

EFFECTIVE NUMERICAL METHODS TO  
MODEL DYNAMIC BEHAVIOR OF  
SPRINGS WITH TORSION.

by  
E. Dilan Fernando

Prof. Stephen Montgomery-Smith, Dissertation  
Supervisor

A Dissertation presented to the Faculty of the  
Graduate School of the University of Missouri

In Partial Fulfillment of the Requirements for the  
Degree Doctor of Philosophy

July 2018

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

**EFFECTIVE NUMERICAL METHODS TO MODEL DYNAMIC  
BEHAVIOR OF SPRINGS WITH TORSION.**

presented by E. Dilan Fernando,

a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

**Committee Member**

**Signature**

Prof. Stephen Montgomery-Smith (Advisor)

\_\_\_\_\_

Prof. Yuri Latushkin

\_\_\_\_\_

Prof. Carmen Chicone

\_\_\_\_\_

Prof. Sam Walsh

\_\_\_\_\_

Prof. David Retzloff

\_\_\_\_\_

# Dedication

*Dedicated to my mother Lalitha for all her love and support*



# Acknowledgments

First I would like to thank my family for their love and support throughout the years. Especially my mother who has sacrificed so much on behalf of me. I will not be where I am today without her.

I also want to thank the staff at the Department of Mathematics for their helpfulness, especially Amy Crews, Gwen Gilpin and Kyle Newell-Groshong. Their hard work and helpfulness made our stay at Mizzou a pleasant one.

I also want to thank all the teachers who have taught me Mathematics over the years. Special thanks goes out to Professor Yuri Latushkin for his support and encouragement during some difficult times. His kindness and help is greatly appreciated.

I want to thank my advisor Professor Stephen Montgomery-Smith for his expertise, patience and generosity of time he showed throughout the preparation of this dissertation. His vast knowledge of both Mathematics and programming was a great source of inspiration and knowledge.

Last but not least I want to thank my wife Sawani for her love and encouragement throughout the preparation of this dissertation.

# Contents

<b>Acknowledgments</b>	<b>ii</b>
<b>List of Figures</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The discretized Hamiltonian . . . . .	3
1.2 Time discretization and the problem of stiffness . . . . .	4
<b>2 Problem of Elastica</b>	<b>7</b>
2.1 Mathematical description . . . . .	7
2.2 Adding damping terms . . . . .	9
2.3 Space discretization and cubic splines . . . . .	10
2.4 Time discretization that preserves the Hamiltonian . . . . .	12
2.5 Computation of matrices $\mathbf{V}, \mathbf{C}_k$ etc . . . . .	15
2.5.1 Computing $\mathbf{V}$ and $\mathbf{v}$ . . . . .	16
2.5.2 Computing $\mathbf{C}_k, \mathbf{c}_k$ and $\chi_k$ . . . . .	16
2.6 System of equations and the associated Jacobian . . . . .	19
<b>3 Springs with torsion</b>	<b>23</b>
3.1 Preliminaries . . . . .	23
3.2 The transported natural curvature and the natural frame. . . . .	24

3.3	Natural Frames . . . . .	27
3.4	Helical springs . . . . .	30
3.4.1	Notations and preliminaries . . . . .	30
3.4.2	Analytical calculation of transported natural curvature for a helical spring . . . . .	32
<b>4</b>	<b>Dynamic equations of a spring</b>	<b>37</b>
4.1	Introducing the variables . . . . .	37
<b>5</b>	<b>Solving a constrained ODE</b>	<b>43</b>
5.1	Notation . . . . .	43
5.2	The space discretization . . . . .	43
5.3	The transported rest curvature . . . . .	53
5.4	Energies and Constraints . . . . .	54
5.5	The differential equation. . . . .	55
5.6	Monotonocity of the Hamiltonian . . . . .	66
5.7	The time discretization . . . . .	70
5.7.1	Discretization of $\mathbf{f}_j(\tau, \mathbf{s})$ and $\mathbf{f}_j(\dot{\tau}, \mathbf{s})$ . . . . .	73
5.7.2	Initial Conditions . . . . .	75
<b>6</b>	<b>Computing the Jacobian of the system</b>	<b>77</b>
6.1	Motivation for computing Jacobian . . . . .	77
6.2	Notation and simplifications . . . . .	78
6.3	Derivatives of the first equation $f_1$ . . . . .	78
6.3.1	Computing $\frac{\partial f_1}{\partial \mathbf{x}_{n+1}}$ : . . . . .	78
6.3.2	Computing $\frac{\partial f_1}{\partial \mathbf{S}_{n+1}}$ : . . . . .	79
6.3.3	Computing $\frac{\partial f_1}{\partial \mathbf{A}_{n+1}}$ . . . . .	80
6.3.4	Computing $\frac{\partial f_1}{\partial \mathbf{B}_{n+1}}$ . . . . .	80
6.3.5	Computing $\frac{\partial f_1}{\partial \alpha_{n+1}}$ . . . . .	80

6.4	Derivatives of the second equation $f_2$ . . . . .	81
6.4.1	Computing $\frac{\partial f_2}{\partial \mathbf{x}_{n+1}}$ : . . . . .	84
6.4.2	Computing $\frac{\partial f_2}{\partial \mathbf{S}_{n+1}}$ . . . . .	85
6.4.3	Computing $\frac{\partial f_2}{\partial \mathbf{A}_{n+1}}$ . . . . .	96
6.4.4	Computing $\frac{\partial f_2}{\partial \mathbf{B}_{n+1}}$ . . . . .	97
6.4.5	Computing $\frac{\partial f_2}{\partial \alpha}$ . . . . .	98
6.5	Derivatives of constraint equations . . . . .	98
6.6	Derivatives of $f_3$ . . . . .	99
6.6.1	Derivative $\frac{\partial f_3}{\partial \mathbf{x}_{n+1}}$ . . . . .	99
6.6.2	Derivative $\frac{\partial f_3}{\partial \mathbf{S}_{n+1}}$ . . . . .	100
6.6.3	Derivatives $\frac{\partial f_3}{\partial \mathbf{A}}, \frac{\partial f_3}{\partial \mathbf{B}}, \frac{\partial f_3}{\partial \alpha}$ . . . . .	100
6.7	Derivatives of $f_4$ . . . . .	101
6.7.1	Derivative $\frac{\partial f_4}{\partial \mathbf{x}_{n+1}}$ . . . . .	101
6.7.2	Derivative $\frac{\partial f_4}{\partial \mathbf{S}_{n+1}}$ . . . . .	101
6.7.3	Derivatives of $\frac{\partial f_4}{\partial \mathbf{A}}, \frac{\partial f_4}{\partial \mathbf{B}}, \frac{\partial f_4}{\partial \alpha}$ . . . . .	102
6.8	Derivatives of $f_5$ . . . . .	102
6.9	The Jacobian . . . . .	103
	<b>Bibliography</b>	<b>105</b>
	<b>Vita</b>	<b>115</b>



## List of Figures

3.1	The natural curvature shown on a spiral spring . . . . .	26
3.2	Corresponding natural curvature . . . . .	26
3.3	A natural frame moving along a curve . . . . .	28

# Abstract

The purpose of this thesis is to find effective algorithms to numerically solve certain systems of differential equations that arise from standard Newtonian mechanics. Numerical models of *elastica* has already been well studied. In this thesis we concentrate on the *Kirchhoff* problem. The goal is to create an effective and robust numerical method to model the dynamic behavior of springs that have a prescribed natural curvature.

In addition to the mathematics, we also provide the implementation details of the numerical method using the computer language *Python 3*. We also discuss in detail the various difficulties of such a software implementation and how certain auxiliary computations can make the software more effective and robust.



# Chapter 1

## Introduction

The purpose of our thesis is to find effective algorithms which solve systems of differential equations that arise from standard Newtonian dynamics. The main piece of mathematics we employ is to introduce algorithms that preserve a discretized version of the associated Hamiltonian. We will try this out on a number of problems which have historically been very difficult to solve numerically.

In addition to the Mathematics of these algorithms, a significant part of this thesis concerns the practical implementation of these algorithms in the computer programming language *Python 3*. We in particular make extensive use of the popular *numpy* and *scipy* packages for our implementations. These packages already boast of highly optimized code that can be readily used for effective matrix computations. However, the use of such packages in a manner that is useful to our particular case has to be handled with care.

In practice we have found that it is one thing to have a mathematically correct algorithm but it is completely another to implement in practice. Many practical difficulties arise in such implementations and our intention was to discuss these difficulties and how to overcome them using Python 3. For example, for these algorithms to be practical one needs to be able to provide an analytical Jacobian to the system.

We discuss extensively how to carry out these computations and implement them in Python.

In the first part of the thesis we re-implement the numerical algorithm introduced in [52]. The rational for re-implementation was this. The computer language used for the implementation was *Mathematica*. There are many shortcomings in Mathematica in addition to being expensive and closed source. The syntax is so arcane that it becomes impossible for anyone other than the original author to read and understand the code. Also the algorithm it uses to solve non linear systems is also what is called a *black box* algorithm in that the source code for the algorithm is not available to the user. Therefore it is not possible to do any fine tuning if the program somehow doesn't do exactly what you would like it to do

We instead implement the algorithm in Python since it is a widely used, interpreted high level programming language for general purpose programming. It has lots of advantages over Mathematica such as availability of highly optimized packages for numerical computing and a large and helpful user base. It is also worth noting that the source code for these packages are open source. In fact Python has now become the most widely used language used in scientific computation throughout the programming community. In Python it is far easier to write easily maintainable, scalable and robust code.

In the second part of the thesis we discuss the specific case of the *Kirchhoff* problem, that is, creating effective and robust software that models the dynamic behavior of springs, including springs which are not necessarily helical. These springs are ones which are thin, inextensible, elastic rods that have a prescribed natural curvature.

## 1.1 The discretized Hamiltonian

The time discretization method we present is designed so that it preserves the energy of the system. There are lots of literature on energy conserving methods for Hamiltonian equations. See for example [9, 12, 18, 24, 40, 62, 65–69]. Many of these explicitly deal with the problems we are attempting in this thesis but from different perspectives. In particular, these methods seek to preserve the symplectic form of the motion rather than the Hamiltonian (c.f. [40]). Historically the notion of representing equations of motion by Hamiltonians came after it was realized that many equations of motion can be derived from a principle of least action. Let us consider the simple case where the kinetic energy is quadratic.

$$\mathcal{E} = \frac{1}{2}m |\mathbf{v}|^2 \tag{1.1}$$

where  $\mathbf{v} = \dot{\mathbf{x}}$  is the velocity in some  $m$ -dimensional space, and potential energy  $\mathcal{V} = \mathcal{V}(\mathbf{x})$  only depends on the position  $\mathbf{x}$ . The Hamiltonian approach is to introduce the momentum  $\mathbf{p} = m\mathbf{v}$ , and the Hamiltonian  $H = \mathcal{E}(\mathbf{p}/m) + \mathcal{V}(x)$  and see that the Hamiltonian is conserved as the equations of motions evolve. However, for prior researchers it has been a difficult task to find effective time discretizations that preserve the Hamiltonian.

Our approach is to go back to the Euler–Lagrange equations from which the Hamiltonian equations were historically derived. More precisely we create the action

$$\mathcal{S}(\mathbf{x}) = \int_{t_1}^{t_2} (-\mathcal{E} + \mathcal{V}) dt. \tag{1.2}$$

Then the Hamilton’s principle is that the equations of motion can be found by calculating the stationary point of the action. That is, find the vector  $\mathbf{x}$ , satisfying the

constraints such that for any infinitesimal perturbation  $\delta\mathbf{x}$  of  $\mathbf{x}$  we have,

$$\delta\mathcal{S}(\mathbf{x}) = \mathcal{S}(\mathbf{x} + \delta\mathbf{x}) - \mathcal{S}(\mathbf{x}) = \mathcal{O}(|\delta\mathbf{x}|^2) \quad (1.3)$$

It is known that the resulting equations of motion conserve the Hamiltonian

$$H(t) = \mathcal{E}(\dot{\mathbf{x}}) + \mathcal{V}(\mathbf{x}) \quad (1.4)$$

We present a time discretization numerical method that computes  $\mathbf{x}_n = \mathbf{x}(t)$  for  $t = nh$  such that the *discretized* Hamiltonian

$$H_n = \mathcal{E}\left(\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{h}\right) + \frac{\mathcal{V}(\mathbf{x}_n) + \mathcal{V}(\mathbf{x}_{n+1})}{2} \quad (1.5)$$

is conserved.

## 1.2 Time discretization and the problem of stiffness

A problem which arises in many software solvers for ODEs is the problem of *stiffness*. In particular the differential equations we present here has that problem. Roughly speaking stiffness happens when one part of the solution to the differential equation evolves faster than another part of the solution. For example when solving the heat equation, sharp spikes of heat will quickly dissipate, whereas broad regions of low density heat do not change at all. The problem here is that a very short time step  $h$  is needed, because otherwise the parts that decay fast will overshoot and create large effects on the final numerical solution. The problem of stiffness has been amply handled in literature. For example see [27].

However, the equations we present here has a different type of stiffness. The

typical stiffness involves linear operators whose eigenvalues are very large, but in the negative direction. Our equation has large eigenvalues but in the imaginary direction. This means that instead of fast parts dissipating quickly we have the problem of fast parts oscillating very quickly. This is the reason for looking for numerical schemes that preserve the discretized Hamiltonian.





# Chapter 2

## Problem of Elastica

We will first give a brief yet sufficient overview of the problem. More details including proofs can be found in [52]. What we do in here is to show in detail the computational aspects leading up to the software implementation. We also show the actual computer algorithm which solves this system of differential equations.

### 2.1 Mathematical description

We assume that the material from which the rod is constructed has two parameters:  $\rho$  which is the mass per unit length of the rod, and  $\epsilon$  which is the elasticity of the material from which the rod is made. For the sake of simplicity we will assume that these quantities are constant throughout the length of the rod. Removing these assumptions do not necessarily complicate the situation but in this instance we chose not to. Let us consider a rod of length  $L$ .

The action  $\mathcal{S}$  consists of three parts: the kinetic energy  $\mathcal{E}$ , the potential energy  $\mathcal{V}$

and the constraint  $\mathcal{C}$  multiplied by the Lagrange multiplier  $\alpha(s)$ . More precisely,

$$\begin{aligned}\mathcal{E} &= \int_0^L \frac{1}{2}\rho |\dot{\mathbf{x}}(s)|^2 ds, \\ \mathcal{V} &= \int_0^L \frac{1}{2} |\mathbf{x}''(s)|^2 ds, \\ \mathcal{C} &= \int_0^L \frac{1}{2}\alpha(s) |\mathbf{x}'(s)|^2 ds.\end{aligned}\tag{2.1}$$

We allow only the following type of boundary conditions: where  $\mathbf{x}(0)$  and  $\mathbf{x}(L)$  are known functions of  $t$  and  $\mathbf{x}'(0)$  and  $\mathbf{x}'(L)$  are not known functions of  $t$ . The steady state solutions of such systems has been known for a long time. For example one can find formulas for solutions involving elliptic integrals in [41] and in Love [43].

The equations of motions are found by applying Hamilton's principle to the action  $\mathcal{S}$ . We create the action  $\mathcal{S}$  as

$$\mathcal{S}(\mathbf{x}) = \int_{t_1}^{t_2} (-\mathcal{E} + \mathcal{V} + \mathcal{C}) dt\tag{2.2}$$

and then we calculate the stationary point of the action. More precisely find the path  $\mathbf{x}(s, t)$ , satisfying the boundary conditions and the constraints, such that for any infinitesimal perturbation  $\delta\mathbf{x}$  of  $\mathbf{x}$  we have  $\delta\mathcal{S}(\mathbf{x}) = \mathcal{S}(\mathbf{x} + \delta\mathbf{x}) - \mathcal{S}(\mathbf{x}) = \mathcal{O}(|\delta\mathbf{x}|^2)$ .

**Theorem 1.** *The variation  $\delta\mathcal{S} = 0$  is solved by the equations*

$$\rho\ddot{\mathbf{x}} + \epsilon\mathbf{x}'''' - (\alpha\mathbf{x}')' = 0\tag{2.3}$$

*with the additional boundary conditions  $\mathbf{x}''(0) = \mathbf{x}''(L) = 0$ . It can be shown that  $\alpha$  satisfies*

$$\alpha'' - |\mathbf{x}''|^2 \alpha = -\rho |\dot{\mathbf{x}}'|^2 - \epsilon(4\mathbf{x}'''' \cdot \mathbf{x}'' + 3|\mathbf{x}''''|^2)\tag{2.4}$$

with the boundary conditions

$$\alpha'(0) = \rho \ddot{\mathbf{x}}(0) \cdot \mathbf{x}'(0), \quad \alpha'(L) = \rho \ddot{\mathbf{x}} \cdot \mathbf{x}'(L) \quad (2.5)$$

Similar equations can be found in [59, 60, 74].

Note that the equation for  $\alpha$  will have unique solutions if and only if the Sturm-Liouville operator

$$-\beta'' + |\mathbf{x}''|^2 \beta, \quad \beta'(0) = \beta'(L) = 0 \quad (2.6)$$

does not have zero in its spectrum. The physical interpretation of that is that the rod isn't a straight line (i.e.,  $\mathbf{x}''$  is not identically zero). In this circumstance the Sturm-Liouville operator will be positive definite.

## 2.2 Adding damping terms

Define the Hamiltonian to be

$$H = \int_0^L \frac{1}{2} \rho |\dot{\mathbf{x}}|^2 + \frac{1}{2} \epsilon |\mathbf{x}''|^2 ds \quad (2.7)$$

Then it can be shown that if (2.3) and

$$|\mathbf{x}'(s)|^2 = 1 \quad (2.8)$$

are satisfied (as well as under suitable boundary conditions) then,  $\dot{H} = 0$ . i.e., the Hamiltonian is conserved. In light of this we can imagine that if we introduce damping terms the new equations will yield  $\dot{H} \leq 0$ . We propose that the damped version of the Hamiltonian to be

$$\dot{H} = -\nu \int_0^L |\dot{\mathbf{x}}''|^2 ds \quad (2.9)$$

where  $\nu$  is the damping constant. It suggests that the damped version of this equation should be

$$\rho\ddot{\mathbf{x}} + \epsilon\mathbf{x}'''' - \nu\dot{\mathbf{x}}'' - (\alpha\mathbf{x}')' = 0 \quad (2.10)$$

The physical rationale for this particular damping term is explained in [52].

## 2.3 Space discretization and cubic splines

We approximate the path of the elastica by a cubic spline. Let  $N$  be some positive integer and set the space mesh size  $\eta = L/N$ . Then define

$$\mathbf{x}_j = \mathbf{x}(\eta j), \quad (0 \leq j \leq N) \quad (2.11)$$

Suppose that  $\mathbf{x}(s)$  is a cubic spline passing through these points. Let us denote the first and second derivatives of the spline

$$\mathbf{x}'_j = \mathbf{x}'(\eta j), \quad \mathbf{x}''_j = \mathbf{x}''(\eta j), \quad (0 \leq j \leq N) \quad (2.12)$$

It is well known that these derivatives can be computed using tridiagonal matrices. (See for example [10]). We obtain  $(N + 1) \times (N + 1)$  matrices  $\mathbf{D}_1$  and  $\mathbf{D}_2$  matrices such that

$$\mathbf{x}'' = \mathbf{D}_2\mathbf{x}, \quad \mathbf{x}' = \mathbf{D}_1\mathbf{x} \quad (2.13)$$

We may now discretize the kinetic, potential energies and constraints as

$$\mathcal{E} = \sum_{j=0}^N w_j \frac{1}{2} \rho |\dot{\mathbf{x}}_j|^2 = \frac{1}{2} \rho \eta \dot{\mathbf{x}}^\top \mathbf{W} \dot{\mathbf{x}} \quad (2.14)$$

$$\mathcal{V} = \sum_{j=0}^N w_j \frac{1}{2} \epsilon |\mathbf{x}_j''|^2 = \frac{1}{2} \epsilon \eta (\mathbf{D}_2 \mathbf{x})^\top \mathbf{W} \mathbf{D}_2 \mathbf{x} \quad (2.15)$$

$$\mathcal{C} = \sum_{j=0}^N w_j \frac{1}{2} \alpha_j |\mathbf{x}'_j|^2 = \frac{1}{2} \eta (\mathbf{D}_1 \mathbf{x})^\top \mathbf{W} (\alpha * \mathbf{D}_1 \mathbf{x}), \quad |\mathbf{x}'|^2 = 1 \quad (2.16)$$

where the Lagrange multiplier  $\alpha = [\alpha_0, \alpha_1, \dots, \alpha_N]$  and the weight  $w_j$  is  $\eta$  for  $j \neq 0, N$  and  $w_j$  is  $\frac{1}{2}\eta$  for  $j = 0, N$ . This is according to the trapezoidal rule for numerical integration, and  $\mathbf{W}$  is an  $(N+1) \times (N+1)$  diagonal matrix whose diagonal entries are 1 except for the top and bottom entries which are  $\frac{1}{2}$ . In here  $*$  denotes the pointwise multiplication of vectors. More explicitly

$$\alpha * \mathbf{x} = [\alpha_0 \mathbf{x}_0, \alpha_1 \mathbf{x}_1, \dots, \alpha_N \mathbf{x}_N] \quad (2.17)$$

$$\mathbf{x} * \mathbf{y} = [\mathbf{x}_0 \cdot \mathbf{y}_0, \mathbf{x}_1 \cdot \mathbf{y}_1, \dots, \mathbf{x}_N \cdot \mathbf{y}_0] \quad (2.18)$$

We can now apply Hamilton's principle, and obtain the equations

$$\rho \ddot{\mathbf{x}} = \mathbf{W}^{-1} (-\epsilon \mathbf{D}_2^\top \mathbf{W} \mathbf{D}_2 \mathbf{x} + \mathbf{D}_1^\top \mathbf{W} (\alpha * (\mathbf{D}_1 \mathbf{x}))) \quad (2.19)$$

$$(2.20)$$

with the restriction that

$$(\mathbf{D}_1 \mathbf{x}) * (\mathbf{D}_1 \mathbf{x}) = \mathbb{1}_{3(N+1)}. \quad (2.21)$$

For the damped equation we have that

$$\rho \ddot{\mathbf{x}} = \mathbf{W}^{-1} (-\epsilon \mathbf{D}_2^\top \mathbf{W} \mathbf{D}_2 \mathbf{x} + \mathbf{D}_1^\top \mathbf{W} (\alpha * (\mathbf{D}_1 \mathbf{x})) + \nu \mathbf{D}_2^\top \mathbf{W} \mathbf{D}_2 \dot{\mathbf{x}} - 6\nu \mathbf{D}_2^\top \mathbf{W} \mathbf{A}^{-1} \dot{\mathbf{z}}). \quad (2.22)$$

## 2.4 Time discretization that preserves the Hamiltonian

The goal here is to apply Hamilton's principle to find the ODE which solves for the  $m$ -dimensional vector  $\mathbf{x}$  using kinetic energy, potential energy, and constraints which are of the following form:

$$\mathcal{E}(\mathbf{x}) = \frac{1}{2} \dot{\mathbf{x}}^\top \mathbf{K} \dot{\mathbf{x}}, \quad (2.23)$$

$$\mathcal{V}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{V} \mathbf{x} + \mathbf{v} \cdot \mathbf{x}, \quad (2.24)$$

$$\mathcal{C}_k(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{C}_k \mathbf{x} + \mathbf{c}_k \cdot \mathbf{x} + \chi_k = 0, \quad 1 \leq k \leq M, \quad (2.25)$$

$$\mathcal{C}(\mathbf{x}) = \sum_{k=1}^M \alpha_k \mathcal{C}_k(\mathbf{x}). \quad (2.26)$$

**Remark.** Notice here that we have shifted the index on  $k$  to start from 1 instead of at 0.

In here  $\mathbf{K}$  and  $\mathbf{V}$  are positive definite matrices and  $\mathbf{C}_k$  for  $1 \leq k \leq M$  are symmetric matrices.  $\mathbf{c}_k$  for  $1 \leq k \leq M$  and  $\mathbf{v}$  are vectors and  $\chi_k$  for  $1 \leq k \leq M$  are scalars. All of these quantities are to be determined.

Hamilton's principle gives the equation

$$\mathbf{K} \ddot{\mathbf{x}} = -\nabla \mathcal{V}(\mathbf{x}) - \sum_{k=1}^M \alpha_k \nabla \mathcal{C}_k(\mathbf{x}). \quad (2.27)$$

and a damped version of the equation is

$$\mathbf{K} \ddot{\mathbf{x}} = -\nabla \mathcal{V}(\mathbf{x}) - \sum_{k=1}^M \alpha_k \nabla \mathcal{C}_k(\mathbf{x}) - \mathbf{g}(\dot{\mathbf{x}}), \quad (2.28)$$

where

$$\mathbf{g}(\dot{\mathbf{x}}) = \mathbf{G}(\dot{\mathbf{x}}) \quad (2.29)$$

is a damping term that satisfy  $\mathbf{G}$  is positive semi definite.

We now explicitly describe the full numerical scheme to solve the ODE. Pick some small time step  $h$ , and abusing notation, denote  $\mathbf{x}$  at  $t = nh$  by  $\mathbf{x}_n$ . The idea behind the time discretization is to find replacements for the gradients, that is, vector valued functions  $\mathbf{D}\mathcal{V}(\mathbf{x}_{n+1}, \mathbf{x}_{n-1})$  and  $\mathbf{D}\mathcal{C}_k(\mathbf{x}_{n+1}, \mathbf{x}_{n-1})$  so that the identities

$$(\mathbf{x}_{n+1} - \mathbf{x}_{n-1}) \cdot \mathbf{D}\mathcal{V}(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) = \mathcal{V}(\mathbf{x}_{n+1}) - \mathcal{V}(\mathbf{x}_{n-1}) \quad (2.30)$$

$$(\mathbf{x}_{n+1} - \mathbf{x}_{n-1}) \cdot \mathbf{D}\mathcal{C}_k(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) = \mathcal{C}_k(\mathbf{x}_{n+1}) - \mathcal{C}_k(\mathbf{x}_{n-1}) \quad (2.31)$$

hold with

$$\mathbf{D}\mathcal{V}(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) = \nabla\mathcal{V}\left(\frac{1}{2}(\mathbf{x}_{n+1} + \mathbf{x}_{n-1})\right) + O(h^2), \quad (2.32)$$

$$\mathbf{D}\mathcal{C}_k(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) = \nabla\mathcal{C}_k\left(\frac{1}{2}(\mathbf{x}_{n+1} + \mathbf{x}_{n-1})\right) + O(h^2). \quad (2.33)$$

In our case  $\mathcal{V}$  and  $\mathcal{C}_k$  are quadratic function and so it can be shown (see [52]) that the following choices work:

$$\mathbf{D}\mathcal{V}(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) = \mathbf{V}\left(\frac{1}{2}(\mathbf{x}_{n+1} + \mathbf{x}_{n-1})\right) + \mathbf{v} \quad (2.34)$$

$$\mathbf{D}\mathcal{C}_k(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) = \mathbf{C}_k\left(\frac{1}{2}(\mathbf{x}_{n+1} + \mathbf{x}_{n-1})\right) + \mathbf{c}_k \quad (2.35)$$

The discrete numerical scheme proposed is the following:



$$\begin{aligned}
& \mathbf{K} \left( \frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{h^2} \right) \\
&= -\mathbf{D}\mathcal{V}(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) - \sum_{k=1}^M \alpha_k \mathbf{D}C_k(\mathbf{x}_{n+1}, \mathbf{x}_{n-1}) \\
&\quad - \mathbf{g} \left( \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2h} \right), \tag{2.36}
\end{aligned}$$

$$\mathcal{C}_k(\mathbf{x}_{n+1}) = 0, \quad 1 \leq k \leq M. \tag{2.37}$$

As for initial conditions on the constraints we assume that  $\mathcal{C}_k(\mathbf{x}_0) = \mathcal{C}_k(\mathbf{x}_1) = 0$  for  $1 \leq k \leq M$ . This scheme requires solving  $M + m$  non linear equations for  $\mathbf{x}_{n+1} \in \mathbb{R}^{3(N+1)}$  and  $(\alpha_k)_{1 \leq k \leq M}$ . Note that  $M = N + 1$  and  $m = 3(N + 1)$ .

Now if one defines the discrete Hamiltonian by

$$H_{n+1} = \frac{1}{2h^2}(\mathbf{x}_{n+1} - \mathbf{x}_n)^\top \mathbf{K}(\mathbf{x}_{n+1} - \mathbf{x}_n) + \frac{1}{2}(\mathcal{V}(\mathbf{x}_{n+1}) + \mathcal{V}(\mathbf{x}_n)) \tag{2.38}$$

**Theorem 2.** *The solution to the difference schemes in (2.36) and (2.37) satisfies the discrete Hamiltonian in (2.38) and is nonincreasing or constant in the case of  $\mathbf{g} = \mathbf{0}$ .*

For a proof of this see [52]. It now follows that the difference scheme stays in a compact set. Hence the solution will be stable and will not *blow up*.

Next, we present the computational and implementation aspects of the algorithm. These computations are necessary for an effective implementation using software.

## 2.5 Computation of matrices $V, C_k$ etc

Energies and constraints are set up as

$$\mathcal{E}(\mathbf{x}) = \sum_{j=0}^N w_j \frac{1}{2} \rho |\dot{\mathbf{x}}_j|^2 = \frac{1}{2} \rho \eta \dot{\mathbf{x}}^\top \mathbf{W} \dot{\mathbf{x}} \quad (2.39)$$

$$\begin{aligned} \mathcal{V}(\mathbf{x}) &= \sum_{j=0}^N w_j \frac{1}{2} \epsilon |\mathbf{x}_j''|^2 \\ &= \frac{1}{2} \epsilon \eta (\mathbf{D}_2 \mathbf{x} - 6\mathbf{A}^{-1} \mathbf{z})^\top \mathbf{W} (\mathbf{D}_2 \mathbf{x} - 6\mathbf{A}^{-1} \mathbf{z}) \end{aligned} \quad (2.40)$$

$$\begin{aligned} \mathcal{C}(\mathbf{x}) &= \sum_{j=0}^N w_j \frac{1}{2} \alpha_j \left( |\mathbf{x}'_j|^2 - 1 \right) \\ &= \frac{1}{2} \eta (\mathbf{D}_1 \mathbf{x} + 3\mathbf{C}\mathbf{A}^{-1} \mathbf{z})^\top \mathbf{W} (\alpha * (\mathbf{D}_1 \mathbf{x} + 3\mathbf{C}\mathbf{A}^{-1} \mathbf{z})) - \sum_{j=0}^N w_j \frac{1}{2} \alpha_j \end{aligned} \quad (2.41)$$

In here  $\mathbf{W}$  is a  $(N+1) \times (N+1)$  diagonal matrix with top and bottom entries  $\frac{1}{2}$  and the rest are 1 and it captures the numerical integration constants that appears in the trapezoidal rule for numerical integration. (see for example [10]).

Our goals in here is to express the equations in (2.39), (2.40) and (2.41) in the form

$$\mathcal{E}(\mathbf{x}) = \frac{1}{2} \dot{\mathbf{x}}^\top \mathbf{K} \dot{\mathbf{x}} \quad (2.42)$$

$$\mathcal{V}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{V} \mathbf{x} + \mathbf{v} \cdot \mathbf{x} \quad (2.43)$$

$$\mathcal{C}_k(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{C}_k \mathbf{x} + \mathbf{c}_k \cdot \mathbf{x} + \chi_k, \quad 1 \leq k \leq M \quad (2.44)$$

$$\mathcal{C}(\mathbf{x}) = \sum_{k=1}^M \alpha_k \mathcal{C}_k(\mathbf{x}) \quad (2.45)$$

More precisely our goal now is to find the matrices  $\mathbf{K}, \mathbf{V}, \mathbf{C}_k, \mathbf{c}_k, \mathbf{v}$  and the scalars  $\chi_k$  for  $1 \leq k \leq M$ .

It is clear that we can set

$$\mathbf{K} = \rho\eta\mathbf{W} \quad (2.46)$$

### 2.5.1 Computing $\mathbf{V}$ and $\mathbf{v}$

Since  $\mathbf{x}'' = \mathbf{D}_2 - 6\mathbf{A}^{-1}\mathbf{z}$  we can write,

$$\begin{aligned} \mathcal{V}(\mathbf{x}) &= \frac{1}{2}\epsilon\eta \left[ (\mathbf{x}^\top \mathbf{D}_2^\top - 6\mathbf{z}^\top (\mathbf{A}^\top)^{-1}) (\mathbf{W}\mathbf{D}_2 - 6\mathbf{W}\mathbf{A}^{-1}\mathbf{z}) \right] \\ &= \frac{1}{2}\epsilon\eta \left[ \mathbf{x}^\top \mathbf{D}_2^\top \mathbf{W}\mathbf{D}_2^\top \mathbf{x} - 12\mathbf{z}^\top (\mathbf{A}^\top)^{-1} \mathbf{W}\mathbf{D}_2^\top \mathbf{x} + \text{constant terms} \right] \end{aligned} \quad (2.47)$$

$$= \frac{1}{2}\mathbf{x}^\top \mathbf{D}_2^\top (\epsilon\eta\mathbf{W}) \mathbf{D}_2 \mathbf{x} - 6\mathbf{z}^\top (\mathbf{A}^\top)^{-1} (\epsilon\eta\mathbf{W}) \mathbf{D}_2^\top \mathbf{x} + \text{constant terms} \quad (2.48)$$

Thus, we can now set

$$\mathbf{V} = \mathbf{D}_2^\top (\epsilon\eta\mathbf{W}) \mathbf{D}_2 \quad (2.49)$$

$$\mathbf{v} = -6\epsilon\eta (\mathbf{D}_2 \mathbf{W} \mathbf{A}^{-1})^\top \quad (2.50)$$

**Remark.** *In here we used the elementary fact that  $(\mathbf{A}^{-1})^\top = (\mathbf{A}^\top)^{-1}$ .*

### 2.5.2 Computing $\mathcal{C}_k$ , $\mathbf{c}_k$ and $\chi_k$

From (2.41) we can write

$$\mathcal{C}(\mathbf{x}) = \frac{1}{2}\eta (\mathbf{D}_1 + 3\mathbf{C}\mathbf{A}^{-1}\mathbf{z})^\top \mathbf{W}\mathbf{\Gamma} (\mathbf{D}_1 + 3\mathbf{C}\mathbf{A}^{-1}\mathbf{z}) - \sum_{k=1}^M w_k \frac{1}{2} \alpha_k \quad (2.51)$$

where  $\mathbf{\Gamma}$  is a diagonal matrix whose diagonal elements are  $\alpha_1, \alpha_2, \dots, \alpha_M$ . i.e.,

$$\mathbf{\Gamma} = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_M)$$

Now denote by  $W_\Gamma := W\Gamma$ . Clearly  $W_\Gamma$  is a diagonal matrix such that

$$W_\Gamma = \text{diag}(\frac{1}{2}\alpha_1, \alpha_2, \dots, \alpha_{M-1}, \frac{1}{2}\alpha_M).$$

We can now re-write (2.51) as follows:

$$\mathcal{C}(\mathbf{x}) = \frac{1}{2}\eta [\mathbf{x}^\top D_1^\top W_\Gamma D_1 \mathbf{x} + 3\mathbf{x}^\top D_1^\top W_\Gamma C A^{-1} \mathbf{z} + 3\mathbf{z}^\top (A^\top)^{-1} C^\top W_\Gamma D_1 \mathbf{x}] \quad (2.52)$$

$$+ \frac{9\eta}{2} (C A^{-1} \mathbf{z})^\top W_\Gamma (C A^{-1} \mathbf{z}) - \sum_{k=1}^M w_k \frac{1}{2} \alpha_k \quad (2.53)$$

$$= \frac{1}{2} \mathbf{x}^\top (D_1^\top (\eta W_\Gamma) D_1) \mathbf{x} + 3(C A^{-1} \mathbf{z})^\top (\eta W_\Gamma) D_1 \mathbf{x} \quad (2.54)$$

$$+ \frac{9}{2} (C A^{-1} \mathbf{z})^\top (\eta W_\Gamma) (C A^{-1} \mathbf{z}) - \sum_{k=1}^M w_k \frac{1}{2} \alpha_k \quad (2.55)$$

Notice that we can write

$$\begin{aligned} & D_1 (\eta W_\Gamma) D_1 \\ &= \eta D_1^\top \left[ \text{diag}(\frac{1}{2}\alpha_1, 0, \dots, 0) + \text{diag}(0, \alpha_2, 0, \dots, 0) + \dots + \text{diag}(0, 0, \dots, 0, \frac{1}{2}\alpha_M) \right] D_1 \\ &= \eta D_1^\top \left( \sum_{k=1}^M \alpha_k W_{\Gamma_k} \right) D_1 \end{aligned} \quad (2.56)$$

where  $W_{\Gamma_k}$  is a series of diagonal matrices whose precise definition is as follows.

$$W_{\Gamma_k} = \begin{cases} \text{diag}(\frac{1}{2}, 0, 0, \dots, 0) & \text{for } k = 1 \\ \text{diag}(0, 0, \dots, 0, 1, 0, \dots, 0) & \text{for } 1 < k < M, \\ \text{diag}(0, 0, \dots, 0, \frac{1}{2}) & \text{for } k = M. \end{cases}$$

**Remark.** In here the diagonal matrix for the case  $1 < k < M$  is one that has a 1 in

its  $k$ th position and zeros everywhere else.

We can similarly show that

$$3(\mathbf{CA}^{-1}\mathbf{z})^\top (\eta \mathbf{W}_\Gamma) \mathbf{D}_1 = 3\eta (\mathbf{CA}^{-1}\mathbf{z})^\top \left( \sum_{k=1}^M \alpha_k \mathbf{W}_{\Gamma_k} \right) \mathbf{D}_1 \quad (2.57)$$

and

$$\frac{9}{2} (\mathbf{CA}^{-1}\mathbf{z})^\top (\eta \mathbf{W}_\Gamma) (\mathbf{CA}^{-1}\mathbf{z}) = \frac{9\eta}{2} (\mathbf{CA}^{-1}\mathbf{z})^\top \left( \sum_{k=1}^M \alpha_k \mathbf{W}_{\Gamma_k} \right) \mathbf{CA}^{-1}\mathbf{z} \quad (2.58)$$

Thus, we can now set

$$\mathbf{C}_k = \eta \mathbf{D}_1^\top \mathbf{W}_{\Gamma_k} \mathbf{D}_1, \quad (2.59)$$

$$\mathbf{c}_k = 3\eta (\mathbf{CA}^{-1}\mathbf{z})^\top \mathbf{W}_{\Gamma_k} \mathbf{D}_1, \quad (2.60)$$

$$\chi_k = \frac{9}{2} \eta (\mathbf{CA}^{-1}\mathbf{z})^\top \mathbf{W}_{\Gamma_k} (\mathbf{CA}^{-1}\mathbf{z}) - \frac{1}{2} w_k, \quad (2.61)$$

for  $1 \leq k \leq M$ . Note that we are now able to write  $\mathcal{C}(\mathbf{x})$  in the intended summation form as follows:

$$\mathcal{C}(\mathbf{x}) = \sum_{k=1}^M \alpha_k \mathcal{C}_k(\mathbf{x}) \quad (2.62)$$

where

$$\mathcal{C}_k(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{C}_k \mathbf{x} + \mathbf{c}_k \cdot \mathbf{x} + \chi_k \quad \text{for } 1 \leq k \leq M. \quad (2.63)$$

## 2.6 System of equations and the associated Jacobian

In light of the previous computations we can now write the equations of motion in a manner that is suitable for computer simulation. Indeed,

$$\begin{aligned} & \mathbf{K} \left( \frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{h^2} \right) + \frac{1}{2}\mathbf{V}(\mathbf{x}_{n+1} + \mathbf{x}_{n-1}) + \mathbf{v} \\ & + \sum_{k=1}^M \alpha_k \left[ \frac{1}{2}\mathbf{C}_k(\mathbf{x}_{n+1} + \mathbf{x}_{n-1}) + \mathbf{c}_k \right] + \mathbf{G} \left( \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2h} \right) = \mathbf{0} \end{aligned} \quad (2.64)$$

$$\frac{1}{2}\mathbf{x}_{n+1}^\top \mathbf{C}_k \mathbf{x}_{n+1} + \mathbf{c}_k^\top \mathbf{x}_{n+1} + \chi_k = 0 \quad \text{for } 1 \leq k \leq M \quad (2.65)$$

Let us also discuss here the initial conditions. These are normally specified by assuming we know  $\mathbf{x} = \mathbf{F}$  and  $\dot{\mathbf{x}} = \mathbf{G}$  at  $t = 0$  where  $\mathbf{F}$  and  $\mathbf{G}$  are given. To guarantee the method is still order 2 in time the discrete version of these conditions are

$$\mathbf{x}_0 = \mathbf{F} \quad (2.66)$$

$$\mathbf{x}_1 - \mathbf{x}_{-1} = 2h\mathbf{G} \quad (2.67)$$

Our goal here is to solve for  $\mathbf{x}_{n+1}$  the system of equations in (2.64) and (2.65). It is clear that  $\mathbf{x}_0$  is immediately known because of (2.66). To find  $\mathbf{x}_1$  we can substitute (2.67) in (2.64) and (2.65) with  $n = 0$ . To find  $\mathbf{x}_{n+1}$  for  $n = 1, 2, \dots$  we can use an iterative algorithm which uses the previous known value for  $\mathbf{x}_n$  to solve for  $\mathbf{x}_{n+1}$  at each iterative step.

Note that at each step we are solving a non linear system of equations of order 2. To effectively use Python programming to carry out this computation we use the Levenberg-Marquardt algorithm. To make the algorithm converge faster to solutions

it is important for us to provide an analytical Jacobian rather than let the solver numerically approximate the Jacobian at each iteration. The reason for this is that the algorithm uses a least squares method to find the root of the system of equation and it numerically approximates the Jacobian of at each call to the function. The downside is that it will make thousands of such calls for computing a single root and thus making it a computationally expensive process.

The way to make this algorithm more efficient we can instead provide the analytical Jacobian explicitly. Let  $\mathbf{J}$  be the Jacobian of the system of equations in (2.64) and (2.65). Let us denote by  $f(\mathbf{x}_{n+1}, \alpha) = \mathbf{0}$  the system of equations in (2.64) and by  $g(\mathbf{x}_{n+1}, \alpha) = 0$  the system of equations in (2.65). Then  $\mathbf{J}$  is a block matrix of the form

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}_{n+1}} & \frac{\partial f}{\partial \alpha} \\ \frac{\partial g}{\partial \mathbf{x}_{n+1}} & \frac{\partial g}{\partial \alpha} \end{bmatrix} \quad (2.68)$$

where,

$$\frac{\partial f}{\partial \mathbf{x}_{n+1}} = \frac{1}{h^2} \mathbf{K} + \frac{1}{2} \mathbf{V} + \sum_{k=1}^M \frac{1}{2} \alpha_k \mathbf{C}_k + \frac{1}{2h} \mathbf{G} \quad (2.69)$$

$$\frac{\partial f}{\partial \alpha} = \left[ \frac{1}{2} \mathbf{C}_k (\mathbf{x}_{n+1} + \mathbf{x}_{n-1}) + \mathbf{c}_k \right]_{1 \leq k \leq M} \quad (2.70)$$

$$\frac{\partial g}{\partial \mathbf{x}_{n+1}} = \left[ \mathbf{x}_{n+1}^\top \mathbf{C}_k + \mathbf{c}_k^\top \right]_{1 \leq k \leq M} \quad (2.71)$$

$$\frac{\partial g}{\partial \alpha} = \mathbf{0} \quad (2.72)$$

It is helpful for computational purposes to explicitly state the dimensions of these matrix blocks. We have,

- $\frac{\partial f}{\partial \mathbf{x}_{n+1}}$  is an  $3(N+1) \times 3(N+1)$  matrix

- $\frac{\partial f}{\partial \alpha}$  is an  $3(N + 1) \times (N + 1)$  matrix
- $\frac{\partial g}{\partial \mathbf{x}_{n+1}}$  is an  $(N + 1) \times 3(N + 1)$  matrix
- $\frac{\partial g}{\partial \alpha}$  is an  $(N + 1) \times (N + 1)$  zero matrix

which results in  $\mathbf{J}$  being a  $4(N + 1) \times 4(N + 1)$  matrix as expected.

The complete software re-implementation of this problem written in Python 3 can be found on the following site. <https://github.com/dilanfd/dynamics-of-springs>





# Chapter 3

## Springs with torsion

### 3.1 Preliminaries

Beginning this chapter we discuss the dynamic *Kirchhoff* problem, that is, the behavior of general springs that have a natural curvature, and for which the equations take into account the (mechanical) torsion of the rod. It turns out that the major contributor to the potential energy stored in the spring is as a result of torsion and not the side to side twisting of the spring as one might expect. Our goal is to build an efficient and robust numerical scheme and consequently implement the said scheme using the computer programming language Python 3.

A complete mathematical description of the Kirchhoff problem can be found in the work of Love [43]. This work however was done much earlier than when computers were even invented and so the problem of numerical modeling of such behavior has not been addressed.

Thomson (the brother of Lord Kelvin) [75] also looked specifically at helical springs and established geometric relationships between the various spring parameters such as pitch angle, static load, coil curvature and the radius of the spring helix. Many later works that model spring behavior [6, 13, 16, 19, 37–39, 57, 63, 73, 77, 78] make

simplifying assumptions. For example, they usually assume that the springs are in a helical shape with constant pitch angle and radius of the spring helix and they only model their static behavior.

## 3.2 The transported natural curvature and the natural frame.

The key to the Kirchhoff problem is to find a way to describe the natural curvature of the dynamic spring.

Let us assume that the equation describing the spring is given by function

$$\mathbf{x}(s) : [0, L] \rightarrow \mathbb{R}^3$$

where,

$$\mathbf{x}(s) = [x_1(s), x_2(s), x_3(s)]^T \tag{3.1}$$

and  $L$  is the arc length of the spring and  $s$  is the arc length of the spring along the rod. Hence the function  $x(s)$  satisfies the equation

$$|\mathbf{x}'(s)|^2 = x_1'(s)^2 + x_2'(s)^2 + x_3'(s)^2 = 1 \tag{3.2}$$

In here the prime denotes derivative with respect to the arc length. Note that we have suppressed the time dependence of the equation. For now we may assume that the function  $\mathbf{x}$  is for a fixed time. We will use the usual convention of denoting the derivative with respect to time by putting a dot over the variable.

We define the *curvature* of the spring to be the vector  $\mathbf{x}''(s)$ .

**Remark.** *The traditional definition for the curvature  $\kappa$  of a space curve is defined*

as

$$\kappa = \left| \frac{d\mathbf{T}}{ds} \right|$$

where  $\mathbf{T}$  is the unit tangent vector given by

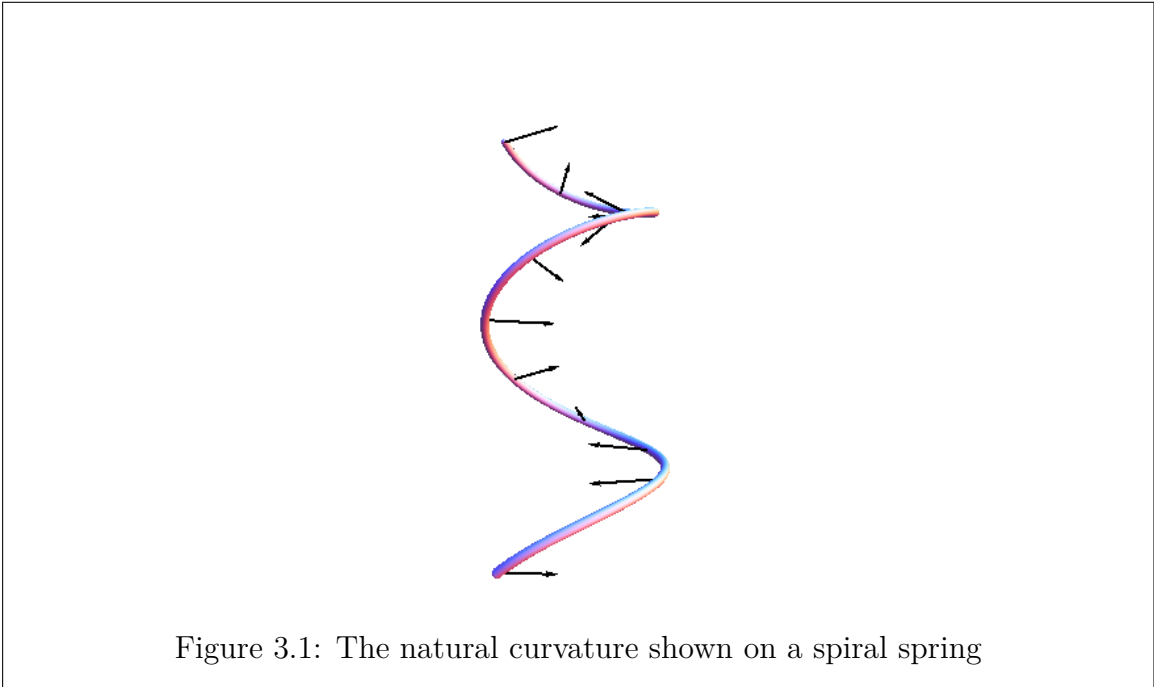
$$\mathbf{T} = \frac{\mathbf{x}'(s)}{|\mathbf{x}'(s)|}.$$

By (3.2), we have  $\kappa = |\mathbf{x}''(s)|$ .

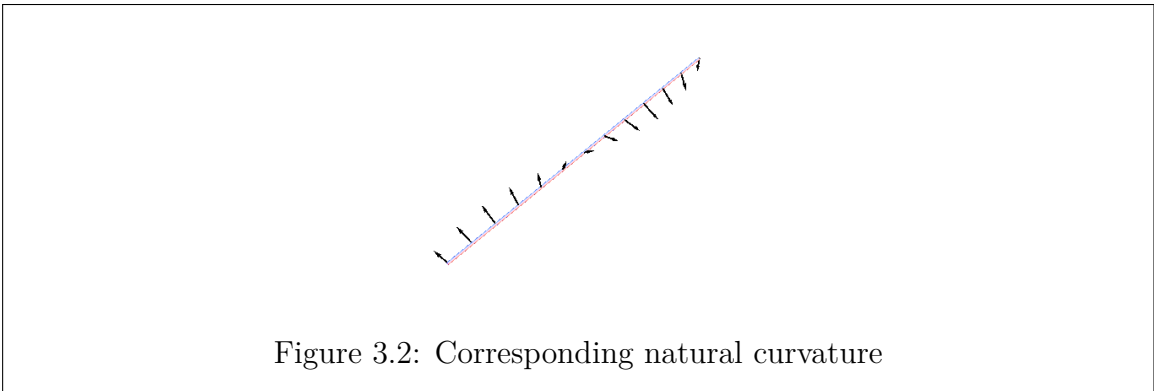
The above remark shows that it makes sense for us to define *curvature* the way we have defined it. By (3.2) again we have  $\mathbf{x}'(s) \cdot \mathbf{x}'(s) = 1$ . Taking derivatives with respect to  $s$  of this equation gives us  $\mathbf{x}'(s) \cdot \mathbf{x}''(s) = 0$ . This shows that  $\mathbf{x}''(s)$  is always perpendicular to the tangent vector  $\mathbf{x}'(s)$ .

When the spring is at rest, and has no forces applied to it, we will denote this state of the spring  $\mathbf{x}$  as  $\mathbf{x}_0$  the path of the spring. (This does not necessarily have to be the path of the spring at time  $t = 0$ ). We call the function  $\mathbf{x}_0''(s)$  the *natural curvature*. Then we try to determine part of the difference in potential energy by seeing how much  $\mathbf{x}''(s)$  differs from  $\mathbf{x}_0''(s)$ . However we cannot actually compute this difference because  $\mathbf{x}(s)$  and  $\mathbf{x}_0(s)$  are at different points.

Thus, in order to have a point of comparison define the concept of *transported natural curvature*. Suppose the spring is at rest state and attach to each point on the spring the curvature vector  $\mathbf{x}_0''(s)$ .



This is illustrated in Figure 3.1. We then stretch out the spring until it is completely straight and we do so without introducing any internal twisting or torsion to the rod. Note that this is only a mathematical idealization. In reality such a maneuver will add internal twisting to the spring.



We then obtain what is illustrated in Figure 3.2. The curvature vectors we previously attached to the spring now move with the points on the spring but they remain perpendicular to the now straight rod. Let us assume that the point  $\mathbf{x}_0(s)$  is

transported to the point  $[s, 0, 0]^T$ , i.e., the spring is stretched out along the positive  $x$ -axis.

**Notation.** Let  $\boldsymbol{\kappa}(s)$  denote the transported natural curvature at  $[s, 0, 0]$ .

Observe that only the second and third coordinates of  $\boldsymbol{\kappa}(s)$  are non zero. This is because for each transported natural curvature vector must have tail coordinates of the form  $(s, 0, 0)$  and its head must have coordinates of the form  $(s, y_s, z_s)$  and so the direction vector must be of the form  $[0, y_s, z_s]$ . To carry out the straightening of the rod we introduce the our next subject.

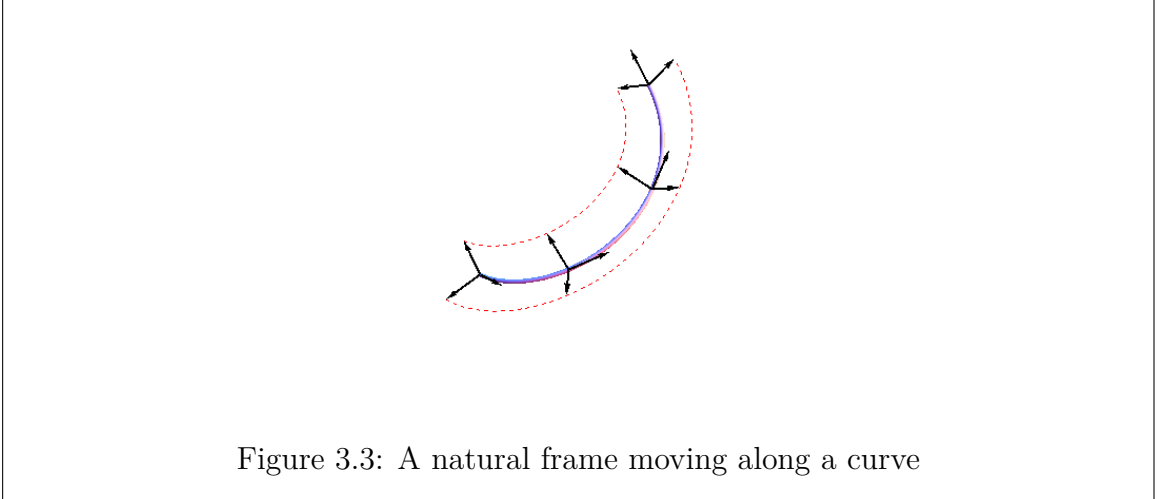
### 3.3 Natural Frames

A *frame* is a triple of vectors  $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$  that are orthonormal. By definition it follows that the square matrix  $\mathbf{R}$  is such that  $\mathbf{R}\mathbf{R}^T = \mathbf{I}_3$  where  $\mathbf{I}_3$  denotes the  $3 \times 3$  identity matrix. In other words the matrix  $\mathbf{R}$  is orthogonal and in particular  $\mathbf{R}^{-1} = \mathbf{R}^T$ .

We now introduce the concept of a *natural frame*. Pick any frame  $\mathbf{R}_0(s)$  attached to the point  $x_0(s)$ . Suppose now, that we move tangent to the spring to the point  $\mathbf{x}_0(s)$  with minimal amount of rotation necessary. Denote by  $\mathbf{R}_0(s)$  the frame that's been carried to the point  $\mathbf{x}_0(s)$ . Once we have chosen  $\mathbf{R}_0(0)$ , the frame  $\mathbf{R}_0(s)$  is uniquely determined. Indeed, for the sake of convenience let us agree that we chose  $\mathbf{R}_0(0)$  as follows:

$$\mathbf{R}_0(0) = \begin{bmatrix} | & | & | \\ \mathbf{x}'_0(0) & \mathbf{m}_2 & \mathbf{m}_3 \\ | & | & | \end{bmatrix}$$

In particular the first column of  $\mathbf{R}_0(0)$  is chosen to be  $\mathbf{x}'_0(0)$  whereas  $\mathbf{m}_2$  and  $\mathbf{m}_3$  are chosen arbitrarily so long as they adhere to the obvious restriction that  $\mathbf{R}_0(0)$  be a frame (i.e., orthogonal). A picture of how a natural frame evolves along the frame is shown in Figure 3.3.



Each column of the natural frame  $\mathbf{R}_0(s)$  can now be computed by solving for  $\mathbf{r}(s)$  the differential equation

$$\mathbf{r}'(s) = (\mathbf{x}'_0(s) \times \mathbf{x}''_0(s)) \times \mathbf{r}(s) \quad (3.3)$$

where  $\mathbf{r}(s)$  denotes an element(column) of  $\mathbf{R}_0(s)$ . We are of course, using each respective column of  $\mathbf{R}_0(0)$  as an initial condition in the solution of ((3.3)).

Note that  $\mathbf{x}'(s)$  is a solution of (3.3). Indeed, if we set  $\mathbf{r}(s) = \mathbf{x}'_0(s)$  on the right hand side of (3.3) we get,

$$\begin{aligned} & \mathbf{x}''_0(s) (\mathbf{x}'_0(s) \cdot \mathbf{x}'_0(s)) - \mathbf{x}'_0(s) (\mathbf{x}'_0(s) \cdot \mathbf{x}''_0(s)) \\ &= \mathbf{x}''_0(s) (1) - \mathbf{x}'_0(s) (0) \\ &= \mathbf{x}''(s) \end{aligned}$$

which is precisely the left hand side of (3.3) with  $\mathbf{r}(s) = \mathbf{x}'(s)$ . It is now clear that the first element(column) of  $\mathbf{R}_0(s)$  will always be  $\mathbf{x}'_0(s)$ . This concept appears in literature as the so called *Fermi–Walker transport* in General Relativity. [46].

We can now write the equation for natural curvature as

$$\boldsymbol{\kappa}(s) = \mathbf{R}_0(s)^{-1} \mathbf{x}_0''(s) = \mathbf{R}_0^T \mathbf{x}_0''(s) \quad (3.4)$$

To describe the spring as it is undergoing motion, it is important not only to decide where each point is, but also how it is oriented there. So we denote by  $\mathbf{R}(s)$  the frame consisting of three vectors in  $\mathbf{R}_0(s)$  transported from  $\mathbf{x}_0(s)$ . Note that we must have that the first element of  $\mathbf{R}(s)$  be  $\mathbf{x}'(s)$ . We are now in a position to see how the curvature of  $\mathbf{x}$  and the curvature of  $\mathbf{x}_0$  differ. We can compare them by using the frames we have introduced. The difference we need to measure is

$$\mathbf{R}^{-1} \mathbf{x}'' - \mathbf{R}_0^{-1} \mathbf{x}_0'' = \mathbf{R}^\top \mathbf{x}'' - \boldsymbol{\kappa}_0.$$

In here we use the subscript 0 on  $\boldsymbol{\kappa}$  to suggest that it is the *natural curvature*, the curvature of the spring at point  $s$  at rest. (Note that the rest state does not necessarily have to be at time  $t = 0$ ).

For a general spring, the transported natural curvature can easily be computed numerically. In fact, for a helical spring the transported natural curvature can even be computed analytically. In the next section we show how this can be done.



## 3.4 Helical springs

### 3.4.1 Notations and preliminaries

First let us introduce some useful notation. For any vector  $\mathbf{v} \in \mathbb{R}^3$ , where  $\mathbf{v} = [v_1, v_2, v_3]^T$  define the anti symmetric matrix

$$\Omega_{\mathbf{v}} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

Notice that for any  $\mathbf{v}, \mathbf{w} \in \mathbb{R}^3$  we have

$$\Omega_{\mathbf{v}} \mathbf{w} = \mathbf{v} \times \mathbf{w}$$

We now prove a couple of useful lemmas.

**Proposition 3.** *For  $s \in \mathbb{R}$ , we have*

$$\exp(s\Omega_{\mathbf{i}}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(s) & -\sin(s) \\ 0 & \sin(s) & \cos(s) \end{bmatrix}$$

where  $\mathbf{i} = [1, 0, 0]^T$ .

*Proof.* Note that

$$s\Omega_{\mathbf{i}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -s \\ 0 & s & 0 \end{bmatrix}$$

Denote  $M = s\Omega_{\mathbf{i}}$  and let  $I$  denote the  $3 \times 3$  identity matrix. Now note that  $M^3 = -s^2M$ . Therefore we must have that  $M^{2n+1} = (-1)^n s^{2n}M$  and  $M^{2n} =$

$(-1)^{n-1}s^{2n-2}M^2$  for  $n \in \mathbb{N}$ . Therefore,

$$\begin{aligned}
\exp(M) &= I + \sum_{k=1}^{\infty} \frac{M^k}{k!} \\
&= I + \sum_{n=0}^{\infty} \frac{M^{2n+1}}{(2n+1)!} + \sum_{n=1}^{\infty} \frac{M^{2n}}{(2n)!} \\
&= I + \sum_{n=0}^{\infty} \frac{(-1)^n s^{2n} M}{(2n+1)!} + \sum_{n=1}^{\infty} \frac{(-1)^{n-1} s^{2n-2} M^2}{(2n)!} \\
&= I + \frac{M}{s} \sum_{n=0}^{\infty} \frac{(-1)^n s^{2n+1}}{(2n+1)!} - \frac{M^2}{s^2} \sum_{n=1}^{\infty} \frac{(-1)^n s^{2n}}{(2n)!} \\
&= I + \frac{\sin(s)}{s} M + \frac{1 - \cos(s)}{s^2} M^2 \\
&= I + \sin(s)\Omega_{\mathbf{i}} + \frac{1 - \cos(s)}{s^2} M^2
\end{aligned}$$

It now easily follows that  $\exp(s\Omega_{\mathbf{i}})$  has the intended matrix form.  $\square$

**Notation.** Let  $\text{SO}(3)$  denote the special orthogonal group of order 3.

**Proposition 4.** For any  $\mathbf{r} \in \text{SO}(3)$  and for  $\mathbf{i} = [1, 0, 0]^T$  we must have

$$\mathbf{r}\Omega_{\mathbf{i}}\mathbf{r}^{-1} = \Omega_{\mathbf{i}} \tag{3.5}$$

*Proof.* The canonical form for any  $\mathbf{r} \in \text{SO}(3)$  is

$$\mathbf{r} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & p & t \\ 0 & q & u \end{bmatrix}$$

Also note that since  $\mathbf{r} \in \text{SO}(3)$  we have  $\mathbf{r}^{-1} = \mathbf{r}^T$  and  $\det(\mathbf{r}) = 1$ . In particular

$pu - qt = 1$ . Thus,

$$\begin{aligned}
 r\Omega_i r^{-1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -pu + tq \\ 0 & -qt + pu & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \\
 &= \Omega_{ri}
 \end{aligned}$$

□

### 3.4.2 Analytical calculation of transported natural curvature for a helical spring

Suppose that a helical spring is given by

$$\mathbf{x}_0(s) = [\alpha s, \beta \cos(\gamma s), \beta \sin(\gamma s)]^T \quad (3.6)$$

where  $\alpha^2 + \beta^2 \gamma^2 = 1$ . Clearly

$$\mathbf{x}'_0(s) = [\alpha, -\gamma \beta \sin(\gamma s), \gamma \beta \cos(\gamma s)] \quad (3.7)$$

and

$$\mathbf{x}''_0(s) = [0, -\gamma^2 \beta \cos(\gamma s), -\gamma^2 \beta \sin(\gamma s)]. \quad (3.8)$$

Then,

$$\begin{aligned}
\mathbf{x}'_0(s) \times \mathbf{x}''_0(s) &= \langle \beta^2 \gamma^3, \alpha \beta \gamma^2 \sin(\gamma s), -\alpha \beta \gamma^2 \cos(\gamma s) \rangle \\
&= \gamma \mathbf{i} - \alpha \gamma \mathbf{x}'_0(s) \\
&= \gamma \mathbf{i} - \mu \mathbf{x}'_0(s)
\end{aligned} \tag{3.9}$$

where  $\mu = \alpha \gamma$ .

For the sake of convenience let us recast (3.3) as a matrix valued differential equation in the obvious way as follows.

$$\mathbf{r}'_0(s) = \Omega_{\mathbf{v}} \mathbf{r}_0(s) \tag{3.10}$$

where  $\mathbf{v} = \mathbf{x}'_0(s) \times \mathbf{x}''_0(s)$  and  $\mathbf{r}_0(0) = R_0(0)$ . Note that,

$$\Omega_{\mathbf{x}'_0(s) \times \mathbf{x}''_0(s)} = \begin{bmatrix} 0 & \alpha \beta \gamma^2 \cos(\gamma s) & \alpha \beta \gamma^2 \sin(\gamma s) \\ -\alpha \beta \gamma^2 \cos(\gamma s) & 0 & -\beta^2 \gamma^3 \\ -\alpha \beta \gamma^2 \sin(\gamma s) & \beta^2 \gamma^3 & 0 \end{bmatrix}. \tag{3.11}$$

In light of (3.9), we claim the following.

**Proposition 5.** *The solution to the matrix valued differential equation in (3.10) is given by*

$$\mathbf{r}_0(s) = \exp(\gamma s \Omega_{\mathbf{i}}) \mathbf{r}_0(0) \exp(-\mu s \Omega_{\mathbf{i}}) \tag{3.12}$$

*Proof.* By definition  $\mathbf{r}_0 \in \text{SO}(3)$ . Taking derivatives with respect to  $s$  in (3.12) we

get,

$$\begin{aligned} \mathbf{r}'_0 &= \gamma \Omega_{\mathbf{i}} \exp(\gamma s \Omega_{\mathbf{i}}) \mathbf{r}_0(0) \exp(-\mu s \Omega_{\mathbf{i}}) \\ &\quad - \mu \exp(\gamma s \Omega_{\mathbf{i}}) \mathbf{r}_0(0) \exp(-\mu s \Omega_{\mathbf{i}}) \Omega_{\mathbf{i}} \end{aligned} \quad (3.13)$$

$$= \gamma \Omega_{\mathbf{i}} \mathbf{r}_0 - \mu \mathbf{r}_0 \Omega_{\mathbf{i}} \quad (3.14)$$

$$\mathbf{r}'_0 \mathbf{r}_0^{-1} = \gamma \Omega_{\mathbf{i}} - \mu (\mathbf{r}_0 \Omega_{\mathbf{i}} \mathbf{r}_0^{-1}) \quad (3.15)$$

$$= \gamma \Omega_{\mathbf{i}} - \mu \Omega_{\mathbf{r}_0 \mathbf{i}} \quad (3.16)$$

Note that  $\mathbf{r}_0(s) \mathbf{i}$  is the first column of  $\mathbf{r}_0(s)$  and we know that the first column of  $\mathbf{r}_0(s)$  must be  $\mathbf{x}'_0(s)$ . Thus,  $\mathbf{r}_0(s) \mathbf{i} = \mathbf{x}'_0(s)$ . It now follows from (3.16) that

$$\begin{aligned} \mathbf{r}'_0 \mathbf{r}_0^{-1} &= \gamma \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} - \mu \begin{bmatrix} 0 & -\beta \gamma \cos(\gamma s) & -\beta \gamma \sin(\gamma s) \\ \beta \gamma \cos(\gamma s) & 0 & -\alpha \\ \beta \gamma \sin(\gamma s) & \alpha & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \alpha \beta \gamma^2 \cos(\gamma s) & \alpha \beta \gamma^2 \sin(\gamma s) \\ -\alpha \beta \gamma^2 \cos(\gamma s) & 0 & \alpha^2 \gamma - \gamma \\ -\alpha \beta \gamma^2 \sin(\gamma s) & \gamma - \alpha^2 \gamma & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \alpha \beta \gamma^2 \cos(\gamma s) & \alpha \beta \gamma^2 \sin(\gamma s) \\ -\alpha \beta \gamma^2 \cos(\gamma s) & 0 & -\beta^2 \gamma^3 \\ -\alpha \beta \gamma^2 \sin(\gamma s) & \beta^2 \gamma^3 & 0 \end{bmatrix} \\ &= \Omega_{\mathbf{x}'_0(s) \times \mathbf{x}''_0(s)} \end{aligned}$$

It is now clear that (3.12) is a solution of (3.10). □

**Proposition 6.** *The transported natural curvature denoted by  $\kappa(s)$  for the helical*

spring given in (3.6) is given by

$$\boldsymbol{\kappa}(s) = |\beta\gamma^2| \begin{bmatrix} 0 \\ \cos(\alpha\gamma(s - s_0)) \\ \sin(\alpha\gamma(s - s_0)) \end{bmatrix} \quad (3.17)$$

where  $s_0$  depends on the choice of the second and third elements (columns) of  $\mathbf{R}_0(0)$ .

*Proof.* Let us choose  $\mathbf{r}_0(0)$  the following way. First column is  $\mathbf{x}'_0(0)$ , second column is  $\frac{\mathbf{x}''_0(0)}{|\mathbf{x}''_0(0)|}$  and its third column is some arbitrary vector  $\mathbf{m}_3$ . (Actually  $\mathbf{m}_3$  is not completely arbitrary we still need to chose it in such a way that  $\mathbf{r}_0(0) \in \text{SO}(3)$ ). Since  $\boldsymbol{\kappa}(0) = \mathbf{r}_0(0)^T \mathbf{x}''_0(0)$  and noticing that  $\mathbf{x}''_0(0)$  is a column vector we get,

$$\begin{aligned} \boldsymbol{\kappa}(0) &= \begin{bmatrix} \mathbf{x}'_0(0) \cdot \mathbf{x}''_0(0) \\ \frac{\mathbf{x}''_0(0)}{|\mathbf{x}''_0(0)|} \cdot \mathbf{x}''_0(0) \\ \mathbf{m}_3 \cdot \mathbf{x}''_0(0) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ |\mathbf{x}''_0(0)| \\ \mathbf{m}_3 \cdot \mathbf{x}''_0(0) \end{bmatrix} \end{aligned} \quad (3.18)$$

Notice that  $\mathbf{x}''_0(0) = [0, -\gamma^2\beta, 0]$  and so  $|\mathbf{x}''_0(0)| = |\gamma^2\beta|$ . Also notice that  $\mathbf{m}_3 \cdot \mathbf{x}''_0(0) = m_2(-\gamma^2\beta)$  where  $m_2$  is the  $\mathbf{j}$  component of vector  $\mathbf{m}_3$ . Since  $\mathbf{r}_0(0)$  is anti-symmetric we must have that  $-m_2$  is equal to the  $\mathbf{k}$  component of the 2nd row of  $\mathbf{r}_0(0)$  which is incidentally zero in this case. i.e.,  $m_2 = 0$ . Thus,

$$\boldsymbol{\kappa}(0) = \begin{bmatrix} 0 \\ |\gamma^2\beta| \\ 0 \end{bmatrix} \quad (3.19)$$

We can now write  $\mathbf{r}_0(s)^T = \exp(\mu s \Omega_{\mathbf{i}}) \mathbf{r}_0(0)^T \exp(-\gamma s \Omega_{\mathbf{i}})$ . Also by explicit computa-

tion we see that  $\exp(-\gamma s \Omega_{\mathbf{i}}) \mathbf{x}_0''(s) = \mathbf{x}_0''(0)$ . Thus,

$$\boldsymbol{\kappa}(s) = \exp(\mu s \Omega_{\mathbf{i}}) \mathbf{r}_0(0)^T \exp(-\gamma s \Omega_{\mathbf{i}}) \mathbf{x}_0''(s) \quad (3.20)$$

$$= \exp(\mu s \Omega_{\mathbf{i}}) \mathbf{r}_0(0)^T \mathbf{x}_0''(0) \quad (3.21)$$

$$= \exp(\mu s \Omega_{\mathbf{i}}) \boldsymbol{\kappa}(0) \quad (3.22)$$

$$= \exp(\mu s \Omega_{\mathbf{i}}) \begin{bmatrix} 0 \\ |\gamma^2 \beta| \\ 0 \end{bmatrix} \quad (3.23)$$

$$= |\gamma^2 \beta| \begin{bmatrix} 0 \\ \cos(\alpha \gamma s) \\ \sin(\alpha \gamma s) \end{bmatrix} \quad (3.24)$$

□

# Chapter 4

## Dynamic equations of a spring

### 4.1 Introducing the variables

A spring is described by the two quantities

$$\mathbf{x}(s) \quad \text{and} \quad R(s) \tag{4.1}$$

where  $s \in [0, L]$  is arc length. Here  $\mathbf{x} \in \mathbb{R}^3$  describes the curve in 3D space, and  $R(s)$  is the rotation matrix we have introduced earlier. By the manner in which we have defined the  $R(s)$  matrix we know that  $R(s) \in \text{SO}(3)$  and in particular  $\mathbf{x}$  and  $R$  are related by

$$\mathbf{x}' = R \mathbf{i} \tag{4.2}$$

Let  $\mathbf{u} = [u_1, u_2, u_3] \in \mathbb{R}^3$  and a matrix  $M \in \mathcal{M}_{3 \times 3}(\mathbb{R})$  with columns  $[\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3]$ .



**Definition 1.** Define their cross product as follows

$$\mathbf{u} \times M := [\mathbf{u} \times_1, \mathbf{u} \times_2, \mathbf{u} \times_3] = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} M \quad (4.3)$$

Since  $R \in \text{SO}(3)$  we have that there exists vectors  $\boldsymbol{\Omega}$  and  $\mathbf{T}$  such that

$$\dot{R} = \boldsymbol{\Omega} \times R \quad (4.4)$$

$$R' = \mathbf{T} \times R \quad (4.5)$$

Differentiating (4.2) with respect to time  $t$  and arc length  $s$  respectively and using (4.4) and (4.5) we get

$$\dot{\mathbf{x}}' = \boldsymbol{\Omega} \times \mathbf{x}' \quad (4.6)$$

$$\mathbf{x}'' = \mathbf{T} \times \mathbf{x}' \quad (4.7)$$

Set

$$\omega := \boldsymbol{\Omega} \cdot \mathbf{x}' \quad (4.8)$$

$$\tau := \mathbf{T} \cdot \mathbf{x}' \quad (4.9)$$

We refer to  $\omega$  as the *angular velocity* of that piece of spring and  $\tau$  as the *torsion* of

that piece of string.<sup>1</sup> By (4.6) and using its cross product with  $\mathbf{x}'$  we have,

$$\begin{aligned}\mathbf{x}' \times \dot{\mathbf{x}}' &= \mathbf{x}' \times (\boldsymbol{\Omega} \times \mathbf{x}') \\ &= \boldsymbol{\Omega}(\mathbf{x}' \cdot \mathbf{x}') - \mathbf{x}'(\mathbf{x}' \cdot \boldsymbol{\Omega})\end{aligned}$$

and thus,

$$\boldsymbol{\Omega} = \omega \mathbf{x}' + \mathbf{x}' \times \dot{\mathbf{x}}' \quad (4.10)$$

Similarly taking the cross product of (4.7) with  $\mathbf{x}'$  we can show that

$$\mathbf{T} = \tau \mathbf{x}' + \mathbf{x}' \times \mathbf{x}'' \quad (4.11)$$

Since there is no torsion in the spring at rest, we have the formula

$$R'_0 = (\mathbf{x}'_0 \times \mathbf{x}''_0) \times R_0. \quad (4.12)$$

**Definition 2.** Define the transported rest curvature  $\boldsymbol{\kappa}$  to be

$$\boldsymbol{\kappa} = RR_0^{-1}\mathbf{x}''_0. \quad (4.13)$$

**Proposition 7.** For all vectors  $\mathbf{u}$ ,  $\mathbf{v}$  and any invertible matrix  $R$  we have,

$$\mathbf{u} \times R = \mathbf{v} \times R \quad \text{if and only if} \quad \mathbf{u} = \mathbf{v} \quad (4.14)$$

*Proof.* Indeed if  $\mathbf{u} \times R = \mathbf{v} \times R$  then,  $[\mathbf{u}]R = [\mathbf{v}]R$  where  $[\mathbf{u}], [\mathbf{v}]$  are the skew symmetric matrices associated with  $\mathbf{u}$  and  $\mathbf{v}$  as defined in (4.3). Since  $R$  is invertible

---

<sup>1</sup>Note that this is not the same traditional definition of torsion given in terms of the Frenet–Serret frame

it now follows that  $[\mathbf{u}] = [\mathbf{v}]$ . Consequently,  $\mathbf{u} = \mathbf{v}$  since there is a clear one-to-one identification with the vector  $\mathbf{u}$  and its associated skew symmetric matrix  $[\mathbf{u}]$ . The converse implication is trivial.  $\square$

**Lemma 8.** *The following identities hold.*

$$\dot{\mathbf{T}} = \boldsymbol{\Omega}' + \boldsymbol{\Omega} \times \mathbf{T} \quad (4.15)$$

$$\dot{\tau} = \omega' + (\mathbf{x}'' \times \dot{\mathbf{x}}') \cdot \mathbf{x}' \quad (4.16)$$

*Proof.* Using the fact that  $\frac{\partial}{\partial t} R' = \frac{\partial}{\partial s} \dot{R}$  and using (4.4) and (4.5) we have the following set of equalities.

$$\begin{aligned} \dot{\mathbf{T}} \times R + T \times (\boldsymbol{\Omega} \times R) &= \boldsymbol{\Omega}' \times R + \boldsymbol{\Omega} \times (\mathbf{T} \times R) \\ \boldsymbol{\Omega} \times (\mathbf{T} \times R) - \mathbf{T} \times (\boldsymbol{\Omega} \times R) &= (\dot{\mathbf{T}} \times R) - (\boldsymbol{\Omega}' \times R) \\ \mathbf{T}(\boldsymbol{\Omega} \cdot R) - R(\boldsymbol{\Omega} \cdot T) - \boldsymbol{\Omega}(\mathbf{T} \cdot R) + R(\mathbf{T} \cdot \boldsymbol{\Omega}) &= (\dot{\mathbf{T}} \times R) - (\boldsymbol{\Omega}' \times R) \end{aligned} \quad (4.17)$$

$$\begin{aligned} \mathbf{T}(\boldsymbol{\Omega} \cdot R) - \boldsymbol{\Omega}(\mathbf{T} \cdot R) &= (\dot{\mathbf{T}} \times R) - (\boldsymbol{\Omega}' \times R) \\ (\boldsymbol{\Omega} \times T) \times R &= (\dot{\mathbf{T}} \times R) - (\boldsymbol{\Omega}' \times R) \end{aligned} \quad (4.18)$$

Thus,

$$\dot{\mathbf{T}} \times R = (\boldsymbol{\Omega}' \times R) + (\boldsymbol{\Omega} \times R) \times R. \quad (4.19)$$

Applying proposition (7) we get

$$\dot{\mathbf{T}} = \boldsymbol{\Omega}' + \boldsymbol{\Omega} \times \mathbf{T}. \quad (4.20)$$

On (4.17) and (4.18) we used the Lagrange's formula for vector triple products. To

show (4.16) we proceed with the following set of qualities.

$$\begin{aligned}
\dot{\tau} - \omega' &= \frac{\partial}{\partial t}(\mathbf{T} \cdot \mathbf{x}') - \frac{\partial}{\partial s}(\boldsymbol{\Omega} \cdot \mathbf{x}') \\
&= \dot{\mathbf{T}} \cdot \mathbf{x}' + \mathbf{T} \cdot \dot{\mathbf{x}}' - \boldsymbol{\Omega}' \cdot \mathbf{x}' - \boldsymbol{\Omega} \cdot \mathbf{x}'' \\
&= (\boldsymbol{\Omega}' + \boldsymbol{\Omega} \times \mathbf{T}) \cdot \mathbf{x}' + \mathbf{T} \cdot \dot{\mathbf{x}}' - \boldsymbol{\Omega}' \cdot \mathbf{x}' - \boldsymbol{\Omega} \cdot \mathbf{x}'' \\
&= (\boldsymbol{\Omega} \times \mathbf{T}) \cdot \mathbf{x}' + \mathbf{T} \cdot \dot{\mathbf{x}}' - \boldsymbol{\Omega} \cdot \mathbf{x}''
\end{aligned} \tag{4.21}$$

Note that we have  $\mathbf{x}' \cdot \mathbf{x}' = 1$ . Taking derivatives of this equation with respect to time  $t$  we get  $\mathbf{x}' \cdot \dot{\mathbf{x}}' = 0$  and taking derivatives with respect to  $s$  we get  $\mathbf{x}' \cdot \mathbf{x}'' = 0$ . Thus,

$$\begin{aligned}
\mathbf{T} \cdot \dot{\mathbf{x}}' - \boldsymbol{\Omega} \cdot \mathbf{x}'' &= (\tau \mathbf{x}' + \mathbf{x}' \times \mathbf{x}'') \cdot \dot{\mathbf{x}}' - (\omega \mathbf{x}' + \mathbf{x}' \times \dot{\mathbf{x}}') \cdot \mathbf{x}'' \\
&= (\mathbf{x}' \times \mathbf{x}'') \cdot \dot{\mathbf{x}}' - (\mathbf{x}' \times \dot{\mathbf{x}}') \cdot \mathbf{x}'' \\
&= \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') - \mathbf{x}' \cdot (\dot{\mathbf{x}}' \times \mathbf{x}'') \\
&= \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') + \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') \\
&= 2 \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}')
\end{aligned} \tag{4.22}$$

We also have

$$\begin{aligned}
(\boldsymbol{\Omega} \times \mathbf{T}) \cdot \mathbf{x}' &= (\mathbf{x}' \times \dot{\mathbf{x}}') \times (\mathbf{x}' \times \mathbf{x}'') \cdot \mathbf{x}' \\
&= (\mathbf{x}' \times \dot{\mathbf{x}}') \cdot (\mathbf{x}' \times \mathbf{x}'') \times \mathbf{x}' \\
&= (\mathbf{x}' \times \dot{\mathbf{x}}') \cdot [\mathbf{x}''(\mathbf{x}' \cdot \mathbf{x}') - \mathbf{x}'(\mathbf{x}' \cdot \mathbf{x}'')] \\
&= (\mathbf{x}' \times \dot{\mathbf{x}}') \cdot \mathbf{x}''
\end{aligned} \tag{4.23}$$

Collecting the results in (4.21), (4.22) and (4.23) we get

$$\begin{aligned}\dot{\tau} - \omega' &= (\mathbf{x}' \times \dot{\mathbf{x}}') \cdot \mathbf{x}'' + 2 \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') \\ &= -\mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') + 2 \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') \\ &= \mathbf{x}' \cdot (\mathbf{x}'' \times \dot{\mathbf{x}}') \\ &= (\mathbf{x}'' \times \dot{\mathbf{x}}') \cdot \mathbf{x}'.\end{aligned}\tag{4.24}$$

□

# Chapter 5

## Solving a constrained ODE

### 5.1 Notation

For any matrix  $\mathbf{a}$  define  $\omega(\mathbf{a})$  to be the vector

$$\omega(\mathbf{a}) = [\mathbf{a}_{23} - \mathbf{a}_{32}, \mathbf{a}_{31} - \mathbf{a}_{13}, \mathbf{a}_{12} - \mathbf{a}_{21}]^T \quad (5.1)$$

Now notice that  $\mathbf{v} \cdot \omega(\mathbf{a}) = \text{trace}(\mathbf{a}\Omega_{\mathbf{v}})$ . Given two vectors  $\mathbf{u} = [u_1, u_2, u_3]$  and  $\mathbf{v} = [v_1, v_2, v_3]$ , we denote by  $\mathbf{u} \otimes \mathbf{v}$  the matrix

$$(\mathbf{u} \otimes \mathbf{v})_{ij} = u_i v_j \quad (5.2)$$

It now follows that  $(\mathbf{u} \otimes \mathbf{v})\mathbf{w} = (\mathbf{u} \cdot \mathbf{v})\mathbf{w}$ .

### 5.2 The space discretization

We approximate the path of the spring by a cubic spline. (This is the same method that was employed in [52]). Pick some positive integer  $N$  and set the space mesh size

$\eta = L/N$ . We then define,

$$\mathbf{x}_j = \mathbf{x}(\eta j), \quad \text{for } 0 \leq j \leq N \quad (5.3)$$

and suppose that  $\mathbf{x}(s)$  is a cubic spline passing through these points. In other words  $\mathbf{x}(s)$  is piecewise cubic function for  $j\eta \leq s \leq (j+1)\eta$  for  $0 \leq j \leq N$ . Moreover the cubic spline has a continuous second derivative. Consider the following types of boundary conditions.

$$\mathbf{x}(0) \text{ and } \mathbf{x}'(0) \text{ are known functions of } t; \text{ or} \quad (5.4)$$

$$\mathbf{x}(0) \text{ is a known function } t \text{ and} \quad (5.5)$$

$$\mathbf{x}'(0) \text{ is not a known function of } t; \text{ or}$$

$$\text{neither } \mathbf{x}(0) \text{ nor } \mathbf{x}'(0) \text{ are known functions of } t \quad (5.6)$$

and

$$\mathbf{x}(L) \text{ and } \mathbf{x}'(L) \text{ are known functions of } t; \text{ or} \quad (5.7)$$

$$\mathbf{x}(L) \text{ is a known function } t \text{ and} \quad (5.8)$$

$$\mathbf{x}'(L) \text{ is not a known function of } t; \text{ or}$$

$$\text{neither } \mathbf{x}(L) \text{ nor } \mathbf{x}'(L) \text{ are known functions of } t. \quad (5.9)$$

Denote the first and second derivatives of the spline

$$\mathbf{x}'_j = \mathbf{x}'(\eta j), \quad \mathbf{x}''_j = \mathbf{x}''(\eta j), \quad \text{for } 0 \leq j \leq N. \quad (5.10)$$

It can be shown that these derivatives can be computed using the following tridiagonal matrices. See, for example [10].

$$\mathbf{A} = \begin{bmatrix} A_{0,0} & A_{0,1} & 0 & 0 & \cdots & 0 & 0 \\ \eta & 4\eta & \eta & 0 & \cdots & 0 & 0 \\ 0 & \eta & 4\eta & \eta & \cdots & 0 & 0 \\ 0 & 0 & \eta & 4\eta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4\eta & \eta \\ 0 & 0 & 0 & 0 & \cdots & A_{N,N-1} & A_{N,N} \end{bmatrix} \quad (5.11)$$

where,

$$A_{0,0} = 2\eta, A_{0,1} = \eta \quad \text{if boundary conditions in (5.4) holds} \quad (5.12)$$

$$A_{0,0} = 1, A_{0,1} = 0 \quad \text{if boundary conditions in (5.5) or (5.6) holds} \quad (5.13)$$

$$A_{N,N} = \eta, A_{N,N-1} = 2\eta \quad \text{if boundary conditions in (5.7) holds} \quad (5.14)$$

$$A_{N,N-1} = 0, A_{N,N} = 1 \quad \text{if boundary conditions in (5.8) or (5.9) holds.} \quad (5.15)$$



$$\mathbf{U} = \frac{1}{\eta} \begin{bmatrix} U_{0,0} & U_{0,1} & 0 & 0 & \cdots & 0 & 0 \\ 3 & -6 & 3 & 0 & \cdots & 0 & 0 \\ 0 & 3 & -6 & 3 & \cdots & 0 & 0 \\ 0 & 0 & 3 & -6 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -6 & 3 \\ 0 & 0 & 0 & 0 & \cdots & U_{N,N-1} & U_{N,N} \end{bmatrix} \quad (5.16)$$

$$U_{0,0} = -3, U_{0,1} = 3 \quad \text{if boundary conditions in (5.4) holds} \quad (5.17)$$

$$U_{0,0} = 0, U_{0,1} = 0 \quad \text{if boundary conditions in (5.5) or (5.6) holds} \quad (5.18)$$

$$U_{N,N} = 3, U_{N,N} = -3 \quad \text{if boundary conditions in (5.7) holds} \quad (5.19)$$

$$U_{N,N-1} = 0, U_{N,N} = 0 \quad \text{if boundary conditions in (5.8) or (5.9) holds.} \quad (5.20)$$

$$\mathbf{B} = \frac{1}{\eta} \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \quad (5.21)$$

$$C = \frac{\eta}{3} \begin{bmatrix} 2 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 2 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & -1 & -2 \end{bmatrix} \quad (5.22)$$

From this point forward we use  $\mathbf{x}$ ,  $\mathbf{x}'$  and  $\mathbf{x}''$  to denote either the functions  $[0, L]$ , or the vectors

$$\mathbf{x} = [\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_N]^\top, \quad (5.23)$$

$$\mathbf{x}' = [\mathbf{x}'_0, \mathbf{x}'_1, \cdots, \mathbf{x}'_N]^\top, \quad (5.24)$$

$$\mathbf{x}'' = [\mathbf{x}''_0, \mathbf{x}''_1, \cdots, \mathbf{x}''_N]^\top. \quad (5.25)$$

In particular, we can compute the derivatives  $\mathbf{x}'$  and  $\mathbf{x}''$  using

$$\mathbf{x}'' = D_2 \mathbf{x} \quad (5.26)$$

$$\mathbf{x}' = D_1 \mathbf{x} \quad (5.27)$$

where,

$$D_2 = 2A^{-1}U, \quad (5.28)$$

$$D_1 = B - CA^{-1}U. \quad (5.29)$$

If  $\mathbf{x} \in \mathbb{R}^{n+1}$  then denote

$$\mathbf{x}' = D_1 \mathbf{x}, \quad \mathbf{x}'' = D_2 \mathbf{x}, \quad \mathbf{x}^\dagger = -D_1^\dagger \mathbf{x}, \quad \mathbf{x}^{\dagger\dagger} = D_2^\dagger \mathbf{x} \quad (5.30)$$

with boundary conditions second derivatives equal to zero.

**Definition 3.** Let  $\mathbf{s} = (\mathbf{s}_j)_{0 \leq j \leq N}$  be a sequence of  $3 \times 2$  matrices satisfying  $\mathbf{s}_j^T \mathbf{s}_j = \mathbf{i}_2$ . Let  $\eta$  be the unit step in space discretization. Now define the sequence of scalars  $\tau = (\tau_j)_{1 \leq j \leq N}$  by

$$\tau_j \mathbf{j} = \frac{(\mathbf{s}_{j-1} + \mathbf{s}_j)^T (\mathbf{s}_j - \mathbf{s}_{j-1})}{2\eta} = \frac{\mathbf{s}_{j-1}^T \mathbf{s}_j - \mathbf{s}_j^T \mathbf{s}_{j-1}}{2\eta} \quad (5.31)$$

where

$$\mathbf{j} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (5.32)$$

and  $\mathbf{i}_2$  denoting the  $2 \times 2$  identity matrix.

In functional terms (5.31) it may be written as

$$\tau \mathbf{j} = \mathbf{s}^T \mathbf{s}'. \quad (5.33)$$

**Definition 4.** Let

$$\mathbf{f}_j(\tau, \mathbf{s}) = -\frac{1}{\eta} (\tau_{j+1} \mathbf{s}_{j+1} - \tau_j \mathbf{s}_{j-1}) \quad (5.34)$$

for  $1 \leq j \leq N - 1$ . Let

$$\mathbf{f}_0(\tau, \mathbf{s}) = -\frac{1}{\eta} \tau_1 \mathbf{s}_1 \quad (5.35)$$

and let

$$\mathbf{f}_N(\tau, \mathbf{s}) = \frac{1}{\eta} \tau_N \mathbf{s}_{N-1} \quad (5.36)$$

In functional terms this would be.

$$\mathbf{f}(\tau, \mathbf{s}) = -\tau \mathbf{s}' - (\tau \mathbf{s})'. \quad (5.37)$$

In light of these definitions we have the following.

**Theorem 9.** *For any function/sequence  $\tau_1$  we have,*

$$\int_0^L 2\tau_1 \delta \tau ds = \int_0^L \text{trace} (\mathbf{f}(\tau_1, \mathbf{s}) \mathbf{j} \delta \mathbf{s}^\top) ds \quad (5.38)$$

and consequently

$$\delta \int_0^L \tau^2 ds = \int_0^L \text{trace} (\mathbf{f}(\tau, \mathbf{s}) \mathbf{j} \delta \mathbf{s}^\top) ds. \quad (5.39)$$

*Proof.* Let  $I = \int_0^L \text{trace} (\mathbf{f}(\tau_1, \mathbf{s}) \mathbf{j} \delta \mathbf{s}^\top) ds$ . Then,

$$\begin{aligned} I &= \int_0^L \text{trace} ((-\tau_1 \mathbf{s}' - (\tau_1 \mathbf{s})') \mathbf{j} (\delta \mathbf{s})^\top) ds \\ &= \int_0^L \text{trace} (-\tau_1 \mathbf{s}' \mathbf{j} (\delta \mathbf{s})^\top + \tau_1 \mathbf{s} \mathbf{j} ((\delta \mathbf{s})^\top)') ds \\ &= \int_0^L \text{trace} (-\tau_1 \mathbf{s}' \mathbf{j} (\delta \mathbf{s})^\top + \tau_1 (\delta \mathbf{s})' \mathbf{j}^\top \mathbf{s}^\top) ds \\ &= \int_0^L \text{trace} (-\tau_1 \mathbf{s}' \mathbf{j} (\delta \mathbf{s})^\top - \tau_1 (\delta \mathbf{s})' \mathbf{j} \mathbf{s}^\top) ds \\ &= \int_0^L \text{trace} (-\tau_1 \delta (\mathbf{s}' \mathbf{j} \mathbf{s}^\top)) ds \\ &= \int_0^L \text{trace} (-\tau_1 \delta (\mathbf{s}^\top \mathbf{s}' \mathbf{j})) ds \\ &= \int_0^L \text{trace} (-\tau_1 \delta (-\tau_1)) ds \\ &= \int_0^L 2\tau_1 \delta (\tau) ds \end{aligned}$$

The equality in (5.39) is now immediate since  $\delta(\tau^2) = 2\tau\delta(\tau)$ . □

We can now prove the discrete version of the above theorem.

**Theorem 10.** *Let  $\tau_1 = (\tau_j)_{0 \leq j \leq N+1}$  be a sequence of values such that  $\tau_0 = \tau_{N+1} = 0$  and  $(\tau_{1,j})_{1 \leq j \leq N}$  is a sequence that satisfies (5.31). Consider the space discretization*

given by

$$\tau = \tau_j \quad (5.40)$$

$$\mathbf{s}^\top = \frac{(\mathbf{s}_{j-1} + \mathbf{s}_j)^\top}{2} \quad (5.41)$$

$$\mathbf{s}' = \frac{\mathbf{s}_j - \mathbf{s}_{j-1}}{\eta} \quad (5.42)$$

$$(\tau \mathbf{s})' = \frac{\tau_{j+1} \mathbf{s}_{j+1} - \tau_j \mathbf{s}_j}{\eta} \quad (5.43)$$

then, definition for  $\mathbf{f}_j$  as given in (5.34) may be rewritten in the equivalent form as,

$$\mathbf{f}_j(\tau, \mathbf{s}) = - \left[ \tau_j \left( \frac{\mathbf{s}_j - \mathbf{s}_{j-1}}{\eta} \right) + \left( \frac{\tau_{j+1} \mathbf{s}_{j+1} - \tau_j \mathbf{s}_j}{\eta} \right) \right] \quad (5.44)$$

and consequently,

$$\sum_{j=0}^N 2\tau_j \delta(\tau_j) = \sum_{j=0}^N \text{trace} [\mathbf{f}_j(\tau, \mathbf{s}) \mathbf{j} (\delta \mathbf{s}_j)^\top] \quad (5.45)$$

*Proof.* Let us denote the right hand side of (5.45) as RHS. Then,

$$\text{RHS} = \frac{1}{\eta} \sum_{j=0}^N \text{trace} \left[ -\tau_j (\mathbf{s}_j - \mathbf{s}_{j-1}) \mathbf{j} (\delta \mathbf{s}_j)^\top - (\tau_{j+1} \mathbf{s}_{j+1} - \tau_j \mathbf{s}_j) \mathbf{j} (\delta \mathbf{s}_j)^\top \right] \quad (5.46)$$

$$= \frac{1}{\eta} \sum_{j=0}^N \text{trace} \left[ -\tau_j (\mathbf{s}_j - \mathbf{s}_{j-1}) \mathbf{j} (\delta \mathbf{s}_j)^\top + \tau_j \mathbf{s}_j \mathbf{j} (\delta \mathbf{s}_j - \delta \mathbf{s}_{j-1})^\top \right] \quad (5.47)$$

$$= \frac{1}{\eta} \sum_{j=0}^N \text{trace} \left[ -\tau_j (\mathbf{s}_j - \mathbf{s}_{j-1}) \mathbf{j} (\delta \mathbf{s}_j)^\top - \tau_j (\delta \mathbf{s}_j - \delta \mathbf{s}_{j-1}) \mathbf{j} \mathbf{s}_j^\top \right] \quad (5.48)$$

$$= \frac{1}{\eta} \sum_{j=0}^N \text{trace} \left[ -\tau_j \delta \left( (\mathbf{s}_j - \mathbf{s}_{j-1}) \mathbf{j} \mathbf{s}_j^\top \right) \right] \quad (5.49)$$

We will now discretize  $\mathbf{s}_j^\top$  as  $(\mathbf{s}_j + \mathbf{s}_{j-1})^\top / 2$ . Then,

$$\text{RHS} = \sum_{j=0}^N \text{trace} \left[ -\tau_j \delta \left( \frac{(\mathbf{s}_j - \mathbf{s}_{j-1})_j (\mathbf{s}_j + \mathbf{s}_{j-1})^\top}{\eta} \right) \right] \quad (5.50)$$

$$= \sum_{j=0}^N \text{trace} \left[ -\tau_j \delta \left( \frac{(\mathbf{s}_j + \mathbf{s}_{j-1})^\top (\mathbf{s}_j - \mathbf{s}_{j-1})_j}{\eta} \right) \right] \quad (5.51)$$

$$= \sum_{j=0}^N \text{trace} [-\tau_j \delta(\tau_j j^2)] \quad (5.52)$$

$$= \sum_{j=0}^N \text{trace} [-\tau_j \delta(-\tau_j \mathbf{i}_2)] \quad (5.53)$$

$$= \sum_{j=0}^N 2\tau_j \delta \tau_j \quad (5.54)$$

Note that on (5.47) we are using summation by parts. On (5.49) we are using the fact that  $\delta$  satisfies the Leibniz rule. Indeed this will be true if we used the following time discretization:

$$a = \frac{a_{n+1} + a_{n-1}}{2} \quad (5.55)$$

$$\delta(a) = \frac{a_{n+1} - a_{n-1}}{2h} \quad (5.56)$$

$$\delta(ab) = \frac{a_{n+1}b_{n+1} - a_{n-1}b_{n-1}}{2h} \quad (5.57)$$

where  $h$  stands for the unit time step and  $n$  stands for the  $n$ th time. Under this discretization it is also immediate that  $\delta(\tau^2) = 2\tau\delta\tau$  also holds. Consequently the discrete version of (5.39), namely

$$\delta \left( \sum_{j=0}^N \tau_j^2 \right) = \sum_{j=0}^N \text{trace} [\mathbf{f}_j(\tau, \mathbf{s})_j \delta(\mathbf{s}_j)^\top] \quad (5.58)$$

also holds true. □

**Proposition 11.** For any sequence/function of matrices  $\mathbf{t}$

$$\int_0^L \text{trace}(\mathbf{f}(\tau, \mathbf{s})\mathbf{j}\mathbf{t}^\top) dt = \int_0^L \text{trace}(\mathbf{f}(\tau, \mathbf{t})\mathbf{j}\mathbf{s}^\top) dt \quad (5.59)$$

*Proof.* Let  $J = \int_0^L \text{trace}(\mathbf{f}(\tau, \mathbf{s})\mathbf{j}\mathbf{t}^\top) ds$ . Then,

$$\begin{aligned} J &= \int_0^L \text{trace}((-\tau\mathbf{s}' - (\tau\mathbf{s})')\mathbf{j}\mathbf{t}^\top) ds \\ &= \int_0^L \text{trace}(-\tau\mathbf{s}'\mathbf{j}\mathbf{t}^\top - (\tau\mathbf{s})'\mathbf{j}\mathbf{t}^\top) ds \\ &= \int_0^L \text{trace}(-\tau\mathbf{s}'\mathbf{j}\mathbf{t}^\top + (\tau\mathbf{s})\mathbf{j}(\mathbf{t}')^\top) ds \\ &= \int_0^L \text{trace}(-\mathbf{s}'\mathbf{j}(\tau\mathbf{t})^\top + \tau\mathbf{t}'\mathbf{j}^\top\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(\mathbf{s}\mathbf{j}(\tau\mathbf{t}^\top)' - \tau\mathbf{t}'\mathbf{j}\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(\mathbf{s}\mathbf{j}(\tau'\mathbf{t}^\top + \tau(\mathbf{t}')^\top) - \tau\mathbf{t}'\mathbf{j}\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(\tau(\mathbf{s}\mathbf{j}(\mathbf{t}')^\top) + \tau'\mathbf{s}\mathbf{j}\mathbf{t}^\top - \tau\mathbf{t}'\mathbf{j}\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(\tau\mathbf{t}'\mathbf{j}^\top\mathbf{s}^\top + \tau'\mathbf{t}\mathbf{j}^\top\mathbf{s}^\top - \tau\mathbf{t}'\mathbf{j}\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(-\tau\mathbf{t}'\mathbf{j}\mathbf{s}^\top - \tau'\mathbf{t}\mathbf{j}\mathbf{s}^\top - \tau\mathbf{t}'\mathbf{j}\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(-\tau\mathbf{t}' - (\tau'\mathbf{t} + \tau\mathbf{t}')\mathbf{j}\mathbf{s}^\top) ds \\ &= \int_0^L \text{trace}(-\tau\mathbf{t}' - (\tau\mathbf{t})')\mathbf{j}\mathbf{s}^\top ds \\ &= \int_0^L \text{trace}(\mathbf{f}(\tau, \mathbf{t})\mathbf{j}\mathbf{s}^\top) ds \end{aligned}$$

□

### 5.3 The transported rest curvature

Suppose the spring at rest is described by the curve  $\mathbf{x}_0(s)$  for  $0 \leq s \leq L$ . We calculate  $\mathbf{r}_0(s)$  of orthogonal matrices by solving the equation

$$\mathbf{r}_0(0)\mathbf{i} = \mathbf{x}'(0) \quad (5.60)$$

$$\mathbf{r}'_0 = \Omega_{\mathbf{x}' \times \mathbf{x}''} \mathbf{r}_0 \quad (5.61)$$

Notice that (5.60) specifies that the first column of  $\mathbf{r}_0$  must be  $\mathbf{x}'_0(0)$ . Notice also that  $\mathbf{r}_0(0)\mathbf{j}$  and  $\mathbf{r}_0(0)\mathbf{k}$  are not specified. The only other constraint being that  $\mathbf{r}_0(0)$  is orthogonal. Further notice that

$$\mathbf{r}_0\mathbf{i} = \mathbf{x}'_0 \quad (5.62)$$

The *transported rest curvature* is the vector defined by

$$\boldsymbol{\kappa}_0 = \mathbf{r}_0^{-1} \mathbf{x}''_0 \quad (5.63)$$

Notice that  $\boldsymbol{\kappa}_0 \in \mathbb{R}^3$  and that the first component of  $\boldsymbol{\kappa}_0$  is always zero since  $\mathbf{x}'' \cdot \mathbf{x}' = 0$  and  $\mathbf{r}_0^{-1} \mathbf{x}'_0 = \mathbf{i}$ . Therefore we may consider  $\boldsymbol{\kappa}_0 \in \mathbb{R}^2$  to be a two-dimensional vector lying in the  $yz$ -plane.

**Remark.** Notice that the sub-index 0 in  $\boldsymbol{\kappa}_0$  denotes that it is the transported natural curvature at time  $t = 0$ . In the following section we abuse notation and use the sub-index  $\boldsymbol{\kappa}_j$  to denote space discretization. We will make explicit mention of what is meant by the sub-index in the appropriate context.



## 5.4 Energies and Constraints

We are solving for the curves  $\mathbf{x}(s)$  of 3-dimensional and  $\mathbf{r}$  of orthogonal matrices. Since  $\mathbf{r}\mathbf{i} = \mathbf{x}'$ , we write  $\mathbf{s}$  as the  $3 \times 2$  matrix consisting of the second and third columns of  $\mathbf{r}$ . The first column of  $\mathbf{r}$  is determined by  $\mathbf{x}'$ . The kinetic energy is

$$\text{K.E.} = \frac{1}{2} \int_0^L \rho |\dot{\mathbf{x}}|^2 ds \quad (5.64)$$

The space discretization is

$$\text{K.E.} = \frac{1}{2} \left( \frac{1}{2} \rho |\dot{\mathbf{x}}_0|^2 + \sum_{j=1}^{N-1} \rho |\dot{\mathbf{x}}_j|^2 + \frac{1}{2} \rho |\dot{\mathbf{x}}_N|^2 \right) \eta \quad (5.65)$$

and the potential energy is

$$\text{P.E.} = \frac{1}{2} \int_0^L \left( \sigma_1 |\mathbf{s}^T \mathbf{x}'' - \boldsymbol{\kappa}|^2 + 2\sigma_2 \tau^2 \right) ds. \quad (5.66)$$

The discretization of the potential energy is

$$\text{P.E.} = \frac{\eta}{2} \left( \frac{1}{2} \sigma_1 |\mathbf{s}_0^T \mathbf{x}_0'' - \boldsymbol{\kappa}_0|^2 + \sum_{j=1}^{N-1} \sigma_1 |\mathbf{s}_j^T \mathbf{x}_j'' - \boldsymbol{\kappa}_j|^2 + \frac{1}{2} \sigma_1 |\mathbf{s}_N^T \mathbf{x}_N'' - \boldsymbol{\kappa}_N|^2 + \sum_{j=1}^N 2\sigma_2 \tau_j^2 \right). \quad (5.67)$$

In here  $\boldsymbol{\kappa}_j$ 's denote the space discretizations.

We propose the following boundary conditions and we will eventually show that these suffice to guarantee equations of motion with a solution.

**Boundary Conditions 1.** *The proposed boundary conditions are*

$$\mathbf{x}(0) \text{ and } \mathbf{x}(L) \text{ are given} \quad (5.68)$$

$$\mathbf{x}''(0) = \mathbf{s}(0)\boldsymbol{\kappa}(0) \quad (5.69)$$

$$\mathbf{x}''(L) = \mathbf{s}(L)\boldsymbol{\kappa}(L) \quad (5.70)$$

$$\boldsymbol{\tau} = \mathbf{s}^T \mathbf{s}' = 0 \text{ at } s = 0, L. \quad (5.71)$$

The fact that  $\mathbf{r}$  is orthogonal is captured by the constraints

$$\mathbf{s}^T \mathbf{s} = \mathbf{i}_2, \quad \mathbf{s}^T \mathbf{x}' = 0, \quad |\mathbf{x}'|^2 = 1. \quad (5.72)$$

Since  $\mathbf{s}^T \mathbf{s}$  is symmetric by construction, this provides six constraints per point.

The Lagrange multipliers for these constraints are a symmetric  $2 \times 2$  matrix valued  $\mathbf{a}(s)$ , a 2-vector valued function  $\boldsymbol{\beta}(s)$  for  $0 \leq s \leq L$ , a scalar valued function  $\alpha(s)$ , and the additional term to the Lagrangian is

$$\text{Constraint} = \frac{1}{2} \int_0^L \left( \text{trace}(\mathbf{a} \mathbf{s}^T \mathbf{s}) + 2\boldsymbol{\beta} \cdot \mathbf{s}^T \mathbf{x}' + \alpha |\mathbf{x}'|^2 \right) ds \quad (5.73)$$

in discretized terms,

$$\text{Constraint} = \frac{1}{2} \left( \sum_{j=0}^N \text{trace}(\mathbf{a}_j \mathbf{s}_j^T \mathbf{s}_j) + 2\boldsymbol{\beta}_j \cdot \mathbf{s}_j^T \mathbf{x}'_j + \alpha_j |\mathbf{x}'_j|^2 \right) \eta \quad (5.74)$$

## 5.5 The differential equation.

We take the variation of the usual Lagrangian  $\mathbf{L} = \text{K.E} - \text{P.E} + \text{Constraint}$  and then apply Hamilton's principle. Taking variation with respect to  $\mathbf{x}$  will yield one system of equations and taking variation with respect to  $\mathbf{s}$  will yield another set of equations. More precisely we have the following theorem.

**Theorem 12** (Equations of motion). *Let the Lagrangian  $\mathbf{L}$  be defined as*

$$\mathbf{L} = \text{K.E} - \text{P.E} + \text{Constraint} \quad (5.75)$$

*Taking variation of  $\mathbf{L}$  with respect to  $\mathbf{x}$  we get*

$$\rho\ddot{\mathbf{x}} = -\sigma_1(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})^{\dagger\dagger} - (\alpha\mathbf{x}')^{\dagger} - (s\boldsymbol{\beta})^{\dagger} \quad (5.76)$$

*and taking variation of  $\mathbf{L}$  with respect to  $\mathbf{s}$  gives*

$$-\sigma_1\mathbf{x}'' \otimes \boldsymbol{\kappa} + \sigma_2\mathbf{f}(\tau, \mathbf{s})\mathbf{j} - \mathbf{x}' \otimes \boldsymbol{\beta} - \mathbf{s}\mathbf{a} = 0. \quad (5.77)$$

*By adding damping to (5.76) we get,*

$$\rho\ddot{\mathbf{x}} = -\sigma_1(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})^{\dagger\dagger} - (\alpha\mathbf{x}')^{\dagger} - (s\boldsymbol{\beta})^{\dagger} - \nu_1(\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa})^{\dagger\dagger} \quad (5.78)$$

*Similarly adding a damping term to the right hand side of (5.77) we get,*

$$-\sigma_1\mathbf{x}'' \otimes \boldsymbol{\kappa} + \sigma_2\mathbf{f}(\tau, \mathbf{s})\mathbf{j} - \mathbf{x}' \otimes \boldsymbol{\beta} - \mathbf{s}\mathbf{a} = (\nu_1(\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa}) \otimes \boldsymbol{\kappa}) - \frac{1}{2}\nu_2\mathbf{f}(\dot{\tau}, \mathbf{s})\mathbf{j} \quad (5.79)$$

*Proof.* Let  $\delta\mathbf{x}(s, t)$  be a small perturbation that is zero at the time end points. i.e.,

$$\delta\mathbf{x}(t = 0) = \delta\mathbf{x}(t = T) \stackrel{\text{def}}{=} 0 \quad (5.80)$$

The variation of the action  $\mathbf{L}$  with respect to  $\mathbf{x}$  is

$$\delta_{\mathbf{x}}\mathbf{L} = \int_{t=0}^T \int_{s=0}^L \rho \delta \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} \, ds dt - \int_{t=0}^L \int_{s=0}^L \sigma_1 \delta \mathbf{x}'' \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \, ds dt \quad (5.81)$$

$$+ \int_{t=0}^T \int_{s=0}^L \delta \mathbf{x}' \cdot \frac{\partial}{\partial \mathbf{x}'} (\boldsymbol{\beta}^\top \mathbf{s}^\top \mathbf{x}') \, ds dt \quad (5.82)$$

$$+ \int_{t=0}^T \int_{s=0}^L \delta \mathbf{x}' \cdot (\boldsymbol{\alpha} \mathbf{x}') \, ds dt \quad (5.83)$$

$$=: E_1 - E_2 + E_3 + E_4 \quad (5.84)$$

Let us take a look at each of these terms one at a time. Note here that the variation of the terms  $\int_0^L \text{trace}(\mathbf{a}\mathbf{s}^\top) \, ds$  and  $\int_0^L 2\sigma_2 \tau^2 \, ds$  with respect to  $\mathbf{x}$  are zero because they do not depend on  $\mathbf{x}$ .

Interchanging the variable of integration and then using integration by parts (with respect to time  $t$ ) we have,

$$\begin{aligned} E_1 &= \int_{t=0}^T \int_{s=0}^L \rho \delta \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} \, ds dt \\ &= \int_{s=0}^L \int_{t=0}^T \rho \delta \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} \, dt ds \\ &= \int_{s=0}^L \int_{t=0}^T -\rho \delta \mathbf{x} \cdot \ddot{\mathbf{x}} \, dt ds + \int_{s=0}^L \rho [\delta \mathbf{x} \cdot \dot{\mathbf{x}}]_{t=0}^{t=T} \, ds \\ &= \int_{t=0}^T \int_{s=0}^L -\rho \delta \mathbf{x} \cdot \ddot{\mathbf{x}} \, ds dt \end{aligned} \quad (5.85)$$

The boundary term on (5.85) disappear because of (5.80).

For  $E_2$  we integrate by parts with respect to  $s$  twice to get

$$\begin{aligned} E_2 &= \int_{t=0}^T \int_{s=0}^L \sigma_1 (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})^{\dagger\dagger} \cdot \delta \mathbf{x} \, ds dt \\ &+ \int_{t=0}^T [(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \delta \mathbf{x}']_{s=0}^{s=L} \, dt \end{aligned} \quad (5.86)$$

$$- \int_{t=0}^T [(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})^\dagger \delta \mathbf{x}]_{s=0}^{s=L} \, dt \quad (5.87)$$

The boundary terms on (5.86) disappear because in boundary condition (5.69) and (5.70) we have chosen the boundary conditions to be  $\mathbf{x}''(0) = \mathbf{s}(0)\boldsymbol{\kappa}(0)$  and  $\mathbf{x}''(L) = \mathbf{s}(L)\boldsymbol{\kappa}(L)$ . Again from *boundary conditions* as stated above we have that  $\mathbf{x}(0)$  and  $\mathbf{x}(L)$  are given. Consequently there is no variation at the end points. Thus,  $\delta\mathbf{x}(s=0) = \delta\mathbf{x}(s=L) = 0$  and so the terms on (5.87) vanish.

Again for  $E_3$  we integrate by parts with respect to  $s$  to get

$$E_3 = \int_{t=0}^T \int_{s=0}^L -\delta\mathbf{x} \cdot (\mathbf{s}\boldsymbol{\beta})^\dagger ds dt + \int_{t=0}^T [\delta\mathbf{x} \cdot (\mathbf{s}\boldsymbol{\beta})]_{s=0}^L dt \quad (5.88)$$

Once again our boundary condition stipulation that there is no variation at the end points makes the boundary terms on (5.88) disappear. As for the last term  $E_4$  yet again, the boundary conditions that there is no variation at the end points gives

$$E_4 = \int_{t=0}^T \int_{s=0}^L -\delta\mathbf{x} \cdot (\boldsymbol{\alpha}\mathbf{x}')^\dagger ds dt \quad (5.89)$$

Consequently,

$$\delta_{\mathbf{x}}\mathbf{L} = \int_{t=0}^T \int_{s=0}^L -\rho \delta\mathbf{x} \cdot \ddot{\mathbf{x}} - \sigma_1 \delta\mathbf{x} \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})^{\dagger\dagger} - \delta\mathbf{x} \cdot (\boldsymbol{\alpha}\mathbf{x}')^\dagger - \delta\mathbf{x} \cdot (\mathbf{s}\boldsymbol{\beta})^\dagger ds dt \quad (5.90)$$

Hamilton's principle now gives (5.76) as desired.

Let us now compute the variation of  $\mathbf{L}$  with respect to  $\mathbf{s}$ . We get,

$$\begin{aligned}
\delta_{\mathbf{s}}\mathbf{L} &= -\frac{1}{2} \int_{t=0}^T \int_{s=0}^L \sigma_1 \frac{\partial}{\partial \mathbf{s}} [(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})] \cdot \delta \mathbf{s} \, ds dt \\
&\quad - \delta_{\mathbf{s}} \left( \int_{t=0}^T \int_{s=0}^L \sigma_2 \tau^2 \, ds dt \right) \\
&\quad + \frac{1}{2} \int_{t=0}^T \int_{s=0}^L \frac{\partial}{\partial \mathbf{s}} \text{trace}(\mathbf{a}\mathbf{s}^\top \mathbf{s}) \cdot \delta \mathbf{s} \, ds dt \\
&\quad + \int_{t=0}^T \int_{s=0}^L \frac{\partial}{\partial \mathbf{s}} (\boldsymbol{\beta} \cdot \mathbf{s}^\top \mathbf{x}') \cdot \delta \mathbf{s} \, ds dt \\
&=: G_1 + G_2 + G_3 + G_4
\end{aligned}$$

We have,

$$\begin{aligned}
(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) &= \mathbf{x}''^\top \mathbf{x}'' - 2\mathbf{x}''^\top \mathbf{s}\boldsymbol{\kappa} + \boldsymbol{\kappa}^\top (\mathbf{s}^\top \mathbf{s}) \boldsymbol{\kappa} \\
&= \mathbf{x}''^\top \mathbf{x}'' - 2\mathbf{x}''^\top \mathbf{s}\boldsymbol{\kappa} + \boldsymbol{\kappa}^\top \boldsymbol{\kappa}
\end{aligned}$$

It now follows that

$$\frac{\partial}{\partial \mathbf{s}} (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) = -2(\mathbf{x}'' \otimes \boldsymbol{\kappa})$$

Thus,

$$G_1 = \int_{t=0}^T \int_{s=0}^L \sigma_1 (\mathbf{x}'' \otimes \boldsymbol{\kappa}) \cdot \delta \mathbf{s} \, ds dt$$

For  $G_2$  we proceed as follows. Note that by (5.33) we have  $\tau \mathbf{j}^2 = \mathbf{s}^\top \mathbf{s}' \mathbf{j}$ . But  $\mathbf{j}^2 = -\mathbf{i}_2$

and so  $\text{trace}(-\tau \mathbf{i}_2) = \text{trace}(\mathbf{s}^\top \mathbf{s}' \mathbf{j})$ . This shows that  $\tau = -\frac{1}{2} \text{trace}(\mathbf{s}^\top \mathbf{s}' \mathbf{j})$ . Consequently,

$$\begin{aligned}
-G_2 &= \int_{t=0}^T \int_{s=0}^L \sigma_1 \delta_s(\tau^2) ds dt \\
&= \int_{t=0}^T \int_{s=0}^L \frac{\partial}{\partial \mathbf{s}}(\tau^2) \cdot \delta \mathbf{s} + \frac{\partial}{\partial \mathbf{s}'}(\tau^2) \cdot \delta \mathbf{s}' ds dt \\
&= - \int_{t=0}^T \int_{s=0}^L \tau \frac{\partial}{\partial \mathbf{s}}(\text{trace}(\mathbf{s}^\top \mathbf{s}' \mathbf{j})) \cdot \delta \mathbf{s} + \tau \frac{\partial}{\partial \mathbf{s}'}(\text{trace}(\mathbf{s}^\top \mathbf{s}' \mathbf{j})) \cdot \delta \mathbf{s}' ds dt \\
&= - \int_{t=0}^T \int_{s=0}^L \tau \mathbf{s}' \mathbf{j} \cdot \delta \mathbf{s} + \tau \mathbf{s} \mathbf{j}^\top \cdot \delta \mathbf{s}' ds dt \\
&= \int_{t=0}^T \int_{s=0}^L -\tau \mathbf{s}' \mathbf{j} \cdot \delta \mathbf{s} - (\tau \mathbf{s})' \mathbf{j} \cdot \delta \mathbf{s} ds dt + \int_{t=0}^T [\tau \mathbf{s} \mathbf{j} \cdot \delta \mathbf{s}]_{s=0}^{s=L} ds dt \tag{5.91}
\end{aligned}$$

$$\begin{aligned}
&= \int_{t=0}^T \int_{s=0}^L (-\tau \mathbf{s}' - (\tau \mathbf{s})') \mathbf{j} \cdot \delta \mathbf{s} ds dt \tag{5.92} \\
&= \int_{t=0}^T \int_{s=0}^L f(\tau, \mathbf{s}) \mathbf{j} \cdot \delta \mathbf{s} ds dt
\end{aligned}$$

On (5.91) we used the fact that  $\mathbf{j}^\top = -\mathbf{j}$  and integration by parts. On (5.92) we used the boundary condition on (5.71).

**Remark.** *The boundary condition in (5.71) states that  $\mathbf{s}^\top \mathbf{s}' = 0$  for  $s = 0$  and  $s = L$ . But then by definition  $\tau \mathbf{j} = \mathbf{s}^\top \mathbf{s}'$  in which case we have that  $\tau = 0$  for  $s = 0$  and  $s = L$ .*

To compute  $G_3$  and  $G_4$  note that  $\frac{\partial}{\partial \mathbf{s}}(\text{trace}(\mathbf{a} \mathbf{s}^\top \mathbf{s})) = \mathbf{s} \mathbf{a}$  and  $\frac{\partial}{\partial \mathbf{s}}(\boldsymbol{\beta} \cdot \mathbf{s}^\top \mathbf{x}') = \mathbf{x}' \otimes \boldsymbol{\beta}$ .

Thus,

$$G_3 = \int_{t=0}^T \int_{s=0}^L (\mathbf{s} \mathbf{a}) \delta \mathbf{s} ds dt \tag{5.93}$$

and

$$G_4 = \int_{t=0}^T \int_{s=0}^L (\mathbf{x}' \otimes \boldsymbol{\beta}) \delta \mathbf{s} ds dt \tag{5.94}$$

Putting all of this together we get

$$\delta_s \mathbf{L} = \int_{t=0}^T \int_{s=0}^L (-\sigma_1 \mathbf{x}'' \otimes \boldsymbol{\kappa} + \sigma_2 \mathbf{f}(\tau, \mathbf{s}) \mathbf{j} - \mathbf{x}' \otimes \boldsymbol{\beta} - \mathbf{s} \mathbf{a}) \delta \mathbf{s} \, ds dt \quad (5.95)$$

Now Hamilton's principle gives (5.77) as desired.

To obtain the damping terms we use *Rayleigh Dissipation Function* as described in [58, p. 23]. For our purposes the Rayleigh Dissipation Function  $R$  may be set up as,

$$R = \frac{1}{2} \nu_1 \int_{s=0}^L \left| \frac{\partial}{\partial t} (\mathbf{x}'' - \mathbf{s} \boldsymbol{\kappa}) \right|^2 ds + \frac{1}{2} \nu_2 \int_{s=0}^L \dot{\tau}^2 ds. \quad (5.96)$$

Thus,

$$\begin{aligned} \frac{\partial R}{\partial \dot{\mathbf{x}}} &= \frac{1}{2} \nu_1 \int_{s=0}^L \frac{\partial}{\partial \dot{\mathbf{x}}} [(\dot{\mathbf{x}}'')^\top \dot{\mathbf{x}}'' - 2(\dot{\mathbf{x}}'')^\top (\dot{\mathbf{s}} \boldsymbol{\kappa}) + (\dot{\mathbf{s}} \boldsymbol{\kappa}) \cdot (\dot{\mathbf{s}} \boldsymbol{\kappa})] ds \\ &= \frac{1}{2} \nu_1 \int_{s=0}^L \frac{\partial}{\partial \dot{\mathbf{x}}''} (\dot{\mathbf{x}}''^\top \dot{\mathbf{x}}'' - 2\dot{\mathbf{x}}''^\top \dot{\mathbf{s}} \boldsymbol{\kappa}) \frac{\partial \dot{\mathbf{x}}''}{\partial \dot{\mathbf{x}}} ds \\ &= \nu_1 \int_{s=0}^L (\dot{\mathbf{x}}'' - \dot{\mathbf{s}} \boldsymbol{\kappa})^\dagger \frac{\partial \dot{\mathbf{x}}}{\partial \dot{\mathbf{x}}} ds + \frac{1}{2} \nu_1 \left[ (\dot{\mathbf{x}}'' - \dot{\mathbf{s}} \boldsymbol{\kappa}) \frac{\partial \dot{\mathbf{x}}}{\partial \dot{\mathbf{x}}} \right]_{s=0}^{s=L} \end{aligned} \quad (5.97)$$

$$- \frac{1}{2} \nu_1 \left[ (\dot{\mathbf{x}}'' - \dot{\mathbf{s}} \boldsymbol{\kappa})^\dagger \frac{\partial \dot{\mathbf{x}}}{\partial \dot{\mathbf{x}}} \right]_{s=0}^{s=L} \quad (5.98)$$

$$= \nu_1 \int_{s=0}^L (\dot{\mathbf{x}}'' - \dot{\mathbf{s}} \boldsymbol{\kappa})^\dagger ds. \quad (5.99)$$

On (5.97) and (5.98) we used integration by parts and the boundary terms disappear because of the **Boundary Conditions** in (5.69) and (5.70).



We also have,

$$\begin{aligned}
\frac{\partial R}{\partial \dot{\mathbf{s}}} &= \frac{1}{2} \nu_1 \int_{s=0}^L \frac{\partial}{\partial \dot{\mathbf{s}}} \left[ \boldsymbol{\kappa}^\top \dot{\mathbf{s}}^\top \dot{\mathbf{s}} \boldsymbol{\kappa} - 2 \dot{\mathbf{x}}''^\top \dot{\mathbf{s}} \boldsymbol{\kappa} + \dot{\mathbf{x}}'' \cdot \dot{\mathbf{x}}'' \right] ds + \frac{1}{2} \nu_1 \int_{s=0}^L \frac{\partial}{\partial \dot{\mathbf{s}}} \dot{\tau}^2 ds \\
&=: \nu_1 \int_{s=0}^L \dot{\mathbf{s}} \boldsymbol{\kappa} \boldsymbol{\kappa}^\top - \dot{\mathbf{x}}'' \boldsymbol{\kappa}^\top ds + K \\
&= \nu_1 \int_{s=0}^L -(\dot{\mathbf{x}}'' - \dot{\mathbf{s}} \boldsymbol{\kappa}) \otimes \boldsymbol{\kappa} ds + K
\end{aligned} \tag{5.100}$$

Computing  $K$  is a little more subtle and we proceed separately.

Indeed,

$$K = \int_{s=0}^L \dot{\tau} \frac{\partial \dot{\tau}}{\partial \dot{\mathbf{s}}} ds$$

and note that we may write  $\tau = -\frac{1}{2} \text{trace}(\mathbf{s}^\top \mathbf{s}'_j)$ . Consequently we have

$$\begin{aligned}
\dot{\tau} &= -\frac{1}{2} [\text{trace}(\dot{\mathbf{s}}^\top \mathbf{s}'_j) + \text{trace}(\mathbf{s}^\top \dot{\mathbf{s}}'_j)] \\
\frac{\partial \dot{\tau}}{\partial \dot{\mathbf{s}}} &= -\frac{1}{2} \left[ \frac{\partial}{\partial \dot{\mathbf{s}}} \text{trace}(\dot{\mathbf{s}}^\top \mathbf{s}'_j) + \frac{\partial}{\partial \dot{\mathbf{s}}'} \text{trace}(\mathbf{s}^\top \dot{\mathbf{s}}'_j) \frac{\partial \dot{\mathbf{s}}'}{\partial \dot{\mathbf{s}}} \right] \\
&= -\frac{1}{2} \left[ \mathbf{s}'_j + \mathbf{s}_j^\top \frac{\partial \dot{\mathbf{s}}'}{\partial \dot{\mathbf{s}}} \right]
\end{aligned}$$

We can now compute  $K$  as follows.

$$K = -\frac{1}{2} \nu_2 \int_{s=0}^L \left[ \dot{\tau} \mathbf{s}'_j - \dot{\tau} \mathbf{s}_j \frac{\partial \dot{\mathbf{s}}'}{\partial \dot{\mathbf{s}}} \right] ds \tag{5.101}$$

$$= -\frac{1}{2} \nu_2 \int_{s=0}^L \left[ \dot{\tau} \mathbf{s}'_j + (\dot{\tau} \mathbf{s})'_j \frac{\partial \dot{\mathbf{s}}}{\partial \dot{\mathbf{s}}} \right] + \frac{1}{2} \nu_2 \left[ \dot{\tau} \mathbf{s}_j \frac{\partial \dot{\mathbf{s}}}{\partial \dot{\mathbf{s}}} \right]_{s=0}^{s=L} \tag{5.102}$$

$$= \nu_2 \int_{s=0}^L \frac{1}{2} [-\dot{\tau} \mathbf{s}'_j - (\dot{\tau} \mathbf{s})'_j] ds \tag{5.103}$$

$$= \nu_2 \int_{s=0}^L \frac{1}{2} \mathbf{f}(\dot{\tau}, \mathbf{s})_j ds \tag{5.104}$$

On (5.102) we used integration by parts. On (5.103) we used the **Boundary Con-**

**dition** (5.71). We may now write that

$$\frac{\partial R}{\partial \dot{\mathbf{s}}} = \nu_1 \int_{s=0}^L -(\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa})^{\dagger\dagger} ds + \nu_2 \int_{s=0}^L \frac{1}{2} \mathbf{f}(\dot{\tau}, \mathbf{s})_j ds.$$

We note that the damping terms that needs to be added to the equations of motion are of the form  $-\frac{\partial R}{\partial \dot{\mathbf{x}}}$  and  $-\frac{\partial R}{\partial \dot{\mathbf{s}}}$ . See for example in [58, chap. 1] and [47].

Finally (5.78) and (5.79) now follows as desired.  $\square$

A similar calculation for the discretized  $\tau, \mathbf{s}$  and  $\mathbf{x}$  case gives the desired formulas for the discrete case. The precise formulation is as follows.

**Theorem 13.** *Let  $\mathbf{L}$  be the discrete Lagrangian given by*

$$\begin{aligned} \mathbf{L} = & \frac{\eta}{2} \sum_{j=0}^N w_j \rho |\dot{\mathbf{x}}_j|^2 - \sigma_1 w_j |\mathbf{x}_j'' - \mathbf{s}_j \boldsymbol{\kappa}_j|^2 - \frac{\eta}{2} \sum_{j=1}^N 2\sigma_2 \tau_j^2 \\ & + \frac{\eta}{2} \sum_{j=0}^N \text{trace}(\mathbf{a}_j \mathbf{s}_j^\top \mathbf{s}_j) + 2\boldsymbol{\beta}_j^\top \mathbf{s}_j^\top \mathbf{x}'_j + \alpha_j |\mathbf{x}'_j|^2 \end{aligned} \quad (5.105)$$

where  $w_j$  is a weight which is  $\frac{1}{2}$  when  $j = 0, N$  and 1 for  $1 \leq j \leq N - 1$ . Taking the variation with respect to  $\mathbf{x}_0$  gives,

$$w_0 \rho \ddot{\mathbf{x}}_0 = -(\mathbf{s}_0 \boldsymbol{\beta}_0)^\dagger - (\alpha_0 \mathbf{x}'_0)^\dagger \quad (5.106)$$

taking the variation with respect to  $\mathbf{x}_j$  for  $1 \leq j \leq N - 1$  gives

$$\rho w_j \ddot{\mathbf{x}}_j = -\sigma_1 w_j (\mathbf{D}_2 \mathbf{x}_j - \mathbf{s}_j \boldsymbol{\kappa}_j)^{\dagger\dagger} - (\mathbf{s}_j \boldsymbol{\beta}_j)^\dagger - (\alpha_j \mathbf{x}'_j)^\dagger \quad (5.107)$$

and variation with respect to  $\mathbf{x}_N$  gives,

$$\rho w_N \ddot{\mathbf{x}}_N = -(\mathbf{s}_N \boldsymbol{\beta}_N)^\dagger - (\alpha_N \mathbf{x}'_N)^\dagger \quad (5.108)$$

With the addition of the damping terms (5.107) becomes

$$\rho w_j \ddot{\mathbf{x}}_j = -\sigma_1 w_j (\mathbf{D}_2 \mathbf{x}_j - \mathbf{s}_j \boldsymbol{\kappa}_j)^{\dagger\dagger} - (\mathbf{s}_j \boldsymbol{\beta}_j)^\dagger - (\alpha_j \mathbf{x}'_j)^\dagger - \nu_1 (\dot{\mathbf{x}}''_j - \dot{\mathbf{s}}_j \boldsymbol{\kappa}_j)^{\dagger\dagger} \quad (5.109)$$

for  $1 \leq j \leq N - 1$ . The equations in (5.108) and (5.106) remain unchanged as no damping terms contribute to it.

**Remark.** Note that the equations (5.109), (5.108) and (5.106) can be written in the following compact form without the use of sub-indices as follows:

$$\rho \mathbf{W} \ddot{\mathbf{x}} = -\sigma_1 \mathbf{W} \mathbf{E} (\mathbf{x}'' - \mathbf{s} \boldsymbol{\kappa})^{\dagger\dagger} - (\alpha \mathbf{x}')^\dagger - (\mathbf{s} \boldsymbol{\beta})^\dagger - \nu_1 \mathbf{E} (\dot{\mathbf{x}}'' - \dot{\mathbf{s}} \boldsymbol{\kappa})^{\dagger\dagger}. \quad (5.110)$$

**Remark.**  $\mathbf{E}$  is an  $(N + 1) \times (N + 1)$  diagonal matrix whose diagonal entries are 1 except for the top and bottom entries which are 0 according to the boundary conditions as stated in **Boundary Conditions** (1).

**Remark.**  $\mathbf{W}$  is an  $(N + 1) \times (N + 1)$  diagonal weight matrix whose entries are 1 except for the top and bottom entries which are  $\frac{1}{2}$ .

**Theorem 14.** Let  $\mathbf{L}$  be as defined in Theorem (13). Taking variation with respect to  $\mathbf{s}_0$  first and then adding damping terms gives

$$\sigma_2 \mathbf{f}_0(\tau, \mathbf{s})_j - \mathbf{D}_1 \mathbf{x}_0 \otimes \boldsymbol{\beta}_0 - \mathbf{s}_0 \mathbf{a}_0 = -\frac{1}{2} \nu_2 \mathbf{f}_0(\dot{\tau}, \mathbf{s})_j \quad (5.111)$$

and variation with respect to  $\mathbf{s}_j$  for  $1 \leq j \leq N - 1$  gives

$$-\sigma_1 w_j (\mathbf{D}_2 \mathbf{x}_j \otimes \boldsymbol{\kappa}_j) + \sigma_2 \mathbf{f}_0(\tau, \mathbf{s})_j - \mathbf{D}_1 \mathbf{x}_j \otimes \boldsymbol{\beta}_j - \mathbf{s}_j \mathbf{a}_j = 0 \quad (5.112)$$

with the addition of damping terms (5.112) becomes.

$$\begin{aligned}
& -\sigma_1 w_j (\mathbf{D}_2 \mathbf{x}_j \otimes \boldsymbol{\kappa}_j) + \sigma_2 \mathbf{f}_j(\tau, \mathbf{s})_j - \mathbf{D}_1 \mathbf{x}_j \otimes \boldsymbol{\beta}_j - \mathbf{s}_j \mathbf{a}_j \\
& = \nu_1 (\mathbf{D}_2 \dot{\mathbf{x}}_j - \dot{\mathbf{s}}_j \boldsymbol{\kappa}_j) \otimes \boldsymbol{\kappa}_j - \frac{1}{2} \nu_2 \mathbf{f}_j(\dot{\tau}, \mathbf{s})_j
\end{aligned} \tag{5.113}$$

and finally taking the variation with respect to  $\mathbf{s}_N$  and subsequently adding damping terms gives

$$\sigma_2 \mathbf{f}_N(\tau, \mathbf{s})_j - \mathbf{D}_1 \mathbf{x}_N \otimes \boldsymbol{\beta}_N - \mathbf{s}_N \mathbf{a}_N = -\frac{1}{2} \nu_2 \mathbf{f}_N(\dot{\tau}, \mathbf{s})_j \tag{5.114}$$

**Remark.** We can write the equations in (5.113), (5.111) and (5.114) in compact form as follows:

$$\begin{aligned}
& -\sigma_1 \mathbf{W} \mathbf{E} (\mathbf{D}_2 \mathbf{x} \otimes^* \boldsymbol{\kappa}) + \sigma_2 \mathbf{f}(\tau, \mathbf{s})_j - \mathbf{D}_1 \mathbf{x} \otimes^* \boldsymbol{\beta} - \mathbf{s} \mathbf{a} \\
& = \nu_1 \mathbf{E} (\mathbf{D}_2 \dot{\mathbf{x}} - \dot{\mathbf{s}} \boldsymbol{\kappa}) \otimes^* \boldsymbol{\kappa} - \frac{1}{2} \nu_2 \mathbf{f}(\dot{\tau}, \mathbf{s})_j
\end{aligned} \tag{5.115}$$

where  $\mathbf{E}$  and  $\mathbf{W}$  are as described in the remarks preceding Theorem (13).

**Remark.** Note that we have,

$$\mathbf{f}_j(\tau, \mathbf{s}) = \begin{cases} -\frac{1}{\eta} \tau_1 \mathbf{s}_1 & \text{for } j = 0, \\ -\frac{1}{\eta} [\tau_{j+1} \mathbf{s}_{j+1} - \tau_j \mathbf{s}_{j-1}] & \text{for } 1 \leq j \leq N-1, \\ \frac{1}{\eta} \tau_N \mathbf{s}_{N-1} & \text{for } j = N. \end{cases} \tag{5.116}$$

and a similar expression holds for  $\mathbf{f}_j(\dot{\tau}, \mathbf{s})$  where each  $\tau$  is replaced with a  $\dot{\tau}$  in (5.116).

**Remark.** We are using a slight abuse of notation in writing terms such as  $\mathbf{D}_1 \mathbf{x}_j$ .

What it actually means is the following:

$$[\mathbf{D}_1]_j \mathbf{x}_j$$

where the sub-index on  $D_1$  stands for taking the  $j$ th triplet of rows from the matrix  $D_1$ . For example  $[D_1]_0$  means take the first three rows and  $[D_1]_1$  means take the 4th, 5th and 6th rows etc. Since  $D_1$  is a  $3(N+1) \times 3(N+1)$  matrix, this makes sense for each  $0 \leq j \leq N$ .

## 5.6 Monotonocity of the Hamiltonian

Before we can prove the monotonicity of the Hamiltonian under the proposed discretization we need some auxiliary results.

**Proposition 15.** *Let  $\mathbf{x} \in \mathbb{R}^{3 \times 1}$ ,  $\mathbf{y} \in \mathbb{R}^{2 \times 1}$  and  $A \in \mathbb{R}^{3 \times 2}$ . Then*

$$\text{trace}(A^\top(\mathbf{x} \otimes \mathbf{y})) = \mathbf{x} \cdot (A\mathbf{y}).$$

In here  $\otimes$  stands for the outer product.

*Proof.* By definition  $\mathbf{x} \otimes \mathbf{y} = \mathbf{xy}^\top$ . Therefore

$$\begin{aligned} \text{trace}(A^\top(\mathbf{x} \otimes \mathbf{y})) &= \text{trace}(A^\top(\mathbf{xy}^\top)) \\ &= \text{trace}(\mathbf{xy}^\top A^\top) \end{aligned} \tag{5.117}$$

$$\begin{aligned} &= \text{trace}(\mathbf{x}(A\mathbf{y})^\top) \\ &= \text{trace}(\mathbf{x}^\top(A\mathbf{y})) \end{aligned} \tag{5.118}$$

Notice that  $\text{trace}(\mathbf{x}^\top(A\mathbf{y}))$  is a scalar and therefore,

$$\begin{aligned} \text{trace}(\mathbf{x}^\top(A\mathbf{y})) &= \mathbf{x}^\top A\mathbf{y} \\ &= \mathbf{x} \cdot A\mathbf{y} \end{aligned}$$

as required. Note that on (5.117) we are using the fact that  $\text{trace}(AB) = \text{trace}(BA)$  and on (5.118) we are using the fact that  $\text{trace}(X^\top Y) = \text{trace}(XY^\top)$ .  $\square$

**Proposition 16.** Let  $\mathbf{x} \in \mathbb{R}^{3 \times 1}$  and  $A \in \mathbb{R}^{3 \times 2}$ . Then,

$$\mathbf{x} \cdot (AA^\top \mathbf{x}) = (A^\top \mathbf{x}) \cdot (A^\top \mathbf{x})$$

*Proof.* Notice that  $\mathbf{x} \cdot (AA^\top \mathbf{x}) = \mathbf{x}^\top (AA^\top \mathbf{x}) = \mathbf{x}^\top AA^\top \mathbf{x}$ . Also note that  $(A^\top \mathbf{x}) \cdot (A^\top \mathbf{x}) = \mathbf{x}^\top AA^\top \mathbf{x}$ . The required equality now follows.  $\square$

**Proposition 17.** Let  $\mathbf{s}$  be as defined in definition (3) and let  $\boldsymbol{\kappa}$  be as defined in (5.63). Then

$$(\dot{\mathbf{s}}\boldsymbol{\kappa}) \cdot (\mathbf{s}\boldsymbol{\kappa}) = 0 \tag{5.119}$$

*Proof.* Note that  $\mathbf{s}^\top \mathbf{s} = \mathbf{i}_2$  and so  $\dot{\mathbf{s}}^\top \mathbf{s} + \mathbf{s}^\top \dot{\mathbf{s}} = 0$ . Thus,

$$\begin{aligned} (\dot{\mathbf{s}}\boldsymbol{\kappa}) \cdot (\mathbf{s}\boldsymbol{\kappa}) &= \frac{1}{2} [(\dot{\mathbf{s}}\boldsymbol{\kappa}) \cdot (\mathbf{s}\boldsymbol{\kappa}) + (\mathbf{s}\boldsymbol{\kappa}) \cdot (\dot{\mathbf{s}}\boldsymbol{\kappa})] \\ &= \frac{1}{2} [\boldsymbol{\kappa}^\top \dot{\mathbf{s}}^\top \mathbf{s} \boldsymbol{\kappa} + (\boldsymbol{\kappa}^\top \mathbf{s}^\top \dot{\mathbf{s}} \boldsymbol{\kappa})] \\ &= \frac{1}{2} [\boldsymbol{\kappa}^\top (\dot{\mathbf{s}}^\top \mathbf{s} + \mathbf{s}^\top \dot{\mathbf{s}}) \boldsymbol{\kappa}] \\ &= 0 \end{aligned}$$

as desired.  $\square$

**Proposition 18.** Let  $\mathbf{a}$  be a symmetric matrix and let  $\mathbf{s}$  be as defined above. then,

$$\text{trace}(\dot{\mathbf{s}}^\top \mathbf{s} \mathbf{a}) = \text{trace} \left( \mathbf{a} \frac{\partial}{\partial t} (\mathbf{s}^\top \mathbf{s}) \right).$$

*Proof.* Note that  $\dot{\mathbf{s}}^\top \mathbf{s} + \mathbf{s}^\top \dot{\mathbf{s}} = 0$  as observed previously. It now follows that  $\mathbf{s}^\top \dot{\mathbf{s}}$  is anti-symmetric. Consequently we have  $\text{trace}(\mathbf{a}(\mathbf{s}^\top \dot{\mathbf{s}})) = 0$ , since the trace of a symmetric matrix times an anti-symmetric matrix vanishes. The given identity now follows.  $\square$

We first prove the continuous version of the fact that the Hamiltonian is monotone

under the given space discretizations. The discrete version follows immediately because all of the arguments follow formally the same. Simply replace integration by parts with summation by parts and use the necessary discrete version of the corresponding propositions and the discrete versions of the **Boundary Conditions 1**.

**Theorem 19** (Monotonicity of the Hamiltonian). *Under the given space discretizations, the Hamiltonian*

$$H(t) := \text{K.E} + \text{P.E}$$

*is monotone.*

*Proof.* Note that

$$\frac{\partial}{\partial t}(\text{K.E}) = \rho \int_0^T \frac{1}{2} \frac{\partial}{\partial t} |\dot{\mathbf{x}}|^2 ds = \rho \int_0^T (\dot{\mathbf{x}} \cdot \ddot{\mathbf{x}}) ds.$$

Now let us dot product equation (5.78) by  $\dot{\mathbf{x}}$ , multiply (5.79) by  $\dot{\mathbf{s}}^\top$  and take the trace,

add the two equations, and integrate from 0 to  $L$ . Then,

$$\begin{aligned}
\frac{\partial}{\partial t}(\text{K.E}) &= \int_0^L \dot{\mathbf{x}} \cdot [-\sigma_1(\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})^{\dagger\dagger} - (\boldsymbol{\alpha}\mathbf{x}')^\dagger - (\mathbf{s}\boldsymbol{\beta})^\dagger] ds \\
&\quad - \text{trace} \left[ \int_0^L \dot{\mathbf{s}}^\top (-\sigma_1(\mathbf{x}'' \otimes \boldsymbol{\kappa}) + \sigma_2 \mathbf{f}(\tau, \mathbf{s}) \mathbf{j} - \mathbf{x}' \otimes \boldsymbol{\beta} - \mathbf{s} \mathbf{a}) \right] ds \\
&\quad + \int_0^L \dot{\mathbf{x}} \cdot [-E\nu_1(\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa})^{\dagger\dagger}] ds \\
&\quad + \text{trace} \left[ \int_0^L \dot{\mathbf{s}}^\top (-E((-\nu_1(\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa}) \otimes \boldsymbol{\kappa}) + \frac{1}{2}\nu_2 \mathbf{f}(\dot{\tau}, \mathbf{s}) \mathbf{j})) \right] ds \\
&= \int_0^L -\sigma_1 \dot{\mathbf{x}}'' \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) + \dot{\mathbf{x}}' \cdot (\boldsymbol{\alpha}\mathbf{x}') + \dot{\mathbf{x}}' \cdot (\mathbf{s}\boldsymbol{\beta}) ds \tag{5.120}
\end{aligned}$$

$$- \int_0^L -\sigma_1 \mathbf{x}'' \cdot \dot{\mathbf{s}}\boldsymbol{\kappa} + \sigma_2 2\tau\dot{\tau} - \mathbf{x}' \cdot (\dot{\mathbf{s}}\boldsymbol{\beta}) - \text{trace}(\dot{\mathbf{s}}^\top \mathbf{s} \mathbf{a}) ds \tag{5.121}$$

$$- \nu_1 \int_0^L \dot{\mathbf{x}}'' \cdot (\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa}) ds + \nu_1 \int_0^L (\dot{\mathbf{x}}'' - \dot{\mathbf{s}}\boldsymbol{\kappa}) \cdot (\dot{\mathbf{s}}\boldsymbol{\kappa}) - \nu_2 \int_0^L \dot{\tau}^2 ds \tag{5.122}$$

$$= \frac{1}{2} \frac{\partial}{\partial t} \int_0^L -\sigma_1 |\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}|^2 - 2\sigma_2 \tau^2 ds \tag{5.123}$$

$$+ \frac{1}{2} \int_0^T \boldsymbol{\alpha} \frac{\partial}{\partial t} |\mathbf{x}'|^2 + 2\boldsymbol{\beta} \cdot \frac{\partial}{\partial t} (\mathbf{s}^\top \mathbf{x}') + \text{trace} \left( \mathbf{a} \frac{\partial}{\partial t} (\mathbf{s}^\top \mathbf{s}) \right) ds \tag{5.124}$$

$$= -\frac{\partial}{\partial t} \text{P.E} - \nu_1 \int_0^L \left| \frac{\partial}{\partial t} (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \right|^2 ds - \nu_2 \int_0^L \dot{\tau}^2 ds.$$

On (5.120) we used integration by parts. For equalities in (5.121) and (5.122) we used theorem (9) and proposition (15) respectively. On line (5.124) all of the terms vanish because of the constraints stated in (5.72). For the equality in line (5.123) we used the following set of equalities:

$$\frac{1}{2} \frac{\partial}{\partial t} ((\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa}) \cdot (\mathbf{x}'' - \mathbf{s}\boldsymbol{\kappa})) = \dot{\mathbf{x}}'' \cdot \mathbf{x}'' - (\dot{\mathbf{x}}'' \cdot (\mathbf{s}\boldsymbol{\kappa}) + \mathbf{x}'' \cdot (\dot{\mathbf{s}}\boldsymbol{\kappa})) + (\dot{\mathbf{s}}\boldsymbol{\kappa}) \cdot (\mathbf{s}\boldsymbol{\kappa})$$

But then by proposition (17) we have that

$$(\dot{\mathbf{s}}\boldsymbol{\kappa}) \cdot (\mathbf{s}\boldsymbol{\kappa}) = 0.$$



□

## 5.7 The time discretization

The time discretization may be done as follows:

$$\begin{aligned}\mathbf{y} &\rightarrow \frac{\mathbf{y}_{n+1} + \mathbf{y}_{n-1}}{2} \\ \dot{\mathbf{y}} &\rightarrow \frac{\mathbf{y}_{n+1} - \mathbf{y}_{n-1}}{2h} \\ \ddot{\mathbf{y}} &\rightarrow \frac{\mathbf{y}_{n+1} - 2\mathbf{y}_n + \mathbf{y}_{n-1}}{h^2}\end{aligned}$$

**Notation.** Let  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ ,  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$  and  $\mathfrak{S} = [\mathfrak{s}_1, \mathfrak{s}_2, \dots, \mathfrak{s}_N]$  with  $\alpha_j \in \mathbb{R}$ ,  $\mathbf{x}_j \in \mathbb{R}^{3 \times 1}$  and  $\mathfrak{s}_j \in \mathbb{R}^{2 \times 3}$  for  $0 \leq j \leq N$ . Then,  $*$  is defined as the context dependent pointwise multiplication given by,

$$\boldsymbol{\alpha} * \mathbf{x} = [\alpha_0 \mathbf{x}_0, \alpha_1 \mathbf{x}_1, \dots, \alpha_N \mathbf{x}_N]$$

$$\mathfrak{S} * \mathbf{x} = [\mathfrak{s}_1 \mathbf{x}_1, \mathfrak{s}_2 \mathbf{x}_2, \dots, \mathfrak{s}_N \mathbf{x}_N].$$

Let  $\mathfrak{K} = [\boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2, \dots, \boldsymbol{\kappa}_N]$  with  $\boldsymbol{\kappa}_j \in \mathbb{R}^{2 \times 1}$  for  $0 \leq j \leq N$ . Then  $\otimes^*$  is defined as the pointwise outer product given by,

$$\mathfrak{S} \otimes^* \mathfrak{K} = [\mathfrak{s}_1 \otimes \boldsymbol{\kappa}_1, \mathfrak{s}_2 \otimes \boldsymbol{\kappa}_2, \dots, \mathfrak{s}_N \otimes \boldsymbol{\kappa}_N].$$

**Notation.** let  $\mathbf{s} = (\mathfrak{s}_j)_{0 \leq j \leq N}$  be as defined in Definition 3. Then by  $\mathfrak{s}_n[j]$  we mean the  $j$ th space discretization step of  $s_j$  at the  $n$ th time discretization step.  $\tau_n[j]$  is similarly defined.

The full discretization of (5.78) is as follows:

$$\begin{aligned}
& \rho W \left( \frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{h^2} \right) \\
& + \text{WE} \left[ \sigma_1 \left( D_2^\dagger \left( D_2 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) - \frac{1}{2} \begin{bmatrix} (\mathbf{s}_{n+1}[0] + \mathbf{s}_{n-1}[0]) \boldsymbol{\kappa}_0[0] \\ (\mathbf{s}_{n+1}[1] + \mathbf{s}_{n-1}[1]) \boldsymbol{\kappa}_0[1] \\ \vdots \\ (\mathbf{s}_{n+1}[N] + \mathbf{s}_{n-1}[N]) \boldsymbol{\kappa}_0[N] \end{bmatrix} \right) \right) \right] \\
& - \left[ D_1^\dagger \left( \boldsymbol{\alpha}_{n+1} * D_1 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) \right) + \frac{1}{2} D_1^\dagger \begin{bmatrix} (\mathbf{s}_{n+1}[0] + \mathbf{s}_{n-1}[0]) \boldsymbol{\beta}[0] \\ (\mathbf{s}_{n+1}[1] + \mathbf{s}_{n-1}[1]) \boldsymbol{\beta}[1] \\ \vdots \\ (\mathbf{s}_{n+1}[N] + \mathbf{s}_{n-1}[N]) \boldsymbol{\beta}[N] \end{bmatrix} \right] \\
& + \nu_1 \text{ED}_2^\dagger \left[ D_2 \left( \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2h} \right) - \frac{1}{2h} \begin{bmatrix} (\mathbf{s}_{n+1}[0] - \mathbf{s}_{n-1}[0]) \boldsymbol{\kappa}_0[0] \\ (\mathbf{s}_{n+1}[1] - \mathbf{s}_{n-1}[1]) \boldsymbol{\kappa}_0[1] \\ \vdots \\ (\mathbf{s}_{n+1}[N] - \mathbf{s}_{n-1}[N]) \boldsymbol{\kappa}_0[N] \end{bmatrix} \right] \quad (5.125)
\end{aligned}$$

$$= \mathbf{0}_{3(N+1) \times 1} \quad (5.126)$$

And the complete discretization of (5.79) is as follows:

$$\begin{aligned}
& \text{WE} \left[ -\sigma_1 \mathbf{D}_2 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) \otimes^* \begin{bmatrix} \boldsymbol{\kappa}_0[0] \\ \boldsymbol{\kappa}_0[1] \\ \vdots \\ \boldsymbol{\kappa}_0[N] \end{bmatrix} + \sigma_2 \begin{bmatrix} \mathbf{f}^{(d)}(\tau, \mathbf{s})[0]j \\ \mathbf{f}^{(d)}(\tau, \mathbf{s})[1]j \\ \vdots \\ \mathbf{f}^{(d)}(\tau, \mathbf{s})[N]j \end{bmatrix} \right] \\
& - \left[ \mathbf{D}_1 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) \otimes^* \begin{bmatrix} \boldsymbol{\beta}[0] \\ \boldsymbol{\beta}[1] \\ \vdots \\ \boldsymbol{\beta}[N] \end{bmatrix} + \frac{1}{2} \begin{bmatrix} (\mathbf{s}_{n+1}[0] + \mathbf{s}_{n-1}[0])\mathbf{a}[0] \\ (\mathbf{s}_{n+1}[1] + \mathbf{s}_{n-1}[1])\mathbf{a}[1] \\ \vdots \\ (\mathbf{s}_{n+1}[N] + \mathbf{s}_{n-1}[N])\mathbf{a}[N] \end{bmatrix} \right] \\
& - \nu_1 \mathbf{E} \left( \mathbf{D}_2 \left( \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2h} \right) - \frac{1}{2h} \begin{bmatrix} (\mathbf{s}_{n+1}[0] - \mathbf{s}_{n-1}[0])\boldsymbol{\kappa}_0[0] \\ (\mathbf{s}_{n+1}[1] - \mathbf{s}_{n-1}[1])\boldsymbol{\kappa}_0[1] \\ \vdots \\ (\mathbf{s}_{n+1}[N] - \mathbf{s}_{n-1}[N])\boldsymbol{\kappa}_0[N] \end{bmatrix} \right) \otimes^* \begin{bmatrix} \boldsymbol{\kappa}_0[0] \\ \boldsymbol{\kappa}_0[1] \\ \vdots \\ \boldsymbol{\kappa}_0[N] \end{bmatrix} \right) \\
& \hspace{15em} (5.127)
\end{aligned}$$

$$+ \frac{1}{2} \nu_2 \begin{bmatrix} \mathbf{f}^{(d)}(\dot{\tau}, \mathbf{s})[0]j \\ \mathbf{f}^{(d)}(\dot{\tau}, \mathbf{s})[1]j \\ \vdots \\ \mathbf{f}^{(d)}(\dot{\tau}, \mathbf{s})[N]j \end{bmatrix} = \mathbf{0}_{3(N+1) \times 2} \quad (5.128)$$

where  $\mathbf{f}^{(d)}(\tau, \mathbf{s})[j]$ , and  $\mathbf{f}^{(d)}(\dot{\tau}, \mathbf{s})[j]$  stands for the full discretization of  $\mathbf{f}$  terms both with respect to space and time. Also note that  $\boldsymbol{\alpha}_{n+1} = [\alpha_0, \alpha_1, \dots, \alpha_N]$  and the subscript  $n + 1$  on the left means the set of values for  $\boldsymbol{\alpha}$  at time  $n + 1$ .

**Remark.** *We have presented the full discretization here for the sake of completeness. The computational implementation will be similar to the original equations in (5.78) and in (5.79).*

### 5.7.1 Discretization of $f_j(\tau, \mathbf{s})$ and $f_j(\dot{\tau}, \mathbf{s})$

#### 5.7.1.1 Full discretization of $f_j(\tau, \mathbf{s})$ :

We have that

$$f_j(\tau, \mathbf{s}) = \begin{cases} -\frac{1}{\eta}\tau_1\mathbf{s}_1 & \text{for } j = 0, \\ -\frac{1}{\eta}[\tau_{j+1}\mathbf{s}_{j+1} - \tau_j\mathbf{s}_{j-1}] & \text{for } 1 \leq j \leq N - 1, \\ \frac{1}{\eta}\tau_N\mathbf{s}_{N-1} & \text{for } j = N. \end{cases} \quad (5.129)$$

The full discretization of  $f_j(\tau, \mathbf{s})$  is as follows:

$$f_j^{(d)}(\tau, \mathbf{s}) = \begin{cases} -\frac{1}{\eta}\bar{\tau}_1\bar{\mathbf{s}}_1 & \text{for } j = 0, \\ -\frac{1}{\eta}[\bar{\tau}_{j+1}\bar{\mathbf{s}}_{j+1} - \bar{\tau}_j\bar{\mathbf{s}}_{j-1}] & \text{for } 1 \leq j \leq N - 1, \\ \frac{1}{\eta}\bar{\tau}_N\bar{\mathbf{s}}_{N-1} & \text{for } j = N. \end{cases} \quad (5.130)$$

where

$$\bar{\tau} = \frac{\tau_{n+1} + \tau_{n-1}}{2} \quad (5.131)$$

$$\bar{\mathbf{s}} = \frac{\mathbf{s}_{n+1} + \mathbf{s}_{n-1}}{2} \quad (5.132)$$

#### 5.7.1.2 Full discretization of $f_j(\dot{\tau}, \mathbf{s})$ :

For  $0 \leq j \leq N$  set,

$$\begin{aligned} f_j^{(d)}(\dot{\tau}, \mathbf{s}) &= -\frac{1}{4\eta h} ((\tau_{n+1}[j+1] - \tau_{n-1}[j+1])(\mathbf{s}_{n+1}[j+1] + \mathbf{s}_{n-1}[j+1])) \\ &\quad + \frac{1}{4\eta h} ((\tau_{n+1}[j] - \tau_{n-1}[j])(\mathbf{s}_{n+1}[j-1] + \mathbf{s}_{n-1}[j-1])) \end{aligned} \quad (5.133)$$

for  $j = 0$  set,

$$\mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s}) = -\frac{1}{4\eta h} [(\tau_{n+1}[1] - \tau_{n-1}[1])(\mathbf{s}_{n+1}[1] + \mathbf{s}_{n-1}[1])] \quad (5.134)$$

and for  $j = N$  set,

$$\mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s}) = \frac{1}{4\eta h} [(\tau_{n+1}[N] - \tau_{n-1}[N])(\mathbf{s}_{n+1}[N-1] + \mathbf{s}_{n-1}[N-1])]. \quad (5.135)$$

The discretization of  $\mathbf{f}_j(\dot{\tau}, \mathbf{s})$  may be written compactly as follows:

$$\mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s}) = \begin{cases} -\frac{1}{\eta} \bar{\tau}_1 \bar{\mathbf{s}}_1 & \text{for } j = 0, \\ -\frac{1}{\eta} [\bar{\tau}_{j+1} \bar{\mathbf{s}}_{j+1} - \bar{\tau}_j \bar{\mathbf{s}}_{j-1}] & \text{for } 1 \leq j \leq N-1, \\ \frac{1}{\eta} \bar{\tau}_N \bar{\mathbf{s}}_{N-1} & \text{for } j = N. \end{cases} \quad (5.136)$$

where  $\bar{\tau}$  stands for

$$\bar{\tau} = \frac{\tau_{n+1} - \tau_{n-1}}{2h} \quad (5.137)$$

and the superscript  $(d)$  suggestively stands for *discretized*.

We can now write the full discretizations of both equations in a more compact way as follows. Let

$$\begin{aligned} \mathbf{S}_n &= [\mathbf{s}_n[0], \mathbf{s}_n[1], \dots, \mathbf{s}_n[N]]^\top, \\ \mathbf{K}_0 &= [\boldsymbol{\kappa}_0[0], \boldsymbol{\kappa}_0[1], \dots, \boldsymbol{\kappa}_0[N]]^\top, \\ \mathbf{B}_{n+1} &= [\boldsymbol{\beta}[0], \boldsymbol{\beta}[1], \dots, \boldsymbol{\beta}[N]]^\top, \\ \mathbf{A}_{n+1} &= [\mathbf{a}[0], \mathbf{a}[1], \dots, \mathbf{a}[N]]^\top, \\ \mathbf{J} &= [j, j, \dots, j]^\top, \\ \mathbf{F}^{(d)}(\tau, \mathbf{s}) &= [\mathbf{f}_0^{(d)}(\tau, \mathbf{s}), \mathbf{f}_1^{(d)}(\tau, \mathbf{s}), \dots, \mathbf{f}_N^{(d)}(\tau, \mathbf{s})]^\top, \end{aligned}$$

where the sub-index  $n$  stands for the  $n$ th time step. We can now rewrite (5.126) as follows. Notice that we are not time discretizing  $\mathbf{a}$ 's,  $\boldsymbol{\beta}$ 's and  $\boldsymbol{\alpha}$ 's.

$$\begin{aligned}
& \rho W \left( \frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{h^2} \right) \\
& + WE \left[ \sigma_1 \left( D_2^\dagger \left( D_2 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) - \frac{1}{2}(\mathbf{S}_{n+1} + \mathbf{S}_{n-1}) * \mathbf{K}_0 \right) \right) \right] \\
& - \left[ D_1^\dagger \left( \boldsymbol{\alpha}_{n+1} \odot D_1 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) \right) + \frac{1}{2} D_1^\dagger \left( (\mathbf{S}_{n+1} + \mathbf{S}_{n-1}) * \mathbf{B}_{n+1} \right) \right] \\
& + E\nu_1 D_2^\dagger \left[ D_2 \left( \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2h} \right) - \frac{1}{2h}(\mathbf{S}_{n+1} - \mathbf{S}_{n-1}) * \mathbf{K}_0 \right] \tag{5.138}
\end{aligned}$$

$$= \mathbf{0}_{3(N+1) \times 1} \tag{5.139}$$

In the preceding equation  $\odot$  stands for the following: The column vector on the right hand side argument of  $\odot$  is first sliced into vectors each of which is a 3-vector. Then take the usual element-wise matrix multiplication with each element from the left hand side.

We can rewrite (5.128) as follows:

$$\begin{aligned}
& WE \left[ -\sigma_1 D_2 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) \otimes^* \mathbf{K}_0 \right] + \sigma_2 (\mathbf{F}^{(d)}(\tau, \mathbf{s}) * \mathbf{J} \\
& - \left[ D_1 \left( \frac{\mathbf{x}_{n+1} + \mathbf{x}_{n-1}}{2} \right) \otimes^* \mathbf{B}_{n+1} + \frac{1}{2} \left( (\mathbf{S}_{n+1} + \mathbf{S}_{n-1}) * \mathbf{A}_{n+1} \right) \right] \\
& - E\nu_1 \left[ \left( D_2 \left( \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2h} \right) - \frac{1}{2h}(\mathbf{S}_{n+1} - \mathbf{S}_{n-1}) * \mathbf{K}_0 \right) \otimes^* \mathbf{K}_0 \right] \\
& + \left[ \frac{1}{2} \nu_2 \mathbf{F}^{(d)}(\dot{\tau}, \mathbf{s}) * \mathbf{J} \right] = \mathbf{0}_{3(N+1) \times 2} \tag{5.140}
\end{aligned}$$

## 5.7.2 Initial Conditions

Let us now discuss the initials conditions. These are normally specified by assuming we know  $\mathbf{x} = \mathbf{F}_1$ ,  $\mathbf{S} = \mathbf{F}_2$  and  $\dot{\mathbf{x}} = \mathbf{G}_1$ ,  $\dot{\mathbf{S}} = \mathbf{G}_2$  at  $t = 0$ , where  $\mathbf{F}_1$ ,  $\mathbf{F}_2$ ,  $\mathbf{G}_1$  and  $\mathbf{G}_2$  are given. To guarantee that the numerical method is still order 2 in time, the discrete

versions of these conditions are

$$\mathbf{x}_0 = \mathbf{F}_1 \tag{5.141}$$

$$\mathbf{S}_0 = \mathbf{F}_2 \tag{5.142}$$

$$\mathbf{x}_1 - \mathbf{x}_{-1} = 2h\mathbf{G}_1 \tag{5.143}$$

$$\mathbf{S}_1 - \mathbf{S}_{-1} = 2h\mathbf{G}_2 \tag{5.144}$$

The physical meaning of  $\mathbf{F}_1$  is that it is the initial state of the spring.  $\mathbf{F}_2$  is extracted from the *initial frame* chosen. Let us assume for now that  $\mathbf{G}_1$  and  $\mathbf{G}_2$  are zero.

# Chapter 6

## Computing the Jacobian of the system

### 6.1 Motivation for computing Jacobian

The software algorithm that we use to model the equations of motion use is the Levenberg-Marquardt algorithm. This method interpolates between the Gauss-Newton algorithm and the gradient descent algorithm. In particular it uses the Jacobian of the system of equations to find the ‘direction’ it needs to descent in order to find the (local) minimum. Therefore if an analytical Jacobian has not been provided to the solver it numerically approximates the Jacobian of the system at every iteration. This calculation happens many thousands of times during this process. Needless to say, this is a computationally a very inefficient process. The solution for this problem is to explicitly provide the Jacobian of the system to the solver so that it can look up the Jacobian instantly rather than approximating it. We have found drastic increases in performance by providing the Jacobian. The results obtained were also of much higher accuracy.

It is with this motivation that we embark on computing the Jacobian of the system



of equations. Even though this calculation has nothing to do with the actual mathematical analysis of the numerical method it plays an integral part in the software implementation of the problem. The calculations are rather tedious. But then can be done with a few results that stem from matrix calculus.

## 6.2 Notation and simplifications

For the sake of convenience let us denote the system of equations given in (5.139) as  $f_1 = \mathbf{0}$  and the equation given in (5.140) as  $f_2 = \mathbf{0}$ . We have that  $\boldsymbol{\alpha}_{n+1} = (\alpha_{n+1}[0], \alpha_{n+1}[1], \dots, \alpha_{n+1}[N])$ . For simplicity, let us also suppress the time dependence on the right hand side of the preceding equation and write  $\boldsymbol{\alpha}_{n+1} = (\alpha_0, \alpha_1, \dots, \alpha_N)$ . Denote by

$$\text{diag}(\boldsymbol{\alpha}_{n+1}) = \text{diag}(\alpha_0, \alpha_1, \dots, \alpha_N) \otimes \mathbf{i}_3 \quad (6.1)$$

Let us also agree that the sub index  $n$  always denotes time discretization. In particular  $n = 0$  means the initial state of the system and  $n \neq 0$  means non initials states of the system. Note that we need to make this distinction clear in our calculations as the equations turn out to be different in each case and the computer program needs to be explicitly made aware of the difference.

## 6.3 Derivatives of the first equation $f_1$

### 6.3.1 Computing $\frac{\partial f_1}{\partial \mathbf{x}_{n+1}}$ :

This computation is fairly straight forward. Indeed for  $n \neq 0$  we have,

$$\frac{\partial f_1}{\partial \mathbf{x}_{n+1}} = \frac{1}{h^2} \rho \mathbf{W} + \frac{1}{2} \sigma_1 \mathbf{W} \mathbf{E} \mathbf{D}_2^\top \mathbf{D}_2 - \frac{1}{2} \mathbf{D}_1^\top \text{diag}(\boldsymbol{\alpha}_{n+1}) \mathbf{D}_1 + \frac{\nu_1}{2h} \mathbf{E} \mathbf{D}_2^\top \mathbf{D}_2. \quad (6.2)$$

Note that by design  $\text{diag}(\boldsymbol{\alpha}_{n+1})$  is an  $3(N+1) \times 3(N+1)$  matrix. Thus,  $\frac{\partial f_1}{\partial \mathbf{x}_{n+1}}$  is also a  $3(N+1) \times 3(N+1)$  matrix as desired. For  $n = 0$  we have,

$$\frac{\partial f_1}{\partial \mathbf{x}_{n+1}} = \frac{2}{h^2} \rho \mathbf{W} + \sigma_1 \mathbf{W} \mathbf{E} \mathbf{D}_2^\top \mathbf{D}_2 - \mathbf{D}_1^\top \text{diag}(\boldsymbol{\alpha}_{n+1}) \mathbf{D}_1. \quad (6.3)$$

### 6.3.2 Computing $\frac{\partial f_1}{\partial \mathbf{S}_{n+1}}$ :

To compute this derivative we proceed as follows. Denote  $\mathbf{K}_0 = [k_0, k_1, \dots, k_N]$ . Then by definition each  $k_j$  for  $0 \leq j \leq N$  is a  $2 \times 1$  vector. Now use the symbol  $\text{diag}(\mathbf{K}; \mathbf{i}_3)$  in the following sense.

$$\begin{aligned} \text{diag}(\mathbf{K}_0; \mathbf{i}_3) &\stackrel{\text{def}}{=} \text{diag}(k_0^\top \otimes \mathbf{i}_3, k_1^\top \otimes \mathbf{i}_3, \dots, k_N^\top \otimes \mathbf{i}_3) \\ &= \text{diag}(k_0, k_1, \dots, k_N)^\top \otimes \mathbf{i}_3 \end{aligned}$$

Note that  $\text{diag}(\mathbf{K}_0; \mathbf{i}_3)$  thus defined is a  $3(N+1) \times 6(N+1)$  matrix.

For  $n \neq 0$  we now have,

$$\begin{aligned} \frac{\partial f_1}{\partial \mathbf{S}_{n+1}} &= -\frac{1}{2} \sigma_1 \mathbf{W} \mathbf{E} \mathbf{D}_2^\top \text{diag}(\mathbf{K}_0; \mathbf{i}_3) - \frac{1}{2} \mathbf{D}_1^\top \text{diag}(\mathbf{B}_{n+1}; \mathbf{i}_3) \\ &\quad - \frac{\nu_1}{2h} \mathbf{E} \mathbf{D}_2^\top \text{diag}(\mathbf{K}_0; \mathbf{i}_3). \end{aligned} \quad (6.4)$$

Note that all of the matrices that involve the operator  $\text{diag}(\cdot; \cdot)$  is an  $3(N+1) \times 6(N+1)$  matrix and so  $\frac{\partial f_1}{\partial \mathbf{S}_{n+1}}$  is an  $3(N+1) \times 6(N+1)$  matrix as required. For  $n = 0$  we have

$$\frac{\partial f_1}{\partial \mathbf{S}_{n+1}} = -\sigma_1 \mathbf{W} \mathbf{E} \mathbf{D}_2^\top \text{diag}(\mathbf{K}_0; \mathbf{i}_3) - \mathbf{D}_1^\top \text{diag}(\mathbf{B}_{n+1}; \mathbf{i}_3). \quad (6.5)$$

### 6.3.3 Computing $\frac{\partial f_1}{\partial \mathbf{A}_{n+1}}$

Since  $f_1$  does not depend on  $\mathbf{A}_{n+1}$  we have,

$$\frac{\partial f_1}{\partial \mathbf{A}_{n+1}} = \mathbf{0}_{3(N+1) \times 3(N+1)} \quad (6.6)$$

for all  $n$ .

### 6.3.4 Computing $\frac{\partial f_1}{\partial \mathbf{B}_{n+1}}$

For  $n \neq 0$  we have,

$$\frac{\partial f_1}{\partial \mathbf{B}_{n+1}} = -\frac{1}{2} \mathbf{D}_1^\top \text{diag}(\mathbf{S}_{n+1} + \mathbf{S}_{n-1}). \quad (6.7)$$

Equation (6.7) is an  $3(N+1) \times 2(N+1)$  matrix as required. For  $n = 0$  we have,

$$\frac{\partial f_1}{\partial \mathbf{B}_{n+1}} = -\mathbf{D}_1^\top \text{diag}(\mathbf{S}_{n+1}). \quad (6.8)$$

### 6.3.5 Computing $\frac{\partial f_1}{\partial \alpha_{n+1}}$

Consider the term

$$y := -\frac{1}{2} \mathbf{D}_1^\top (\boldsymbol{\alpha}_{n+1} \odot \mathbf{D}_1(\mathbf{x}_{n+1} + \mathbf{x}_{n-1})).$$

Using the notation used in (6.1) we can rewrite this as follows.

$$y = -\frac{1}{2} \mathbf{D}_1^\top (\text{diag}(\alpha_0, \alpha_1, \dots, \alpha_N) \otimes \mathbf{i}_3) \mathbf{D}_1(\mathbf{x}_{n+1} + \mathbf{x}_{n-1}) \quad (6.9)$$

For  $n \neq 0$  we have that,

$$\frac{\partial f_1}{\partial \alpha_i} = -\frac{1}{2} \mathbf{D}_1^\top (\text{diag}[i] \otimes \mathbf{i}_3) \mathbf{D}_1 (\mathbf{x}_{n+1} + \mathbf{x}_{n-1}) \quad (6.10)$$

and for  $n = 0$ ,

$$\frac{\partial f_1}{\partial \alpha_i} = -\mathbf{D}_1^\top (\text{diag}[i] \otimes \mathbf{i}_3) \mathbf{D}_1 (\mathbf{x}_{n+1}). \quad (6.11)$$

In here  $\text{diag}[i]$  denotes a  $(N+1) \times (N+1)$  diagonal matrix with 1 in the  $i$ th position and zeros everywhere else for  $0 \leq i \leq N$ . Note that  $\frac{\partial f_1}{\partial \alpha_i}$  thus computed is a  $3(N+1) \times 1$  matrix. We can now write,

$$\frac{\partial f_1}{\partial \boldsymbol{\alpha}_{n+1}} = \left[ \frac{\partial f_1}{\partial \alpha_0}, \frac{\partial f_1}{\partial \alpha_1}, \dots, \frac{\partial f_1}{\partial \alpha_N} \right]_{3(N+1) \times (N+1)} \quad (6.12)$$

where we take care to use one of (6.10) or (6.11) depending on whether  $n \neq 0$  or  $n = 0$ .

## 6.4 Derivatives of the second equation $f_2$

**Notation.** Let  $\mathbf{D}$  be an  $3(N+1) \times 3(N+1)$  matrix. Then note that we can group the rows of  $\mathbf{D}$  in to  $N+1$  groups where each group contain 3 consecutive rows. Denote by  $[\mathbf{D}]_j$  the  $j$ th such group for  $0 \leq j \leq N$ . More precisely  $[\mathbf{D}]_0$  denotes the first three rows,  $[\mathbf{D}]_1$  denotes the group consisting of 4th, 5th and 6th rows etc. In particular  $[\mathbf{D}]_j$  is a  $3 \times 3(N+1)$  matrix.

**Proposition 20.** Let  $\mathbf{D}$  be a  $3(N+1) \times 3(N+1)$  matrix and let  $\mathbf{x} \in \mathbb{R}^{3(N+1)}$ . Let  $\mathbf{K}_0 = [k_0, k_1, \dots, k_N]^\top$  where each  $k_i$  is a  $2 \times 1$  vector. Now consider the expression

$$\mathbf{y} := (\mathbf{D}\mathbf{x}) \otimes^* \mathbf{K}_0 \quad (6.13)$$

Then,

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = [k_0 \otimes [\mathbf{D}]_0, k_1 \otimes [\mathbf{D}]_1, \dots, k_N \otimes [\mathbf{D}]_N]^\top \quad (6.14)$$

Notice that each  $k_i \otimes [\mathbf{D}]_i$  is an  $6 \times 3(N + 1)$  matrix and this results in  $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$  being a  $6(N + 1) \times 3(N + 1)$  matrix.

*Proof.* We will show the result for the case when  $N = 1$ . The general result easily follows by induction on  $N$ . In this case  $\mathbf{D}$  will be a  $6 \times 6$  matrix and  $\mathbf{x}$  will be a  $6 \times 1$  column vector. Let  $\mathbf{D} = (d_{ij})$  and  $\mathbf{x} = (x_i)$ . Let  $k_i = [k_1^{(i)}, k_2^{(i)}]^\top$ . With this notation  $\mathbf{y}$  will be of the form

$$y = \begin{bmatrix} \begin{pmatrix} r_1 \cdot \mathbf{x} \\ r_2 \cdot \mathbf{x} \\ r_3 \cdot \mathbf{x} \end{pmatrix} \begin{pmatrix} k_1^{(0)} & k_2^{(0)} \end{pmatrix} \\ \begin{pmatrix} r_4 \cdot \mathbf{x} \\ r_5 \cdot \mathbf{x} \\ r_6 \cdot \mathbf{x} \end{pmatrix} \begin{pmatrix} k_1^{(1)} & k_2^{(1)} \end{pmatrix} \end{bmatrix}$$

In here  $r_m$ , stands for the  $m$ th row of the matrix  $\mathbf{D}$  where  $1 \leq m \leq 6$ . We can

rewrite  $\mathbf{y}$  as

$$\mathbf{y} = \begin{bmatrix} k_1^{(0)} r_1^\top \mathbf{x} & k_2^{(0)} r_1^\top \mathbf{x} \\ k_1^{(0)} r_2^\top \mathbf{x} & k_2^{(0)} r_2^\top \mathbf{x} \\ k_1^{(0)} r_3^\top \mathbf{x} & k_2^{(0)} r_3^\top \mathbf{x} \\ k_1^{(1)} r_4^\top \mathbf{x} & k_2^{(1)} r_4^\top \mathbf{x} \\ k_1^{(1)} r_5^\top \mathbf{x} & k_2^{(1)} r_5^\top \mathbf{x} \\ k_1^{(1)} r_6^\top \mathbf{x} & k_2^{(1)} r_6^\top \mathbf{x} \end{bmatrix}$$

Note that  $\mathbf{y}$  is a  $6 \times 2$  matrix. Then our objective is to find the derivative of the function  $\mathbf{y} : \mathbb{R}^6 \rightarrow \mathbb{R}^{12}$  with respect to  $\mathbf{x}$ . In other words we have

$$\mathbf{y} : (x_1, x_2, \dots, x_6) \mapsto (y_1, y_2, \dots, y_{12})$$

For consistencies sake we choose to (internally) encode  $\mathbf{y}$  in the following manner:

$$\mathbf{y} = \begin{bmatrix} y_1 & y_4 \\ y_2 & y_5 \\ y_3 & y_6 \\ y_7 & y_{10} \\ y_8 & y_{11} \\ y_9 & y_{12} \end{bmatrix}$$

We can now compute (the Jacobian),

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \begin{pmatrix} k_1^{(0)} r_1 \\ k_1^{(0)} r_2 \\ k_1^{(0)} r_3 \end{pmatrix} \\ \begin{pmatrix} k_2^{(0)} r_1 \\ k_2^{(0)} r_2 \\ k_3^{(0)} r_3 \end{pmatrix} \\ \begin{pmatrix} k_1^{(1)} r_4 \\ k_1^{(1)} r_5 \\ k_1^{(1)} r_6 \\ k_2^{(1)} r_4 \\ k_2^{(1)} r_5 \\ k_3^{(1)} r_6 \end{pmatrix} \end{bmatrix}$$

We can simplify this representation by using outer product notation as

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = [k_0 \otimes [\mathbf{D}]_0, k_1 \otimes [\mathbf{D}]_1]^\top.$$

This is the Jacobian we set out to compute. Note that the preceding expression is an  $12 \times 6$  matrix as desired. The general case now follows easily by induction on  $N$ .  $\square$

#### 6.4.1 Computing $\frac{\partial f_2}{\partial \mathbf{x}_{n+1}}$ :

For ease of notation (and ease of computer implementation) let us denote

$$[k_0 \otimes [\mathbf{D}]_0, k_1 \otimes [\mathbf{D}]_1, \dots, k_N \otimes [\mathbf{D}]_N]^\top =: \mathbf{K}_0 \otimes^* \mathbf{D}.$$

For  $n \neq 0$  we get,

$$\frac{\partial f_2}{\partial \mathbf{x}_{n+1}} = -\frac{1}{2}\sigma_1 \mathbf{K}_0 \otimes^* (\text{WED}_2) - \frac{1}{2}\mathbf{B}_{n+1} \otimes^* \mathbf{D}_1 - \frac{\nu_1}{2h} \mathbf{K}_0 \otimes^* (\text{ED}_2) \quad (6.15)$$

and for  $n = 0$ ,

$$\frac{\partial f_2}{\partial \mathbf{x}_{n+1}} = -\sigma_1 \mathbf{K}_0 \otimes^* (\text{WED}_2) - \mathbf{B}_{n+1} \otimes^* \mathbf{D}_1. \quad (6.16)$$

The computed derivative in (6.15) is an  $6(N+1) \times 3(N+1)$  matrix as expected.

### 6.4.2 Computing $\frac{\partial f_2}{\partial \mathcal{S}_{n+1}}$ .

The following identity is crucial.

**Proposition 21.** *For  $\tau_j$  as defined in (5.31) with  $1 \leq j \leq N$ , we have*

$$\tau_j = \frac{1}{2\eta} \text{trace}(\mathbf{s}_j(\mathbf{s}_{j-1}\mathbf{j})^\top) \quad (6.17)$$

*Proof.* Throughout the following set of equalities we use the fact that  $\mathbf{j}^\top = -\mathbf{j}$  and  $\mathbf{j}^2 = -\mathbf{i}_2$  without explicit mention. From (5.31) we have,

$$\begin{aligned} \tau_j \mathbf{j}\mathbf{j}^\top &= \frac{1}{2\eta} (\mathbf{s}_{j-1}^\top \mathbf{s}_j \mathbf{j}^\top - \mathbf{s}_j^\top \mathbf{s}_{j-1} \mathbf{j}^\top) \\ \text{trace}(\tau_j \mathbf{j}\mathbf{j}^\top) &= \frac{1}{2\eta} \text{trace}(\mathbf{s}_{j-1}^\top \mathbf{s}_j \mathbf{j}^\top - \mathbf{s}_j^\top \mathbf{s}_{j-1} \mathbf{j}^\top) \\ 2\tau_j &= \frac{1}{2\eta} (\text{trace}(\mathbf{s}_{j-1}^\top \mathbf{s}_j \mathbf{j}^\top) - \text{trace}(\mathbf{s}_j^\top \mathbf{s}_{j-1} \mathbf{j}^\top)) \\ \tau_j &= \frac{1}{4\eta} (\text{trace}(\mathbf{s}_j(\mathbf{s}_{j-1}\mathbf{j})^\top) + \text{trace}(\mathbf{s}_j^\top \mathbf{s}_{j-1}\mathbf{j})) \end{aligned} \quad (6.18)$$

$$\tau_j = \frac{1}{2\eta} \text{trace}(\mathbf{s}_j(\mathbf{s}_{j-1}\mathbf{j})^\top). \quad (6.19)$$

On (6.19) we used the fact that  $\text{trace}(XY^\top) = \text{trace}(X^\top Y)$  for matrices  $X$  and  $Y$ .

On (6.18) we used the cyclic property of the trace operator.  $\square$



**Theorem 22.** Let  $\mathbf{f}_j(\tau, \mathbf{s})$  be as defined in (5.34). Then

$$\frac{\partial \mathbf{f}_j(\tau, \mathbf{s})}{\partial \mathbf{s}_j} = \frac{1}{2\eta^2} (\text{vec}(\mathbf{s}_{j+1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j+1}\mathbf{j}) + \text{vec}(\mathbf{s}_{j-1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j-1}\mathbf{j})) \quad (6.20)$$

for  $1 \leq j \leq N-1$ . For the boundary cases where  $j = 0, N$  we have

$$\frac{\partial \mathbf{f}_j(\tau, \mathbf{s})}{\partial \mathbf{s}_j} = \frac{1}{2\eta^2} \begin{cases} (\text{vec}(\mathbf{s}_1\mathbf{j}) \otimes \text{vec}(\mathbf{s}_1\mathbf{j})), & \text{for } j = 0 \\ (\text{vec}(\mathbf{s}_{N-1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{N-1}\mathbf{j})), & \text{for } j = N. \end{cases} \quad (6.21)$$

In here  $\text{vec}(X)$  stands for the  $mn \times 1$  column vector obtained by stacking the columns of the  $m \times n$  matrix  $X$  on top of each other (from left to right). Thus, the equations in (6.20) and (6.21) are all  $6 \times 6$  matrices as expected.

*Proof.* Let us first consider the case  $1 \leq j \leq N-1$ . Using (6.17) we get

$$\tau_j = \frac{1}{2\eta} \text{trace}(\mathbf{s}_j \mathbf{j}^\top \mathbf{s}_{j-1}^\top) = -\frac{1}{2\eta} \text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j-1}^\top)$$

and

$$\tau_{j+1} = \frac{1}{2\eta} \text{trace}(\mathbf{s}_{j+1}(\mathbf{s}_j \mathbf{j})^\top) = \frac{1}{2\eta} \text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j+1}^\top)$$

Now using the definition of  $\mathbf{f}_j(\tau, \mathbf{s})$  in (5.34) we get

$$\mathbf{f}_j(\tau, \mathbf{s})\mathbf{j} = -\frac{1}{2\eta^2} [\text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j+1}^\top) \mathbf{s}_{j+1}\mathbf{j} + \text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j-1}^\top) \mathbf{s}_{j-1}\mathbf{j}] \quad (6.22)$$

Let us consider the term  $Y_1 := \text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j+1}^\top) \mathbf{s}_{j+1}\mathbf{j}$ . Note that  $Y$  is a  $3 \times 2$  matrix. We can view this as a function of the form  $Y_1 : \mathbb{R}^6 \rightarrow \mathbb{R}^6$  if we vectorize  $\text{vec}(Y_1) = \text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j+1}^\top) \text{vec}(\mathbf{s}_{j+1}\mathbf{j})$ . We can use the identity  $\text{trace}(X^\top Y) = \text{vec}(Y)^\top \text{vec}(X)$  and the fact that  $\text{trace}(XY) = \text{trace}(YX)$  to write,

$$\text{trace}(\mathbf{s}_j \mathbf{j} \mathbf{s}_{j+1}^\top) = \text{trace}(\mathbf{s}_j (\mathbf{s}_{j+1}\mathbf{j}^\top)^\top) = \text{vec}(\mathbf{s}_{j+1}\mathbf{j}^\top)^\top \text{vec}(\mathbf{s}_j)$$

and since we can factor out scalars from  $\text{vec}$  operator,

$$\text{vec}(Y_1) = \text{vec}(\mathbf{s}_{j+1}\mathbf{j}^\top)^\top \text{vec}(\mathbf{s}_j) \text{vec}(\mathbf{s}_{j+1}\mathbf{j}).$$

Therefore

$$\frac{\partial \text{vec}(Y_1)}{\partial \text{vec}(\mathbf{s}_j)} = \text{vec}(\mathbf{s}_{j+1}\mathbf{j}) \text{vec}(\mathbf{s}_{j+1}\mathbf{j}^\top)^\top.$$

But note that  $\mathbf{j}^\top = -\mathbf{j}$  and so

$$\frac{\partial \text{vec}(Y_1)}{\partial \text{vec}(\mathbf{s}_j)} = -\text{vec}(\mathbf{s}_{j+1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j+1}\mathbf{j}).$$

Similarly if  $Y_2 := \text{trace}(\mathbf{s}_j\mathbf{s}_{j-1}^\top)\mathbf{s}_{j-1}\mathbf{j}$  we can then show that,

$$\frac{\partial \text{vec}(Y_2)}{\partial \text{vec}(\mathbf{s}_j)} = -\text{vec}(\mathbf{s}_{j-1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j-1}\mathbf{j})$$

(6.20) now follows. To compute the derivatives of  $\mathbf{f}_0$  and  $\mathbf{f}_N$  note that we have set  $\tau_0 = \tau_{N+1} = 0$  as boundary conditions. It is now natural to define

$$\mathbf{f}_0(\tau, \mathbf{s}) = -\frac{1}{\eta}(\tau_1\mathbf{s}_1)$$

and

$$\mathbf{f}_N(\tau, \mathbf{s}) = -\frac{1}{\eta}(-\tau_N\mathbf{s}_{N-1})$$

We can now proceed similarly as we did before by noting that

$$\tau_1 = -\frac{1}{2\eta} \text{trace}(\mathbf{s}_1\mathbf{j}\mathbf{s}_0^\top),$$

and

$$\tau_N = -\frac{1}{2\eta} \text{trace}(\mathbf{s}_N\mathbf{j}\mathbf{s}_{N-1}^\top).$$

Thus,

$$\frac{\partial f_0(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_0} = \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_1\mathbf{j}) \otimes \text{vec}(\mathbf{s}_1\mathbf{j})]$$

Similarly,

$$\frac{\partial f_N(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_N} = \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_{N-1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{N-1}\mathbf{j})].$$

□

**Theorem 23.** Let  $f_j(\tau, \mathbf{s})$  be as defined in (5.34). Then,

$$\frac{\partial f_j(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j+1}} = -\frac{1}{2\eta^2} [\text{trace}(\mathbf{s}_j\mathbf{j}\mathbf{s}_{j+1}^\top)(\mathbf{j}^\top \otimes \mathbf{i}_3) + \text{vec}(\mathbf{s}_j\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j+1}\mathbf{j})] \quad (6.23)$$

and

$$\frac{\partial f_j(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j-1}} = -\frac{1}{2\eta^2} [\text{trace}(\mathbf{s}_j\mathbf{j}\mathbf{s}_{j-1}^\top)(\mathbf{j}^\top \otimes \mathbf{i}_3) + \text{vec}(\mathbf{s}_j\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j-1}\mathbf{j})] \quad (6.24)$$

for  $1 \leq j \leq N-1$ . For  $j=0$  and  $j=N$  we have,

$$\frac{\partial f_0(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_1} = -\frac{1}{2\eta^2} [\text{trace}(\mathbf{s}_0\mathbf{j}\mathbf{s}_1^\top)(\mathbf{j}^\top \otimes \mathbf{i}_3) + \text{vec}(\mathbf{s}_0\mathbf{j}) \otimes \text{vec}(\mathbf{s}_1\mathbf{j})] \quad (6.25)$$

and

$$\frac{\partial f_N(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{N-1}} = -\frac{1}{2\eta^2} [\text{trace}(\mathbf{s}_N\mathbf{j}\mathbf{s}_{N-1}^\top)(\mathbf{j}^\top \otimes \mathbf{i}_3) + \text{vec}(\mathbf{s}_N\mathbf{j}) \otimes \text{vec}(\mathbf{s}_{N-1}\mathbf{j})] \quad (6.26)$$

*Proof.* Let us first consider derivative of  $f_j$  with respect to  $\mathbf{s}_{j+1}$ . From (6.22) (and product rule) we get,

$$\frac{\partial f_j(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j+1}} = -\frac{1}{2\eta^2} \left[ \text{trace}(\mathbf{s}_j\mathbf{j}\mathbf{s}_{j+1}^\top) \frac{\partial(\mathbf{s}_{j+1}\mathbf{j})}{\partial \mathbf{s}_{j+1}} + \frac{\partial(\text{trace}(\mathbf{s}_j\mathbf{j}\mathbf{s}_{j+1}^\top))}{\partial \mathbf{s}_{j+1}} (\mathbf{s}_{j+1}\mathbf{j}) \right] \quad (6.27)$$

$$=: -\frac{1}{2\eta^2} [E_1 + E_2]. \quad (6.28)$$

Using a bit of matrix calculus (see for example [31] chapter 4),

$$E_1 = \text{trace}(\mathbf{s}_j \mathbf{s}_{j+1}^\top) (\mathbf{j}^\top \otimes \mathbf{i}_3).$$

To compute  $E_2$  we may proceed using vectorization as we did in Theorem (22) and we get

$$E_2 = \text{vec}(\mathbf{s}_j \mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j+1} \mathbf{j}).$$

(6.23) now follows. The proofs of (6.24), (6.25) and (6.26) are similar.  $\square$

The theorems (22) and (23) shows how to compute derivatives of terms that involve  $\mathbf{f}_j(\tau, \mathbf{s}) \mathbf{j}$ . We will now explicitly show how these computations can be used to find the derivatives of the discretized  $\mathbf{f}^{(d)}(\tau, \mathbf{s})$ .

**Corollary 24.** *Let  $\mathbf{f}_j^{(d)}(\tau, \mathbf{s})$  as discretized in (5.130). Then,*

**Case I:  $n \neq 0$**

$$\begin{aligned} \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_j} &= \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_{j+1} \mathbf{j}) + \text{vec}(\mathbf{s}_{j+1} [n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_{j+1} \mathbf{j}) \\ &\quad + (\text{vec}(\mathbf{s}_{j-1} \mathbf{j}) + \text{vec}(\mathbf{s}_{j-1} [n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_{j-1} \mathbf{j})] \end{aligned} \quad (6.29)$$

for  $1 \leq j \leq N-1$ . For  $j=0$  and  $j=N$  we have,

$$\frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_0} = \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_1 \mathbf{j}) + \text{vec}(\mathbf{s}_1 [n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_1 \mathbf{j})] \quad (6.30)$$

and

$$\frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_N} = \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_{N-1} \mathbf{j}) + \text{vec}(\mathbf{s}_{N-1} [n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_{N-1} \mathbf{j})] \quad (6.31)$$

**Case II:  $n = 0$**

$$\frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_j} = \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_{j+1} \mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j+1} \mathbf{j}) + \text{vec}(\mathbf{s}_{j-1} \mathbf{j}) \otimes \text{vec}(\mathbf{s}_{j-1} \mathbf{j})] \quad (6.32)$$

for  $1 \leq j \leq N - 1$ . For  $j = 0$  and  $j = N$  we have,

$$\frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_0} = \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_1 \mathbf{j}) \otimes \text{vec}(\mathbf{s}_1 \mathbf{j})] \quad (6.33)$$

and

$$\frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_N} = \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_{N-1} \mathbf{j}) \otimes \text{vec}(\mathbf{s}_{N-1} \mathbf{j})]. \quad (6.34)$$

**Corollary 25.** Let  $\mathbf{f}_j^{(d)}(\tau, \mathbf{s})$  as discretized in (5.130). Then,

**Case I:  $n \neq 0$**

$$\begin{aligned} \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_{j+1}} &= -\frac{1}{4\eta} [(\tau_{j+1} + \tau_{j+1}[n-1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_{j+1} \mathbf{j}) + \text{vec}(\mathbf{s}_{j+1}[n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_j \mathbf{j})] \end{aligned} \quad (6.35)$$

and

$$\begin{aligned} \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_{j-1}} &= \frac{1}{4\eta} [(\tau_j + \tau_j[n-1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_{j-1} \mathbf{j}) + \text{vec}(\mathbf{s}_{j-1}[n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_j \mathbf{j})] \end{aligned} \quad (6.36)$$

for  $1 \leq j \leq N - 1$ . For  $j = 0$  and  $j = N$  we have,

$$\begin{aligned} \frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_1} &= -\frac{1}{4\eta} [(\tau_1 + \tau_1[n-1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_1 \mathbf{j}) + \text{vec}(\mathbf{s}_1[n-1] \mathbf{j})) \otimes \text{vec}(\mathbf{s}_0 \mathbf{j})] \end{aligned} \quad (6.37)$$

and

$$\begin{aligned} \frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{N-1}} &= \frac{1}{4\eta} [(\tau_N + \tau_N[n-1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2} [(\text{vec}(\mathbf{s}_{N-1}\mathbf{j}) + \text{vec}(\mathbf{s}_{N-1}[n-1]\mathbf{j})) \otimes \text{vec}(\mathbf{s}_N\mathbf{j})] \end{aligned} \quad (6.38)$$

**Case II:  $n = 0$**

$$\frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j+1}} = -\frac{1}{\eta} [\tau_{j+1}(\mathbf{j}^\top \otimes \mathbf{i}_3)] - \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_{j+1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_j\mathbf{j})] \quad (6.39)$$

and

$$\frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j-1}} = \frac{1}{\eta} [\tau_j(\mathbf{j}^\top \otimes \mathbf{i}_3)] - \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_{j-1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_j\mathbf{j})] \quad (6.40)$$

for  $1 \leq j \leq N-1$ . For  $j = 0$  and  $j = N$  we have,

$$\frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_1} = -\frac{1}{\eta} [\tau_1(\mathbf{j}^\top \otimes \mathbf{i}_3)] - \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_1\mathbf{j}) \otimes \text{vec}(\mathbf{s}_0\mathbf{j})] \quad (6.41)$$

and

$$\frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{N-1}} = \frac{1}{\eta} [\tau_N(\mathbf{j}^\top \otimes \mathbf{i}_3)] - \frac{1}{2\eta^2} [\text{vec}(\mathbf{s}_{N-1}\mathbf{j}) \otimes \text{vec}(\mathbf{s}_N\mathbf{j})]. \quad (6.42)$$

We also need to compute derivatives of terms of the form  $\mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}$ . We now present that. Note that  $\mathbf{f}_j(\dot{\tau}, \mathbf{s})\mathbf{j}$  differs from  $\mathbf{f}_j^{(d)}(\tau, \mathbf{s})\mathbf{j}$  only slightly. Therefore slight adjustments to Corollaries (24) and (25) gives us the desired derivatives. Indeed we have,

**Corollary 26.** *Let  $\mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})$  be as discretized in (5.136). Then,*

Case I:  $n \neq 0$

$$\frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_j} = \frac{1}{h} \left( \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_j} \right) \quad (6.43)$$

for  $1 \leq j \leq N - 1$ . For  $j = 0$  and  $j = N$  we have,

$$\frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_0} = \frac{1}{h} \left( \frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_0} \right) \quad (6.44)$$

and

$$\frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_N} = \frac{1}{h} \left( \frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_N} \right) \quad (6.45)$$

Case II:  $n = 0$

$$\frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_j} = \frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_0} = \frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_N} = \mathbf{0}_{6 \times 6} \quad (6.46)$$

**Corollary 27.** Let  $\mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})$  be as discretized in (5.136). Then,

Case I:  $n \neq 0$

$$\begin{aligned} \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_{j+1}} &= -\frac{1}{4\eta h} [(\tau_{j+1} - \tau_{j+1}[n - 1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2 h} [(\text{vec}(\mathbf{s}_{j+1}\mathbf{j}) + \text{vec}(\mathbf{s}_{j+1}[n - 1]\mathbf{j})) \otimes \text{vec}(\mathbf{s}_j\mathbf{j})] \end{aligned} \quad (6.47)$$

and

$$\begin{aligned} \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_{j-1}} &= \frac{1}{4\eta h} [(\tau_j - \tau_j[n - 1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2 h} [(\text{vec}(\mathbf{s}_{j-1}\mathbf{j}) + \text{vec}(\mathbf{s}_{j-1}[n - 1]\mathbf{j})) \otimes \text{vec}(\mathbf{s}_j\mathbf{j})] \end{aligned} \quad (6.48)$$

for  $1 \leq j \leq N - 1$ . For  $j = 0$  and  $j = N$  we have,

$$\begin{aligned} \frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_1} &= -\frac{1}{4\eta h} [(\tau_1 - \tau_1[n-1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2 h} [(\text{vec}(\mathbf{s}_1\mathbf{j}) + \text{vec}(\mathbf{s}_1[n-1]\mathbf{j})) \otimes \text{vec}(\mathbf{s}_0\mathbf{j})] \end{aligned} \quad (6.49)$$

and

$$\begin{aligned} \frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{N-1}} &= \frac{1}{4\eta h} [(\tau_N - \tau_N[n-1])(\mathbf{j}^\top \otimes \mathbf{i}_3)] \\ &\quad - \frac{1}{8\eta^2 h} [(\text{vec}(\mathbf{s}_{N-1}\mathbf{j}) + \text{vec}(\mathbf{s}_{N-1}[n-1]\mathbf{j})) \otimes \text{vec}(\mathbf{s}_N\mathbf{j})] \end{aligned} \quad (6.50)$$

**Case II:  $n = 0$**

$$\frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j+1}} = \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{j-1}} = \frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_1} = \frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})\mathbf{j}}{\partial \mathbf{s}_{N-1}} = \mathbf{0}_{6 \times 6} \quad (6.51)$$

**Remark.** In the preceding corollaries we used a slight abuse of notation for the sake of simplicity. For each  $0 \leq j \leq N$  we used  $\tau_j$  instead of  $\tau_j[n+1]$  and  $\mathbf{s}_j$  instead of  $\mathbf{s}_j[n+1]$ . The absence of a square parenthetical subscript means that the time step at that point is  $n+1$ . When the time step at that point is  $n-1$  we have explicitly used that.

Also note that if  $\mathbf{y} = \mathbf{s}\mathbf{a}$  where  $\mathbf{s}$  is a  $3 \times 2$  matrix and  $\mathbf{a}$  is an  $2 \times 2$  symmetric matrix then

$$\frac{\partial \mathbf{y}}{\partial \mathbf{s}} = \mathbf{a} \otimes \mathbf{i}_3 \quad (6.52)$$

**Notation.** Let us denote the  $j$ th component of the function  $f_2$  by  $f_2[j]$  for  $0 \leq j \leq N$ .



Then we write

$$f_2[j] = \sigma_2 \mathbf{f}_j^{(d)}(\tau, \mathbf{s}) \mathbf{j} - \frac{1}{2}(\mathbf{s}_j[n+1] + \mathbf{s}_j[n-1]) \mathbf{a}_j + \frac{\nu_1}{2h}(\mathbf{s}_j[n+1] - \mathbf{s}_j[n-1]) \boldsymbol{\kappa}_j \boldsymbol{\kappa}_j^\top + \frac{1}{2} \nu_2 \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s}) \mathbf{j} \quad (6.53)$$

for  $1 \leq j \leq N-1$ . For  $j=0$  and  $j=N$  we write,

$$f_2[0] = \sigma_2 \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j} - \frac{1}{2}(\mathbf{s}_0[n+1] + \mathbf{s}_0[n-1]) \mathbf{a}_0 + \frac{1}{2} \nu_2 \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s}) \mathbf{j} \quad (6.54)$$

and

$$f_2[N] = \sigma_2 \mathbf{f}_N^{(d)}(\tau, \mathbf{s}) \mathbf{j} - \frac{1}{2}(\mathbf{s}_N[n+1] + \mathbf{s}_N[n-1]) \mathbf{a}_N + \frac{1}{2} \nu_2 \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s}) \mathbf{j} \quad (6.55)$$

Note here that we have dropped all of the terms which do not depend on  $\mathbf{s}_j$  since those vanish after taking derivatives with respect to  $\mathbf{s}_j$  anyway.

We are now in a position to compute  $\frac{\partial f_2}{\partial \mathbf{s}_{n+1}}$ . Notice that  $\frac{\partial f_2}{\partial \mathbf{s}_{n+1}}$  is a  $6(N+1) \times 6(N+1)$  matrix. Let us call this matrix  $\mathbf{M}$ . Let us denote the  $(j, k)$ th block of  $\mathbf{M}$  with  $\mathbf{M}_{j,k}$  for  $0 \leq j, k \leq N$ . Note that each  $\mathbf{M}_{j,k}$  is a  $6 \times 6$  matrix.

#### 6.4.2.1 The zeroth row of $\mathbf{M}$ .

$$\text{for } n \neq 0: \quad \mathbf{M}_{0,0} = \frac{\partial f_2[0]}{\partial \mathbf{s}_0} = \sigma_2 \frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_0} - \frac{1}{2}(\mathbf{a}_0 \otimes \mathbf{i}_3) + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_0} \quad (6.56)$$

$$\text{for } n = 0: \quad \mathbf{M}_{0,0} = \sigma_2 \frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_0} - (\mathbf{a}_0 \otimes \mathbf{i}_3) + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_0} \quad (6.57)$$

$$\mathbf{M}_{0,1} = \frac{\partial f_2[0]}{\partial \mathbf{s}_1} = \sigma_2 \frac{\partial \mathbf{f}_0^{(d)}(\tau, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_1} + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_0^{(d)}(\dot{\tau}, \mathbf{s}) \mathbf{j}}{\partial \mathbf{s}_1} \quad (6.58)$$

$$\mathbf{M}_{0,k} = \frac{\partial f_2[0]}{\partial \mathbf{s}_k} = \mathbf{0}_{6 \times 6}, \quad \text{for } 2 \leq k \leq N. \quad (6.59)$$

### 6.4.2.2 The $j$ th row of $\mathbf{M}$ for $1 \leq j \leq N - 1$ .

$$\mathbf{M}_{j,j-1} = \frac{\partial f_2[j]}{\partial \mathbf{s}_{j-1}} = \sigma_2 \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_{j-1}} + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_{j-1}} \quad (6.60)$$

$$\begin{aligned} \text{for } n \neq 0 : \quad \mathbf{M}_{j,j} &= \frac{\partial f_2[j]}{\partial \mathbf{s}_j} \\ &= \sigma_2 \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_j} - \frac{1}{2} (\mathbf{a}_j \otimes \mathbf{i}_3) + \frac{\nu_1}{2h} ((\boldsymbol{\kappa}_j \boldsymbol{\kappa}_j^\top) \otimes \mathbf{i}_3) + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_j} \end{aligned} \quad (6.61)$$

$$\text{for } n = 0 : \quad \mathbf{M}_{j,j} = \frac{\partial f_2[j]}{\partial \mathbf{s}_j} = \sigma_2 \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_j} - (\mathbf{a}_j \otimes \mathbf{i}_3) + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_j} \quad (6.62)$$

$$\mathbf{M}_{j,j+1} = \frac{\partial f_2[j]}{\partial \mathbf{s}_{j+1}} = \sigma_2 \frac{\partial \mathbf{f}_j^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_{j+1}} + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_j^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_{j+1}} \quad (6.63)$$

$$\mathbf{M}_{j,k} = \mathbf{0}_{6 \times 6} \quad \text{for } k \neq j - 1, j, j + 1. \quad (6.64)$$

### 6.4.2.3 The $N$ th row of $\mathbf{M}$

$$\mathbf{M}_{N,k} = \mathbf{0}_{6 \times 6}, \quad \text{for } 0 \leq k \leq N - 2. \quad (6.65)$$

$$\mathbf{M}_{N,N-1} = \frac{\partial f_2[N]}{\partial \mathbf{s}_{N-1}} = \sigma_2 \frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_{N-1}} + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_{N-1}} \quad (6.66)$$

$$\text{for } n \neq 0 : \quad \mathbf{M}_{N,N} = \frac{\partial f_2[N]}{\partial \mathbf{s}_N} = \sigma_2 \frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_N} - \frac{1}{2} (\mathbf{a}_N \otimes \mathbf{i}_3) + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_N} \quad (6.67)$$

$$\text{for } n = 0 : \quad \mathbf{M}_{N,N} = \frac{\partial f_2[N]}{\partial \mathbf{s}_N} = \sigma_2 \frac{\partial \mathbf{f}_N^{(d)}(\tau, \mathbf{s})_j}{\partial \mathbf{s}_N} - (\mathbf{a}_N \otimes \mathbf{i}_3) + \frac{1}{2} \nu_2 \frac{\partial \mathbf{f}_N^{(d)}(\dot{\tau}, \mathbf{s})_j}{\partial \mathbf{s}_N} \quad (6.68)$$

**Remark.** *It is worth noting here that we have explicitly written formulas for the cases  $n \neq 0$  and  $n = 0$  only when the equations are different. Whenever there is no explicit distinction care should be taken in looking up the appropriate equation.*

It now follows that the matrix  $\mathbf{M}$  is an  $6(N + 1) \times 6(N + 1)$  block tridiagonal matrix which has the following form.

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{0,0} & \mathbf{M}_{0,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{M}_{1,0} & \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_{N-2,N-3} & \mathbf{M}_{N-2,N-2} & \mathbf{M}_{N-2,N-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{M}_{N-1,N-2} & \mathbf{M}_{N-1,N-1} & \mathbf{M}_{N-1,N} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{N,N-1} & \mathbf{M}_{N,N} \end{pmatrix}$$

### 6.4.3 Computing $\frac{\partial f_2}{\partial \mathbf{A}_{n+1}}$

We can effectively set

$$f_2[j] = -\frac{1}{2}(\mathbf{s}_j[n+1] + \mathbf{s}_j[n-1])\mathbf{a}_j$$

since this is the only term that depend on  $\mathbf{a}_j$ .

**Proposition 28.** *Let  $\mathbf{y} = \mathbf{t}\mathbf{a}$  where  $\mathbf{t} = (t_{ij})$  is a  $3 \times 2$  matrix and  $\mathbf{a}$  is a  $2 \times 2$  symmetric matrix. Then*

$$\frac{\partial \mathbf{y}}{\partial \mathbf{a}} = \begin{pmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 0 \\ 0 & t_{11} & t_{12} \\ 0 & t_{21} & t_{22} \\ 0 & t_{31} & t_{32} \end{pmatrix}_{6 \times 3} \quad (6.69)$$

Moreover the expression in (6.69) can be written as

$$\frac{\partial \mathbf{y}}{\partial \mathbf{a}} = (\mathbf{i}_2 \otimes \mathbf{t}) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} =: (\mathbf{i}_2 \otimes \mathbf{t}) \mathbf{k}$$

Let us denote

$$\text{for } n \neq 0 : \quad \mathbf{t}^{(j)} = -\frac{1}{2}(\mathbf{s}_j[n+1] + \mathbf{s}_j[n-1])$$

$$\text{for } n = 0 : \quad \mathbf{t}^{(j)} = -\mathbf{s}_j[n+1]$$

with suppressed time dependence. Then  $\mathbf{t}^{(j)}$  is a  $3 \times 2$  matrix. It now follows that

$$p_j =: \frac{\partial f_2[j]}{\partial \mathbf{a}_j} = (\mathbf{i}_2 \otimes \mathbf{t}^{(j)}) \mathbf{k}$$

for  $0 \leq j \leq N$ . It is now easy to see that

$$\frac{\partial f_2}{\partial \mathbf{A}_{n+1}} = \text{diag}(p_0, p_1, \dots, p_N) \tag{6.70}$$

The matrix in (6.70) is a  $6(N+1) \times 3(N+1)$  block diagonal matrix as expected.

#### 6.4.4 Computing $\frac{\partial f_2}{\partial \mathbf{B}_{n+1}}$

We may write  $f_2 = -\frac{1}{2} \mathbf{D}_1(\mathbf{x}_{n+1} + \mathbf{x}_{n-1}) \otimes^* \boldsymbol{\beta}$ .

**Proposition 29.** *Let  $\mathbf{y} = \mathbf{x} \otimes \boldsymbol{\beta}$  where  $\mathbf{x}$  is a  $3 \times 1$  vector and  $\boldsymbol{\beta}$  is a  $2 \times 1$  vector.*

*Then*

$$\frac{\partial \mathbf{y}}{\partial \boldsymbol{\beta}} = \mathbf{i}_2 \otimes \mathbf{x}$$

It is now easy to see that

$$\frac{\partial f_2}{\partial \mathbf{B}_{n+1}} = \text{diag}(q_0, q_1, \dots, q_N) \quad (6.71)$$

where

$$\text{for } n \neq 0: \quad q_j = \mathbf{i}_2 \otimes \left(-\frac{1}{2}[\mathbf{D}_1]_j(\mathbf{x}_{n+1} + \mathbf{x}_{n-1})\right) \quad (6.72)$$

$$\text{for } n = 0: \quad q_j = \mathbf{i}_2 \otimes \left(-[\mathbf{D}_1]_j \mathbf{x}_{n+1}\right) \quad (6.73)$$

In here we are using the notation  $[\mathbf{D}]_j$  as introduced in Section 6.4. Note that each  $q_j$  is an  $6 \times 2$  matrix and consequently the equation in (6.71) is a  $6(N+1) \times 2(N+1)$  matrix.

#### 6.4.5 Computing $\frac{\partial f_2}{\partial \alpha}$

$f_2$  has no  $\alpha$  dependence and thus,

$$\frac{\partial f_2}{\partial \alpha} = \mathbf{0}_{6(N+1) \times (N+1)}. \quad (6.74)$$

### 6.5 Derivatives of constraint equations

We now concentrate on finding the derivatives of the three constraint equations. Note that we are not discretising the constraints.

$$f_3[j] := (\mathbf{s}_j[n+1])^\top \mathbf{s}_j[n+1] - \mathbf{i}_2 = \mathbf{0}_{2 \times 2} \quad (6.75)$$

$$f_4[j] := (\mathbf{s}_j[n+1])^\top [\mathbf{D}_1]_j \mathbf{x}_{n+1} = \mathbf{0}_{2 \times 1} \quad (6.76)$$

$$f_5[j] := ([\mathbf{D}_1]_j \mathbf{x}_{n+1} \cdot [\mathbf{D}_1]_j \mathbf{x}_{n+1}) - 1 = 0 \quad (6.77)$$

for  $0 \leq j \leq N$ .

## 6.6 Derivatives of $f_3$

Some preliminaries first.

**Definition 5** (Commutation Matrix). *Let  $A$  be an  $m \times n$  matrix. Then  $K_{m,n}$  is the  $mn \times mn$  matrix which transforms  $\text{vec}(A)$  into  $\text{vec}(A^\top)$ :*

$$K_{m,n} \text{vec}(A) = \text{vec}(A^\top).$$

*In here  $\text{vec}(A)$  is the  $mn \times 1$  column vector obtained by stacking the columns of  $A$  on top of each other.*

**Proposition 30.** *Let  $f(\mathbf{s}) = \mathbf{s}^\top \mathbf{s}$  where  $\mathbf{s}$  is a  $3 \times 2$  matrix. Then,*

$$\frac{\partial f}{\partial \mathbf{s}} = \mathbf{i}_2 \otimes \mathbf{s}^\top + K_{2,2}(\mathbf{i}_2 \otimes \mathbf{s}^\top) \quad (6.78)$$

For a more detailed discussion of *commutation matrix* and matrix calculus in general see Horn and Johnson [31][p. 267] and/or Magnus and Neudecker [45][pp. 205-207].

### 6.6.1 Derivative $\frac{\partial f_3}{\partial \mathbf{x}_{n+1}}$

The function  $f_3$  has no  $\mathbf{x}$  dependence and therefore

$$\frac{\partial f_3}{\partial \mathbf{x}_{n+1}} = \mathbf{0}_{3(N+1) \times 3(N+1)} \quad (6.79)$$

Note here that it is still important to get the dimension of  $\mathbf{0}$  since we need this for the actual computer implementation.

### 6.6.2 Derivative $\frac{\partial f_3}{\partial \mathcal{S}_{n+1}}$

In light of Proposition (30) we can write

$$r_j := \frac{\partial f_3[j]}{\partial \mathbf{s}_j} = (\mathbf{i}_2 \otimes \mathbf{s}_j^\top) + K_{2,2}(\mathbf{i}_2 \otimes \mathbf{s}_j^\top) = (\mathbf{i}_4 + K_{2,2})(\mathbf{i}_2 \otimes \mathbf{s}_j^\top) \quad (6.80)$$

The expression in (6.80) is a  $4 \times 6$  matrix. However the 2nd and 3rd rows of this matrix are identical since  $\mathbf{s}_j^\top \mathbf{s}_j$  is symmetric. We need to remove duplicate rows so that the Jacobian computation is accurate. This *row removal* can be easily by achieved pre-multiplying  $r_j$  with the following matrix:

$$\mathbf{m} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.81)$$

(Note this particular matrix,  $\mathbf{m}$  removes the third row.) It is now easy to see that

$$\frac{\partial f_3}{\partial \mathcal{S}_{n+1}} = \text{diag}(\mathbf{m}r_0, \mathbf{m}r_1, \dots, \mathbf{m}r_N) \quad (6.82)$$

The equation in (6.82) is now a  $3(N+1) \times 6(N+1)$  matrix as desired.

### 6.6.3 Derivatives $\frac{\partial f_3}{\partial \mathbf{A}}, \frac{\partial f_3}{\partial \mathbf{B}}, \frac{\partial f_3}{\partial \alpha}$

Once again  $f_3$  has no  $\mathbf{a}, \beta$  or  $\alpha$  dependence and therefore we have

$$\frac{\partial f_3}{\partial \mathbf{A}_{n+1}} = \mathbf{0}_{3(N+1) \times 3(N+1)} \quad (6.83)$$

$$\frac{\partial f_3}{\partial \mathbf{B}_{n+1}} = \mathbf{0}_{3(N+1) \times 2(N+1)} \quad (6.84)$$

$$\frac{\partial f_3}{\partial \alpha_{n+1}} = \mathbf{0}_{3(N+1) \times (N+1)} \quad (6.85)$$

## 6.7 Derivatives of $f_4$

### 6.7.1 Derivative $\frac{\partial f_4}{\partial \mathbf{x}_{n+1}}$

It is clear that

$$\frac{\partial f_4[j]}{\partial \mathbf{x}_{n+1}} = \mathbf{s}_j^\top [\mathbf{D}_1]_j \quad (6.86)$$

for  $0 \leq j \leq N$ . The expression in (6.86) yields  $2 \times 3(N+1)$  matrix. It now follows that

$$\frac{\partial f_4}{\partial \mathbf{x}_{n+1}} = [\mathbf{s}_0^\top [\mathbf{D}_1]_0, \dots, \mathbf{s}_N^\top [\mathbf{D}_1]_N]^\top. \quad (6.87)$$

Consequently the expression in (6.87) yields a  $2(N+1) \times 3(N+1)$  matrix as expected.

### 6.7.2 Derivative $\frac{\partial f_4}{\partial \mathcal{S}_{n+1}}$

**Proposition 31.** *Let  $f(\mathbf{s}) = \mathbf{s}^\top \mathbf{D} \mathbf{x}$  where  $\mathbf{s}$  is a  $3 \times 2$  matrix,  $\mathbf{D}$  is an  $3 \times (3(N+1))$  matrix and  $\mathbf{x}$  is a  $3(N+1) \times 1$  vector. Then,*

$$\frac{\partial f}{\partial \mathbf{s}} = ((\mathbf{D} \mathbf{x})^\top \otimes \mathbf{i}_2) K_{3,2}. \quad (6.88)$$

It is now easy to see that

$$u_j := \frac{\partial f_4[j]}{\partial \mathbf{s}_j} = (([\mathbf{D}_1]_j \mathbf{x}_{n+1})^\top \otimes \mathbf{i}_2) K_{3,2}. \quad (6.89)$$

The expression in (6.89) is an  $2 \times 6$  matrix. Finally,

$$\frac{\partial f_4}{\partial \mathcal{S}_{n+1}} = \text{diag}(u_0, u_1, \dots, u_N) \quad (6.90)$$

which results in a  $2(N+1) \times 6(N+1)$  matrix as expected.



### 6.7.3 Derivatives of $\frac{\partial f_4}{\partial \mathbf{A}}, \frac{\partial f_4}{\partial \mathbf{B}}, \frac{\partial f_4}{\partial \alpha}$ .

Clearly  $f_4$  does not depend on any of  $\mathbf{A}, \beta$  or  $\alpha$ . Thus,

$$\frac{\partial f_4}{\partial \mathbf{A}_{n+1}} = \mathbf{0}_{2(N+1) \times 3(N+1)} \quad (6.91)$$

$$\frac{\partial f_4}{\partial \mathbf{B}_{n+1}} = \mathbf{0}_{2(N+1) \times 2(N+1)} \quad (6.92)$$

$$\frac{\partial f_4}{\partial \alpha_{n+1}} = \mathbf{0}_{2(N+1) \times (N+1)} \quad (6.93)$$

## 6.8 Derivatives of $f_5$

We have that

$$v_j := \frac{\partial f_5[j]}{\partial \mathbf{x}_{n+1}} = 2\mathbf{x}_{n+1}^\top ([\mathbf{D}_1]_j^\top [\mathbf{D}_1]_j) \quad (6.94)$$

which results in a  $1 \times 3(N+1)$  row vector. It now easily follows that

$$\frac{\partial f_5}{\partial \mathbf{x}_{n+1}} = [v_0, v_1, \dots, v_N]^\top \quad (6.95)$$

where the expression in (6.95) results in a  $(N+1) \times 3(N+1)$  matrix as desired. We also have the following,

$$\frac{\partial f_5}{\partial \mathbf{S}_{n+1}} = \mathbf{0}_{(N+1) \times 6(N+1)} \quad (6.96)$$

$$\frac{\partial f_5}{\partial \mathbf{A}_{n+1}} = \mathbf{0}_{(N+1) \times 3(N+1)} \quad (6.97)$$

$$\frac{\partial f_5}{\partial \mathbf{B}_{n+1}} = \mathbf{0}_{(N+1) \times 2(N+1)} \quad (6.98)$$

$$\frac{\partial f_5}{\partial \alpha_{n+1}} = \mathbf{0}_{(N+1) \times (N+1)}. \quad (6.99)$$

## 6.9 The Jacobian

For use in the computer implementation we state here the full Jacobian of the system along with its block sizes. Let us denote the Jacobian by  $\mathbf{J}$ .

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial \mathbf{x}_{n+1}} & \frac{\partial f_1}{\partial \mathbf{S}_{n+1}} & \frac{\partial f_1}{\partial \mathbf{A}_{n+1}} & \frac{\partial f_1}{\partial \mathbf{B}_{n+1}} & \frac{\partial f_1}{\partial \boldsymbol{\alpha}_{n+1}} \\ \frac{\partial f_2}{\partial \mathbf{x}_{n+1}} & \frac{\partial f_2}{\partial \mathbf{S}_{n+1}} & \frac{\partial f_2}{\partial \mathbf{A}_{n+1}} & \frac{\partial f_2}{\partial \mathbf{B}_{n+1}} & \frac{\partial f_2}{\partial \boldsymbol{\alpha}_{n+1}} \\ \frac{\partial f_3}{\partial \mathbf{x}_{n+1}} & \frac{\partial f_3}{\partial \mathbf{S}_{n+1}} & \frac{\partial f_3}{\partial \mathbf{A}_{n+1}} & \frac{\partial f_3}{\partial \mathbf{B}_{n+1}} & \frac{\partial f_3}{\partial \boldsymbol{\alpha}_{n+1}} \\ \frac{\partial f_4}{\partial \mathbf{x}_{n+1}} & \frac{\partial f_4}{\partial \mathbf{S}_{n+1}} & \frac{\partial f_4}{\partial \mathbf{A}_{n+1}} & \frac{\partial f_4}{\partial \mathbf{B}_{n+1}} & \frac{\partial f_4}{\partial \boldsymbol{\alpha}_{n+1}} \\ \frac{\partial f_5}{\partial \mathbf{x}_{n+1}} & \frac{\partial f_5}{\partial \mathbf{S}_{n+1}} & \frac{\partial f_5}{\partial \mathbf{A}_{n+1}} & \frac{\partial f_5}{\partial \mathbf{B}_{n+1}} & \frac{\partial f_5}{\partial \boldsymbol{\alpha}_{n+1}} \end{pmatrix} \quad (6.100)$$

The dimensions of each of the blocks are:

- $\frac{\partial f_1}{\partial \mathbf{x}_{n+1}}$  is an  $3(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_1}{\partial \mathbf{S}_{n+1}}$  is an  $3(N + 1) \times 6(N + 1)$  matrix
- $\frac{\partial f_1}{\partial \mathbf{A}_{n+1}}$  is an  $3(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_1}{\partial \mathbf{B}_{n+1}}$  is an  $3(N + 1) \times 2(N + 1)$  matrix.
- $\frac{\partial f_1}{\partial \boldsymbol{\alpha}_{n+1}}$  is an  $3(N + 1) \times (N + 1)$  matrix.
- $\frac{\partial f_2}{\partial \mathbf{x}_{n+1}}$  is an  $6(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_2}{\partial \mathbf{S}_{n+1}}$  is an  $6(N + 1) \times 6(N + 1)$  matrix
- $\frac{\partial f_2}{\partial \mathbf{A}_{n+1}}$  is an  $6(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_2}{\partial \mathbf{B}_{n+1}}$  is an  $6(N + 1) \times 2(N + 1)$  matrix.

- $\frac{\partial f_2}{\partial \boldsymbol{\alpha}_{n+1}}$  is an  $6(N + 1) \times (N + 1)$  matrix.
- $\frac{\partial f_3}{\partial \mathbf{x}_{n+1}}$  is an  $3(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_3}{\partial \mathbf{S}_{n+1}}$  is an  $3(N + 1) \times 6(N + 1)$  matrix
- $\frac{\partial f_3}{\partial \mathbf{A}_{n+1}}$  is an  $3(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_3}{\partial \mathbf{B}_{n+1}}$  is an  $3(N + 1) \times 2(N + 1)$  matrix.
- $\frac{\partial f_3}{\partial \boldsymbol{\alpha}_{n+1}}$  is an  $3(N + 1) \times (N + 1)$  matrix.
- $\frac{\partial f_4}{\partial \mathbf{x}_{n+1}}$  is an  $2(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_4}{\partial \mathbf{S}_{n+1}}$  is an  $2(N + 1) \times 6(N + 1)$  matrix
- $\frac{\partial f_4}{\partial \mathbf{A}_{n+1}}$  is an  $2(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_4}{\partial \mathbf{B}_{n+1}}$  is an  $2(N + 1) \times 2(N + 1)$  matrix.
- $\frac{\partial f_4}{\partial \boldsymbol{\alpha}_{n+1}}$  is an  $2(N + 1) \times (N + 1)$  matrix.
- $\frac{\partial f_5}{\partial \mathbf{x}_{n+1}}$  is an  $(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_5}{\partial \mathbf{S}_{n+1}}$  is an  $(N + 1) \times 6(N + 1)$  matrix
- $\frac{\partial f_5}{\partial \mathbf{A}_{n+1}}$  is an  $(N + 1) \times 3(N + 1)$  matrix.
- $\frac{\partial f_5}{\partial \mathbf{B}_{n+1}}$  is an  $(N + 1) \times 2(N + 1)$  matrix.
- $\frac{\partial f_5}{\partial \boldsymbol{\alpha}_{n+1}}$  is an  $(N + 1) \times (N + 1)$  matrix.

Note that  $\mathbf{J}$  is a  $15(N + 1) \times 15(N + 1)$  matrix.

The complete software implementation of this problem in Python 3 can be found on <https://github.com/dilanfd/dynamics-of-springs>

# Bibliography

- [1] B. A. Agboola and D. A. Jack. *Comparison of short-fiber orientation and predicted materials properties from closure approximation and spherical harmonics approach*. ECTC Preceedings of the ASME Early Career Technical Conference, Oral Roberts University, 2010.
- [2] B. A. Agboola and D. A. Jack. *Comparison of Short-Fiber Orientation and Predicted Materials Properties from Closure Approximations and Spherical Harmonics*. Early Career Technical Conference, Tulsa, Oklahoma, 2010.
- [3] B. A. Agboola and D. A. Jack. Fiber Orientation and Stiffness Predictions for Thin Geometries Using the Fast Exact Closure. *Proceedings of the Society of Rheology's 82nd Annual Meeting, Santa Fe, NM*, October 2010.
- [4] B. A. Agboola, D. A. Jack, S. Montgomery-Smith, and D. E. Smith. Investigation of the Effectiveness and Efficiency of the Exact Closures: Comparison with Industrial Closures and Spherical Harmonic Solutions. *Proceedings of ASME IMECE'10, Vancouver, British Columbia, Canada*, November 2010.
- [5] Babatunde O Agboola, David A Jack, and Stephen Montgomery-Smith. Effectiveness of recent fiber-interaction diffusion models for orientation and the part stiffness predictions in injection molded short-fiber reinforced composites. *Composites Part A: Applied Science and Manufacturing*, 43(11):1959–1970, 2012.

- [6] C. J. Ancker, Jr. and J. N. Goodier. Pitch and curvature corrections for helical springs. *ASME JOURNAL OF APPLIED MECHANICS*, 25:466–495, 1958.
- [7] S. S. Antman. *Nonlinear problems in elasticity*. Springer, New York, 2nd edition, 2005.
- [8] N. H. Asmar. *Partial Differential Equations with Fourier Series and Boundary Value Problems*. Prentice Hall, 2nd edition, 2004.
- [9] O. A. Bauchau and C. L. Bottasso. On the design of energy preserving and decaying schemes, for flexible, nonlinear multi-body systems. *Computer Methods in Applied Mechanics and Engineering*, 169:61–79, 1999.
- [10] R. L. Burden and J. D. Faires. *Numerical methods*, volume 8. 8th Edition, Thomson Brooks/Cole Publishing Co, 2005.
- [11] R. E. Caflisch and J. H. Maddocks. Nonlinear dynamical theory of the elastica. *Proceedings of the Royal Society of Edinburgh*, 99:1–23, 1984.
- [12] E. Celledoni, V. Grimm, R. I. McLachlan, D. I. McLaren, D. O’Neale, B. Owren, and G. R. W. Quispel. Preserving energy resp. dissipation in numerical pdes using the ‘average vector field’ method. preprint, 2012.
- [13] W. H. Chen and P. Tsai. On static and dynamic finite element analysis of helical springs and experimental verifications. *Chung Kuo Kang Ch’eng Hsueh K’an*, 6:1, 1983.
- [14] B. D. Coleman and E. H. Dill. Flexure waves in elastic rods. *J. Acoust. Soc. Am*, 91:2663–2673, 1992.
- [15] Wikipedia contributors. Spring (device) — wikipedia, the free encyclopedia, 2018. [Online; accessed 16-January-2018].

- [16] G. A. Costello. Radial expansion of impacted helical springs. *ASME JOURNAL OF APPLIED MECHANICS*, 42:789–792, 1975.
- [17] D. J. Dichmann, J. H. Maddocks, and R. L. Pego. Hamiltonian dynamics of an elastica and the stability of solitary waves. *Arch. Rational Mech. Anal*, 135:357–396, 1996.
- [18] F. Diele and B. Pace. Energy-preserving runge-kutta methods, 2010.
- [19] W. T. Donkin and H. H. Clark. The electric telemeter and valve- spring surge. *Trans. SAE*, 24:185–196, 1929.
- [20] N. Dunford and J. T. Schwartz. *Linear operators. Part II. Spectral theory*. John Wiley & Sons Inc., New York, 1963.
- [21] R. S. Falk and J.-m. Xu. Convergence of a second-order scheme for the non-linear dynamical equations of elastic rods. *SIAM J. Numer. Anal*, 32(4):1185–1209, 1995.
- [22] E. Fermi, J. Pasta, and S. Ulam. Studies of nonlinear problems. *Document LA-*, 1940, 1955.
- [23] F. Frenet. Sur les courbes à double courbure. *Thèse, Toulouse, Abstract in J. de Math*, 17:1852, 1847.
- [24] T. Fütterer, A. Klar, and R. Wegener. An energy conserving numerical scheme for the dynamics of hyperelastic rods. *International Journal of Differential Equations*, 8308:15, 2012.
- [25] G Gallavotti, editor. *The Fermi–Pasta–Ulam Problem: A Status Report*. Springer, Lecture Notes in Physics 728, 2008.
- [26] V. G. A. Goss. *Snap buckling, writhing and loop formation in twisted rods*. PhD thesis, University College London, 2003.

- [27] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, Series in Computational Mathematics, 2nd edition, 2002.
- [28] Brian C. Hall. *Lie groups, Lie algebras, and representations: An Elementary Introduction*, volume 222. Springer, 2015.
- [29] Harold V. Henderson and S. R. Searle. The vec-permutation matrix, the vec operator and kronecker products: a review. *Linear and Multilinear Algebra*, 9(4):271–288, 1981.
- [30] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.
- [31] Roger A Horn and Charles R Johnson. *Topics in matrix analysis*. Cambridge University Press, 1994.
- [32] W. Huang. Inspection satellite orbit design using differential evolution, third prize winner in third mizzou-cis student poster competition, nov, 2008.
- [33] K. Ito. Uniform stabilization of the dynamic elastica by boundary feedback. *SIAM J. Control Optim.*, 37:319–329, 1998.
- [34] K. Ito. Stabilization of the dynamic elastica only by damping torque. *Nonlinear Analysis: Real World Applications*, 10(5):3122–3131, 2009.
- [35] D. A. Jack, S. Montgomery-Smith, and D. E. Smith. Anisotropic Diffusion Model for Suspensions of Short-Fiber Composite Processes. in. *The XVth International Congress on Rheology, the Society of Rheology 80th Annual Meeting*, August 2008.
- [36] G. B. Jeffery. The Motion of Ellipsoidal Particles Immersed in a Viscous Fluid. *Proceedings of the Royal Society of London A*, 102:161–179, March 1923.

- [37] F. Jehle and W. R. Spiller. Idiosyncrasies of valve mechanisms and their causes. *Trans. SAE*, 24:197–212, 1929.
- [38] Michio Kato. Study on dynamic properties of a varying-pitch helical compression spring. *Bulletin of the JSME*, 17(110):1015–1022, 1974.
- [39] E. Lehr. Schwingungen in ventiltfedern. *VDI*, 77(18):457–462, 1933.
- [40] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, 2004.
- [41] R. Levian. The elastica: A mathematical history. Technical report, University of California, Berkeley, 2008.
- [42] Y. Lin and A. P. Pisano. General dynamic equations of helical springs with static solution and experimental verification. *Transactions of the ASME*, 54:910–917, 1987.
- [43] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity*. Dover, New York, 4th edition, 1927.
- [44] J. H. Maddocks and D. J. Dichmann. Conservation laws in the dynamics of rods. *Journal of Elasticity*, 34:83–96, 1994.
- [45] Jan R Magnus and Heinz Neudecker. Matrix differential calculus with applications in statistics and econometrics. *Wiley series in probability and mathematical statistics*, 1988.
- [46] C. W. Misner, K. S. Torne, and J. A. Wheeler. Gravitation. *Freeman, San Francisco*, 1973, 1973.
- [47] S. Montgomery-Smith. Hamiltonians representing equations of motion with damping due to friction. *ArXiv e-prints*, May 2013.



- [48] S. Montgomery-Smith, D. A. Jack, and D. E. Smith. Fast Solutions for the Fiber Orientation of Concentrated Suspensions of Short-Fiber Composites Using the Exact Closure Methods. *Proceedings of ASME IMECE'10, Vancouver, British Columbia, Canada*, November 2010.
- [49] S. J. Montgomery-Smith. Perturbations of the coupled Jeffery-Stokes equations. *J. of Fluid Mechanics*, 681:622–638, 2011.
- [50] S. J. Montgomery-Smith, W. He, D. A. Jack, and D. E. Smith. Exact tensor closures for the three dimensional Jeffery’s Equation. *J. of Fluid Mechanics*, 680:321–335, 2011.
- [51] S. J. Montgomery-Smith and W. Huang. *Dynamic Equations of Non-Helical Springs*. in preparation, 2012.
- [52] S. J. Montgomery-Smith and W. Huang. A numerical method to model dynamic behavior of thin inextensible elastic rods in three dimensions. *Journal of Computational and Nonlinear Dynamics*, 9,1, 2014.
- [53] S. J. Montgomery-Smith, D. A. Jack, and D. E. Smith. Spherical Harmonic Solutions of Fiber Orientation Probability Distributions for Composite Processes. *The XV<sup>th</sup> International Congress on Rheology, the Society of Rheology 80<sup>th</sup> Annual Meeting, Monterey, California*, 15, August 2008.
- [54] S. J. Montgomery-Smith, D. A. Jack, and D. E. Smith. A Systematic Approach to Obtaining Numerical Solutions of Jeffery’s Type Equations using Spherical Harmonics. *Composites Part A*, 41:827–835, 2010.
- [55] S. J. Montgomery-Smith, D. A. Jack, and D. E. Smith. The Fast Exact Closure for Jeffery’s Equation with Diffusion. *Journal of Non-Newtonian Fluid Mechanics*, 166:343–353, 2011.

- [56] P Frank Pai. *Highly flexible structures: modeling, computation, and experimentation*. AIAA (American Institute of Aeronautics & Ast, 2007.
- [57] J. W. Phillips and G. A. Costello. Large deflections of impacted helical springs. *Journal of the Acoustical Society of America*, 51(3):967–973, 1972.
- [58] Charles Poole, John Safko, and Albert Goldstein. *Classical mechanics*. Addison-Wesley, 1996.
- [59] Stephen C Preston. The motion of whips and chains. *Journal of Differential Equations*, 251(3):504–550, 2011.
- [60] Stephen C Preston. The geometry of whips. *Annals of Global Analysis and Geometry*, 41(3):281–305, 2012.
- [61] N. Qadir and D. A. Jack. Modeling Fibre Orientation in Short Fibre Suspensions Using the Neural Network-Based Orthotropic Closure. *Composites, Part A*, 40:1524–1533, 2009.
- [62] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian problems*. Chapman & Hall, 1994.
- [63] M. F. Sayre and A. V. De Forest. New spring formulas and new materials for precision spring scales. *Transactions of ASME*, 58:379–387, 1936.
- [64] J. A. Serret. Sur quelques formules relatives à la théorie des courbes à double courbure. *J. De Math*, 16, 1851.
- [65] J. C. Simo, J. E. Marsden, and P. S. Krishnaprasad. The hamiltonian structure of nonlinear elasticity: The material and convective representations of solids, rods, and plates. *Archive for Rational Mechanics and Analysis*, 104:125–183, 1988.

- [66] J. C. Simo and N. Tarnow. The discrete energy-momentum method. conserving algorithms for nonlinear elastodynamics. *Zeitschrift für Angewandte Mathematik und Physik (ZAMP)*, 43:757–792, 1992.
- [67] J. C. Simo, N. Tarnow, and M. Doblare. Non-linear dynamics of three-dimensional rods: Exact energy and momentum conserving algorithms. *International Journal for Numerical Methods in Engineering*, 38:9, 1995.
- [68] J. C. Simo, N. Tarnow, and K. K Wong. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics and Engineering*, 100:1, 1992.
- [69] J. C. Simo and K. K Wong. Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum. *International Journal for Numerical Methods in Engineering*, 31:1, January 1991.
- [70] D. E. Smith and D. A. Jack. *Modeling Orientational Diffusion in Short Fiber Composite Processing Simulations*. National Science Foundation CMMI Engineering Research and Innovation Conference, Honolulu, HI, 2009.
- [71] D. E. Smith, D. A. Jack, and S. J. Montgomery-Smith. Phenomenological-based constitutive model and simulation of fiber interaction for short fiber composite processing. *Poster presentation at the 2008 NSF Design, Service and Manufacturing Grantees and Research Conference, Knoxville, Tennessee*, January 2008.
- [72] Michael Spivak. *A comprehensive introduction to differential geometry*, volume 2. Publish or Perish, Inc, Houston, TX, 1999.
- [73] V. K. Stokes. On the dynamic radial expansion of helical springs due to longitudinal impact. *Journal of Sound and Vibration*, 35:77–99, 1974.

- [74] A. Thess, O. Zikanov, and A. Nepomnyashchy. Finite-time singularity in the vortex dynamics of a string. *Physical Review E*, 59:3637–3640, 1999.
- [75] J. Thomson. On the elasticity and strength of spiral springs. *Cambridge Philosophical and Mathematical Journal*, 3:258–266, 1848.
- [76] G. H. M. van der Heijden.
- [77] A. M. Wahl. *Mechanical Springs*. Penton Publishing Co, Cleveland, Ohio, 1963.
- [78] W. H. Wittrick. On elastic wave propagation in helical springs. *International Journal of Mechanical Science*, 8:25–47, 1966.
- [79] C. Zhang and D. A. Jack. Fiber orientation for suspensions of long fibers: Comparison of rigid and flexible models. *Proceedings of the Society of Rheology’s 82nd Annual Meeting, Santa Fe, NM*, October 2010.
- [80] C. Zhang and D. A. Jack. *Flexible Fiber Suspension Orientation Model Predictions: Comparison Between Rod-Chain Model and Classical Jeffery’s Rigid Ellipsoid Model*. Early Career Technical Conference, Tulsa, Oklahoma, 2010.
- [81] C. Zhang and D. A. Jack. Modeling of flexible fiber motion and orientation in simple shear flow: A comparison between rod-chain model and jeffery’s equation. *Proceedings of ASME IMECE’10, Vancouver, British Columbia, Canada*, November 2010.
- [82] C. Zhang and D. A. Jack. Discontinuous long fiber reinforced composite stiffness predictions for injection molded parts. *Composites, Part A*, 2011, October 2011.
- [83] C. Zhang and D. A. Jack. *Flexible Fiber Waviness as a Role in Fiber Motion Simulation and Material Property Prediction*. Early Career Technical Conference, Fayetteville, Arkansas, 2011.

- [84] D. Zhang, D. E. Smith, D. A. Jack, and S. Montgomery-Smith. Numerical evaluation of single fiber motion for short fiber composites materials processing. *Proceedings of ASME IMECE'10, Vancouver, British Columbia, Canada*, November 2010.
- [85] D. Zhang, D. E. Smith, D. A. Jack, and S. Montgomery-Smith. Rheological study on multiple fiber suspensions for fiber reinforced composite materials processing. *Proceedings of ASME IMECE'11, Denver, Colorado*, November 2011.
- [86] D. Zhang, D. E. Smith, D. A. Jack, and S. J. Montgomery-Smith. Numerical evaluation of single fiber motion for short fiber composite materials processing. *Journal of Manufacturing Science and Engineering*, 133:10021–9, 2011.

# Vita

Dilan Fernando obtained his Bachelors degree in Mathematics from the University of Colombo in Sri Lanka in 2008. After obtaining a Masters degree in Mathematics from Sam Houston State University in Huntsville, Texas he continued his Mathematical studies at the University of Missouri by enrolling in the Mathematics PhD program. Currently he is studying numerical solutions to problems that arise in certain Newtonian mechanical contexts under the direction of Professor Stephen Montgomery-Smith. Starting August 2018, he will be a visiting assistant professor of mathematics at Trinity University, San Antonio, Texas.