theorem these results can be extended to $J$ being a constant skew-symmetric matrix [39, pg. 61].

By using similar techniques as above (where the determinate of the change of variables formula is one), it can be shown that a symplectic map preserves phase space volume. However, not all volume preserving maps are symplectic: symplectic area conserving maps are a special case of volume preserving maps where the *shape* of phase space is also preserved in some way. A volume preserving map is allowed to distort phase space in any way whatsoever so long as the volume remains identical. An example is illustrated by Gromov's non-squeezing theorem [22], in which a sphere cannot be embedded into a cylinder of smaller radius by a symplectic map, but by a volume preserving map the sphere could be squashed. This theorem is also known as the *principle of the symplectic camel* [69] because of the biblical parable regarding camels going through the eye of a needle.

This section has introduced the ODE symplectic conservation law, now some examples of numerical integrators that preserve it are given. The Störmer–Verlet, or leapfrog, method is an example of a numerical integrator that preserves a discrete version of the symplectic conservation law. Theorem 2.2.2 shows that the implicit mid-point method applied to the generalized autonomous Hamiltonian ODE, is a symplectic integrator; although, the proof relies on properties of wedge products and exterior calculus. A

more elementary proof using tangent vectors is now presented. Note that this is a new result as $J$ is allowed to be singular.

**Theorem 1.2.11.** *The implicit mid-point method*

$$Jz_1 = Jz_0 + \Delta t \nabla H \left( \frac{z_0 + z_1}{2} \right),$$

*applied to a generalized autonomous Hamiltonian ODE is a symplectic integrator.*

*Proof.* By Proposition 1.2.10 it must be shown that $u^T Jv$ is preserved. Let $u_0, v_0 \in T_{z_0}\mathscr{M}$, then the implicit mid-point maps these forward to $u_1, v_1 \in T_{z_1}\mathscr{M}$ according to

$$\begin{aligned} Ju_1 &= Ju_0 + \Delta t H'' \left( \frac{z_0 + z_1}{2} \right) \cdot \left( \frac{u_0 + u_1}{2} \right) \\ Jv_1 &= Jv_0 + \Delta t H'' \left( \frac{z_0 + z_1}{2} \right) \cdot \left( \frac{v_0 + v_1}{2} \right). \end{aligned}$$

Rearrange these to give

$$\begin{aligned} J(u_1 - u_0) &= K(u_0 + u_1) \\ J(v_1 - v_0) &= K(v_0 + v_1), \end{aligned} \tag{1.10}$$

where $K = \frac{\Delta t}{2} H''(\frac{z_0 + z_1}{2})$. Now

$$\begin{aligned} 2(u_1^T Jv_1 - u_0^T Jv_0) &= (u_0 + u_1)^T J(v_1 - v_0) - (v_0 + v_1)^T J(u_1 - u_0) \\ &= (u_0 + u_1)^T K(v_0 + v_1) - (v_0 + v_1)^T K(u_0 + u_1) && \text{(using (1.10))} \\ &= 0 && \text{(as } K^T = K\text{).} \end{aligned}$$

□

### 1.2.4 PDE symplecticness

As in the ODE case, the symplectic conservation law involves solutions to the variational equation

$$Ku_t + Lu_x = S''(z)u,$$

where $u$ is a tangent vector to a solution $z$ and $S''(z)$ is the Hessian of $S$ at $z$. It is convenient to have the variational equation expressed in terms of the one-form $dz$

$$Kdz_t + Ldz_x = S''(z)dz.$$

**Definition 1.2.12.** *The multisymplectic conservation law is*

$$\omega_t + \kappa_x = 0 \tag{1.11}$$

*where $\omega$ and $\kappa$ are the* pre-*symplectic (to indicate that the matrices $K$ and $L$ might be singular) two-forms:*

$$\omega(u,v) = \frac{1}{2}(dz \wedge Kdz)(u,v),$$
$$\kappa(u,v) = \frac{1}{2}(dz \wedge Ldz)(u,v).$$

As with the $\omega$ appearing in the Hamiltonian ODE setting, the two-forms $\omega$ and $\kappa$ here are oriented areas for the multi-Hamiltonian wave equation: $\omega(u,v)$ is the oriented area between the first and second components (that is the wave location and speed) of $u$ and $v$, and $\kappa(u,v)$ is the oriented area between the first and third components (wave location and slope). In higher dimensions the wedge product is a sum, so, as in the ODE case, $\omega$ and $\kappa$ are sums of oriented areas of projections onto the lower dimensional subspaces. These areas are not individually conserved: $\omega$ can change, but any temporal change must be balanced by a spatial change.

**Proposition 1.2.13.** *The symplectic conservation law given in* (1.11) *is satisfied by the solution to* (1.4)*.*

*Proof.* Take the variation of (1.4) to obtain

$$Kdz_t + Ldz_x = S''(z)dz,$$

then wedge with $dz$ on the left to get

$$dz \wedge Kdz_t + dz \wedge Ldz_x = dz \wedge S''(z)dz.$$

Since $S''$ is symmetric the right hand size is 0. The matrices $K$ and $L$ are antisymmetric so the derivatives can be pulled out with a leading $\frac{1}{2}$ term. □

Two global symplectic conservation laws can be constructed by integrating the multisymplectic conservation law, over space or time, with suitable boundary conditions. Integrating over space gives

$$\int \omega_t + \kappa_x \, dx = 0 \Rightarrow \frac{d}{dt}\int \omega dx = 0,$$

so $\omega$ integrated over all space does not change in time. Similarly,

$$\int \omega_t + \kappa_x \, dt = 0 \Rightarrow \frac{d}{dx} \int \kappa dt = 0,$$

so $\kappa$ integrated over all time does not change in space. These two global symplectic conservation laws do not say that multisymplecticness is conserved globally: the multisymplectic conservation law is strictly a local law.

A numerical method that satisfies a discrete version of (1.11) is called a *multisymplectic integrator*; see [39, 9] for reviews of multisymplectic integration. An example of a multisymplectic integrator is the Keller/Preissman box scheme [39, pg. 344], a two dimensional version of the implicit midpoint method, given by

$$K\delta_t^+ z_{i+1/2}^j + L\delta_x^+ z_i^{j+1/2} = \nabla S(z_{i+1/2}^{j+1/2}), \tag{1.12}$$

where $z_{i+1/2}^j = \frac{z_{i+1}^j + z_i^j}{2}$, $z_i^{j+1/2} = \frac{z_i^{j+1} + z_i^j}{2}$, $\delta_t^+ z_i^j = z_i^{j+1} - z_i^j$ is the forward difference operator in time, and $\delta_x^+ z_i^j = z_{i+1}^j - z_i^j$ is the forward difference operator in space. The scheme is implicit, and satisfies energy and momentum conservation laws and a discrete version of the multisymplectic conservation law, namely

$$\frac{\omega_i^{n+1} - \omega_i^n}{\Delta t} + \frac{\kappa_{i+1}^n - \kappa_i^n}{\Delta t} = 0,$$

where

$$\omega_i^n = \frac{1}{2} dz_{i+1/2}^n \wedge K dz_{i+1/2}^n \quad \text{and} \quad \kappa_i^n = \frac{1}{2} dz_i^{n+1/2} \wedge L dz_i^{n+1/2}.$$

Chapter 3 introduces the diamond scheme; a new class of multisymplectic integrators presented in this thesis. The diamond scheme has a number of advantages over the box scheme; these will be discussed fully from Chapter 3 onwards.

## 1.3 Constrained Hamiltonian systems

Chapter 2 explores symplectic integrators for Hamiltonian ODE systems with so-called 'index 1' constraints, which are a generalization of *holonomic* (positional) constraints. This section will introduce these holonomically constrained Hamiltonian systems.

**Definition 1.3.1.** *The Hamiltonian ODE with k holonomic constraints is*

$$\begin{aligned} \dot{q} &= \nabla_p H \\ \dot{p} &= -\nabla_q H + g'(q)^T \lambda \\ g(q) &= 0, \end{aligned}$$

*where* $g = (g_1, \ldots, g_k)$, $g_i \colon \mathbb{R}^d \to \mathbb{R}$, $g'$ *is the Jacobian of* $g$, *and* $\lambda = (\lambda_1, \ldots, \lambda_k)$.

Note that the $q$ and $p$ expressions can be recast using the canonical structure matrix $J$ and $z = (q, p) \in \mathbb{R}^{2d}$ as

$$J\dot{z} = \nabla \left( H(z) - \sum \lambda_i g_i(q) \right).$$

Secondary constraints are implied by this definition. The solutions start on the constraint manifold $\{q \in \mathbb{R}^d \colon g_i(q) = 0\}$ and must remain there, so

$$0 = \frac{d}{dt} g_i(q(t)) = \nabla g_i(q(t)) \cdot \dot{q}(t) = \nabla g_i(q(t)) \cdot H_p(q, p).$$

Thus the full constraint manifold is

$$\{(q, p) \in \mathbb{R}^n \colon g_i(q) = 0,\ \nabla g_i(q) \cdot H_p(q, p) = 0,\ 1 \leq i \leq k\}.$$

As an example of an holonomically constrained system, consider a number of particles, all of whose positions are given by $q \in \mathbb{R}^d$, moving on a constraint surface, $h \colon \mathbb{R}^d \to \mathbb{R}$ with $h(q) = 0$, in a potential $V \colon \mathbb{R}^d \to \mathbb{R}$. D'Alembert's principle says there is a constraint force $F \colon \mathbb{R}^d \to \mathbb{R}^d$ acting in the normal direction to the constraint surface to keep the particles on that surface [39, pg. 7], thus

$$F(q) = \lambda \nabla h(q),$$

where $\lambda \in \mathbb{R}$ is the *Lagrange* multiplier. To use the notation from Definition 1.3.1 the column $\nabla h(q)$ is replaced with $h'(q)^T$ (the same thing in this example as $k = 1$). Using Newton's second law of motion, the equations of motion are

$$\begin{aligned} M\ddot{q} &= -\nabla_q V(q) + \lambda h'(q)^T \\ h(q) &= 0, \end{aligned}$$

where $M$ is the diagonal mass matrix. Introduce the velocity $v = \dot{q}$ and Hamiltonian $H = \frac{1}{2} v^T M v + V(q)$, then this can be written as

$$\begin{aligned} M\dot{q} &= \nabla_v H \\ M\dot{v} &= -\nabla_q H + \lambda h'(q)^T \\ h(q) &= 0. \end{aligned}$$

By replacing the velocity with generalized momentum, $p = Mv$, this can be rewritten

as

$$\begin{aligned} \dot{q} &= \nabla_p H \\ \dot{p} &= -\nabla_q H + \lambda h'^T \\ h(q) &= 0, \end{aligned}$$

which is in the form given in Definition 1.3.1.

### 1.3.1 Numerical methods for constrained Hamiltonian ODEs

SHAKE and RATTLE [2, 61, 68] are two commonly used numerical integration methods for Hamiltonian problems subject to holonomic constraints. To present these two methods, let $H(q,p) = \frac{1}{2}p^T M^{-1} p + V(q)$ in Definition 1.3.1 to give

$$\begin{aligned} \dot{q} &= v \\ M\dot{v} &= -\nabla V(q) + g'(q)^T \lambda \\ g(q) &= 0, \end{aligned} \tag{1.13}$$

where $p = Mv$. SHAKE applied to this problem is

$$\begin{aligned} q_{n+1} &= q_n + \Delta t v_{n+1/2} \\ Mv_{n+1/2} &= Mv_{n-1/2} - \Delta t \nabla V(q_n) - \Delta t g'(q_n)^T \lambda_n \\ g(q_{n+1}) &= 0 \\ v_n &= \tfrac{1}{2}(v_{n+1/2} + v_{n-1/2}). \end{aligned}$$

It is apparent from the definition that the SHAKE algorithm preserves the constraint $g(q) = 0$ at each time step. However the secondary, implied, constraint is not preserved. RATTLE is

$$\begin{aligned} q_{n+1} &= q_n + \Delta t v_{n+1/2} \\ Mv_{n+1/2} &= Mv_n - \tfrac{\Delta t}{2}\nabla V(q_n) - \tfrac{\Delta t}{2} g'(q_n)^T \lambda_n^{(r)} \\ g(q_{n+1}) &= 0 \\ Mv_{n+1} &= Mv_{n+1/2} - \tfrac{\Delta t}{2}\nabla V(q_{n+1}) - \tfrac{\Delta t}{2} g'(q_{n+1})^T \mu_{n+1} \\ g'(q_{n+1}) v_{n+1} &= 0, \end{aligned} \tag{1.14}$$

where $\mu$ are the velocity multipliers. It is clear that the secondary constraints are preserved by RATTLE. It can be shown [68] that RATTLE and SHAKE are equivalent with $\lambda_n$ from SHAKE replaced with $\frac{1}{2}(\lambda_n^{(r)} + \mu_n)$ from RATTLE. In other words, under the correct initialization RATTLE produces the same sequence of $q_n$ as SHAKE, however

the velocities, $v_n$, differ at the end of each time step.

SHAKE is a step of the Störmer–Verlet leapfrog method applied to the unconstrained problem (that is, (1.13) without the $g'^T\lambda$ part), followed by a projection onto the constraint manifold along the normal to the constraint at the previous time step [39, pg. 176]. In the course of research on index 1 constraints, to be presented in Chapter 2, a geometric formulation of RATTLE was developed. For example, one result is that RATTLE can be viewed as a composition of three methods [47]: first a step along the flow of $J\dot{z} = \lambda\nabla g$; then a leapfrog step for the system $J\dot{z} = \nabla H$, with the $\lambda$ in the first step chosen so that $g(q) = 0$; finally a step along the flow of $J\dot{z} = \mu\nabla g$ where $\mu$ is chosen so that the second implied constraint is satisfied. The proof of this for $H(q,p) = V(q)$ is shown here.

**Theorem 1.3.2.** *One step of* RATTLE *applied to the constrained system given in* (1.13) *is equivalent to*

$$e^{(\Delta t/2)\mu J^{-1}\nabla g} \circ \phi \circ e^{(\Delta t/2)\lambda J^{-1}\nabla g},$$

*where* $e^{(\Delta t/2)\lambda J^{-1}\nabla g}$ *is a* $\frac{\Delta t}{2}$ *step of the (exact) flow of the system* $J\dot{z} = \sum \lambda_i \nabla g_i$, $\lambda$ *is chosen so the intermediary* $q$ *that results from* $\phi \circ e^{(\Delta t/2)\lambda J^{-1}\nabla g}$ *satisfies the constraints* $g(q) = 0$*;* $\phi$ *is one step of Störmer–Verlet leapfrog applied to* $J\dot{z} = \nabla H$*; and* $e^{(\Delta t/2)\mu J^{-1}\nabla g}$ *is a* $\frac{\Delta t}{2}$ *step of the (exact) flow of the system* $J\dot{z} = \sum \mu_i \nabla g_i$, $\mu$ *is chosen so that, for the new* $q$ *and* $v$*, the secondary constraints* $g'(q)v = 0$ *are satisfied.*

*Proof.* The system $J\dot{z} = \sum \lambda_i \nabla g_i$ is

$$\begin{aligned} \dot{q} &= 0 \\ \dot{p} &= -g'(q)^T\lambda, \end{aligned}$$

and a $\frac{\Delta t}{2}$ step along the exact flow of it is

$$\begin{aligned} q_1 &= q_0 \\ p_1 &= p_0 - \frac{\Delta t}{2} g'(q_0)^T\lambda. \end{aligned}$$

The leap-frog step applied to $J\dot{z} = \nabla H$ is

$$\begin{aligned} p_2 &= p_1 - \frac{\Delta t}{2}\nabla V(q_1) \\ q_2 &= q_1 + \Delta t v_2 \\ p_3 &= p_2 - \frac{\Delta t}{2}\nabla V(q_2), \end{aligned}$$

and $\lambda$ is chosen so $g(q_2) = 0$. Combining the above steps, except the $p_3$ update, gives

$$\begin{aligned} q_2 &= q_0 + \Delta t v_2 \\ p_2 &= p_0 - \frac{\Delta t}{2}\nabla V(q_0) - \frac{\Delta t}{2} g'(q_0)^T \lambda. \end{aligned}$$

Making the associations $q_n = q_0$, $p_n = p_0$, $q_{n+1} = q_2$, and $p_{n+\frac{1}{2}} = p_2$ shows this is identical to the position update in (1.14).

The final part of RATTLE, the $\frac{\Delta t}{2}$ step along the exact flow of $J\dot{z} = \sum \mu_i \nabla g_i$, is the correction to ensure the secondary constraints are satisfied. It is given by

$$\begin{aligned} q_3 &= q_2 \\ p_4 &= p_3 - \frac{\Delta t}{2} g'(q_3)^T \mu, \end{aligned}$$

with $g'(q_3)p_4 = 0$. Combining these steps with the $p_3$ update gives

$$p_4 = p_2 - \frac{\Delta t}{2}\nabla V(q_2) - \frac{\Delta t}{2} g'(q_3)^T \mu,$$

which, after making the association $p^{n+1} = p_4$, is identical to the velocity update in (1.14). □

## 1.4 Rationale for using symplectic methods

The solutions to a Hamiltonian ODE live in a symplectic manifold: a smooth manifold equipped with the two-form $\omega$. A symplectic integrator preserves this manifold and two-form. Solutions to a Hamiltonian PDE live in a multisymplectic manifold: again, a smooth manifold but this time with the associated multisymplectic conservation law. A multisymplectic integrator preserves this manifold and conservation law. It is one of many possible attributes a numerical scheme can have, but it is clearly desirable to preserve the symplectic (multisymplectic) manifold. These integrators have many other favourable properties, some of which given below.

As mentioned in Section 1.2.3, the flow of a Hamiltonian ODE with a divergence-free vector field is volume preserving. Symplectic integrators are also volume preserving. This is a good feature when modelling incompressible fluid flow [49, pg. 5265] for example. A Hamiltonian system cannot have asymptotically stable points. This feature is reproduced by a volume preserving numerical method, as it cannot take a region of trajectories to a point of zero volume.

There is an extensively developed theory called backward error analysis, a widely used tool for studying behaviour of numerically computed solutions, which can be used to establish various results including that a symplectic method produces the exact

flow of a nearby Hamiltonian [3, pg. 196]. The difference between this new flow and the flow of the original Hamiltonian is exponentially small in the time step and only grows linearly with time. This leads to approximate conservation laws [8]. Also, even if such numerical schemes have larger error than a nonsymplectic scheme, they often more stable and are qualitatively more accurate [49]. Because the error growth is small, large time steps can be used in simulation. This allows simulations that could not otherwise be performed. As an example, a solar system simulation, that included moons and covered the past 2 million years, was performed using a symplectic integrator [65]. Accuracy was good enough that climatic events such as ice ages could be predicted from the numerically computed orbits.

There is a discrete analogue of the KAM theorem that holds for symplectic integrators [66]. Symplectic integrators preserve invariant tori (a generic feature of Hamiltonian ODE systems is they contain an infinite number of these) when applied to integrable Hamiltonian systems [43].

Although multisymplecticity by itself does not ensure good performance on traditional criteria like accuracy and stability, multisymplectic methods do have a number of advantages.

- They are (essentially) variational; the variational principle being considered fundamental to this class of physical laws.
- They are the standard discrete model in the relevant parts of physics, such as the lattice dynamics of solid state physics.
- For applications such as weather prediction it is important to conserve the potential vorticity (proportional to the product of the vorticity and the gradient of temperature), it was shown in [28] that multisymplectic integrators have two discrete conservation laws that imply conservation of potential vorticity.
- Symplecticity in space is essential for a semidiscretization to be amenable to symplectic time integration, whose advantages in long-time integration are well known.
- At the opposite extreme, steady-state solutions of (3.1) obey a Hamiltonian ODE in space. Methods that are not symplectic in space will typically preserve only the very simplest steady-state solutions, such as constants; methods that are symplectic in space can also preserve periodic, quasi-periodic, and heteroclinic solutions [43]. Standard methods for simple equations, such as those in [15], are multisymplectic.[3]

[3]Cohen [15, p. 34] writes "... all the schemes described below will be centered, since uncentered schemes generate numerical dissipation for wave equations which satisfy a principle of energy conservation." For the linear PDEs he considers, centered symmetric schemes are multisymplectic.

For further discussion, and numerical comparisons of multisymplectic with nonsymplectic integrators, see [5, 30].

However, multisymplectic integrators can only preserve some conservation laws, and these do not typically include those of energy and momentum, which can be important for nonlinear stability. The issue is discussed further in [13, 67]; a full understanding of the relations between discrete conservation laws and discrete multisymplecticity, and their numerical implications, is still lacking.

## 1.5 General overview of this thesis

Chapter 2 extends the class of symplectic integrators for constrained Hamiltonian systems. It starts in Section 2.1 by introducing the class of constrained Hamiltonian ODE problems that will be discussed. It goes onto to develop a class of symplectic integrators for index 1 systems. Section 2.4 generalizes these integrators to include both holonomic and nonholonomic constraints. Finally examples are given in Sections 2.5 and 2.6.

Chapter 3 starts with an introduction of the diamond scheme. Section 3.1 explains in the detail the simple diamond scheme, including a numerical test and one approach to non-periodic boundary conditions. Section 3.2 discusses the full diamond scheme. A description of the algorithm is given, then theorems covering the solvability of the scheme, a discrete symplectic conservation law, and the relationship between the $r = 1$ diamond scheme and the simple diamond scheme are presented. Dispersion analysis for both the simple and full diamond schemes is performed in Section 3.3. Linear stability for the wave equation is presented and Section 3.3.1 details linear stability results for Schrödinger equation.

Chapter 4 explains some of the implementation details. In particular, the tensor like mathematical equations are converted to matrix equations that are more amenable to treatment on a computer in Section 4.1. Numerical solver details follow, and then Section 4.2.1 details data structures for the computer code. The chapter closes in Section 4.3 with results from a parallel implementation.

Chapter 5 opens with an initialization scheme for the diamond scheme in Section 5.1, then goes on to present numerical results and scheme order in Section 5.2. How the diamond scheme lends itself to different boundary condition treatments is explored in Section 5.3 including numerical results.

Conclusions summarizing both the constrained Hamiltonian system work and the diamond scheme appear in Chapter 6.

**Aims and Objectives**

This thesis aims to contribute to two areas of geometric integration; one concerning ODEs and one concerning PDEs. The objective of the ODE contribution is to develop an integrator for the vakanomic equations, and possibly generalize it to other

constrained Hamiltonian systems. The integrator should be symplectic; its order and stability should be established either by proof or numerical experiment; and its feasibility demonstrated numerically. The objective of the PDE contribution is to develop a multisymplectic integrator for the multi-Hamiltonian equations. It should be shown that the integrator leads to solvable equations. The numerical order and conditions for numerical stability should be found, either through numerical experiment or proof. Schemes for dealing with non-periodic boundary conditions should be presented. The numerical efficiency, including the feasibility of running the integrator in parallel, should be demonstrated.

**Publications**

Theorem 1.3.2 has been published in "Geometric Generalisations of SHAKE and RATTLE" [47]. The results found in Chapter 2 have been published in "Symplectic integrators for index 1 constraints" [46]. Most of Chapter 3 and Section 5.2 have been published in the paper "The Multisymplectic Diamond Scheme" [45]. Chapters 4 and 5, and the results pertaining to Schrödinger equation, have not been published yet.

# Chapter 2

# Constrained Hamiltonian systems

Here, symplectic Runge–Kutta methods are shown to provide effective symplectic integrators for Hamiltonian systems with index 1 constraints. These include the Hamiltonian description of variational problems subject to position and velocity constraints that are nondegenerate in the velocities, such as those arising in sub-Riemannian geometry and control theory.

## 2.1 Introduction to constrained Hamiltonian systems

There are several different types of constrained dynamical systems. First, the constraints themselves can be *holonomic* (depending only on position) or *nonholonomic* (depending on position and velocity, but not the derivative of a holonomic constraint). Holonomic constraints were introduced in section 1.3 and methods for solving these systems discussed in section 1.3.1. Nonholonomic constraints are associated with two main types of dynamical system (using the terminology of [7]):

1. The *dynamic nonholonomic equations*, also known as the Lagrange–d'Alembert equations, which describe many mechanical systems in rolling and sliding contact. They have been the subject of several studies in geometric numerical integration (see, for example, [16, 18, 48, 52] and references therein), but as the equations are not in general Hamiltonian or variational and their geometric properties are not fully understood, there is no consensus as to their best discrete version.

2. The *variational nonholonomic equations*, also known as the vakonomic equations, which are discussed in this chapter. They arise in two main contexts in dynamics. The first is from the *Lagrange problem*, that of finding curves $q(t)$ in the configuration manifold $M$ that make the action

$$\int_{t_0}^{t_1} L(q, \dot{q})\, dt,$$

where $L(q,\dot{q})$ is the *Lagrangian*, stationary amongst all curves satisfying the fixed endpoint conditions $q(t_0)=q_0$, $q(t_1)=q_1$ and satisfying the nonholonomic constraints $g(q,\dot{q})=0$. If the action has the form

$$\int_{t_0}^{t_1} \langle \dot{q},\dot{q}\rangle_q \, dt$$

for some metric $\langle , \rangle_q$ on $M$ and the constraint is that $q(t)$ is tangent to a maximally nonintegrable distribution on $M$, then the Lagrange problem becomes the *sub-Riemannian geodesic problem* that defines the sub-Riemannian geometry of $M$ [38, 53][1], an active branch of geometry. Some Lie groups (the Carnot groups, of which the Heisenberg group is an example) have a natural sub-Riemannian structure.

The second is in optimal control problems. Under (quite weak[2]) conditions, the Lagrange problem is equivalent to the optimal control problem $\min_u \int_{t_0}^{t_1} h(q,u)\,dt$ subject to $q(t_0)=q_0$, $q(t_1)=q_1$, and $\dot{q}=f(q,u)$—the control $u$ can be eliminated ([7], Thm 7.3.3). Many applications of the variational nonholonomic equations—to kinematic sub-Riemannian optimal control problems, to control on semi-simple Lie groups and symmetric spaces, to the motion of a particle in a magnetic field, and to optimal control on Riemannian manifolds and Lie groups—are discussed in detail in Chapter 7 of [7]. Many familiar situations—from parking a car (an example modelled numerically in Section 2.5), riding a bike, rolling a ball, to controlling a satellite or a falling cat controlling itself—are described using the variational nonholonomic equations.

A Hamiltonian formulation of the variational nonholonomic equations is to consider

$$J\dot{z} = \nabla H(z), \quad z \in C \subset \mathbb{R}^n \tag{2.1}$$

where $z\in\mathbb{R}^n$, $\omega := \frac{1}{2} dz \wedge Jdz$ is a closed 2-form[3], $H\colon \mathbb{R}^n \to R$ is a Hamiltonian, and $C$ is a constraint submanifold such that $i^*\omega$ (where $i\colon C\to\mathbb{R}^n$ is the inclusion of $C$ in $\mathbb{R}^n$, and the $*$ denotes the *pull-back*) is nondegenerate, i.e., such that $(C, i^*\omega)$ is a symplectic manifold. The dynamics on $C$ depends only on the restricted Hamiltonian $i^*H$ and restricted symplectic form $i^*\omega$. Indeed, systems with holonomic constraints also take this form, with $z=(q,p)$, $\omega = dq\wedge dp$, and $C=\{(q,p)\colon h_i(q)=0,\ Dh_i(q)H_p(q,p)=0,\ 1\le$

[1] Also known as Carnot geometry in France and nonholonomic Riemannian geometry in Russia [77]; the metric induced by sub-Riemannian geodesics is known as the Carnot-Carathéodory metric.

[2] (i): $g(q,\dot{q})=0$ iff $\exists_u\colon \dot{q}=f(q,u)$, (ii): $L(q,f(q,u))=h(q,u)$, and (iii): $\exists!_{u^*}\colon \frac{\partial \hat{H}}{\partial u}(q,p,u^*)=0$ where $\hat{H}(q,p,u)=\langle p, f(q,u)\rangle - h(q,u)$.

[3] Vector notation is used in wedge products, writing $dq\wedge dp$ for $\sum_{i=1}^n dq_i\wedge dp_i$ and $dz\wedge Jdz$ for $\sum_{i,j=1}^n J_{ij} dz_i \wedge dz_j$, where the dimension $n$ is determined from the context.

$i \leq k\}$ consisting of primary and secondary constraints; a nondegeneracy assumption ensures that $C$ is symplectic. The widely used RATTLE method [39, 47] provides a (class of) symplectic integrators for this case when $J$ is constant: it integrates in coordinates $z$ with Lagrange multipliers to enforce the constraints. However, it is striking that there are no known symplectic integrators for general constrained Hamiltonian systems of the form (2.1).

This chapter develops a class of symplectic integrators for a class of Hamiltonian systems of the form (2.1) containing constraints that can depend on both position and velocity. The systems are those of index 1 and the integrators are given in Propositions 2.2.2 and 2.3.1 which are the main results. The class includes the Hamiltonian description of the variational nonholonomic equations, including the sub-Riemannian geodesic equations, and this application is given first, in Propositions 2.1.1 and 2.1.2, as it motivates the consideration of index 1 Hamiltonian systems. The construction is generalized to include both holonomic and nonholonomic constraints in Section 2.4. Sample applications are given to calculating the sub-Riemannian geodesics of a wheeled vehicle (the 'parallel parking' problem) in Section 2.5 and of the Heisenberg group in Section 2.6.

In the following proposition, the linear independence assumption on the constraints is equivalent to constraining the velocities to lie in an $(n-k)$-dimensional distribution of the tangent space of the positions[4].

**Proposition 2.1.1.** *Let $M$ be a symmetric nonsingular $n \times n$ mass matrix, $V\colon \mathbb{R}^n \to \mathbb{R}$ a smooth potential, $g_i\colon \mathbb{R}^n \to \mathbb{R}^n$, $i = 1, \ldots, k$ be smooth functions whose values are linearly independent for all arguments, and $q$ be a smooth extremal with fixed endpoints for the functional*

$$S(q) = \int_{t_0}^{t_1} L(t, q, \dot{q}) dt = \int_{t_0}^{t_1} \left( \frac{1}{2} \dot{q}^T M \dot{q} - V(q) \right) dt \tag{2.2}$$

*subject to the constraints $g_i(q) \cdot \dot{q} = 0, i = 1, \ldots, k$. Then*

$$J\dot{z} = \nabla H(z) \tag{2.3}$$

[4]Velocities are always tangent to positions, the dimension is $n-k$ because there are $k$ independent $g_i$ constraint functions.

*where*

$$J = \begin{pmatrix} 0 & -\mathbb{I}_{n\times n} & 0 \\ \mathbb{I}_{n\times n} & 0 & 0 \\ 0 & 0 & 0_{k\times k} \end{pmatrix}, \quad z = \begin{pmatrix} q \\ p \\ \lambda \end{pmatrix} \in \mathbb{R}^{2n+k},$$

$$p = M\dot{q} - \sum_{i=1}^{k} \lambda_i g_i(q),$$

$$H(z) = \frac{1}{2}\left(p + \sum_{i=1}^{k} \lambda_i g_i(q)\right)^T M^{-1} \left(p + \sum_{i=1}^{k} \lambda_i g_i(q)\right) + V(q),$$

*and, furthermore, the Euler–Lagrange equations for (2.2) are equivalent to the generalized Hamiltonian system (2.3). Equation* (2.3) *forms a constrained Hamiltonian system of the type (2.1) with constraint submanifold $C$ a graph over $(q,p)$, i.e., $C := \{(q,p,\lambda) : \lambda = \tilde{\lambda}(q,p)\}$ and restricted symplectic form $i^*\omega = dq \wedge dp$.*

*Proof.* Introducing Lagrange multipliers $\lambda_1, \ldots, \lambda_k$, the Euler–Lagrange equations for (2.2) are

$$\frac{d}{dt}\left(\nabla_{\dot{q}} F\right) - \nabla_q F = 0, \tag{2.4}$$

$$g_i(q) \cdot \dot{q} = 0, \quad i = 1, \ldots, k, \tag{2.5}$$

where

$$F(q, \dot{q}, \lambda) = \frac{1}{2}\dot{q}^T M \dot{q} - V(q) - \sum_{i=1}^{k} \lambda_i g_i(q) \cdot \dot{q}.$$

Expanding out equation (2.4) gives the Euler–Lagrange equations

$$\frac{d}{dt}\left(M\dot{q} - \sum_{i=1}^{k} \lambda_i g_i(q)\right) - \nabla_q F = 0,$$

$$\frac{d}{dt}\left(M\dot{q} - \sum_{i=1}^{k} \lambda_i g_i(q)\right) + \left(\nabla V(q) + \sum_{i=1}^{k} \lambda_i D g_i(q)\dot{q}\right) = 0. \tag{2.6}$$

Define the conjugate momentum $p \in \mathbb{R}^n$ using the standard Legendre transform

$$p := \nabla_{\dot{q}} F = M\dot{q} - \sum_{i=1}^{k} \lambda_i g_i(q) \tag{2.7}$$

so that

$$\dot{q} = M^{-1}\left(p + \sum_{i=1}^{k} \lambda_i g_i(q)\right). \tag{2.8}$$

Using equations (2.7) and (2.8) in equation (2.4) gives

$$\dot{p} = -\nabla V(q) - \sum_{i=1}^{k} \lambda_i Dg_i(q) M^{-1} \left( p + \sum_{j=1}^{k} \lambda_j g_j(q) \right). \tag{2.9}$$

Defining $H(q,p,\lambda) := \dot{q} \cdot p - F(q,\dot{q},\lambda)$ gives

$$\begin{aligned}
H &= \dot{q} \cdot p - F(q,\dot{q},\lambda) \\
&= \dot{q} \cdot p - \frac{1}{2}\dot{q}^T M \dot{q} + V(q) + \sum_{i=1}^{k} \lambda_i g_i(q) \cdot \dot{q} \\
&= \dot{q} \cdot \left( p - \frac{1}{2} M\dot{q} + \sum_{i=1}^{k} \lambda_i g_i(q) \right) + V(q) \\
&= \dot{q} \cdot \left( M\dot{q} - \sum_{i=1}^{k} \lambda_i g_i(q) - \frac{1}{2} M\dot{q} + \sum_{i=1}^{k} \lambda_i g_i(q) \right) + V(q) \\
&= \dot{q} \cdot \left( \frac{1}{2} M\dot{q} \right) + V(q) \\
&= \frac{1}{2} \left( p + \sum_{i=1}^{k} \lambda_i g_i(q) \right)^T M^{-1} \left( p + \sum_{i=1}^{k} \lambda_i g_i(q) \right) + V(q).
\end{aligned}$$

A calculation shows the equivalence of the right hand side of (2.8) and $\nabla_p H$; of the right hand side of (2.9) and $-\nabla_q H(q,p,\lambda)$; and of constraints $g_i(q) \cdot \dot{q} = 0$ and $0 = \nabla_\lambda H(q,p,\lambda)$.

The constraints $0 = \nabla_\lambda H(q,p,\lambda)$ are the following set of equations linear in $\lambda$:

$$\begin{pmatrix} g_1 \cdot M^{-1} g_1 & \cdots & g_1 \cdot M^{-1} g_k \\ \vdots & \ddots & \vdots \\ g_k \cdot M^{-1} g_1 & \cdots & g_k \cdot M^{-1} g_k \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = - \begin{pmatrix} g_1 \cdot M^{-1} p \\ \vdots \\ g_k \cdot M^{-1} p \end{pmatrix} \tag{2.10}$$

which has a unique solution for $\lambda$ for all $q$, $p$ because the matrix is $GM^{-1}G^T$ where $G$ is the $k \times n$ matrix whose $i$th row is $g_i^T$. The assumption that the $g_i$ are linearly independent means that $G$ has full rank $k$ and hence that $GM^{-1}G^T$ is nonsingular. The constraints therefore have a unique solution for $\lambda$ that can be written as $\lambda = \tilde{\lambda}(q,p)$, that is, the constraint submanifold is a graph over $(q,p)$. Differentiating these constraints with respect to $t$ then yields ODEs for $\dot{\lambda}$, that is, the system (2.3) has (differentiation) index 1. The symplectic form on $C$ is $\frac{1}{2} dz \wedge J dz = dq \wedge dp$. □

Although Proposition 2.1.1 is not original, it should be emphasised that the usual treatment is to go one step further and eliminate the Lagrange multipliers $\lambda$ to get a canonical Hamiltonian system in $(q,p)$ ([7], Thm. 7.3.1). This step may not be

desirable either analytically or numerically.

Under certain conditions, namely that the Legendre transform that defines the conjugate momenta is invertible to give $\dot{q}$, Proposition 2.1.1 can be generalized to allow a general Lagrangian and general constraints. A very thorough geometric treatment of this type of constraint, applying the Gotay–Nestor geometric version of the Dirac–Bergmann constraint algorithm, can be found in [42].

**Proposition 2.1.2.** *If the Legendre transform mapping $(\dot{q}, q, \lambda) \to (p, q, \lambda)$ given in equation (2.12) is invertible then the Euler–Lagrange equations for the action*

$$S(q) = \int_{t_0}^{t_1} L(t, q, \dot{q}) dt$$

*subject to the constraints $g_i(q, \dot{q}) = 0, i = 1, \ldots, k$ are equivalent to the generalized Hamiltonian system*

$$J\dot{z} = \nabla H(z) \tag{2.11}$$

*where*

$$J = \begin{pmatrix} 0 & -\mathbb{I}_{n\times n} & 0 \\ \mathbb{I}_{n\times n} & 0 & 0 \\ 0 & 0 & 0_{k\times k} \end{pmatrix}, \quad z = \begin{pmatrix} q \\ p \\ \lambda \end{pmatrix}, \quad p = \nabla_{\dot{q}} F(q, \dot{q}, \lambda), \tag{2.12}$$

$$H(z) = \dot{q} \cdot p - F(q, \dot{q}, \lambda), \quad F(q, \dot{q}, \lambda) = L(t, q, \dot{q}) - \sum_{i=1}^{k} \lambda_i g_i(q, \dot{q}).$$

*If, in addition, the matrix $G(q, \dot{q})$ given by $G_{ij} = \partial g_i(q, \dot{q})/\partial \dot{q}_j$ has full rank $k$ for all $q$, $\dot{q}$, then the system* (2.11) *has index 1, i.e., can be solved for $\lambda = \tilde{\lambda}(q, p)$.*

*Proof.* The Euler-Lagrange equations are

$$\frac{\partial}{\partial t} (\nabla_{\dot{q}} F) - \nabla_q F = 0, \tag{2.13}$$

$$g_i(q, \dot{q}) = 0, \quad i = 1, \ldots, k, \tag{2.14}$$

where

$$F(q, \dot{q}, \lambda) = L(t, q, \dot{q}) - \sum_{i=1}^{k} \lambda_i g_i(q, \dot{q}). \tag{2.15}$$

Define the conjugate momentum $p \in \mathbb{R}^n$ using the standard Legendre transform

$$p := \nabla_{\dot{q}} F. \tag{2.16}$$

By assumption, equation (2.16) can be rearranged to give $\dot{q}$

$$\dot{q} = f(q, p, \lambda). \tag{2.17}$$

Using equation (2.16) in equation (2.13) gives the expression for $\dot{p}$

$$\dot{p} = \nabla_q F. \tag{2.18}$$

Define the Hamiltonian as

$$\begin{aligned} H(q, p, \lambda) &:= \dot{q} \cdot p - F(q, \dot{q}, \lambda) \\ &= p \cdot f(q, p, \lambda) - F(q, \dot{q}, \lambda). \qquad \text{using (2.17)} \end{aligned} \tag{2.19}$$

Then (where $f_p$ is the Jacobian derivative $\partial f_i(q, p, \lambda)/\partial p_j$)

$$\begin{aligned} \nabla_p H &= f + f_p p - f_p \nabla_{\dot{q}} F \\ &= f + f_p p - f_p p \qquad \text{using (2.16)} \\ &= \dot{q} \qquad \text{using (2.17)} \end{aligned} \tag{2.20}$$

$$\begin{aligned} \nabla_q H &= f_q p - \nabla_q F - f_q \nabla_{\dot{q}} F \\ &= f_q p - \nabla_q F - f_q p \qquad \text{using (2.16)} \\ &= -\dot{p} \qquad \text{using (2.18)} \end{aligned} \tag{2.21}$$

$$\begin{aligned} \nabla_\lambda H &= f_\lambda p - f_\lambda \nabla_{\dot{q}} F + g \\ &= f_\lambda p - f_\lambda p + g \qquad \text{using (2.16)} \\ &= 0 \qquad \text{using (2.14)} \end{aligned} \tag{2.22}$$

establishing the equivalence of the Euler–Lagrange equations to (2.11). The constraints in the Hamiltonian formulation are

$$0 = \nabla_\lambda H(q, p, \lambda) = g(q, f(q, p, \lambda))$$

and their Jacobian derivative with respect to $\lambda$ is the matrix

$$\sum_{j=1}^{n} \frac{\partial g_i}{\partial \dot{q}_j} \frac{\partial \dot{q}_j}{\partial \lambda_k}.$$

The first factor is $G$. The second factor is the derivative of the inverse Legendre transform (assumed invertible) with respect to $\lambda$ ($f_\lambda$). The forward Legendre transform is $p = \frac{\partial L}{\partial \dot{q}} + \sum_{i=1}^{k} \lambda_i \frac{\partial g_i}{\partial \dot{q}}$ and its derivative with respect to $\lambda$ is $G^T$. By the chain rule[5],

[5] $f^{-1}(f(x)) = x \Rightarrow (f^{-1})_\lambda f_\lambda = \mathbb{I}$, so $(f_\lambda)^{-1}$ exists if $f^{-1}$ exists.

if $G$ has rank $k$ then the Jacobian is nonsingular for all $q$, $\dot{q}$ and the constraints have a unique solution $\lambda = \tilde{\lambda}(q,p)$ for all $q,p$, that is, the system has index 1. □

## 2.2 Symplectic integrators for generalized Hamiltonian systems

The Hamiltonian form (2.3) suggests considering generalized Hamiltonian systems of the form

$$J\dot{z} = \nabla H(z), \quad z \in \mathbb{R}^n, \tag{2.23}$$

where $J$ is a constant antisymmetric matrix, and the constraints are not specified. Note that many kinds of constrained Hamiltonian systems (including those with holonomic constraints) can be written in this form; the constraint manifold $C$ is constructed as the subset of initial conditions for which the equations have a solution. In general, these equations may not have solutions for all initial conditions; in the extreme case $J = 0$, the equations are purely algebraic. However, it is easily seen that any solutions that do exist preserve the ('pre-symplectic') 2-form $\frac{1}{2}dz \wedge Jdz$, which is degenerate when $J$ is singular; this does not require the invertibility of $J$.

**Lemma 2.2.1.** *Any solutions to* (2.23) *preserve the 2-form* $\frac{1}{2}dz \wedge Jdz$.

*Proof.*

$$\begin{aligned}\tfrac{1}{2}(dz \wedge Jdz)_t &= \tfrac{1}{2}(dz \wedge Jdz_t + dz_t \wedge Jdz)\\ &= dz \wedge Jdz_t\\ &= dz \wedge H_{zz}(z)dz\\ &= 0.\end{aligned}$$

□

In the particular case of Proposition 2.1.1, the generalized Hamiltonian system that is obtained is equivalent to a canonical Hamiltonian system obtained by eliminating the Lagrange multipliers $\lambda$. Let $\lambda = \tilde{\lambda}(q,p)$ be the solution to (2.10), and $\tilde{H}(q,p) :=$

$H(q,p,\tilde{\lambda}(q,p))$. Then

$$\begin{aligned}\frac{\partial \tilde{H}}{\partial p_i}(q,p) &= \frac{\partial H}{\partial p_i}(q,p,\tilde{\lambda}(q,p)) + \sum_{j=1}^{k} \frac{\partial H}{\partial \lambda_j}(q,p,\tilde{\lambda}(q,p))\frac{\partial \tilde{\lambda}_j}{\partial p_i}(q,p) \\ &= \frac{\partial H}{\partial p_i}(q,p,\tilde{\lambda}(q,p)) \\ &= \dot{q}_i \\ \text{and} \quad \frac{\partial \tilde{H}}{\partial q_i}(q,p) &= \frac{\partial H}{\partial q_i}(q,p,\tilde{\lambda}(q,p)) + \sum_{j=1}^{k} \frac{\partial H}{\partial \lambda_j}(q,p,\tilde{\lambda}(q,p))\frac{\partial \tilde{\lambda}_j}{\partial q_i}(q,p) \\ &= \frac{\partial H}{\partial q_i}(q,p,\tilde{\lambda}(q,p)) \\ &= -\dot{p}_i\end{aligned}$$

which are equivalent to (2.3). That is, the two operations of eliminating the Lagrange multipliers and mapping the Hamiltonian to its Hamiltonian vector field commute.

Certain Runge–Kutta methods, e.g. the midpoint rule, are known to be symplectic when the structure matrix $J$ is invertible [39]. However, it will now be shown that $J$ need not be invertible for such methods to be symplectic. This new result is analogous to the new continuous time result given in Theorem 1.2.8.

**Proposition 2.2.2.** *Any solutions of any symplectic Runge–Kutta method applied to $J\dot{z} = \nabla H$ preserve the 2-form $\frac{1}{2}dz \wedge Jdz$, where $J$ is any constant antisymmetric matrix.*

*Proof.* The $s$ stage symplectic Runge–Kutta method is

$$JZ_i = Jz_0 + \Delta t \sum_{j=1}^{s} a_{ij} JF_j, \tag{2.24}$$

$$Jz_1 = Jz_0 + \Delta t \sum_{j=1}^{s} b_j JF_j, \tag{2.25}$$

where

$$JF_j = \nabla H(Z_j). \tag{2.26}$$

Here $\Delta t$ is the time step and the method maps $z_0$ to $z_1$. The coefficients of a symplectic Runge–Kutta method obey

$$b_i b_j - b_j a_{ji} - b_i a_{ij} = 0. \tag{2.27}$$

Taking the exterior derivative of equations (2.24), (2.25), and (2.26) gives

$$Jdz_0 = JdZ_i - \Delta t \sum_{j=1}^{s} a_{ij} JdF_j, \tag{2.28}$$

$$Jdz_1 = Jdz_0 - \Delta t \sum_{j=1}^{s} b_j JdF_j, \tag{2.29}$$

$$JdF_j = H_{zz}(Z_j) dZ_j. \tag{2.30}$$

From equation (2.30),

$$dZ_j \wedge JdF_j = dZ_j \wedge H_{zz}(Z_j) dZ_j = 0. \tag{2.31}$$

Substitution now gives, in the same way as in the original study of B-stability by Burrage and Butcher [10],

$$\begin{aligned}
&dz_1 \wedge Jdz_1 - dz_0 \wedge Jdz_0 \\
&\quad = dz_1 \wedge J(dz_0 + \Delta t \sum_{j=1}^{s} b_j dF_j) - dz_0 \wedge Jdz_0 \text{ (using (2.29))} \\
&\quad = -Jdz_1 \wedge \left( dz_0 + \Delta t \sum_{j=1}^{s} b_j dF_j \right) - dz_0 \wedge Jdz_0 \\
&\quad = -\left( Jdz_0 + \Delta t \sum_{j=1}^{s} b_j JdF_j \right) \wedge \left( dz_0 + \Delta t \sum_{j=1}^{s} b_j dF_j \right) - dz_0 \wedge Jdz_0 \\
&\quad = \Delta t dz_0 \wedge J \sum_{j=1}^{s} b_j dF_j + \Delta t \sum_{j=1}^{s} b_j dF_j \wedge Jdz_0 + \Delta t^2 \sum_{j=1}^{s} b_j dF_j \wedge J \sum_{j=1}^{s} b_j dF_j \\
&\quad = -\Delta t \sum_{j=1}^{s} b_j Jdz_0 \wedge dF_j + \Delta t \sum_{i=1}^{s} b_i dF_i \wedge Jdz_0 + \Delta t^2 \sum_{j=1}^{s} b_j dF_j \wedge J \sum_{j=1}^{s} b_j dF_j.
\end{aligned}$$

Now, use (2.28) to substitute in for $Jdz_0$ in the first two sums,

$$\begin{aligned}
&dz_1 \wedge Jdz_1 - dz_0 \wedge Jdz_0 \\
&\quad = -\Delta t \sum_{j=1}^{s} b_j J \left( dZ_j - \Delta t \sum_{i=1}^{s} a_{ji} dF_i \right) \wedge dF_j \\
&\qquad + \Delta t \sum_{i=1}^{s} b_i dF_i \wedge J \left( dZ_i - \Delta t \sum_{j=1}^{s} a_{ij} dF_j \right) \\
&\qquad + \Delta t^2 \sum_{j=1}^{s} b_j dF_j \wedge J \sum_{j=1}^{s} b_j dF_j \\
&\quad = -\Delta t^2 \left[ \sum_{j,i=1}^{s,s} b_j a_{ji} dF_i \wedge JdF_j + \sum_{i,j=1}^{s,s} b_i a_{ij} dF_i \wedge JdF_j \right] \text{ (using (2.31))} \\
&\qquad + \Delta t^2 \sum_{j=1}^{s} b_j dF_j \wedge J \sum_{j=1}^{s} b_j dF_j \\
&\quad = \Delta t^2 \left[ \sum_{i,j=1}^{s,s} \left( b_i b_j - b_j a_{ji} - b_i a_{ij} \right) dF_i \wedge JdF_j \right] \\
&\quad = 0 \quad \text{(using (2.27))}.
\end{aligned}$$

This establishes the proposition. □

Note that the underlying structure can be seen very clearly in the case of the midpoint rule

$$\frac{Jz_1 - Jz_0}{\Delta t} = \nabla H \left( \frac{z_0 + z_1}{2} \right) := \nabla H(\bar{z}). \tag{2.32}$$

for which

$$\begin{aligned}
dz_1 \wedge Jdz_1 - dz_0 \wedge Jdz_0 &= (dz_0 + dz_1) \wedge J(dz_1 - dz_0) \\
&= (dz_0 + dz_1) \wedge \frac{1}{2} \Delta t H_{zz}(\bar{z})(dz_0 + dz_1) \\
&= 0.
\end{aligned} \tag{2.33}$$

A full study of the geometry of the relations $(z_0, z_1)$ generated in Proposition 2.2.2 remains to be undertaken. Unfortunately, the relations $(z_0, z_1)$ in Proposition 2.2.2 do not yield good integrators for arbitrary $J$ and $H$. For example, holonomic constraints can be specified as generalized Hamiltonian systems with $H = \tilde{H}(q,p) + \sum_{i=1}^{k} \lambda_i h_i(q)$. In this case the midpoint rule, say, generates maps from *all* $(q_0, p_0)$ to $(q_1, p_1)$ with the constraints satisfied at the midpoint[6]. Not only is the phase space 'wrong', this method

[6]The third component ($\lambda$) of (2.32) gives $0 = \nabla_\lambda H(\bar{z}) = (h_1(\bar{q}), \ldots, h_k(\bar{q}))$.

is known to be not convergent in general [24]. The situation is much better for index 1 constraints.

## 2.3 Symplectic integrators for index 1 constraints

**Proposition 2.3.1.** *Let $J$ be any constant antisymmetric matrix and let $H$ be a Hamiltonian such that the generalized Hamiltonian system*

$$J\dot{z} = \nabla H(z) \tag{2.34}$$

*has index 1, i.e., such that when the system is written in Darboux coordinates $(q, p, \lambda)$, the constraint $H_\lambda = 0$ has a unique solution for $\lambda$ for all $q$ and $p$. Then any symplectic Runge–Kutta method (2.24)–(2.27) applied to (2.34) is well-defined for sufficiently small $\Delta t$, convergent of the same order as the Runge–Kutta method, preserves the constraint submanifold, and preserves the symplectic form on the constraint submanifold.*

*Proof.* By linear covariance of Runge–Kutta methods[7] $J$ is assumed to be in Darboux form (although the theorem holds in any basis). Then the constraint part of the Runge–Kutta equations read

$$\begin{aligned} 0 &= \nabla_\lambda H(Q_i, P_i, \Lambda_i),\ i = 1, \ldots, k, \\ 0 \frac{\lambda_1 - \lambda_0}{\Delta t} &= \sum_{i=1}^{s} b_i \nabla_\lambda H(Q_i, P_i, \Lambda_i), \end{aligned}$$

where $Q_i$, $P_i$, and $\Lambda_i$ are the stage variables. Therefore the Lagrange multipliers $\Lambda_i$ at each stage are given by the *exact* Lagrange multipliers evaluated at $(Q_i, P_i)$, i.e. $\Lambda_i = \tilde{\lambda}(Q_i, P_i)$, and $\lambda_1$ is arbitrary. For convenience, the extra equations $\lambda_0 = \tilde{\lambda}(q_0, p_0)$, $\lambda_1 = \tilde{\lambda}(q_1, p_1)$ are added; these do not affect the method at all. The resulting method is equivalent to that obtained by eliminating the Lagrange multipliers in the Hamiltonian, applying a symplectic Runge–Kutta method, and lifting back to the constraint manifold by $\lambda = \tilde{\lambda}(q, p)$. It is therefore well-defined for sufficiently small $\Delta t$ and convergent of the same order as the Runge–Kutta method. Because $\frac{1}{2} dz \wedge J dz = dq \wedge dp$, the symplectic form $dq \wedge dp$ is preserved on the constraint manifold. □

Note that the assumptions are satisfied if $|H_{\lambda\lambda}| \neq 0$. The constraints may be nonlinear in $\lambda$, and need not be solved analytically; the entire Runge–Kutta system for $(Q_i, P_i, \Lambda_i)$ can be numerically solved simultaneously.

[7] The method is invariant under a change of variables

**Corollary 2.3.2.** *Symplectic Runge–Kutta methods yield convergent constraint-preserving symplectic integrators for the Hamiltonian formulation of the Lagrange and sub-Riemannian problems given in Propositions 2.1.1 and 2.1.2. When velocities are calculated using the Legendre transform, the constraints $g_i(q)\cdot\dot{q}=0$ (resp. $g(q,\dot{q})=0$) are satisfied exactly at the stages, and if the endpoint Lagrange multipliers are defined by $H_\lambda(q_n,p_n,\lambda_n)=0$, then the constraints are satisfied exactly at the endpoints.*

## 2.4 Variational problems with holonomic and nonholonomic constraints

Proposition 2.1.1 allowed a nonholonomic variational problem to be converted into an index 1 constrained Hamiltonian system that can be integrated using the symplectic midpoint rule. If *holonomic* constraints are added to the original variational problem, it is shown here that the resulting Hamiltonian system is a simple *holonomically* constrained system. This system can be solved by a symplectic method such as RATTLE [47, 68].

**Proposition 2.4.1.** *Let $M$ be a symmetric nonsingular $n\times n$ mass matrix, $V\colon\mathbb{R}^n\to\mathbb{R}$ a smooth potential, $g_i\colon\mathbb{R}^n\to\mathbb{R}^n$, $i=1,\dots,k$ be $k$ smooth functions , and $q$ be a smooth extremal with fixed endpoints for the functional*

$$S(q)=\int_{t_0}^{t_1}L(t,q,\dot{q})dt=\int_{t_0}^{t_1}\left(\frac{1}{2}\dot{q}^TM\dot{q}-V(q)\right)dt \tag{2.35}$$

*subject to the* velocity *constraints $g_i(q)\cdot\dot{q}=0, i=1,\dots,k$ and the* holonomic *constraints $h_i(q)=0, i=1,\dots,l$. Then*

$$J\dot{z}=\nabla H(z) \tag{2.36}$$

*where*

$$J=\begin{pmatrix}0&-\mathbb{I}_{n\times n}&0&0\\ \mathbb{I}_{n\times n}&0&0&0\\ 0&0&0_{k\times k}&0\\ 0&0&0&0_{l\times l}\end{pmatrix},\quad z=\begin{pmatrix}q\\p\\\lambda\\\lambda^h\end{pmatrix},$$

$$p=M\dot{q}-\sum_{i=1}^{k}\lambda_ig_i(q),$$

$$H(z)=\frac{1}{2}\left(p+\sum_{i=1}^{k}\lambda_ig_i(q)\right)^TM^{-1}\left(p+\sum_{i=1}^{k}\lambda_ig_i(q)\right)+V(q)+\sum_{i=1}^{l}\lambda_i^hh_i(q),$$

*and, furthermore, the Euler–Lagrange equations for (2.35) are equivalent to the generalized Hamiltonian system (2.36). If, in addition, the velocity constraints are linearly independent for all $q$, then* (2.36) *is equivalent to a canonical holonomically constrained Hamiltonian system.*

*Proof.* Similarly to Proposition 2.1.1 the extended Lagrangian $F$, the conjugate momenta $p$, and the Hamiltonian $H(q, p, \lambda, \lambda^h)$ are defined by

$$\begin{aligned} F &:= \frac{1}{2}\dot{q}^T M\dot{q} - V(q) - \sum_{i=1}^{k} \lambda_i g_i(q) \cdot \dot{q} - \sum_{i=1}^{l} \lambda_i^h h_i(q), \\ p &:= \nabla_{\dot{q}} F = M\dot{q} - \sum_{i=1}^{k} \lambda_i g_i(q), \\ H &:= \dot{q} \cdot p - F. \end{aligned}$$

The rest of the proof is a calculation along the same lines as for Proposition 2.1.1. □

**Proposition 2.4.2.** *Subject to standard nondegeneracy assumptions on the Hamiltonian, the following algorithm yields a convergent, second-order integrator that is symplectic on the constraint manifold defined by the (primary) holonomic constraints and the secondary constraints induced by them: (i) apply* RATTLE *using the holonomic constraints; (ii) in the inner step of* RATTLE*, when a time step of the unconstrained system is required, apply the midpoint rule to the generalized Hamiltonian system with Hamiltonian $H(q, p, \lambda, 0)$.*

*Proof.* Eliminating the velocity constraints by solving for the Lagrange multipliers $\lambda_i$ yields a standard holonomically constrained system. Applying RATTLE (with the midpoint rule in the inner step) to this system yields a convergent second order integrator on the constraint surface. Applying the midpoint rule in the inner step is equivalent to applying the midpoint rule to the generalized Hamiltonian system with Hamiltonian $H(q, p, \lambda, 0)$. □

## 2.5 Example: sub-Riemannian geodesics

Trajectories of a two-wheeled vehicle with a front steering wheel and a non-steering back wheel, moving on a smooth surface, will be modelled. This two-wheeled vehicle can be considered an idealized car, or considering it only has two wheels, a bicycle. Consider the two-wheeled vehicle shown in Fig. 2.1 with length $\ell$, back wheel at $(z, w)$, and front wheel at $(x, y)$. The front wheel is at an angle $\phi$ and the vehicle is at an angle $\theta$.

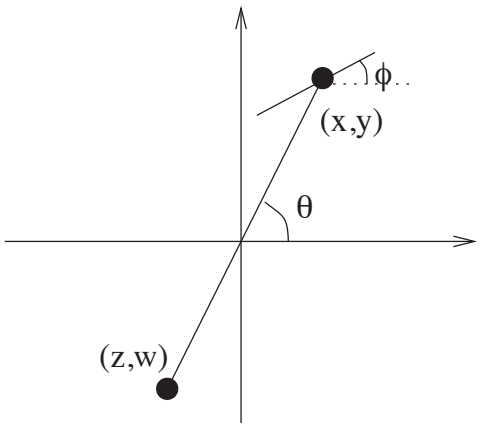



Figure 2.1: A two wheeled vehicle showing the front wheel angle $\phi$ and the vehicle angle of $\theta$.

If the speed of the front wheel is $v$, its velocity of the front wheel must obey

$$\dot{x} = v\cos\phi, \qquad \dot{y} = v\sin\phi.$$

Eliminating $v$, the velocity of the front wheel obeys the constraint

$$\dot{x}\sin\phi - \dot{y}\cos\phi = 0. \tag{2.37}$$

Similarly, the velocity of the back wheel obeys the constraint

$$\dot{z}\sin\theta - \dot{w}\cos\theta = 0. \tag{2.38}$$

Equation (2.38), and thus the variables $z$ and $w$, can be eliminated using the distance between the two wheels, $\ell$, which relates the four variables. Notice that

$$x - z = \ell\cos\theta, \qquad y - w = \ell\sin\theta$$

so that

$$\dot{z} = \dot{x} + \ell\dot{\theta}\sin\theta, \qquad \dot{w} = \dot{y} - \ell\dot{\theta}\cos\theta$$

which substituted into (2.38) gives

$$\dot{x}\sin\theta - \dot{y}\cos\theta + \ell\dot{\theta} = 0. \tag{2.39}$$

The Lagrangian is

$$L = \frac{1}{2}\left(\dot{x}^2 + \dot{y}^2 + \alpha\dot{\theta}^2 + \beta\dot{\phi}^2\right) - V(x,y) \tag{2.40}$$

where the potential $V(x, y)$ is the (scaled) height of the surface, which together with the constraints (2.37) and (2.39) gives an index 1 system as in Proposition 2.1.1. That is, when $V = 0$ the geodesics of the sub-Riemannian metric defined by (2.40), (2.37), and (2.39) are being calculated. To put it another way, the problem is to seek the shortest paths that move the vehicle from one configuration to another subject to the constraints of its geometry—the 'parallel parking' problem. In the numerics, the midpoint rule is used.

The potential $V(q) = -\cos r$, where $r$ is the midpoint of the vehicle, was first used to numerically compare the second-order convergence of the method, to a reference solution computed by MATLAB's `ode15s`, numerical conservation of the symplectic form, exact conservation of the original constraints (up to round-off error), and behaviour of the energy error. A sample result is shown in Fig. 2.2, from which the energy errors appear to be bounded, as expected for a symplectic integrator.

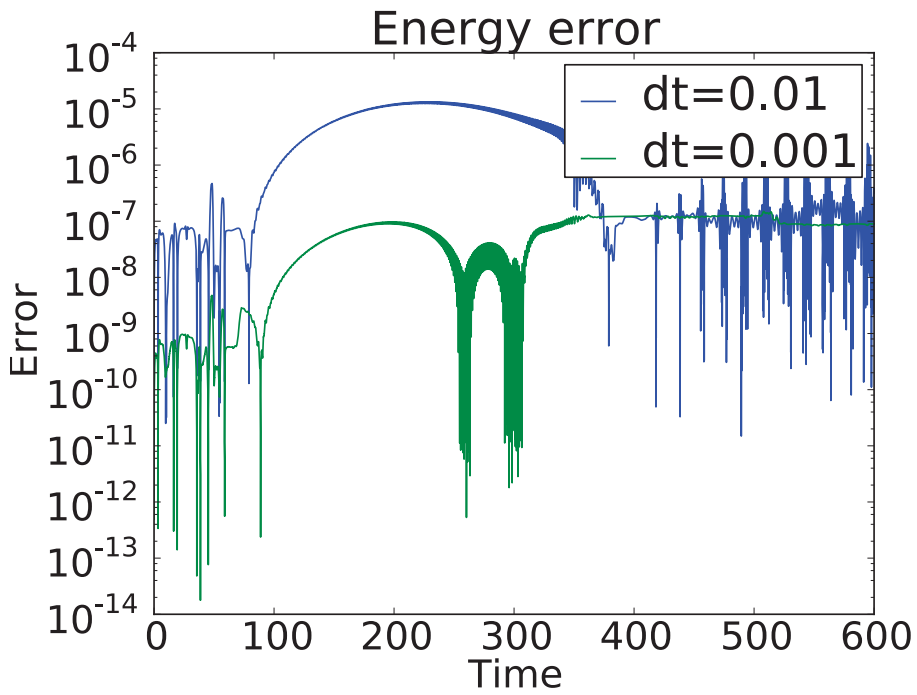



Figure 2.2: The energy error over time. This is the energy at each step minus the initial energy. The vehicle is trapped in the potential bowl $-V(\mathbf{q})$, and the energy error does not show a linear growth in time.

Then the special solutions of the free motion case $V = 0$ were studied. There are two simple solutions that are relative equilibria for the translation and rotation symmetries of the problem: namely straight line and circular motion. If the vehicle starts with $\theta = \phi = 0$, the exact solution is a straight line motion. Let $\theta = \phi = 0$, $\dot{x} = 1$, and $\dot{y} = 0$. The constraints in equations (2.37) and (2.39) are satisfied. Equation (2.8) gives the

initial generalized momenta values: all are zero except $p_x = 1$. The discretization gives the exact solution. However, as the solution is unstable, round-off errors eventually cause the vehicle to wander.

For the circular motion, let $\theta = at$, $\phi = at + \frac{\pi}{2}$, $\dot{x} = -c\sin(\theta)$, and $\dot{y} = c\cos(\theta)$. There are two constants, $a$ and $c$, to be determined. Equation (2.37) gives $\lambda = (1, -c)$, and equation (2.39) gives $a\ell = c$. Using these values in equation (2.7) gives the initial generalized momenta values: $(p_x, p_y, p_\theta, p_\phi) = (0, 0, a(1 + \ell^2), a)$. For this simple trajectory $a$ is chosen to be 1. In Fig. 2.3 the circle trajectory of the vehicle is confirmed. If the trajectory is computed for larger times the vehicle leaves the circle; the solution

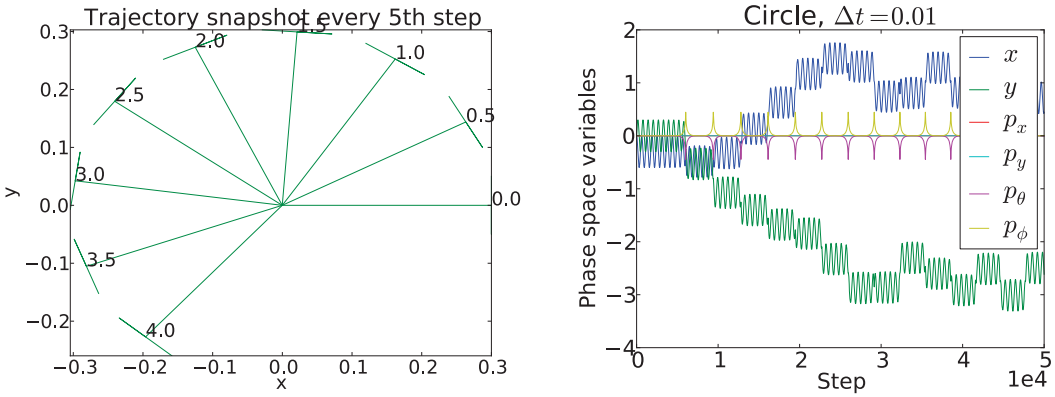



Figure 2.3: Snapshots taken every fifth step of the two-wheeled vehicle starting at $(x, y, \theta, \phi, p_x, p_y, p_\theta, p_\phi) = (0.3, 0, 0, \pi, 0, 1.09, 1)$, with $\Delta t = 0.1$, and zero potential. The vehicle stays in a circle for many revolutions (not shown for clarity), but the geodesics is not stable, so eventually the vehicle wanders, as shown in the longer orbit on the right. The solutions for $p_\theta$ and $p_\phi$ suggest a relative homoclinic orbit.

appears to be unstable, but, interestingly, appears to repeatedly return to the circular orbit, indicating a possible relative homoclinic structure in this problem.

## 2.6 Example: the Heisenberg problem

A previous study of geometric integrators for sub-Riemannian variational problems used a discrete variational approach to obtain constrained symplectic integrators [6]. The approach here, applying symplectic integrators to the Hamiltonian formulation, yields geometric integrators with the same geometric properties, but uses standard integrators that allow any order with standard implementations, and does not require an approximation of $\dot{q}$, that is, it naturally yields first-order trajectories in $(q, p)$ instead of second-order trajectories in $q$.

What follows is a repeat of the numerical illustration of [6, p. 12], the Heisenberg problem, using the Hamiltonian formulation and the midpoint rule. This is to find the

extremal $q(t) = (x(t), y(t), z(t))$ of

$$S(q) = \int_{t_0}^{t_1} L(t, q, \dot{q}) dt = \int_{t_0}^{t_1} \left( \frac{1}{2} \dot{q}^T \dot{q} - V(q) \right) dt$$

subject to the constraint $g(q) \cdot \dot{q} = 0$, where $g(q) = (-y, x, 1)$.

Equation (2.8) gives $\dot{q}$:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \lambda \begin{pmatrix} -y \\ x \\ 1 \end{pmatrix}. \tag{2.41}$$

Using equation (2.9), $\dot{p}$ can be written

$$\begin{pmatrix} \dot{p_x} \\ \dot{p_y} \\ \dot{p_z} \end{pmatrix} = -\nabla V(q) - \lambda \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \left[ \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \lambda \begin{pmatrix} -y \\ x \\ 1 \end{pmatrix} \right] \tag{2.42}$$

and the constraint $g \cdot (p + \lambda g) = 0$, gives

$$\lambda = -\frac{g \cdot p}{g \cdot g}.$$

A simple trajectory starting with the same initial conditions as in [6, pg. 15] is shown in Fig. 2.4. Their initial conditions are $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \lambda) = (0, 0, 0, 0.1, 0.3, 0, 1)$, which when converted to generalized momenta variables are $(x, y, z, p_x, p_y, p_z, \lambda) = (0, 0, 0, 0.1, 0.3, 1, 1)$. The results are qualitatively consistent with [6, pg. 14].

This chapter has shown the equivalence between various variational problems with nonholonomic constraints, and generalized Hamiltonian systems with index 1 constraints. Symplectic Runge–Kutta methods have been shown to be effective methods when applied to these systems. For a full discussion and conclusions see Chapter 6. Attention will now be turned to the other major part of this research: the multisymplectic diamond scheme.

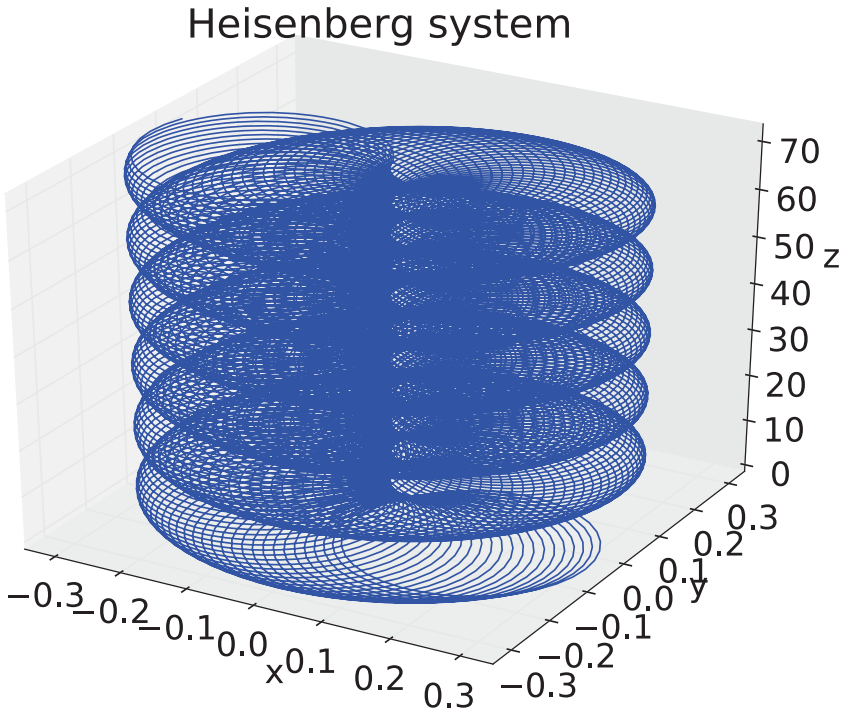



Figure 2.4: The Heisenberg example starting at $(x, y, z, \dot{x}, \dot{y}, \dot{z}) = (0, 0, 0, 0.1, 0.3, 0)$ or $(x, y, z, p_x, p_y, p_z, \lambda) = (0, 0, 0, 0.1, 0.3, 1, 1)$. Qualitatively the results look like [6, pg.14].

# Chapter 3

# The diamond scheme

Recall the multi-Hamiltonian PDE given in (1.4),

$$Kz_t + Lz_x = \nabla S(z), \tag{3.1}$$

where $K$ and $L$ are constant $n \times n$ real skew-symmetric matrices, $z\colon \Omega \to \mathbb{R}^n$, $\Omega \subset \mathbb{R}^2$, and $S\colon \mathbb{R}^n \to \mathbb{R}$. Note that in (1.4) the domain $\Omega$ was implicitly defined to be $\mathbb{R}^2$. Despite much research, no well-defined high-order multisymplectic integrators are known for this class. This chapter introduces a class of general purpose multisymplectic integrators for (3.1) that are locally well-defined and can have high order in space and time.

The (Preissman or Keller) box scheme, a multisymplectic integrator introduced at the end of Section 1.2.4, is simply the implicit midpoint rule (a Runge–Kutta method) applied in space and in time on a rectangular grid. It is called the *simple box scheme* to distinguish it from other Runge–Kutta box-based schemes. There are plenty of multisymplectic low-order methods applicable to Schrödinger equation [14, 27, 29, 40, 72, 41]. Most are based on box-like schemes and are second-order. LingHua's [35] method for the Klein–Gordon–Schrödinger equation has spectral accuracy in space and is second-order in time. Jia–Xiang [32] present a multisymplectic, low-order, implicit/explicit method for the Klein–Gordon–Schrödinger equation. Hong [26] presents a box-like multisymplectic method for the nonlinear Dirac equation. For the Korteweg–de Vries equation there are numerous [26, 79, 80, 4] multisymplectic low order box-like schemes. Moore [54] give a multisymplectic box-like low-order scheme that can be applied to any multi-Hamiltonian system. Bridges and Reich [8] present a staggered-grid multisymplectic method that is based on the symplectic Störmer–Verlet scheme. They discuss possible extensions to higher-order methods.

Instead of discretizing one particular PDE, in this chapter, methods will be developed that are applicable to the entire class (3.1), specializing to a particular equation or

family as late as possible. The simple box scheme is simple to define, can in principle be applied to any PDE of the form (3.1), and has several appealing properties, including the unconditional preservation of dispersion relations (up to a diffeomorphic remapping of continuous to discrete frequencies) with consequent lack of parasitic waves [5] and preservation of the sign of group velocities [19], and lack of spurious reflections at points where the mesh size changes [20]. These properties are related to the linearity of the box scheme; a feature that will be retained in the diamond scheme.[1]

However, the simple box scheme also has some less positive features. It is fully implicit, which makes it expensive; for equations where the CFL condition is not too restrictive, the extra (linear and sometimes nonlinear) stability this provides is not needed. The implicit equations may not have a solution: with periodic boundary conditions, solvability requires that the number of grid points be odd [62]; there is no general treatment of Dirichlet, Neumann, or mixed boundary conditions in the literature that leads to a well-posed method. It is only second-order in space and time.

The latter issue can be avoided by applying higher-order Runge–Kutta methods in space and in time [60]. As the dependent variables are the internal stages, typically one obtains the stage order in space, for example, order $r$ for $r$-stage Gauss Runge–Kutta [50]. However, the scheme is still fully implicit and this time leads to singular ODEs for periodic boundary conditions unless $r$ and $n$ are *both* odd [62, 50].

The first two issues, implicitness and boundary treatment, are related. They can be avoided for some PDEs, like the nonlinear wave equation, by applying suitable partitioned Runge–Kutta methods [62, 44, 63, 64]. *When they apply*, they can be excellent methods, leading to explicit ODEs amenable to explicit time-stepping, arbitrary order, and local boundary handling [50]. For example, the 5-point stencil for $u_{tt} - u_{xx} = f(u)$, equivalent to leapfrog in space and time, is a method of this class (2-stage Lobatto IIIA–IIIB). This class coincides in some cases with staggered grid methods, which again rely on the detailed structure of the equation, and can be extremely effective. However, the partitioning/staggering means that they are not linear methods, which can lead to, for example, discontinuous dispersion relations [50].

For Hamiltonian ODEs there are methods that are symplectic and methods that are not; symplectic methods that are implicit and explicit; methods that are linearly equivariant and methods that are not; and methods defined for all ODEs and methods defined only for particularly structured ODEs. All approaches have their domains of applicability. While there are truly fundamental differences between Hamiltonian PDEs and ODEs, this situation indicates that there may be multisymplectic integrators

[1] A method is *linear in $z$* if it is equivariant with respect to linear changes of the dependent variables, $z \mapsto \tilde{z} := Az$. Runge–Kutta methods and linear multistep methods are linear in this sense, while partitioned Runge–Kutta methods are not. For this concept to make sense, the method has to be defined for all systems obtained by such changes of variables; it does not apply to methods that are defined for a *single* differential equation.

based on Runge–Kutta discretizations that respect the structure of the PDE better than those previously known, and lead to broader applicability. The class of *diamond schemes* introduced for (3.1) will demonstrate this is the case. It is based on the following observation. Let the PDE (3.1) be discretized on a square cell by a Runge–Kutta method in space and time. At each internal point there are $n$ equations and $n$ unknowns. At each pair of opposite edge points there is one equation. Therefore, to get a closed system with the same number of equations as unknowns, data should be specified on exactly *half* of the edge points. It will be shown later that for the nonlinear wave equation, specifying $z$ at the edge points on two adjacent edges leads to a properly determined system for the two opposite edges. What remains is to arrange the cells so that the information flow is consistent with the initial value problem.

**Definition 3.0.1.** *A* diamond scheme *for the PDE* (3.1) *is a quadrilateral mesh in space-time together with a mapping of each quadrilateral to a square upon which a set of Runge–Kutta methods are applied in each dimension. Initial data is specified at sufficient edge points such that the solution can be propagated forward in time by locally solving for pairs of adjacent edges.*

For equations that are symmetric in $x$, the quadrilaterals are typically diamonds, this scheme is first outlined in Section 3.1 in the simplest case, the analogue of the simple box scheme that is dubbed the *simple diamond scheme*. The validity of the method is illustrated on a nonlinear wave equation. Then the general diamond scheme, based on $r$-stage Gauss Runge–Kutta, is introduced in Section 3.2. It is proved to be multisymplectic and, for the nonlinear wave equation (1.5), it is well-defined, i.e., the algebraic equations have a solution for sufficiently small time step. (The method does lead to a closed system of $m$ equations in $m$ unknowns for any multi-Hamiltonian PDE, but their solvability could depend on the PDE.) The validity of the method is illustrated on a nonlinear wave equation, and high convergence orders are observed; as the dependent variables are the *stage* variables of the underlying Runge–Kutta method, this order is related to the stage order, not the classical order, of the Runge–Kutta method. The dispersion relation, and hence the linear stability, of the simple diamond scheme is determined in Section 3.3 for all multi-Hamiltonian PDEs. As the diamond scheme is implicit in $z$, but explicit in $x$, its obeys a CFL-type stability restriction typical of fully explicit methods.

The diamond scheme is inspired by and has some similarities with the *staircase method* in discrete integrability [76]. In both cases, initial data is posed on a subset of a quadrilateral graph such that the remaining data can be filled in uniquely. In discrete integrability, this fill-in is usually explicit, whereas for the diamond schemes it depends on the PDE and is usually implicit. A second key point is that in the diamond method, there is a stability condition that the fill-in must be such that the numerical domain

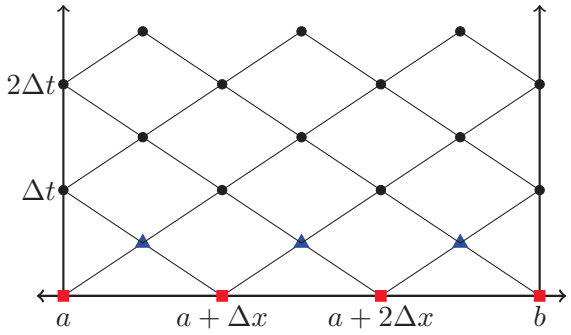



Figure 3.1: The domain divided into diamonds by the simple diamond method. The solution, $z$, is calculated at the corners of the diamonds. The scheme is started using the initial condition, which gives the solution along the $x$ axis at the red squares, and the solution at $t = \frac{\Delta t}{2}$ (the blue triangles) which is calculated using a forward Euler step (or the exact solution if known). After this initialization the simple diamond scheme proceeds, step by step, to update the top of a diamond using the other three known points in that diamond.

of dependence includes the analytic domain of dependence. Thus the characteristics of the PDE determine the geometry of the mesh: if they all pointed to the right, then one could indeed use a simple rectangular mesh and fill in from left to right. (Indeed, this was how early versions of the box scheme proceeded.)

## 3.1 The simple diamond scheme

Consider solving (3.1) numerically on the domain $x \in [a, b]$, $t \geq 0$, with periodic boundary conditions. Unlike a typical finite difference scheme, which uses a rectilinear grid aligned with the $(x, t)$ axes, the simple diamond scheme uses a mesh comprising diamonds, as shown in Figure 3.1.

To describe the simple diamond scheme consider a more detailed view of a single diamond in Figure 3.2: $z_0^1$ is the solution at the top, $z_1^0$ the right-most point, $z_0^{-1}$ the bottom, and $z_{-1}^0$ the left. The point in the centre of the diamond, $z_0^0$, is defined as the average of the corner values.

The discrete version of (3.1) is

$$K\left(\frac{z_0^1 - z_0^{-1}}{\Delta t}\right) + L\left(\frac{z_1^0 - z_{-1}^0}{\Delta x}\right) = \nabla S(z_0^0), \tag{3.2}$$

$$z_0^0 = \frac{1}{4}\left(z_0^1 + z_1^0 + z_0^{-1} + z_{-1}^0\right). \tag{3.3}$$

The values $z_1^0$, $z_0^{-1}$, and $z_{-1}^0$ are known from the preceding step, and $z_0^0$ is determined from (3.3), leaving the $n$ unknowns $z_0^1$ to be determined from the $n$ equations (3.2).

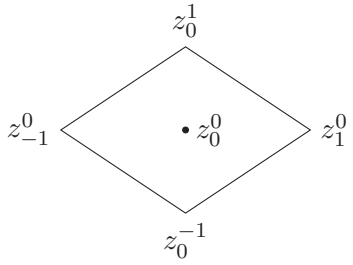



Figure 3.2: A single diamond in the simple diamond scheme. A diamond has a width of $\Delta x$ and height of $\Delta t$.

The simple diamond scheme solves this system of equations independently for each diamond at each time step, then advances to the next step. To determine the local truncation error of this scheme, substitute the exact solution $z(x+i\frac{\Delta x}{2}, t+j\frac{\Delta t}{2})$ for $z_i^j$ into (3.2) and expand in Taylor series:

$$K\left(z_t + \frac{\Delta t^2}{4} z_{ttt} + \mathcal{O}\left(\Delta t^3\right)\right) + L\left(z_x + \frac{\Delta x^2}{4} z_{xxx} + \mathcal{O}\left(\Delta x^3\right)\right) = \nabla S(z) \tag{3.4}$$

$$\Rightarrow K z_t + L z_x = \nabla S(z) + \mathcal{O}\left(\Delta t^2 + \Delta x^2\right); \tag{3.5}$$

thus the order is $\mathcal{O}\left(\Delta t^2 + \Delta x^2\right)$.

For the one-dimensional wave equation defined by (3.1) and (1.5) the simple diamond scheme becomes

$$\frac{u_1^0 - u_{-1}^0}{\Delta x} = \frac{w_0^{-1} + w_1^0 + w_{-1}^0 + w_0^1}{4}, \tag{3.6}$$

$$\frac{u_0^1 - u_0^{-1}}{\Delta t} = \frac{v_0^{-1} + v_1^0 + v_{-1}^0 + v_0^1}{4}, \tag{3.7}$$

$$\frac{v_0^1 - v_0^{-1}}{\Delta t} = \frac{w_1^0 - w_{-1}^0}{\Delta x} + f'(u_0^0). \tag{3.8}$$

At each time step, for each diamond, (3.6) is first solved to give the new $w_0^1$. Then (3.7) is solved for $v_0^1$ and substituted into (3.8) to give a scalar equation of the form

$$u_0^1 = C + \frac{(\Delta t)^2}{4} f'(u_0^0) \tag{3.9}$$

for $u_0^1$, where $C$ depends on the known data. This equation has a solution $u_0^1 = C + \mathcal{O}\left((\Delta t)^2\right)$ for sufficiently small $\Delta t$ when $f'$ is Lipschitz. Thus, although the scheme is implicit, it is only *locally* implicit within each cell; a set of $N$ uncoupled scalar equations is typically much easier to solve than a system of $N$ coupled equations.

**Proposition 3.1.1.** *The simple diamond scheme shown in* (3.2) *satisfies the discrete*

*conservation law*

$$\frac{1}{4\Delta t}\left((dz^0_{-1}+dz^1_0+dz^0_1)\wedge K dz^1_0-(dz^0_{-1}+dz^{-1}_0+dz^0_1)\wedge K dz^{-1}_0\right)$$
$$+\frac{1}{4\Delta x}\left((dz^1_0+dz^0_1+dz^{-1}_0)\wedge L dz^0_1-(dz^1_0+dz^0_{-1}+dz^{-1}_0)\wedge L dz^0_{-1}\right)=0.$$

*Proof.* Take the exterior derivative and apply $dz^0_0\wedge$ on the left of (3.2) to give

$$\tfrac{1}{4\Delta t}(dz^0_{-1}+dz^1_0+dz^0_1+dz^{-1}_0)\wedge K(dz^1_0-dz^{-1}_0)$$
$$+\tfrac{1}{4\Delta x}(dz^1_0+dz^0_1+dz^{-1}_0+dz^0_{-1})\wedge L(dz^0_1-dz^0_{-1})=0.$$

Expanding and simplifying leads to the result. □

Although a recursive implementation of the simple diamond scheme does not run particularly fast, it does clearly define the algorithm:

Let $N$ be the number of diamonds across the domain $[a,b]$, and `z[i,j]` approximate $z(a+i\frac{\Delta x}{2}, j\frac{\Delta t}{2})$, where $i\in\{0,1,\dots,2N-1\}, j\in\{0,1,\dots\}$.
The functions `z0(x)` and $\texttt{z0}_\texttt{t}\texttt{(x)}$ give the initial conditions.
There are periodic boundary conditions.
The function `fun(i, j)` defines `z[i,j]` recursively:

```
fun(i,j)
  if i is -1 then i = 2N-1
  if i is 2N then i = 0
  if j is 0 then
    return z0(a + iΔx/2)
  else if j is 1 then
    return z0(a + iΔx/2) + Δt/2 z0_t(a + iΔx/2)
  else
    Use a numerical solver to find z such that
```

$$K\left(\frac{\texttt{z}-\texttt{fun(i, j}-2\texttt{)}}{\Delta t}\right)+L\left(\frac{\texttt{fun(i}-1\texttt{, j}-1\texttt{)}-\texttt{fun(i}-1\texttt{, j}-1\texttt{)}}{\Delta x}\right)$$
$$=\nabla S\left(\tfrac{\texttt{z+fun(i,j}-2\texttt{)+fun(i}-1\texttt{,j}-1\texttt{)+fun(i}-1\texttt{,j}-1\texttt{)}}{4}\right)$$

```
    return z
```

The above algorithm can be considerably sped up by caching the results of the calculation of `z`.

**Numerical test.** To illustrate the validity of the method, the Sine–Gordon equation,

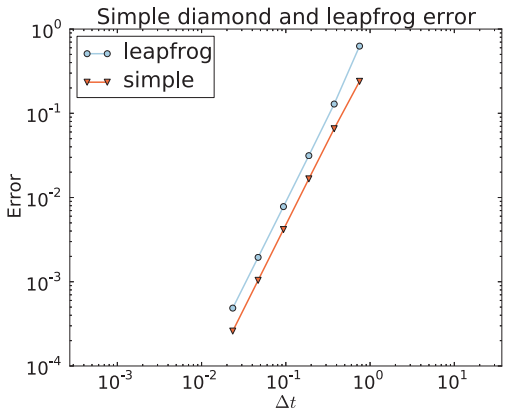



Figure 3.3: The error of the simple diamond scheme applied to the multi-symplectic Hamiltonian PDE arising from the Sine–Gordon equation, and the error of the leapfrog scheme applied to the Sine–Gordon equation. The exact solution is the so-called *breather* on the domain $[-30, 30]$. The Courant number is fixed at $\frac{1}{2}$ as $\Delta t$, is decreased. Both methods appear to have order 2.

$u_{tt} - u_{xx} = -\sin(u)$ will be solved using the leapfrog and the simple diamond schemes. The leapfrog scheme used is

$$\frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{(\Delta t)^2} - \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{(\Delta x)^2} = -\sin(u_i^j),$$

where $\Delta t$ and $\Delta x$ are the same as in the diamond grid, so the five-point stencil contains four diamonds. An exact solution is the so-called *breather*,

$$u(x,t) = 4\arctan\left(\frac{\sin\left(\frac{t}{\sqrt{2}}\right)}{\cosh\left(\frac{x}{\sqrt{2}}\right)}\right). \tag{3.10}$$

The domain is taken significantly large, $[-30, 30]$, so the solution can be assumed periodic. The initial conditions are calculated using the exact solution. The error is the discrete 2-norm of $u$,

$$E^2 = \frac{b-a}{N}\sum_{i=1}^{N}\left(\tilde{u}_i - u(a + i\Delta x, T)\right)^2. \tag{3.11}$$

Figure 3.3 shows the error of both schemes as $\Delta t$ is reduced while keeping the Courant number $\frac{\Delta t}{\Delta x} = \frac{1}{2}$. The integration time $T = 1.5$ is twice the largest time step. It is apparent that for this problem, both schemes appear to have order 2.

**Boundary conditions.** Because the simple diamond scheme is locally defined, boundary conditions can be treated locally without affecting the solvability of the discrete

equations. The numerical method requires $n$ conditions at each boundary, and the PDE typically provides fewer than $n$. Extra conditions can be found by (i) differentiating the boundary conditions in the direction of the boundary; (ii) using and/or differentiating the PDE at the boundary; and (iii) utilizing a phantom diamond that straddles the boundary. Here is an illustration of just one possible treatment of Dirichlet and Neumann boundary conditions. For the nonlinear wave equation with Dirichlet condition $u(a,t) = f(t)$, the values of $u(a, n\Delta t) = f(n\Delta t)$ are given, and by differentiating the boundary condition, $v(a, n\Delta t) = f'(n\Delta t)$ can be found. The PDE (1.5), and using the approximation

$$A = v_x\left(a, \left(n - \tfrac{1}{2}\right)\Delta t\right) \approx \frac{v(a + \frac{1}{2}\Delta x, (n - \frac{1}{2})\Delta t) - f'((n - \frac{1}{2})\Delta t)}{\frac{1}{2}\Delta x},$$

gives the update $w(a, n\Delta t) = w(a, (n-1)\Delta t) + \Delta t A$. For the Neumann condition $u_x(b,t) = g(t)$, a phantom diamond is used. First fill in the values of $\mathbf{z}_1^0$ (see Figure 3.2): the boundary condition provides $w_0^0$, and linear extrapolation provides $w_1^0$; the boundary condition provides $u_x$ at the centre point, and linear extrapolation from $u_{-1}^0$ provides $u_1^0$; the PDE implies $v_x = w_t$, where $w_t$ is provided by the boundary condition at the centre point, and linear extrapolation from $v_{-1}^0$ provides $v_1^0$. The boundary diamond is then filled in like the internal diamonds. Figure 3.4 shows the results of 150,000 steps of the simple diamond scheme together with this boundary treatment for the linear wave equation for a solution that contains both left- and right-moving waves.

## 3.2 The diamond scheme

The diamond scheme refines the simple diamond scheme discretization by using the multisymplectic Runge–Kutta collocation method given by Reich [60] within each diamond. It is easier to apply this method to a square that is aligned with the axes, so the first step is to transform the $(x,t)$ coordinate space. Each diamond in Figure 3.1 is transformed to a square of side length one using the linear transformation $\Phi$ defined by

$$\Phi\colon \quad \tilde{x} = \tfrac{1}{\Delta x}x + \tfrac{1}{\Delta t}t \quad \text{and} \quad \tilde{t} = -\tfrac{1}{\Delta x}x + \tfrac{1}{\Delta t}t. \tag{3.12}$$

Because (3.1) has no dependence on $x$ or $t$ it doesn't matter where the square is located in $(\tilde{x}, \tilde{t})$ space, so the same transformation can be used for all the diamonds. Let $\tilde{z}(\tilde{x}, \tilde{t}) = z(x,t)$. By the chain rule

$$z_x = \tilde{z}_{\tilde{x}}\tfrac{1}{\Delta x} - \tilde{z}_{\tilde{t}}\tfrac{1}{\Delta x} \quad \text{and} \quad z_t = \tilde{z}_{\tilde{x}}\tfrac{1}{\Delta t} + \tilde{z}_{\tilde{t}}\tfrac{1}{\Delta t}, \tag{3.13}$$

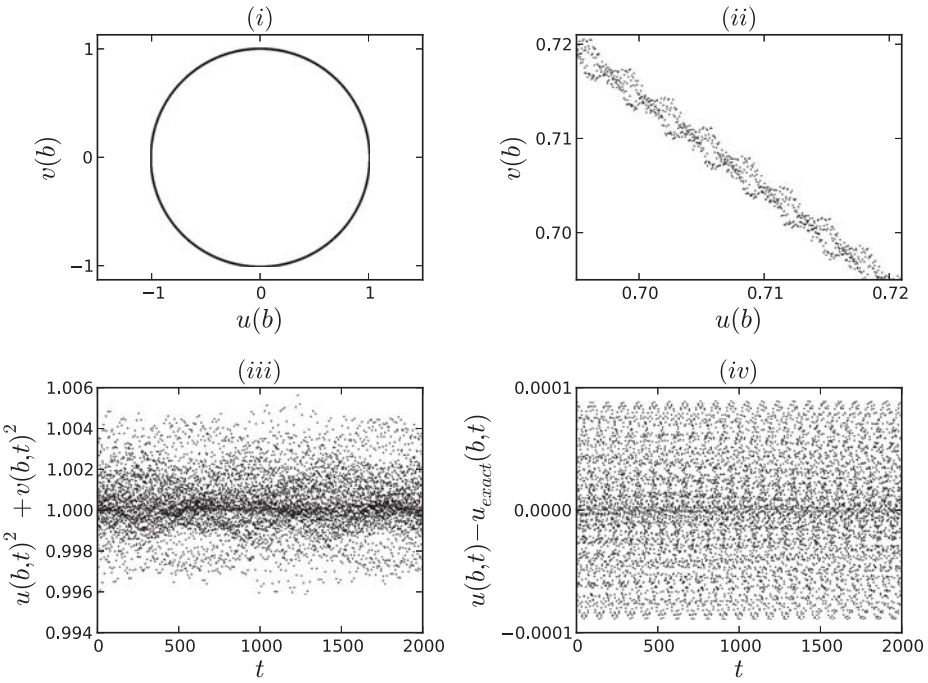



Figure 3.4: Solution of the linear wave equation on $[a, b]$ with Dirichlet boundary condition $u(a, t) = \sin a \cos t$, Neumann boundary condition $u_x(b, t) = -\cos a \cos t$, $a = 0.5$, $b = \pi/2$, and exact solution $u(x, t) = \sin x \cos t$ which contains both left- and right-moving waves. There are $N = 40$ diamonds in space and the Courant number is 0.5; 150,000 time steps are shown. See text for the handling of the boundary conditions. (i): phase portrait of $(u(b, t), v(b, t))$, which is a circle in the exact solution. (ii): close-up of (i) showing some blurring of the orbit, which does not increase in time. (iii): Evolution of local energy $u(b, t)^2 + v(b, t)^2$ at $x = b$ vs. time (exact value is 1); the error does not increase in time. (iv): Global error in $u(b, t)$ vs. time. The error does not increase in time. (Note that $u(b, t)$ is not fixed by the boundary condition.)

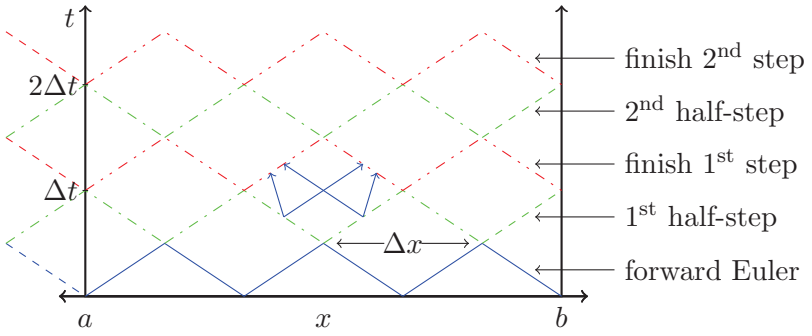



Figure 3.5: Information flows upwards as indicated by the solid blue arrows for a typical diamond. The solution, $z$, is initialized on the solid blue zig-zag line. A step of the diamond scheme consists of two half-steps. The first half-step calculates $z$ along the green dash-dot line, which by periodicity is extended to the dashed line to the right. The second half-step uses the green dash-dot line to calculate the red dash-double-dotted line, which again by periodicity is extended to the left-hand dashed segment.

so

$$\begin{aligned} Kz_t + Lz_x &= K\left(\tilde{z}_{\tilde{x}}\tfrac{1}{\Delta t} + \tilde{z}_{\tilde{t}}\tfrac{1}{\Delta t}\right) + L\left(\tilde{z}_{\tilde{x}}\tfrac{1}{\Delta x} - \tilde{z}_{\tilde{t}}\tfrac{1}{\Delta x}\right), \\ &= \left(\tfrac{1}{\Delta t}K - \tfrac{1}{\Delta x}L\right)\tilde{z}_{\tilde{t}} + \left(\tfrac{1}{\Delta t}K + \tfrac{1}{\Delta x}L\right)\tilde{z}_{\tilde{x}}, \end{aligned}$$

thus (3.1) transforms to

$$\tilde{K}\tilde{z}_{\tilde{t}} + \tilde{L}\tilde{z}_{\tilde{x}} = \nabla S(\tilde{z}), \tag{3.14}$$

where

$$\tilde{K} = \tfrac{1}{\Delta t}K - \tfrac{1}{\Delta x}L \quad \text{and} \quad \tilde{L} = \tfrac{1}{\Delta t}K + \tfrac{1}{\Delta x}L. \tag{3.15}$$

**Outline of method.** Figure 3.5 illustrates the diamond scheme for a sample initial-boundary value problem on $[a, b] \times \mathbb{R}^+$ with periodic boundary conditions. The solution $z$ is calculated at grid points located on the solid diamond edges; dashed edges indicate where values are inferred by periodicity. Information follows from the bottom left and right edges of a diamond to the top left and right edges of the same diamond. An initialization step must provide values for $z$ along the bottom edges of the first row of diamonds (the first solid blue zig-zag line in the figure). This is extended using periodicity beyond the left-hand boundary (the blue dashed line). The initialization should not reduce the overall order, nor undo special numerical properties of the scheme. For much of the research, the known exact solution was used; however, an initialization method of high-order, presented in Section 5.1, was used for the numerical results presented in this thesis. A step of the diamond scheme consists of two half-steps. The first half-step calculates $z$ along the top edges of the first row of diamonds (the green dash-dot zig-zag), which by periodicity is extended to the right-hand boundary (the

green dashed line). The second half-step uses values on the top edges of the first row of diamonds (the green dash-dot line) to calculate the new values of $z$ on the top edges of the second row (the red dash-double-dotted line). Again by periodicity the values from the top right edge of the most right diamond are copied outside the left-hand boundary of the domain (the red dashed segment). Another step can now be performed using the red dash-double-dotted zig-zag as initial data (the very-right-hand line segment is not used except to provide values for the dashed line).

**Updating one diamond.** Let $(A, b, c)$ be the parameters of an $r$-stage Runge–Kutta method, in what follows this will be assumed to be a Gauss Runge–Kutta method. Figure 3.6 shows a diamond with $r = 3$ and its transformation to the unit square. The square contains $r \times r$ internal grid points, as determined by the Runge–Kutta coefficients $c$, and internal stages $Z_i^j$, which are analogous to the usual internal grid points and stages in a Runge–Kutta method. The internal stages also carry the variables $X_i^j$ and $T_i^j$ which approximate $z_x$ and $z_t$, respectively, at the internal stages.

The dependent variables of the method are the values of $z$ at the edge grid points. To be able to distinguish the internal edge points from all the edge points let $I$ be the set of indices $\{1, \ldots, r\}$. Then, for example, $\tilde{z}_I^{\mathrm{b}}$ refers to $\tilde{z}_i^{\mathrm{b}}, i = 1, \ldots, r$. If the $I$ qualifier does not appear, then the left- or bottommost corner is included, for example, $\tilde{z}^{\mathrm{b}}$ refers to the points $\tilde{z}_i^{\mathrm{b}}, i = 0, \ldots, r$, but does not include $\tilde{z}_{r+1}^{\mathrm{b}}$ which is $\tilde{z}_{\mathfrak{r}}^0$. Note $\tilde{z}_0^{\mathrm{b}} = \tilde{z}_\ell^0$ (which is also the same as $z_0^{-1}$). The Runge–Kutta discretization is

$$Z_i^j = \tilde{z}_\ell^j + \sum_{k=1}^{r} a_{ik} X_k^j, \tag{3.16}$$

$$Z_i^j = \tilde{z}_i^{\mathrm{b}} + \sum_{k=1}^{r} a_{jk} T_i^k, \tag{3.17}$$

$$\nabla S(Z_i^j) = \tilde{K} T_i^j + \tilde{L} X_i^j, \tag{3.18}$$

together with the update equations

$$\tilde{z}_{\mathfrak{r}}^i = \tilde{z}_\ell^i + \sum_{k=1}^{r} b_k X_k^i, \tag{3.19}$$

$$\tilde{z}_i^{\mathrm{t}} = \tilde{z}_i^{\mathrm{b}} + \sum_{k=1}^{r} b_k T_i^k, \tag{3.20}$$

for $i, j \in I$. The $\tilde{z}_\ell^I$ and $\tilde{z}_I^{\mathrm{b}}$ are known. Equations (3.16)–(3.18) are first solved for the internal stage values $Z_i^j$, $X_i^j$, and $T_i^j$, then equations (3.19) and (3.20) are used to calculate $\tilde{z}_I^{\mathrm{t}}$ and $\tilde{z}_{\mathfrak{r}}^I$. Equations (3.16)–(3.18) are $3r^2$ equations in $3r^2$ unknowns $Z$, $X$, and $T$. Equations (3.16) and (3.17) are linear in $X$ and $T$. Thus in practice the method requires solving a set of $nr^2$ nonlinear equations for $Z$ in each diamond.

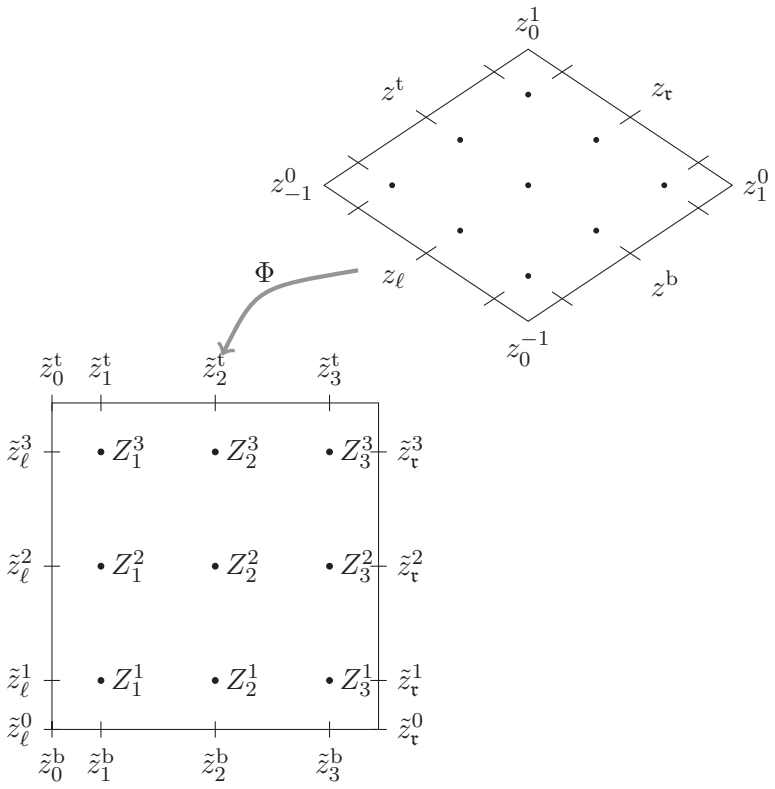



Figure 3.6: The diamond transformed by a linear transformation, $\Phi$, to the unit square. The square contains $r \times r$ ($r = 3$ in this example) internal stages, $Z_i^j$. The solution is known along the bottom and left-hand sides. The method proceeds as two sets of $r$ Gauss Runge–Kutta $r$-step methods: internal stage values, $Z_i^j$, $X_i^j$, $T_i^j$, are calculated, then the right and top updated.

The method does not need the values at the corners. However, if solutions are wanted at the corners (particularly desirable when comparing computed solutions) then the method can be extended by: allowing $j = 0$ in (3.16) and associating $Z_i^0$ with $\tilde{z}_i^{\mathrm{b}}$; and allowing $i = 0$ in (3.17) and associating $Z_0^j$ with $\tilde{z}_\ell^j$. This extension gives equations for $X_k^0$ and $T_0^k$, which can be used in the update equations (3.19) and (3.20) which are extended by allowing $i = 0$. On this extended domain (3.16)–(3.18) are $2r(r+1)+r^2$ equations. This is because: (3.16) is extended onto the bottom boundary ($r(r+1)$ equations), (3.17) onto the left boundary ($r(r+1)$ equations), but there is no need to extend (3.18) onto either boundaries because $T$ was not extended onto the bottom boundary, and $X$ was not extended onto the left boundary. The number of unknowns is $2r(r+1)+r^2$, so again there is the same number of equations as unknowns. Corner points are shared by two adjacent diamonds, and typically $\tilde{z}_{r+1}^{\mathrm{t}} \neq \tilde{z}_{\mathtt{r}}^{r+1}$. In practice, the mean of these two approximations is used.

Here is a summary of the diamond scheme algorithm that includes computing the corner values:

Let $\mathtt{z}$ and $\mathtt{z_n}$ be $N(2r+1)$ length vectors with each element in $\mathbb{R}^n$. These vectors contain the $\tilde{z}$ values for two particular edges of each diamond. Each of the two edges has $r$ nodes, plus there is the value at the bottom, hence $2r+1$ values per $N$ diamonds.

1. Initialize $\mathtt{z}$. It now contains $\tilde{z}$ for the blue zig-zag at the bottom in Figure 3.5.
2. The half-step. For each diamond;
   (a) associate $\tilde{z}_\ell$ and $\tilde{z}^{\mathrm{b}}$ with the correct values in $\mathtt{z}$ (periodicity is used at the edges);
   (b) solve (3.16)–(3.18);
   (c) use (3.19) and (3.20) to find $\tilde{z}_{\mathtt{r}}$ and $\tilde{z}^{\mathrm{t}}$;
   (d) associate $\tilde{z}_{\mathtt{r}}$ and $\tilde{z}^{\mathrm{t}}$ with the current diamond's section of $\mathtt{z_n}$ (periodicity is used at the edges).
3. $\mathtt{z} = \mathtt{z_n}$. Now $\mathtt{z}$ contains $\tilde{z}$ values for the second/green zig-zag in Figure 3.5.
4. Perform step 2 again.
5. $\mathtt{z} = \mathtt{z_n}$. Now $\mathtt{z}$ contains $\tilde{z}$ values for the third/red zig-zag in Figure 3.5.
6. If final time not reached go to step 2.

Extending the diamond scheme onto the corners is mainly for computational convenience, and as the remainder of this chapter is predominantly theoretical in nature, the corners will not be included in the scheme unless explicitly stated.

**Theorem 3.2.1.** *For the multi-Hamiltonian one-dimensional wave equation defined by* (3.1) *and* (1.5) *with the conditions:*

- $f'$ *is Lipschitz with constant* $L$*;*
- *the matrix* $A$ *of coefficients of the underlying Runge–Kutta scheme is invertible;*
- *the matrix*

$$B = (1-\lambda^2)(I \otimes A^{-2}) + 2(1+\lambda^2)(A^{-1} \otimes A^{-1}) + (1-\lambda^2)(A^{-2} \otimes I),$$

  *is invertible, where* $\lambda = \frac{\Delta t}{\Delta x}$ *is the Courant number; and*

- $\Delta t < 1/(L\|B^{-1}\|_\infty)^{1/2}$,

*Equations* (3.16)–(3.18) *are solvable, and thus the diamond scheme is well-defined.*

*Proof.* Equations (3.16) and (3.17) relate components of matrices and tensors. Writing these equations in tensor form, then multiplying on the left by $A^{-1}$, gives expressions for $X_i^j$ and $T_i^j$. Substituting these into (3.18) gives

$$\nabla S(Z_i^j) = \tilde{K} \sum_{k=1}^{r} m_{jk}(Z_i^k - z_i^{\mathrm{b}}) + \tilde{L} \sum_{k=1}^{r} m_{ik}(Z_k^j - z_\ell^k),$$

where $m_{ij}$ are the elements of $A^{-1}$, and the tildes on the $z$ values have been dropped for clarity. Using (3.15) for $\tilde{K}$ and $\tilde{L}$, and adopting the summation convention, this becomes

$$\begin{pmatrix} -f'(u_i^j) \\ v_i^j \\ -w_i^j \end{pmatrix} = \begin{pmatrix} 0 & \frac{-1}{\Delta t} & \frac{-1}{\Delta x} \\ \frac{1}{\Delta t} & 0 & 0 \\ \frac{1}{\Delta x} & 0 & 0 \end{pmatrix} m_{jk}(Z_i^k - z_i^{\mathrm{b}}) + \begin{pmatrix} 0 & \frac{-1}{\Delta t} & \frac{1}{\Delta x} \\ \frac{1}{\Delta t} & 0 & 0 \\ \frac{-1}{\Delta x} & 0 & 0 \end{pmatrix} m_{ik}(Z_k^j - z_\ell^k).$$

The solution for $v_i^j$ and $w_i^j$,

$$\begin{aligned} v_i^j &= \frac{1}{\Delta t} m_{jk}(u_i^k - u_i^{\mathrm{b}}) + \frac{1}{\Delta t} m_{ik}(u_k^j - u_\ell^k), \\ w_i^j &= \frac{-1}{\Delta x} m_{jk}(u_i^k - u_i^{\mathrm{b}}) + \frac{1}{\Delta x} m_{ik}(u_k^j - u_\ell^k), \end{aligned}$$

is substituted into the equation for $u_i^j$, which after simplification gives

$$(1-\lambda^2)m_{jk}m_{kp}u_i^p + 2(1+\lambda^2)m_{jk}m_{ip}u_p^k + (1-\lambda^2)m_{ik}m_{kp}u_p^j = b + (\Delta t)^2 f'(u_i^j),$$

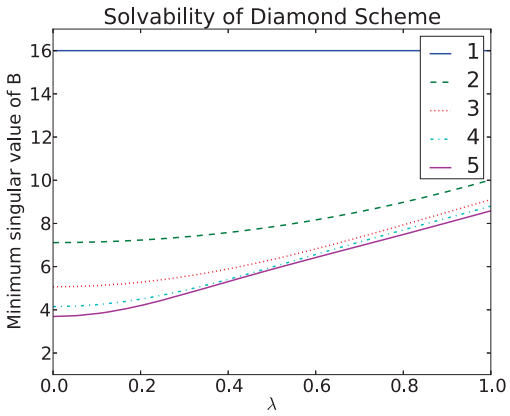



Figure 3.7: How the minimum singular value of $B$ varies with different Courant numbers. Because there are no zero singular values, the diamond scheme is solvable for the wave equation for all $\lambda \in [0,1]$ and $r$ up to 5. It is easy to check this holds for larger $r$.

where the vector $b$ is a constant term depending on $A^{-1}$ and the initial data $z_\ell$ and $z^{\mathrm{b}}$. Let $u = (u_1^1, u_1^2, \ldots, u_1^r, u_2^1, \ldots, u_3^1, \ldots, u_r^r)$ and $f'(u) = (f'(u_1^1), \ldots, f'(u_r^r))$; then this simplifies to

$$Bu = b + \Delta t^2 f'(u), \tag{3.21}$$

where $B$ is given in the conditions of the theorem. To complete the proof it must be shown that this equation has a solution. Because $B$ is invertible, $G(u) = B^{-1}(b + \Delta t^2 f'(u))$ exists. Consider $G$ applied to the two points $u^1$ and $u^2$,

$$\begin{aligned}\|G(u^1) - G(u^2)\|_\infty &= \Delta t^2 \|B^{-1}(f'(u^1) - f'(u^2))\|_\infty \\ &\leq \Delta t^2 \|B^{-1}\|_\infty L \|u^1 - u^2\|_\infty.\end{aligned}$$

By the contraction mapping theorem and the condition on $\Delta t$, $G$ must have a fixed point $u = G(u)$, thus (3.21) has a solution. □

For a particular Runge–Kutta method it is straightforward to calculate the matrix $B$ and determine the conditions on $\lambda$ that lead to solvability. Figure 3.7 shows that for Gauss Runge–Kutta, $r = 1, \ldots, 5$ and $\lambda \in [0,1]$, the minimum singular value of $B$ is nonzero. This calculation can be performed for larger $r$.

**Theorem 3.2.2.** *The diamond scheme satisfies the discrete symplectic conservation law*

$$\frac{1}{\Delta t}\sum_{i=1}^{r} b_i(\omega_i^{\mathrm{t}} + \omega_{\mathfrak{r}}^i - (\omega_\ell^i + \omega_i^{\mathrm{b}})) + \frac{1}{\Delta x}\sum_{i=1}^{r} b_i(\kappa_{\mathfrak{r}}^i + \kappa_i^{\mathrm{b}} - (\kappa_i^{\mathrm{t}} + \kappa_\ell^i)) = 0,$$

*where* $\omega_n^m = \frac{1}{2} dz_n^m \wedge K dz_n^m$, $\kappa_n^m = \frac{1}{2} dz_n^m \wedge L dz_n^m$, *and* $m, n \in [0,r] \cup \{\mathrm{t}, \mathfrak{r}, \mathrm{b}, \ell\}$ *(refer to Figure 3.6 for the definition of those labels).*

*Proof.* The solver within each square satisfies the discrete multisymplectic conservation law [62]

$$\Delta x \sum_{i=1}^{r} b_i(\tilde{\omega}_i^{\mathrm{t}} - \tilde{\omega}_i^{\mathrm{b}}) + \Delta t \sum_{i=1}^{r} b_i(\tilde{\kappa}_{\mathfrak{r}}^i - \tilde{\kappa}_\ell^i) = 0,$$

where $\Delta x = \Delta t = 1$ because the square has side length one. Substituting in $\tilde{\omega}_n^m = \frac{1}{2} d\tilde{z}_n^m \wedge \tilde{K} d\tilde{z}_n^m$ and $\tilde{\kappa}_n^m = \frac{1}{2} d\tilde{z}_n^m \wedge \tilde{L} d\tilde{z}_n^m$ gives

$$\frac{1}{2} \sum_{i=1}^{r} b_i(d\tilde{z}_i^{\mathrm{t}} \wedge \tilde{K} d\tilde{z}_i^{\mathrm{t}} - d\tilde{z}_i^{\mathrm{b}} \wedge \tilde{K} d\tilde{z}_i^{\mathrm{b}} + d\tilde{z}_{\mathfrak{r}}^i \wedge \tilde{L} d\tilde{z}_{\mathfrak{r}}^i - d\tilde{z}_\ell^i \wedge \tilde{L} d\tilde{z}_\ell^i) = 0.$$

Using $d\tilde{z}_n^m = dz_n^m$, and (3.15) this becomes

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^{r} b_i(dz_i^{\mathrm{t}} \wedge (\tfrac{1}{\Delta t} K - \tfrac{1}{\Delta x} L) dz_i^{\mathrm{t}} - dz_i^{\mathrm{b}} \wedge (\tfrac{1}{\Delta t} K - \tfrac{1}{\Delta x} L) dz_i^{\mathrm{b}} \\ + dz_{\mathfrak{r}}^i \wedge (\tfrac{1}{\Delta t} K + \tfrac{1}{\Delta x} L) dz_{\mathfrak{r}}^i - dz_\ell^i \wedge (\tfrac{1}{\Delta t} K + \tfrac{1}{\Delta x} L) dz_\ell^i) = 0 \\ \Rightarrow \frac{1}{\Delta t} \sum_{i=1}^{r} b_i(\omega_i^{\mathrm{t}} - \omega_i^{\mathrm{b}} + \omega_{\mathfrak{r}}^i - \omega_\ell^i) + \frac{1}{\Delta x} \sum_{i=1}^{r} b_i(-\kappa_i^{\mathrm{t}} + \kappa_i^{\mathrm{b}} + \kappa_{\mathfrak{r}}^i - \kappa_\ell^i) = 0 \end{aligned}$$

□

Now the relationship between the simple diamond scheme (which uses corner values only) and the $r = 1$ diamond scheme (which uses edge values only) will be examined. To relate the two, note that the extension to the corners of the $r = 1$ diamond scheme discussed previously, in which (3.16), (3.17), (3.19) and (3.20) are applied with $i = j = 0$, leads on a single diamond to

$$\begin{aligned} \tilde{z}^{\mathrm{b}} &= \frac{z_0^{-1} + z_1^0}{2}, \quad & \tilde{z}^{\mathrm{t}} &= \frac{z_{-1}^0 + z_0^1}{2}, \\ \tilde{z}_\ell &= \frac{z_0^{-1} + z_{-1}^0}{2}, \quad & \tilde{z}_{\mathfrak{r}} &= \frac{z_1^0 + z_0^1}{2}, \end{aligned} \tag{3.22}$$

where the sub/superscript 1 has been dropped.

**Theorem 3.2.3.** *(i) Any solution of the simple diamond scheme, mapped to edge midpoint values according to Eqs.* (3.22)*, satisfies the equations of the $r = 1$ diamond scheme. (ii) Any solution of the $r = 1$ diamond scheme corresponds under* (3.22) *locally to a 1-parameter family of solutions to the simple diamond scheme. With periodic boundary conditions, the correspondence is global iff the solution satisfies $\sum_i z_{\ell i,j}^1 = \sum_i z_{1i,j}^{\mathrm{b}}$ for all $j$, where the subscript $_{i,j}$ refers to the $i$th diamond at the $j$th time step.*

*Proof.* When $r = 1$, equations (3.16)–(3.20) become

$$Z = \tilde{z}_\ell + \frac{1}{2}X, \tag{3.23}$$

$$Z = \tilde{z}^{\mathrm{b}} + \frac{1}{2}T, \tag{3.24}$$

$$\nabla S(Z_1^1) = \tilde{K}T + \tilde{L}X, \tag{3.25}$$

$$\tilde{z}_{\mathfrak{r}} = \tilde{z}_\ell + X, \tag{3.26}$$

$$\tilde{z}^{\mathrm{t}} = \tilde{z}^{\mathrm{b}} + T, \tag{3.27}$$

where $\tilde{K}$ and $\tilde{L}$ are the transformed $K$ and $L$ given in (3.15), and the sub/superscript 1 has been omitted in $\tilde{z}_\ell$, $\tilde{z}^{\mathrm{b}}$, $\tilde{z}_{\mathfrak{r}}$, $\tilde{z}^{\mathrm{t}}$, $Z$, $X$, and $T$.

Eliminating $X$, $T$, and $Z$ from the five equations (3.23)–(3.27) yields the equivalent formulation

$$\tilde{K}\left(\tilde{z}^{\mathrm{t}} - \tilde{z}^{\mathrm{b}}\right) + \tilde{L}\left(\tilde{z}_{\mathfrak{r}} - \tilde{z}_\ell\right) = \nabla S\left(\frac{\tilde{z}^{\mathrm{t}} + \tilde{z}^{\mathrm{b}} + \tilde{z}_{\mathfrak{r}} + \tilde{z}_\ell}{4}\right), \tag{3.28}$$

$$\tilde{z}^{\mathrm{t}} - \tilde{z}_{\mathfrak{r}} + \tilde{z}^{\mathrm{b}} - \tilde{z}_\ell = 0. \tag{3.29}$$

(i) Substituting the relations (3.22) into the equations of the simple diamond scheme (3.2) and (3.3) gives

$$\begin{aligned}
\nabla S\left(\frac{z_0^{-1} + z_{-1}^0 + z_1^0 + z_0^1}{4}\right) &= K\left(\frac{z_0^1 - z_0^{-1}}{\Delta t}\right) + L\left(\frac{z_1^0 - z_{-1}^0}{\Delta x}\right) \\
\Rightarrow \nabla S\left(\frac{\tilde{z}^{\mathrm{t}} + \tilde{z}^{\mathrm{b}} + \tilde{z}_{\mathfrak{r}} + \tilde{z}_\ell}{4}\right) &= \tfrac{1}{\Delta t}K\left(\tilde{z}^{\mathrm{t}} - \tilde{z}^{\mathrm{b}} + \tilde{z}_{\mathfrak{r}} - \tilde{z}_\ell\right) + \tfrac{1}{\Delta x}L\left(\tilde{z}_{\mathfrak{r}} - \tilde{z}_\ell - \tilde{z}^{\mathrm{t}} + \tilde{z}^{\mathrm{b}}\right) \\
&= \left(\tfrac{1}{\Delta t}K - \tfrac{1}{\Delta x}L\right)\left(\tilde{z}^{\mathrm{t}} - \tilde{z}^{\mathrm{b}}\right) + \left(\tfrac{1}{\Delta t}K + \tfrac{1}{\Delta x}L\right)\left(\tilde{z}_{\mathfrak{r}} - \tilde{z}_\ell\right) \\
&= \tilde{K}\left(\tilde{z}^{\mathrm{t}} - \tilde{z}^{\mathrm{b}}\right) + \tilde{L}\left(\tilde{z}_{\mathfrak{r}} - \tilde{z}_\ell\right)
\end{aligned}$$

using (3.15). That is, the equations (3.28) of the $r = 1$ diamond scheme are satisfied. Equation (3.29) follows directly from (3.22).

(ii) Using (3.22), the corner values of one diamond can be recovered uniquely from the edge values and one corner value. From these values, adjacent diamonds can be filled in, continuing to get a unique solution for the corner values in any simply connected region. The same calculation as in part (i) now shows that these corner values satisfy the equations of the simple diamond scheme. For a global solution with periodic boundary conditions, the edge values at one time level $j$ must lie in the range of the mean value operator in (3.22), which gives the condition in the theorem. (If the condition holds at $j = 1$, then it holds for all $j$, from (3.29).) In both cases one corner value parameterizes the solutions. □

Theorem 3.2.3 implies that under (3.22), the multisymplectic conservation laws of the simple and $r = 1$ diamond schemes are equivalent. This is now proved directly.

**Corollary 3.2.4.** *Under the relations* (3.22)*, the simple diamond scheme and the* $r = 1$ *diamond scheme have equivalent discrete multisymplectic conservation laws.*

*Proof.* Substitute $r = 1$ into Theorem 3.2.2, note $b_1 = 1$, and differentiate (3.22) to get $dz_1^{\mathrm{t}} = (dz_0^1 + dz_{-1}^0)/2$, $dz_{\mathrm{t}}^1 = (dz_0^1 + dz_1^0)/2$, $dz_1^{\mathrm{b}} = (dz_1^0 + dz_0^{-1})/2$, and $dz_\ell^1 = (dz_0^{-1} + dz_{-1}^0)/2$, leading to

$$\begin{aligned}
&\frac{1}{8\Delta t}\left((dz_0^1 + dz_{-1}^0) \wedge K(dz_0^1 + dz_{-1}^0) + (dz_0^1 + dz_1^0) \wedge K(dz_0^1 + dz_1^0)\right. \\
&\qquad \left. -(dz_0^{-1} + dz_{-1}^0) \wedge K(dz_0^{-1} + dz_{-1}^0) - (dz_1^0 + dz_0^{-1}) \wedge K(dz_1^0 + dz_0^{-1})\right) \\
&+ \frac{1}{8\Delta x}\left((dz_0^1 + dz_1^0) \wedge L(dz_0^1 + dz_1^0) + (dz_1^0 + dz_0^{-1}) \wedge L(dz_1^0 + dz_0^{-1})\right. \\
&\qquad \left. -(dz_0^1 + dz_{-1}^0) \wedge L(dz_0^1 + dz_{-1}^0) - (dz_0^{-1} + dz_{-1}^0) \wedge L(dz_0^{-1} + dz_{-1}^0)\right) = 0,
\end{aligned}$$

which upon expanding and simplifying leads to the simple diamond scheme conservation law given in Proposition 3.1.1. □

Also the order of the $r = 1$ method should be the same as the simple diamond scheme, as a check, this is now proved directly.

**Theorem 3.2.5.** *The* $r = 1$ *diamond scheme is order two in space and time.*

*Proof.* Substituting the exact solution, $z$, into (3.28), and using Taylor series to expand around the centre of the diamond gives

$$\begin{aligned}
&\tilde{K}\left[z - \tfrac{\Delta x}{4}z_x + \tfrac{\Delta t}{4}z_t + \tfrac{\Delta x^2}{32}z_{xx} - \tfrac{\Delta x\Delta t}{16}z_{xt} + \tfrac{\Delta t^2}{32}z_{tt} + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right) -\right. \\
&\qquad \left.\left(z + \tfrac{\Delta x}{4}z_x - \tfrac{\Delta t}{4}z_t + \tfrac{\Delta x^2}{32}z_{xx} - \tfrac{\Delta x\Delta t}{16}z_{xt} + \tfrac{\Delta t^2}{32}z_{tt} + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right)\right)\right] + \\
&\qquad \tilde{L}\left[z + \tfrac{\Delta x}{4}z_x + \tfrac{\Delta t}{4}z_t + \tfrac{\Delta x^2}{32}z_{xx} + \tfrac{\Delta x\Delta t}{16}z_{xt} + \tfrac{\Delta t^2}{32}z_{tt} + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right) -\right. \\
&\qquad\quad \left.\left(z - \tfrac{\Delta x}{4}z_x - \tfrac{\Delta t}{4}z_t + \tfrac{\Delta x^2}{32}z_{xx} + \tfrac{\Delta x\Delta t}{16}z_{xt} + \tfrac{\Delta t^2}{32}z_{tt} + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right)\right)\right] \\
&\qquad\qquad = \nabla S\left(z + \mathcal{O}\left(\Delta t^2 + \Delta x^2\right)\right) \\
&\qquad\qquad \Rightarrow \tilde{K}\left[-\tfrac{\Delta x}{2}z_x + \tfrac{\Delta t}{2}z_t + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right)\right] + \\
&\qquad\qquad\quad \tilde{L}\left[\tfrac{\Delta x}{2}z_x + \tfrac{\Delta t}{2}z_t + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right)\right] \\
&\qquad\qquad\qquad\qquad = \nabla S(z) + \mathcal{O}\left(\Delta t^2 + \Delta x^2\right).
\end{aligned}$$

Using (3.15) to substitute for $\tilde{K}$ and $\tilde{L}$ gives

$$\begin{aligned}\left(\tfrac{1}{\Delta t}K - \tfrac{1}{\Delta x}L\right)\left[-\tfrac{\Delta x}{2}z_x + \tfrac{\Delta t}{2}z_t + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right)\right] + \\ \left(\tfrac{1}{\Delta t}K + \tfrac{1}{\Delta x}L\right)\left[\tfrac{\Delta x}{2}z_x + \tfrac{\Delta t}{2}z_t + \mathcal{O}\left(\Delta t^3 + \Delta x^3\right)\right] = \nabla S(z) + \mathcal{O}\left(\Delta t^2 + \Delta x^2\right) \\ \Rightarrow Kz_t + Lz_x = \nabla S(z) + \mathcal{O}\left(\Delta t^2 + \Delta x^2\right).\end{aligned}$$

□

The $r = 1$ diamond scheme extended to the corners is also order two in space and time. Substituting (3.28) into (3.22) gives

$$\tilde{K}\left(\frac{z_{-1}^0 + z_0^1}{2} - \frac{z_0^{-1} + z_1^0}{2}\right) + \tilde{L}\left(\frac{z_1^0 + z_0^1}{2} - \frac{z_0^{-1} + z_{-1}^0}{2}\right) = \nabla S\left(\frac{z_0^{-1} + z_1^0 + z_{-1}^0 + z_0^1}{4}\right).$$

Substituting in the exact solution and using Taylor expansion gives the result.

## 3.3 Dispersion analysis

**Lemma 3.3.1.** *For the linear multi-Hamiltonian*

$$Kz_t + Lz_x = Sz, \tag{3.30}$$

*where $S$ is a constant $n \times n$ real symmetric matrix, the dispersion relation between the wave number $\xi \in \mathbb{R}$ and frequency (not the two-form) $\omega \in \mathbb{R}$ is given by*

$$p(\xi, \omega) = \det(-i\omega K + i\xi L - S) = 0.$$

*Proof.* Assume $z = e^{i(\xi x - \omega t)}c$ where $c$ is a constant vector, is a solution to (3.30). Substitution yields

$$(-i\omega K + i\xi L - S)c = 0.$$

For non-trivial solutions the matrix on the left must have zero determinant. □

If there are any solutions to $p(\xi, \omega) = 0$ with $\xi$ real and $\omega$ complex but not real, then the PDE has solutions that grow without bound. For example, the dispersion relation for the wave equation, $u_{tt} - u_{xx} = 0$, is $p(\xi, \omega) = \omega(\omega^2 - \xi^2) = 0$, so all solutions are bounded. For the equation $u_{tt} + u_{xx} = 0$, the dispersion relation is $p(\xi, \omega) = \omega(\omega^2 + \xi^2) = 0$, so there are unbounded solutions.

**Theorem 3.3.2.** *The simple diamond scheme applied to the linear multi-Hamiltonian*

*equation has the dispersion relation between* $\mathcal{X}\Delta x, \Omega\Delta t \in [-2\pi, 2\pi]$ *defined by*

$$P(\mathcal{X}\Delta x, \Omega\Delta t) = p(h(\mathcal{X}\Delta x, \Omega\Delta t)) = p(h_1(\mathcal{X}\Delta x, \Omega\Delta t), h_2(\mathcal{X}\Delta x, \Omega\Delta t)) = 0,$$

*where* $p$ *is given in Lemma 3.3.1 and*

$$\begin{aligned} h(x,y) &= (h_1(x,y), h_2(x,y)), \\ h_1(x,y) &= \frac{4\sin\left(\frac{1}{2}x\right)}{\Delta x\left(\cos\left(\frac{1}{2}x\right) + \cos\left(\frac{1}{2}y\right)\right)}, \\ h_2(x,y) &= \frac{4\sin\left(\frac{1}{2}y\right)}{\Delta t\left(\cos\left(\frac{1}{2}x\right) + \cos\left(\frac{1}{2}y\right)\right)}. \end{aligned} \tag{3.31}$$

*Proof.* Assume that a solution to the simple diamond scheme given in (3.2) is $z_j^n = e^{i(\mathcal{X}j\Delta x - \Omega n\Delta t)}c$, where $c$ is a constant vector. If $j, n \in \mathbb{Z}$ then adding a multiple of $2\pi$ to either of $\Omega\Delta t$ or $\mathcal{X}\Delta x$ would not change $z_j^n$ so they could be restricted to $[-\pi, \pi]$. This situation occurs in the proof to the similar theorem for the simple box scheme. However, here $j, n \in \{\ldots -1, -1/2, 0, 1/2, 1, 3/2, \ldots\}$ so $\Omega\Delta t$ and $\mathcal{X}\Delta x$ can only be restricted to $[-2\pi, 2\pi]$. Substituting $z_j^n$ into (3.2) yields

$$\begin{aligned} &\left[\frac{1}{\Delta t}\left(e^{-i\Omega\frac{1}{2}\Delta t} - e^{i\Omega\frac{1}{2}\Delta t}\right)K + \frac{1}{\Delta x}\left(e^{i\mathcal{X}\frac{1}{2}\Delta x} - e^{-i\Omega\frac{1}{2}\Delta x}\right)L - \right. \\ &\qquad \left. \frac{1}{4}S\left(e^{-i\Omega\frac{1}{2}\Delta t} + e^{i\Omega\frac{1}{2}\Delta t} + e^{i\mathcal{X}\frac{1}{2}\Delta x} + e^{-i\mathcal{X}\frac{1}{2}\Delta x}\right)\right]c = 0 \\ \Rightarrow &\left[\frac{-2i}{\Delta t}\sin\left(\tfrac{1}{2}\Omega\Delta t\right)K + \frac{2i}{\Delta x}\sin\left(\tfrac{1}{2}\mathcal{X}\Delta x\right)L - \frac{S}{2}\left(\cos\left(\tfrac{1}{2}\Omega\Delta t\right) + \cos\left(\tfrac{1}{2}\mathcal{X}\Delta x\right)\right)\right]c = 0 \\ \Rightarrow &\left[\frac{-4i\sin\left(\frac{1}{2}\Omega\Delta t\right)K}{\Delta t\left(\cos\left(\frac{1}{2}\Omega\Delta t\right) + \cos\left(\frac{1}{2}\mathcal{X}\Delta x\right)\right)} + \frac{4i\sin\left(\frac{1}{2}\mathcal{X}\Delta x\right)L}{\Delta x\left(\cos\left(\frac{1}{2}\Omega\Delta t\right) + \cos\left(\frac{1}{2}\mathcal{X}\Delta x\right)\right)} - S\right]c = 0 \\ &\qquad \Rightarrow \left[-ih_2(\mathcal{X}\Delta x, \Omega\Delta t)K + ih_1(\mathcal{X}\Delta x, \Omega\Delta t)L - S\right]c = 0 \end{aligned}$$

For non-trivial solutions the matrix on the left must have zero determinant, so $p(h_1(\mathcal{X}\Delta x, \Omega\Delta t), h_2(\mathcal{X}\Delta x, \Omega\Delta t)) = 0$. □

**Theorem 3.3.3.** *The* $r = 1$ *diamond scheme applied to the linear multi-Hamiltonian equation has a dispersion relation between* $\tilde{\Omega}, \tilde{\mathcal{X}} \in [-2\pi, 2\pi]$ *defined by*

$$\tilde{P}(\tilde{\mathcal{X}}, \tilde{\Omega}) = \det\left(-2i\tan\left(\tfrac{1}{2}\tilde{\Omega}\right)\tilde{K} + 2i\tan\left(\tfrac{1}{2}\tilde{\mathcal{X}}\right)\tilde{L} - S\right) = 0.$$

*The tildes are reminders that this dispersion relation is in the* $(\tilde{x}, \tilde{t})$ *coordinates.*

*Proof.* The diamond method operates on a staggered grid: at each half step the current set of diamonds being worked upon are shifted across $\frac{\Delta x}{2}$ compared to the previous diamonds. So the $r = 1$ diamond scheme is not defined for every set of four points

$(j,n),(j+1,n),(j,n+1),(j+1,n+1)$. This means the usual *ansatz* $e^{i(\mathcal{X}j\Delta x-\Omega n\Delta t)}c$ cannot be used, rather *two* need to be employed. Secondly, the proof will be clearer in $(\tilde{x},\tilde{t})$ (or the discrete $(\tilde{j},\tilde{n})$) coordinates. The two *ansätze* are: $\tilde{z}^{\tilde{n}}_{\tilde{j}} = e^{i(\tilde{\mathcal{X}}\tilde{j}-\tilde{\Omega}\tilde{n})}c_1$ when $\tilde{n}$ is an integer, and $\tilde{z}^{\tilde{n}}_{\tilde{j}} = e^{i(\tilde{\mathcal{X}}\tilde{j}-\tilde{\Omega}\tilde{n})}c_2$ when $\tilde{j}$ is an integer. Where these two ansätze are defined, and the $(\tilde{j},\tilde{n})$ coordinate space are illustrated in Figure 3.8. Note, by periodicity, the variables $\tilde{\Omega}$ and $\tilde{\mathcal{X}}$ can be restricted to $[-2\pi,2\pi]$, and in $(\tilde{x},\tilde{t})$ coordinates $\Delta x=\Delta t=1$.

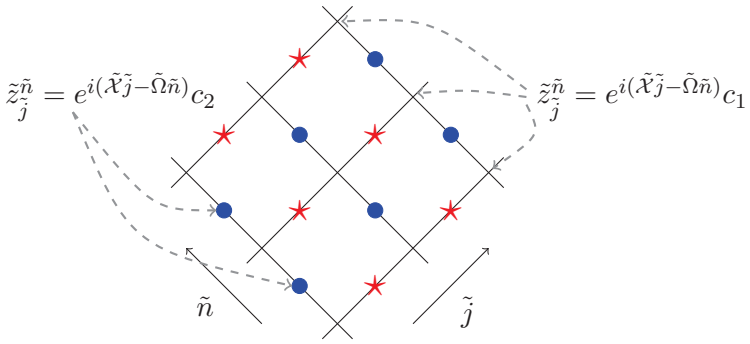



Figure 3.8: The $r=1$ diamond mesh, shown with $(\tilde{j},\tilde{n})$ coordinate axes. Because the solution is defined on the red stars and blue circles, which do not form a regular grid in $(\tilde{j},\tilde{n})$ space, the ansatz solution is given by $\tilde{z}^{\tilde{n}}_{\tilde{j}} = e^{i(\tilde{\mathcal{X}}\tilde{j}-\tilde{\Omega}\tilde{n})}c_1$ when $\tilde{n}$ is an integer (red stars), and $\tilde{z}^{\tilde{n}}_{\tilde{j}} = e^{i(\tilde{\mathcal{X}}\tilde{j}-\tilde{\Omega}\tilde{n})}c_2$ when $\tilde{j}$ is an integer (blue circles).

Substituting these two ansätze into (3.16)–(3.20) (or (3.28) because $r=1$) yields, after some simplification,

$$\begin{aligned}\tilde{K}\left(e^{-i\frac{1}{2}\tilde{\Omega}}-e^{i\frac{1}{2}\tilde{\Omega}}\right)c_1+\tilde{L}\left(e^{i\frac{1}{2}\tilde{\mathcal{X}}}-e^{-i\frac{1}{2}\tilde{\mathcal{X}}}\right)c_2 &= \\ \frac{S}{4}\left(e^{-i\frac{1}{2}\tilde{\Omega}}c_1+e^{i\frac{1}{2}\tilde{\Omega}}c_1+e^{i\frac{1}{2}\tilde{\mathcal{X}}}c_2+e^{-i\frac{1}{2}\tilde{\mathcal{X}}}c_2\right)&\\ \Rightarrow -2i\sin(\tfrac{1}{2}\tilde{\Omega})\tilde{K}c_1+2i\sin(\tfrac{1}{2}\tilde{\mathcal{X}})\tilde{L}c_2=\frac{S}{4}\left(2\cos(\tfrac{1}{2}\tilde{\Omega})c_1+2\cos(\tfrac{1}{2}\tilde{\mathcal{X}})c_2\right).&\end{aligned}\tag{3.32}$$

The alternating sum result (3.29) gives

$$\begin{aligned}e^{-i\frac{1}{2}\tilde{\Omega}}c_1-e^{-i\frac{1}{2}\tilde{\mathcal{X}}}c_2+e^{i\frac{1}{2}\tilde{\Omega}}c_1-e^{i\frac{1}{2}\tilde{\mathcal{X}}}c_2&=0\\ \Rightarrow\cos(\tfrac{1}{2}\tilde{\Omega})c_1-\cos(\tfrac{1}{2}\tilde{\mathcal{X}})c_2&=0.\end{aligned}$$

If $c_1$ and $c_2$ are independent eigenvectors, then $\cos(\frac{1}{2}\tilde{\Omega})=\cos(\frac{1}{2}\tilde{\mathcal{X}})=0$, and the solution collapses to only $\tilde{\Omega}=\pm\pi=\tilde{\mathcal{X}}$. So now let $c_1$ and $c_2$ be dependent: $c_1=\frac{\cos(\frac{1}{2}\tilde{\mathcal{X}})}{\cos(\frac{1}{2}\tilde{\Omega})}c_2$.

Substituting this into (3.32) gives

$$-2i\sin(\tfrac{1}{2}\tilde{\Omega})\tilde{K}\frac{\cos(\frac{1}{2}\tilde{\mathcal{X}})}{\cos(\frac{1}{2}\tilde{\Omega})}c_2 + 2i\sin(\tfrac{1}{2}\tilde{\mathcal{X}})\tilde{L}c_2 = \frac{S}{2}\left(\cos(\tfrac{1}{2}\tilde{\Omega})\frac{\cos(\frac{1}{2}\tilde{\mathcal{X}})}{\cos(\frac{1}{2}\tilde{\Omega})}c_2 + \cos(\tfrac{1}{2}\tilde{\mathcal{X}})c_2\right)$$
$$\Rightarrow -2i\tan(\tfrac{1}{2}\tilde{\Omega})\tilde{K}\cos(\tfrac{1}{2}\tilde{\mathcal{X}})c_2 + 2i\sin(\tfrac{1}{2}\tilde{\mathcal{X}})\tilde{L}c_2 = S\cos(\tfrac{1}{2}\tilde{\mathcal{X}})c_2.$$

Dividing through by $\cos(\frac{1}{2}\tilde{\mathcal{X}})$ gives the result. □

Recall Theorem 3.2.3: modulo initial conditions, the $r = 1$ diamond scheme and simple diamond scheme are equivalent. The following theorem shows that instead of directly calculating the dispersion relation for the $r = 1$ diamond scheme, the dispersion relation from the simple diamond scheme can simply be transformed from $(x, t)$ coordinates to $(\tilde{x}, \tilde{t})$ coordinates.

**Theorem 3.3.4.** *The simple and the $r = 1$ diamond schemes have identical dispersion relations, that is, $\tilde{P}(\tilde{\mathcal{X}}, \tilde{\Omega}) = P(\mathcal{X}, \Omega)$.*

*Proof.*

$$\begin{aligned}
z_j^n &= \tilde{z}_{\tilde{j}}^{\tilde{n}} \\
\Rightarrow e^{i(\mathcal{X}j\Delta x - \Omega n\Delta t)}c &= e^{i(\tilde{\mathcal{X}}\tilde{j} - \tilde{\Omega}\tilde{n})}c \\
\Rightarrow e^{i(\mathcal{X}x - \Omega t)}c &= e^{i(\tilde{\mathcal{X}}\tilde{x} - \tilde{\Omega}\tilde{t})}c \\
\Rightarrow e^{i(\mathcal{X}\frac{\Delta x(\tilde{x}-\tilde{t})}{2} - \Omega\frac{\Delta t(\tilde{t}+\tilde{x})}{2})}c &= e^{i(\tilde{\mathcal{X}}\tilde{x} - \tilde{\Omega}\tilde{t})}c \qquad \text{using (3.12)} \\
\Rightarrow e^{i(\frac{\Delta x\mathcal{X} - \Omega\Delta t}{2}\tilde{x} - \frac{\Delta x\mathcal{X} + \Omega\Delta t}{2}\tilde{t})}c &= e^{i(\tilde{\mathcal{X}}\tilde{x} - \tilde{\Omega}\tilde{t})}c,
\end{aligned}$$

thus

$$\tilde{\mathcal{X}} = \tfrac{\Delta x\mathcal{X} - \Delta t\Omega}{2} \quad \text{and} \quad \tilde{\Omega} = \tfrac{\Delta x\mathcal{X} + \Delta t\Omega}{2}. \tag{3.33}$$

Now

$$\begin{aligned}
&-i2\tan\left(\tfrac{\tilde{\Omega}}{2}\right)\tilde{K} + i2\tan\left(\tfrac{\tilde{\mathcal{X}}}{2}\right)\tilde{L} - S \\
&\quad = -i2\tan\left(\tfrac{\Delta x\mathcal{X} + \Omega\Delta t}{4}\right)\tilde{K} + i2\tan\left(\tfrac{\Delta x\mathcal{X} - \Omega\Delta t}{4}\right)\tilde{L} - S \\
&\quad = -i2\tan\left(\tfrac{\Delta x\mathcal{X} + \Omega\Delta t}{4}\right)\left(\tfrac{1}{\Delta t}K - \tfrac{1}{\Delta x}L\right) + i2\tan\left(\tfrac{\Delta x\mathcal{X} - \Omega\Delta t}{4}\right)\left(\tfrac{1}{\Delta t}K + \tfrac{1}{\Delta x}L\right) - S \\
&\quad = -i\frac{2}{\Delta t}\left(\tan\left(\tfrac{\Delta x\mathcal{X} + \Omega\Delta t}{4}\right) - \tan\left(\tfrac{\Delta x\mathcal{X} - \Omega\Delta t}{4}\right)\right)K + \\
&\qquad i\frac{2}{\Delta x}\left(\tan\left(\tfrac{\Delta x\mathcal{X} - \Omega\Delta t}{4}\right) + \tan\left(\tfrac{\Delta x\mathcal{X} + \Omega\Delta t}{4}\right)\right)L - S \\
&\quad = -i\frac{2}{\Delta t}\frac{2\sin\left(\frac{\Delta t\Omega}{2}\right)}{\cos\left(\frac{\Delta x\mathcal{X}}{2}\right) + \cos\left(\frac{\Delta t\Omega}{2}\right)}K + i\frac{2}{\Delta x}\frac{2\sin\left(\frac{\Delta x\mathcal{X}}{2}\right)}{\cos\left(\frac{\Delta x\mathcal{X}}{2}\right) + \cos\left(\frac{\Delta t\Omega}{2}\right)}L - S,
\end{aligned}$$

using (3.33), (3.15), and (in the last step) $\tan(\frac{a+b}{2}) = \frac{\sin(a)+\sin(b)}{\cos(a)+\cos(b)}$. □

The discrete dispersion relation determined in Theorem 3.3.2 allows the linear stability of the method to be determined for any PDE for which the continuous dispersion relation is known. This is now illustrated for the linear wave equation.

**Theorem 3.3.5.** *The simple diamond scheme applied to the wave equation is linearly stable when* $\lambda = \frac{\Delta t}{\Delta x} \leq 1$.

*Proof.* It must be shown that for all $\mathcal{X}\Delta x \in [-2\pi, 2\pi]$ the solutions $\Omega\Delta t \in [-2\pi, 2\pi]$ to $P(\mathcal{X}\Delta x, \Omega\Delta t) = 0$ are all real. From Theorem 3.3.2, $P(x, y) = p(h(x, y)) = p(h_1(x, y), h_2(x, y))$, and for the wave equation $p(\xi, \omega) = \xi^2 - \omega^2$. So

$$\begin{aligned} & P(\mathcal{X}\Delta x, \Omega\Delta t) = 0 \\ \Leftrightarrow\ & p(h_1(\mathcal{X}\Delta x, \Omega\Delta t), h_2(\Delta x, \Omega\Delta t)) = 0 \\ \Leftrightarrow\ & h_1(\mathcal{X}\Delta x, \Omega\Delta t) \pm h_2(\Delta x, \Omega\Delta t) = 0 \\ \Leftrightarrow\ & \frac{\sin(\frac{1}{2}\Omega\Delta t)}{\sin(\frac{1}{2}\mathcal{X}\Delta x)} = \pm\frac{\Delta t}{\Delta x} \\ \Leftrightarrow\ & \tfrac{1}{2}\Omega\Delta t = \sin^{-1}\left(\pm\lambda \sin(\tfrac{1}{2}\mathcal{X}\Delta x)\right) \\ \Leftrightarrow\ & \tfrac{1}{2}\Omega\Delta t = \pm\sin^{-1}\left(\lambda \sin(\tfrac{1}{2}\mathcal{X}\Delta x)\right). \end{aligned} \tag{3.34}$$

When $\lambda \leq 1$, for all $\mathcal{X}\Delta x \in [-2\pi, 2\pi]$, the right-hand side yields real results $\Omega\Delta t \in [-2\pi, 2\pi]$. □

The astute reader will have noticed that the last statement in the preceeding proof to Theorem 3.3.5 can be elaborated upon. It is true that when $\lambda \leq 1$ the right-hand side of (3.34) yields real results, and the scheme is stable. But how many real solutions? Due to the multi-valued nature of $\sin^{-1}$ there are in fact two solutions (or four solutions if one takes into account the $\pm$, or the second branch of the continuous dispersion relation). This other solution causes spurious, but stable, waves. These spurious waves were not noticed for most of the research. It wasn't until the diamond scheme was applied to the Schrödinger equation (Section 3.3.1) that the spurious solutions were noticed; simply because the spurious waves are unstable in that case. Spurious solutions are often the hallmark of a multi-step scheme [3], and it apears that the diamond scheme is multi-step in nature. See the conclusions in Chapter 6 for more discussion on this topic.

Note that the dispersion relation of the simple box scheme for the linear wave equation, given in (3.34), is the same as that of the explicit 5-point method.

### 3.3.1 Dispersion relation for linear Schrödinger equation

The linear Schrödinger equation is

$$i\Psi_t(x, t) + \Psi_{xx}(x, t) = 0,$$

where $x, t \in \mathbb{R}$. To express this in a multi-Hamiltonian form, let $\Psi = p+iq$ and consider the real and imaginary components

$$\begin{aligned} -q_t + p_{xx} &= 0 \\ p_t + q_{xx} &= 0, \end{aligned}$$

separately. Introduce $v = p_x, w = q_x$, then a multi-Hamiltonian form of the linear Schrödinger equation is

$$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_t \\ q_t \\ v_t \\ w_t \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_x \\ q_x \\ v_x \\ w_x \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} p \\ q \\ v \\ w \end{pmatrix}.$$

Recall in Lemma 3.3.1, the ansatz $z(x,t) = e^{i(\xi x - \omega t)}c$ is made, giving

$$(-i\omega K + i\xi L - S)c = 0,$$

which results in the dispersion relation

$$\begin{aligned} p(\xi, \omega) &= \det(-i\omega K + i\xi L - S) \\ &= (\omega - \xi^2)(\omega + \xi^2) = 0. \end{aligned} \tag{3.35}$$

The second branch of the dispersion relation, $\omega = -\xi^2$, is surprising because if the ansatz $\Psi(x,t) = e^{i(\xi x - \omega t)}$ is made and substituted into, not the multi-Hamiltonian form, but rather the original linear Schrödinger equation, the dispersion relation comes out to be $\omega = \xi^2$: there is no $\omega = -\xi^2$ branch. To explore this further, the eigenvector, $c$, belonging to the $\omega = -\xi^2$ branch will be found.

$$\begin{aligned} &(-i\omega K + i\xi L - S)c = 0 \\ \Rightarrow\ &(i\xi^2 K + i\xi L - S)c = 0 \qquad \text{by substituting in } \omega = -\xi^2 \\ \Rightarrow\ &\begin{pmatrix} 0 & -i\xi^2 & i\xi & 0 \\ i\xi^2 & 0 & 0 & i\xi \\ -i\xi & 0 & 1 & 0 \\ 0 & -i\xi & 0 & 1 \end{pmatrix} c = 0, \end{aligned}$$

which gives eigenvectors $c = (\frac{-1}{\xi}, \frac{-i}{\xi}, -i, 1)$. Only one is needed: choose $(-1, -i, -i, 1)$. Substituting this back into the ansatz and recovering the first and second components

($p$ and $q$) of $z$ gives

$$p(x,t) = -e^{i(\xi x+\xi^2 t)}$$
$$q(x,t) = -ie^{i(\xi x+\xi^2 t)},$$

which finally leads to

$$\Psi(x,t) = p(x,t) + iq(x,t) = 0,$$

which is the trivial solution. Hence the branch $\omega = -\xi^2$ can be ignored since it belongs to the trivial solution.

### 3.3.2 Stability of the simple and $r = 1$ diamond schemes

To assess the stability of the simple and $r = 1$ diamond schemes, recall from Theorem 3.3.2 that the discrete dispersion relation is

$$P(\mathcal{X}\Delta x, \Omega\Delta t) = p(h(\mathcal{X}\Delta x, \Omega\Delta t)) = p(h_1(\mathcal{X}\Delta x, \Omega\Delta t), h_2(\mathcal{X}\Delta x, \Omega\Delta t)) = 0,$$

where the $h$ map is defined in (3.31), and $p$ is defined in (3.35). Substituting these definitions in gives the discrete dispersion relation

$$\begin{aligned} h_1^2(\mathcal{X}\Delta x, \Omega\Delta t) &= h_2(\mathcal{X}\Delta x, \Omega\Delta t) \\ \Rightarrow \frac{4\sin^2\left(\frac{1}{2}\mathcal{X}\Delta x\right)}{\Delta x^2\left(\cos\left(\frac{1}{2}\mathcal{X}\Delta x\right)+\cos\left(\frac{1}{2}\Omega\Delta t\right)\right)^2} &= \frac{\sin\left(\frac{1}{2}\Omega\Delta t\right)}{\Delta t\left(\cos\left(\frac{1}{2}\mathcal{X}\Delta x\right)+\cos\left(\frac{1}{2}\Omega\Delta t\right)\right)} \\ \Rightarrow \frac{4\Delta t\sin^2\left(\frac{1}{2}\mathcal{X}\Delta x\right)}{\Delta x^2} &= \sin\left(\tfrac{1}{2}\Omega\Delta t\right)\left(\cos\left(\tfrac{1}{2}\mathcal{X}\Delta x\right)+\cos\left(\tfrac{1}{2}\Omega\Delta t\right)\right). \end{aligned} \tag{3.36}$$

The schemes will be stable iff for $\frac{1}{2}\mathcal{X}\Delta x \in [-\pi, \pi]$ the solutions $\frac{1}{2}\Omega\Delta t \in [-\pi, \pi]$ to (3.36) are all real.

**Theorem 3.3.6.** *The simple and $r = 1$ diamond schemes applied to the linear multi-Hamiltonian Schrödinger equation with periodic boundary conditions are stable when*

$$\Delta t < \frac{\Delta x^3}{24\sqrt{3}}.$$

*Proof.* To ease the notation let $x = \frac{1}{2}\mathcal{X}\Delta x$ and $y = \frac{1}{2}\Omega\Delta t$ in (3.36). Into (3.36) substitute complex exponentials for the trigonometric functions on $y$, and multiply by $e^{2iy}$, to give

$$\frac{1}{4}ie^{4iy} + \frac{1}{2}i\cos(x)e^{3iy} + \frac{4\Delta t\sin^2(x)}{\Delta x^2}e^{2iy} - \frac{1}{2}i\cos(x)e^{iy} - \frac{i}{4} = 0.$$

Making the substitution $z = e^{iy}$ yields the quartic equation

$$\frac{1}{4}iz^4 + \frac{1}{2}i\cos(x)z^3 + \frac{4\Delta t\sin^2(x)}{\Delta x^2}z^2 - \frac{1}{2}i\cos(x)z - \frac{i}{4} = 0.$$

The schemes will be stable iff the roots of this equation all lie on the unit circle ($\|z\| = 1 \Leftrightarrow y \in \mathbb{R}$). There are well-known conditions on the coefficients of polynomials to guarantee the roots are real, so map the unit circle to the real line using the substitution $z = \frac{w+i}{iw+1}$ and multiply through by $(1+iw)^4$ to give

$$\frac{\Delta x^2\cos(x) + 4\Delta t\sin^2(x)}{\Delta x^2}w^4 + 2w^3 + \frac{8\Delta t\sin^2(x)}{\Delta x^2}w^2 - 2w + \frac{4\Delta t\sin^2(x) - \Delta x^2\cos(x)}{\Delta x^2} = 0. \quad (3.37)$$

The schemes will be stable iff the four roots of this equation are real. From [59, pg. 51] if the discriminant

$$\Delta = \frac{128\sin^4(x)}{\Delta x^8}\left(3072\Delta t^4 - 564\Delta t^2\Delta x^4 + \Delta x^8 - \left(4096\Delta t^4 + 304\Delta t^2\Delta x^4 + \Delta x^8\right)\cos(2x) + 4\Delta t^2\left(256\Delta t^2 + \Delta x^4\right)\cos(4x)\right),$$

is positive, and the two polynomials

$$P = 8w_4w_2 - 3w_3^2,$$
$$\text{and } D = 64w_4^3w_0 - 16w_4^2w_2^2 + 16w_4w_3^2w_2 - 16w_4^2w_3w_1 - 3w_3^4,$$

where $w_i$ is the coefficient of $w^i$ in (3.37), are both negative, then the roots are real (and distinct). When $\Delta t = 0$,

$$\Delta = 256\sin^6(x),$$
$$P = -12,$$
$$\text{and } D = -8(5 + \cos(4x)).$$

So the roots are all real at $\Delta t = 0$. By expanding in Taylor series in $\Delta x$, for sufficiently small $\Delta x$, the smallest positive zeros of $\Delta$, $P$, and $D$ occur when $\Delta t$ is

$$\frac{\mathcal{X}\Delta x^3}{24\sqrt{3}} + \mathcal{O}\left(\Delta x^5\right),\ \frac{3}{2\mathcal{X}^2} + \mathcal{O}\left(\Delta x^2\right),\ \text{and } \frac{2+\sqrt{10}}{4\mathcal{X}^2} + \mathcal{O}\left(\Delta x^2\right)$$

respectively (the calculation was performed using Mathematica). Thus the schemes

will be stable when

$$\Delta t < \min\left(\frac{\mathcal{X}\Delta x^3}{24\sqrt{3}}, \frac{2+\sqrt{10}}{4\mathcal{X}^2}\right).$$

If the spatial frequencies are continuous, in other words $\mathcal{X} \in (0, \infty)$, no matter how small $\Delta t$ gets there will always be some low and high frequencies that are unstable. However, a discrete grid can only support a finite number of spatial frequencies. If a domain of width $\ell$ is discretized into $N$ diamonds of width $\Delta x = \frac{\ell}{N}$, then $\mathcal{X}$ is restricted to $1, 2, \ldots, N$. Thus

$$\min\left(\frac{2+\sqrt{10}}{4\mathcal{X}^2}\right) = \frac{2+\sqrt{10}}{4\ell^2}\Delta x^2,$$

and the constraint on $\Delta t$ for stability is

$$\Delta t < \frac{\Delta x^3}{24\sqrt{3}}.$$

□

Because the continuous dispersion relation is quadratic, it is somewhat surprising, and disappointing, that the stability condition involves the ratio $\frac{\Delta t}{\Delta x^3}$ and not $\frac{\Delta t}{\Delta x^2}$. However this stability result is straightforward to verify numerically: see confirmation by numerical experiment in Section 5.2.1. Also, this proof has shown that the discrete dispersion relation is a quartic. Thus, there are four possible frequencies ($\Omega\Delta t$) for each spatial frequency ($\mathcal{X}\Delta x$). This gives rise to spurious waves. See the conclusions in Chapter 6 for more discussion on spurious waves and the multi-step nature of the diamond scheme.

This has chapter introduced the diamond scheme. Foundational results regarding solvability, order, symplecticness and stability have been presented. A full discussion and conclusions can be found in Chapter 6. The next chapter covers some of the implementation details.

# Chapter 4

# Diamond implementation

This chapter discusses implementation details including data structures, numerical solvers, and how the diamond scheme runs in parallel.

## 4.1 Notation for diamond update equations

Recall that in equations (3.16)–(3.20) the variables $Z_i^j$, $X_i^j$, and $T_i^j$, are all in $\mathbb{R}^n$. So $Z$, $X$, and $T$ are $3^{\text{rd}}$-order tensors, and it would seem natural to treat them as such while translating these equations to computer code. However such a treatment can lead to overly complicated and difficult to read code. For instance, in Python using NumPy [78], solving (3.16)–(3.20) is equivalent to solving $f(Z) = 0$ where $f$ is defined by

```
def f(Z):
 X = np.tensordot(ainv, Z.transpose((1,0,2)) - Zl, 1).transpose((1,0,2))
 T = np.tensordot(ainv, Z - Zb, 1)
 return R(Z) -
    np.sum(K*T.reshape(r*r,-1)[:,np.newaxis],2).reshape(r,r,-1) -
    np.sum(L*X.reshape(r*r,-1)[:,np.newaxis],2).reshape(r,r,-1),
```

where `ainv` is $A^{-1}$, `Zl` is $\tilde{z}_\ell$, and `Zb` is $\tilde{z}^{\text{b}}$. While this is succinct (after all, there are three equations, so to represent this with only three lines of code is very tidy), it is not easy to read or maintain. Also, there is the issue of portability. Two versions of the code, one in Python and one in C, were developed; it would not have been easy to convert the Python 'tensor logic' to the C code. Moreover the author could not find a good fully-featured tensor library for C. To remedy this, the tensors were flattened to matrices.

To express equations (3.16)–(3.20) using matrices, define $Z \in \mathbb{R}^{n\times r^2}$

$$Z = \begin{pmatrix} Z^1_{11} & Z^1_{21} & \cdots & Z^1_{r1} & \cdots & Z^r_{11} & Z^r_{21} & \cdots & Z^r_{r1} \\ Z^1_{12} & Z^1_{22} & \cdots & Z^1_{r2} & \cdots & Z^r_{12} & Z^r_{22} & \cdots & Z^r_{r2} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ Z^1_{1n} & Z^1_{2n} & \cdots & Z^1_{rn} & \cdots & Z^r_{1n} & Z^r_{2n} & \cdots & Z^r_{rn} \end{pmatrix}, \tag{4.1}$$

where $Z^k_{ji}$ is the $i^{\text{th}}$ component of $Z^k_j$, which is the value of $Z$ at the $(j,\, k)$-position. In other words each row of $Z$ has the $Z$ values in row major order starting with the bottom (smallest time) row, and similarly for $X$ and $T$. For the wave equation (where the multi-Hamiltonian form has $n = 3$), these matrices would have three rows, and $r^2$ columns.

Let $zl \in \mathbb{R}^{n\times r}$ be the matrix $[\tilde{z}^j_{\ell i}]$

$$zl = \begin{pmatrix} \tilde{z}^1_{\ell 1} & \tilde{z}^2_{\ell 1} & \cdots & \tilde{z}^r_{\ell 1} \\ \tilde{z}^1_{\ell 2} & \tilde{z}^2_{\ell 2} & \cdots & \tilde{z}^r_{\ell 2} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{z}^1_{\ell n} & \tilde{z}^2_{\ell n} & \cdots & \tilde{z}^r_{\ell n} \end{pmatrix},$$

and similarly for $zb$, $zt$, and $zr$. Again these matrices have one row per component, but now there are only $r$ columns. Let $Zl \in \mathbb{R}^{n\times r^2}$ be $zl$ tiled $r$ times

$$Zl = \begin{pmatrix} zl & zl & \cdots & zl \end{pmatrix},$$

and similar for $Zb$. Define $Z[i, a, b]$ as the $i^{\text{th}}$ row of $Z$ reshaped into an $a \times b$ matrix, and as a shortcut $Z[i] = Z[i, r, r]$,

$$Z[i] = \begin{pmatrix} Z^1_{1i} & Z^1_{2i} & \cdots & Z^1_{ri} \\ Z^2_{1i} & Z^2_{2i} & \cdots & Z^2_{ri} \\ \vdots & \vdots & \ddots & \vdots \\ Z^r_{1i} & Z^r_{2i} & \cdots & Z^r_{ri} \end{pmatrix}.$$

Equations (3.16)–(3.18) can now be written as $n$ matrix equations

$$Z[i]^T = Zl[i] + AX[i]^T, \tag{4.2}$$

$$Z[i] = Zb[i] + AT[i], \tag{4.3}$$

$$R(Z) = \tilde{K}T + \tilde{L}X, \tag{4.4}$$

for $i = 1, \ldots, n$, where $R(Z) \in \mathbb{R}^{n \times r^2}$ is a matrix with columns $\nabla S(Z_j^k)$

$$R(Z) = \begin{pmatrix} \nabla S(Z_1^1) & \cdots & \nabla S(Z_r^1) & \cdots & \nabla S(Z_1^r) & \cdots & \nabla S(Z_r^r) \end{pmatrix}. \tag{4.5}$$

Equation (4.2) can be solved to give $X[i] = (Z[i] - Zl[i]^T)A^{-T}$, and (4.2) to give $T[i] = A^{-1}(Z[i] - Zb[i]^T)$. Using these, $X$ and $T$ can be formed, row by row, and eliminated from (4.4).

Update equations (3.19) and (3.20) can be written as

$$\begin{aligned} zr_{i,:} &= zl_{i,:} + b^T X[i]^T, \\ zt_{i,:} &= zb_{i,:} + b^T T[i], \end{aligned}$$

for $i = 1, \ldots, n$, where the MATLAB index, or slice, notation $i,:$ has been used to denote the $i^{\text{th}}$ row.

For (3.16)–(3.20) on the extended domain (corners included), define $Z_e \in \mathbb{R}^{n \times (r+1)^2}$, $X_e, T_e \in \mathbb{R}^{n \times r(r+1)}$,

$$Z_e = \begin{pmatrix} \tilde{z}_{01}^b & \tilde{z}_{11}^b & \cdots & \tilde{z}_{r1}^b & \tilde{z}_{\ell 1}^1 & Z_{11}^1 & \cdots & Z_{r1}^1 & \cdots & \tilde{z}_{\ell 1}^r & Z_{11}^r & \cdots & Z_{r1}^r \\ \tilde{z}_{02}^b & \tilde{z}_{12}^b & \cdots & \tilde{z}_{r2}^b & \tilde{z}_{\ell 2}^1 & Z_{12}^1 & \cdots & Z_{r2}^1 & \cdots & \tilde{z}_{\ell 2}^r & Z_{12}^r & \cdots & Z_{r2}^r \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{z}_{0n}^b & \tilde{z}_{1n}^b & \cdots & \tilde{z}_{rn}^b & \tilde{z}_{\ell n}^1 & Z_{1n}^1 & \cdots & Z_{rn}^1 & \cdots & \tilde{z}_{\ell n}^r & Z_{1n}^r & \cdots & Z_{rn}^r \end{pmatrix}, \tag{4.6}$$

$$X_e = \begin{pmatrix} X_{11}^0 & \cdots & X_{r1}^0 & X_{11}^1 & \cdots & X_{r1}^1 & \cdots & X_{11}^r & \cdots & X_{r1}^r \\ X_{12}^0 & \cdots & X_{r2}^0 & X_{12}^1 & \cdots & X_{r2}^1 & \cdots & X_{12}^r & \cdots & X_{r2}^r \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ X_{1n}^0 & \cdots & X_{rn}^0 & X_{1n}^1 & \cdots & X_{rn}^1 & \cdots & X_{1n}^r & \cdots & X_{rn}^r \end{pmatrix},$$

$$T_e = \begin{pmatrix} T_{01}^1 & T_{11}^1 & \cdots & T_{r1}^1 & \cdots & T_{01}^r & T_{11}^r & \cdots & T_{r1}^r \\ T_{02}^1 & T_{12}^1 & \cdots & T_{r2}^1 & \cdots & T_{02}^r & T_{12}^r & \cdots & T_{r2}^r \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ T_{0r}^1 & T_{1r}^1 & \cdots & T_{rr}^1 & \cdots & T_{0r}^r & T_{1r}^r & \cdots & T_{rr}^r \end{pmatrix}.$$

Note that $X_e$ extends onto only the bottom boundary, and $T_e$ extends onto only the left boundary. Extend $zl$ by including the bottom most point, so $zl_e \in \mathbb{R}^{n\times(r+1)}$

$$zl_e = \begin{pmatrix} \tilde{z}^0_{\ell 1} & \tilde{z}^1_{\ell 1} & \tilde{z}^2_{\ell 1} & \cdots & \tilde{z}^r_{\ell 1} \\ \tilde{z}^0_{\ell 1} & \tilde{z}^1_{\ell 2} & \tilde{z}^2_{\ell 2} & \cdots & \tilde{z}^r_{\ell 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{z}^0_{\ell 1} & \tilde{z}^1_{\ell n} & \tilde{z}^2_{\ell n} & \cdots & \tilde{z}^r_{\ell n} \end{pmatrix},$$

and similarly for $zr_e$, $zb_e$, and $zt_e$. The tiled matrices $Zl_e, Zb_e \in \mathbb{R}^{n\times r(r+1)}$ are defined similarly to $Zl$ and $Zb$

$$Zl_e = \begin{pmatrix} zl_e & zl_e & \cdots & zl_e \end{pmatrix},$$

and similar for $Zb_e$. The extended versions of (4.2) and (4.3) are

$$(Z_e[i, r+1, r+1]_{:,1:})^T = Zl_e[i, r, r+1] + AX_e[i, r+1, r]^T, \tag{4.7}$$

$$Z_e[i, r+1, r+1]_{1:,:} = Zb_e[i, r, r+1] + AT_e[i, r, r+1]. \tag{4.8}$$

When unextended equations (4.2)–(4.4) have been solved, $Z_e$ can be formed using (4.6) then (4.7) and (4.8) solved to give $X_e$ and $T_e$. Finally these are the extended update equations

$$zr_{e_{i,:}} = zl_{e_{i,:}} + b^T X_e[i, r+1, r]^T,$$
$$zt_{e_{i,:}} = zb_{e_{i,:}} + b^T T_e[i, r, r+1].$$

## 4.2 Numerically solving diamond update equations

Solving (4.2)–(4.4) numerically requires a nonlinear solver, such as Newton's method. All Python code used the SciPy [33] routine `fsolve`, which is a wrapper around the MINPACK [55] `hydrd` and `hybrj` algorithms, which are based on the Powell hybrid method [58]. The diamond scheme was also implemented in the C language. The C code used the GNU Scientific Library [21] routines `gsl_multiroot_fsolver_hybrids` and `gsl_multiroot_fdfsolver_gnewton` which again are wrappers around the MINPACK `hydrd` and `hybrj` algorithms.

The `hydrd` algorithm approximates the Jacobian, whereas `hybrj` requires the exact Jacobian. To calculate that, (4.2)–(4.4) need to be functions of vectors, not matrices

or tensors. To that end, flatten, row by row, the matrix given in (4.1), to give

$$Z = (Z^1_{11}, Z^1_{21}, \ldots, Z^1_{r1}, Z^2_{11}, Z^2_{21}, \ldots, Z^2_{r1}, \ldots, Z^r_{11}, Z^r_{21}, \ldots, Z^r_{r1}, \ldots, \\ Z^1_{1n}, Z^1_{2n}, \ldots, Z^1_{rn}, Z^2_{1n}, Z^2_{2n}, \ldots, Z^2_{rn}, \ldots, Z^r_{1n}, Z^r_{2n}, \ldots, Z^r_{rn}) \in \mathbb{R}^{nr^2}.$$

Now define $g\colon \mathbb{R}^{nr^2} \to \mathbb{R}^{nr^2}$ as

$$g(Z) = (R(Z) - \tilde{K}T - \tilde{L}X)_{\text{-}}$$

with

$$X[i] = (Z[i] - Zl[i]^T)A^{-T}$$
$$T[i] = A^{-1}(Z[i] - Zb[i]^T),$$

where the subscript - in the definition of $g$ means to flatten the matrix row by row as $Z$ was flattened. Now solving (4.2)–(4.4) is simply solving $g(Z) = 0$.

By writing out the components of $A^{-1}$ and proceeding element by element it is laborious, but not complicated, to calculate the Jacobian of $g$

$$g'(Z) = (R(Z)_{\text{-}})' - \tilde{K} \otimes A^{-1} \otimes I_{r\times r} - \tilde{L} \otimes I_{r\times r} \otimes A^{-1}.$$

Using the definition of $R$ in (4.5), for the one-dimensional wave equation given in (1.5)

$$R(Z) = \begin{pmatrix} -f'(u^1_1) & \cdots & -f'(u^1_r) & \cdots & -f'(u^r_1) & \cdots & -f'(u^r_r) \\ v^1_1 & \cdots & v^1_r & \cdots & v^r_1 & \cdots & v^r_r \\ -w^1_1 & \cdots & -w^1_r & \cdots & -w^r_1 & \cdots & -w^r_r \end{pmatrix},$$

thus

$$R(Z)_{\text{-}} = \left(-f'(u^1_1), \ldots, -f'(u^r_r), v^1_1, \ldots, v^r_r, -w^1_1, \ldots, -w^r_r\right),$$
$$\Rightarrow (R(Z)_{\text{-}})' = \operatorname{diag}(-f''(u^1_1), \ldots, -f''(u^r_r), 1, \ldots, 1, -1, \ldots, -1).$$

If $\tilde{K}$ and $\tilde{L}$ are not constant, but depend on $\tilde{x}$ and $\tilde{t}$, then both $g$ and its Jacobian are more complicated. Evaluate $\tilde{K}$ at the internal grid points $(\tilde{x}, \tilde{t}) = (c_i, c_j)$ for $i, j = 1, \ldots, r$ (the same locations as $Z^j_i$ are evaluated at), and put into a vector with elements $\tilde{K}_{r(j-1)+i}$. Despite overloading the $\tilde{K}$ notation, call this vector $\tilde{K}$. Similarly with $\tilde{L}$. The products $\tilde{K}T$ and $\tilde{L}X$ that appear in (3.18) must be modified: the $i^{\text{th}}$

column of $\tilde{K}T$ is now $\tilde{K}_i T$, similarly for $\tilde{L}X$. The Jacobian of $g$ is now

$$g'(Z) = (R(Z)_{\cdot})' - \left(\sum \tilde{K}_i \otimes \Delta_i\right) \star \left(\mathbf{1}_{n\times n} \otimes (A^{-1} \otimes I_{r\times r})\right) - \left(\sum \tilde{L}_i \otimes \Delta_i\right) \star \left(\mathbf{1}_{n\times n} \otimes (I_{r\times r} \otimes A^{-1})\right),$$

where $\Delta_i$ is a $r^2 \times r^2$ matrix of zeros except with the $i^{\text{th}}$ row all ones, the operator $\star$ is point-wise multiplication, and $\mathbf{1}_{n\times n}$ is the $n \times n$ matrix of ones.

### 4.2.1 Data structures

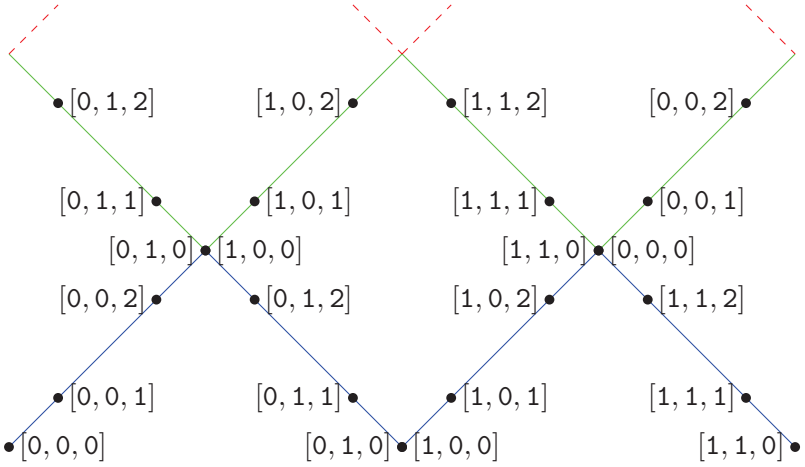



Figure 4.1: The domain divided into diamonds annotated with labels showing how each point is indexed. In this example $r = 2$ and there are only two diamonds across the domain. The points are indexed by diamond number counting from the left, leg number (0 for left leg, 1 for right leg), and finally point number, counting from the bottom. For example $[\mathtt{i}, 1, \mathtt{j}]$ is the $\mathtt{j}$th point from the bottom on the right leg of the $\mathtt{i}$th diamond. Because the boundary conditions are periodic the right-most green edge is considered the left edge of the left-most diamond.

This section details the $z$ data structure and how the algorithm works across all the diamonds. Figure 4.1 expands upon Figure 3.5 to show how the data structures are laid out. The domain is divided into two diamonds, the boundary conditions are periodic, and each point is annotated to show how each point is indexed. The variable `z` contains the values along the blue lines, `zhalf` along the green lines, and `znew` contains the values along the red lines (removed for clarity). The points are indexed by diamond number counting from the left, leg number (0 for left leg, 1 for right leg), and finally point number, counting from the bottom. For example $\mathtt{z}[\mathtt{i}, 1, \mathtt{j}]$ is the value of `z` at the $\mathtt{j}^{\text{th}}$ point from the bottom on the right leg of the $\mathtt{i}^{\text{th}}$ diamond.

At each step, by solving the equations given in Section 4.1, the `z` values are used to

calculate the `zhalf` values, which are used in turn to calculate the `znew` values. Finally `znew` is copied to `z` and the step is complete. If the solution is to be printed out at a step the values $\mathtt{z}[:,:,0]$ are used after taking the arithmetic mean of coinciding points. In other words the $\mathtt{i}^{\text{th}}$ value of the solution is the mean of $\mathtt{z}[\mathtt{i}-1,1,0]$ and $\mathtt{z}[\mathtt{i},0,0]$ except when $\mathtt{i}=0$ and then by periodicity the value is the mean of $\mathtt{z}[-1,1,0]$ (last point) and $\mathtt{z}[0,0,0]$.

## 4.3 Parallel diamond scheme

The diamond scheme lends itself to being implemented in parallel. To calculate the values on the top two edges of a diamond only the values on the bottom two edges are needed. There is no inter-diamond communication necessary during this calculation, and this calculation is where most of the work is.

For a parallel diamond scheme implementation, the domain is divided into strips, finite in width in the $x$ direction and potentially infinite in the $t$ direction. The width of the domain, $L$ is assumed to be a multiple of $\Delta x$. There are $N = \frac{L}{\Delta x}$ diamonds in the each row, and $p$ processors. The $N$ diamonds are divided as equally as possible into $p$ contiguous regions. If $N$ is a multiple of $p$ then each processor will get an equal number of diamonds to work on. Otherwise $p-n$ processors get $k$ diamonds, and $n$ processors get $k+1$ diamonds, where $n = N - pk$ and $k = \lfloor \frac{N}{p} \rfloor$.

Each processor calculates the solution on its strip of diamonds. At each half time-step the solution at the very left or right edge of the strip must be passed to a neighbour. This results in $p$ transmits of a vector $r+1$ in length (in the one-dimensional wave equation case each element of the vector would have length 3). At the next half step the left hand edge must be passed to the left and received from the right, another $p$ transmits. If the solution is to be printed each processor must send a number of values—twice the number of diamonds in its sub-domain—to one processor. This communication was achieved with a MPI gather. The receiving processor forms the solution by taking the arithmetic mean of coinciding points, as discussed at the end of Section 4.2.1, and prints the result. Since this processor has very slightly more work to perform it is desirable to ensure it is one of the $p-n$ processors that receives $k$ diamonds to work on. The communication costs of this gather could be roughly halved if each processor resolved the ambiguities in the solution values at the coinciding points before sending the results to the single processor. However, the ambiguities at the edges of each sub-domain can not be resolved without communication with a neighbour. This communication occurs at the start of the next step, so no extra communication would be involved, but it is considerably tidier to get the single processor to resolve all these ambiguities.

To test how much the diamond scheme could benefit from running in parallel it was run on the New Zealand eScience Infrastructure's (NeSI's) Pan Intel Linux cluster

physically located at the University of Auckland, New Zealand. At the time of use (2015) the cluster had approximately 6000 cores each running somewhere between 2GHz and 3GHz with most around 2.7GHz or 2.8GHz. Most cores had at least 10GB of RAM available, far more than the diamond scheme ever required. Due to the busyness of the cluster it was impractical to request specific CPUs for each job, thus there was a certain variability in timing tests simply because of the different speeds of the CPUs available. However the uni-processor jobs, arguably the most important while testing parallel speed-ups, did run on the most common 2.7GHz or 2.8GHz processors.

Because the Pan cluster was a completely different environment to where the diamond scheme code was developed and mostly run, the output from the first run was compared to output from a test run done locally. This was simply to check that the code behaved in an identical manner. After this initial successful test no output was generated from the parallel test runs. The diamond scheme was initialized with the diamond initialization method detailed in Section 5.1, $r$ was set to 5, the number of iterations was 1000, $\Delta t = 0.05$, and the periodic Sine–Gordon problem from Table 5.2 was used. Despite the cluster having approximately 6000 cores, by trial and error it was apparent that only about a maximum of 300 or 400 cores could be readily available on demand. So each trial consisted of nine runs with the number of cores being 1, 3, 7, 20, 56, 100, 150, 300, and 350. For each run the wall-clock time was recorded using the Unix `date` command, the program run, and then the wall-clock time checked again. Each trial (set of nine runs) was performed twice with a couple of days in between each trial, and the time results averaged.

According to Amdahl's law [1], for a particular problem size, if $n$ is the number of cores, and $B \in [0, 1]$ is the fraction of the algorithm that is strictly serial, then the theoretical time $T(n)$ for the algorithm to run on $n$ cores is

$$T(n) = T(1)\left(B + \frac{1}{n}(1 - B)\right).$$

Thus the theoretical speed-up $S(n)$ is

$$S(n) = \frac{T(1)}{T(n)} = \frac{1}{B + \frac{1}{n}(1 - B)}. \tag{4.9}$$

Letting $n \to \infty$ gives a theoretical maximum speed-up of $\frac{1}{B}$. By increasing $n$ until the speed-up begins to tail-off it is possible to estimate $B$. For a perfectly parallelizeable algorithm the speed-up should be equal to the number of cores used. In the first trial, shown in Figure 4.2, $\Delta x$ was such that there were 4000 diamonds across the domain. As the ratio of the number of cores to the amount of work (number of diamonds across the domain) increases, one would expect the speed-up to diverge from the perfect speed-up

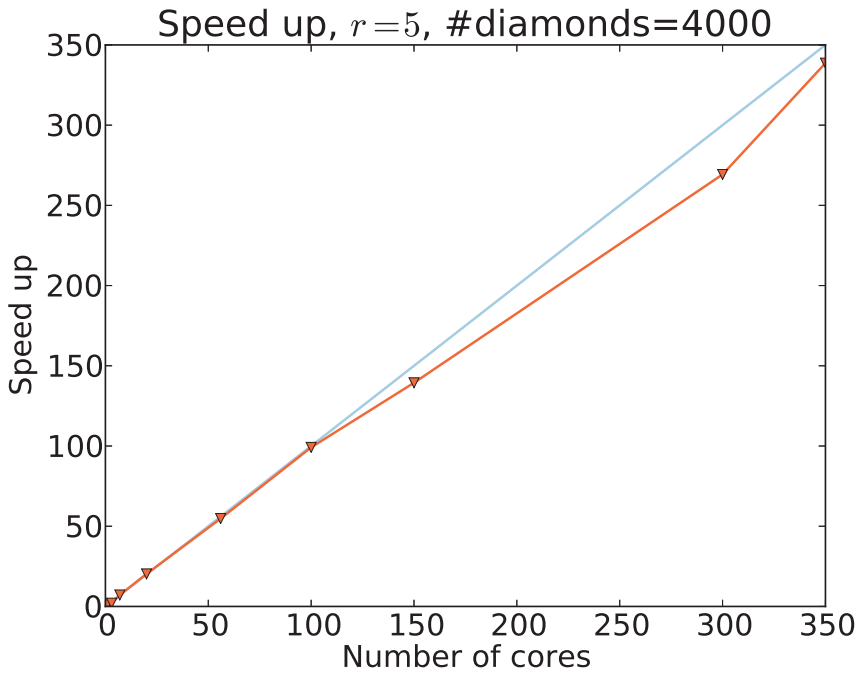



Figure 4.2: Speed-up of the diamond scheme versus the number of cores for the code running on the NeSI Pan cluster. Code that was perfectly parallelizeable would have the speed-up equal to the number of cores (the blue line). As the ratio of the number of cores to the amount of work (number of diamonds across the domain) increases one would expect the speed-up to deviate from the perfect blue line. In this region, though, the speed-up is very good.

line. This is because the overhead in communication will gradually swamp the gains in computation time. For this trial the speed-up is still very good and it is impossible to estimate $B$, the fraction of the algorithm that is strictly serial. Ideally, the number of cores would be increased until a deviation from the perfectly parallelizable line could be reliably detected, however no more cores were easily available.

So, instead of increasing the number of cores, the number of diamonds was decreased. Figure 4.3 shows the second trial where the number of diamonds across the domain was decreased to 1000. One of the trials included two extra runs with $n = 450$ and $n = 499$. Because it took many days for these runs to begin executing, the second trial did not include these large runs, and no averaging could take place. This figure shows the speed-up reaching approximately 250 before beginning to tail off. So for this size of problem, from (4.9) this equates to $B \approx 0.4\%$, which is remarkably low. The conclusion is that the diamond scheme is exceptionally parallelizable.

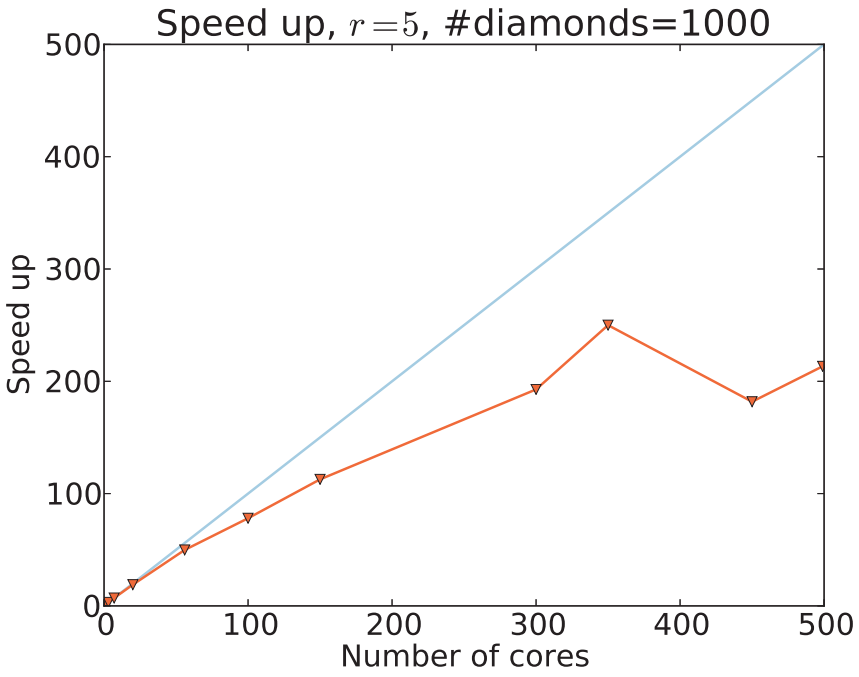



Figure 4.3: Speed-up of the diamond scheme versus the number of cores for the code running on the NeSI Pan cluster. Code that was perfectly parallelizeable would have the speed-up equal to the number of cores (the blue line). The number of diamonds across the domain is 1000, so at the maximum number of cores here, most cores have only two diamonds to work on. The overheads in communicating between cores is causing the speed-up to deviate from the perfect blue line. Note that the speed up was calculated from a single run (not an average of two runs) for the 450 and 499 number of cores runs.

This chapter has discussed implementation details including translating the mathematical equations into computer code, numerical solvers, and data structures. Results from a parallel implementation have been presented. For a full discussion and conclusions see Chapter 6. The next chapter covers the initialization scheme, boundary conditions and numerical testing.

# Chapter 5

# Diamond initialization and boundary treatment

This chapter discusses implementation details, including an initialization scheme, treatment of boundary conditions, and presents results of numerical tests.

## 5.1 Diamond scheme initialization

Instead of initializing the diamond scheme with a forward Euler step, which will affect the order of the scheme, or the exact solution, which may not be known, the diamond scheme itself can be used for initialization. One such initialization method is developed and tested in this section. It will be referred to as the *diamond initialization* scheme. The $x$-axis cuts the first row of diamonds in half, yielding a row of triangles. Each triangle can be mapped via a nonlinear mapping to the unit square, then the usual set of equations (3.16)– (3.20) solved, giving values for $z$ on the top-left and top-right edges of the triangle.

The transformation, $x = \tilde{x} - \tilde{t}, t = \tilde{x}\tilde{t}$, illustrated in Figure 5.1, takes the unit square to the triangle $(-1, 0), (1, 0), (0, 1)$. Adding a translation and scaling results in the map

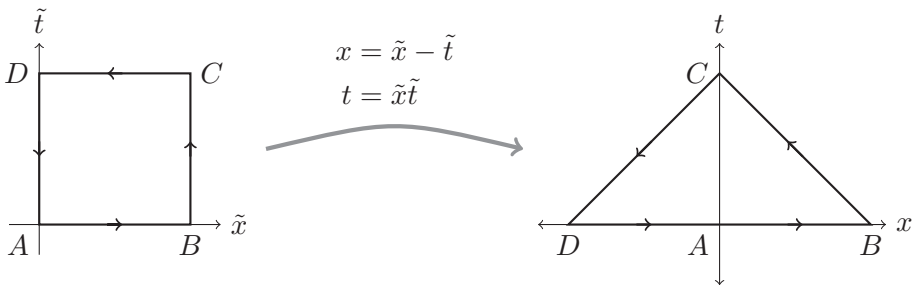



Figure 5.1: The unit square under the map $x = \tilde{x} - \tilde{t}, t = \tilde{x}\tilde{t}$.

$$x = \frac{\Delta x}{2}(\tilde{x} - \tilde{t}) + b \tag{5.1}$$

$$t = \frac{\Delta t}{2}\tilde{x}\tilde{t} \tag{5.2}$$

which takes the unit square to the triangle $(b - \frac{\Delta x}{2}, 0), (b + \frac{\Delta x}{2}, 0), (b, \frac{\Delta t}{2})$. Recall that the transformed $K$ and $L$ are given by

$$\tilde{K} = g_t K + g_x L$$
$$\tilde{L} = f_t K + f_x L,$$

where $(\tilde{x}, \tilde{t}) = (f(x,t), g(x,t))$. The inverse functions $f$ and $g$ are not needed[1], only their partial derivatives. The chain rule is used to calculate these. Let $X = (x,t)$, $\tilde{X} = (\tilde{x}, \tilde{t})$, and a mapping $T$ such that $\tilde{X} = T(X)$. So

$$T^{-1}(\tilde{X}) = \begin{pmatrix} \frac{\Delta x}{2}(\tilde{x} - \tilde{t}) + b \\ \frac{\Delta t}{2}\tilde{x}\tilde{t} \end{pmatrix},$$

and the inverse of its Jacobian is

$$(T^{-1})'^{-1}(\tilde{X}) = \frac{2}{\Delta x \Delta t(\tilde{x} + \tilde{t})} \begin{pmatrix} \tilde{x}\Delta t & \Delta x \\ -\tilde{t}\Delta t & \Delta x \end{pmatrix}.$$

The chain rule is

$$T'(X) = (T^{-1})'^{-1}(\tilde{X})$$
$$\Rightarrow \begin{pmatrix} f_x & f_t \\ g_x & g_t \end{pmatrix} = \frac{2}{\Delta x \Delta t(\tilde{x} + \tilde{t})} \begin{pmatrix} \tilde{x}\Delta t & \Delta x \\ -\tilde{t}\Delta t & \Delta x \end{pmatrix}.$$

Using these partial derivatives the transformed $K$ and $L$ can be written as

$$\tilde{K} = \tfrac{2}{\Delta x \Delta t(\tilde{x}+\tilde{t})} \left(\Delta x K - \tilde{t}\Delta t L\right)$$
$$\tilde{L} = \tfrac{2}{\Delta x \Delta t(\tilde{x}+\tilde{t})} \left(\Delta x K + \tilde{x}\Delta t L\right).$$

For initializing the diamond scheme, $z$ values are needed on the bottom zig-zag (Figure 3.5) spaced according to the Runge–Kutta vector $c$. Because the above map (5.1) (and its inverse of course) are linear along the edges, this same spacing can be used in

[1] but here they are if required for some other purpose

$$\tilde{x} = \frac{x-b}{\Delta x} + \sqrt{\tfrac{(x-b)^2}{\Delta x^2} + \tfrac{2t}{\Delta t}} \equiv f(x,t)$$
$$\tilde{t} = \frac{b-x}{\Delta x} + \sqrt{\tfrac{(x-b)^2}{\Delta x^2} + \tfrac{2t}{\Delta t}} \equiv g(x,t)$$

$(\tilde{x}, \tilde{t})$ space.

Figure 5.2 shows the error of exact and diamond initialization as $\Delta t$ is reduced while keeping the Courant number $\frac{\Delta t}{\Delta x} = \frac{1}{2}$. The integration time $T = 1.5$ is twice the largest time step. It is apparent that for this problem the diamond initialization is equal, or better, than exact initialization. Table 5.1 shows the numerical order of the two initialization methods.

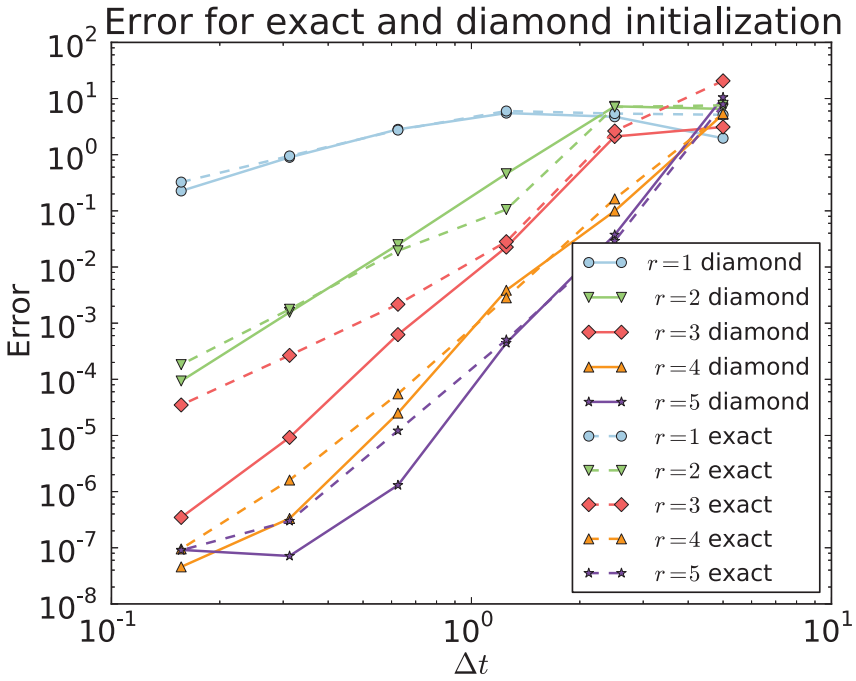



Figure 5.2: The error of the diamond scheme initialized using the exact and diamond methods applied to the multisymplectic Hamiltonian PDE arising from the Sine–Gordon equation. The true solution was the so-called *breather* on the domain $[-30, 30]$. The Courant number is fixed at $\frac{1}{2}$ as $\Delta t$ is decreased. For this problem the diamond initialization is equal or better than exact initialization.

## 5.2 Numerical testing

The extended-to-the-corners diamond scheme (4.6)–(4.8), for various values of $r$, is applied to four sample problems with periodic boundary conditions, numerical errors calculated, and the numerical order of the scheme presented. When checking how well a numerical scheme is performing it is useful to know the exact solution to the PDE. These and the sample problems are summarized in Table 5.2. To calculate the error

| $r$ | Order | |
|---|---|---|
| | exact | diamond |
| 1 | 1 | 2 |
| 2 | 3 | 3 |
| 3 | 3 | 5 |
| 4 | 5 | 5 |
| 5 | 5 | 6 |

Table 5.1: Numerical order, read off Figure 5.2, of the diamond scheme initialized with the exact solution and the diamond scheme initialization. It is apparent that for this problem the diamond initialization performs as well, or better, than the exact initialization. The order appears to be $r+1$ for most $r$ (for $r = 4$ the order is $r+2$ for some reason), whereas for the exact initialization the order is $r$ ($r$ odd) and $r+1$ ($r$ even).

| Name | Equation | Range | Solution |
|---|---|---|---|
| Esin | $u_{tt} + u_{xx} = 0$ | $0 \le x \le 2\pi$ | $e^{2\sin(x-t-3)}$ |
| Sincos | $u_{tt} + u_{xx} = 0$ | $0 \le x \le 2\pi$ | $\sin(x)\cos(t)$ |
| Coscos | $u_{tt} + u_{xx} = -u$ | $0 \le x \le \pi$ | $\cos(2x)\cos(\sqrt{5}t)$ |
| Sine–Gordon | $u_{tt} + u_{xx} = -\sin(u)$ | $-200 \le x \le 200$ | $4\arctan\left(\frac{\sin\left(\frac{t}{\sqrt{2}}\right)}{\cosh\left(\frac{x}{\sqrt{2}}\right)}\right)$ |

Table 5.2: Sample problems.

between the computed solution, $\tilde{u}$, and the true solution, $u$, the discrete 2-norm

$$E^2 = \frac{b-a}{N}\sum_{i=1}^{N}\left(\tilde{u}_i - u(a+i\Delta x, T)\right)^2, \tag{5.3}$$

was used. The number of diamonds at each time level is $N = 10, 20, \ldots, 1280$, and the integration time, $T = 1.5$ is twice the largest time step. The Courant number $\frac{\Delta t}{\Delta x} = \frac{1}{2}$ is held fixed as $\Delta t$ was decreased. The initialization scheme from Section 5.1 was used.

The results for the global numerical error are shown in Figures 5.3–5.6. Table 5.3 shows the numerical order of the diamond scheme for these problems. It is apparent that for this problem, the order is $r+1$. The super-convergence property of the underlying Gauss Runge–Kutta method, which has order $2r$, has been lost. But this is expected, since the underlying scheme has super-convergence at the end points, but not at the internal stages where it has order $r+1$. It is these internal stages, the diamond edge points, that the diamond scheme operates on.

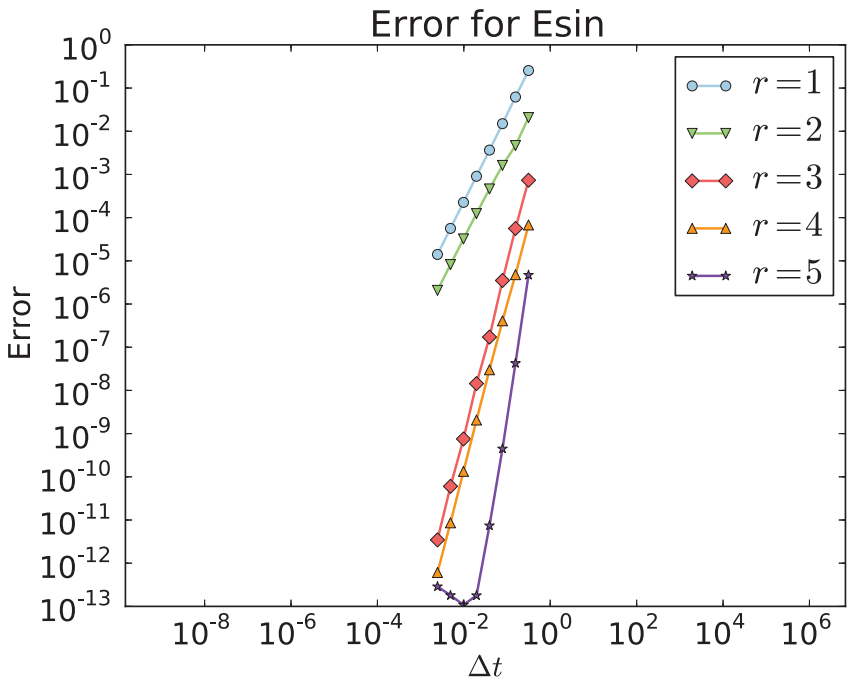



Figure 5.3: The error of the diamond scheme with varying $r$ applied to the Esin problem (see Table 5.2). The Courant number is fixed at $\frac{1}{2}$ as $\Delta t$ is decreased. Table 5.3 summarizes the numerical order by reporting the slope of these lines. The numerical method cannot reduce the error any further than machine precision ($\approx 10^{-16}$ for double floating point arithmetic). When the error has reached approximately machine precision, increasing $r$ and/or decreasing $\Delta t$, just causes 'noise', and these parts of the error plot can be safely ignored.

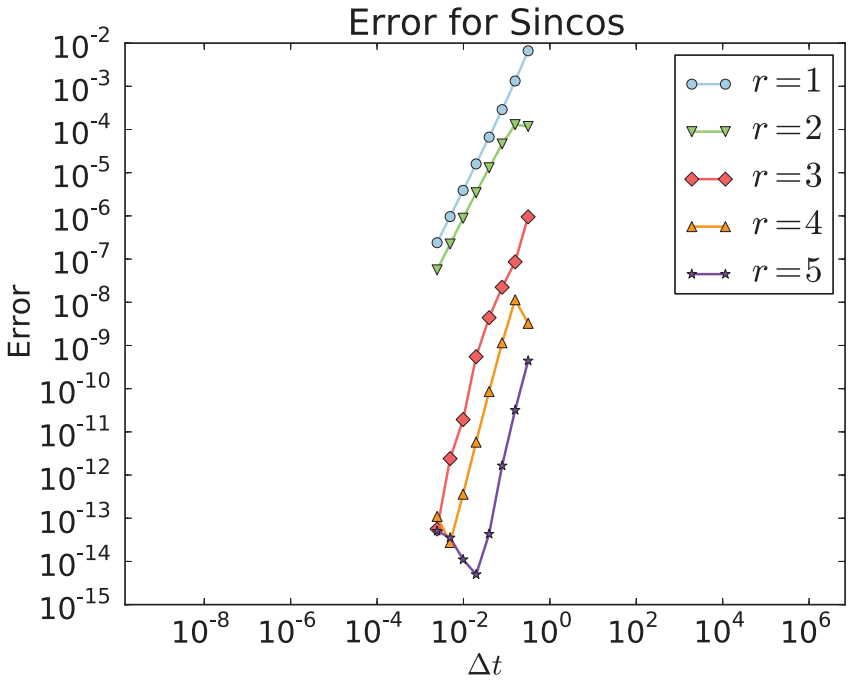



Figure 5.4: The error of the diamond scheme with varying $r$ applied to the Sincos problem (see Table 5.2). See Figure 5.3 for general comments regarding these error plots.

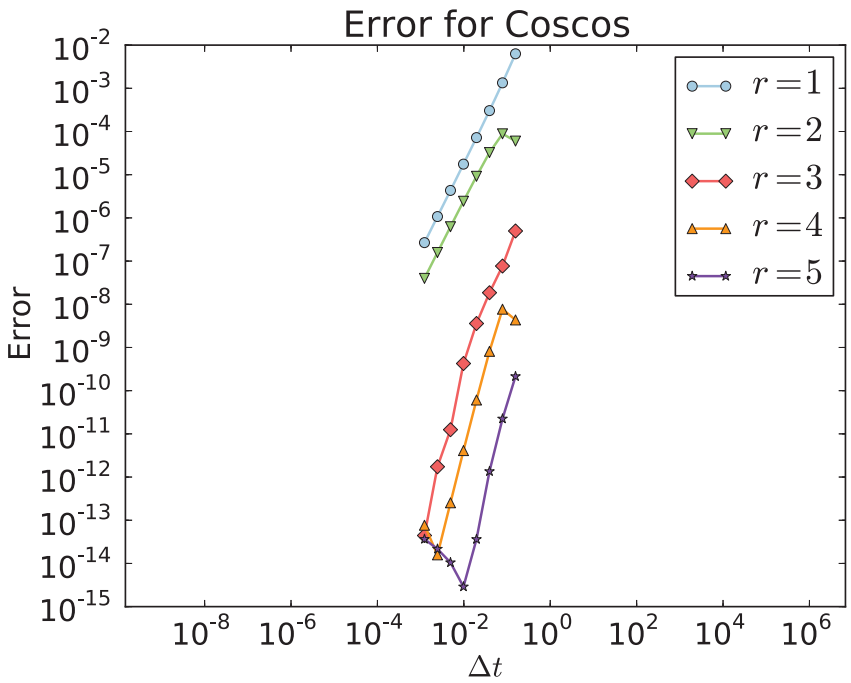



Figure 5.5: The error of the diamond scheme with varying $r$ applied to the Coscos problem (see Table 5.2). See Figure 5.3 for general comments regarding these error plots.

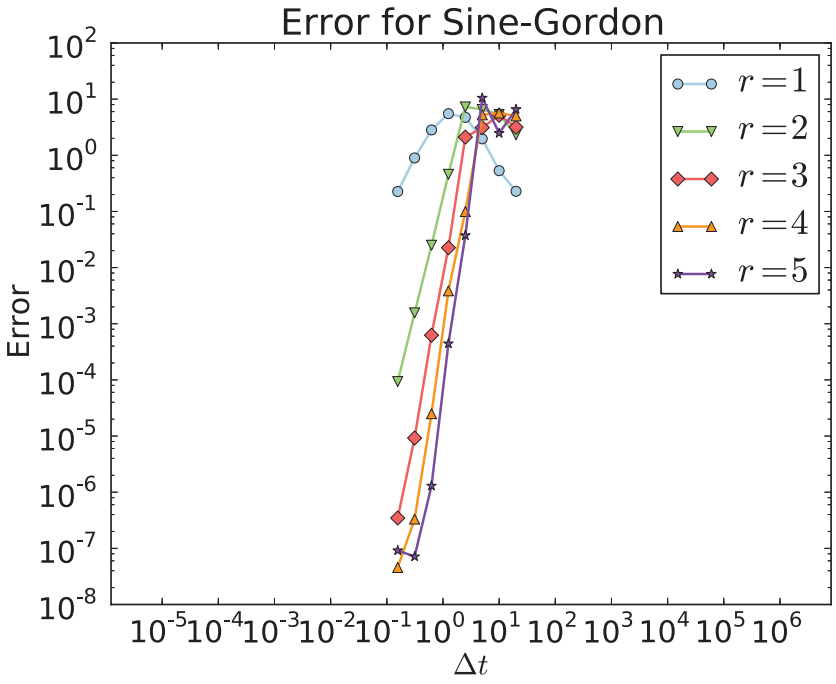



Figure 5.6: The error of the diamond scheme with varying $r$ applied to the Sine–Gordon problem (see Table 5.2). At the larger values of $\Delta x$ and $\Delta t$ the error is not behaving as expected. However this is because $\Delta x$ and $\Delta t$ have not got small enough to cope with the much larger (than the other three problems) spatial domain. See Figure 5.3 for general comments regarding these error plots.

| $r$ | Order | | | |
|---|---|---|---|---|
| | Esin | Sincos | Coscos | Sine–Gordon |
| 1 | 2.0 | 2.1 | 2.1 | 1.8 |
| 2 | 1.9 | 1.9 | 1.9 | 4.1 |
| 3 | 4.0 | 3.4 | 3.3 | 5.6 |
| 4 | 3.8 | 3.7 | 3.8 | 5.4 |
| 5 | 6.2 | 4.8 | 4.6 | 6.8 |

Table 5.3: Numerical order read from Figures 5.3–5.6 of the problems given in Table 5.2. To one significant figure the order appears to be $r$, although in many cases it exceeds this.

### 5.2.1 Schrödinger equation

It is interesting to apply the diamond scheme to a different multi-Hamiltonian. Consider the cubic nonlinear Schrödinger non-dimensionalized equation [25],

$$i\Psi_t(x,t) + \Psi_{xx}(x,t) = |\Psi(x,t)|^2 \Psi(x,t), \tag{5.4}$$

where $x \in [0,\pi]$ and $t > 0$. The boundary conditions will be periodic. The right hand side of this equation does not have a negative sign, so this is the so-called *defocussing* case. To express this in a multi-Hamiltonian form, let $\Psi = p + iq$ and consider the real and imaginary components

$$\begin{aligned} -q_t + p_{xx} &= (p^2+q^2)p \\ p_t + q_{xx} &= (p^2+q^2)q \end{aligned} \tag{5.5}$$

separately. Introduce $v = p_x, w = q_x$; then a multi-Hamiltonian form of Schrödinger equation is

$$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_t \\ q_t \\ v_t \\ w_t \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_x \\ q_x \\ v_x \\ w_x \end{pmatrix} = \begin{pmatrix} (p^2+q^2)p \\ (p^2+q^2)q \\ -v \\ -w \end{pmatrix} = \nabla S, \tag{5.6}$$

where $S = \frac{1}{4}(p^2+q^2)^2 - \frac{1}{2}(v^2+w^2)$. Note a practical advantage of the diamond scheme: the underlying equation is different (Schrödinger equation compared to the wave equation), but all that needs changing in the code is $K$, $L$, and $S$.

As in the wave equation numerical experiments (see Table 5.2) it is much easier to have an exact solution of the differential equation available. This exact solution can be used to provide initial conditions and be used as a reference solution for calculating errors. A solution to the cubic nonlinear Schrödinger non-dimensionalized equation

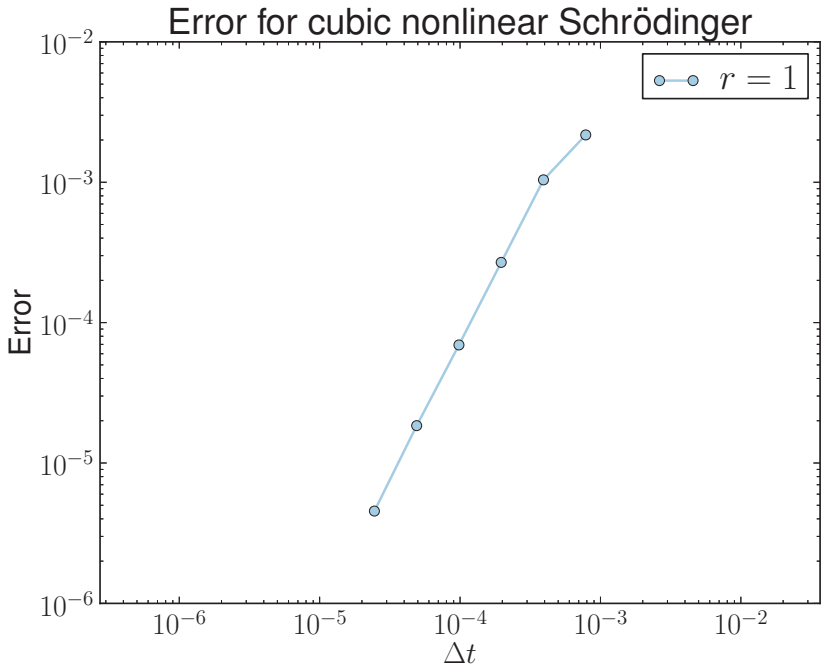



Figure 5.7: The error of the diamond scheme with $r = 1$ applied to the cubic nonlinear Schrödinger non-dimensionalized equation. $\frac{\Delta t}{\Delta x} = \frac{1}{2000}$ to keep $\Delta t < \frac{\Delta x^3}{24\sqrt{3}}$ over the range of $\Delta x$. The error appears to be approximately order 2.

from which the initial conditions were taken is

$$\Psi(x,t) = e^{i(2x-5t+3)},$$

where $x \in [0, \pi]$. Note that this is periodic, and it is straightforward to check that it is indeed a solution to (5.4).

The extended-to-the-corners (4.6)–(4.8), $r = 1$ diamond scheme is applied to this problem and the numerical error calculated as in (5.3). The diamond scheme with $r > 1$ was applied to this problem, but with the very small $\Delta t$ that is required, as the number of diamonds across the domain increases, the numerical solver finds it progressively more difficult to solve within each diamond. Since Theorem 3.3.6 only applies for the $r = 1$ scheme, only results for $r = 1$ will be shown. The number of diamonds at each time level was $N = 2, 4, 8, 16, 32, 64$, $\Delta x = \frac{\pi}{N}$, $\Delta t = \frac{\Delta x}{2000}$, and the integration time was calculated as twice the largest time step. Notice that $\Delta t$ remains less than $\frac{\Delta x^3}{24\sqrt{3}}$ over the entire range of $\Delta x$. The initialization scheme from Section 5.1 was used. The results for the global numerical error shown in Figure 5.7 indicate that the $r = 1$ scheme has order two.

The stability condition

$$\Delta t < \frac{\Delta x^3}{24\sqrt{3}}$$

is very stringent. To check that the scheme is indeed stable, the energy,

$$E = \int_0^\pi \frac{1}{4}(p^2+q^2)^2 + \frac{1}{2}(p_x^2+q_x^2)dx \tag{5.7}$$

is plotted for a long run to see if it drifts or stays bounded. Note that $E$ is a conserved quantity on solutions because

$$\begin{aligned}
E_t &= \int (p^2+q^2)(pp_t+qq_t) + (p_xp_{xt}+q_xq_{xt}) \\
&= \int (p^2+q^2)\left(-pq_{xx}+(q^2+p^2)qp+qp_{xx}-(q^2+p^2)qp\right) \\
&\quad + p_x\left(-q_{xx}+(q^2+p^2)q\right)_x + q_x\left(p_{xx}-(q^2+p^2)p\right)_x \\
&= \int [(p^2+q^2)p]_x q_x - [(q^2+p^2)q]_x p_x + p_x\left(-q_{xxx}+(2qq_x+2pp_x)q+(q^2+p^2)q_x\right) \\
&\quad + q_x\left(p_{xxx}-(2qq_x+2pp_x)p-(q^2+p^2)p_x\right) \\
&= \int [(p^2+q^2)p]_x q_x - [(q^2+p^2)q]_x p_x + (2qq_x+2pp_x)(qp_x-pq_x) \\
&= \int [(p^2+q^2)p]_x q_x - [(q^2+p^2)q]_x p_x + (q^2+p^2)_x(qp_x-pq_x) \\
&= \int (q^2+p^2)_x pq_x + (q^2+p^2)p_xq_x - (q^2+p^2)_x qp_x - (q^2+p^2)q_xp_x + (q^2+p^2)_x(qp_x-pq_x) \\
&= 0,
\end{aligned}$$

where (5.5) was used to get to line two, and integration by parts (noting periodic boundary conditions) was used in lines three and four.

Figure 5.8 shows this energy normalized by initial energy for a run of approximately 690,000 steps with $\Delta x = \frac{\pi}{8}$, and $\Delta t = \frac{1}{2}\frac{\Delta x^3}{24\sqrt{3}} \approx 0.0007$. The energy error is large, but $\Delta x$ is large, so this is not surprising. The important fact is the energy error stays bounded.

## 5.3 Dirichlet and Neumann boundary conditions

This section discusses development of boundary schemes. In general, for finite different schemes, dealing with boundaries is considered troublesome [23]. However, the diamond scheme does have the advantage of being spatially decoupled, which allows for the use of a different scheme on the boundary diamonds, without interfering with the internal diamonds. It is well known that, for a numerical scheme (including an accompanying boundary scheme) to have spatial order $n$, the interior numerical scheme must have

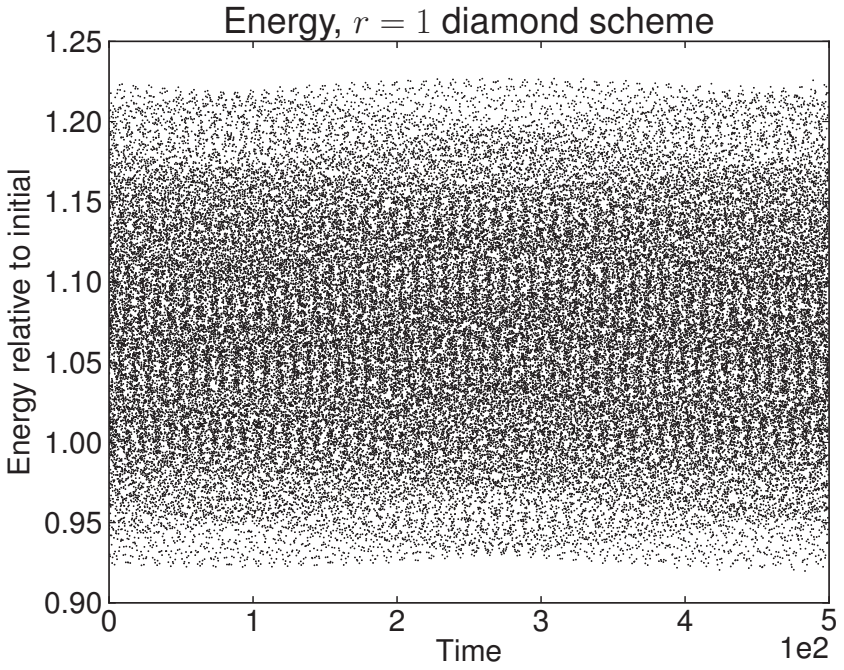



Figure 5.8: The energy from (5.7) for a cubic nonlinear Schrödinger non-dimensionalized $r = 1$ diamond scheme run. There are approximately 690,000 steps with $\Delta x = \frac{\pi}{8}$ and $\Delta t = \frac{1}{2}\frac{\Delta x^3}{24\sqrt{3}} \approx 0.0007$. The energy error is large, but $\Delta x$ is large, so this is not surprising. The important fact is the energy error stays bounded.

spatial order $n$, and the boundary scheme must have a spatial order of at least $n-1$ [23]. To deal with the boundary, some possible strategies are to define: a completely new scheme near the boundary, a new asymmetric scheme by modifying the interior scheme, or *phantom* nodes outside the domain. The constructed boundary scheme must be stable; however, proving that is no easy task: "...determination of the numerical stability of a fully discrete approximation (including boundary schemes) for a linear hyperbolic partial differential equation is a difficult task" [34].

The first possible strategy, of defining a completely new scheme near the boundary, was not seriously pursued in this research. A finite element scheme could perhaps have been employed. Another, possible approach would have been to use the approximate spatial finite different operators that are defined by the *summation by parts* and *simultaneous approximation term* methods [36, 37, 70, 75]. These finite different operators approximate the $x$-derivative at all points including boundary points. Various stable boundary treatments are given by Carpenter et al. [11, 12], Strand et al. [71], and Olsson et al. [56, 57]. However, the order of these schemes is fixed; but, the order of the diamond scheme varies with $r$. So this approach would have not been suitable, as the order of the boundary scheme should depend on $r$.

The second strategy, of some how modifying the interior scheme so it is defined on only half a diamond, was ultimately unsuccessful. More details on the failed attempts will be given after some notation has been defined. The third strategy, involving *phantom diamonds*, can successfully accommodate Dirichlet and Neumann boundary conditions for certain multi-Hamiltonian equations. This will now be discussed.

Along the $x = a$ and $x = b$ boundaries the diamonds are cut in half, so they are triangles. These triangles can be extended past the boundary to be phantom diamonds, half outside the domain and half inside. Recall that in the internal diamonds $z$ is known on the two bottom edges: the south-west (SW) and south-east (SE) edges. The unknown internal stage values $Z$, $X$, and $T$ are first solved for, then the update equations give $z$ on the two top edges. For concreteness, consider the left hand boundary from now on. On this phantom diamond the values of $z$ are not known on the SW edge, as this is outside the domain. Without using the boundary conditions there are simply too many unknowns to find the internal stages $Z$, $X$, and $T$, and perform the update. However, if the boundary conditions can give some of the internal stage values, then it is possible that all the internal stages and the previously unknown values of $z$ on the SW edge might be found, and the update proceed. What is necessary is that the boundary conditions give the correct number of pieces of information (whether this is *sufficient* for a stable, accurate boundary treatment is another question). For the $r$ method there are $nr$ pieces of missing information, where $n$ is the dimension of the problem.

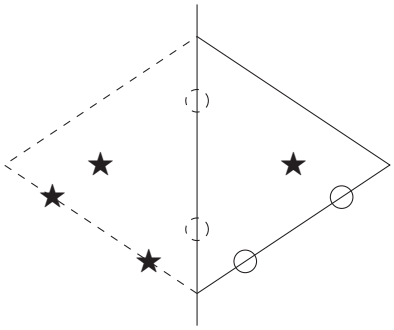


Figure 5.9: A left-hand boundary phantom diamond in the $r = 2$ scheme. The solution, $z$, is known at the circles on the SE edge. For an internal diamond, $z$ would also be known at the stars on the SW edge, but not in this case. Internally, nothing is known at the stars, and some information is known at the dashed circles. There needs to be enough known at these points marked with dashed circles to match the missing information on the SW stars.

Before preceding with the successful method, an exploration of the problem and a few failed alternatives will be given. At any time, $t$, let the boundary condition specify $k$ pieces of information. For the wave equation, $n = 3$, and if $u(a, t) = g(t)$ was given, then $k = 1$. If $k = n$ then $n$ extra pieces of information are known at any point on the boundary, by picking the usual $r$ points (dashed circles in Figure 5.9), $nr$ pieces of information are known which makes up for not knowing the $nr$ pieces of information on the SW edge. The counting in the system of equations (3.16)–(3.18) is correct and the system might be solvable. However, in general $k < n$. Some possible approaches are now given.

- Since $k$ pieces of information are known for *all* $t$, extra points on the boundary can be picked. Pick as many points as required to get the counting in equations (3.16)–(3.18) correct. However, a problem immediately arises. The diamond scheme is defined on a regular (albeit rotated) grid in each diamond, it is unclear how these extra points fit into (3.16)–(3.18). Extra equations may be defined relating various components of $Z$, $X$, and $T$ at these new locations. This approach was developed, but it was found that often the system was unsolvable or the scheme unstable.

- Use the extra $k$ pieces of information at the $r$ points on the boundary in a 2-dimensional least-squares polynomial fit to extrapolate these $k$ pieces of information to somewhere else. The somewhere else could be the diamond interior, the SW boundary, or directly to the NE boundary. Unfortunately, the resulting equations were sometimes unsolvable or the scheme unstable.

- Use information contained in the diamond located to the SE of the current diamond. The information was used in the least-squares fit in the previous idea. The scheme was often unstable due to using too much interior information.

- Nonlinearly map the triangle (boundary, NE boundary and SE boundary) to a square and perform the solve and update on this square. The $2r$ boundary points are located in such a way that after the nonlinear map, $r$ are located correctly on the left edge and $r$ on the *top* edge. This results in knowing $rk$ pieces of information on each of the left and *top* edges of the resulting square. This scheme was often unstable, perhaps because of information flowing backwards in time or the nonlinearity of the transformation.

Combinations of the above ideas were also tried; unfortunately whilst sometimes a method appeared to work on a particular problem it failed on another, or the scheme was so convoluted that it was unclear how it could be generalized to a different $k$ or $n$. A considerable portion (more than represented by the length of this section) of research time and energy was expended on this problem; the reason being, on the surface it appears that the diamond scheme should easily lend itself to flexible boundary treatment. However it was not to be; a more restricted approach had to be taken. What follows is the boundary treatment for the wave equation with Dirichlet and Neumann boundary conditions.

Consider the multi-Hamiltonian arising from the wave equation (1.5) with a Dirichlet boundary condition on the left, $u(a,t) = g(t)$ with a smooth $g(t)$. The right hand boundary treatment is similar and will not be detailed. A phantom diamond is illustrated in Figure 5.9 for the case $r = 2$. On the SW edge there are $nr$ ($3r$ for this wave equation example), missing pieces of information. As in the internal case, the values of $z$ are known on the SE edge. For the counting in the system of equations (3.16)–(3.18) to be correct, an extra $3r$ pieces of information are needed. There are $r$ internal stage value points along the boundary, if 3 pieces of information could be found at each of these $r$ points, equations (3.16)–(3.18) would have the correct number of unknowns. Now the value of $u$ (first component of $Z$) is known along the boundary; differentiating $g(t)$ with respect to time gives $v(a,t) = g'(t)$ (second component of $Z$); and using the PDE, $u_{tt} - u_{xx} = f'(u)$, gives $w_x(a,t) = g''(t) - f'(g(t))$ (third component of $X$). That is 3 extra pieces of information.

Neumann boundary conditions can be treated in a similar manner. All that is required is the correct ($3r$ for the wave equation) number of pieces of information. Suppose that $u_x(a,t) = g(t)$, in other words $w$, or the third component of $Z$, is known on the boundary. Differentiating with respect to time gives $w_t$, or the third component of $T$. By equality of mixed partial derivatives $v_x = w_t$, so the second component of $X$ is also known on the boundary. This gives the required number of extra pieces of

information to make up for the missing values of $z$ on the SW edge.

Although the above has been applied to the wave equation it can be extended to any multi-Hamiltonian, so long as the counting is correct. For example Schrödinger equation in Section 5.2.1 has $n = 4$, so four pieces of extra information are needed on the boundary. If Dirichlet boundary conditions were applied, then $p$ and $q$ would be specified, but differentiating with respect to time gives two more conditions, thus giving the four pieces of extra information needed.

The above ideas on how to treat boundary conditions can be applied to initializing the diamond scheme. Here a phantom diamond is constructed about the $t = 0$ axis. Now both the SW and SE edges are missing values of $z$, so more information ($2nr$ to be exact) must be garnered from the initial, or $t = 0$ boundary, conditions. This situation is illustrated in Figure 5.10. For the wave equation example $n = 3$, so 6

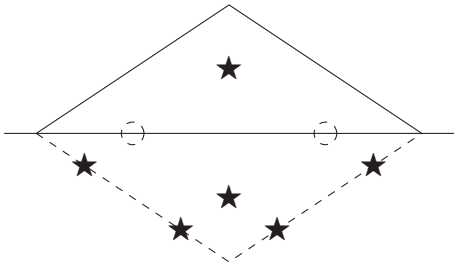


Figure 5.10: An initial phantom diamond in the $r = 2$ scheme. The solution, $z$, is not known on the SW or SE edges. Internally, nothing is known at the stars, and some information is known at the dashed circles. There needs to be enough known at these points marked with dashed circles to match the missing information on the SW and SE stars.

pieces of information are required at each internal stage values along the $t = 0$ axis. The solution $u$ and $v = u_t$ are normally specified. Differentiating along the axis gives $w = u_x$ and $v_x = u_{tx}$. Because $u_{tx} = u_{xt}$, $w_t$ is also known. The sixth, and last, piece can be obtained by differentiating $w$ by $x$ to give $w_x = u_{xx}$. So, now there are four ways to initialize the diamond scheme: forward Euler (not accurate for high $r$), the exact solution (may be impractical), the so-called diamond method in Section 5.1, and the boundary method. The boundary method only works for certain multi-Hamiltonian PDEs where the correct number of pieces of information are available.

The developed boundary scheme is now applied to the four sample problems given in Table 5.2 with a mix of Dirichlet and Neumann boundary conditions. Table 5.4 summarizes the problems. While attempting to find a good boundary scheme, two problems repeatedly arose: the scheme in question turned out to be unstable, and/or equations (3.16)– (3.18) were not solvable. Detecting the second problem was fairly easy: the numerical solvers (Section 4.2) would report an error if they failed to converge. If the first problem occured, it normally occurred quickly and dramatically: the

| Name | Range | Left boundary | Right boundary |
|---|---|---|---|
| Esin D-D | $0.2 \leq x \leq \frac{\pi}{3}$ | Dirichlet | Dirichlet |
| Sincos D-D | $0.2 \leq x \leq \frac{\pi}{3}$ | Dirichlet | Dirichlet |
| Sincos D-N | $0.2 \leq x \leq \frac{\pi}{3}$ | Dirichlet | Neumann |
| Coscos D-D | $0.2 \leq x \leq \frac{\pi}{3}$ | Dirichlet | Dirichlet |
| Coscos D-N | $0.2 \leq x \leq \frac{\pi}{3}$ | Dirichlet | Neumann |
| Sine–Gordon D-D | $-2 \leq x \leq 2$ | Dirichlet | Dirichlet |

Table 5.4: Sample non-periodic problems. See Table 5.2 for the exact equations and solutions. The Dirichlet boundary conditions are found using the exact solution, and the Neumann conditions by differentiating the exact solution with respect to $x$. Note there are no reasons, other than lack of resources/space/interest, for missing out the results for Esin D-N and Sine–Gordon D-N.

solution would rapidly grow (and then the second problem was often triggered to boot). Extraordinarily long runs, consisting of many millions of steps, were performed on the boundary schemes presented here to numerically confirm the schemes are stable and lead to solvable equations. However it still remains to prove this. One possible avenue of attack would be to establish a local discrete energy conservation law for the linear wave equation. The Reich method that is used within each diamond is known to have quadratic invariants [60, 73].

Because the exact solution is known for all the sample problems it is easy to impose whatever boundary condition are desired on any spatial domain. The domains were chosen so the solutions were not periodic or symmetric in any way, because while testing other potential methods it became apparent that using 'easy' problems gave false confidence in the method. To ensure only one thing was tested at a time the initialization scheme used was the diamond method. As a comparison, for one problem (Coscos D-N), the boundary initialization was tried. Because the domains are smaller than the periodic tests, a smaller number of diamonds was used, $N = 2, 4, \ldots, 128$. Figures 5.11–5.16 show the error for the various test problems as $\Delta t$ is decreased. The estimated numerical error, which is read off from these figures, is summarized in Table 5.5. The boundary method does not appear to adversely affect the order of the scheme: it is at least $r$. Figure 5.17 shows the error for the Coscos D-N problem initialized with the usual diamond method, and the boundary method. It is apparent that the boundary initialization method does as well, or better, than the diamond initialization. Unfortunately the boundary initialization method can only be applied to certain problems (such as the multi-Hamiltonian derived from the wave equation) where the counting is correct.

This chapter presented a successful initialization scheme, and an approach for dealing with non-periodic boundary conditions. Results from numerical tests were presented. Full conclusions are in the following chapter.

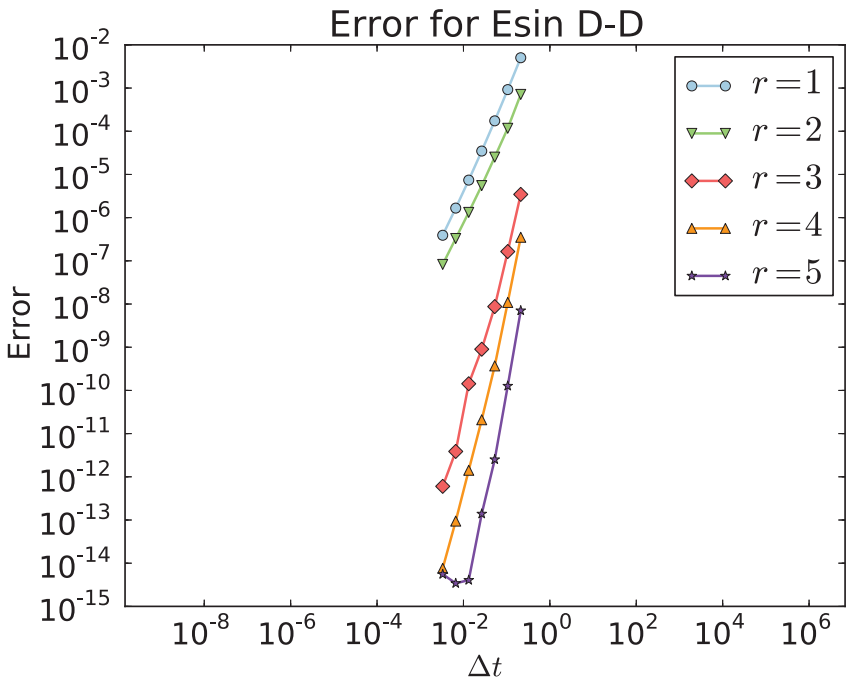



Figure 5.11: The error of the diamond scheme with varying $r$ applied to the Esin D-D problem (see Table 5.4). The Courant number is fixed at $\frac{1}{2}$ as $\Delta t$ is decreased. Table 5.5 summarizes the numerical order by reporting the slope of these lines. The numerical method cannot reduce the error any further than machine precision ($\approx 10^{-16}$ for double floating point arithmetic). When the error has reached approximately machine precision, increasing $r$ and/or decreasing $\Delta t$, just causes 'noise', and these parts of the error plot can be safely ignored.

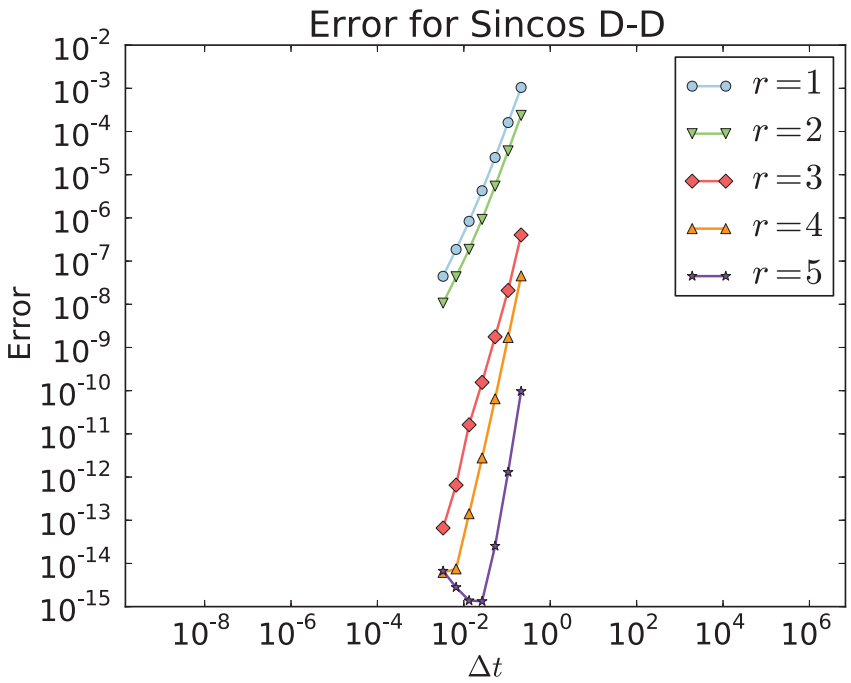



Figure 5.12: The error of the diamond scheme with varying $r$ applied to the Sincos D-D problem (see Table 5.4). See Figure 5.11 for general comments regarding these error plots.

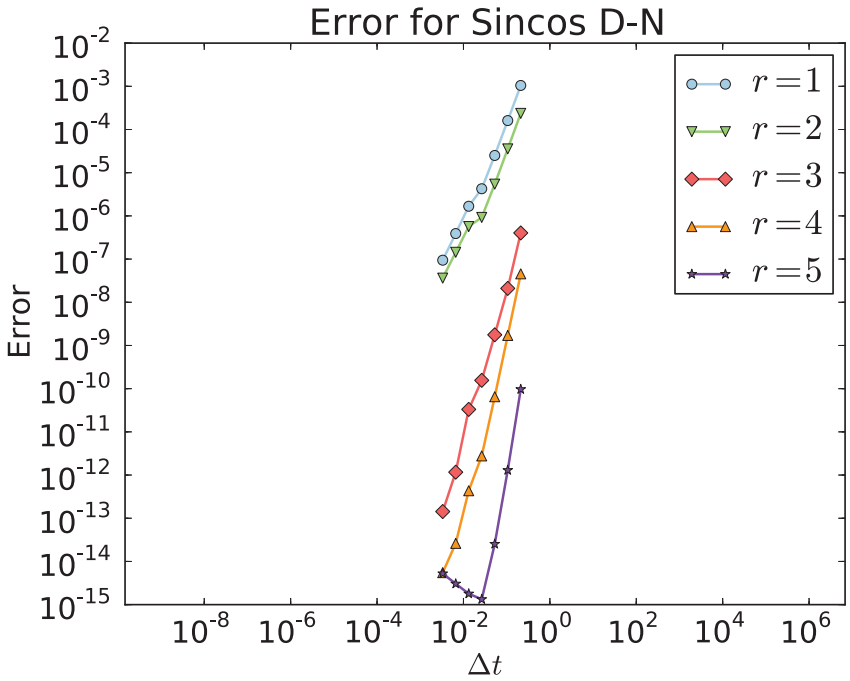



Figure 5.13: The error of the diamond scheme with varying $r$ applied to the Sincos D-N problem (see Table 5.4). See Figure 5.11 for general comments regarding these error plots.

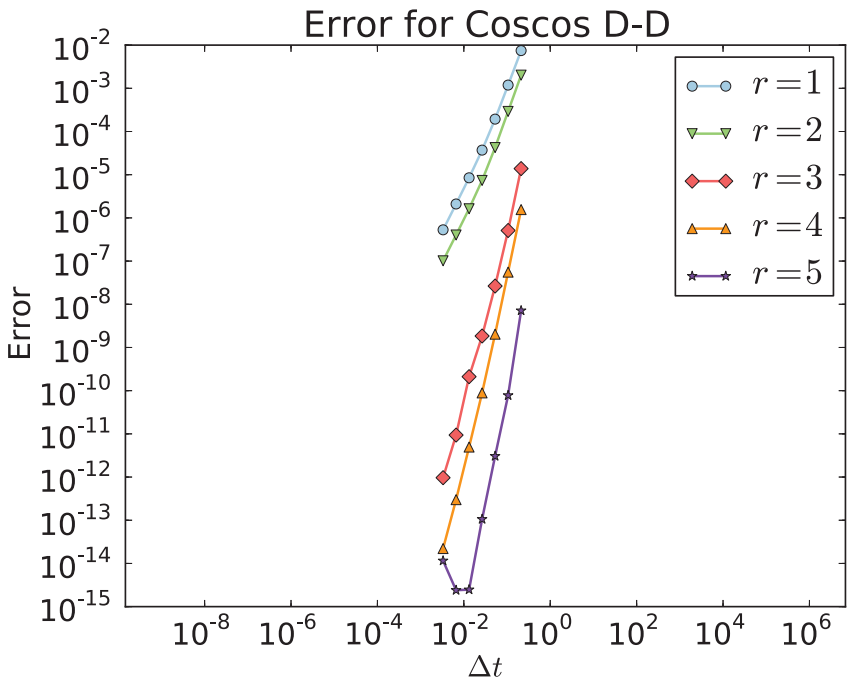



Figure 5.14: The error of the diamond scheme with varying $r$ applied to the Coscos D-D problem (see Table 5.4). See Figure 5.11 for general comments regarding these error plots.

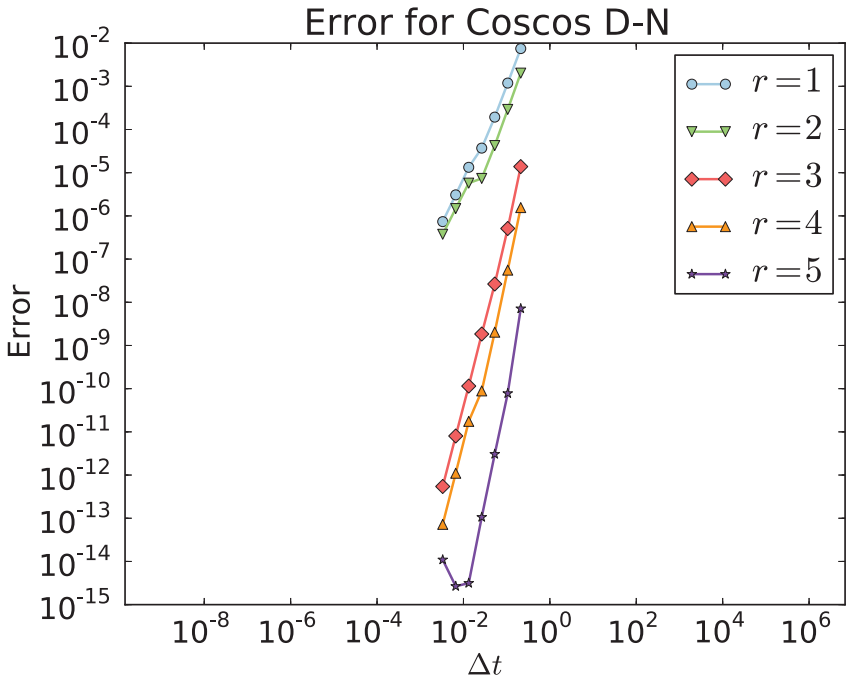



Figure 5.15: The error of the diamond scheme with varying $r$ applied to the Coscos D-N problem (see Table 5.4). See Figure 5.11 for general comments regarding these error plots.


Figure 5.16: The error of the diamond scheme with varying $r$ applied to the Sine–Gordon D-D problem (see Table 5.4). See Figure 5.11 for general comments regarding these error plots.

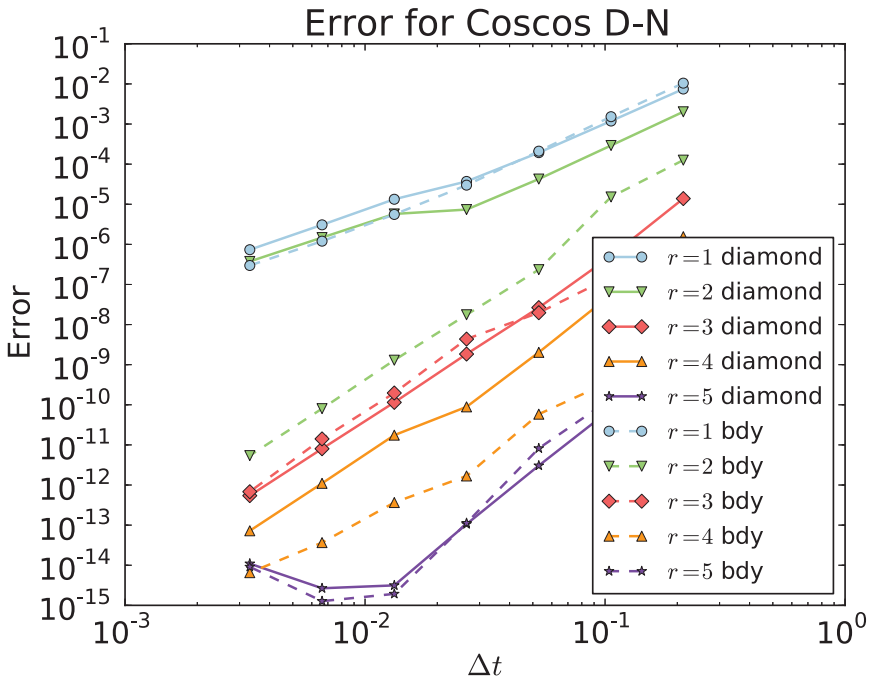



Figure 5.17: The error of the diamond scheme initialized with the diamond method (Section 5.1) and the boundary initialization method (Section 5.3), with varying $r$ applied to the Coscos D-N problem (see Table 5.4). The Courant number is fixed at $\frac{1}{2}$ as $\Delta t$ is decreased. The boundary initialization method is as good as, or better, than the diamond initialization method.

| $r$ | Order | | | | | |
|---|---|---|---|---|---|---|
| | Esin DD | Sincos DD | Sincos DN | Coscos DD | Coscos DN | Sine–Gordon DD |
| 1 | 2.3 | 2.4 | 2.2 | 2.3 | 2.2 | 2.0 |
| 2 | 2.2 | 2.4 | 2.1 | 2.4 | 2.1 | 3.2 |
| 3 | 3.7 | 3.8 | 3.6 | 4.0 | 4.1 | 4.4 |
| 4 | 4.2 | 4.5 | 3.8 | 4.3 | 4.1 | 4.5 |
| 5 | 5.2 | 5.4 | 5.4 | 5.4 | 5.3 | 6.3 |

Table 5.5: Numerical order read from Figures 5.11–5.16 for the problems given in Table 5.4. To one significant figure the order appears to be $r$, although in some cases it exceeds this.

# Chapter 6

# Conclusions

Two classes of differential equations:

$$J\dot{z} = \nabla H$$

and

$$Jz_t + Lz_x = \nabla H,$$

have been discussed in this thesis. They are linked, in fact: the first is a special case of the second, and both are generalizations of classical Hamiltonian systems. This thesis has made contributions to each of them.

## 6.1 Constrained Hamiltonian systems

Three different types of constrained dynamical system were discussed in this thesis:

- Those with only *holonomic*, in other words positional, constraints. Systems with these constraints were introduced in Section 1.3. Well-known methods for solving these systems were discussed in Section 1.3.1, and these types of constrained dynamical system were not discussed further.
- The *dynamic nonholonomic equations*, also known as the Lagrange–d'Alembert equations, that describe many mechanical systems in rolling and sliding contact. These systems have *nonholonomic* constraints, constraints that depend on position and velocity, but not on the derivative of a holonomic constraint. As these equations are not in general Hamiltonian or variational they were not discussed any further.
- The *variational nonholonomic equations*, also known as the vakonomic equations. These systems have nonholonomic constraints, and occur in the *Lagrange problem* with nonholonomic constraints $g(q, \dot{q}) = 0$, and in optimal control problems.

> Control problems are important and widely found: kinematic sub-Riemannian optimal control problems; control on semi-simple Lie groups and symmetric spaces; motion of a particle in a magnetic field; optimal control on Riemannian manifolds and Lie groups; parking a 'car' (an example modelled numerically in Section 2.5); riding a bike; rolling a ball; controlling a satellite or a falling cat controlling itself. These are all described using the variational nonholonomic equations.

It was this third type of constrained dynamical system—the variational nonholonomic equations—that was the focus of Chapter 2.

Proposition 2.1.1 shows the equivalence between the variational equations with constraints $g_i(q) \cdot \dot{q} = 0$ to a generalized ($J$ not necessarily invertible) Hamiltonian system with index 1 constraints. Although Proposition 2.1.1 was not original, it should be emphasised that the usual treatment is to go one step further and eliminate the Lagrange multipliers $\lambda$ to get a canonical Hamiltonian system in $(q, p)$. This step may not be desirable either analytically or numerically. Under certain invertibility conditions this proposition was generalized to Proposition 2.1.2 that has constraints of the form $g_i(q, \dot{q}) = 0$. Further generalizations may be attempted: the Lagrangian could be generalized to any singular Lagrangian $L(q, \dot{q}, \lambda)$, and still further to Lagrangians $L(z, \dot{z})$ where $|L_{\dot{z}\dot{z}}| = 0$, but the required nondegeneracy assumptions may not be as geometrically transparent as those in Proposition 2.1.2.

It was shown in Proposition 2.4.1 that if *holonomic* constraints were added to the original variational problem in Proposition 2.1.1, the resulting Hamiltonian system is a simple *holonomically* constrained system. This system can be solved by a symplectic method such as RATTLE [47, 68].

Chapter 2 went further to discuss what symplectic methods could be applied to generalized Hamiltonian systems that may not even arise from constrained Hamiltonian systems. First, Lemma 2.2.1 showed that the flows of these generalized Hamiltonian systems are in fact symplectic. It was also shown that in the particular case of Proposition 2.1.1, the generalized Hamiltonian system that is obtained is equivalent to a canonical Hamiltonian system obtained by eliminating the Lagrange multipliers $\lambda$. Proposition 2.2.2 showed that any symplectic Runge–Kutta method (in other words a Runge–Kutta method that when applied to a canonical Hamiltonian system is symplectic) applied to these generalized Hamiltonian systems is still symplectic. A simplified proof of this result was shown for the implicit midpoint rule. Proposition 2.3.1 showed that a symplectic Runge–Kutta method applied to a generalized Hamiltonian with index 1 constraints is well-defined and convergent with the same order of the Runge–Kutta method.

Two sample applications were given. The sub-Riemannian geodesics of a idealized wheeled vehicle (the 'parallel parking' problem) were found in Section 2.5, and the

sub-Riemannian geodesics of the Heisenberg group were found in Section 2.6.

**Future work**

Many directions for future research are suggested by the results and the approach taken.

- In this thesis, attention has been restricted to symplectic Runge–Kutta methods; a generalization to partitioned methods in which different Runge–Kutta coefficients are used for $q$, $p$, and $\lambda$ could be attempted.

- The nondegeneracy conditions in Propositions 2.1.1, 2.1.2, and 2.4.1 are essential for the integrators in Proposition 2.3.1, indeed, for the entire approach, to work. It is not clear to what extent the approach can be extended to handle more general constraints, for example, to the system $J\dot{z} = \nabla H + \lambda \nabla g$, where the constraint submanifold $g(z) = 0$ is symplectic. No symplectic, constraint-preserving method is known for this problem.

- As remarked before Proposition 2.3.1, a full study of the geometry of the relations $(z_0, z_1)$ generated in Proposition 2.2.2 remains to be undertaken. Any solutions are symplectic, so this gives access to a much larger class of symplectic maps than do traditional generating functions. Note that new variables (analogous to $\lambda$) can be added as needed to generate larger classes of maps.

## 6.2 The diamond scheme

Section 3.1 defined the simple diamond scheme, showed it has order two in both space and time, has solvable equations for the one-dimensional wave equation, and, in Proposition 3.1.1, satisfies a discrete conservation law. The simple diamond scheme was tested numerically on the Sine–Gordon equation, and compared favourably to the leapfrog scheme (Figure 3.3). The scheme was extended to allow Dirichlet and Neumann boundary conditions for the wave equation.

The diamond scheme, defined in Section 3.2, is a process that uses the multisymplectic Runge–Kutta collocation method given by Reich [60]. The domain is overlaid by a mesh of diamonds, and within each diamond the Reich scheme is used. The unique quality of the diamond scheme lies in the arrangement of these diamonds, and how the information flows from a set of diamonds at one time level to the next.

Theorem 3.2.3 shows how the simple diamond and $r = 1$ diamond schemes are related. In particular, it shows how any solution of the simple diamond scheme, mapped to mid-points of the diamond edges, satisfies the $r = 1$ diamond scheme. This result is repeatedly used to show, or explain, why results applicable to the simple diamond also apply to the $r = 1$ diamond scheme. Theorem 3.2.5 directly proves that the order

of the $r = 1$ diamond scheme is two in space and time; which, given the relationship with the simple diamond scheme, is not surprising as the simple diamond scheme is also order two.

Many features of the diamond scheme can be seen immediately from its definition. Unlike many other schemes (see the beginning of Chapter 3) that are only defined for particular instances of the multi-Hamiltonian, the diamond scheme is defined for *all* multi-Hamiltonian systems (3.1).

The diamond scheme is only locally implicit within each diamond. Such locality is suitable for hyperbolic systems with finite wave speeds. Compared to fully implicit schemes like Runge–Kutta box schemes, this leads to the following:

- The nonlinear equations are solvable. This was proved in Theorem 3.2.1 for the one-dimensional wave equation.
- Faster nonlinear solves, since an entire time level of boxes or diamonds is not coupled together in one large implicit equation.
- Better parallelization, as no communication between diamonds is required during solves, and all diamonds can be solved in parallel. Experiments in Section 4.3 indicate that the scheme scales exceedingly well until each processor has approximately three diamonds to solve at each time step.
- Local treatment of boundary conditions. Schemes for the multi-Hamiltonian arising from the wave equation with Dirichlet and Neumann boundary conditions can been found in Section 5.3.

On the other hand, the implicitness within a diamond should improve stability compared to fully explicit methods in cases where $S(z)$ contributes (a moderate amount of) stiffness to the equation. Despite this, as shown in Theorem 3.3.6, for the multi-Hamiltonian arising from the linear Schrödinger equation, the stability conditions on $\Delta t$ turn out to be very stringent.

The diamond scheme is linear in $z$. This is expected to lead to the following:

- Preservation of conservation laws associated with linear symmetries. Theorem 3.2.2 presents the discrete symplectic conservation law that the diamond scheme satisfies.
- Possibly better transmission of waves at mesh boundaries, as has been found for the (linear) simple box scheme [20].
- Easier handing of dispersion relations, which can be determined once and for all, for *all* multi-Hamiltonian PDEs. Theorem 3.3.2 finds the discrete dispersion relation for the simple diamond and $r = 1$ diamond schemes in terms of the

continuous dispersion relation. The coordinate transformation ($h$ given in (3.31)) between the discrete and continuous dispersion relations is unfortunately not a continuous, easy to visualize, change of coordinates. This means that stability results cannot just be 'read off' from the continuous dispersion relation. However, the stability conditions for the simple diamond and $r = 1$ schemes were found in Theorem 3.3.5.

Numerical results are presented in Section 5.2. The diamond scheme is applied to five different problems: four problems are wave equations with different forcing terms and/or initial conditions, and the fifth problem is the cubic nonlinear Schrödinger non-dimensionalized equation. The order of the diamond scheme is numerically shown to be order $r$, although for odd $r$ the order is often $r + 1$.

The diamond mesh does introduce some complications or problems. The implementation is slightly more involved than on a standard mesh; however, with carefully chosen data structures and a modern programming language, such as Python, this has not been found to be significant. In fact, due to the decoupled (between diamonds) nature of the algorithm, the parallel implementation is generally easier than on a standard mesh. The other two complications, boundaries and the multi-step nature of the scheme, are discussed in the next two paragraphs.

The interaction of the mesh with the boundaries (both 'vertical' and 'horizontal') means that they need special treatment. A bootstrap method is needed to get the initial conditions 'up onto the first zigzag' (see Figure 3.5). For much of the development of the diamond scheme, that bootstrap method consisted of 'cheating' by using the known exact solution. Section 5.1 developed a robust initialization scheme that involves mapping the initial triangles to diamonds, and then using the diamond scheme itself on these diamonds. Table 5.1 shows that the order of the diamond scheme initialized with this method, is as good as, or often better, than the diamond scheme initialized with the known exact solution. It is more difficult to develop a general purpose method for handling arbitrary ('vertical') boundary conditions. Section 5.3 details a successful method for handling Dirichlet and Neumann boundary conditions for the one-dimensional wave equation. The method hinges upon garnering enough information on the boundary to account for the lack of information on the outside edge of the boundary diamond. Various schemes to obtain this information through extrapolation or interpolation were unsuccessful. Nonlinear mappings of the edge triangle to a diamond (as in the initialization scheme) were also unsuccessful, most likely due to not having the right amount of information along the boundary. The successful (but not general) method resorts to acquiring the required information from the PDE. Perhaps this is not surprising: PDE boundary treatment is often difficult, and the correct number of boundary conditions for existence and uniqueness of the solution is inherently tied to the PDE.

The final, and perhaps most troubling, complication that the diamond mesh causes is the multi-step nature of the scheme. This fact was not appreciated during most of the research. It was not until the diamond scheme was applied to the multi-Hamiltonian arising from Schrödinger equation that spurious modes were noticed (they were unstable modes—the only reason they were noticed!). Spurious modes are often the hallmark of a multi-step scheme [3]. A single-step ODE scheme maps a single point to a single point, a multi-step scheme maps a set of $r$ points to $r$ points. The extra $(r-1)$ points allow for the possibility of extra or spurious modes to enter. The diamond scheme uses the Reich scheme within each diamond, and although the PDE has been transformed into tilde coordinates, it is still a multi-Hamiltonian PDE. So the diamond scheme is almost the Reich scheme with a different multi-Hamiltonian. The difference is, in the Reich scheme, information is known only on the bottom edge; in the diamond scheme information is known on *two* edges. This is perhaps the key feature of the diamond scheme: this extra information allows one to solve each diamond independently, but it also allows for spurious modes to enter in. To see the multi-step nature another way, consider applying the diamond scheme to an ODE by taking the limit as $\Delta x \to 0$. The diamonds get squashed onto a vertical line with information known at $r$ points in the lower half of the each time step, and $r$ points in the top half. This is exactly an ($r$-step) ODE solver. If $\Delta x \to 0$ in the Reich scheme, all the points at one time level collapse to a single point: a single-step ODE scheme. A multi-step scheme is still a useful scheme: many of the widely-used ODE solvers mentioned in the Introduction are multi-step. A multi-step scheme does not necessarily lead to instability, but it does allow for the possibility of spurious modes, or wiggles, in the solution.

**Future work**

- The principle of the diamond method is extremely general and can be applied to a very wide range of PDEs; it may have applications beyond the multisymplectic PDE (3.1). It extends easily to $2d$-hedral meshes for PDEs in $d$-dimensional space-time, again subject to the CFL condition.

- At present, to prove existence of solutions of the nonlinear equations the class of equations must be restricted; ideally one would like to establish existence of numerical solutions for all PDEs (3.1) and relate them to the existence of solutions to the PDE itself.

- More work could be expended on developing boundary schemes for different PDEs, or even a general boundary treatment for the multi-Hamiltonian system.

- Prove that the boundary schemes developed in Section 5.3 are stable and lead to solvable equations.

- The order of the scheme has only been demonstrated numerically, a theoretical proof of the order needs to be established.

- Finally, is there a general purpose multisymplectic scheme that is not multi-step in nature? Perhaps a scheme could be developed for a slightly less general PDE than the multi-Hamiltonian PDE (3.1): one where $K$ and $L$ are fixed, but the right hand side is allowed to vary. Or, the scheme could be modified so at the end of each time step certain information is thrown away, so as to create a single-step scheme.

## 6.3 Summary

Many of the aims and objectives that were stated at the end of Chapter 1 have been met. In particular, symplectic Runge–Kutta methods have been shown to be effective symplectic integrators for the vakonomic equations and generalized Hamiltonian systems. Because a well-established method, such as symplectic Runge–Kutta, was employed there was no need to present stability or order proofs. The main difficulty was proving that symplectic Runge–Kutta could be used on the generalized Hamiltonian systems, and the variational equations with various constraints could be represented as generalized Hamiltonian systems.

Secondly, a multisymplectic integrator for the multi-Hamiltonian equations was developed. Both the simple diamond and diamond schemes were shown to lead to solvable equations for the wave equation (this result has not been generalized to the multi-Hamiltonian equations). The order was found for the simple diamond and $r = 1$ diamond schemes, and numerical results presented for the remaining ($r > 1$) diamond schemes. The discrete dispersion relation for the simple diamond and $r = 1$ diamond schemes was found, and stability conditions presented for the wave equation and Schrödinger equation. The discrete dispersion relation for the $r > 1$ diamond scheme has not been found. Approaches for dealing with Dirichlet and Neumann boundary conditions for the wave equation were presented. Finally, the diamond scheme was shown to be particularly amenable to parallel implementation.

# Bibliography


[1] G. M. Amdahl, *Validity of the single-processor approach to achieving large scale computing capabilities*, AFIPS Conference Proceedings, 30 (1967), pp. 493–485.

[2] H. C. Andersen, *Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations*, Journal of Computational Physics, 52 (1983), pp. 24–34.

[3] U. Ascher, *Numerical Methods for Evolutionary Differential Equations*, Computational Science and Engineering, Cambridge University Press, 2008.

[4] U. Ascher and R. McLachlan, *On symplectic and multisymplectic schemes for the KdV equation*, J. Sci. Comput., 25 (2005), pp. 83–104.

[5] U. M. Ascher and R. I. McLachlan, *Multisymplectic box schemes and the Korteweg–de Vries equation*, Appl. Numer. Math., 48 (2004), pp. 255–269.

[6] R. Benito and D. M. de Diego, *Discrete vakonomic mechanics*, Journal of Mathematical Physics, 46 (2005). eid. 083521.

[7] A. M. Bloch, *Nonholonomic mechanics and control*, vol. 24 of Interdisciplinary Applied Mathematics, Springer-Verlag, New York, 2003.

[8] T. J. Bridges and S. Reich, *Multi-symplectic integrators: numerical schemes for Hamiltonian PDEs that conserve symplecticity*, Phys. Lett. A, 284 (2001), pp. 184–193.

[9] T. J. Bridges and S. Reich, *Numerical methods for Hamiltonian PDEs*, J. Phys. A, 39 (2006), pp. 5287–5320.

[10] K. Burrage and J. C. Butcher, *Stability criteria for implicit Runge–Kutta methods*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 46–57.

[11] M. Carpenter, D. Gottlieb, and S. Abarbanel, *Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes*, Journal of Computational Physics, 111 (1994), pp. 220–236.

[12] M. H. Carpenter, D. Gottlieb, and S. Abarbanel, *Stable and accurate boundary treatments for compact, high-order finite-difference schemes*, Applied Numerical Mathematics, 12 (1993), pp. 55–87.

[13] E. Celledoni, V. Grimm, R. I. McLachlan, D. McLaren, D. O'Neale, B. Owren, and G. Quispel, *Preserving energy resp. dissipation in numerical PDEs using the "average vector field" method*, J. Comput. Phys., 231 (2012), pp. 6770–6789.

[14] J.-B. Chen, M.-Z. Qin, and Y.-F. Tang, *Symplectic and multi-symplectic methods for the nonlinear Schrödinger equation*, Comp. Math. Appl., 43 (2014), pp. 339–370.

[15] G. Cohen, *Higher-Order Numerical Methods for Transient Wave Equations*, Springer, New York, 2002.

[16] J. Cortés Monforte, *Geometric, control and numerical aspects of nonholonomic systems*, vol. 1793 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 2002.

[17] K. Feng and M. Qin, *Symplectic Geometric Algorithms for Hamiltonian Systems*, Springer, 2010.

[18] O. E. Fernandez, A. M. Bloch, and P. J. Olver, *Variational integrators from Hamiltonizable nonholonomic systems*, J. Geom. Mech., 4 (2012), pp. 137–163.

[19] J. Frank, B. E. Moore, and S. Reich, *Linear PDEs and numerical methods that preserve a multisymplectic conservation law*, SIAM J. Sci. Comput., 28 (2006), pp. 260–277.

[20] J. Frank and S. Reich, *On Spurious Reflections, Nonuniform Grids and Finite Difference Discretizations of Wave Equations.* CWI Report MAS-E0406, Center for Mathematics and Computer Science, 2004.

[21] B. Gough and M. Galassi, *GNU Scientific Library Reference Manual*, Network Theory Ltd, 3 ed., 2009.

[22] M. Gromov, *Pseudo holomorphic curves in symplectic manifolds*, Inventiones Mathematicae, 82 (1985), p. 307.

[23] B. Gustafsson, *The convergence rate for difference approximations to mixed initial boundary value problems*, Mathematics of Computation, 29 (1975), pp. 396–406.

[24] E. Hairer and L. O. Jay, *Implicit Runge–Kutta methods for higher index differential-algebraic systems*, in Contributions in Numerical Mathematics, vol. 2, River Edge, N.J., 1993, World Scientific, pp. 213–224.

[25] B. Herbst and M. J. Ablowitz, *Numerically induced chaos in the nonlinear Schrödinger equation*, Physical Review Letters, 62 (1989), p. 2065.

[26] J. Hong and C. Li, *Multi-symplectic Runge–Kutta methods for nonlinear Dirac equations*, J. Comput. Phys., 211 (2006), pp. 448–472.

[27] J. Hong, Y. Liu, H. Munthe-Kaas, and A. Zanna, *Globally conservative properties and error estimation of a multi-symplectic scheme for Schrödinger equations with variable coefficients*, Appl. Numer. Math., 56 (2006), pp. 814–843.

[28] P. Hydon, *Multisymplectic conservation laws for differential and differential-difference equations*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 461 (2005), pp. 1627–1637.

[29] A. Islas, D. Karpeev, and C. Schober, *Geometric integrators for the nonlinear Schrödinger equation*, J. Comput. Phys., 173 (2001), pp. 116–148.

[30] A. Islas and C. Schober, *On the preservation of phase space structure under multisymplectic discretization*, J. Comput. Phys., 197 (2004), pp. 585–609.

[31] A. L. Islas and C. M. Schober, *Multi-symplectic methods for generalized schrödinger equations*, Future Gener. Comput. Syst., 19 (2003), pp. 403–413.

[32] C. Jia-Xiang, Y. Bin, and L. Hua, *Multisymplectic implicit and explicit methods for Klein–Gordon–Schrödinger equations*, Chinese Physics B, 22 (2013), p. 030209.

[33] E. Jones, T. Oliphant, and P. Peterson, *SciPy: Open source scientific tools for Python*, 2001–. http://www.scipy.org/.

[34] N. Kampanis, A. Vassilios, and J. Ekaterinaris, *Effective Computational Methods for Wave Propagation*, Boca Raton: Chapman & Hall/CRC, 1 ed., 2008.

[35] L. Kong, L. Wang, S. Jiang, and Y. Duan, *Multisymplectic Fourier pseudospectral integrators for Klein–Gordon–Schrödinger equations*, Sci. China Math., 56 (2013), pp. 915–932.

[36] H. Kreiss and G. Scherer, *On the existence of energy estimates for difference approximations for hyperbolic systems*, tech. rep., Technical report, Dept. of Scientific Computing, Uppsala University, 1977.

[37] H.-O. Kreiss and G. Scherer, *Finite element and finite difference methods for hyperbolic partial differential equations*, Mathematical aspects of finite elements in partial differential equations, Academic Press, New York, (1974), pp. 195–212.

[38] E. Le Donne, *Lecture notes on sub-Riemannian geometry.* Unpublished.

[39] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics*, Cambridge Monogr. Appl. Comput. Math., Cambridge University Press, Cambridge, UK, 2004.

[40] C. Liao and X. Ding, *Nonstandard finite difference variational integrators for nonlinear Schrödinger equation with variable coefficients*, Adv. Difference Equ., 2013 (2013), pp. 1–22.

[41] Z.-Q. Lv, Y.-S. Wang, and Y.-Z. Song, *A new multi-symplectic integration method for the nonlinear Schrödinger equation*, Chinese Physics Letters, 30 (2013), p. 030201.

[42] S. Martínez, J. Cortés, and M. de León, *The geometrical theory of constraints applied to the dynamics of vakonomic mechanical systems: The vakonomic bracket*, Journal of Mathematical Physics, 41 (2000), pp. 2090–2120.

[43] R. McLachlan and D. O'Neale, *Preservation and destruction of periodic orbits by symplectic integrators*, Numerical Algorithms, 53 (2010), pp. 343–362.

[44] R. McLachlan, Y. Sun, and P. Tse, *Linear stability of partitioned Runge–Kutta methods*, SIAM J. Numer. Anal., 49 (2011), pp. 232–263.

[45] R. McLachlan and M. Wilkins, *The multisymplectic diamond scheme*, SIAM Journal on Scientific Computing, 37 (2015), pp. A369–A390.

[46] R. I. McLachlan, K. Modin, O. Verdier, and M. Wilkins, *Symplectic integrators for index 1 constraints*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2150–A2162.

[47] R. I. McLachlan, K. Modin, O. Verdier, and M. C. Wilkins, *Geometric generalisations of* shake *and* rattle, Found. Comput. Math., (2013), pp. 1–32. DOI:10.1007/s10208-013-9163-y.

[48] R. I. McLachlan and M. Perlmutter, *Integrators for nonholonomic mechanical systems*, J. Nonlinear Sci., 16 (2006), pp. 283–328.

[49] R. I. Mclachlan, G. Reinout, and W. Quispel, *Geometric integrators for ODEs*, J. Phys. A, 39 (2006), pp. 5251–5285.

[50] R. I. McLachlan, Y. Sun, and B. Ryland, *High order multisymplectic Runge–Kutta methods*, SIAM J. Sci. Comput., 36 (2014), pp. A2199–A2226.

[51] K. Meyer and G. Hall, *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem*, Springer Science+Business Media, 2 ed., 1990.

[52] K. Modin and O. Verdier, *Integrability of nonholonomically coupled oscillators*, Discrete and Continuous Dynamical Systems, 34 (2014), pp. 1121–1130.

[53] R. Montgomery, *A tour of sub-Riemannian geometries, their geodesics and applications*, vol. 91 of Mathematical Surveys and Monographs, American Mathematical Society, Providence, RI, 2002.

[54] B. E. Moore and S. Reich, *Multi-symplectic integration methods for Hamiltonian PDEs*, Future Generation Computer Systems, 19 (2003), pp. 395–402.

[55] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *User Guide for MINPACK-1*, ANL-80-74, Argonne National Laboratory, (1980).

[56] P. Olsson, *Summation by parts, projections, and stability. I*, Mathematics of Computation, 64 (1995), pp. 1035–1065.

[57] ——, *Summation by parts, projections, and stability. II*, Mathematics of Computation, 64 (1995), pp. 1473–1493.

[58] M. J. D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, The Computer Journal, 7 (1964), pp. 155–162.

[59] E. L. Rees, *Graphical discussion of the roots of a quartic equation*, The American Mathematical Monthly, 29 (1922), pp. 51–55.

[60] S. Reich, *Multi-symplectic Runge–Kutta collocation methods for Hamiltonian wave equations*, J. Comput. Phys., 157 (2000), pp. 473–499.

[61] J.-P. Ryckaert, G. Ciccotti, and H. J. Berendsen, *Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes*, Journal of Computational Physics, 23 (1977), pp. 327–341.

[62] B. Ryland, *Multisymplectic integration*, PhD thesis, Massey University, New Zealand, 2007.

[63] B. N. Ryland and R. I. McLachlan, *On multisymplecticity of partitioned Runge–Kutta methods*, SIAM J. Sci. Comput., 30 (2008), pp. 1318–1340.

[64] B. N. Ryland, R. I. Mclachlan, and J. Frank, *On the multisymplecticity of partitioned Runge–Kutta and splitting methods*, Int. J. Comput. Math., 84 (2007), pp. 847–869.

[65] S. Edvardsson, K. G. Karlsson, and M. Engholm, *Accurate spin axes and solar system dynamics: Climatic variations for the earth and mars*, Astronom. and Astrophys., 384 (2002), pp. 689–701.

[66] Z.-j. Shang, *KAM theorem of symplectic algorithms for Hamiltonian systems*, Numerische Mathematik, 83 (1999), pp. 477–496.

[67] J. Simo and N. Tarnow, *The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics*, Z. Angew. Math. Phys., 43 (1992), pp. 757–792.

[68] R. D. Skeel and B. J. Leimkuhler, *Symplectic numerical integrators in constrained Hamiltonian systems*, Journal of Computational Physics, 112 (1994), pp. 117–125.

[69] I. Stewart, *The symplectic camel*, Nature, 329 (1987), pp. 17–18.

[70] B. Strand, *Summation by parts for finite difference approximations for d/dx*, Journal of Computational Physics, 110 (1994), pp. 47–67.

[71] B. Strand, *Numerical studies of hyperbolic ibvp with high-order finite difference operators satisfying a summation by parts rule*, Applied Numerical Mathematics, 26 (1998), pp. 497–521.

[72] J.-Q. Sun and M.-Z. Qin, *Multi-symplectic methods for the coupled 1D nonlinear Schrödinger system*, Comput. Phys. Comm., 155 (2003), pp. 221–235.

[73] Y. Sun, *Quadratic invariants and multi-symplecticity of partitioned Runge–Kutta methods for Hamiltonian PDEs*, Numerische Mathematik, 106 (2007), pp. 691–715.

[74] Y. Sun and P. Tse, *Symplectic and multisymplectic numerical methods for maxwell's equations*, Journal of Computational Physics, 230 (2011), pp. 2076–2094.

[75] M. Svärd and J. Nordström, *Review of summation-by-parts schemes for initial-boundary-value problems*, Journal of Computational Physics, 268 (2014), pp. 17–38.

[76] P. H. van der Kamp and G. R. W. Quispel, *The staircase method: Integrals for periodic reductions of integrable lattice equations*, J. Phys. A, 43 (2010). eid. 465207.

[77] A. M. Vershik and V. Y. Gershkovich, *Nonholonomic dynamical systems, geometry of distributions and variational problems*, vol. 16 of Encyclopaedia of Mathematical Sciences, Dynamical systems VII, Springer-Verlag, Berlin, 1994, pp. 1–81.

[78] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, *The numpy array: A structure for efficient numerical computation*, Computing in Science and Engineering, 13 (2011), pp. 22–30.

[79] J.-J. Wang and L.-T. Wang, *Multi-symplectic Preissmann scheme for a high-order wave equation of KdV type*, Appl. Math. Comput., (2013), pp. 4400–4409.

[80] P. F. Zhao and M. Z. Qin, *Multisymplectic geometry and multisymplectic Preissmann scheme for the KdV equation*, J. Phys. A, 33 (2000), pp. 3613–3626.