Faculty of Science

Local numerical discretization of Euler-Poisson reduction

MASTER'S THESIS

H.J. Balk

6186599

Mathematical Sciences

Supervisor:
Prof. dr. ir. J.E. Frank

Second reader:
Dr. P.A. Zegeling

Abstract

In 1993 McLachlan published a paper discussing Poisson discretizations of Hamiltonian PDE's; these discretizations are approximate systems which have a particular structure given by a Poisson bracket. In this thesis we derive a Poisson discretization of Burgers' equation. First, we use the Clebsch method to derive the Euler-Poincaré equations from the action principle associated to Burgers' equation. We then discretize the Euler-Poincaré equations. This discretization turns out to not be useful since the solution is discontinuous right away. Because of this, we will use constraint equations to enforce continuity. Then we use theory by Dirac to derive a Poisson system on the manifold defined by the constraint equations.

Acknowledgements

I would firstly like to thank my advisor Jason Frank, for suggesting the subject and for providing guidance along the way. I also want to thank Paul Zegeling, my second reader, for taking the time to read my thesis. Furthermore i would like to thank Esther Steenkamer for her insights into Poisson geometry, and Tess van Leeuwen along with the rest of my fellow students for keeping me motivated.

Contents

Introduction **4**
0.1 Background 4
0.2 Notation 6
0.3 Outline 7

1 Poisson systems **8**
1.1 Hamiltonian systems 8
1.1.1 Lagrangian mechanics 8
1.1.2 Hamilton's equations 9
1.1.3 Symplecticity 10
1.1.4 Symplectic integration 11
1.2 Poisson systems 13
1.2.1 Poisson brackets and Poisson structure 13
1.2.2 Differential equations on the dual of a Lie algebra 15
1.3 Poisson Integrators 17
1.3.1 Poisson maps 17
1.3.2 Example of a Poisson integrator 17

2 Clebsch variational principle and Burgers' equation **19**
2.1 Clebsch variational principle 19
2.1.1 Clebsch action principle and the Clebsch equations 19
2.1.2 Elimination of the Clebsch variables 20
2.2 Application to Burgers' equation 22
2.2.1 Euler-Poincaré equation 22
2.2.2 Discrete solution space 23
2.2.3 Poisson formulation 24
2.2.4 Constrained Poisson formulation 24

3 Constrained Poisson systems **26**
3.1 Constrained formulation 26
3.2 Index reduction 27
3.3 Dirac bracket approach 28
3.3.1 Dirac bracket 29
3.4 Reduction to the constraint manifold 31
3.4.1 Description of the method for slaving constraints 32
3.4.2 Dirac coordinates 34
3.5 Projection onto the constraint manifold 35
3.6 Integration of the constrained formulation 36
3.6.1 Reich's method 37
3.6.2 Poisson systems 38

4 Poisson discretization of Burgers' equation 40
4.1 Dirac's procedure . . . 40
4.2 Reduction to the constraint manifold . . . 41
4.3 Projection onto the constraint manifold . . . 42
4.3.1 Direct projection . . . 43
4.3.2 Projection after inverting . . . 44
4.4 Numerical results . . . 44

Conclusion 47

A Matlab codes 48
A.1 Example 1.20 . . . 48
A.2 Example 3.9 . . . 49
A.3 System (4.7) . . . 51
A.4 System (4.4) . . . 52

Introduction

0.1 Background

In many applications, such as classical mechanics and fluid dynamics, partial differential equations arise which have conserved quantities. With conserved quantities, also called invariants, we mean functions of the solution u which do not change with respect to time. In a physical context these conserved quantities can for example be the total energy or linear momentum. A simple example of systems with conserved quantities are conservation equations in one dimension. These are given by

$$\frac{\partial}{\partial t}u + a(u)\frac{\partial}{\partial x}u = 0, \tag{1}$$

for some function a. The conserved quantity in this case is given by $\int a(u(x))dx$ where the integral is taken over the domain of u. When approximating these kinds of equations, using for example a finite difference discretization, we want the discretized system to also have this property. So the conserved quantity should still be conserved by the approximate system. For this we will look into structure preserving algorithms.

A lot of work has been done on structure preserving algorithms, in the case of ODE's see for example the book by Hairer, Lubich and Wanner [1]. These algorithms, next to conserving the invariants of the differential equations, preserve the specific structure corresponding to the differential equations. To explain this we will look at the following system describing the evolution of a variable u

$$\frac{\partial u}{\partial t} = \mathscr{J}\frac{\delta \mathscr{H}[u]}{\delta u}, \tag{2}$$

where $\mathscr{J}$ is some operator and H is a functional. The *functional derivative* $\frac{\delta \mathscr{H}[u]}{\delta u}$ represent how much the functional $\mathscr{H}[u]$ changes when perturbing u,

$$\langle \frac{\delta \mathscr{H}[u]}{\delta u}, v\rangle = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\left[H[u+\varepsilon v] - H[u]\right].$$

Here $\frac{\delta \mathscr{H}[u]}{\delta u}$ is also a functional, we use notation $\langle .,.\rangle$ to denote the application of this functional to an element v. We will refer to $\langle .,.\rangle$ as the *dual representation*.

Systems like (2) have a conserved quantity given by $\mathscr{H}[u]$. There is also a structure associated to the operator $\mathscr{J}$, which we also would like to preserve. As an example we will look at Hamiltonian systems, these are systems of the form (2) where $\mathscr{J}$ is chosen to be the following constant operator

$$\mathscr{J} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \tag{3}$$

where I denotes the identity matrix, and the functional H is called the *Hamiltonian*. This gives the following system

$$\frac{\partial u}{\partial t} = J\nabla H(u).$$

We can see that H is conserved using of the following calculation,

$$\begin{aligned}\frac{\partial}{\partial t}H(u) = \nabla H(u)^T \frac{\partial}{\partial t}u = \nabla H(u)^T J \nabla H(u) = (\nabla H(u)^T J \nabla H(u))^T &= \nabla H(u)^T J^T \nabla H(u) \\ &= -\nabla H(u)^T J \nabla H(u).\end{aligned}$$

Now since $-\nabla H(u)^T J \nabla H(u) = \nabla H(u)^T J \nabla H(u) = 0$ we get $\frac{\partial}{\partial t}H(u) = 0$ so the Hamiltonian is conserved. For these types of systems a special class of numerical integrators, called *symplectic integrators*, exist. These not only preserve the invariant $H(u)$ but also preserve the specific structure given by the operator J. Later we will define this structure preservation more rigorously. Integrators such as this one are useful since they have been shown to have excellent long term behaviour. They might not be as accurate in the short term as more general methods such as high order Runge-Kutta methods, but they behave better when we want to integrate systems over longer periods of time. McLachlan & Atela studied the accuracy of symplectic methods in [2].

It was also McLachlan who started to look at Poisson discretizations of Hamiltonian PDE's in 1993 [3]. He showed that if the operator $\mathscr{J}$ is independent of u, we can find a skew-symmetric approximation for $\mathscr{J}$ and approximate the functional $\mathscr{H}$ with any quadrature. This will give us a semi-discrete Hamiltonian system. For non-constant operators $\mathscr{J}$ there is no recipe for finding such approximations in general. In this thesis we aim to find a skew-symmetric approximation for one such non-constant operator $\mathscr{J}$ which defines a *Poisson structure*, the precise definition of which we will see later. For now assume $J(u)$ is the approximation of $\mathscr{J}$, and this approximation defines a bilinear map $\{.,.\}$ in the following way

$$\{F,G\} = \nabla F^T J(u) \nabla G,$$

where ∇F denotes the gradient with respect to u. For now we want this bracket to satisfy the *Jacobi identity*, given by

$$\{f,\{g,h\}\} + \{g,\{h,f\}\} + \{h,\{f,g\}\} = 0.$$

These types of systems can be integrated in a similar way to Hamiltonian systems. We can construct so called *Poisson integrators*, which will then preserve any invariants and the Poisson structure $J(u)$ associated to the system. It has long been known that fluid dynamics have a non-linear Poisson structure. and for applications in for example ocean science or meteorology this structure is important since we would like the total energy to be conserved when doing simulations in these areas. There have been many attempts at finding such a Poisson structure for fluids. The only successful attempt was due to Vladimir Zeitlin in 1991 [4]. However, this Poisson structure is very restrictive. One downside being that this structure only works for imcompressible fluids on periodic domains. Another downside is that it is a two dimensional structure which cannot be generalized to three dimensions. Both reasons are due to the reliance on the vorticity-stream function formulation for fluids and the use of Fourier representation in deriving this Poisson structure.

In this thesis instead of looking for the Poisson structure of fluids we will try to find a Poisson structure for Burgers' equation, which is much simpler but still a nonlinear equation. Burgers' equation is a partial differential equation with applications to many areas in applied mathematics. It was first introduced by Harry Bateman in 1915 and later studied by Johannes Martinus Burgers in 1948. It is a one dimensional equation, for $u : \mathbb{R} \to \mathbb{R}$ the viscous Burgers' equation is given by the following partial differential equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \tag{4}$$

for x on some domain $[0,L]$ and ν the diffusion coefficient. For $\nu = 0$ we get the inviscid Burgers' equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0. \tag{5}$$

This equation is a particular example of a conservation equation, this can be seen by substituting $a(u) = u$ into equation (1).

To find this Poisson structure we want to start by writing Burgers' equation in the form of (2). The most obvious way to do this is by defining the operator $\mathscr{J}$ and the functional $\mathscr{H}$ in the following way

$$\mathscr{J} = \frac{\partial}{\partial x}, \qquad \mathscr{H}[u] = \int_{[0,L]} \frac{u^3}{6} dx.$$

We can then use finite differences to approximate these objects. To do this we discretize the domain, with periodic boundary conditions, into points $0 = x_1 < x_2 < ... < x_n = L$ and denote the values of the solution u at the points by u_i for $i = 1, ..., n$. Using this discretization we can approximate $\mathscr{J}$ and $\mathscr{H}$ in the following way

$$\mathscr{J} \approx \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & -1 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & \ddots & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & \vdots & \ddots & 0 & 1 \\ 1 & 0 & \dots & \dots & -1 & 0 \end{pmatrix}, \qquad \mathscr{H} \approx \sum_{i=1}^{n} \frac{u_i^3}{6}$$

This is one possible approach to approximating Burgers' equation, but since we want the procedure to be applicable to fluids we look for a different approximation. In the above case we have a constant approximation for $\mathscr{J}$, which will not be the case for the Poisson structure of fluid dynamics. Instead we will define the operator $\mathscr{J}$ in a way dependent on u

$$\mathscr{J}(u)v = u(\frac{\partial}{\partial x}v) + (\frac{\partial}{\partial x}u)v.$$

If we now define the Hamiltonian as

$$\mathscr{H} = \int_{[0,L]} \frac{u^2}{2} dx, \tag{6}$$

we can again write Burgers' equation in the form of equation (2). For this operator McLachlan stated in [3] that there exist no finite-difference Poisson discretizations, meaning discretizations which have a Poisson structure. We will investigate a non-standard approach to approximating the operator $\mathscr{J}$ in order to find a Poisson discretization of Burgers' equation.

In this thesis we will look to discretize Burgers' equation using the Clebsch method, which can be found in Cotter & Holm (2007) [5]. Since this approach looks to also work for fluids, we will attempt to use it to find a Poisson discretization Burgers' equation, which will hopefully also be applicable to fluid dynamics.

0.2 Notation

We will, in addition to the Leibniz notation, use Newton notation for the derivative with respect to time,

$$\dot{u} := \frac{\partial}{\partial t} u.$$

Furthermore we will mostly be using subscript notation to denote spacial derivatives,

$$u_x := \frac{\partial}{\partial x} u.$$

When taking gradients with respect to only a part of the variables we will also denote this by a subscript,

$$\nabla_u f = (\frac{\partial}{\partial u_1} f, ..., \frac{\partial}{\partial u_n} f)^T, \quad u = (u_1, .., u_n)^T.$$

0.3 Outline

In Chapter 1 we will start by looking at Hamiltonian systems and structure preserving algorithms for these systems. Then we will look at the generalization given by Poisson systems, and the corresponding integrators. Chapter 2 will contain the description of the Clebsch method, which is then applied to Burgers' equation. After that in Chapter 3 we will look at some theory surrounding constrained Poisson systems, this theory will then be applied to Burgers' equation in Chapter 4. Chapter 4 will contain the main result of this thesis, a Poisson discretization of Burgers' equation. In the same chapter we will also go over numerical results.

Chapter 1

Poisson systems

In this chapter we will formalize the notions found in the introduction. We will start by looking at Hamiltonian systems, then we will look at the generalization of Hamiltonian systems given by Poisson systems. For both of types we will look at ways to integrate them numerically.

1.1 Hamiltonian systems

Before looking at Poisson systems we will look at Hamiltonian systems, which are a specific example of Poisson systems. The reason for this is that Hamiltonian systems arise in a natural way from Lagrangian mechanics, as we will see. After this we will look at symplectic integrators, which were mentioned in the introduction, and why they are useful.

1.1.1 Lagrangian mechanics

We consider a mechanical system with position coordinates given by $q = (q_1, ..., q_n) \in Q \subset \mathbb{R}^n$, denote the kinetic energy of the system by $T(q, \dot{q})$ and the potential energy by $U(q)$. Lagrange's method provides a general way to find evolution equations corresponding to systems defined by the kinetic and potential energy. The first step of the method is to define the *Lagrangian* $L = T - U$, the goal now is to minimize the integral with respect to time of the Lagrangian.

$$\int_0^t L(q, \dot{q}) dt. \tag{1.1}$$

By the principle of least action this can be done by requiring that the first variation is equal to zero. In this case this is known as the *variational principle of Hamilton*. So we need the following to be true:

$$\delta \int_0^T L(q, \dot{q}) dt = 0. \tag{1.2}$$

We can calculate this variation along a curve $q_i(t)$ connecting two points in Q for $t \in [0, T]$. Denote by δq_i an arbitrary variation, then we get the following

$$\sum_{i=1}^n \int_0^t \frac{\partial L}{\partial q_i} \delta q_i + \frac{\partial L}{\partial \dot{q}_i} \delta \dot{q}_i dt.$$

Since $\delta \dot{q}_i = \frac{d}{dt} \delta q_i$ we can apply integration by parts

$$\sum_{i=1}^n \int_0^t \left[\frac{\partial L}{\partial q_i} - \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}_i) \right] \delta q_i dt = 0. \tag{1.3}$$

Now since the variation δq_i was chosen arbitrarily we get

$$\sum_{i=1}^n \frac{\partial L}{\partial q_i} - \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}_i) = 0. \tag{1.4}$$

These equations are known as the *Euler-Lagrange equations*. This concludes the method found by Lagrange, in the next section we will look how Hamilton rewrote these equations.

1.1.2 Hamilton's equations

Starting where we left of in the last section with the Euler-Lagrange equations we will now introduce new coordinates, this will simplify the equations and we can see a specific structure appear. Hamilton noticed that by introducing the conjugate momenta p_i defined by

$$p_i = \frac{\partial L}{\partial \dot{q}_i}, \quad 1 \leq i \leq n,$$

the Euler-Lagrange equations can be rewritten in a particularly nice way. Using these new coordinates we will define the *Hamiltonian*

$$H(p,q) = p^T \dot{q} - L(q,\dot{q}). \tag{1.5}$$

From here we will calculate the partial derivatives of this function which will allow us to rewrite (1.4)

$$\frac{\partial H}{\partial p_i} = \dot{q}_i + \sum_{j=1}^{n} p_j \frac{\partial \dot{q}_j}{\partial p_i} - \frac{\partial L}{\partial \dot{q}_j}\frac{\partial \dot{q}_j}{\partial p_i} = \dot{q}_i + \sum_{j=1}^{n} \frac{\partial L}{\partial \dot{q}_j}\frac{\partial \dot{q}_j}{\partial p_i} - \frac{\partial L}{\partial \dot{q}_j}\frac{\partial \dot{q}_j}{\partial p_i} = \dot{q}_i, \tag{1.6}$$

$$\frac{\partial H}{\partial q_i} = \sum_{j=1}^{n} p_j \frac{\dot{q}_j}{\partial q_i} - \frac{\partial L}{\partial q_i} - \sum_{j=1}^{n} \frac{\partial L}{\partial \dot{q}_j}\frac{\dot{q}_j}{\partial q_i} = \sum_{j=1}^{n} \frac{\partial L}{\partial \dot{q}_j}\frac{\dot{q}_j}{\partial q_i} - \frac{\partial L}{\partial \dot{q}_j}\frac{\dot{q}_j}{\partial q_i} - \frac{\partial L}{\partial q_i}\frac{\partial L}{\partial \dot{q}_j}\frac{\dot{q}_j}{\partial q_i} = -\frac{\partial L}{\partial q_i}. \tag{1.7}$$

Substituting the second expression into (1.4) we get the following equivalent equation

$$\frac{\partial H}{\partial q_i} = -\frac{d}{dt} p_i,$$

Together with the first expression from (1.6) we get **Hamilton's equations**

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \tag{1.8}$$

$$\dot{p}_i = -\frac{\partial H}{\partial q_i}. \tag{1.9}$$

As we have seen in the introduction Hamiltonian systems can be written in the form of equation (2)

$$\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = J \nabla H(p,q), \qquad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \tag{1.10}$$

These types of systems are called *Hamiltonian systems*. We can introduce the following bilinear map of functions of (p,q),

$$\{F,G\} = \sum_{i=1}^{n} \left(\frac{\partial F}{\partial q_i}\frac{\partial G}{\partial p_i} - \frac{\partial F}{\partial p_i}\frac{\partial G}{\partial q_i}\right). \tag{1.11}$$

This map is known as the canonical *Poisson bracket*. Using this bracket we can simplify (1.10) even further, because the Lie derivative of a function F along the solution of (1.10) is given by

$$\frac{\partial}{\partial t} F = \{F,H\}, \tag{1.12}$$

this expression can be used instead of the expression in (1.10). For $F(p,q) = q_i$ and $F(p,q) = p_i$ we get back equations (1.8) and (1.9), respectfully. In the introduction we have already seen that these types of systems have invariants, we will now formalize this notion.

Definition 1.1 (First integral)**.** Consider the following system

$$\dot{x} = f(x), \tag{1.13}$$

A non-constant function $I(x)$ is called a **first integral** (also known as a conserved quantity or invariant) when the following condition holds

$$I'(x)f(x) = 0, \quad \text{for all } x.$$

Given a solution $x(t)$ of system (1.13) we can look at the derivative of I along this solution

$$\frac{d}{dt}I(x(t)) = I'(x(t))\dot{x}(t) = I'(x(t))f(x(t)) = 0.$$

Since $I(x(t))$ does not change with respect to time we have $I(x(t)) = C \in \mathbb{R}$ for all t, so first integrals stay constant along solutions of a system. In the case of Hamiltonian systems a function I is a first integral if $\{I,H\} = 0$, since then $\frac{\partial}{\partial t}I = 0$ so I again stays constant along solutions. Since the bracket defined in equation (1.11) is skew-symmetric we can see that the Hamiltonian H is always a first integral for Hamiltonian systems

$$\frac{\partial}{\partial t}H = \{H,H\} = 0,$$

So the Hamiltonian of our system is constant along solutions. This is one of the properties of Hamiltonian systems we would like to keep while numerically integrating our systems. Like we noted in the introduction we would also like to keep the structure associated to Hamiltonian systems, we will look at how to do this in the next section.

1.1.3 Symplecticity

In the previous section we defined Hamiltonian systems and noted that the Hamiltonian H is conserved along solutions of the system. In this section we will look at another property of Hamiltonian systems, namely the *symplecticity* of its flow. First we will define what symplecticity means, after which we will show that the flow of a Hamiltonian system is a symplectic map whenever the Hamiltonian is twice continuously differentiable.

Let $z = (p,q)$, then we can write the Hamiltonian system (1.10) as

$$\dot{z} = J\nabla H(z). \tag{1.14}$$

Consider a transformation $\phi : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ and write $\tilde{z} = \phi(z)$. We have the following

$$\dot{\tilde{z}} = \phi'(z)\dot{z} = \phi'(z)J\nabla H(z) = \phi'(z)J\phi'(z)^T\nabla\tilde{H}(\tilde{z}),$$

where $\phi'(z)$ is the Jacobian of ϕ and $\tilde{H}(\tilde{z}) = H(z)$. The resulting system for $\tilde{z}$ is Hamiltonian if and only if

$$\phi'(z)J\phi'(z)^T = J. \tag{1.15}$$

Taking the determinant of this expression gives

$$\begin{aligned}\det(J) = \det(\phi'(z)J\phi'(z)^T) &= \det(\phi'(z))\det(J)\det(\phi'(z)^T),\\ 1 = \det(\phi'(z))\det(\phi'(z)^T) &= \det(\phi'(z))^2,\end{aligned}$$

so $\det(\phi'(z)) = \pm 1$. Therefore $\phi'(z)$ is invertible, so we get the following for the inverse of (1.15)

$$\begin{aligned}&\phi'(z)^{-T}J^{-1}\phi'(z)^{-1} = J^{-1},\\ &\phi'(z)^{-T}J\phi'(z)^{-1} = J,\\ &\phi'(z)^{T}J\phi'(z) = J,\end{aligned}$$

where from the first to the second equality we used that $J^{-1} = -J$.

We can therefore transform a Hamiltonian system using a transformation ϕ into another Hamiltonian system whenever $\phi'(z)^T J\phi'(z) = J$. This is the structure preservation property which was mentioned in the introduction. This motivates the following definition

Definition 1.2 (Symplectic map)**.** A differentiable mapping ϕ is called **symplectic** if and only if the following condition holds

$$\phi'(z)^T J\phi'(z) = J. \tag{1.16}$$

As was mentioned in the beginning of this section the flows of Hamiltonian systems are symplectic maps. This result is stated in the following theorem. It was first proved by Poincaré in 1899 and the theorem along with its proof can be found in [1]

Theorem 1.3. *Assume that the Hamiltonian $H(z)$ is twice continuously differentiable. Then the flow of the corresponding Hamiltonian system is a symplectic map.*

Proof. Denote by $\varphi_t(z_0) = (z(t,z_0))$ the flow of system (1.14), which is defined by

$$\frac{d}{dt}\varphi_t(z_0) = J\nabla H(\varphi_t(z_0)). \tag{1.17}$$

We want to prove that the identity in (1.16) holds for φ_t. Note that for $t = 0$ we have $\varphi_t = \text{id}$ which satisfies (1.16) trivially, so we just have to prove that the time derivative of $\varphi_t'(z_0)^T J\varphi_t'(z_0)$ vanishes.

First we will differentiate equation (1.17) with respect to z to get the following variational equation

$$\frac{d}{dt}\varphi_t'(z_0) = J\nabla^2 H(\varphi_t(z_0))\varphi_t'(z_0),$$

where $\nabla^2 H$ denotes the Hessian matrix. Now we can take the time derivative of $\varphi_t'(z_0)^T J\varphi_t'(z_0)$ to get the following

$$\begin{aligned}\frac{d}{dt}(\varphi_t'(z_0)^T J\varphi_t'(z_0)) &= (\frac{d}{dt}\varphi_t')(z_0)^T J\varphi_t'(z_0) + \varphi_t'(z_0)^T J(\frac{d}{dt}\varphi_t'(z_0)) \\ &= (\varphi_t'(z_0))^T\nabla^2 H(\varphi_t(z_0)J^T J\varphi_t'(z_0) + \varphi_t'(z_0)^T JJ\nabla^2 H(\varphi_t(z_0))\varphi_t'(z_0) \\ &= (\varphi_t'(z_0))^T\nabla^2 H(\varphi_t(z_0)\varphi_t'(z_0) - \varphi_t'(z_0)^T\nabla^2 H(\varphi_t(z_0))\varphi_t'(z_0) \\ &= 0,\end{aligned}$$

where we used that $\nabla^2 H$ is symmetric and $JJ = -J^T J = -I$. This implies $\varphi_t'(z_0)^T J\varphi_t'(z_0)$ is constant for all t, and since $\varphi_0'(z_0)^T J\varphi_0'(z_0) = J$ we can conclude that the flow of a Hamiltonian system is a symplectic map for all t. □

1.1.4 Symplectic integration

In the last section we have seen that the flow of Hamiltonian systems is symplectic. Because we would like to keep this property when approximating our differential equations we have to look at a special class of integrators called *symplectic integrators*. We will start by defining such integrators.

Definition 1.4 (Symplectic integrator)**.** A one-step map $z_{k+1} = \phi(z_k)$ is called **symplectic** if ϕ is a symplectic map.

Note that the definition of symplectic integrators is independent of the Hamiltonian, so once we have found a symplectic integrator it will work for all Hamiltonian systems with the same structure matrix J. In the rest of this section we will look at two examples of symplectic integrators.

Example 1.5. The first example of a symplectic integrator is one of the most simple examples. It is a combination of the forward and backward Euler method, both of which are not symplectic by themselves. The method is given by the following recursion

$$p_{k+1} = p_k + \Delta t \nabla_q H(p_{k+1}, q_k), \qquad q_{k+1} = q_k - \Delta t \nabla_p H(p_{k+1}, q_k),$$

or

$$p_{k+1} = p_k + \Delta t \nabla_q H(p_k, q_{k+1}), \qquad q_{k+1} = q_k - \Delta t \nabla_p H(p_k, q_{k+1}),$$

where Δt denotes the stepsize. Note that the method is explicit in one variable and implicit in the other. This method is symplectic, which one can show by determining the Jacobian matrix of the method and then verifying condition (1.16). $\triangle$

For the second example we will need the next theorem. This theorem can also be found in [1].

Theorem 1.6. *Let ϕ_1, ϕ_2 be symplectic maps, then their composition $\phi_1 \circ \phi_2$ is also symplectic.*

Proof. Define $\psi(z) = \phi_1 \circ \phi_2(z)$, by the chain rule we have the following

$$\psi'(z) = \frac{\partial}{\partial z} \phi_1 \circ \phi_2(z) = \phi_1'(\phi_2(z))\phi_2'(z).$$

Using this we can show the identity (1.16)

$$\begin{aligned} \psi'(z) J \psi'(z)^T = (\phi_1'(\phi_2(z))\phi_2'(z)) J (\phi_1'(\phi_2(z))\phi_2'(z))^T &= \phi_1'(\phi_2(z))(\phi_2'(z) J \phi_2'(z)^T)\phi_1'(\phi_2(z))^T \\ &= \phi_1'(\phi_2(z)) J \phi_1'(\phi_2(z))^T \\ &= J. \end{aligned}$$

So the composition ψ is also symplectic. $\square$

Example 1.7. Suppose the Hamiltonian can be split into a number of parts

$$H = H^{(1)} + H^{(2)} + ... + H^{(m)}, \quad m \in \mathbb{N},$$

and all of the subsystems $\dot{z} = J \nabla H^{(i)}(z)$ can be solved explicitly. We can then compose all of the flows of the subsystems, and as we have seen in the theorem above this will give us a symplectic integrator. Now what is left to show is that this is a consistent approximation of the original system. We will show this for a splitting into two subsystems. Consider the following differential equation,

$$\dot{y} = f^{(1)}(y) + f^{(2)(y)}.$$

Let $\phi_{\Delta t}^{(1)}, \phi_{\Delta t}^{(2)}$ be the flows corresponding to $\dot{y} = f^{(1)}(y)$ and $\dot{y} = f^{(2)}(y)$, respectfully. We have the following splitting

$$\varphi_{\Delta t} = \phi_{\Delta t}^{(1)} \circ \phi_{\Delta t}^{(2)}.$$

By Lemma III.5.1 of [1] we have

$$\varphi_{\Delta t} = \exp(\Delta t D_1) \exp(\Delta t D_2) I, \tag{1.18}$$

where the Lie derivative D_i is defined as

$$D_i g(y) = g'(y) f^{(i)}(y), \quad i = 1, 2.$$

Using the Baker-Campbell-Hausdorff formula (Section III.4.2 from [1]) we can rewrite equation (1.18),

$$\varphi_{\Delta t} = \exp(\Delta t \tilde{D}) I,$$

where

$$\tilde{D} = D_1 + D_2 + O(\Delta t).$$

This equation implies $\varphi_{\Delta t}$ is the flow corresponding to a modified differential equation given by

$$\dot{\tilde{y}} = \tilde{f}(y),$$

where $\tilde{f} = f^{(1)} + f^{(2)} + O(\Delta t)$. Therefore the splitting method yields consistent approximations of the original system. $\triangle$

1.2 Poisson systems

In this section we look at a generalization of Hamiltonian systems. They are of the same form, but the constant matrix J is replaced by a non-constant matrix $B(y)$. Such systems are called **Poisson systems**, and where introduced by Sophus Lie in 1888. These types of systems arise in many contexts, two examples are the Lotka-Volterra predator-prey model and rigid body dynamics.

In the first section we will look at the formal definition of the Poisson bracket and define what a Poisson system is. After that we will look at Poisson integrators, which are similar to symplectic maps in that they preserve the Hamiltonian and Poisson structure. Furthermore Poisson integrators will preserve the *Casimirs*, which are invariant functions which, as we will see, arise from the specific Poisson structure.

1.2.1 Poisson brackets and Poisson structure

We will begin by giving the formal definition for the Poisson bracket. We have already seen the canonical Poisson bracket corresponding to Hamiltonian systems in section 1.1.2, in the following definition we will see the general form of the bracket. After this we will see that equation (1.12) will define a Poisson system whenever we are working with a Poisson bracket.

Definition 1.8 (Poisson Bracket)**.** A **Poisson bracket** on a smooth manifold M is a bilinear map

$$\{.,.\} : C^\infty(M) \times C^\infty(M) \to C^\infty(M)$$

Satisfying the following conditions

1. Skew-symmetry: $\{f,g\} = -\{g,f\}$,
2. Jacobi identity: $\{f,\{g,h\}\} + \{g,\{h,f\}\} + \{h,\{f,g\}\} = 0$,
3. Leibniz' rule: $\{fg,h\} = f\{g,h\} + g\{f,h\}$,

where $f,g,h \in C^\infty(M)$.

In local coordinates on the manifold M we get the following expression for the Poisson bracket

$$\{f,g\} = \sum_{i,j=1}^{n} \frac{\partial f}{\partial x_i} b_{ij}(x) \frac{\partial g}{\partial x_j} = \nabla f^T B(x) \nabla g, \tag{1.19}$$

where the **Structure matrix** $B(x) = (b_{ij}(x))_{ij}$ is defined by $b_{ij}(x) = \{x_i, x_j\}$. If we take $B(x) = J$ where J was defined in (1.10) we get back the canonical Poisson bracket.

The following lemma from [1] gives the equivalent conditions for which matrices B define a Poisson structure, so that (1.19) is a Poisson bracket.

Lemma 1.9. *The bracket in (1.19) for a structure matrix $B(x) = (b_{ij}(x))_{ij}$ is a Poisson bracket if and only if the following two conditions hold:*

1. $b_{ij} = -b_{ji}$, *for all* i, j,
2. $\sum_{l=1}^{n}(\frac{\partial b_{ij}(x)}{\partial x_l}b_{lk}(x) + \frac{\partial b_{jk}(x)}{\partial x_l}b_{li}(x) + \frac{\partial b_{ki}(x)}{\partial x_l}b_{lj}(x)) = 0,$ *for all* i, j, k.

Using the above definitions we can define what a Poisson system is.

Definition 1.10 (Poisson systems)**.** Given a matrix $B(x)$ satisfying the conditions from lemma (1.9) the following is called a **Poisson system**;

$$\dot{x} = B(x)\nabla H(x), \tag{1.20}$$

where the function $H : M \to \mathbb{R}$ is again called the **Hamiltonian**. Just like in the Hamiltonian case we can rewrite this system using the Poisson bracket

$$\dot{F} = \{F, H\},$$

for $\{f, g\} = \nabla f^T B(x)\nabla g$. We can again get back the original system by substituting $F(x) = x_i$ for $1 \leq i \leq n$.

Since the bracket is again anti-symmetric we can see that just like in the Hamiltonian case the Hamiltonian is a first integral for Poisson systems. There can also be other invariants for Poisson systems, given by so called **Casimirs**, like we mentioned in the beginning of this section these depend on the specific Poisson structure.

Definition 1.11 (Casimirs)**.** A function $C : M \to \mathbb{R}$ is called a **Casimir** if the following holds

$$\nabla C(y)^T B(y) = 0, \qquad \text{for all } y.$$

Remark 1.12. Note that Casimirs are first integrals for all Poisson systems with structure matrix given by B, independent of the choice of Hamiltonian. In addition there may be some first integrals $I(y)$ for which the Poisson bracket with a certain Hamiltonian $H(y)$ vanish,

$$\{I, H\} = 0.$$

Next we will look at an example of a Poisson system, this example will be reused in the following chapters.

Example 1.13 (Lorentz 86)**.** The following system was first derived by Lorenz in 1986 [6] and has applications to atmospheric science.

$$\begin{aligned} \dot{x}_1 = -x_2x_3 + bx_2x_5, \qquad \dot{x}_2 = x_1x_3 - bx_1x_5, \qquad \dot{x}_3 = -x_1x_2, \\ \dot{x}_4 = -\frac{x_5}{\varepsilon}, \qquad \dot{x}_5 = \frac{x_4}{\varepsilon} + bx_1x_2, \end{aligned} \tag{1.21}$$

where $b, \varepsilon \in \mathbb{R}$ are constants. This is a Poisson system with Poisson bracket given by

$$\begin{aligned} \{F, G\} = \frac{\partial F}{\partial x_1}x_2(b\frac{\partial G}{\partial x_5} - \frac{\partial G}{\partial x_3}) + \frac{\partial F}{\partial x_2}x_1(\frac{\partial G}{\partial x_3} - b\frac{\partial G}{\partial x_5}) + \frac{\partial F}{\partial x_3}(x_2\frac{\partial G}{\partial x_1} - x_1\frac{\partial G}{\partial x_2}) - \frac{1}{\varepsilon}\frac{\partial F}{\partial x_4}\frac{\partial G}{\partial x_5} \\ + \frac{\partial F}{\partial x_5}(-bx_2\frac{\partial G}{\partial x_1} + bx_1\frac{\partial G}{\partial x_2} + \frac{1}{\varepsilon}\frac{\partial G}{\partial x_4}). \end{aligned}$$

The corresponding structure matrix is

$$B(x) = \begin{pmatrix} 0 & 0 & -x_2 & 0 & bx_2 \\ 0 & 0 & x_1 & 0 & -bx_1 \\ x_2 & -x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\varepsilon} \\ -bx_2 & bx_1 & 0 & \frac{1}{\varepsilon} & 0 \end{pmatrix}.$$

Together with the Hamiltonian given by $H(x) = \frac{1}{2}(x_1^2 + 2x_2^2 + x_3^2 + x_4^2 + x_5^2)$ equation (1.21) can now be written as

$$\dot{x} = B(x)\nabla H(x).$$

A Casimir for this Poisson structure is given by:

$$C(x) = \frac{1}{2}(x_1^2 + x_2^2).$$

△

Changing the coordinates of Poisson system results in another equivalent Poisson system. Since this is quite useful we will give the new equations corresponding to the Poisson system after the coordinate transformation as a theorem.

Theorem 1.14 (Change of coordinates formula)**.** *Consider the system given in (1.20), and new coordinates $\tilde{x}_1, ..., \tilde{x}_n$ such that the Jacobian $A(x) = \frac{\partial \tilde{x}}{\partial x}$ is invertible. Then the Poisson structure in the new coordinates is given by*

$$\{F, G\} = \nabla F(\tilde{x})^T \tilde{B}(\tilde{x}) \nabla G(\tilde{x}),$$

Where $\tilde{B}(\tilde{x}) = A(x)B(x)A(x)^T$. System (1.20) is then equivalent to

$$\dot{\tilde{x}} = \tilde{B}(\tilde{x}) \nabla K(\tilde{x}),$$

where the Hamiltonian K of the new Poisson system is given by

$$K(\tilde{x}) = H(x)$$

1.2.2 Differential equations on the dual of a Lie algebra

In this section we will look at differential equations on the dual of Lie algebras. These types of equations are important since they are equivalent to a specific type of Poisson system, as we will see later. This section is based on the proof of Theorem VII.5.5 from [1]. The theorem itself will be given at the end of this section. First we will recall the definition of a Lie-algebra:

Definition 1.15 (Lie algebra)**.** A **Lie algebra** is a vector space V together with a bilinear map $[.,.] : V \times V \to V$. This map is called the **Lie bracket** and satisfies the following properties:

- Skew-symmetry:
$$[f, g] = -[g, f],$$
- Jacobi identity:
$$[f, [g, h]] + [g, [h, f]] + [h, [f, g]] = 0,$$

for $f, g, h \in V$.

Let V be a Lie algebra, the dual V^* is defined as all linear forms $Y : V \to \mathbb{R}$. For $Y \in V^*$ and $X \in V$ the *duality pairing* is defined by $\langle Y, X \rangle$, meaning that we denote $Y(X) = \langle Y, X \rangle$. Let $\phi_1, ..., \phi_n$ be a basis for V and $\varphi_1, ..., \varphi_n$ be the dual basis in V^* defined by $\langle \varphi_i, \phi_j \rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta function. Assume that the Lie bracket associated to V is given by the following

$$[\phi_i, \phi_j] = \sum_{k=1}^{n} C_{ij}^k \phi_k,$$

here $C_{ij}^k \in \mathbb{R}$ are referred to as the **structure constants** of the Lie algebra.

Consider the following differential equation on V^*

$$\langle \dot{Y}, X \rangle = \langle Y, [\frac{\delta H^*}{\delta Y}(Y), X] \rangle, \quad \text{for all } X \in V, \tag{1.22}$$

where $H^* : V^* \to \mathbb{R}$, and $\frac{\delta H^*}{\delta Y}$ is the variational derivative defined as

$$\langle \frac{\delta H^*}{\delta Y}, v \rangle = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \left[H(Y + \varepsilon v) - H(Y) \right],$$

for $v \in V$. Note that $H^* \in (V^*)^*$, which we have identified with V. To some vector $y = (y_1, ..., y_n) \in \mathbb{R}^n$ we can associate an element $Y = \sum_{i=1}^n y_i \varphi_i \in V^*$, we can use this to define a Hamiltonian $H : \mathbb{R}^n \to \mathbb{R}$ by $H(y) = H^*(Y)$. Differentiating this last equality gives

$$\frac{\partial H(y)}{\partial y_j} = \langle H'(Y), \phi_j \rangle, \qquad (H^*)'(Y) = \sum_{i=1}^n \frac{\partial H(y)}{\partial y_i} \phi_i. \tag{1.23}$$

We can use this to rewrite the left and right hand sides of (1.22). For the left hand side we have

$$\langle \dot{Y}, \phi_j \rangle = \dot{y}_j$$

and for the right hand side we have

$$\begin{aligned}\langle Y, [(H^*)'(Y), \phi_j] \rangle = \langle Y, \sum_{i=1}^n \frac{\partial H(y)}{y_i} [\phi_i, \phi_j] \rangle = \langle Y, \sum_{i=1}^n \frac{\partial H(y)}{y_i} \sum_{k=1}^n C_{ij}^k \phi_k \rangle &= \sum_{i=1}^n \sum_{k=1}^n C_{ij}^k \langle Y, \phi_k \rangle \frac{\partial H(y)}{y_i} \\ &= \sum_{i=1}^n (\sum_{k=1}^n C_{ij}^k y_k) \frac{\partial H(y)}{y_i}.\end{aligned}$$

So we get a system of the following form

$$\dot{y}_j = \sum_{i=1}^n (\sum_{k=1}^n C_{ij}^k y_k) \frac{\partial H(y)}{y_i}, \quad \text{for all } j. \tag{1.24}$$

This system is equivalent to $\dot{y} = B(y) \nabla H(y)$ when we define $B(y) = (b_{ij}(y))$ with

$$b_{ij}(y) = \sum_{k=1}^n C_{ji}^k y_k.$$

This structure matrix B satisfies the conditions from (1.9), so the corresponding system is a Poisson system. Note that every entry of the matrix B is a linear function of y. Systems with this property are called **Lie-Poisson** systems. We will summarize this section in the following theorem, which together with the proof given above can be found in [1].

Theorem 1.16. *Let V be a Lie algebra with basis given by $\phi_1, ..., \phi_n$ such that*

$$[\phi_i, \phi_j] = \sum_{k=1}^n C_{ij}^k \phi_k,$$

for some constants $C_{ij}^k \in \mathbb{R}$. Furthermore let $\varphi_1, ..., \varphi_n$ be the dual basis in V^*, such that for $Y \in V^*$ we have $Y = \sum_{i=1}^n y_i \varphi_i$, where $y = (y_1, ..., y_n) \in \mathbb{R}^n$. Then the differential equation given by

$$\langle \dot{Y}, X \rangle = \langle Y, [\frac{\delta H^*}{\delta Y}(Y), X] \rangle, \tag{1.25}$$

for some Hamiltonian H^*, is equivalent to the Lie-Poisson system

$$\dot{y} = B(y) \nabla H(y),$$

where H is defined such that $H(y) = H^*(Y)$ and $B(y) = (b_{ij}(y))_{ij}$ where

$$b_{ij}(x) = \sum_{k=1}^n C_{ji}^k x_k.$$

1.3 Poisson Integrators

Unlike in the Hamiltonian case there is no general integrator which works for every Poisson system, for every system we have to exploit the specific Poisson structure. In this section we will look at possible strategies to obtain Poisson integrators. First we define *Poisson maps*, which are similar to symplectic maps for the Hamiltonian case. We will see that just as in the Hamiltonian case where we wanted our integrators to be symplectic maps, we now want our integrators to be Poisson maps when working with Poisson systems.

1.3.1 Poisson maps

Definition 1.17 (Poisson map)**.** A transformation $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ is a **Poisson map** if the following holds:

$$\varphi'(x)B(x)\varphi'(x)^T = B(\varphi(x)), \tag{1.26}$$

where $\varphi'(x)$ denotes the Jacobian matrix of φ. We can reformulate this conditions in terms of the Poisson bracket in the following way

$$\{F\circ\varphi, G\circ\varphi\}(x) = \{F,G\}(\varphi(x)), \tag{1.27}$$

for all smooth functions F,G defined on $\varphi(U)$.

From this definition it is obvious that Poisson maps are the analog of symplectic maps in the Poisson case. The definitions are equivalent for the canonical Poisson structure given by $B(x) = J^{-1}$. For Hamiltonian systems we have seen that their flow φ_t is a symplectic mapping for all t, similarly for Poisson systems we have the following theorem from [1].

Theorem 1.18. *The flow φ_t of a Poisson system is a Poisson map for all $t \geq 0$.*

When integrating Hamiltonian systems we wanted the integrator to be a symplectic map. For Poisson systems we again want this to be the case, except we want it to be a Poisson map instead of a symplectic map. This leads to the following definition,

Definition 1.19 (Poisson Integrator)**.** A one-step map $x_{k+1} = \phi(x_k)$ is called a **Poisson integrator** when it is a Poisson map.

1.3.2 Example of a Poisson integrator

Like we mentioned in the beginning of this section there is no general integrator which works for every Poisson structure. It might for example be the case that the symplectic Euler method is Poisson, but we would have to check that using Definition (1.17) for the specific Poisson structure we are working with. There is however one method which works in a lot of cases. This method is given by the splitting method, which we have already seen in section 1.1.4.

Splitting methods Just like in the Hamiltonian case we can split up the Hamiltonian into integrable pieces and then compose the flows corresponding to the splitting. If we solve the subsystems exactly the flows will be Poisson maps and since compositions of Poisson maps are again Poisson maps we will get a Poisson integrator. The proof of the fact that compositions of Poisson maps are again Poisson maps is very similar to the proof of Theorem 1.6. Furthermore, the proof that the splitting method yields a consistent approximation of the system still holds here. In the next example we will see how to apply the splitting method to a Poisson system.

Example 1.20. Consider the Lorenz 86 system from example (1.13). We can split the Hamiltonian into pieces $H = H^{(1)} + H^{(2)} + H^{(3)}$ where

$$H^{(1)} = \frac{1}{2}(x_1^2 + 2x_2^2), \qquad H^{(2)} = \frac{1}{2}(x_3^2 + x_4^2), \qquad H^{(3)} = \frac{1}{2}(x_5^2).$$

The corresponding systems are

$$\begin{cases} \dot{x}_1 &= 0, \\ \dot{x}_2 &= 0, \\ \dot{x}_3 &= -x_1x_2, \\ \dot{x}_4 &= 0, \\ \dot{x}_5 &= bx_1x_2, \end{cases} \qquad \begin{cases} \dot{x}_1 &= -x_2x_3, \\ \dot{x}_2 &= x_1x_3, \\ \dot{x}_3 &= 0, \\ \dot{x}_4 &= 0, \\ \dot{x}_5 &= \frac{1}{\varepsilon}x_4, \end{cases} \qquad \begin{cases} \dot{x}_1 &= bx_2x_5, \\ \dot{x}_2 &= -bx_1x_5, \\ \dot{x}_3 &= 0, \\ \dot{x}_4 &= -\frac{1}{\varepsilon}x_5, \\ \dot{x}_5 &= 0. \end{cases}$$

Each of these subsystems are explicitly solvable. We will solve the system on the interval $[0,25]$ with $\Delta t = 2.5 \cdot 10^{-4}$. When we take $b = 3$ and $\varepsilon = 0.01$ we get the following figures representing the change in the Hamiltonian and Casimir with respect to their initial values.

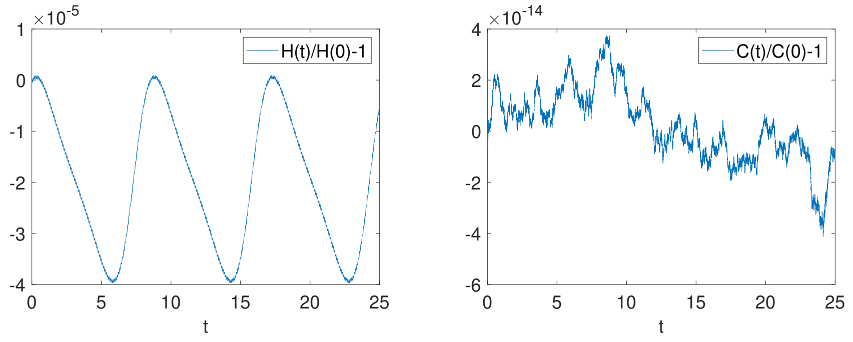

Figure 1.1: Conservation of the Hamiltonian H and Casimir C.

$\triangle$

Chapter 2

Clebsch variational principle and Burgers' equation

In this chapter we will first outline the Clebsch method as described in [5]. Then we will apply this theory to approximate Burgers' equation as a Poisson system.

2.1 Clebsch variational principle

The Clebsch method is an approach similar to Lagrange's method outlined in section 1.1.1. We again start with a Lagrangian with an associated variational principle, but instead of minimizing this principle along an arbitrary curve we enforce the path within the action principle using Lagrange multipliers. This will give us extra differential equations for the Lagrange multipliers and the position variables on the manifold. Given some conditions we can then eliminate these extra equations. This method will then produce the Euler-Poincaré equations corresponding to our action principle, which are the analogous equations to the Euler-Lagrange equations in Lagrange's method.

2.1.1 Clebsch action principle and the Clebsch equations

Let V be a vector space, we will consider the following variational principle

$$S = \int l[\xi(t)]dt, \tag{2.1}$$

where $l : V \to \mathbb{R}$ is the Lagrangian. For the principle of least action we needed S to be minimal along a path q, for the Clebsch variational principle we will enforce this path within the integral. In order to constrain our dynamics in this way we will define the *velocity map*. Let $q \in M$ be a point on a manifold M and let T_qM be the tangent space at that point, the velocity map is defined as follows

$$\mathscr{L}_\xi : V \times M \to T_qM,$$

such that $\mathscr{L}_\xi q$ is linear in ξ. This map defines the path which the element $q \in M$ will take in time

$$\dot{q}(t) = \mathscr{L}_\xi q(t). \tag{2.2}$$

As mentioned we want to minimize S subject to the condition given by (2.2). We can do this by using *Lagrange multipliers*, defined in the following theorem.

Theorem 2.1 (Lagrange multiplier theorem)**.** *Let $f : V \to \mathbb{R}$ be some continuously differentiable objective function and let $g : V \to \mathbb{R}^m$, $m \leq \dim(V)$, be the constraints function, also continuously differentiable. Now let $x^* \in V$ be the minimal value for f such that $g(x^*) = 0$. Then there exists a unique $\lambda^* \in \mathbb{R}^m$ such that:*

$$Df(x^*) = (\lambda^*)^T Dg(x^*).$$

*The value λ^* is called the **Lagrange multiplier**.*

In our case we can let $f = l$ and $g = \dot{q}(t) - \mathscr{L}_\xi q(t)$. To implement this theorem we will change the Lagrangian in our variational principle in the following way.

$$l^*[\xi] = l[\xi] + \langle p, \dot{q} - \mathscr{L}_\xi q \rangle_{T^*M},$$

where $p \in T^*M$ is the Lagrange multiplier and $\langle .,. \rangle_{T^*M}$ denotes the duality pairing on T^*M. The Lagrange multiplier theorem now states that there exists some value for p such that when we minimize l^* we get a minimum for l which satisfies the constraint given in (2.2). Using this fact we can define the *Clebsch action principle*.

Definition 2.2 (Clebsch action principle)**.** For a functional $l : V \to \mathbb{R}$, the **Clebsch action principle** is given by:

$$\delta \int_{t_1}^{t_2} l[\xi(t)] + \langle p(t), \dot{q}(t) - \mathscr{L}_\xi q(t) \rangle_{T^*M} dt = 0. \tag{2.3}$$

In order to write down the general solution to this action principle we need the following definition.

Definition 2.3 (Diamond operator)**.** Let $\mathscr{L}$ be a velocity map from V to M. The **Diamond operator** is an operator $\diamond : T^*M \to V^*$ which satisfies:

$$\langle p \diamond Q, \xi \rangle_V = -\langle p, \mathscr{L}_\xi Q \rangle_{T^*M},$$

where $\langle .,. \rangle_V$ is the duality pairing between V and V^*.

Now we can minimize the Clebsch variational principle by calculating the first variation and equating it to zero, just like we did when minimizing the Lagrange action principle.

$$\begin{aligned} 0 &= \delta \int_{t_1}^{t_2} l[\xi] + \langle p, \dot{q} - \mathscr{L}_\xi q \rangle_{T^*M} dt \\ &= \int_{t_1}^{t_2} \langle \frac{\delta l}{\delta \xi}, \delta\xi \rangle_V + \langle \delta p, \dot{q} - \mathscr{L}_\xi q \rangle_{T^*M} + \langle p, \delta\dot{q} - (T_q \mathscr{L})\delta q - \mathscr{L}_{\delta\xi} q \rangle_{T^*M} dt, \end{aligned}$$

where $\delta l = \langle \frac{\delta l}{\delta \xi}, \delta\xi \rangle_V$ and $T_q \mathscr{L}_\xi = \lim_{t \downarrow 0} \frac{1}{h}[\mathscr{L}_{(\xi + h\delta q)} - \mathscr{L}_\xi]$. Using integration by parts and the definition of the diamond operator we can rewrite this in the following way.

$$0 = \int_{t_1}^{t_2} \langle \frac{\delta l}{\delta \xi} + p \diamond q, \delta\xi \rangle_V + \langle \delta p, \dot{q} - \mathscr{L}_\xi q \rangle_{T^*M} + \langle -\dot{p} - (T_q \mathscr{L}_\xi)^T p, \delta q \rangle_{T^*M} dt,$$

here we also used that $\langle p, T_q \mathscr{L}_\xi \delta q \rangle = \langle (T_q \mathscr{L}_\xi)^T, \delta q \rangle$. The variations $\delta\xi, \delta p$ and δq are chosen arbitrarily, so the above equation is equivalent to the three following equations

$$\frac{\delta l}{\delta \xi} = -p \diamond q, \tag{2.4}$$

$$\dot{q} = \mathscr{L}_\xi q, \tag{2.5}$$

$$\dot{p} = -(T_q \mathscr{L}_\xi)^T p. \tag{2.6}$$

These equations are known as the *Clebsch equations*. The main result of [5] is that under certain conditions we can eliminate the equations for q and p and get an equation just for l. In the next section we will see how this is done.

2.1.2 Elimination of the Clebsch variables

To be able to eliminate the Clebsch variables we need the velocity map defined previously to induce a Lie bracket. We will first see how the velocity map can be used to define a bracket under a certain condition and then we will show that the resulting bracket is actually a Lie bracket. The condition which needs to be satisfied by the velocity map in order to eliminate the variables q and p is given in the following definition.

Definition 2.4 (Closed)**.** Let V be a vector space and $\mathscr{L}$ be a velocity map ($\mathscr{L} : V \times M \to TM$). The velocity map is called **closed** when for all $u, v \in V$ there is some $w \in V$ such that for all $q \in M$:

$$\mathscr{L}_w q = ((T_q\mathscr{L}_v)\mathscr{L}_u - (T_q\mathscr{L}_u)\mathscr{L}_v)q.$$

When this is the case $\mathscr{L}$ defines a bracket in V:

$$\mathscr{L}_{[u,v]} = (T_q\mathscr{L}_v)\mathscr{L}_u - (T_q\mathscr{L}_u)\mathscr{L}_v \tag{2.7}$$

Using the definition of a Lie bracket (definition 1.15) we will check that this bracket actually defines a Lie bracket. The skew-symmetry follows from the linearity of $\mathscr{L}$:

$$\mathscr{L}_{[u,v]} = (T_q\mathscr{L}_v)\mathscr{L}_u - (T_q\mathscr{L}_u)\mathscr{L}_v = -((T_q\mathscr{L}_u)\mathscr{L}_v - (T_q\mathscr{L}_v)\mathscr{L}_u) = -\mathscr{L}_{[v,u]} = \mathscr{L}_{-[v,u]},$$

so $[u,v] = -[v,u]$. The Jacobi identity is also satisfied, which we will verify next. Note first that $T_q(T_q\mathscr{L}_s) = 0$ since $\mathscr{L}$ is linear in V, and therefore

$$T_q(\mathscr{L}_{[r,s]}) = T_q((T_q\mathscr{L}_s)\mathscr{L}_r - (T_q\mathscr{L}_r)\mathscr{L}_s) = T_q\mathscr{L}_s T_q\mathscr{L}_r - T_q\mathscr{L}_r T_q\mathscr{L}_s = 0,$$

for all $r, s \in V$. The Jacobi identity now follows by the next calculation.

$$\begin{aligned}
\mathscr{L}_{[u,[v,w]]+[v,[w,u]]+[w,[u,v]]} =& \mathscr{L}_{[u,[v,w]]} + \mathscr{L}_{[v,[w,u]]} + \mathscr{L}_{[w,[u,v]]} \\
=& (T_q\mathscr{L}_u)\mathscr{L}_{[v,w]} - (T_q\mathscr{L}_{[v,w]})\mathscr{L}_u \\
& + (T_q\mathscr{L}_v)\mathscr{L}_{[w,u]} - (T_q\mathscr{L}_{[w,u]})\mathscr{L}_v \\
& + (T_q\mathscr{L}_w)\mathscr{L}_{[u,v]} - (T_q\mathscr{L}_{[u,v]})\mathscr{L}_w. \\
=& (T_q\mathscr{L}_u)\mathscr{L}_{[v,w]} + (T_q\mathscr{L}_v)\mathscr{L}_{[w,u]} \\
& + (T_q\mathscr{L}_w)\mathscr{L}_{[u,v]} \\
=& (T_q\mathscr{L}_u)(T_q\mathscr{L}_w)\mathscr{L}_v - (T_q\mathscr{L}_u)(T_q\mathscr{L}_v)\mathscr{L}_w \\
& + (T_q\mathscr{L}_v)(T_q\mathscr{L}_u)\mathscr{L}_w - (T_q\mathscr{L}_v)(T_q\mathscr{L}_w)\mathscr{L}_u \\
& + (T_q\mathscr{L}_w)(T_q\mathscr{L}_v)\mathscr{L}_u - (T_q\mathscr{L}_w)(T_q\mathscr{L}_u)\mathscr{L}_v \\
=& 0.
\end{aligned}$$

Therefore since $\mathscr{L}$ is linear we have $[u,[v,w]] + [v,[w,u]] + [w,[u,v]] = 0$. So the bracket $[.,.]$ satisfies the conditions from definition (1.15) and therefore defines a Lie bracket, and this bracket makes V into a Lie algebra.

As we will see next we can use the fact that the velocity map is closed to eliminate the Clebsch variables p and q; the following is theorem 1 from [5]. Take $w \in V$, then

$$\begin{aligned}
\frac{d}{dt}\langle\frac{\delta l}{\delta\xi}(\xi), w\rangle_V &= -\frac{d}{dt}\langle p \diamond q, w\rangle_V \\
&= \frac{d}{dt}\langle p, \mathscr{L}_w q\rangle_{T^*M} \\
&= \langle \dot{p}, \mathscr{L}_w q\rangle_{T^*M} + \langle p, (T_q\mathscr{L}_w)\dot{q}\rangle_{T^*M} \\
&= \langle -(T_q\mathscr{L}_\xi)^T p, \mathscr{L}_w q\rangle_{T^*M} + \langle p, (T_q\mathscr{L}_w)\mathscr{L}_\xi q\rangle_{T^*M} \\
&= \langle p, (-(T_q\mathscr{L}_\xi)\mathscr{L}_w + (T_q\mathscr{L}_w)\mathscr{L}_\xi)q\rangle_{T^*M} \\
&= \langle p, -\mathscr{L}_{\xi,w} q\rangle_{T^*M} \\
&= \langle p \diamond q, [\xi, w]\rangle_V \\
&= -\langle\frac{\delta l}{\delta\xi}(\xi), [\xi, w]\rangle_V.
\end{aligned}$$

Note that in [5] this is taken one step further by defining the ad- and ad$^*-$ operators to acquire an equation which is not in a weak formulation. For our purposes the weak equation is fine, so we have shown that the Clebsch equations (2.4) are equivalent to the following weak equation,

$$\frac{d}{dt}\langle\frac{\delta l}{\delta\xi}(\xi),w\rangle_V = -\langle\frac{\delta l}{\delta\xi}(\xi),[\xi,w]\rangle_V \quad \text{for all } w\in V, \tag{2.8}$$

whenever the velocity map $\mathscr{L}$ is closed in V. This equation is known as the *Euler-Poincaré* equation. Note that this is an equation on the dual of a Lie algebra, where in this case the Lie algebra is given by V together with the bracket defined by the velocity map. In Section 1.2.2 we have seen that in some cases this is equivalent to a Poisson system. In the next section we will exploit this fact to get a Poisson discretization for Burgers' equation.

2.2 Application to Burgers' equation

In this section we will first find the Euler-Poincaré equation corresponding to Burgers' equation by using the Clebsch method described previously in this chapter. We will then discretize this equation and use the theory from Section 1.2.2 to rewrite it as a Poisson system.

2.2.1 Euler-Poincaré equation

First we will derive the Euler-Poincaré equation corresponding to Burgers' equation. Recall that Burgers' equation was given by

$$\dot{u}+uu_x=0, \tag{2.9}$$

for $u:\mathbb{R}\to\mathbb{R}$, $x\in[0,L]$.

Let V be a vector space, and let the manifold M be the domain of u, so $M=[0,L]$. When we define the velocity map $\mathscr{L}_u$ such that it is closed with respect to the vector space V we know that the Euler-Poincaré equation will be given by

$$\frac{d}{dt}\langle\frac{\delta l}{\delta u}(u),w\rangle_V = -\langle\frac{\delta l}{\delta u}(u),[\xi,w]\rangle_V \quad \text{for all } w\in V,$$

The Lagrangian corresponding to Burgers' equation is given by $l[u]=\int_M \frac{u^2}{2}dx$. Substituting this into the above equation gives us the Euler-Poincaré equation,

$$\langle\dot{u},w\rangle_V = \langle u,[u,w]\rangle_V, \quad \text{for all } w\in V. \tag{2.10}$$

Recall from the introduction (6) that the Hamiltonian for Burgers' equation was defined as $H(u)=\frac{1}{2}\langle u,u\rangle_V$, the functional derivative of H is equal to

$$\langle\frac{\delta H}{\delta u},w\rangle_V = \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}[H(u+\varepsilon w)-H(u)] = \lim_{\varepsilon\to 0}[\langle u+\varepsilon w,u+\varepsilon w\rangle_V - \langle u,u\rangle_V] = \langle u,w\rangle_V.$$

We can substitute this into equation (2.10) to get the following equation.

$$\langle\dot{u},w\rangle_V = \langle u,[\frac{\delta H}{\delta u},w]\rangle_V. \tag{2.11}$$

Note that this differential equation is in the form of equation (1.25), in order to discretize this differential equation we will therefore make use of Theorem 1.16. First we will have to set up the Lie algebra V and its dual V^*, which we will do in the next section.

2.2.2 Discrete solution space

In the previous section we found the Euler-Poincaré equation associated to Burgers' equation for an arbitrary vector space V. In this section we will define a vector space V in which we can find an approximate solution to Burgers' equation. Along with this we will define a velocity map $\mathscr{L}_u$ in such a way that the closure condition from definition 2.4 is satisfied.

We will start by defining our vector space V, which we will do by discretizing the domain $[0,L]$. Define $x_i = i\Delta x$ for $\Delta x = L/n$, $i = 0,...,n$ and $n \in \mathbb{N}$, further define the intervals $I_i = [x_i, x_{i+1}]$. Using this discretization we will define V as follows,

$$V = \{v \in H^1(M) : v|_{I_i} \in P^1(I_i)\}, \tag{2.12}$$

where P^1 is the space of first degree polynomials. Note that V is just the space of piecewise linear functions. We can construct a basis for this space using the following functions

$$\phi_i^c(x) = \begin{cases} \frac{1}{\sqrt{\Delta x}} & x \in I_i, \\ 0 & x \notin I_i, \end{cases} \qquad \phi_i^l(x) = \begin{cases} \sqrt{\frac{12}{\Delta x^3}}(x - \frac{x_i + x_{i+1}}{2}) & x \in I_i, \\ 0 & x \notin I_i. \end{cases} \tag{2.13}$$

Note that $\{\phi_i^c, \phi_i^l\}_i$ form an orthonormal basis of V with respect to the L^2 inner product. Here, ϕ_i^c can be interpreted as the mean over a gridcell and ϕ_i^l the slope of the function on the gridcell.

Since the basis $\{\phi_i^c, \phi_i^l\}_i$ is orthonormal with respect to this inner product we can use it to define the dual basis $\{\varphi_i^c, \varphi_i^l\}_i$ for V^* in the following way,

$$\varphi_i^c(v) = \langle \phi_i^c, v \rangle, \quad \varphi_i^l(v) = \langle \phi_i^l, v \rangle,$$

for all $i = 1,..,n$. Here, we have denoted the L^2 inner product using $\langle .,. \rangle$ without subscript. If we now define the dual representation using the L^2 inner product we can identify V^* with V, so an element $v \in V^*$ is actually an element of V where the action of the element on any other element $w \in V$ is defined by the L^2 inner product between them. The dual basis for V^* will then be the same basis as V, given by $\{\phi_i^c, \phi_i^l\}_i$.

We will define the velocity map $\mathscr{L}_u : V \times M \to T_qM$ in the following way

$$\mathscr{L}_u(q) = u(q). \tag{2.14}$$

Using formula (2.7) we can find the associated bracket. For the tangent map $T_q\mathscr{L}_u$ we have the following

$$T_q\mathscr{L}_u(q) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} [\mathscr{L}_u(q + \varepsilon\delta q) - \mathscr{L}_u(q)] = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} [u(q + \varepsilon\delta q) - u(q)] = u_x(q).$$

Take $u, v \in V$, using the above expression we get

$$((T_q\mathscr{L}_v)\mathscr{L}_u - (T_q\mathscr{L}_u)\mathscr{L}_v)(q) = u(q)v_x(q) - u_x(q)v(q) = \mathscr{L}_{(uv_x - u_xv)}(q).$$

Which implies the bracket in this case is given by

$$[u, v] = uv_x - u_xv, \quad u, v \in V. \tag{2.15}$$

This bracket is the standard commutator of vector fields, which is the reason for defining the velocity map like we did.

We can check that V is closed with respect to this bracket by showing that the brackets of the basis functions can be written as linear combinations of the basis functions.

$$[\phi_i^c, \phi_j^l] = -\delta_{ij}\sqrt{\frac{12}{\Delta x^3}}\phi_i^c = -[\phi_j^l, \phi_i^c],$$

where δ denotes the Dirac delta function. So by Theorem 1 from [5] vector space V together with the Lie bracket given by $[.,.]$ define a Lie algebra. Here the structure constants corresponding to the Lie algebra are given by $C_{ij}^k = \pm\delta_{ij}\sqrt{\frac{12}{\Delta x^3}}$, they are only non-zero when taking the bracket of basis functions defined on the same interval.

2.2.3 Poisson formulation

In the last section we have set up the Lie algebra V, and its dual V^*. We will now apply Theorem 1.16 to get a Poisson system approximating Burgers' equation. We start again where we left off in section 2.2.1, with the differential equation on the dual of the Lie algebra V. Take $u \in V^*$, we write $u = \sum_i u_i \phi_i^c + v_i \phi_i^l$ for $\underline{u} = (u_1, .., u_n)^T, \underline{v} = (v_1, ..., v_n)^T \in \mathbb{R}^n$. The Hamiltonian H such that $H(\underline{u}, \underline{v}) = H^*(u)$ can be calculated in the following way,

$$\begin{aligned} H^*(u) = \frac{1}{2}\langle u, u\rangle &= \frac{1}{2}\sum_{i,j} u_i u_j \langle \phi_i^c, \phi_j^c\rangle + u_i v_j \langle \phi_i^c, \phi_j^l\rangle + v_i u_j \langle \phi_i^l, \phi_j^c\rangle + v_i v_j \langle \phi_i^l, \phi_j^l\rangle \\ &= \frac{1}{2}\sum_i u_i^2 + v_i^2 = H(\underline{u}, \underline{v}). \end{aligned} \tag{2.16}$$

Applying Theorem 1.16 we now get the following Poisson system equivalent to equation (2.11).

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & B_1(u) \\ -B_1(u) & 0 \end{pmatrix} \nabla H(u, v) = B(u) \nabla H(u, v), \tag{2.17}$$

where we have omitted the underlines from u and v, and $B_1(u)$ is given by

$$B_1(u) = \sqrt{\frac{12}{\Delta x^3}} \begin{pmatrix} u_1 & 0 & \dots & \dots & 0 \\ 0 & u_2 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & u_n \end{pmatrix} = C \begin{pmatrix} u_1 & 0 & \dots & \dots & 0 \\ 0 & u_2 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & u_n \end{pmatrix},$$

for $C = \sqrt{\frac{12}{\Delta x^3}}$. Note that this matrix depends only on the coefficients u since in the previous section we found that only the structure coefficients corresponding to the ϕ_i^l basis functions are nonzero. The theorem also guarantees that this system is a Poisson system. The main problem with this system is that it is decoupled into the grid cells. When we look at the individual equations for the coefficients we have

$$\dot{u}_i = \sqrt{\frac{12}{\Delta x^3}} u_i v_i, \quad \dot{v}_i = -\sqrt{\frac{12}{\Delta x^3}} u_i^2, \qquad i = 0, ..., n.$$

We can see using the interpretation we had for u as the mean of the cell and v as the slope in the cell that the equation for the cell mean is a good approximation of Burgers' equation. So we have a decoupled system which should behave like Burgers' equation in the grid cells. Since the vector space V we choose allows for discontinuous functions we can expect the solution to this system to be discontinuous. Therefore we would like to constrain this system to be continuous without losing the Poisson structure.

2.2.4 Constrained Poisson formulation

The way we will try to enforce the acquired Poisson system to be continuous is by again using Lagrange multipliers. We will first set up a couple of equations, called the *constraint equations*, which will vanish for continuous functions. Using these equations we will restrict the class of solutions to the manifold of continuous solutions. In order to enforce the constraints we can add the constraint equations to our Hamiltonian using Lagrange multipliers. In the basis given in (2.13) these equations are given by the following

$$g_i = u_{i+1} - u_i - \sqrt{3}(v_{i+1} + v_i), \qquad i = 1, ..., n.$$

Which can be written as

$$g(u, v) = E_1 u - E_2 v,$$

where

$$E_1 = \begin{pmatrix} -1 & 1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & 1 \\ 1 & 0 & \dots & \dots & 0 & -1 \end{pmatrix} \qquad E_2 = \sqrt{3}\begin{pmatrix} 1 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & 1 \\ 1 & 0 & \dots & \dots & 0 & 1 \end{pmatrix}$$

Using Lagrange multipliers λ these equations can now be added to the Hamiltonian,

$$H^*(u,v) = H(u,v) + \lambda^T g(u,v),$$

where $\lambda \in \mathbb{R}^n$. Giving us the following constrained system

$$\begin{aligned} \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} &= B(u)\nabla H^*(u,v), \\ g(u,v) &= 0. \end{aligned} \tag{2.18}$$

In the next chapter we will look at constrained Poisson systems more generally. We will see that we can preserve the Poisson structure by adding the constraint equations to our Hamiltonian. Afterwards we will look at ways to integrate such systems.

Chapter 3

Constrained Poisson systems

In the last chapter we found a Poisson system equivalent to Burgers' equation. This system was useless however since it was decoupled into the grid cells of the discretization. The way we want to solve this problem is by adding continuity constraints to our system. In this section we will look in general at constrained Poisson systems. In the case of Hamiltonian systems these have been studied quite extensively, see for example Leimkuhler & Reich (1994) [7] or Reich (1996) [8].

We will start by looking at why we can add the constraint equations to the Hamiltonian using Lagrange multipliers.

3.1 Constrained formulation

Consider the system given by some Poisson bracket $\{.,.\}$ and Hamiltonian $H : \mathbb{R}^n \to \mathbb{R}$

$$\frac{\partial}{\partial t}F(y) = \{F,H\}(y), \qquad y \in \mathbb{R}^n, \tag{3.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}$. We want to constrain this system using constraint functions $g : \mathbb{R}^n \to \mathbb{R}^m$. One way to enforce this constraint is to add it as a potential to the Hamiltonian

$$H^*(y) = H(y) + \frac{1}{2\varepsilon}\|g\|^2, \tag{3.2}$$

for some small positive number ε. Suppose we start with initial values $y_0 \in \mathbb{R}^n$ such that $g(y_0) = 0$, if we can then integrate system (3.1) using the augmented Hamiltonian (3.2) with an integrator which preserves the Hamiltonian we can choose ε such that the constraint is also approximately preserved. This procedure is known as the penalty method in [9]. Assume we have an integrator given by a one step map $y_{n+1} = \phi(y_n)$, if this integrator preserves the Hamiltonian and $g(y_n) = 0$ we have

$$H(y_n) = H^*(y_n) = H^*(y_{n+1}) = H(y_{n+1}) + \frac{1}{2\varepsilon}\|g(y_{n+1})\|^2 > \frac{1}{2\varepsilon}\|g(y_{n+1})\|^2.$$

Say we want to have $\|g(y_{n+1})\|^2 < \delta$ for some $\delta > 0$, then we can choose

$$\varepsilon = \frac{\delta}{2H(y_n)} = \frac{\delta}{2H(y_0)}.$$

We can try to immediately numerically integrate the system using this new Hamiltonian. However, a drawback of this is that if we want the constrained to be satisfied almost exactly the value for ε will be very small, which in turn makes $\frac{1}{2\varepsilon}$ big, in general leading to stiffness. We will now look at how to avoid this issue. Assume the Poisson bracket is given by $\{F,G\}(y) = \nabla F^T B(y)\nabla G$ for some matrix B

satisfying the conditions from lemma (1.9). If we now replace H with H^* from (3.2) into (3.1) and write out the bracket we get

$$\frac{\partial}{\partial t}F(y) = \{F,H^*\}(y) = \{F,H\} + \{F, \frac{1}{2\varepsilon}\|g\|^2\} = \nabla F^T B(y)(\nabla H + \frac{1}{\varepsilon}G^T(y)g(y)),$$

where $G(y) = g'(y)$. If we now set $\lambda = \frac{1}{\varepsilon}g(y) \in \mathbb{R}^n$ we get the system

$$\begin{aligned}\frac{\partial}{\partial t}F(y) &= \{F, H + \lambda^T g\}(y),\\ \lambda &= \frac{1}{\varepsilon}g(y).\end{aligned}$$

If we now take $\varepsilon \to 0$ we get the following system

$$\begin{aligned}\frac{\partial}{\partial t}F(y) &= \{F, H + \lambda^T g\}(y),\\ g(y) &= 0.\end{aligned} \tag{3.3}$$

So we can replace (3.2) with a Hamiltonian given by

$$H^* = H + \lambda^T g. \tag{3.4}$$

In the case of Hamiltonian systems derived from mechanical systems system (3.3) could be derived directly by adding the constraint to the Lagrangian using Lagrange multipliers. The reason for deriving it like we did in this section is that we can add constraints to the system after we have worked out the equations of motion corresponding to the Lagrangian.

In the remainder of this chapter we will see how we can handle constrained systems like (3.3). There are two ways to go about integrating such systems. One of the ways is by considering the Lagrange multipliers as extra variables, then solving the whole system. The other way is to find a way to express the Lagrange multipliers in terms of the other variables and substituting this into the equations. We will start by looking at the latter.

3.2 Index reduction

Systems such as (3.3) are known differential-algebraic equations. We will look at more general systems of this form first

$$\begin{aligned}\dot{y} &= f(y,\lambda),\\ 0 &= g(y),\end{aligned} \tag{3.5}$$

for $y \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^m$. One way to handle such systems is by replacing the constraint $g(y) = 0$ with some equivalent expression which can be solved for λ. If we then substitute this expression into the first equation from (3.5) we get an equivalent unconstrained formulation. For system (3.5) we can for example replace the constraint function with its derivative, because when for $t = 0$ we have $g(y(0)) = 0$ and $\dot{g} = 0$ we have $g(y(t)) = 0$ for all $t \geq 0$. We then get the following

$$0 = \frac{d}{dt}g(y) = G(y)^T\dot{y} = G(y)^T f(y,\lambda),$$

where again $G(y) = g'(y)$. If we can solve this equation for λ we will say that this system has **differentiation index** equal to 1. In general we will define:

Definition 3.1 (Differentiation index)**.** The **differentiation index** is the number of times we have to differentiate the constraint equation in (3.5) to get a system which is solvable for λ.

The procedure of solving for λ from some derivative of the constraint function and then substituting λ to get an unconstrained formulation is known as **index reduction**. The main issue with index reduction is what is known as the drift-off phenomenon [10]. Assume for example (3.5) is an index 1 problem, when we replace the constraint with its time derivative the resulting system will satisfy the following:

$$
\begin{aligned}
G(y(t))\dot{y}(t) &= G(y(0))\dot{y}(0), \\
g(y(t)) &= g(y(0)) + tG(y(0))\dot{y}(0).
\end{aligned}
$$

So if we get a linear error growth in the first equation this can lead to a quadratic growth in error for the original constraint g.

Next we will look at what index reduction looks like for Poisson systems. Consider again the system given in (3.3), using the Leibniz' rule as well as linearity of the Poisson bracket we get

$$\frac{\partial}{\partial t}F = \{F, H + \lambda^T g\} = \{F,H\} + \{F, \sum_{j=1}^{m} \lambda_j g_j\} = \{F,H\} + \sum_{j=1}^{m} (\{F, g_j\}\lambda_j + \{F, \lambda_j\} g_j).$$

Now since we are considering this differential equation together with $g = 0$, the second term in the sum vanishes resulting in

$$\frac{\partial}{\partial t}F = \{F,H\} + \sum_{j=1}^{m} \{F, g_j\}\lambda_j. \tag{3.6}$$

We now want to choose λ such that $g = 0$, but we again can't solve for λ using this condition. Instead we will again replace it with its derivative, $\dot{g} = 0$, which in this case is also referred to as checking the **consistency conditions**. We can do this by substituting $F = g_i$ in (3.6),

$$\dot{g}_i = \{g_i, H\} + \sum_{j=1}^{m} \{g_i, g_j\}\lambda_j,$$

for $1 \leq i \leq m$. Now we can solve for λ if the matrix $\{g,g\} := (\{g_i, g_j\})_{ij}$ is invertible. If we denote $\{g,H\} = (\{g_1,H\}, ..., \{g_m,H\})^T$ we get

$$\lambda = -\{g,g\}^{-1}\{g,H\}.$$

Substituting this into (3.6) we get the unconstrained formulation:

$$\frac{\partial}{\partial t}F = \{F,H\} - \{F,g\}\{g,g\}^{-1}\{g,H\}, \tag{3.7}$$

where $\{F,g\}$ is defined in the same way as $\{g,H\}$. The main question now is whether the matrix $\{g,g\}$ is invertible. If the matrix is not invertible we might have to replace the constraints with their derivatives until we do get an invertible matrix such that we can solve for the Lagrange multipliers. In the next section we will formalize this approach, which was first found by Dirac.

3.3 Dirac bracket approach

In the last section we left off with the unconstrained formulation in (3.7). As we mentioned we want to find a way to guarantee the invertibility of the matrix $\{g,g\}$. We will see in this section that we can do this by adding derivatives of our constraints to the set of constraints and then finding out which constraints actually matter for the dynamics. The system (3.7) can then be rewritten in the form of (3.1) using a bracket defined by

$$\{F,G\}_* = \{F,G\} + \{F,G\}\{g,g\}^{-1}\{g,H\}.$$

This bracket is known as the *Dirac bracket*, and we will see that the method by Dirac which we will outline in this section, guarantees that this bracket is actually a Poisson bracket.

3.3.1 Dirac bracket

We will start by going through the method by Dirac, which he gave a lecture series about at Yeshiva university, New York in 1964 [11].

Consider a Poisson bracket $\{.,.\}$ and Hamiltonian H defining a system of differential equations

$$\frac{\partial}{\partial t}F(y) = \{F, H^*\}(y), \tag{3.8}$$

together with constraints $g_i(y) = 0$, $i = 1, ..., m$ and the augmented Hamiltonian $H^* = H + \sum_{i=1}^{m} \lambda_i g_i$. The constraints $(g_i)_i$ are called the *primary constraints*. We want g_i to remain zero over time for all i so we again enforce the consistency conditions $\dot{g}_i = 0$. By substituting $F = g_i$ into (3.8) these conditions give rise to the following equations:

$$\{g_i, H\} + \sum_{j=1}^{m} \lambda_j \{g_i, g_j\} = 0, \qquad i = 1, ..., m. \tag{3.9}$$

For a given constraint g_i the above equation gives rise to three cases:

- **Case 1**
  If the Poisson bracket of g_i with the Hamiltonian as well as the Poisson brackets of g_i with all of the primary constraints vanish, i.e.

  $$\{g_i, H\} = 0, \qquad \{g_i, g_j\} = 0 \qquad \text{for all } j,$$

  then equation (3.9) reduces to $0 = 0$. The constraint is therefore already satisfied when all of the other primary constraints are satisfied. So this constraint vanishes on the manifold defined by the other constraints, and we can remove it.

- **Case 2**
  For the second case assume that only the Poisson brackets of g_i with the rest of the constraints vanishes

  $$\{g_i, g_j\} = 0 \qquad \text{for all } j.$$

  Then equation (3.9) reduces to $\{g_i, H\} = 0$, so the constraint gives rise to another constraint:

  $$\chi = \{g_i, H\},$$

  called a **secondary constraint**. These constraints have to be treated like the primary constraints, so we have to again set up consistency equations for the secondary constraints to see to which case they belong. It might happen that the secondary constraint again falls into the second case, leading to an additional secondary constraint.

- **Case 3**
  Constraints belonging to the third case do not reduce in the two above ways. Meaning that we do not have to work out secondary constraints in this case because we do not need them to solve for λ in the end.

After going through these cases for all of the constraints g_i we end up with a bunch of primary constraints and corresponding secondary constraints. Note that this procedure is similar to index reduction, but instead of replacing constraints with their time derivatives we are adding them to the total constraints, the goal will now also be to get an expression for λ using the resulting system.

Since the secondary constraints have to be treated the same as primary constraints we will denote them using the same notation $\{g_j : j = m+1,...,m+\tilde{m}\}$, where $\tilde{m}$ is the total number of secondary constraints, we will set $M = m+\tilde{m}$. For these constraints we get the following consistency conditions

$$\{g_j, H\} + \lambda_k \{g_j, g_k\} = 0, \tag{3.10}$$

where $j = 1,...,M$ and $k = 1,...,m$. Which can be written as a linear system for λ

$$\begin{bmatrix} 0 & \{g_1,g_2\} & \cdots & \cdots & \{g_1,g_m\} \\ \{g_2,g_1\} & 0 & \{g_2,g_3\} & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \{g_{M-1},g_m\} \\ \{g_M,g_1\} & \cdots & \cdots & \{g_M,g_{m-1}\} & 0 \end{bmatrix} \lambda = \begin{bmatrix} \{g_1,H\} \\ \{g_2,H\} \\ \vdots \\ \vdots \\ \{g_M,H\} \end{bmatrix}. \tag{3.11}$$

We will again denote the matrix of Poisson brackets of constraints by $\{g,g\}$ and the vector on the left hand side of the above equation by $\{g,H\}$. We will assume this system has a solution, denote this solution by $\lambda = \Lambda(y)$. To any solution of (3.11) we can add a solution to the homogeneous system, given by $\{g,g\}\lambda = 0$. We can write all independent solutions to the homogeneous system as V_j for $j = 1,...,J$. Then the solution to system (3.11) can be written as

$$\lambda = \Lambda(y) + \sum_j v_j V_j,$$

where the v_j's are arbitrary constants. This gives the following expression for the Hamiltonian

$$H^* = H + \lambda^T g = H + \sum_i \Lambda_i g_i + \sum_j v_j V_{ji} g_i,$$

where V_{ji} denote the components of V_j. We can write $\tilde{g}_j = \sum_i V_{ji} g_i$,

$$H^* = H + \lambda^T g = H + \sum_i \Lambda_i g_i + \sum_j v_j \tilde{g}_j. \tag{3.12}$$

Now that we have collected all the primary and secondary constraints we want to see which constraints matter for the dynamics. In order to do this we need the following definition dividing the constraints into two separate classes.

Definition 3.2 (First and second class functions)**.** A function F is called **first class** if the Poisson bracket of the function with all of the constraints vanishes, i.e.

$$\{F, g_i\} = 0, \qquad \text{for all } i.$$

If this is not the case the function is called **second class**.

All the constraints given by $\tilde{g}_j$ are first class, for example, because the Poisson bracket of these constraints with other constraints g_k is given by

$$\{\tilde{g}_j, g_k\} = \sum_i V_{ji} \{g_i, g_k\} = 0,$$

since V_j was defined to be in the kernel of the right hand side matrix of (3.11).

Once we have our set of constraints we can take a linear combination of them which defines a new constraint:

$$\chi(y) = \sum_{i=1}^{M} b_i g_i + \sum_{j=1}^{J} \tilde{b}_j \tilde{g}_j.$$

Replacing one of the constraints with χ will not change the constraints. We can now replace some of our constraints with linear combinations of the constraints in such a way that as many of them will become first class functions. Assuming we have done this we have the following theorem.

Theorem 3.3. *Let $\{g_i : 1 \leq i \leq m\}$ be a set of constraints. Assume that we have already taken linear combinations of the constraints in such a way that as many of them as possible have been brought into the first class. Then the following matrix is invertible:*

$$\begin{bmatrix} 0 & \{g_1,g_2\} & \cdots & \cdots & \{g_1,g_m\} \\ \{g_2,g_1\} & 0 & \{g_2,g_3\} & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \{g_{m-1},g_m\} \\ \{g_m,g_1\} & \cdots & \cdots & \{g_m,g_{m-1}\} & 0 \end{bmatrix}. \tag{3.13}$$

The proof of this theorem can be found in [11]. The basic idea is that when (3.13) is not invertible it will have some rank $T < m$, we can then set up the following determinant:

$$\text{A=}\begin{vmatrix} g_1 & 0 & \{g_1,g_2\} & 0 & \cdots & \{g_1,g_{T+1}\} \\ g_2 & \{g_2,g_1\} & 0 & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \{g_T,g_{T+1}\} \\ g_{T+1} & \{g_{T+1},g_1\} & \cdots & \cdots & \{g_{T+1},g_T\} & 0 \end{vmatrix}.$$

It can then be shown that A is first class, and since it is a linear combination of the constraints this contradicts our assumption that as many constraints have been replaced with first class constraints. We can also use this determinant A to find the linear combinations of constraints which are first class, so we can keep setting up the determinant until we end up with an invertible matrix.

Now that we have found a way to set up our constraints in such a way that the matrix $\{g,g\}$ from (3.7) is non-singular we can use it to define a new bracket.

Definition 3.4 (Dirac bracket)**.** For a bracket $\{.,.\}$ and a set of linearly independent constraints $\{g_i : 1 \leq i \leq m\}$ the **Dirac bracket** is defined by the following

$$\{\xi,\eta\}_* = \{\xi,\eta\} - \{\xi,g\}\{g,g\}^{-1}\{g,\eta\}.$$

This bracket defines a Poisson bracket, the proof for this can be found in [12]. Using this fact we have that the constrained Poisson system in (3.8) is equivalent to the Poisson system

$$\frac{\partial}{\partial t}F = \{F,H\}_* = \{F,H\} - \{F,g\}\{g,g\}^{-1}\{g,H\}. \tag{3.14}$$

This new Poisson structure is defined on the *constraint manifold*, which is defined in terms of the constraint functions in the following way

$$\mathscr{M} := \{y \in \mathbb{R}^n : g_i(y) = 0, \text{ for } i = 1,...,n\}. \tag{3.15}$$

In the next section we will look at a way to rewrite the bracket in such a way that we get a reduced number of equations just on this manifold. Then after that we will see how we can directly project onto this manifold.

3.4 Reduction to the constraint manifold

Like we mentioned at the end of the last section we can rewrite the Dirac bracket to get a reduced system defined on the constraint manifold. This can only be done for a specific type of constraint, which we will refer to as *slaving constraints*. We will start at the Poisson system given by the Dirac bracket (3.14) and rewrite it in terms of gradients restricted to the manifold. Since we are just rewriting a Poisson bracket, this will result in a Poisson system. This way of reduction to the constraint manifold is due to Vanneste & Bokhove [13].

3.4.1 Description of the method for slaving constraints

Let $u \in \mathbb{R}^m, v \in \mathbb{R}^r$ be variables and let $m+r=n$, consider the following Poisson system

$$\frac{d}{dt}F = \{F,H\}_* := \{F,H\} + \sum_{i=1}^{m} \lambda_i \{F, g_i\}, \tag{3.16}$$

where $\{g_i : 1 \leq i \leq m\}$ are the constraints, and $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers. Let the Poisson bracket $\{.,.\}$ be given by

$$\begin{aligned} \{F,G\} &= \nabla_u F B^{uu}(u,v) \nabla_u G + \nabla_u F B^{uv}(u,v) \nabla_v G + \nabla_v F B^{vu}(u,v) \nabla_u G, \\ &= \begin{pmatrix} \nabla_u F & \nabla_v F \end{pmatrix} \begin{pmatrix} B^{uu}(u,v) & B^{uv}(u,v) \\ B^{vu}(u,v) & 0 \end{pmatrix} \begin{pmatrix} \nabla_u G \\ \nabla_v G \end{pmatrix}, \end{aligned} \tag{3.17}$$

where $B^{uu}(u,v) \in \mathbb{R}^{r \times r}, B^{uv}(u,v) \in \mathbb{R}^{r \times s}$ and $B^{vu} = -(B^{uv})^T$. Note that there is no interaction between the gradients with respect to v, this is important for this procedure to work. We also do not want the v_i's to be dependent on each other through the constraints, so we will only look at *slaving constraints*. These are given by

$$g_j(u,v) = u_j - P_j(v), \qquad 0 \leq j \leq m, \tag{3.18}$$

where P_j are given possibly nonlinear functions of v. On the manifold defined by these constraints the values of the u_j's are uniquely determined by the values of the v_j's. We assume all primary and secondary constraints acquired from the theory of Dirac are in this form. The consistency conditions for these constraints are given by

$$0 = \frac{d}{dt} g_j = \{g_j, H\} + \sum_{i=1}^{m} \lambda_i \{g_j, g_i\}, \qquad 0 \leq j \leq m.$$

We can rewrite this to get

$$\sum_{i=1}^{m} \lambda_i \{g_j, g_i\} = -\{g_j, H\}, \qquad 0 \leq j \leq m. \tag{3.19}$$

Note that this equation is solvable for λ if the matrix $\{g,g\} = (\{g_i, g_j\})_{ij}$ is invertible, which is guaranteed if we have gone through the procedure of Dirac outlined in Section 3.3.1. Using the notation $\{g,g\}$ we can rewrite (3.19) as

$$\{g,g\}\lambda = -\{g,H\}, \tag{3.20}$$

where $\{g,H\} = (\{g_1,H\},...,\{g_m,H\})^T$.

On the manifold defined by the constraints we have $u_j = P_j(v)$ so the derivative of a function F restricted to the constraint manifold, which we will denote by $\frac{\partial F}{\partial v_j}|_c$, is given by

$$\left.\frac{\partial F}{\partial v_j}\right|_c = \frac{\partial F}{\partial v_j} + \sum_{i=1}^{m} \frac{\partial F}{\partial u_i} \frac{\partial P_i}{\partial v_j} = \frac{\partial F}{\partial v_j} - \sum_{i=1}^{m} \frac{\partial F}{\partial u_i} \frac{\partial g_i}{\partial v_j}. \tag{3.21}$$

Note that this is just a consequence of the chain rule, and the derivative with respect to u restricted to the constraint manifold coincides with the usual derivative with respect to u. We can then define the gradient restricted to the manifold as

$$\nabla_v F|_c = (\left.\frac{\partial F}{\partial v_r}\right|_c, ..., \left.\frac{\partial F}{\partial v_j}\right|_c)^T.$$

The following theorem now gives the equivalent system to (3.16) in terms of the restricted gradients. The theorem along with its proof is based on the procedure in [13].

Theorem 3.5. *Assume the matrix $\{g,g\}$ is invertible, such that (3.19) is solvable for λ, and denote the inverse by $\{g,g\}^{-1}$. Then the Poisson system (3.16) can be rewritten in terms of gradients on the constraint manifold in the following way.*

$$\frac{d}{dt}F = \{F,H\}_* = -(\nabla_v F|_c)^T B^{vu}\{g,g\}^{-1}B^{uv}(\nabla_v H|_c). \tag{3.22}$$

Proof. We start by rewriting the right hand side of (3.16); define $\{F,g\} = (\{F,g_1\},...,\{F,g_m\})$ and $\nabla_u g = (\nabla_u g_1|...|\nabla_u g_m)$.

$$\begin{aligned}\{F,H\}_* &= \{F,H\} + \{F,g\}\lambda \\ &= \{F,H\} + (\nabla_u F^T B^{uu}\nabla_u g + \nabla + \nabla_u F^T B^{uv}\nabla_v g + \nabla_v F^T B^{vu}\nabla_u g)\lambda.\end{aligned}$$

Now using (3.21) we get

$$\{F,H\}_* = \{F,H\} + (\nabla_u F^T B^{uu}\nabla_u g + \nabla_u F^T B^{uv}\nabla_v g + (\nabla_v F|_c)B^{vu}\nabla_u g + \nabla_u F^T \nabla_v g^T B^{vu}\nabla_u g)\lambda.$$

Note that $\nabla_u g = I$, so we have

$$\begin{aligned}\{F,H\}_* &= \{F,H\} + \nabla_u F^T(\nabla_u g^T B^{uu}\nabla_u g + \nabla_u g^T B^{uv}\nabla_v g + \nabla_v g^T B^{vu}\nabla_u g)\lambda + (\nabla_v F|_c)^T B^{vu}\lambda \\ &= \{F,H\} + \nabla_u F^T\{g,g\}\lambda + (\nabla_v F|_c)^T B^{vu}\lambda \\ &= \{F,H\} - \nabla_u F^T\{g,H\} + (\nabla_v F|_c)^T B^{vu}\lambda,\end{aligned}$$

where in the last equality we used equation (3.20). Now we can introduce expressions for $\{F,H\}$ and $\{g,H\}$.

$$\begin{aligned}\{F,H\}_* =&(\nabla_u F^T B^{uu}\nabla_u H + \nabla_u F^T B^{uv}\nabla_v H + \nabla_v F^T B^{vu}\nabla_u H) \\ &- \nabla_u F^T(\nabla_u g^T B^{uu}\nabla_u H + \nabla_u g^T B^{uv}\nabla_v H + \nabla_v g^T B^{vu}\nabla_u H) + (\nabla_v F|_c)^T B^{vu}\lambda \\ =&(\nabla_u F^T B^{uu}\nabla_u H + \nabla_u F^T B^{uv}\nabla_v H + \nabla_v F^T B^{vu}\nabla_u H) \\ &- \nabla_u F^T B^{uu}\nabla_u H - \nabla_u F^T B^{uv}\nabla_v H - \nabla_u F^T\nabla_v g^T B^{vu}\nabla_u H + (\nabla_v F|_c)^T B^{vu}\lambda \\ =&\nabla_v F^T B^{vu}\nabla_u H - \nabla_u F^T\nabla_v g^T B^{vu}\nabla_u H + (\nabla_v F|_c)^T B^{vu}\lambda \\ =&(\nabla_v F|_c)^T B^{vu}(\nabla_u H + \lambda).\end{aligned}$$

For $\nabla_u H + \lambda$ we have the following

$$\begin{aligned}\{g,g\}(\nabla_u H + \lambda) =&(\nabla_u g^T B^{uu}\nabla_u g + \nabla_u g^T B^{uv}\nabla_v g + \nabla_v g^T B^{vu}\nabla_u g)\nabla_u H + \{g,g\}\lambda \\ =&(B^{uu} + B^{uv}\nabla_v g + \nabla_v g^t B^{vu})\nabla_u H - \{g,H\} \\ =&(B^{uu} + B^{uv}\nabla_v g + \nabla_v g^t B^{vu})\nabla_u H \\ &- (\nabla_u g^T B^{uu}\nabla_u H + \nabla_u g^T B^{uv}\nabla_v H + \nabla_v g^T B^{vu}\nabla_u H) \\ =&(B^{uu} + B^{uv}\nabla_v g + \nabla_v g^t B^{vu})\nabla_u H - (B^{uu}\nabla_u H + B^{uv}\nabla_v H + \nabla_v g^T B^{vu}\nabla_u H) \\ =&B^{uv}\nabla_v g\nabla_u H - B^{uv}\nabla_v H \\ =&-B^{uv}(\nabla_v H - \nabla_v g\nabla_u H) \\ =&-B^{uv}\,(\nabla_v H|_c).\end{aligned}$$

Substituting this into the equation for $\{F,H\}$ we arrive at the desired result

$$\{F,H\}_* = -(\nabla_v F|_c)^T B^{vu}\{g,g\}^{-1}B^{uv}(\nabla_v H|_c).$$

□

This theorem gives a way to rewrite constrained Poisson systems on the constraint manifold. Furthermore because we just rewrote the Dirac bracket, the resulting system is guaranteed to be a Poisson system.

3.4.2 Dirac coordinates

In this section we will look at a special type of coordinates, where we replace some of our variables with the constraint functions. The remaining coordinates will then be local coordinates on the constraint manifold. We will then look when we can apply the theory from the last section. Lastly we will look at an example.

Consider again a Poisson system given by (3.16), but let the bracket now be given by

$$\{F,G\} = \begin{pmatrix}\nabla_u F & \nabla_v F\end{pmatrix}\begin{pmatrix}B^{uu}(u,v) & B^{uv}(u,v)\\ B^{vu}(u,v) & B^{vv}(u,v)\end{pmatrix}\begin{pmatrix}\nabla_u G\\ \nabla_v G\end{pmatrix}$$

for some matrix valued function $B^{vv} \in \mathbb{R}^{r\times r}$. We now consider constraints $g : \mathbb{R}^n \to \mathbb{R}^m$ simple enough such that there exists a function $h : \mathbb{R}^n \to \mathbb{R}^r$ such that the following change of coordinates is invertible

$$\begin{pmatrix}u\\ v\end{pmatrix} \to \begin{pmatrix}\tilde{u}\\ \tilde{v}\end{pmatrix} = \begin{pmatrix}g(u,v)\\ h(u,v)\end{pmatrix}. \tag{3.23}$$

Here we have replaced some of our coordinates with the constraints g_i, so the rest given by $\tilde{v}_i$ now provide local coordinates on the constraint manifold. This is useful because if we now do reduction to the constraint manifold the evolution equations for $\tilde{u}$ will be given by $\dot{\tilde{u}} = 0$. We will call these coordinates *Dirac coordinates* as is done by Marsden & Ratiu [14].

The Jacobian of the change of coordinates (3.23) is then given by

$$(\frac{\partial(\tilde{u},\tilde{v})}{\partial(u,v)}) = \begin{pmatrix}\frac{\partial g}{\partial u} & \frac{\partial g}{\partial v}\\ \frac{\partial h}{\partial u} & \frac{\partial h}{\partial v}\end{pmatrix}.$$

Using the change of coordinates formula in (1.14) we get the following structure matrix in the new coordinates

$$\begin{pmatrix}\frac{\partial g}{\partial u} & \frac{\partial g}{\partial v}\\ \frac{\partial h}{\partial u} & \frac{\partial h}{\partial v}\end{pmatrix}\begin{pmatrix}B^{uu} & B^{uv}\\ B^{vu} & B^{vv}\end{pmatrix}\begin{pmatrix}(\frac{\partial g}{\partial u})^T & (\frac{\partial h}{\partial v})^T\\ (\frac{\partial g}{\partial u})^T & (\frac{\partial h}{\partial v})^T\end{pmatrix} = \begin{pmatrix}-\{g,g\}^T & \{h,g\}\\ -\{h,g\}^T & -\{h,h\}^T\end{pmatrix}.$$

If we have $\{h,h\} = 0$ we are in the situation of the previous section where our constraints are given by (3.18) with $P_j(v) = 0$ for all $0 \leq j \leq m$. The constraint in the new coordinates will just be given by $0 = g(\tilde{u}) = \tilde{u}$. Let K be the Hamiltonian in the new coordinates, then we have the following expression for the constrained system

$$\begin{pmatrix}\dot{\tilde{u}}\\ \dot{\tilde{v}}\end{pmatrix} = \begin{pmatrix}-\{g,g\}^T & \{h,g\}\\ -\{h,g\}^T & 0\end{pmatrix}\begin{pmatrix}\nabla_{\tilde{u}}K - \lambda\\ \nabla_{\tilde{v}}K\end{pmatrix}.$$

The consistency equations are now just $\dot{\tilde{u}} = 0$ so we can immediately solve for λ

$$\lambda = \{g,g\}^{-T}\{h,g\}\nabla_{\tilde{v}}K - \nabla_{\tilde{u}}K.$$

Substituting this into the equation for $\tilde{v}$ we get

$$\dot{\tilde{v}} = -\{h,g\}^T\{g,g\}^{-1}\{g,h\}\nabla_{\tilde{v}}K. \tag{3.24}$$

Note that when integrating this system numerically we should no longer have to worry about the drift-off phenomenon we mentioned in Section 3.2, because we now have a system in local coordinates on the constraint manifold.

Next we will look at how to use Dirac coordinates in an example.

Example 3.6. We will again look at the Lorenz 86 system. In this example we will see how to integrate the system with two linear constraints using the method described in this section. Later we will see a projection method which allows the integration of this system with one constraint. Due to the condition on h in the coordinate change to Dirac coordinates we will limit ourselves to constraints of the form

$$g_1(x) = x_3 + a_1x_1 + a_2x_2 + a_3x_4, \quad g_2(x) = x_5 + c_1x_1 + c_2x_2 + c_3x_4.$$

This is because in this case we can define $h(x) = (x_1, x_2, x_4)^T$, and in that way guarantee that $\{h, h\} = 0$. The coordinate change will then be

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \to \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \end{pmatrix} = \begin{pmatrix} g_1(x) \\ g_2(x) \\ x_1 \\ x_2 \\ x_4 \end{pmatrix}.$$

The matrix $\{g, g\}$ can now be calculated to be

$$\begin{aligned}\{g,g\} &= \begin{pmatrix} 0 & (a_1b+c_1)x_2 - a_2bx_1 - \frac{a_3}{\varepsilon} \\ \frac{a_3}{\varepsilon} - (a_1b+c_1)x_2 + a_2bx_1 & 0 \end{pmatrix}, \\ &= \begin{pmatrix} 0 & (a_1b+c_1)\tilde{x}_4 - a_2b\tilde{x}_3 - \frac{a_3}{\varepsilon} \\ \frac{a_3}{\varepsilon} - (a_1b+c_1)\tilde{x}_4 + a_2b\tilde{x}_3 & 0 \end{pmatrix}.\end{aligned}$$

In order for this to be invertible we need to have $p(\tilde{x}) = (a_1b+c_1)\tilde{x}_4 - a_2b\tilde{x}_3 - \frac{a_3}{\varepsilon} \neq 0$. We can calculate $\{g, h\}$ and then substitute this together with the expression for $\{g, g\}$ into (3.24), which gives the following expression for the reduced system.

$$\begin{pmatrix} \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \\ \dot{\tilde{x}}_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{1}{\varepsilon}\tilde{x}_4p(\tilde{x}) \\ 0 & 0 & -\frac{1}{\varepsilon}\tilde{x}_3p(\tilde{x}) \\ -\frac{1}{\varepsilon}\tilde{x}_4p(\tilde{x}) & \frac{1}{\varepsilon}\tilde{x}_3p(\tilde{x}) & 0 \end{pmatrix} \nabla K(\tilde{x}),$$

where K is defined such that $K(\tilde{x}) = H(x)$. △

3.5 Projection onto the constraint manifold

Another way of dealing with the constraints is by projecting the differential equation onto the constraint manifold. In this section we will look at how we can do this.

Consider the following Poisson system with constraints

$$\begin{aligned}\dot{y} &= B(y)(\nabla H(y) + \lambda^T \nabla g(y)), \\ g(y) &= 0,\end{aligned} \tag{3.25}$$

where $B(y) \in \mathbb{R}^{n\times n}$ is the structure matrix, $H : \mathbb{R}^n \to \mathbb{R}$ is the Hamiltonian and $g : \mathbb{R}^n \to \mathbb{R}^m$ is some function defining the constraints. As we have seen the constraint functions define the following manifold

$$\mathscr{M} = \{y \in \mathbb{R}^n : g(y) = 0\}. \tag{3.26}$$

If we can find charts for this manifold we can use them to express our differential equation in local coordinates. If we can then solve this equation we can again use the chart to project back and we will have a solution to the original differential equation. We do have to be careful about how we do the projection, since we want the system to remain a Poisson system.

Let $\chi : \mathbb{R}^{(n-m)} \to \mathscr{M}$ be a chart for $\mathscr{M}$ and suppose $x \in \mathbb{R}^{(n-m)}$ provide local coordinates on $\mathscr{M}$ via this chart, meaning $y = \chi(x)$. Then we have

$$\dot{y} = X(x)\dot{x} = B(\chi(x))(\nabla H(\chi(x)) + \lambda^T \nabla g(\chi(x))) = B(\chi(x))\nabla H(\chi(x)),$$

where $X(x) = \chi'(x)$, note that the term involving g vanishes since $\chi(y) \in \mathscr{M}$. We can immediately see that this is not of the form of a Poisson system. If $X(x)$ has a left inverse $X(x)^{-1}$ then we could write it in the form of (3.25)

$$\dot{x} = \tilde{B}(x)\nabla K(x),$$

where $\tilde{B} = X(x)^{-1}B(x)X(x)^{-T}$, $K(x) = H(\chi(x))$. Note that we approximate $X(x)^{-T}X(x)^T$ with the identity here. The new structure matrix $\tilde{B}$ does not necessarily satisfy the rules from Lemma 1.9, so we will not get a Poisson system in this way in general.

If we instead assume the structure matrix is invertible, i.e. $B(y) \in \mathrm{GL}(n,\mathbb{R})$, we can invert B and look at the reduction of our equation to the manifold

$$X(x)^T B(\chi(x))^{-1} X(x)\dot{x} = X(x)^T(\nabla H(\chi(x)) + \lambda^T g(\chi(x))) = X(x)^T \nabla H(\chi(x)).$$

In order to get it into the form of a Poisson system again we now need the matrix $X(x)^T B(\chi(x))^{-1} X(x)$ to be invertible. For our manifold $\mathscr{M}$ this is equivalent to the invertibility of $g'(y)B(y)g'(y)^T$ which is in turn equivalent to the matrix of Poisson brackets of the components of g being invertible

$$(\{g_i, g_j\})_{ij} \in \mathrm{GL}(n,\mathbb{R}).$$

This happens to be the same condition we had when constructing the Dirac bracket. The resulting system will be

$$\dot{x} = \tilde{B}(x)\nabla K(x), \tag{3.27}$$

where $K(x) = H(\chi(x))$ and

$$\tilde{B}(x) = (X(x)^T B(\chi(x))^{-1} X(x))^{-1}. \tag{3.28}$$

We now want to know if the new structure matrix $\tilde{B}$ is a Poisson matrix. In the case of Hamiltonian systems, i.e. $B(x) = J^{-1}$, we have the following theorem from [1],

Theorem 3.7. *Consider system (3.25) where $B(y) = J^{-1}$ for all y. If the matrix of canonical Poisson brackets of the constraint functions $\{g_i, g_j\}(y)$ is invertible for all y then the matrix given in (3.28) defines a Poisson structure on the manifold defined by the constraints.*

The proof to this theorem can be found in [1]. The theorem also holds for Poisson systems with invertible structure matrix because a full rank Darboux form exists.

3.6 Integration of the constrained formulation

In section 3.2 we have seen that when we replace constraints with their derivative in order to find an expression for the Lagrange multipliers this will not guarantee that the numerical solution stays on the constraint manifold. We also cannot just project the solution back onto the manifold afterwards, since this in general will not yield a Poisson integrator. In this section we will look at a method due to Reich (1996) [8], which was developed for Hamiltonian systems with a particular type of constraints. The method works by splitting the system into two projections plus the original unconstrained system, which in this case will result in a symplectic constraint-preserving algorithm. We will first outline this method and then look to what extend the results of the paper apply to constrained Poisson systems.

3.6.1 Reich's method

Consider again the Hamiltonian system given in (1.8)

$$\dot{q} = \frac{\partial H}{\partial p},$$
$$\dot{p} = -\frac{\partial H}{\partial q},$$

where $p, q \in \mathbb{R}^n$. We constrain the dynamics of this system by enforcing $g(q) = 0$ for some function $g : \mathbb{R}^n \to \mathbb{R}^m$. Such constraints are known as *holonomic constraints*, in this case this means the constraint is only on the position variable q.

With the constraint the system now turns into

$$\begin{aligned} \dot{q} &= \frac{\partial H}{\partial p}, \\ \dot{p} &= -\frac{\partial H}{\partial q} - G(q)^T\lambda, \\ 0 &= g(q), \end{aligned} \tag{3.29}$$

where $G(q) = g'(q)$. The Hamiltonian of this system is given by

$$H^*(p,q) = H(p,q) + \lambda^T g(q). \tag{3.30}$$

The main idea now is to split the Hamiltonian, $H^* = H_1^* + H_2^*$, into the constraint $H_1^* = \lambda^T g(q)$ and the original Hamiltonian $H_2^* = H(p,q)$. The scheme can now be partitioned into the three following steps. The first step will ensure that the final outcome of the recursion step is on the manifold, the second step will go through the flow of the unconstrained problem, and the third step will ensure that the next step is in the tangent space of the constraint manifold. For the third step we also have to consider the time derivative of g, given by

$$f(q,p) = \dot{g}(q) = \{g, H\}(q,p),$$

where $\{.,.\}$ denotes the canonical Poisson bracket. We will now look at the three steps defining the method.

1. The flow corresponding to H_1^* can be approximated by numerically solving the following system

$$\begin{aligned} \dot{q} &= 0, \\ \dot{p} &= -G(q)^T\lambda. \end{aligned} \tag{3.31}$$

   Since we want the resulting scheme to be symplectic we want to solve this system symplectically. Since we are working with a Hamiltonian system here we can just use the symplectic Euler method (1.5)

$$\begin{aligned} \tilde{q}_k &= q_k, \\ \tilde{p}_k &= p_k - (\Delta t/2)G(q_k)^T\lambda_k, \end{aligned} \tag{3.32}$$

   where h is the time step. Here we choose the method to be implicit in q and explicit in p since this results in an explicit scheme in this case, note that $\tilde{q}_k = q_k$ so $G(q_k)^T\lambda(q_k, p_k) = G(\tilde{q}_k)^T\lambda(\tilde{q}_k, p_k)$.

2. Denote by Ψ any symplectic integrator for the unconstrained system corresponding to $\tilde{H}_2$, the second step is then given by

$$\begin{pmatrix} \tilde{q}_{k+1} \\ \tilde{p}_{k+1} \end{pmatrix} = \Psi_{\Delta t}(\begin{pmatrix} \tilde{q}_k \\ \tilde{p}_k \end{pmatrix}) \tag{3.33}$$

3. For the third step we repeat step 1, now using different Lagrange multipliers given by Λ.

$$\begin{aligned} q_{k+1} &= \tilde{q}_{k+1}, \\ p_{k+1} &= \tilde{p}_{k+1} - (\Delta t/2) G(\tilde{q}_{k+1})^T \Lambda_k. \end{aligned} \tag{3.34}$$

After this we have to choose λ_k such that $g(\tilde{q}_{k+1}) = 0$ and Λ_k such that $f(q_{k+1}, p_{k+1}) = 0$.

Remark 3.8. 1. In the first step the value for q does not change, so the constraint is still identically satisfied.

2. The third step is not really necessary, since in the next iteration we first project onto the constraint manifold again anyway. It is, however, useful as postprocessing to ensure $p \in T_qM$. For a comparison of the method with and without the third step see [15].

3. Note that in the first step λ is a function of p and q. Since we are using symplectic Euler this doesn't matter since we do not need an explicit expression for λ. If we try to solve this step explicitly this might matter.

3.6.2 Poisson systems

Now we will look at general Poisson structures with general constraints

$$\begin{aligned} \dot{x} &= B(x)(\nabla H(x) + G(x)^T \lambda(x)), \\ 0 &= g(x). \end{aligned}$$

In this section we will look at how we can possibly use the idea in the previous section for these more general systems.

The first problem we will have to deal with is that we can no longer use symplectic Euler in the first step. In the case of general Poisson systems we could try explicitly solving the first step. This of course restricts us to quite simple constraints since it has to be solvable, but we also run into the problem that we do not know how λ depends on p and q. To solve this we can treat λ as another variable, like is done in [7]. The second problem is that it is no longer guaranteed that during the first step of the method the flow stays on the constraint manifold. Next we will look at an example for which Reich's method does work.

Example 3.9. Recall the Lorenz 86 system from example (1.13), given by

$$\dot{x} = \begin{pmatrix} 0 & 0 & -x_2 & 0 & bx_2 \\ 0 & 0 & x_1 & 0 & -bx_1 \\ x_2 & -x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\varepsilon} \\ -bx_2 & bx_1 & 0 & \frac{1}{\varepsilon} & 0 \end{pmatrix} \nabla H(x),$$

for $H(x) = \frac{1}{2}(x_1^2 + 2x_2^2 + x_3^2 + x_4^2 + x_5^2)$. We will consider a single linear constraint, for simplicity we will take

$$g(x_4, x_5) = x_4 - x_5,$$

but we could take any linear constraint. The flow through $H_1^* = \lambda g$ will then be:

$$\dot{x} = \begin{pmatrix} 0 & 0 & -x_2 & 0 & bx_2 \\ 0 & 0 & x_1 & 0 & -bx_1 \\ x_2 & -x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\varepsilon} \\ -bx_2 & bx_1 & 0 & \frac{1}{\varepsilon} & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \lambda \\ -\lambda \end{pmatrix},$$

which is explicitly solvable. We can compose the flow of the first step with the splitting found in example (1.20), and ignore the third step. We can then use fixed point iteration to find λ such that the constraint is satisfied at the end of the flow. We now get a solution which satisfies the constraint, and the Hamiltonian and Casimir are still conserved as can be seen in the next figures.

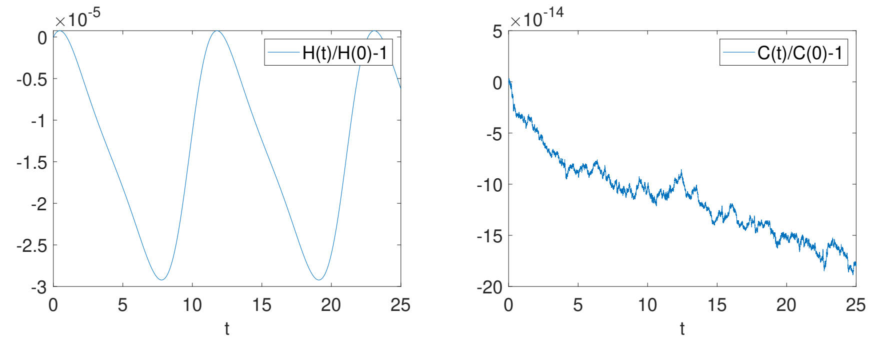

Figure 3.1: Relative change of the Hamiltonian (left) and Casimir (right) for the constrained Lorenz 86 model ($\Delta t = 2.5 \cdot 10^{-4}$).

We will also look at the effect of the constraint on the last two variables. The following figures are the values of x_4 and x_5 for the unconstrained and constrained systems.

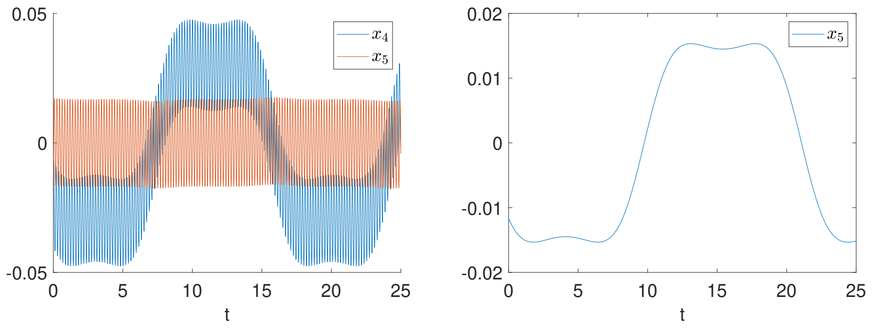

Figure 3.2: Flow of the unconstrained system (left) and the constrained system (right).

△

In this example during the first step the constraint will be preserved, because the Hamiltonian during the first step is given by λg and we are working with a single constraint. Denote by $\tilde{x}$ the solution of the first step and x_0 the initial value, then

$$\lambda g(\tilde{x}) = \lambda g(x_0) = 0. \tag{3.35}$$

Therefore it follows that $g(\tilde{x}) = 0$. When working with multiple constraints this no longer works since (3.35) then only implies $\lambda g(\tilde{x}) = 0$, which means the flow can move in the direction of the left nullspace of λ. After the first step the solution will then no longer be on the manifold so the flow of the unconstrained problem will not start on the manifold. Therefore the solution given by this method will no longer be equivalent to the solution on the constrained system. For a given set of constraints $\{g_i\}_i$ we could define $g = \sum_i g_i^2$, which is a single constraint defining the same constraint manifold. In theory this will work but in practice the first step will most likely no longer be explicitly solvable.

Chapter 4

Poisson discretization of Burgers' equation

In this chapter we will use the theory in Chapter 3 to get a Poisson discretization of Burgers' equation. Recall the discretized Poisson system we left off with at the end of Chapter 2

$$
\begin{aligned}
\frac{d}{dt}F(u,v) &= \{F,H^*\}(u,v), \\
0 &= g(u,v),
\end{aligned}
\tag{4.1}
$$

where the bracket $\{.,.\}$, the Hamiltonian H^* and the constraints g were defined as

$$
\begin{aligned}
\{F,G\} &= \nabla_u F^T B_1(u) \nabla_v G - \nabla_v F^T B_1(u) \nabla_u G, \\
H^*(u,v) &= H(u,v) + \lambda^T g(u,v), \\
g(u,v) &= E_1 u - E_2 v,
\end{aligned}
$$

for B_1, E_1 and E_2 defined in section (2.2.3).

We will start by going through Dirac's procedure given in Section 3.3.1 with our constraints. Then we will be deriving the Poisson system that can be obtained from reducing system (4.1) to the constraint manifold, using the method in section 3.4. Then we will apply the methods from 3.5, giving us two systems. As we will see one of the systems will coincide with the reduced system. Lastly we will look at some numerical results.

4.1 Dirac's procedure

In this section we will look at our set of constraints and apply the theory from Section 3.3.1 to them. Our constraints where given by

$$
g_i(u,v) = u_{i+1} - u_i - \sqrt{3}(v_{i+1} + v_i), \quad \text{for all } i = 1, ..., n.
$$

For all of these constraints we have $\{g_i, g_{i+1}\} \neq 0$, so all of the constraints fit into case 3. So we do not have any secondary constraints, and we can look at the consistency conditions right away. According to formula (3.9) this will give us the following system.

$$
\begin{pmatrix}
0 & u_1 & 0 & \dots & 0 & -u_n \\
-u_1 & 0 & u_2 & 0 & \dots & 0 \\
0 & -u_2 & 0 & \ddots & \dots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & 0 \\
0 & \vdots & \vdots & \ddots & 0 & u_{n-1} \\
u_n & 0 & \dots & 0 & -u_{n-1} & 0
\end{pmatrix}
\lambda =
\begin{pmatrix}
u_1 v_1 - u_n v_n + \sqrt{3}(u_1^2 + u_n^2) \\
u_2 v_2 - u_1 v_1 + \sqrt{3}(u_2^2 + u_1^2) \\
\vdots \\
\vdots \\
\vdots \\
u_n v_n - u_{n-1} v_{n-1} + \sqrt{3}(u_n^2 + u_{n-1}^2)
\end{pmatrix}
\tag{4.2}
$$

We can solve this system for λ. Note that for some values of u the left hand side matrix will have a nullspace, but since the corresponding constraints made of a linear combination of the other constraints will be first class we do not have to do anything with them. We could try replacing some of our original constraints with these linear combinations, but since our constraints correspond to the continuity across grid cells removing them will cause the solution to be discontinuous across those cells due to numerical error. In the next section where we do the reduction to the constraint manifold we will change coordinates to Dirac coordinates. In these coordinates the conditions for Theorem 3.3 should be satisfied since the constraints will be the new variables, so linear combinations of the constraints will not yield first class constraints.

4.2 Reduction to the constraint manifold

In the current formulation of the system (4.1) we can't directly apply the method from section (3.4) because in this case we do not have slaving constraints, so we will start by changing to Dirac coordinates. In this case these coordinates will be given by

$$\begin{pmatrix} u \\ v \end{pmatrix} \to \begin{pmatrix} z \\ w \end{pmatrix} = \begin{pmatrix} h(u,v) \\ g(u,v) \end{pmatrix},$$

where $h(u,v) = (u_1 + \sqrt{3}v_1, \dots, u_n + \sqrt{3}v_n)^T$. We are working with quite a sparse Poisson matrix, so it is easy to calculate that $\{h,h\} = 0$ in this case. In the new coordinates our new system will be

$$\begin{aligned} \begin{pmatrix} \dot{z} \\ \dot{w} \end{pmatrix} &= \begin{pmatrix} 0 & \tilde{B}_1(z,w) \\ -\tilde{B}_1(z,w)^T & \tilde{B}(z,w) \end{pmatrix} \begin{pmatrix} \nabla_z K(z,w) \\ \nabla_w K(z,w) - \lambda \end{pmatrix}, \\ 0 &= g(w) (= w), \end{aligned} \tag{4.3}$$

where K can be calculated in the following way

$$K(z,w) = H(u,v) = \frac{1}{2}\sum_{i=1}^{n}(u_i^2 + v_i^2) = \sum_{i=1}^{n}(\frac{z_i + w_i + z_{i+1}}{2})^2 + (\frac{z_i + w_i - z_{i+1}}{2\sqrt{3}})^2.$$

The matrices $\tilde{B}_1(z,w)$ and $\tilde{B}(z,w)$ are given by

$$\begin{aligned} \tilde{B}_1(z,w) &= \tilde{C}\begin{pmatrix} 0 & z_1+w_1+z_2 & 0 & \dots & \dots & 0 \\ \vdots & 0 & z_2+w_2+z_3 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & z_{n-1}+w_{n-1}+z_n \\ z_n+w_n+z_1 & \dots & \dots & \dots & \dots & 0 \end{pmatrix}, \\ \tilde{B}(z,w) &= \tilde{B}_1(z,w) - \tilde{B}_1^T(z,w), \end{aligned}$$

for $\tilde{C} = \sqrt{3}C$. In the rest of this section $\{.,.\}$ is the Poisson bracket corresponding to this new structure matrix; we also omit the tildes for simplicity. As we mentioned in the last section the constraints all occupy different dimensions of our space so there is no way to take linear combinations of the constraints which bring them into first class in the Dirac theory. Therefore by Theorem 3.3 the following matrix is invertible

$$\{g,g\} = \{w,w\} = B(z,w).$$

We can now replace our constraint by its time derivative $0 = \frac{d}{dt}g(w)$.

$$0 = \frac{d}{dt}g(w(t)) = \dot{w} = -B_1(z,w)^T\nabla_z K(z,w) + B(z,w)\nabla_w K(z,w) - B(z,w)\lambda.$$

We can now use the invertibility of B to solve for λ, giving $\lambda = B(z,w)^{-1}(-B_1(z,w)^T\nabla_z K + B(z,w)\nabla_w K)$. Substituting into (4.3) gives

$$\begin{aligned}\dot{z} = B_1(z,w)\nabla_w K - B_1(z,w)\lambda &= B_1(z,w)\nabla_w K + B_1(z,w)B(z,w)^{-1}B_1(z,w)^T\nabla_z K - B_1(z,w)\nabla_w K \\ &= B_1(z,w)B(z,w)^{-1}B_1(z,w)^T\nabla_z K,\end{aligned}$$

Together with $\dot{w} = 0$. Since $w(0) = 0$ we can now substitute $w = 0$ everywhere, which gives an equation only for z:

$$\dot{z} = B_1(z)B(z)^{-1}B_1(z)^T\nabla_z K(z), \tag{4.4}$$

where $K(z) = K(z,0)$. This system is guaranteed to be a Poisson system by the procedure of Dirac, and it is defined in terms of local coordinates on the constraint manifold. We do not have an explicit inverse for B, but we do have one for B_1 given by

$$B_1^{-1}(z) = \frac{1}{C}\begin{pmatrix} 0 & \cdots & \cdots & \cdots & \cdots & \frac{1}{z_n+z_1} \\ \frac{1}{z_1+z_2} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{1}{z_2+z_3} & 0 & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \frac{1}{z_{n-1}+z_n} & 0 \end{pmatrix}$$

Which allows us to rewrite system (4.4) as

$$M(z)\dot{z} = \nabla K(z), \tag{4.5}$$

where $M(z)$ is given by

$$M(z) = B_1(z)^{-T}B(z)B_1(z)^{-1} = \begin{pmatrix} 0 & \frac{1}{z_1+z_2} & 0 & \cdots & \cdots & 0 & -\frac{1}{z_n+z_1} \\ -\frac{1}{z_1+z_2} & 0 & \frac{1}{z_2+z_3} & 0 & \cdots & \cdots & 0 \\ 0 & -\frac{1}{z_2+z_3} & 0 & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & \vdots & \vdots & \ddots & \ddots & \frac{1}{z_{n-1}+z_n} \\ \frac{1}{z_n+z_1} & 0 & \cdots & \cdots & 0 & -\frac{1}{z_{n-1}+z_n} & 0 \end{pmatrix}$$

This Poisson system is the main result of this thesis. As we will see in the numerical results at the end of this chapter it unfortunately does not yield a stable approximation of Burgers' equation.

4.3 Projection onto the constraint manifold

In this section we will apply the theory from section (3.5). First we will change coordinates, which will allow us to project directly without inverting the structure matrix. As was noted in Section 3.5 this will in general not lead to a structure matrix which is a Poisson matrix. This case, as we will see in the next section, does give the dynamics we would expect from Burgers' equation. Then we will try inverting the structure matrix and projecting afterwards, which will give us a Poisson matrix but as we will see will not give us the correct dynamics.

4.3.1 Direct projection

Consider the following change of coordinates

$$u_i^- = u_i - \sqrt{3}v_i, \qquad u^+ = u_i + \sqrt{3}v_i,$$

and denote $u^- = (u_1^-, ..., u_n^-)^T$, $u^+ = (u_1^+, ..., u_n^+)^T$. In these new coordinates system (4.1) becomes:

$$\begin{pmatrix} \dot{u^-} \\ \dot{u^+} \end{pmatrix} = \begin{pmatrix} 0 & B_1(u^-, u^+) \\ -B_1(u^-, u^+) & 0 \end{pmatrix} \nabla H(u^-, u^+), \tag{4.6}$$

where $B_1(u^-, u^+)$ and $H(u^-, u^+)$ are defined as

$$B_1(u^-, u^+) = \sqrt{\frac{36}{\Delta x^3}} \begin{pmatrix} u_1^- + u_1^+ & 0 & \dots & \dots & \dots & 0 \\ 0 & u_2^- + u_2^+ & 0 & \dots & \dots & \vdots \\ \vdots & 0 & \ddots & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \dots & 0 & u_n^- + u_n^+ \end{pmatrix},$$

$$H(u^-, u^+) = \sum_i (\frac{u_i^- + u_i^+}{2})^2 + (\frac{u_i^- - u_i^+}{2\sqrt{3}})^2 = \frac{1}{3}\sum_i (u_i^-)^2 + (u_i^+)^2 + u_i^- u_i^+.$$

The constraints are now given by $g_i(u^-, u^+) = u_{i+1}^- - u_i^+$ for $i = 1, ..., n$, which can also be written as $g(u^-, u^+) = u^- - Eu^+$ for

$$E = \begin{pmatrix} 0 & \dots & \dots & 0 & 1 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}.$$

Note that in this case we do have slaving constraints, if we now go through the method from section 3.4 we will also end up with system (4.5).

The constraint manifold given by $\mathscr{M} = \{(u^-, u^+) \in \mathbb{R}^{2n} : u^- - Eu^+ = 0\}$ admits a chart $\chi : \mathbb{R}^n \to \mathscr{M}$ given by $\chi(y) = (y, Ey)^T$. The Jacobian of this chart is given by

$$X(y) = \begin{pmatrix} I \\ E \end{pmatrix},$$

which admits a left inverse given by

$$X(y)^{-1} = \frac{1}{2}\begin{pmatrix} I & E^T \end{pmatrix}.$$

So we can write our system on the manifold using this chart as

$$\dot{y} = \tilde{B}(y) \nabla \tilde{K}(y), \tag{4.7}$$

Where $\tilde{B}$ and $\tilde{K}$ are given by

$$\tilde{B}(y) = X(y)^{-1}\begin{pmatrix} 0 & B_1(\chi(y)) \\ -B_1(\chi(y)) & 0 \end{pmatrix} X(y)^{-T}$$

$$= \sqrt{\frac{36}{16\Delta x^3}}\begin{pmatrix} 0 & y_1+y_2 & 0 & \dots & 0 & -(y_n+y_1) \\ -(y_1+y_2) & 0 & y_2+y_3 & 0 & \dots & 0 \\ 0 & -(y_2+y_3) & 0 & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & \vdots & \ddots & 0 & y_{n-1}+y_n \\ y_n+y_1 & 0 & \dots & 0 & -(y_{n-1}+y_n) & 0 \end{pmatrix},$$

$$\tilde{K}(y) = \sum_i \frac{2}{3}y_i^2 + \frac{1}{3}y_iy_{i+1}.$$

Recall we noted that this procedure of directly projecting onto the constraint manifold will not produce Poisson systems in general. We can see this here because the matrix $\tilde{B}$, apart from being skew-symmetric, does not satisfy the conditions from Lemma 1.9. We can for example take $i = 1, j = 2$ and $k = 3$ in the second condition from the lemma, which will not yield an identity.

4.3.2 Projection after inverting

Because our structure matrix corresponding to system (4.1) is invertible we can also project using Theorem 3.7. We will start from the system in Dirac coordinates (4.3) because this is convenient. The matrix B_1 is invertible so the inverse of the structure matrix from (4.3) is given by

$$\begin{pmatrix} -B_1(z,w)^{-T}B(z,w)B_1^{-1} & -B_1^{-T} \\ B_1^{-1} & 0 \end{pmatrix}.$$

Since our constraints are in this case given by $g(w) = w$ a chart for the constraint manifold is given by $\chi(y) = (y,0)^T$. Let $X(y) = \chi'(y)$, then the system after projecting will be

$$\dot{y} = \tilde{B}(y)\nabla\tilde{K}(y), \tag{4.8}$$

where $\tilde{B}$ is given by

$$\tilde{B}(y) = -(B_1(\chi(y))^{-T}B(\chi(y))B_1(\chi(y))^{-1})^{-1},$$

and $\tilde{K}(y) = H(\chi(y))$. Note that this is the same system we found in section 4.2. So the resulting system from this projection is a Poisson system, and Theorem 3.7 works at least in this case.

4.4 Numerical results

In this section we will discuss the numerical integration of systems found in this chapter. Unfortunately the only system giving the correct dynamics is the system in (4.7) with a structure matrix which was only skew-symmetric and not Poisson. After we plot the dynamics of this system we will discuss the time integration of system (4.8). This system is actually a Poisson system but as we will see does not produce the correct dynamics.

We can integrate system (4.7) using the midpoint rule, since this will preserve quadratic invariants and our Hamiltonian is quadratic. For the initial condition we have chosen $\sin(x)$ on the domain $[0,2\pi]$. The domain is discretized into 100 points and we have used a time step of $\Delta t = 3\cdot 10^{-3}$. In the following figure the solution to Burgers' equation at different time points is shown.

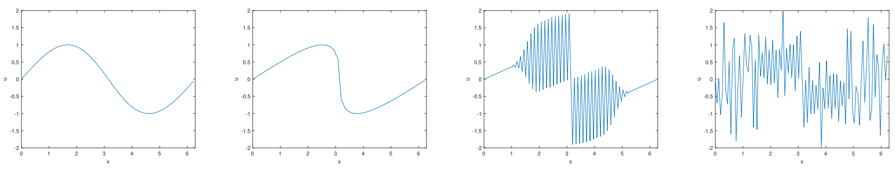

Figure 4.1: The initial value, the solution before the shock, the solution after the shock and the solution some time after the shock for system (4.7).

We can see here that the system actually produces the dynamics we would expect from an energy conserving approximation of Burgers' equation. The positive part moves to the right and the negative part to the left until they meet in the center, which produces a shockwave which then fizzles out into noise after a while.

We will now have a look at the Poisson system given in (4.4). The Hamiltonian $K(z) = K(z,0)$ can be approximated in the following way

$$K(z) = K(z,0) = \sum_{i=1}^{n} (\frac{z_i + z_{i+1}}{2})^2 + (\frac{z_i - z_{i+1}}{2\sqrt{3}})^2 \approx \frac{1}{3}\sum_{i=1}^{n} z_i^2$$

In the limit as n goes to infinity this approximation will become exact. We can now split the Hamiltonian into $K = \sum_{i=1}^{n} K_i$

$$K_i(z) = \frac{1}{3}z_i^2.$$

All the subsystems corresponding to this system are explicitly solvable. We will show the procedure for $K_1(z) = \frac{1}{3}z_1^2$. The system corresponding to K_1 is given by

$$M(z)\dot{z} = \frac{2}{3}z_1 e_1,$$

For the odd indices of z we have equations

$$\frac{\dot{z}_{2i+1}}{z_{2i+1} + z_{2i}} - \frac{\dot{z}_{2i-1}}{z_{2i-1} + z_{2i}} = 0,$$

for all values of i. Now since $K_1 = \frac{1}{3}z_1^2$ is conserved in this subsystem we have $\dot{z}_1 = 0$, this together with the above equations forces $\dot{z}_{2i-1} = 0$ for all i. For the even indices of z we have the following equations

$$\begin{aligned} \frac{\dot{z}_2}{z_1 + z_2} - \frac{\dot{z}_n}{z_n + z_1} &= \frac{2}{3}z_1, \\ \frac{\dot{z}_{2i}}{z_{2i-1} + z_{2i}} - \frac{\dot{z}_{2i-2}}{z_{2i-2} + z_{2i-1}} &= 0, \qquad \text{for all } i \neq 1. \end{aligned}$$

Since we have $\dot{z}_{2i-1} = 0$ for all i all of the odd indices of z remain constant, allowing us to rewrite the above equations as

$$\begin{aligned} \frac{d}{dt}\log(z_1 + z_2) - \frac{d}{dt}\log(z_n + z_1) &= \frac{2}{3}z_1, \\ \frac{d}{dt}\log(z_{2i-1} + z_{2i}) - \frac{d}{dt}\log(z_{2i-2} + z_{2i-1}) &= 0, \qquad \text{for all } i \neq 1. \end{aligned}$$

We can integrate all of these equations and take the exponent, leading to

$$\begin{aligned} \frac{z_n(0) + z_1}{z_2(0) + z_1}\frac{z_2(t) + z_1}{z_n(t) + z_1} &= e^{\frac{2}{3}z_1 t}, \\ \frac{z_{2i-2}(0) + z_{2i-1}}{z_{2i}(0) + z_{2i-1}}\frac{z_{2i}(t) + z_{2i-1}}{z_{2i-2}(t) + z_{2i-1}} &= 1, \qquad \text{for all } i \neq 1. \end{aligned}$$

Since we want to know the solution after a certain time step Δt we can substitute $t = \Delta t$, which will give us a linear system which we can solve.

$$(z_n(0)+z_1)z_2(\Delta t) - e^{\frac{2}{3}z_1\Delta t}(z_2(0)+z_1)z_n(\Delta t) = e^{\frac{2}{3}z_1\Delta t}(z_2(0)+z_1)z_1 - (z_n(0)+z_1)z_1$$
$$(z_{2i-2}(0)+z_{2i-1})z_{2i}(\Delta t) - (z_{2i}(0)+z_{2i-1})z_{2i-2}(\Delta t) = z_{2i-1}(z_{2i}(0)-z_{2i-2}(0)).$$

Solving this will give us the solution of the subsystem after time step Δt. All of the other subsystems can be solved in a similar way.

Like we mentioned this will not give the correct dynamics. The following figures show the time evolution of the system.

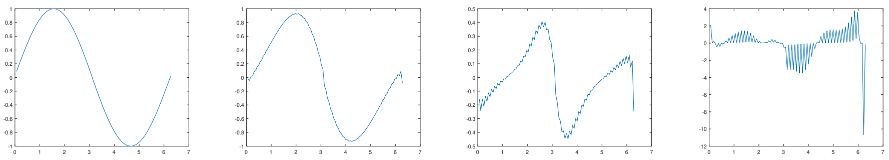
Figure 4.2: Time integration of system (4.4).

As we can see the system does initially start moving in a way we would expect, the maximum and minimum start moving towards each other. Then it starts developing spikes on the edges. After the third figure it will start to oscillate rapidly until it settles down into the shape seen in the fourth figure.

Conclusion

In this thesis we wanted to find a Poisson discretization of Burgers'equation. We used the Clebsch variational principle to get a weak formulation of Burgers'equation, which we then discretized into a Poisson system. But, as we have seen, this Poisson formulation was not useful since we ended up with a system which was not coupled across grid cells. This was a problem since we wanted to get a continuous solution, so we tried to enforce continuity of the solution trough constraint equations. This gave us a system of differential algebraic equations. We then looked at Poisson systems with constraints, and how to reduce the system to the constrained manifold in a way that again gave us a Poisson system. The solution ended up being the procedure by Dirac, which guaranteed a Poisson system on the constrained manifold by altering the original Poisson bracket. Then we applied this theory to Burgers'equation, which gave us a Poisson system. Then we tried to integrate the Poisson discretization in time, but we found that this did not give us the dynamics we would expect from Burgers'equation. So the Poisson system we derived is not a stable approximation of Burgers' equation. We did manage to integrate a different formulation, which we acquired by directly projecting onto the constraint manifold. This formulation was not a Poisson system, but it was skew-symmetric so it does preserve quadratic invariants like the Hamiltonian corresponding to Burgers'equation.

Appendix A

Matlab codes

A.1 Example 1.20

```
%time discretization
m=100000;
T=25;
dt=T/m;

%parameters
b=3;
eps=0.01;

%initial values
x=zeros(5,m);
x=[1.3;0.6;1;-0.0115;-0.0115];

%Hamiltonian H and Casimir C
H_f=@(x) 1/2*(x(1)^2+2*x(2)^2+x(3)^2+x(4)^2+x(5)^2);
H=zeros(m+1,1);
H(1)=H_f(x(:,1));
C_f=@(x) 1/2*(x(1)^2+x(2)^2);
C=zeros(m+1,1);
C(1)=C_f(x(:,1));


for i=1:m
    %H1
    x_1=zeros(5,1);
    x_1(1)=x(1,i);
    x_1(2)=x(2,i);
    x_1(3)=-x(1,i)*x(2,i)*dt/3+x(3,i);
    x_1(4)=x(4,i);
    x_1(5)=b*x(1,i)*x(2,i)*dt/3+x(5,i);

    %H2
    x_2=zeros(5,1);
    x_2(1)=x_1(1)*cos(x_1(3)*dt/3)-x_1(2)*sin(x_1(3)*dt/3);
    x_2(2)=x_1(1)*sin(x_1(3)*dt/3)+x_1(2)*cos(x_1(3)*dt/3);
    x_2(3)=x_1(3);
    x_2(4)=x_1(4);
    x_2(5)=dt*x_1(4)/(eps*3)+x_1(5);

    %H3
    x(1,i+1)=x_2(1)*cos(b*x_2(5)*dt/3)+x_2(2)*sin(b*x_2(5)*dt/3);
    x(2,i+1)=x_2(2)*cos(b*x_2(5)*dt/3)-x_2(1)*sin(b*x_2(5)*dt/3);
    x(3,i+1)=x_2(3);
    x(4,i+1)=-x_2(5)*dt/(eps*3)+x_2(4);
    x(5,i+1)=x_2(5);

    H(i+1)=H_f(x(:,i+1));
    C(i+1)=C_f(x(:,i+1));
end
%plots
figure;
plot(0:dt:T,H/H(1)-1)
xlabel("t")
legend('H(t)/H(0)-1')
figure;
plot(0:dt:T,C/C(1)-1)
xlabel("t")
legend('C(t)/C(0)-1')
figure; hold;
plot(0:dt:T,x(4,:))
plot(0:dt:T,x(5,:))
xlabel("t")
legend('$x_4$','$x_5$','Interpreter','latex')
```

A.2 Example 3.9

```
    %time discretization
m=100000;
T=25;
dt=T/m;

%parameters
b=3;
eps=0.01;

%fixed point iteration parameters
mu=1;
maxiter=10000;
tol=1e-15;

%initial values
x=[1.3;0.6;1;-0.0115;-0.0115];


H_f=@(x) 1/2*(x(1)^2+2*x(2)^2+x(3)^2+x(4)^2+x(5)^2);
H=zeros(m+1,1);
H(1)=H_f(x(:,1));
```

```
C_f=@(x) 1/2*(x(1)^2+x(2)^2);
C=zeros(m+1,1);
C(1)=C_f(x(:,1));

%constraints
g=@(x,y) x-y;

%Lagrange multipliers
L=zeros(m,1);
lambda=0;

for i=1:m
    for k=1:maxiter
    %flow trough constraint
    x_constr=zeros(5,1);
    x_constr(1)=x(1,i)*cos(b*lambda)-x(2,i)*sin(b*lambda);
    x_constr(2)=x(1,i)*sin(b*lambda)+x(2,i)*cos(b*lambda);
    x_constr(3)=x(3,i);
    x_constr(4)=lambda/eps+x(4,i);
    x_constr(5)=lambda/eps+x(5,i);

    %H1
    x_2=zeros(5,1);
    x_2(1)=x_constr(1);
    x_2(2)=x_constr(2);
    x_2(3)=-x_constr(1)*x_constr(2)*dt/4+x_constr(3);
    x_2(4)=x_constr(4);
    x_2(5)=b*x_constr(1)*x_constr(2)*dt/4+x_constr(5);

    %H2
    x_3=zeros(5,1);
    x_3(1)=x_2(1)*cos(x_2(3)*dt/4)-x_2(2)*sin(x_2(3)*dt/4);
    x_3(2)=x_2(1)*sin(x_2(3)*dt/4)+x_2(2)*cos(x_2(3)*dt/4);
    x_3(3)=x_2(3);
    x_3(4)=x_2(4);
    x_3(5)=x_2(4)/eps*dt/4+x_2(5);

    %H3
    x_5=zeros(5,1);
    x_5(1)=x_3(1)*cos(b*x_3(5)*dt/4)+x_3(2)*sin(b*x_3(5)*dt/4);
    x_5(2)=x_3(2)*cos(b*x_3(5)*dt/4)-x_3(1)*sin(b*x_3(5)*dt/4);
    x_5(3)=x_3(3);
    x_5(4)=-x_3(5)*dt/(eps*4)+x_3(4);
    x_5(5)=x_3(5);

    %check if constraint is satisfied
    if abs(g(x_5(4),x_5(5)))<tol
        x(:,i+1)=x_5;
        fprintf("converged in %i iterations\n",k)
        break;
    end
    %update lambda
    lambda=lambda+mu*g(x_5(4),x_5(5));
    if k==maxiter
        fprintf("did not converge\n")
        return
    end
    end
    H(i+1)=H_f(x(:,i+1));
    C(i+1)=C_f(x(:,i+1));
end
%plots
figure;
plot(dt:dt:T,H(2:end)/H(1) - 1)
xlabel("t");
legend('H(t)/H(0)-1')
figure;
plot(dt:dt:T,C(2:end)/C(1)-1);
xlabel("t")
legend('C(t)/C(0)-1')
figure;
plot(0:dt:T,x(4,:))
xlabel("t")
legend('$x_5$','Interpreter','latex')
```

A.3 System (4.7)

```
%space discretization
L = 2*pi;
N = 100;
dx = L/N;

%projection matrices
P = kron(eye(N),[1; 1]);
P = P([2:end,1],:);
R = 1/2*P';
D = kron(eye(N),[-1 1]);
D = D(:,[2:end,1]);
M = 2*kron(eye(N),[1/3 1/6;1/6 1/3]);
M = R*M*P;

%initial value
x = [0:N-1]'/N*L;
u = sin(x);

%time discretization
Nsteps = 100*N;
T = 30;
dt = T/Nsteps;

maxIter = 100;
tol = 1e-14;
```

```
H(1) = 1/2 * u'*M*u;


for n=1:Nsteps
    %midpoint rule
    B = 1/dx * P'*poissonB(P*u)*P;
    u1 = (M - dt/2*B) \ M*u;

    %fixed point iteration
    for iter = 1:maxIter
        B = 1/dx * P'*poissonB(P*u1)*P;
        r = M * (u1 - u) - dt/2 * B*u1;
        u1 = u1 - M\r;
        if norm(r)<tol
            break
        end
    end
    if iter>maxIter
        disp ('no conv');
    end
    u = 2*u1 - u;

    %plot solution during iteration
    plot(x,u);
    ylabel('u')
    xlabel('x')
    axis([0 L -2 2])
    H(n+1) = 1/2 * u'*M*u;
    drawnow;
end


figure(2); plot(H/H(1) - 1)
```

A.4 System (4.4)

```
%space discretization
n=102;
L=2*pi;
dx=L/n;

%time discretization
T=1;
m=10000;
dt=T/m;

%constants
C=sqrt(36/dx^3);

%initial value
x = (1:n)'/n*L;
```

```
u=sin(x);

%Hamiltonian
H=zeros(m,1);
H(1)=norm(u+u([n,1:n-1]))^2;
for i=1:m
    for j=1:n/2
        %even
        b_e=arrayfun(@(k) u(2*k-1)*(u(2*k)-u(mod(2*k-3,n)+1)),(1:
           n/2)');
        b_e(j)=u(2*j-1)*(-(u(mod(2*j-3,n)+1)+u(2*j-1))+exp(C*u(2*
           j-1)*dt/n)*(u(2*j)+u(2*j-1)));

        d_e=u([n,2:2:n-2])+u(1:2:n-1);
        od_e=-(u(2:2:n)+u(1:2:n-1));
        od_e(j)=-exp(C*u(2*j-1)*dt/n)*(u(2*j)+u(2*j-1));

        M_e=(diag(d_e)+diag(od_e(2:end),-1)+diag(od_e(1),n/2-1));

        u(2:2:end)=M_e\b_e;

        %odd
        b_o=arrayfun(@(k) u(2*k)*(u(mod(2*k+1,n))-u(2*k-1)),(1:n
           /2)');
        b_o(j)=u(2*j)*((u(2*j)+u(mod(2*j+1,n)))*exp(C*u(2*j)*dt/n
           )-(u(2*j-1)+u(2*j)));


        d_o=-(u(2:2:end)+u([3:2:end-1,1]));
        d_o(j)=-(u(2*j)+u(mod(2*j+1,n)))*exp(C*u(2*j)*dt/n);
        od_o=u(2:2:end)+u(1:2:end-1);

        M_o=diag(d_o)+diag(od_o(1:end-1),1)+diag(od_o(end),1-n/2)
           ;

        u(1:2:end-1)=M_o\b_o;
    end
    H(i+1)=norm(u+u([n,1:n-1]))^2;
    plot(x,1/2*(u+u([2:end,1])));
    drawnow;
end
```

Bibliography

[1] E. Hairer, Christian Lubich, and Gerhard Wanner. Geometric numerical integration : structure-preserving algorithms for ordinary differential equations, 2006.

[2] Robert I McLachlan and Pau Atela. The accuracy of symplectic integrators. *Nonlinearity*, 5(2):541, 1992.

[3] Robert McLachlan. Symplectic integration of Hamiltonian wave equations. *Numerische Mathematik*, 66:465–492, 1993.

[4] V Zeitlin. Finite-mode analogs of 2d ideal hydrodynamics: Coadjoint orbits and local canonical structure. *Physica D: Nonlinear Phenomena*, 49(3):353–362, 1991.

[5] Colin J Cotter and Darryl D Holm. Continuous and discrete Clebsch variational principles. *Foundations of Computational Mathematics*, 9:221–242, 2009.

[6] Edward N Lorenz. On the existence of a slow manifold. *Journal of Atmospheric Sciences*, 43(15):1547–1558, 1986.

[7] Benedict Leimkuhler and Sebastian Reich. Symplectic integration of constrained Hamiltonian systems. *Mathematics of Computation*, 63(208):589–605, 1994.

[8] Sebastian Reich. Symplectic integration of constrained Hamiltonian systems by composition methods. *SIAM journal on numerical analysis*, 33(2):475–491, 1996.

[9] S Leyendecker, P Betsch, and P Steinmann. Energy-conserving integration of constrained Hamiltonian systems–a comparison of approaches. *Computational Mechanics*, 33:174–185, 2004.

[10] E. Hairer and Gerhard Wanner. Solving ordinary differential equations ii, 2010.

[11] Paul Adrien Maurice Dirac. *Lectures on quantum mechanics*, volume 2. Courier Corporation, 2001.

[12] Paul Adrien Maurice Dirac. Generalized Hamiltonian dynamics. *Canadian journal of mathematics*, 2:129–148, 1950.

[13] J Vanneste and Onno Bokhove. Dirac-bracket approach to nearly geostrophic Hamiltonian balanced models. *Physica D: Nonlinear Phenomena*, 164(3-4):152–167, 2002.

[14] Jerrold E. Marsden and Tudor S. Ratiu. *Introduction to mechanics and symmetry : a basic exposition of classical mechanical systems*. Texts in applied mathematics; 17. Springer, New York, 2nd ed edition, 2011.

[15] Robert I McLachlan, Klas Modin, Olivier Verdier, and Matt Wilkins. Geometric generalisations of SHAKE and RATTLE. *Foundations of Computational Mathematics*, 14:339–370, 2014.