DEVELOPMENT OF A PALPABLE VIRTUAL NYLON THREAD AND HANDLING OF BIFURCATIONS

ANKUR DHANIK

(Bachelor of Technology, IIT Kanpur, India)

A THESIS SUBMITTED FOR
THE DEGREE OF MASTERS OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2005

To my grandparents and parents

Acknowledgements

I wish to express my deepest gratitude and appreciation to my supervisors, Dr. Etienne Burdet and Dr. Teo Chee Leong for their instructive guidance and constant personal encouragement during every stage of this research. I greatly respect their inspiration, unwavering examples of hard work, professional dedication and scientific ethos.

I am highly indebted to Dr. Tim Poston, NIAS, India for his continued involvement in my research. He was always the light at the end of a dark tunnel. His mathematical expertise and wide range of knowledge and experience, were always useful. It was great to work with him and learn so many things.

I would like to thank my parents, brother and sister for their continued love and support at every stage of my life.

I gratefully acknowledge the financial support provided by the National University of Singapore through Research Scholarship that makes it possible for me to study for academic purpose.

My gratitude also goes to Mr. Yee Choon Seng, Mrs. Liaw , Mrs. Too, Mrs. Ooi, Ms. Tshin, Ms. Salmah and Mr. Zhang for the helps on facility support in the laboratory so that the project may be completed smoothly.

Last but not the least I will like to thank people with whom I enjoyed a lot

collaborating during my research. I thank Wang Fei for his thoughts, ideas and providing motivation. I also thank Dr. Lim Kian Meng, James K Rappel, Teo Cheng Yong William, Ganesh Gowrishankar, Brice Rebsamen and Long Bo for their inputs and ideas. I was lucky to have a bunch of friends who always kept me cheerful. I thank Tirthankar Bandopadhyay, Naveen Agarwal, M K Saravanan, Ashok M. Prabhu, Desingh Devibalan Balasubramaniam, Dr. K. Bhupal Redddy, Talasila Sateesh, Li Yuan Ping, Koh Niak Wu, Beatrice and Nandagopal for the nice time spent together.

Table of contents

Acknowledgements **iii**

Abstract **viii**

List of Figures **ix**

1 Introduction **1**

2 Handling Bifurcations in Haptics **8**

2.1 Introduction . . . 8

2.2 Instability Studies . . . 11

2.3 Elastic buckling . . . 12

2.3.1 Elastic curve model . . . 12

2.3.2 Equilibrium surface . . . 13

2.3.3 Zeeman machine . . . 15

2.3.4 Haptic forces on the controls . . . 18

2.3.5 Computational instability . . . 20

2.4 Multi-variable states . . . 21

2.4.1 Curved separation of quadratic and degenerate directions . . 23

2.4.1.1 Two internal variables . . . 24

2.4.1.2 Non-axis degeneracy . . . 27

2.4.1.3 Multiple internal variables . . . 28

2.4.2 Around a bifurcation point . . . 32

2.5 Experiments and Results . . . 33

2.6 Discussion . . . 37

3 Palpable Virtual Nylon Thread **39**

3.1 Introduction . . . 39

3.2 Literature review . . . 40

3.3 Development of nylon thread dynamics . . . 41

3.3.1 Geometric Descriptors . . . 42

3.3.2 Dynamics . . . 43

3.4 Experiments and Results . . . 46

3.5 Discussion . . . 49

4 Conclusion **51**

References **53**

Appendices **54**

A k-determinacy **55**

B Splitting Lemma **56**

C Transversality **59**

D The Morse Lemma **60**

Abstract

This research is motivated by a need to simulate surgical suturing which will be used in a microsurgery training system currently under development in a collaboration between Control and Mechatronics laboratory, NUS and National University Hospital. We have developed a palpable (the thread can be felt using a haptic device) virtual nylon thread, which will be used in suturing training. This work on thread led to discovery of problem of vibrations in haptics when bifurcations are encountered. We have developed a novel technique to handle bifurcations in haptics. The usual numerical techniques involve using second order Taylor series approximation for energy of the system, and finding the equilibria using standard Newton's method. This fails near bifurcations, places where number of available equilibria change suddenly. And leads to vibrations when feeling such a system using haptic techniques. We use higher order energy approximation to solve this problem. We show that higher order terms are necessary, but using bifurcation theory prove that third and fourth derivatives of energy (second and third of force) are sufficient.

We model first a single variable system which bifurcates, a Zeeman machine. To our knowledge this is the first haptic realization for it. We demonstrate using Zeeman machine that using third and fourth derivatives in energy approximation, leads to elimination of vibrations . For a multi-variable system, like a 2D elastic curve, which simulates a tape-like thread which has preferred plane of bending, the number of third and fourth derivatives are huge and finding all of them is computationally expensive, an important consideration in haptics for avoiding vibrations due to delayed response. We make use of splitting lemma and prove that it is suf-

ficient to look for higher derivatives along a specific direction. This significantly reduces the computational load as the higher derivatives can be found easily using numerical differentiation along this direction. The results demonstrate that the algorithm works excellently well.

We have developed an energy based method for simulating a nylon thread. The nylon thread is an example of non-linear dynamics. It shows phenomenon such as bifurcations, leading to 'snap-through' jumps and large flexible deformation, which is in essence of knotting. We model the thread energy using stretching and bending energies, and find equilibria. The algorithm developed for handling bifurcations is applied, and works extremely well. We successfully demonstrate the phenomena associated with a real nylon thread in our virtual thread. Smooth haptic experience is also achieved. A full description of thread will require inclusion of twisting energy and self-collision detection, and will be dealt with in future. Nevertheless, our technique for the first time brings out the characteristic features associated with a nylon thread, which are ignored sometimes for more visual realism.

List of Figures

1.1 Right topology for loop formation 1
1.2 Equilibrium states and bifurcation 3
1.3 Elimination of unphysical chatter at snap points 6
1.4 A user manipulating virtual nylon thread using PHANToM haptic device. 7

2.1 Buckling of a tape-like thread 9
2.2 Two-dimensional elastic curve model. 12
2.3 The Zeeman machine 14
2.4 The equilibrium surface 17
2.5 Vibration experienced in thread, Zeeman machine and standard cusp simulation 20
2.6 Newton's method 21
2.7 Higher-order computed equilibria for the Zeeman machine and standard cusp 22
2.8 Curve on which equilibria lie 25
2.9 Equilibria convergence 34
2.10 Force profiles obtained while moving the control point of 2D curve . 35
2.11 Refresh rate as a function of number of internal variables 36

3.1 Pressing down a compressible upright 41
3.2 Three dimensional curve model 42
3.3 Creating a loop 46
3.4 Snap sequence 47
3.5 3D thread smooth haptics 48
3.6 Refresh rate vs number of segments in thread model 48

Chapter 1

Introduction

Many studies have investigated interaction with, walls, hard surfaces, soft textured surfaces, springy frictional surfaces. These are always surfaces of fixed shape, and the local deformation depends on the user's current input. On the other hand many simulations - for example surgery - involve large scale changes of shape. Pull a membrane, tear it off a gall bladder and you feel where it is still attached, inches away. Pulling and pushing a nylon thread, such as that used in suturing is a simpler but similar problem. The forces depend on the current shape and the user's input history. The current shape changes smoothly, but sometimes it jumps.

Surgical simulation [2] requires simulating suturing process, wherein the surgeon manipulates a nylon thread to form a surgical knot. To make this thread palpable, by which we mean that the thread can be felt using a force feedback haptic device, dynamics based simulation of thread is required. Where a suture

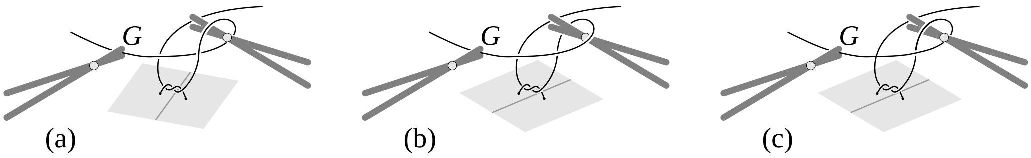


Figure 1.1: The surgeon must create exactly the right loop (c) to reach through for a knot, and does *not* trace its curve through the air, in contrast to knot tying simulation based on geometry [1]. G refers to the grasped end of thread

needle creates a track in tissue that the suture thread must follow, it is enough to define a curve [1] by pulling a needle through it. However, a key step in surgical knotting is to grasp one thread with the left forceps and create a loop through which the right forceps reach the other (Fig. 1.1). The first grasp taken far to the left, pushes grasped point rightwards in untraced space. Nylon mechanics forms the loop, to the right of any point where the forceps tip has gone. A 'following simulation', where the thread is like a multi-linked chain with one node leading rest of the nodes, does not match the task. This example shows the unavoidable need to simulate the bending of an elastic curve fixed at one point and subject to a changing user-controlled constraint at another. The shape must come from mechanics, not by geometric tracking, and the surgeon must master this mechanics - including the propensity to snap.

A nylon thread exhibits snap-through jumps (Fig. 1.3A), which are caused whenever it passes through energy bifurcation. Understanding of this phenomenon requires first defining a few concepts:

Equilibrium: A physical system is said to be in equilibrium when the derivative of its energy with respect to the variables of the system is equal to zero. A physical system not disturbed by external forces always reaches its equilibrium state corresponding to a minimum of energy. A system can have single or multiple equilibria.

Bifurcation: A system is said to bifurcate when the number of equilibrium states available to it change suddenly, with very slight change in the system parameters. For example, a system can have two stable equilibria at a given time, with the system resting in one of them, and on changing the system parameters slightly one will vanish.

The red ball in Fig. 1.2 illustrates the variable which defines the state of the system. The curves show the energy as a function of system variables, for a fixed set of system parameters. In Fig. 1.2A, the system has three equilibria (three places on the energy curve where derivative of energy is zero), and the system is in one of the two local minima. With slight change in parameters, the equilibrium states of the system change (Fig. 1.2B) and finally one stable minimum vanishes and the system jumps to the remaining (now global) minimum (Fig. 1.2C). A similar phenomenon is encountered when the tectonic plates of earth shift suddenly leading to massive earthquakes [15] or when nylon thread undergoes snap-through jump.

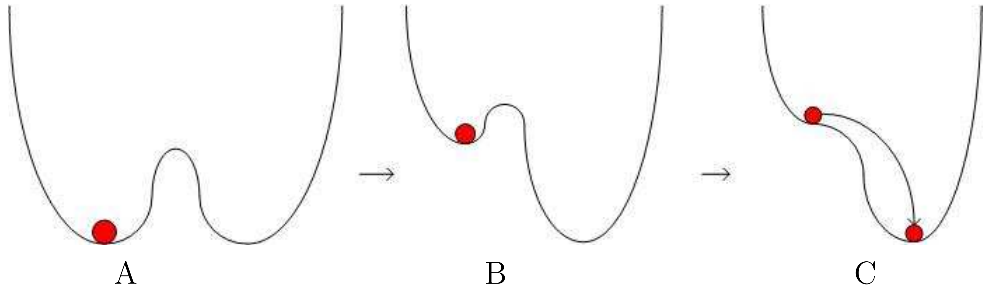


Figure 1.2: Number of equilibrium states of the system changes near bifurcations.

In 3D simulation, just as with real nylon (Fig. 1.3), the shape taken by the thread does not always change smoothly as G (the grasped end) smoothly moves. This leads to a completely different class of problem in haptic simulation. Haptics to date has generally concerned itself with systems where the response force F is a function $F(u)$ of the user's immediate input $u(t)$ at time t (typically a spatial position (x, y, z), perhaps including orientation data for the grip, and sometimes applied force, which may include torque). Ideally this would be an instantaneous response — reasoning with Newton's third law usually assumes that action and reaction are equal, opposite, and also simultaneous — but real haptics prevents this. Delayed response easily creates vibrations or chaos, and a large literature has developed on preventing this. For example, [5], [6], [7], [8], [9], [10], [11] discuss stability criteria for a force response law $F(u)$, generally posing questions equivalent to "If the user attempts to hold the device steady, or apply a steady force, will the system converge to a steady state?". We here address a different source of instability, in a wider context, where the required returned force is a function of input history (not merely of $u(t_{\text{now}})$, but of previous events).

The problem of handling bifurcation related instabilities can be solved by using higher order Taylor series approximation of thread energy, used in the energy minimizing scheme for simulating dynamics of thread. But in a multi-variable system such as nylon thread it is not computationally feasible to calculate higher derivatives in so many variables. This presents a technical challenge for the solution scheme that has to be devised. In the first part of this thesis, we use 'quasi-static' mechanics to bring out the role of bifurcation (change in the set of available equilibria), and the ways that bifurcation theory can reduce the computational load, but emphatically preserving the 'snap' buckling instability which still arises in such systems (Fig. 1.3). We illustrate this with the first haptic realization of the Zeeman machine [12], an example with minimal degrees of freedom and an elastic 2D curve, example with many degrees of freedom. By using the second and third derivatives of force instead of only the first one as in usual algorithms, we can suppress the physical shaking both in numerical tests and experimental implementations [13].

Several schemes have been used to model a rope or nylon for simulating knot tying. In [1] Joel Brown proposed a geometric approach to realize real time knot tying, in which the rope is modelled as a series of nodes linked together. The grasped node (the leader) is followed by rest of the nodes (followers) such that the internodal length remains constant. This follow-the-leader approach leads to a visually nice but (even visually) incorrect knot-tying simulation, because the thread's mechanics, which play an important role in loop formation (Fig. 1.1) are not considered. A dynamic approach was recently developed in our lab [3],[4], in which the nylon thread is modelled as a series of masses and springs. Springs implement stretching, bending and twisting. When the grasped node is perturbed, the nodes move in direction of resultant force at each node until the force is zero. Real time simulation is achieved with this approach, with effects such as twisting are realized, however essential non-linearities in the mechanics of thread such as snaps have not been dealt with. The second part of this thesis describes details of loop formation which involves minimization of thread's energy. We develop a 3D curve as a paradigm for 3D thread. The 3D curve is broken into segments and computationally simple bending and stretching energies are associated with it. We find the equilibria of this system using the standard Newton method to find equilibria of the thread subject to the constraints, which include the position and direction of the grasped node and segment controlled using a PHANToM haptic device(Fig. 1.4). Multiple equilibria lead to snap-through jumps which is demonstrated by our nylon thread model as in Fig. 1.3C, 1.3D. When the currently followed equilibrium vanishes, the nylon thread descends to another equilibrium in a 'snap-through'. The situation near bifurcations is handled with higher order methods, where we use the scheme developed in the first part of this thesis, which emphasizes its significance.

The thesis is organized as follows. Chapter 2 addresses the problem of vibrations near bifurcations , illustrates it on Zeeman machine and 2D curve, introduces a novel scheme for handling bifurcations, thereby eliminating these vibrations, and demonstrates its effectiveness. In Chapter 3 we describe energy based scheme developed by us and our three dimensional thread model. We emphasize the application and significance of the scheme developed in Chapter 2 and demonstrate the possibilities offered by the virtual thread.

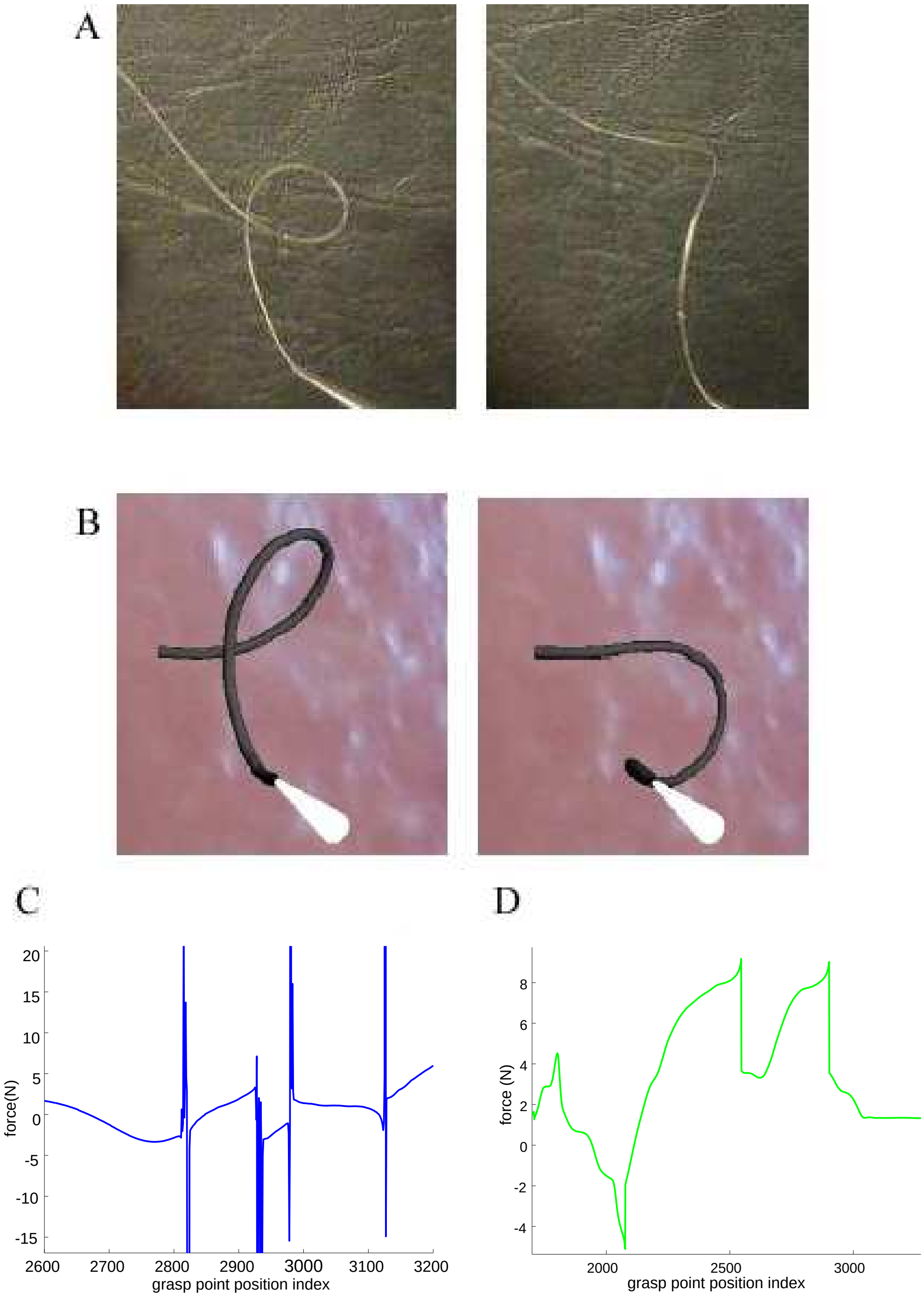


Figure 1.3: A manipulated nylon suture moves sometimes smoothly with the forceps, sometimes jumps as the grip crosses an unseen boundary. *A* shows this on a real nylon thread, and *B* in our simulation of a virtual nylon thread. *C* shows unphysical chatter at the snap points occuring when using a standard Newton algorithm approach. We developed a novel approach based on bifurcation theory to successfully address this problem, as is illustrated in *D*. Visualizing the movement requires movies available at *http://guppy.mpe.nus.edu.sg/∼eburdet/People/ankur/video.html* .

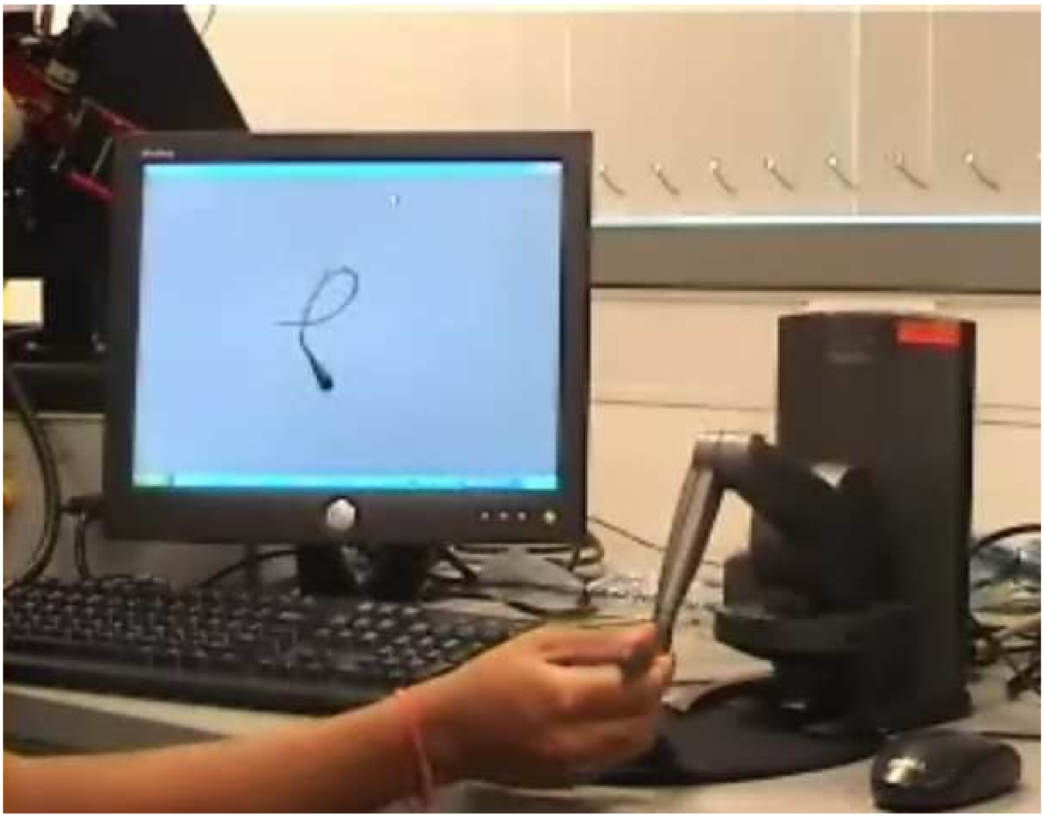


Figure 1.4: A user manipulating virtual nylon thread using PHANToM haptic device.

Chapter 2

Handling Bifurcations in Haptics

2.1 Introduction

Haptics to date has generally concerned itself with systems where the response force F is a function $F(u)$ of the user's immediate input $u(t)$ at time t (typically a spatial position (x, y, z), perhaps including orientation data for the grip, and sometimes applied force, which may include torque). Ideally this would be an instantaneous response — reasoning Newton's third law usually assumes that action and reaction are equal, opposite, and also simultaneous — but real haptics prevents this. Delayed response easily creates vibrations or chaos, and a large literature has developed on preventing this. For example, [5], [6], [7], [8], [9], [10], [11], discuss stability criteria for a force response law $F(u)$, generally posing questions equivalent to "If the user attempts to hold the device steady, or apply a steady force, will the system converge to a steady state?".

We in this thesis address a different source of instability, in a wider context, where the required returned force is a function of input history (not merely of $u(t_{\text{now}})$, but of previous events). The system model includes an internal state σ of positions, momenta, *etc.*, whose dynamics $\dot{\sigma}(\sigma, u)$ depends on the current state but is also influenced by u, and response force F is a function $F(u, \sigma)$ of both the current input and the state which the system has reached. There can be instability from a number of sources, notably:

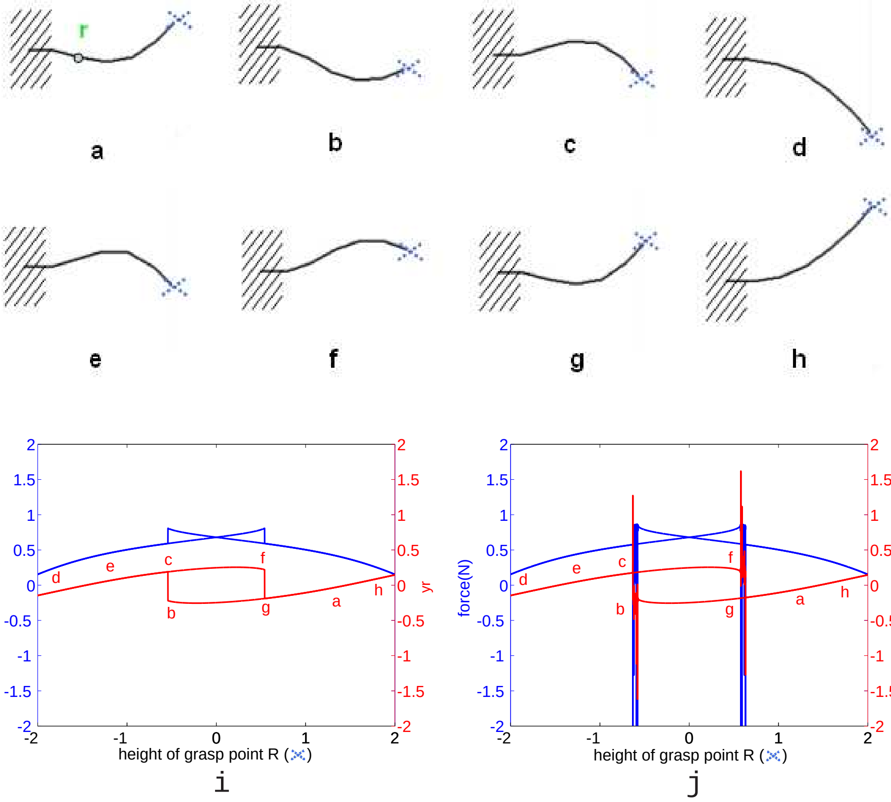


Figure 2.1: We consider a more tape-like thread, whose buckling can stay in its preferred-bending plane, with left end clamped. Moving the right end R vertically causes snaps between 'bent up' and 'bent down' states a–h. Corresponding heights for a reference point r on the curve jump up or down *after* R has crossed the mid-level. In i the evolution is tracked using our algorithms: j, computed by a standard approach, shows unphysical chatter at the snap points. Related video is available at [14].

1. The dynamics $\dot{\sigma}$ may be oscillatory or chaotic even for constant u, and even for the real system modelled. (A fibrillating heart feels chaotic, like a handful of excited worms, and a simulation should feel the same.)

2. Delay in computing F may introduce inappropriate instability, in interaction with the changes in u resulting from the user-side response law $\dot{u}\,(u, F)$. (We include under $\dot{u}$ such neuromotor effects as change in position under the combined forces of the haptic device and the user's muscles, tendons, *etc.* These affect the dynamics even where force due to the user is not mechanically sensed as an input to the haptics.)

3. Instability in computing $\dot{\sigma}$, with consequent fluctuations in $F\,(u, \sigma)$, even if F has no impact on the evolution of u. This is particularly liable to occur transiently where u is changing and the dynamics of $\dot{\sigma}$ pass through a

bifurcation point. The convergence criterion used in much of the literature, which assumes a user is not attempting to change σ, is unhelpful for such transients.

We here address (3), barring oscillations and chaos by 'quasi-static' mechanics to bring out the role of bifurcation, and the ways that bifurcation theory can reduce the computational load, but emphatically preserving the 'snap' buckling instability which still arises in such systems (Fig. 2.1). We compute the haptic response F, replacing $F(u)$ by an implicit force law, but do not in our simulations allow u to respond to it. For a clear focus on source (3), we change u in a pre-programmed, unresponsive way and thus exclude (2), though user experiments with a real haptic device cannot uphold such an exclusion. The physical shaking that prompted this study is in fact suppressed in both our numerical and our experimental tests, supporting the relevance of our analysis and the computational strategy it gives rise to.

Most equilibria — vanishing points for force — are efficiently found by Newton's method, based on the first derivative of force as a function of state. This linearization of force works well wherever the linear approximation is non-degenerate, which is true near most individual equilibria. However, bifurcation occurs at points where the linear approximation *is* degenerate, and in sustained interaction such points can be topologically unavoidable. Buckling, in particular, is a feature of real manipulated systems such as surgical stents and sutures, and buckling is rooted in bifurcation.

We describe the system in which we first encountered this problem, and an illustrative example (the 'Zeeman machine' [12] with minimal internal degrees of freedom. In each case the reduction techniques of bifurcation theory show that higher derivatives of force than the first become important numerically, but that the second and third are sufficient: we illustrate this with a haptic realization of the Zeeman machine. For n internal degrees of freedom, an energy-minimizing force has $n\,(n+1)\,/2$ first derivatives but $n\,(n+1)\,(n+2)\,(n+7)\,/24$ second and third, which would confine use of this to small systems, except that theory also drastically limits the ones actually needed. We illustrate this with a 2D elastic curve simulation.

2.2 Instability Studies

Most of the literature generally poses questions equivalent to "If the user attempts to hold the device steady, or apply a steady force, will the system converge to a steady state?" Fundamental stability and performance issues associated with haptic interaction are addressed in [5], [6]. Necessary and sufficient conditions for the stability of a haptic simulation are developed, assuming the human operator and virtual environment are passive. In the dissertation [7] and [8], [9], the problem of guaranteed stability in the haptic display of virtual environments is addressed. An implementation of stiff virtual wall is being focussed upon. A wall represents boundary between zone of high stiffness and low stiffness. It has been observed that humans are adept at adjusting their behavior to destabilize a virtual wall, if possible. In other words, users will quickly find ways to setup a sustained or growing oscillations by gripping the haptic device lightly or firmly, as necessary. The ability to set up oscillations is evidence of *active walls*: because the frequencies of these oscillations are often outside the range of voluntary motion, and because this behavior is not observed with physical walls, it is evident that the energy supply for the oscillations is the virtual wall, not the human. Suitable criteria have been developed to make the wall passive. Colgate et.al. in [10] and [11] investigated non-linear mass/spring/damper virtual environments designed to prevent oscillations in haptic display and other chaotic behavior in the signal presented to human operator.

The scheme developed in this thesis is, to the best of our knowledge, the first *in haptics* to address oscillations arising near bifurcations. No previous work for comparison is available.

2.3 Elastic buckling

This study was prompted by work towards simulation of surgical suturing [2]. The details of loop formation involve minimization of energy, an essentially 3D matter discussed in detail in next chapter. The shape must come from the mechanics, not by geometric tracking, and the surgeon must master this mechanics — including the propensity to snap.

In 2D or 3D simulation (Fig. 2.1), just as with real nylon, the shape taken by the thread does not always change smoothly as grasped end smoothly moves. The change in the set of available equilibria (the *bifurcation*) that makes it jump causes numerical failure, in techniques that work well at points of smooth behavior. We take the 2D case here for clarity, as it well exemplifies snaps and continuous buckling, and we can handle them without sacrificing haptic speed.

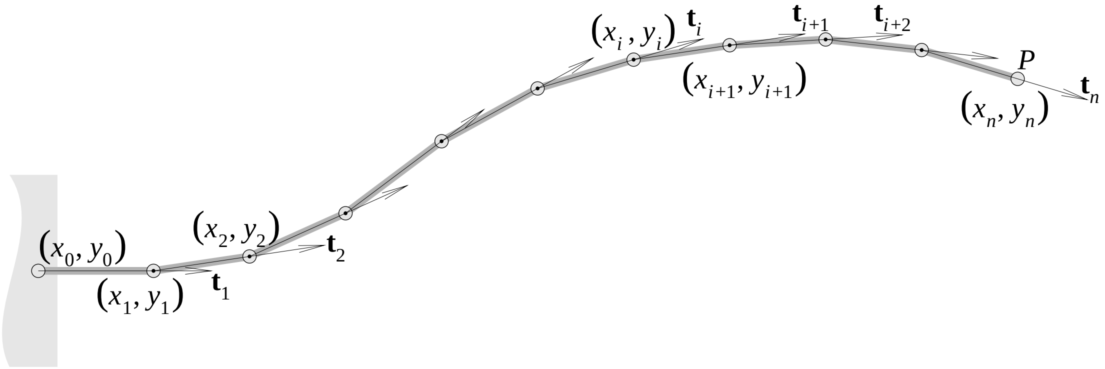


Figure 2.2: Two-dimensional elastic curve model.

2.3.1 Elastic curve model

We here provide a brief insight into a 2D curve model, which is sufficient to understand the concepts involved in this chapter. This resembles a more-tape-like thread, whose buckling can stay in its preferred bending plane. A detailed analysis in three dimensions will be described in the next chapter. We model the 2D curve (Fig. 2.2) using nodes (x_i, y_i), separated by vectors $\mathbf{v}_i = (x_i, y_i) - (x_{i-1}, y_{i-1})$ giving unit tangent vectors $\mathbf{t}_i = \mathbf{v}_i / \|\mathbf{v}_i\|$. A configuration c is given by the $2n - 1$ non-constants in $(0, 0, x_1, 0, \ldots, x_n, y_n)$. For computational speed we take bending energy proportional to $b_i(c) = 1 - \mathbf{t}_{i-1} \cdot \mathbf{t}_i$ at node i: to third order this matches the square of the node's bending angle, which normally remains small. An extension energy $e_i(c) = \gamma \left(\|\mathbf{v}_i\| - l_i\right)^2$, where l_i is a reference length, allows for compression and stretching. We fix $(x_0, y_0) = (0, 0)$ and clamp the end by setting $y_1 \equiv 0$. We 'grasp' by letting the user (or a program simulating the user) fix a point (X, Y), giving

$$(x_0, y_0, y_1, x_n, y_n) = (0, 0, 0, X, Y) \tag{2.1}$$

as current constraint. The total energy is

$$E(c) = \sum_{i=1}^{n} \Big(b_i(c) + e_i(c) \Big) . \tag{2.2}$$

2.3.2 Equilibrium surface

We thus identify an equilibrium — subject to (2.1) — with a state c such that

$$\begin{aligned} &\left[\begin{array}{cccccc} \frac{\partial E}{\partial x_1}(c) & \frac{\partial E}{\partial x_2}(c) & \frac{\partial E}{\partial y_2}(c) & \cdots & \frac{\partial E}{\partial x_{n-1}}(c) & \frac{\partial E}{\partial y_{n-1}}(c) \end{array} \right] \\ &= \left[\begin{array}{ccc} 0 & \cdots & 0 \end{array} \right], \end{aligned} \tag{2.3}$$

These $2n-3$ equations and (2.1) typically define a set of isolated points in $(0, 0, x_1, 0, \ldots, x_{n-1}, y_{n-1})$-space, for fixed (X, Y), and a 2D surface M in the $(2n-1)$-dimensional space of $(0, 0, x_1, 0, \ldots, x_{n-1}, y_{n-1}, X, Y)$ space with all (X, Y) considered. For each m in M there is a force $f(m)$ appropriate for the haptic device to deliver to the user (see §2.3.4 below), but since there is not a unique m for each (X, Y), f does not define a haptic response force $F(X, Y)$ as a function of user position. This is clearest in the analogous case of a simpler elastic system, the Zeeman machine (Fig. 2.3), which has often been simulated but not previously (to our knowledge) haptically. It is not a flexible-cruve system - except in the ignored sense that the elastic strings could bend - but undergoes an bifurcation identical to the nylon thread. The following account (§ 2.3.3) follows [12], and serves here as an introduction to this bifurcation, and the methods available for its analysis.

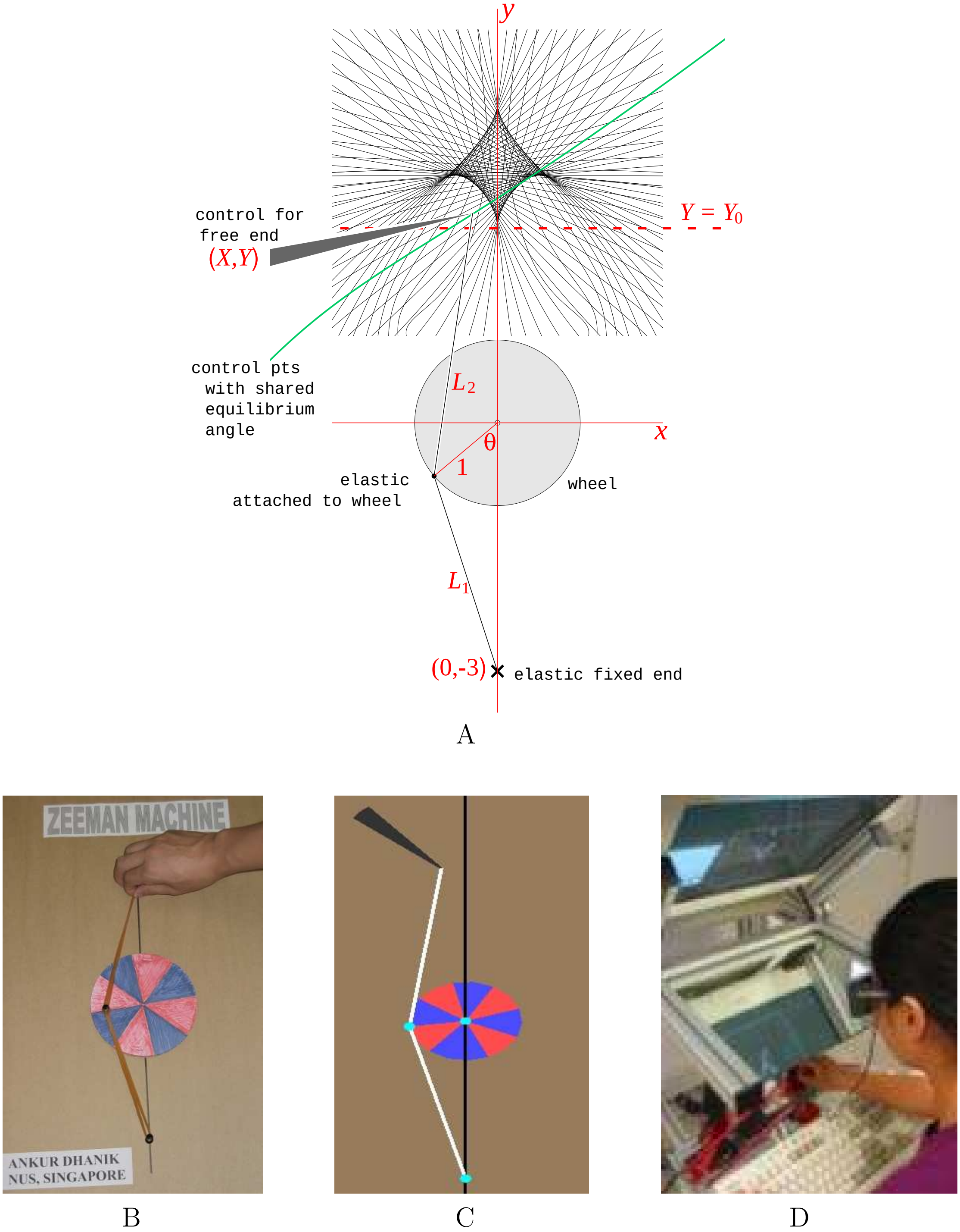


Figure 2.3: The Zeeman machine: a unit-radius wheel (diagram A) turns about $(x, y) = (0, 0)$ attached to elastica held at $(0, -3)$ and at (X, Y) with unextended lengths 1 and Young's Modulus $\gamma = 2$. (Other dimensions would give equivalent results.) For the quasi-static behavior discussed here the wheel should not be heavy or frictionless, but static friction should be low. B shows the physical prototype we have built, C the virtual one, and D the reach-in haptic environment for interaction with it. Related videos are available at [14].

2.3.3 Zeeman machine

Zeeman Machine is a simple example of systems demonstrating energy bifurcations. Bifurcation is a phenomenon where a system encounters change in the number of equilibrium states available, with a slight change in the control parameters of the system. Fig. 2.3A explains the Zeeman Machine. Moving the control (X, Y) in Fig. 2.3A modifies the energy dependence on θ,

$$\begin{aligned} E_{(X,Y)}(\theta) &= \left(\sqrt{\sin^2\theta + (\cos\theta - 3)^2} - 1\right)^2 \\ &+ \left(\sqrt{(X + \sin\theta)^2 + (Y + \cos\theta)^2} - 1\right)^2 . \end{aligned} \tag{2.4}$$

Expanding in θ and $\left(X, \widetilde{Y}\right) = (X, Y_0 - Y)$, where $(0, Y_0) = \left(0, \left(3 + \sqrt{33}\right)/4\right)$ is the lower cusp point in Fig. 2.3A, gives

$$\begin{aligned} E_{\left(X,\widetilde{Y}\right)}(\theta) &= C\left(X, \widetilde{Y}\right) + \left(\frac{141 - 21\sqrt{33}}{16}\right)\frac{\theta^4}{4} \\ &+ \left(\sqrt{33} - 3\right)\widetilde{Y}\frac{\theta^2}{2} + \left(\frac{33 - 7\sqrt{33}}{8}\right) X\,\theta + \text{tayl} \end{aligned} \tag{2.5}$$

where 'tayl' represents terms above fourth order in θ or above linear in $\left(X, \widetilde{Y}\right)$. We chose Y_0 to give a vanishing θ^2 term when $\widetilde{Y} = 0$, so that when $X = 0$ the angle $\theta = 0$ gives a local minimum for E when $\widetilde{Y} > 0$ and a local maximum when $\widetilde{Y} < 0$. With these non-zero coefficients, 'k-determinacy' test (Appendix A, [15]) guarantees the following: there exists, for small θ, X and $\widetilde{Y}$, a change of coordinates to

$$\begin{aligned} &\widehat{X} = X + O^2\left(X, \widetilde{Y}\right) \qquad \widehat{Y} = \widetilde{Y} + O^2\left(X, \widetilde{Y}\right) \\ &t = \theta + O^2\left(\theta, X, \widetilde{Y}\right) \end{aligned} \tag{2.6}$$

which smoothly replaces X and $\widetilde{Y}$ by coordinates that to first order are the same, and reparametrizes θ in an $\left(X, \widetilde{Y}\right)$-dependent way that also leaves it unchanged to first order, and gives

$$e_{\left(\widehat{X},\widehat{Y}\right)}(t) = \widehat{C}\left(\widehat{X},\widehat{Y}\right) + \left(\frac{141-21\sqrt{33}}{16}\right)\frac{t^4}{4} \tag{2.7}$$
$$+\left(\sqrt{33}-3\right)\widehat{Y}\frac{t^2}{2} + \left(\frac{33-7\sqrt{33}}{8}\right)\widehat{X}\,t \quad \textit{exactly,}$$

with no neglected higher terms unless we approximate the $\left(\widehat{X},\widehat{Y}\right)$-dependent constant $\widehat{C}$ by linear expansion[1] For convenience we rescale to

$$\overline{Y} = \frac{4\sqrt{33}-12}{\sqrt{141-21\sqrt{33}}}\widehat{Y}, \quad X = \frac{33-7\sqrt{33}}{4\sqrt[4]{141-21\sqrt{33}}}\widehat{X}\,,$$
$$T = t\frac{\sqrt[4]{141-21\sqrt{33}}}{2} \tag{2.8}$$

giving

$$e_{\left(\overline{X},\overline{Y}\right)}(T) = \overline{C}\left(\overline{X},\overline{Y}\right) + \frac{T^4}{4} + \overline{Y}\frac{T^2}{2} + \overline{X}T \tag{2.9}$$

The constant-free version $T^4/4 + \overline{Y}\,T^2/2 + \overline{X}T$ is known as the **standard cusp**: very many bifurcations reduce to it in this way. For small T the T-positions of equilibria, where $\partial e/\partial T = 0$, correspond closely to the positions of points where

$$\frac{\partial E_{\left(\widehat{X},\widehat{Y}\right)}}{\partial\theta} = 0 \tag{2.10}$$

exactly, that truly exist for small enough $\left(\overline{X},\overline{Y}\right)$. This is in contrast to the zeros of an example like

$$u^2 + 4uv^2 + O^4(u,v)\, : \tag{2.11}$$

the bare cubic u^2+ $4uv^2$ vanishes on the u-axis and on $u = -2v^2$, while $u^2 + 4uv^2 + 5v^4 = \left(u+2v^2\right)^2 + v^4$ vanishes only at $(0,0)$. Since u^2+ $4uv^2 + O^4(u,v)$ allows both possibilities, little can be said about its zeroes, unlike those of $\partial E/\partial\theta$ above. However,

$$W\left(u,v\right) = u^2 + 4uv^2 + O^5(u,v) \tag{2.12}$$

[1] Most expositions of unfolding theory discard this ‘variable constant’ term, as merely representing a change of origin in energy space. This is reasonable if one is concerned only with the θ values at equilibria, but for haptics we are concerned also with actual energy.

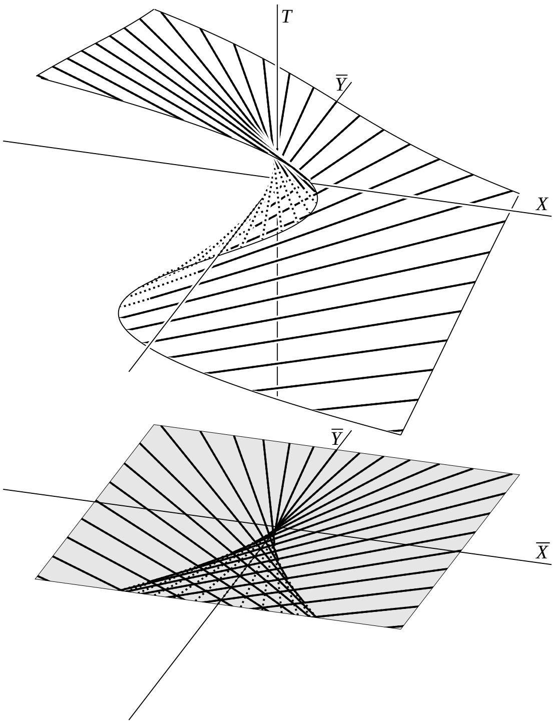


Figure 2.4: The equilibrium surface (2.13), in $\left(\overline{X}, \overline{Y}, t\right)$ space and in projection to $\left(\overline{X}, \overline{Y}\right)$ space. The lines are derived from (2.13) and lie on the equilibrium surface. For any fixed value of T, a line is defined by the equation. The lines help in visualizing the correspondence between the equilibrium surface and its projection. The points on dotted lines, and their projections, correspond to unstable equilibria (energy maxima). Any control with $\left(\overline{X}, \overline{Y}\right)$ with $27\overline{X}^2 < 4\overline{Y}^3$ gives an unstable equilibrium between two stable ones

is locally exactly reducible — for any $O^5(u, v)$ remainder — to the form U^2+4UV^2 with no tayl, by a smooth coordinate change $(U, V) = (U(u, v), V(u, v))$ which to first order is the identity. The Taylor polynomial $u^2 + 4uv^2$ is *4-determinate*, considered as a 4^{th} order expansion with 0 quartic term: any 5^{th} addition can locally be transformed away. It is not 3-determinate, since quartic additions can change it.

The equilibrium condition, for varying $\left(\overline{X}, \overline{Y}\right)$, gives

$$T^3 \ + \overline{Y}T + \overline{X} = 0. \tag{2.13}$$

Moving the control $\left(\overline{X}, \overline{Y}\right)$ has interesting effects on the wheel, which generally stays near the equilibrium surface until a jump is forced by meeting a fold. To-and-fro $\overline{X}$ motion with $\overline{Y} < 0$ gives hysteresis, while looping clockwise around the cusp point gives repeated upward jumps in T as $\left(\overline{X}, \overline{Y}\right)$ crosses the $\overline{X} < 0$ branch of the cusp curve, alternating with smooth decrease. Looping anti-clockwise gives smooth increase, alternating with repeated jump decreases when crossing the $\overline{X} > 0$ branch [14].

We model this with the 'quasi-static' assumption that as $\left(\overline{X}, \overline{Y}\right)$ varies, T in (2.13) or θ in (2.4) moves continuously — if possible — to occupy always a local minimum, neglecting inertial terms. This extends the common assumption that a single elastic string always has the minimal energy configuration (straight, and stretched evenly between its end points) that as in (2.4) makes its energy a quadratic function of the end positions, ignoring transient vibrations through curved states. Here, however, the function is of higher order, with multiple minima, and as $\left(\overline{X}, \overline{Y}\right)$ varies the set of equilibria can *bifurcate*, changing its count and topology.

2.3.4 Haptic forces on the controls

As with a plain elastic string, the force on the free end P may be defined as the 'covector' f mapping infinitesimal changes $\overline{\delta} = \left(\delta\overline{X}, \delta\overline{Y}\right)$ in the position $\left(\overline{X}, \overline{Y}\right)$ of P to the energy $f\left(\overline{\delta}\right)$ such δ subtract from the system, most often via the unique vector F such that $F \cdot \overline{\delta} = f\left(\overline{\delta}\right)$. One result of bifurcating minima is that the equilibrium equation (2.3, 2.10, or 2.13) does not define the state as a function $\theta_{\text{equ}}(x, y)$ of the controlled end. Nor, therefore, does a unique energy $e\left(\overline{X}, \overline{Y}\right) = E_{\left(\overline{X}, \overline{Y}\right)}\left(\theta_{\text{equ}}\left(\overline{X}, \overline{Y}\right)\right)$ for each $\left(\overline{X}, \overline{Y}\right)$, exist as for a spring. We cannot therefore compute the energy change required in adding $\overline{\delta}$ to $\left(\overline{X}, \overline{Y}\right)$ as a δe found by differentiation of e, since e does not exist. We must implicitly differentiate $E_{\left(\overline{X}, \overline{Y}\right)}(\theta)$ subject to the equilibrium condition. This is straightforward at an equilibrium point $\left(\overline{x}, \overline{y}, \overline{\theta}\right)$ where $\partial^2 E/\partial\theta^2 \neq 0$, giving a non-degenerate linearization in $\left(\widetilde{x}, \widetilde{y}, \widetilde{\theta}\right) = \left(x - \overline{x}, y - \overline{y}, \theta - \overline{\theta}\right)$

$$\left.\frac{\partial^2 E}{\partial x \partial \theta}\right|_{\left(\overline{x}, \overline{y}, \overline{\theta}\right)} \widetilde{x} + \left.\frac{\partial^2 E}{\partial y \partial \theta}\right|_{\left(\overline{x}, \overline{y}, \overline{\theta}\right)} \widetilde{y} + \left.\frac{\partial^2 E}{\partial \theta^2}\right|_{\left(\overline{x}, \overline{y}, \overline{\theta}\right)} \widetilde{\theta} = 0 \tag{2.14}$$

$$\theta \approx \overline{\theta} - \frac{\left.\frac{\partial^2 E}{\partial x \partial \theta}\right|_{\left(\overline{x},\overline{y},\overline{\theta}\right)} \widetilde{x} + \left.\frac{\partial^2 E}{\partial y \partial \theta}\right|_{\left(\overline{x},\overline{y},\overline{\theta}\right)} \widetilde{y}}{\left.\frac{\partial^2 E}{\partial \theta^2}\right|_{\left(\overline{x},\overline{y},\overline{\theta}\right)}} \tag{2.15}$$

The Implicit Function Theorem ensures that there is a smooth function $\theta\left(x,y\right)$ locally approximated to $O^2\left(\widetilde{x},\widetilde{y}\right)$ by the linear one (2.15), giving θ's local equilibrium value as a function of x and y, and thus that there is no bifurcation — the equilibrium moves, but persists. Inserting (2.15) into E for a locally defined energy $e\left(\widetilde{x},\widetilde{y}\right)$ suffices, by the Chain Rule, to find $\begin{bmatrix} \partial e/\partial\widetilde{x} & \partial e/\partial\widetilde{y} \end{bmatrix}$ at $\left(\overline{x},\overline{y}\right)$ and thus the force there. (With N internal variables x_i, the corresponding condition is non-degeneracy of $[\partial^2 E/\partial x_i \partial x_j]$.) The logic is more subtle where $\partial^2 E/\partial\theta^2$ vanishes, since the theorem does not apply and the number of equilibria can change discontinuously.

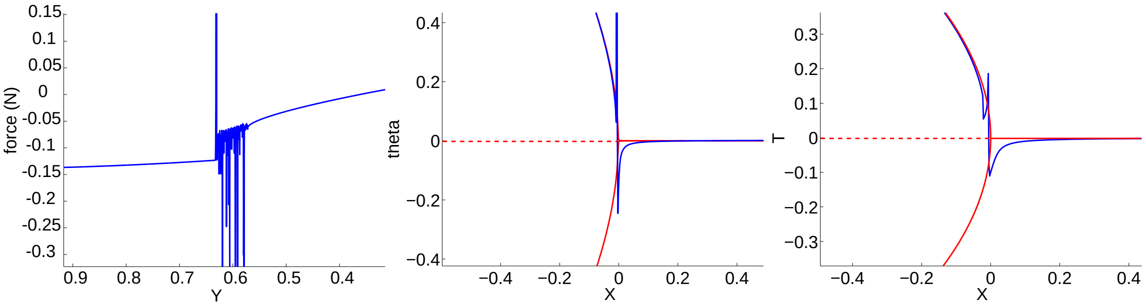
As (2.9) still exactly represents the original energy of the system, merely relabelling the equilibria by reparametrizing the states and controls, the energy cost of a change is unaltered. First derivatives in the $\overline{X}$ and $\overline{Y}$ directions suffice to compute force, so we replace (2.9) by

$$E_{\left(\overline{X},\overline{Y}\right)}\left(T\right) = C_X\overline{X} + C_Y\overline{Y} + \frac{T^4}{4} + \overline{Y}\frac{T^2}{2} + \overline{X}T \tag{2.16}$$

with Taylor C_X and C_Y. Along any $\left(\overline{X}\left(s\right),\overline{Y}\left(s\right),T\left(s\right)\right)$ with $\left(\overline{X}\left(0\right),\overline{Y}\left(0\right),T\left(0\right)\right) = \left(0,0,0\right)$ that satisfies the constraint (2.13), the derivative

$$\frac{d}{ds}\left(\frac{T^4\left(s\right)}{4} + \overline{Y}\left(s\right)\frac{T^2\left(s\right)}{2} + \overline{X}\left(s\right)T\left(s\right)\right) \tag{2.17}$$

exists and is zero. In a haptic simulation, the force to be displayed is thus simply $\left(-C_X,-C_Y\right)$: this holds true for general bifurcation points in a quasi-static system. Where there is no jump between different equilibrium branches, the force is continuous in the control points, and indeed is slightly simpler to compute than at a regular point, where the slope of the equilibrium surface enters the computation.



(a) (b) (c)

Figure 2.5: Force computed with (2.18) for (a) the thread and results of equilibrium search (blue) for (b) the Zeeman machine and (c) standard cusp, against analytical equilibria (red), as hand position is varied. The chattering due to a very flat vanishing of the gradient appears even for the reduced polynomial case. Position of hand is represented by (X,Y).

2.3.5 Computational instability

Our first physical implementation of the model (2.2), with the user's hand or a program loop varying the position of the free end P, gave major vibrations around the transition between stretching and buckling the thread. Ten un-held nodes in (2.2) give nineteen state variables, so in Fig. 2.5a we plot the y-component of force returned to the hand. To isolate the problem in a fewer-variable system we applied the same solution method (2.18) to the Zeeman machine and the polynomial standard cusp, with the similar results shown.

This instability is essentially linked to the change of control point: with constant user position or applied force, the system converges to an equilibrium. The criteria in [5], [6], [7], [8], [9], [10], [11], for convergence with fixed inputs, do not connect with it. The problem is strictly in the solution algorithm.

The normal second-order Newton search[16] for a minimum of an energy function f is

$$\begin{array}{c}\text{linearly approximate the field } \nabla f \text{ at guess } x_i \\ \searrow \ \text{solve for a zero-gradient point } x_{i+1} \nearrow\end{array} \tag{2.18}$$

Fig. 2.6 shows the one-variable case. This converges fast where the minimum is dominated by its quadratic term (Fig. 2.6a), but the solution step becomes unreliable (Fig. 2.6b) for f near a bifurcating minimum, where the quadratic term vanishes. We must thus invoke higher terms.

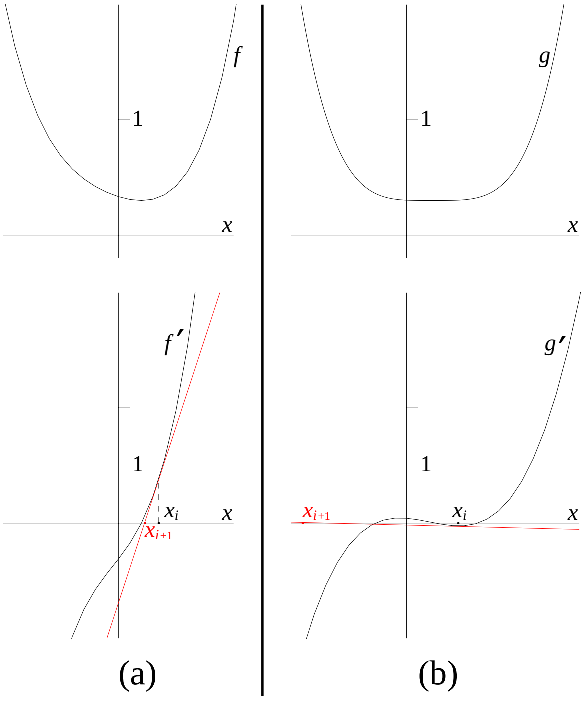


Figure 2.6: Newton's method (2.18) robustly converges a minimum of $f = 0.13 + 0.69\,x + 0.64x^2 + 0.2x^3 + x^4$, but for $g = 0.3 - 0.048\,x + 0.31x^2 - 0.9x^3 + x^4$ its successive solutions move far from any minimum. Drastic step size limits can reduce jumping about, but this reduces the method to a slow gradient-descent Euler method.

Where the lower terms vanish the higher terms 'generically' do not (simultaneous vanishing would simultaneously satisfy too many equations in too few state unknowns $(x_1, \ldots, x_n)$ and control unknowns $(X, Y, \ldots)$).We use the bifurcation theory 'k-determinacy' test (Appendix A) for the order k of terms needed, numerically replacing 'not zero' by 'not small', and solve for a zero of the derivative of the corresponding higher Taylor polynomial instead of the quadratic one. For the Zeeman machine (Fig. 2.7a) we iterate to find equilibria, as in the Newton iteration of linear solving; for the standard cusp (Fig. 2.7b) the polynomial solution is by definition exact. In both cases we get dynamically appropriate behavior and haptic forces, smooth except where a fold curve forces a jump.

2.4 Multi-variable states

In a single variable the k-determinacy test is simply 'first x^k not vanishing', but for examples like (2.11) it is more subtle. Moreover, for a 20-variable state like (2.2)

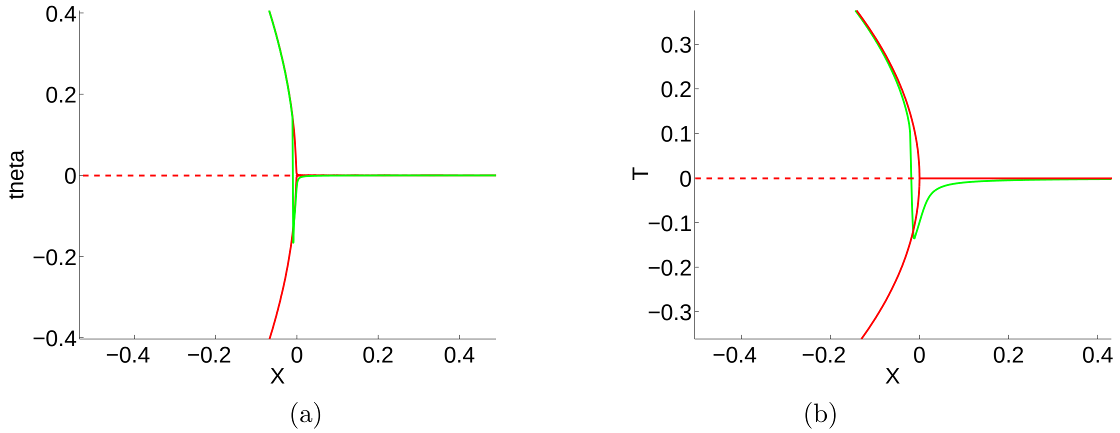


Figure 2.7: Higher-order computed equilibria (green) for (a) the Zeeman machine and (b) standard cusp, as hand position moves along a line very near the plane of symmetry of the equilibrium set found analytically (red). Position of hand is represented by (X,Y).

with 10 free nodes, it is impractical to solve a cubic problem with 10,395 terms, as a full quartic expansion of the energy would require. We therefore exploit the Splitting Lemma (see Appendix B). This guarantees that around a point $(\overline{a}_1, \ldots, \overline{a}_c, \overline{x}_1, \ldots, \overline{x}_n)$ where a minimum of a function $f_{(a_1,\ldots,a_c)}(x_1, \ldots, x_n)$ of n internal variables with c controls has a Hessian matrix $\left[\partial^2 f_{(a_1,\ldots,a_c)}/\partial x_i \partial x_j\right]$ of rank $n-q$, there is locally an $(a_1, \ldots, a_c)$-dependent reparametrization of $(x_1, \ldots, x_n)$ as $(\widehat{x}_1, \ldots, \widehat{x}_n)$ to give it the form

$$\begin{aligned} & C(a_1, \ldots, a_c) + (\widehat{x}_1)^2 + \ldots + (\widehat{x}_{n-q})^2 \\ & \quad + \widehat{f}_{(a_1,\ldots,a_c)}(\widehat{x}_{n-q+1}, \ldots, \widehat{x}_n), \end{aligned} \tag{2.19}$$

with the quadratic expansion of $\widehat{f}_{(\overline{a}_1,\ldots,\overline{a}_c)}$ around $(\overline{x}_1, \ldots, \overline{x}_n)$ exactly zero, as are all higher terms in $(\widehat{x}_1, \ldots \widehat{x}_{n-q})$. Moreover, for the 'corank' q to stably occur we must satisfy $q(q+1)$ equations in the $\partial^2 f_{(a_1,\ldots,a_c)}/\partial x_i \partial x_j$ as well as the n equations $\partial f_{(a_1,\ldots,a_c)}/\partial x_1 = \ldots = \partial f_{(a_1,\ldots,a_c)}/\partial x_1 = 0$, which needs $c \geq q(q+1)/2$ additional unknowns $(a_1, \ldots, a_c)$. Corank $q = 2$, needing three unknowns, cannot stably occur with a 2D-control system like (2.2). Corank $q = 3$ requires six degrees of freedom in control, and so on, independently of n as long as it is finite[2]. We thus

[2] Where n is better handled as infinite, ([17]) illustrates that splitting off a low-dimensional bifurcation variable may stably fail. However, it can also stably be possible, and often is.

expect bifurcations of minima of (2.2) to be reducible to

$$C\left(X,Y\right)+\left(\widehat{x}_1\right)^2+\ldots+\left(\widehat{x}_{19}\right)^2+\widehat{f}_{(X,Y)}\left(\widehat{x}_{20}\right), \tag{2.20}$$

as is the case here since the stiffness matrix has only one near-zero eigenvalue. To use this numerically we track the condition of $\left[\partial^2 f_{(a_1,\ldots,a_c)}/\partial x_i \partial x_j\right]$. When its eigenvalues $\lambda_1,\ldots,\lambda_n$ are well separated from 0, we use the quadratic/linear Newton method, as in [16]. When one or more λ_i approach zero, we split $(x_1,\ldots,x_n)$-space into a sum E_δ of their eigenspaces and a complement $E_\delta^{\intercal}$ to it. The orthogonal complement $\left(E_\delta\right)^\perp$ can robustly be used for this, but one can with less numerical effort use whatever subspace defined by restricting q of the x_i to zero is most orthogonal to E_δ, which provides coordinates $(\widetilde{x}_1,\ldots,\widetilde{x}_{n-q})$ and already-computed $\partial^2 f/\partial\widetilde{x}_i\partial\widetilde{x}_j$. The eigenvectors of the λ_i give a basis and thus coordinates for E_δ. We solve by the standard method in $E_i^\perp$ directions and by higher polynomial approximations along E_i. When $q=1$, this is the same polynomial solution process as for the single-internal-variable Zeeman machine.

2.4.1 Curved separation of quadratic and degenerate directions

We, however, use higher derivatives in more than the straight λ_i eigendirection as it is not quite sufficient to take derivatives in that direction only. The example

$$x^2-4xy^2+3y^4 \tag{2.21}$$

has quadratic approximation simply x^2, degenerate along the y-axis L. Along this line the function reduces to $3y^4$, which suggests a strict (though degenerate) minimum at $(0,0)$. An approach to energy-minimisation that used $3y^4$ as the approximation in this direction would find (when the function is slightly perturbed) a candidate minimum near $(0,0)$, so no large jumps would occur. However, rewriting the example as

$$\left(x - 2y^2\right)^2 - y^4, \tag{2.22}$$

making clear that it has a strict *maximum* in the degenerate direction, and energy minimisation should carry us far away. (Higher terms can introduce new equilibria to jump to, at a distance, but cannot affect the local topology of this example.) If we change to coordinates $(u, v) = (x - 2y^2, y)$ the function becomes exactly $u^2 - v^4$, and it is clear that any minimum points must lie on the v-axis $u = 0$, which in the original coordinates is the curve $x = 2y^2$.

2.4.1.1 Two internal variables

Before treating general n, we discuss the needed computation for two internal variables. Without loss of generality we can choose the origin at the degenerate equilibrium we are interested in, which generically has only one degenerate direction. We linearly choose coordinates to make that direction the y-axis. In this form we have

$$\begin{bmatrix} \frac{\partial f}{\partial x}(0,0) & \frac{\partial f}{\partial y}(0,0) \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{2.23}$$

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x^2}(0,0) & \frac{\partial^2 f}{\partial x \partial y}(0,0) \\ \frac{\partial^2 f}{\partial x \partial y}(0,0) & \frac{\partial^2 f}{\partial y^2}(0,0) \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.24}$$

for $\alpha \neq 0$. Any nearby equilibria are necessarily on the curve C (Fig. 2.8) defined by

$$\frac{\partial f}{\partial x}(x, y) = 0, \tag{2.25}$$

which exists and is tangent to the y-axis by the Implicit Function Theorem. We can parametrize C as

$$c(s) = (X(s), s), \text{ with } X(0) = \frac{dX}{ds}(0) = 0 \tag{2.26}$$

Moreover, since the non-degeneracy condition $\frac{\partial^2 f}{\partial x^2}(0,0) \neq 0$ must still hold for small perturbations of f, the curve C persists and moves around smoothly with

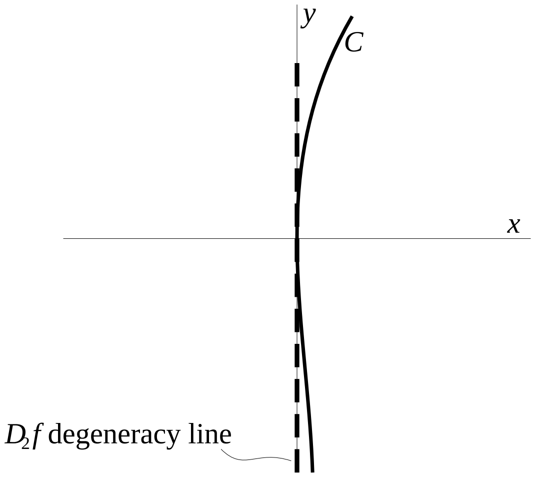


Figure 2.8: Degeneracy direction and the curve on which equilibria lie.

any smooth control parameter a.

To approximate *where* on C the equilibria are, we approximate the restricted function $\widehat{f} = f \mid C$ by more than the vanishing quadratic terms along C, and hence must find some higher derivatives along C: generically, derivatives up to fourth order will suffice for ‘elementary catastrophe’ bifurcations if a is only 2-dimensional. These are not in general the same as the derivatives along the straight line tangent to it, in this example the y-axis. From (2.26) we have the expansion

$$X\left(s\right) = \xi_2 s^2 + \xi_3 s^3 + \xi_4 s^4 + O(5), \tag{2.27}$$

with vanishing constant and linear terms. We substitute this into the expansion of $f\left(x, y\right)$,

$p_{xx}x^2 + (p_{xxx}x^3 + p_{xxy}x^2y + p_{xyy}xy^2 + p_{yyy}y^3) +$
$(p_{xxxx}x^4 + p_{xxxy}x^3y + p_{xxyy}x^2y^2 + p_{xyyy}xy^3 + p_{yyyy}y^4) + \ldots$

where

$$p_{xx} = \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\left(0,0\right) = \frac{\alpha}{2}, \tag{2.28}$$

$$p_{xxx} = \frac{1}{6}\frac{\partial^3 f}{\partial x^3}\left(0,0\right), \quad p_{xxy} = \frac{1}{2}\frac{\partial^3 f}{\partial x^2 \partial y}\left(0,0\right), \quad p_{xyy} = \frac{1}{2}\frac{\partial^3 f}{\partial x \partial y^2}\left(0,0\right)$$

$$p_{yyy} = \frac{1}{6}\frac{\partial^3 f}{\partial y^3}\left(0,0\right), \quad p_{xxxx} = \frac{1}{24}\frac{\partial^4 f}{\partial x^4}\left(0,0\right), \quad p_{xxy} = \frac{1}{6}\frac{\partial^4 f}{\partial x^3 \partial y}\left(0,0\right)$$

$$p_{xyy} = \frac{1}{4}\frac{\partial^4 f}{\partial x^2 \partial y^2}\left(0,0\right), \quad p_{yyy} = \frac{1}{6}\frac{\partial^4 f}{\partial x \partial y^3}\left(0,0\right), \quad p_{yyyy} = \frac{1}{24}\frac{\partial^4 f}{\partial x \partial y^3}\left(0,0\right)$$

to get

$$\widehat{f} = p_{yyy}s^3 + \left(p_{yyyy} + \xi_2^2 p_{xx} + \xi_2 p_{xyy}\right) s^4 + O(5),$$

which shows us that only the coefficient ξ_2 in (2.27) is needed here. To find it, we substitute (2.27) into the defining equation (2.25) of C, using the expansion (2.28). We have, for all small s,

$$\begin{aligned} 0 = \frac{\partial f}{\partial x} &= 2p_{xx}x + \left(3p_{xxx}x^2 + 2p_{xxy}xy + p_{xyy}y^2\right) + \\ &\left(4p_{xxxx}x^3 + 3p_{xxxy}x^2y + 2p_{xxyy}xy^2 + p_{xyyy}y^3\right) + \dots \\ 0 &= \left(p_{xyy} + 2\xi_2 p_{xx}\right) s^2 + \left(p_{xyyy} + 2\xi_3 p_{xx} + 2\xi_2 p_{xxy}\right) s^3 + \dots \end{aligned} \quad (2.29)$$

Equating coefficients,

$$p_{xyy} + 2\xi_2 p_{xx} = 0 \quad (2.30)$$

$$\xi_2 = -\frac{p_{xyy}}{2p_{xx}}\,, \quad (2.31)$$

so that the energy along the curve C can be written as

$$\begin{aligned} \widehat{f} &= p_{yyy}s^3 + \left(p_{yyyy} + \left(-\frac{p_{xyy}}{2p_{xx}}\right)^2 p_{xx} + \left(-\frac{p_{xyy}}{2p_{xx}}\right) p_{xyy}\right) s^4 + O(5) \\ &= p_{yyy}s^3 + \left(p_{yyyy} - \frac{p_{xyy}^2}{4p_{xx}}\right) s^4 + O(5) \end{aligned} \quad (2.32)$$

In particular, for the example (2.21) above,$x^2 - 4xy^2 + 3y^4$, we have

$$p_{xx} = 1, \qquad p_{xyy} = -4, \qquad p_{yyyy} = 3$$

and all other coefficients zero. This gives to fourth order

$$\widehat{f} = 0s^3 + \left(3 - \frac{(-4)^2}{4}\right) s^4 = -s^4, \quad (2.33)$$

revealing the local maximum along C. Without a need for us to spot the rearrangement as (2.22), we have found the fourth-order coefficient -1 that it reveals along the degeneracy curve.

2.4.1.2 Non-axis degeneracy

Any two-variable quadratic expression that is degenerate in one direction can be diagonalized to the form (2.24) by a linear change of variables, but we used that form above for expository clarity, not for computational necessity. Numerically, it is better to avoid it.

Consider an unreduced case

$$\begin{bmatrix} \frac{\partial g}{\partial x}(0,0) & \frac{\partial g}{\partial y}(0,0) \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{2.34}$$

$$\begin{bmatrix} \frac{\partial^2 g}{\partial x^2}(0,0) & \frac{\partial^2 g}{\partial x \partial y}(0,0) \\ \frac{\partial^2 g}{\partial x \partial y}(0,0) & \frac{\partial^2 g}{\partial y^2}(0,0) \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}, \tag{2.35}$$

where $\alpha\gamma - \beta^2 = 0$. This is degenerate in the direction of the vector $\mathbf{v} = (\beta, -\alpha)$:

$$\begin{bmatrix} s\beta \\ -s\alpha \end{bmatrix}^T \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} s\beta \\ -s\alpha \end{bmatrix} = \alpha\left(\alpha\gamma - \beta^2\right) s^2 \equiv 0 \tag{2.36}$$

We assume $|\alpha| \geq |\beta|$, so that $\mathbf{v}$ cannot lie along the x-axis. (If $|\beta| \geq |\alpha|$, switch the names of x and y.) A linear coordinate change that would diagonalise this case is to set $\mathbf{u} = (1,0)$, the unit x-axis vector, and choose $(\mathbf{u}, \mathbf{v})$ as a basis. This simply replaces $p_{xyy} = \frac{1}{2}\frac{\partial^3 f}{\partial x \partial y^2}(0,0)$ above by the second derivative

$$\frac{1}{2}\frac{d^2}{ds^2}\left(\frac{\partial g}{\partial x}(s\mathbf{v})\right), \tag{2.37}$$

which is straightforward to find numerically.

2.4.1.3 Multiple internal variables

Now consider the general one-dimension-of-degeneracy case, still choosing the point of interest as origin.

$$\begin{bmatrix} \frac{\partial g}{\partial x_1}(0,0) & \dots & \frac{\partial g}{\partial x_n}(0,0) \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}, \tag{2.38}$$

$$\begin{bmatrix} sv_1 \\ \vdots \\ sv_n \end{bmatrix}^T \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2}(0,\dots,0) & \dots & \frac{\partial^2 g}{\partial x_1 \partial x_n}(0,\dots,0) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 g}{\partial x_1 \partial x_n}(0,\dots,0) & \dots & \frac{\partial^2 g}{\partial x_n^2}(0,\dots,0) \end{bmatrix} \begin{bmatrix} sv_1 \\ \vdots \\ sv_n \end{bmatrix} \equiv 0 \tag{2.39}$$

for some non-zero $\mathbf{v} = (v_1, \dots, v_n)$. We write the Taylor expansion to order 4 of g as

$$\gamma(x_1, \dots, x_n) = \sum_{\substack{i=1,\dots,n \\ j=1,\dots,n}} p_{ij} x_i x_j + \sum_{\substack{i=1,\dots,n \\ j=1,\dots,n \\ k=1,\dots,n}} p_{ijk} x_i x_j x_k + \sum_{\substack{i=1,\dots,n \\ j=1,\dots,n \\ k=1,\dots,n \\ l=1,\dots,n}} p_{ijkl} x_i x_j x_k x_l\,, \tag{2.40}$$

where

$$p_{ij} = \frac{1}{2}\frac{\partial^2 g}{\partial x_i \partial x_j}(0,\dots,0)\,, \tag{2.41}$$

$$p_{ijk} = \frac{1}{6}\frac{\partial^3 g}{\partial x_i \partial x_j \partial x_k}(0,\dots,0)\,, \tag{2.42}$$

$$p_{ijkl} = \frac{1}{24}\frac{\partial^4 g}{\partial x_i \partial x_j \partial x_k \partial x_l}(0,\dots,0)\,. \tag{2.43}$$

Note that the p_{ijk} *etc.*, do not exactly correspond to the p_{xxy} and so on of the two-variable case above, since the sums are more simply expressed with repetitions. (For example, $x_1^2 x_2$ occurs with the three (equal) coefficients p_{112}, p_{121} and p_{211}, and similarly others.)

We define the curve C by requiring equilibrium in a hyperplane H of directions transverse to $\mathbf{v}$ (see Appendix C for definition on transverse). Generically this is true for all possible choices of $n-1$ axes (the typical line is transverse to both standard axes in (x,y)-space, to the (x,y), (y,z) and (x,z) planes in (x,y,z)-space, and so on), but we make certain of this and make it numerically robust by assuming that v_1 is the smallest-modulus component of $\mathbf{v}$ and v_n the largest. We rescale $\mathbf{v}$ to arrange $v_n = 1$, and parametrize the line L:

$$l(s) = s\mathbf{v} = (v_1 s, v_2 s, v_3 s, \ldots, s) \tag{2.44}$$

As in the the two-variable case, only the second-order expansion of the curve C is needed. Cubic terms in s, substituted in (2.40), give terms order 5 and higher in s, which are outside the expansion we need. We parametrise C (tangent to L) as

$$c(s) = s\mathbf{v}+s^2\mathbf{w}+O(3) = \big(v_1 s + w_1 s^2 + O(3), v_2 s + w_2 s^2 + O(3),, \ldots, s\big), \tag{2.45}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_{n-1}, 0) \in H$ is to be determined. The n^{th} component of $s\mathbf{v} + s^2\mathbf{w}$ is simply s, as in (2.26). We have

$$\frac{\partial \gamma}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\sum_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}} p_{ij}x_i x_j + \sum_{\substack{i=1,\ldots,n\\ j=1,\ldots,n\\ k=1,\ldots,n}} p_{ijk}x_i x_j x_k + \sum_{\substack{i=1,\ldots,n\\ j=1,\ldots,n\\ k=1,\ldots,n\\ l=1,\ldots,n}} p_{ijkl}x_i x_j x_k x_l \right) \tag{2.46}$$

$$= 2\sum_{j=1,\ldots,n} p_{ij}x_j + 3\sum_{\substack{j=1,\ldots,n\\ k=1,\ldots,n}} p_{ijk}x_j x_k + 4\sum_{\substack{j=1,\ldots,n\\ k=1,\ldots,n\\ l=1,\ldots,n}} p_{ijkl}x_j x_k x_l \tag{2.47}$$

On C, to fourth order in s this gives

$$2\sum_{j=1,\ldots,n} p_{ij}\left(sv_j+s^2w_j\right)+3\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}\left(sv_j+s^2w_j\right)\left(sv_k+s^2w_k\right)+$$

$$4\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n\\l=1,\ldots,n}} p_{ijkl}\left(sv_j+s^2w_j\right)\left(sv_k+s^2w_k\right)\left(sv_l+s^2w_l\right) \tag{2.48}$$

$$=\delta_1 s+\delta_2 s^2+\delta_3 s^3+\delta_4 s^4+O(5),\ \text{where} \tag{2.49}$$

$$\delta_1=2\sum_{j=1,\ldots,n} p_{ij}v_j$$

$$\delta_2=2\sum_{j=1,\ldots,n} p_{ij}w_j+3\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}v_jv_k$$

$$\delta_3=3\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}\left(v_jw_k+v_kw_j\right)+4\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n\\l=1,\ldots,n}} p_{ijkl}v_jv_kv_l$$

$$\delta_3=3\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}w_jw_k+4\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n\\l=1,\ldots,n}} p_{ijkl}\left(v_jv_kw_l+v_jv_lw_k+v_kv_lw_j\right)$$

We require that this vanishes identically for every i. The linear term $\delta_1 s$ vanishes by construction of $\mathbf{v}$ as a solution of (2.39). Equating quadratic coefficients gives us

$$0=2\sum_{j=1,\ldots,n} p_{ij}w_j+3\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}v_jv_k \tag{2.50}$$

$$2\begin{bmatrix} p_{11} & \cdots & p_{1n}\\ \vdots & \ddots & \vdots\\ & & \\ p_{1n} & \cdots & p_{nn}\end{bmatrix}\begin{bmatrix} w_1\\ \vdots\\ w_{n-1}\\ 0\end{bmatrix}=-3\begin{bmatrix}\sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{1jk}v_jv_k\\ \vdots\\ \sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{njk}v_jv_k\end{bmatrix} \tag{2.51}$$

analogous to (2.30). The Hessian matrix $P=[p_{ij}]$ has no inverse, but there is a unique solution $\mathbf{w}$ to

$$2\begin{bmatrix} p_{11} & \cdots & p_{1(n-1)} \\ \vdots & \ddots & \vdots \\ p_{1(n-1)} & \cdots & p_{(n-1)(n-1)} \end{bmatrix}\begin{bmatrix} w_1 \\ \vdots \\ w_{n-1} \end{bmatrix} = -3\begin{bmatrix} \sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{1jk}v_jv_k \\ \vdots \\ \sum_{\substack{j=1,\ldots,n\\k=1,\ldots,n}} p_{(n-1)jk}v_jv_k \end{bmatrix} \tag{2.52}$$

because this matrix gives the restriction of the Hessian to the hyperplane H, which does not contain the degeneracy direction $\mathbf{v}$ of P, and is thus invertible on H. Cramer's rule would give us an explicit form for $\mathbf{w}$, which we could substitute into (2.40) in analogy with (2.32), but the resulting closed formula is overcomplicated. In this case it is better to solve the (usually sparse) problem (2.52) numerically, and substitute the resulting $\mathbf{w}$ numerically into (2.40). We have

$$\begin{aligned}\widehat{g}(s) = &\sum_{\substack{i=1,\ldots,n\\j=1,\ldots,n}} p_{ij}\left(sv_i+s^2w_i\right)\left(sv_j+s^2w_j\right) \\ &+\sum_{\substack{i=1,\ldots,n\\j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}\left(sv_i+s^2w_i\right)\left(sv_j+s^2w_j\right)\left(sv_k+s^2w_k\right) \\ &+\sum_{\substack{i=1,\ldots,n\\j=1,\ldots,n\\k=1,\ldots,n\\l=1,\ldots,n}} p_{ijkl}\left(sv_i+s^2w_i\right)\left(sv_j+s^2w_j\right)\left(sv_k+s^2w_k\right)\left(sv_l+s^2w_l\right)\end{aligned} \tag{2.53}$$

$$= \left(\sum_{\substack{i=1,\ldots,n\\j=1,\ldots,n}} p_{ij}v_iv_j\right)s^2 + \left(\sum_{\substack{i=1,\ldots,n\\j=1,\ldots,n}} p_{ij}\left(v_iw_j+v_jw_i\right) + \sum_{\substack{i=1,\ldots,n\\j=1,\ldots,n\\k=1,\ldots,n}} p_{ijk}v_iv_jv_k\right)s^3+$$

$$+ \left(\sum_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}} p_{ij} w_i w_j + \sum_{\substack{i=1,\ldots,n\\ j=1,\ldots,n\\ k=1,\ldots,n}} p_{ijk} \left(v_i v_j w_k + v_i v_k w_j + v_j v_k w_i \right) \right.$$

$$\left. + \sum_{\substack{i=1,\ldots,n\\ j=1,\ldots,n\\ k=1,\ldots,n\\ l=1,\ldots,n}} p_{ijkl} v_i v_j v_k v_l \right) s^4 \tag{2.54}$$

$$= 0s^2 + \gamma_3 s^3 + \gamma_4 s^4 \text{ for short.} \tag{2.55}$$

Note that the cubic coefficient γ_3 is not given simply by the third derivative in the $\mathbf{v}$ direction, since $\mathbf{w}$ contributes to it.

2.4.2 Around a bifurcation point

In numerical work, we detect that we are effectively near an equilibrium by establishing that the derivative of the energy is small. We then detect whether we are near a point where the above calculations apply by testing whether the Hessian is close to degenerate (by its small determinant). We find $\mathbf{v}$ that belongs to the near-zero eigenvalue, and follow the analysis above to give us the expansion (2.54). The resulting $\gamma_3 s^3 + \gamma_4 s^4$ gives a local 'background' non-quadraticity in the $\mathbf{v}$ direction.

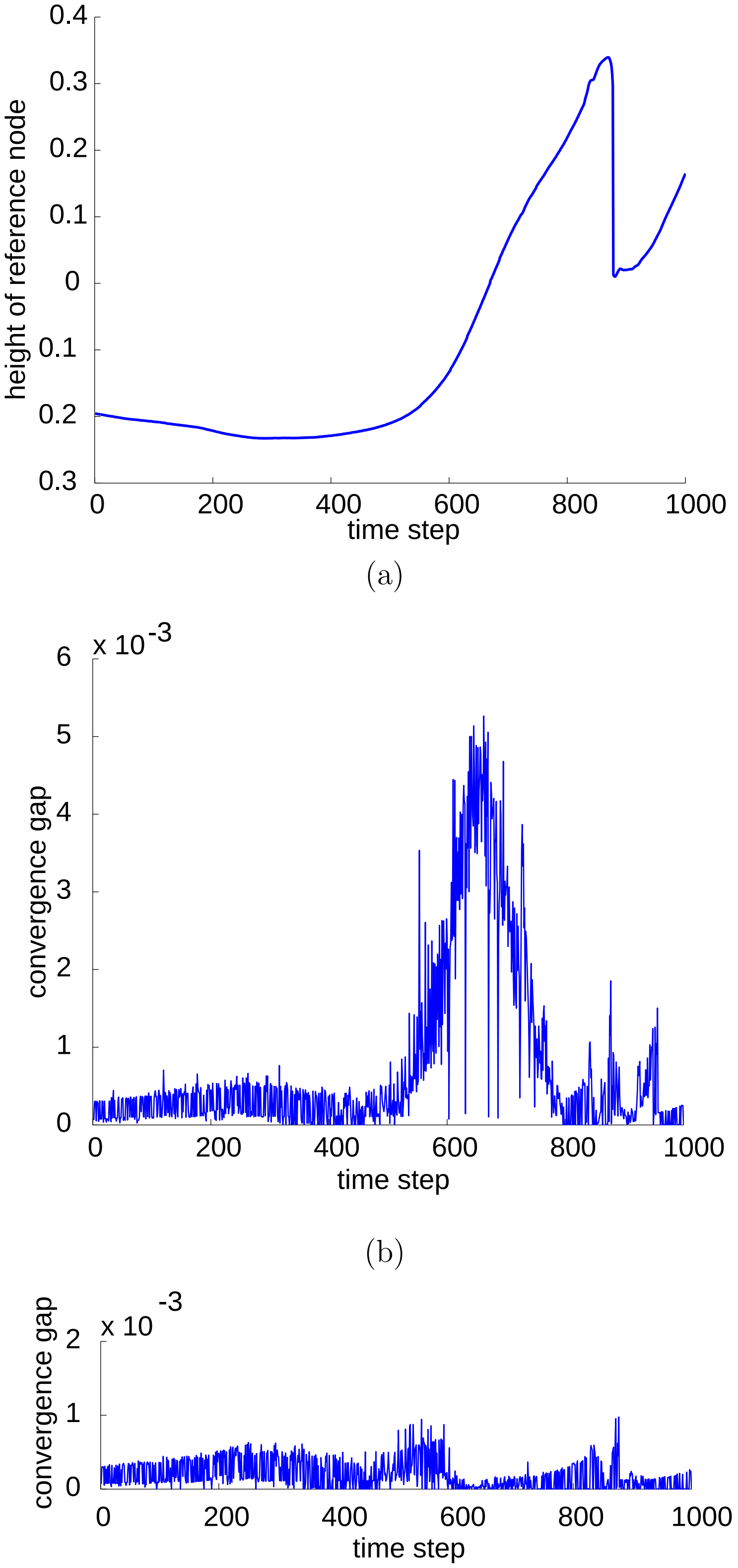
We then add the estimated first and second derivatives in the $\mathbf{v}$ direction, which are small relative to the 'background' coefficients and do change fast relative to their current scale (for example, passing through 0 and so changing sign) to give us a non-linear equilibrium problem in s, using single-variable quadratic or cubic polynomial methods to solve it.

2.5 Experiments and Results

We have simulated 2D curve dynamics using the ideas in § 2.3.1 and § 2.3.2. The user moves an end of the 'thread' by a haptic controller, implemented both with a Delta haptic interface [18] and a desktop PHANToM [19], on a Dell workstation with Dual 2.6 XG Xeon processors.

Unwanted vibrations (§2.3.5) led us to a systematic study of bifurcating systems, using simulations instead of haptic device implementations to eliminate instability sources such as delay and motor noise. For computational clarity, we first studied single-parameter systems such as the Zeeman Machine and Standard Cusp described in § 2.3.3. We developed a virtual quasi-static haptic Zeeman Machine, calculating forces by changes in minima of the energy in (2.4), as explained in §2.3.4. In two sets of simulations, we found energy minima by iterative quadratic fitting, and also by higher order Taylor terms (requiring more derivatives). In experiments we moved the grasped 'control point' across the fold or through the cusp point, and similarly for the Standard Cusp simulation. Typical results for both sets (Figs. 2.5b, 2.7a for the Zeeman Machine, Figs. 2.5c, 2.7b for the standard cusp) clearly show such higher order terms eliminating vibrations.

Our iterative search for equilibrium (quadratic Newton at non-degenerate points, higher order near bifurcations) stops when the computed internal force is below 10^{-6}N, allowing haptic speeds. Around a non-bifurcating equilibrium the expansion of this force is dominated by its linear term, making accuracy comparable in force-vanishing and in equilibrium-location. At a bifurcation larger location errors can occur with near-zero force, precisely because the linear relation is degenerate. Fig. 2.9 shows (a) a sequence of computed equilibria, including a snap, (b) the differences between these equilibria and those found more leisurely with a 10^{-10}N cutoff, and (c) those with 10^{-8}N and 10^{-10}N cutoffs. The small differences in (c) indicate that both these cutoffs give near-true values, so that (b) is a good indicator of error levels at practical speed. Since the haptic force at each user-constrained equilibrium $\mathbf{x} = (x_1, \ldots, x_n)$ follows from $\mathbf{x}$ and from analyti-

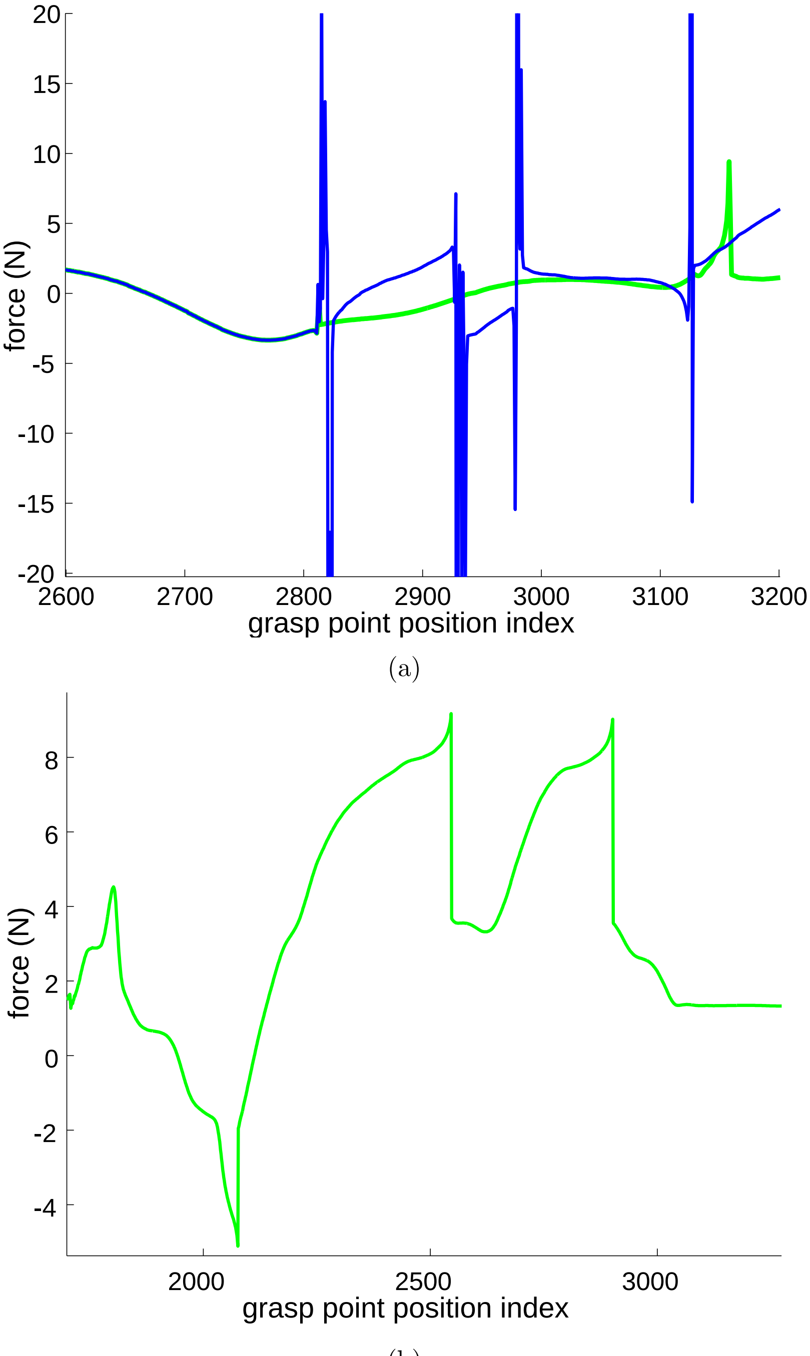


(a)

(b)

(c)

Figure 2.9: Typical time course (a) of equilibria computed with a 10^{-6}N convergence cutoff, with error estimated (b) by differences from 10^{-10}N cut-off values, which differ only as in (c) from 10^{-8}N cut-off values.



(a)

(b)

Figure 2.10: Force profiles while manipulating the 2D curve's grasp point, computed with the quadratic/linear Newton's method (blue) and the bifurcation-algebra-based method introduced in this paper (green). Figs. (a) and (b) show responses to hand manipulation (using the Delta haptic device). Fig. (a, blue) is live, while green curve is the chatter-free force profile found with the scheme described in this paper, for the same grasp point history. Fig. (b) shows the force profile of a separate live Delta manipulation, with chatter-free snap-jumps, found by the scheme described in this paper.

cally known derivatives, given the quasi-static assumption, the forces are similarly accurate.

With a multi-parameter energy function such as that of a curve, it is too time consuming for real-time display to use all derivatives to this higher order. We developed the scheme in §2.4 for an effective solution process, much faster as it requires far fewer derivatives. In similar experiments, we moved the grasped end across bifurcation points, numerically and via a haptic device. Fig. 2.10a (blue curve) shows the results with iterative quadratic Taylor fit for energy and Fig. 2.10a (green curve) shows the results with our scheme, which completely eliminates vibrations. Fig. 2.1i also shows the implementation of our scheme on a 2D curve (or asymmetrical 3D thread moving in its preferred bending plane) when it meets fold points (Fig. 2.1a–h). The difference between Figs. 2.1j and 2.1i shows the benefits of our scheme. Finally Fig. 2.10b clearly illustrates vibration free haptic experience of the palpable 2D curve resulting from our algorithms.

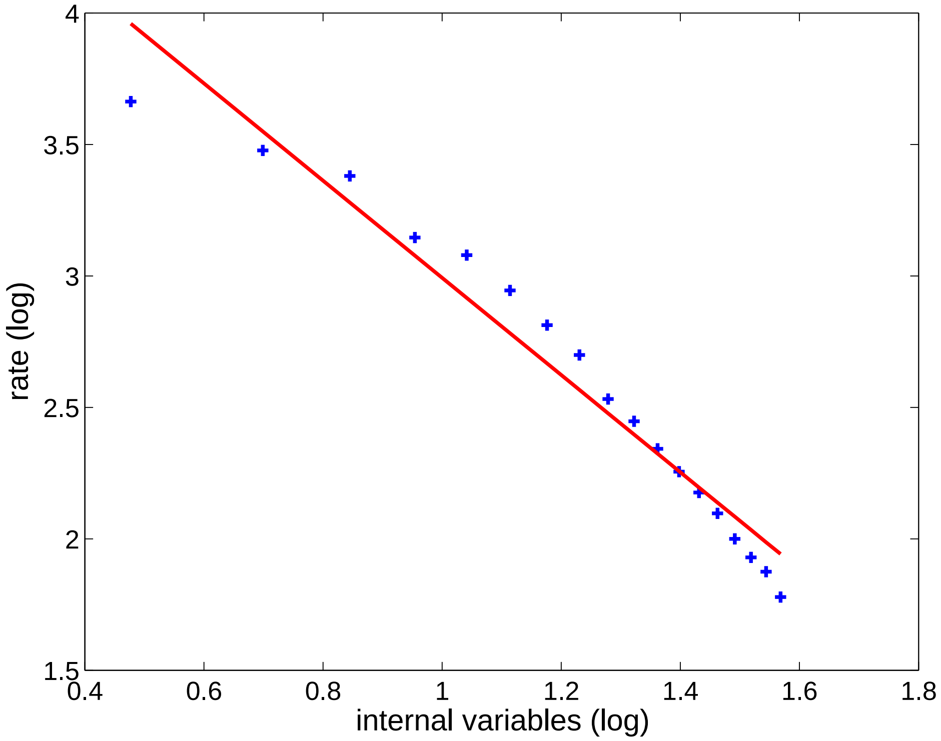


Figure 2.11: Measured numerical refresh rate as a function of the number n of internal variables (free node coordinates), on a log-log scale. The straight line shows the best-fit power law $n^{-1.9}$, but a better CPU cost model is needed.

2.6 Discussion

The haptic wall shows that delay can give improper vibration: [8] shows how to manage it. This instability would vanish when using a perfectly stiff interface with negligible feedback delay. Buckling and jumps cause improper vibration quite differently, by the failure of the standard Newton's Method. Bifurcation theory is the tool to manage this.

The method developed here is, to the best of our knowledge, the first to compute haptic forces around snap discontinuities: no previous work for comparison is available. It operates at haptic speeds, giving appropriate forces for display. This paper applied it both in the paradigmatic Zeeman machine and in a model of thread, whose characteristic jumps have been neglected by the haptics and simulation communities.

Our simulations showed that bifurcation theory provides appropriate approximations in a numerically cost-efficient way, limiting the use of variables and higher derivatives to exactly those required. As a function of the number n of free internal variables (node coordinates) x_i and y_i, the computational load grows comparably with the Newton quadratic fit method, since it uses the stiffness matrix to identify degenerate directions and then does $O(n)$ computations in those directions, rather than handling cubic problems with $O(n^4)$ coefficients. Its speed is thus similar to that of the standard snap-incapable Newton approach, making it usable on models of comparable scale. A general stiffness matrix is up to $O(n^2)$ in its entry count, though a thread's (organized by its sequential topology, unlike a membrane's) is sparse and near-diagonal. Fig. 2.11 shows our achieved rate versus n, on a log-log scale. The fitted line has a slope of -1.9, but it is clear that an $n^{-1.9}$ power law is over-optimistic for large n. More work to identify the growth bottlenecks would be useful.

Bifurcation-theory-based numerics open the door to a wider range of practical simulations, and to education on critical aspects of nonlinear dynamics, where students should literally 'get a feel for' bifurcating systems. A general theorem, and a theory of interaction with the user's dynamics, remain for further work.

In next chapter, we develop the 2D curve dynamics further into a palpable 3D nylon thread. The issue of bifurcations comes up and is handled efficiently, using the scheme developed on this chapter.

Chapter 3

Palpable Virtual Nylon Thread

3.1 Introduction

Suturing is a critical part of surgery, where surgeons suture vessels *1mm* across. It is a particularly skilled task. Fine manipulation under an optical microscope is different from the macro world and requires significant training. Surgeons use nylon thread to create sutures, first passing a needle through vessels or other tissues and then tying knots. Knot tying is difficult when the thread must be handled indirectly with forceps that are seen via a microscope or a laparoscopic camera, where the visible direction of motion and rotation are turned form those felt by the users' hands. The dynamics of nylon thread is non-linear and is an example of large flexible deformation, unlike feeling a surface as hard of soft, that cannot be treated by a computed response using only properties near the user's contact point.

Large deformation is in essence of the knotting task. Where a suture needle creates a track that the suture thread must follow, as in [1], it is enough to define a curve by pulling a needle through it. However, a key step in surgical knotting is to grasp one thread with the left forceps and create a loop through which the right forceps grasps the other (Fig. 1.1). The first grasp taken far to the left, pushes the grasped point rightwards into untraced space. Nylon mechanics forms the loop, to the right of any point where the forceps tip has gone: a 'following' simulation does not match the task. How to simulate these large deformations? To our knowledge no work has been done on global deformation of a flexible nylon thread - including visualizing and feeling such topologically essential non-linearities as ‘snap through’ jumps. With these jumps, comes the issue of handling bifurcations and applying the scheme developed in previous chapter, evaluating its effectiveness. This chapter deals with dynamically realistic thread simulation, which is important for learning the skills required for knot tying. We model the nylon thread as a geometric curve and associate energy with it. During thread manipulation, we find an equilibrium state.

3.2 Literature review

Several schemes have been used to model a rope or nylon for simulating knot tying. In [1] Joel Brown proposed a geometric approach to realize real time knot tying, in which the rope is modelled as a series of nodes linked together. The grasped node (the leader) is followed by rest of the nodes (followers) such that the internodal length remains constant. This follow-the-leader approach leads to a visually nice but (even visually) incorrect knot-tying simulation, because the thread’s mechanics, which play an important role in loop formation (Fig. 1.1) are not considered. Thread simulation based on dynamics was presented in [20], in which the authors simulate knot tying with a spline of linear springs and control points at their ends. The focus was on handling self-collisions when the knot is being pulled and tied, while conserving energy, mass and momentum of the system. Conservation of energy may create problems, because real thread dynamics are dissipative. A dynamic approach was recently developed in our lab [3], [4] in which the nylon thread is modelled as a series of masses and springs. Springs implement stretching, bending and twisting. When the grasped node is perturbed, the nodes move in direction of resultant force at each node until the force is zero. Real time simulation is achieved with this approach, with realistic effects such as twisting realized, however essential non-linearities in the mechanics of thread such as snaps

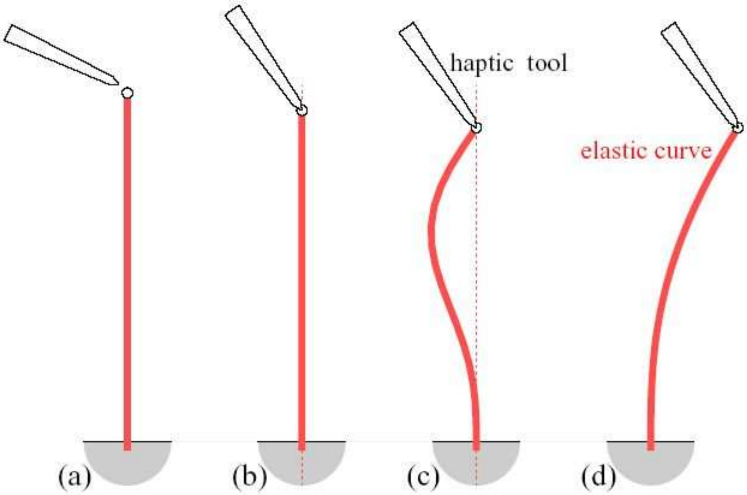


Figure 3.1: Press down a little on the compressible upright (a) and it shortens (b). Press further and it bends to one side or the other: move to the other side and the bend jumps across (d). We want to feel such phenomena.

have not been dealt with. The details of loop formation involve minimization of stretching, bending and twisting energies, and we use precisely this approach to model thread dynamics, though this thesis includes only bending and stretching. We model these energies in a computationally simple way and find the equilibria. We introduce a 3D thread model on which we apply this technique. Multiple equilibria lead to snap-through jumps which is demonstrated by our nylon thread model as in Fig. 1.3B.

3.3 Development of nylon thread dynamics

We develop model of a flexible 3D curve as a paradigm for development of a virtual nylon thread. We are more concerned here with matter like feeling the way a 3D curve like Fig. 3.1 moves and jumps as the user controls an end point, and similar experiments. The user would hold the tip by a freely rotating grip, or a fixed grip, where the curve end is force to a particular angle. We first explain the curve modelled with stretching and bending and then explain modelling of twist.

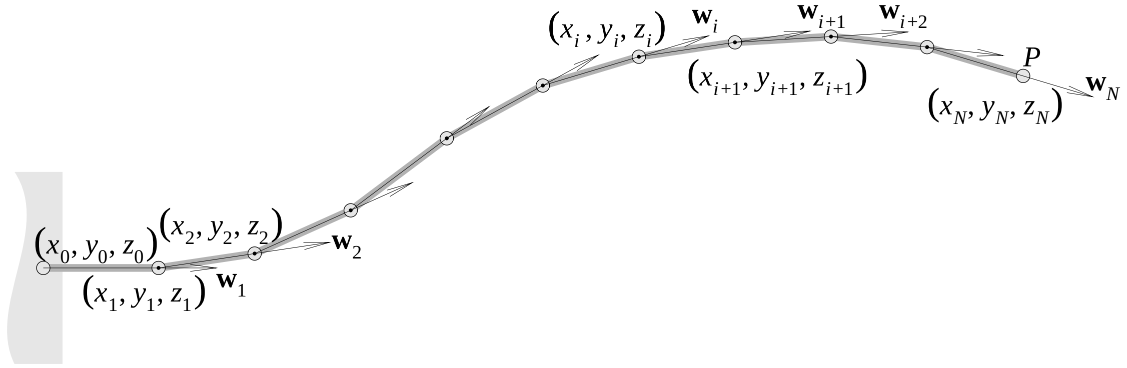


Figure 3.2: Three dimensional curve model, a simple extension of the two-dimensional model.

3.3.1 Geometric Descriptors

We use a reference interval $\Pi = [P_0, P_N] \subset \mathbb{R}$, containing node points $P_0, \ldots, P_i, \ldots, P_N$ mapped by curve state S to (Fig. 3.2)

$$S(P_i) = p_i = (x_i, y_{\,i}, z_i)\,, \quad i = 0, \ldots, N \tag{3.1}$$

Each sub-interval $\Pi_i = [P_{i-1}, P_i] \subset \Pi$ of parameter length $L_i = P_i - P_{i-1}$ has in $\mathbb{R}^3$ an associated image vector (abstractly and in coordinates)

$$\mathtt{v}_i = p_i - p_{i-1} \qquad\qquad (\mathtt{v}_i^x, \mathtt{v}_i^y, \mathtt{v}_i^z) = (x_i - x_{i-1}, y_i - y_{i-1}, z_i - z_{i-1}) \tag{3.2}$$

of geometric length

$$l_i = \|\mathtt{v}_i\| = \sqrt{(\mathtt{v}_i^x)^2 + (\mathtt{v}_i^y)^2 + (\mathtt{v}_i^z)^2}\,. \tag{3.3}$$

and a normalized direction vector

$$w_i = \left(\frac{v_i}{l_i}\right) \tag{3.4}$$

$$(w_i^x, w_i^y, w_i^z) = \left(\frac{v_i^x}{l_i}, \frac{v_i^y}{l_i}, \frac{v_i^z}{l_i}\right) \tag{3.5}$$

There is no natural,local assignment of a normal frame[1], so we must propagate a starting frame by minimal rotations. It is not needed for the present thread model based only on stretching and bending.

We associate bending with the sub intervals Π_i. We could compute the single joint angles, but the inverse trigonometric functions involved are more costly in several ways. Further, we are more concerned with the large global change of shape, and the associated non linearities, than with extreme localized bending. We need our bending formulation to behave in a way that we have zero bend in configuration where adjacent image vectors are 0° apart and maximum bend in configuration where adjacent image vectors are 180° apart. $(1 - w_i \cdot w_{i-1})$ behaves in a similar fashion and to third order matches the square of the node's angle, which normally remains small. Since we associate bending with the sub intervals, the bend is defined as

$$b_i = \left(\frac{(1 - w_{i+1} \cdot w_i) + (1 - w_{i-1} \cdot w_i)}{2} \right) \tag{3.6}$$

Each sub interval Π_i has a reference length L_i (which may be, but need not be, the length $\|V_i\|$), and a reference bend B_i (zero for all in the simplest case).

3.3.2 Dynamics

We take an energy formulation, associating energy $E_i\left(l_i, b_i\right)$ with each segment. This formalism allows for the interactions between bending and length change.

$$E_i\left(l_i, b_i\right) = E_i^{length}\left(l_i\right) + E_i^{bend}\left(b_i\right) \tag{3.7}$$

$$E_i^{length}\left(l_i\right) = \frac{\gamma}{2}\left(l_i - L_i\right)^2 \tag{3.8}$$

$$E_i^{bend}\left(b_i\right) = \frac{\beta}{2}\left(b_i - B_i\right) \tag{3.9}$$

[1] The Frenet choice of bending direction as a first normal direction breaks down where bending is zero, creates major difficulties where bending passes near zero, and is irrelevant to *mechanical* twisting.

where γ and β are stretching and bending stiffness coefficients respectively.

We ignore inertia for the present, assuming quasi-static dynamics that minimizes

$$E\left(S\right)=\sum_{i=1}^{i=m}E_i\left(l_i,b_i\right) \tag{3.10}$$

subject to appropriate constraints. We need to compute equilibrium positions minimizing E, and the forces of the device to deliver.

For a particular curve state $\overline{S}$ given by vertex image points $\overline{p_i}=\left(\overline{x_i},\overline{y_i},\overline{z_i}\right)$, we have a quadratic expansion

$$E\left(S\right)=E\left(\overline{S}\right)+D_{\overline{S}}E\left(\Delta\left(S\right)\right)+\frac{1}{2}D^2_{\overline{S}}E\left(\Delta\left(S\right)\right)+... \tag{3.11}$$

for E in general S in terms of vector $\Delta\left(S\right)$ of differences between the vertex image points $p_i=\left(x_i,y_i\,z_i\right)$ of S and those of $\overline{S}$.

$$\Delta=\Delta\left(S\right)=\begin{pmatrix}\delta x_0\\ \delta y_0\\ \delta z_0\\ \delta x_1\\ \delta y_1\\ \delta z_1\\ \vdots\\ \delta x_N\\ \delta y_N\\ \delta z_N\end{pmatrix}=\begin{pmatrix}x_0-\overline{x_0}\\ y_0-\overline{y_0}\\ z_0-\overline{z_0}\\ x_1-\overline{x_1}\\ y_1-\overline{y_1}\\ z_1-\overline{z_1}\\ \vdots\\ x_N-\overline{x_N}\\ y_N-\overline{y_N}\\ z_N-\overline{z_N}\end{pmatrix} \tag{3.12}$$

$$DE\left(\Delta\left(S\right)\right)=\Lambda\Delta \tag{3.13}$$

$$D^2E\left(\Delta\left(S\right)\right)=\Delta^TQ\Delta \tag{3.14}$$

Where Λ and Q are $(3N+3)\times 1$ and $(3N+3)\times(3N+3)$ square matrices given respectively by the first and second derivatives at $\overline{S}$ of E with respect to x_i, y_i and z_i. With no constraints, Q is necessarily degenerate, but if the coefficients γ and

β are strictly positive we can clamp one or both ends, pin both ends, or clamp one end and grasp another point with the haptic device, to give non degenerate Q. We fix $(x_0, y_0, z_0) = (0, 0, 0)$ and clamp the end by setting $y_1 = 0, z_1 = 0$. We 'grasp' by letting the user fix a point (X, Y, Z) thereby giving

$$(x_0, y_0, z_0, y_1, z_1, x_N, y_N, z_N) = (0, 0, 0, 0, 0, X, Y, Z) \tag{3.15}$$

as the current constraint. Writing C for the uninteresting constant $E\left(\overline{S}\right)$, the quadratic approximation in (3.11) becomes

$$E = C + \Lambda\Delta + \frac{1}{2}\Delta^T Q \Delta \tag{3.16}$$

To find equilibrium (unique in this local quadratic approximation) we set the derivative of (3.16) equal to zero

$$\Lambda + \Delta^T Q = 0 \tag{3.17}$$

We iterate given a current guess $\overline{S}$. The solution of (3.17) gives us a candidate $\overline{S} + \Delta_\circ$, which would be true equilibrium if E were truly quadratic. If $\overline{S}$ was a reasonable guess,$\overline{S} + \Delta_\circ$ is a much better one. When changing constraints modify E, a previous minimum is usually a good starting guess for the current one. When the currently followed minimum vanishes, the process descends to another in a 'snap-through'. Near bifurcations, where we have 'snap-through', we use scheme that was developed in the previous chapter for multi-variable states. We have explicitly demonstrated the scheme for a 2D curve, for a 3D thread (curve) it follows analogously. The forces are obtained using the scheme explained in §2.3.4, except that now the force vector is 3D.

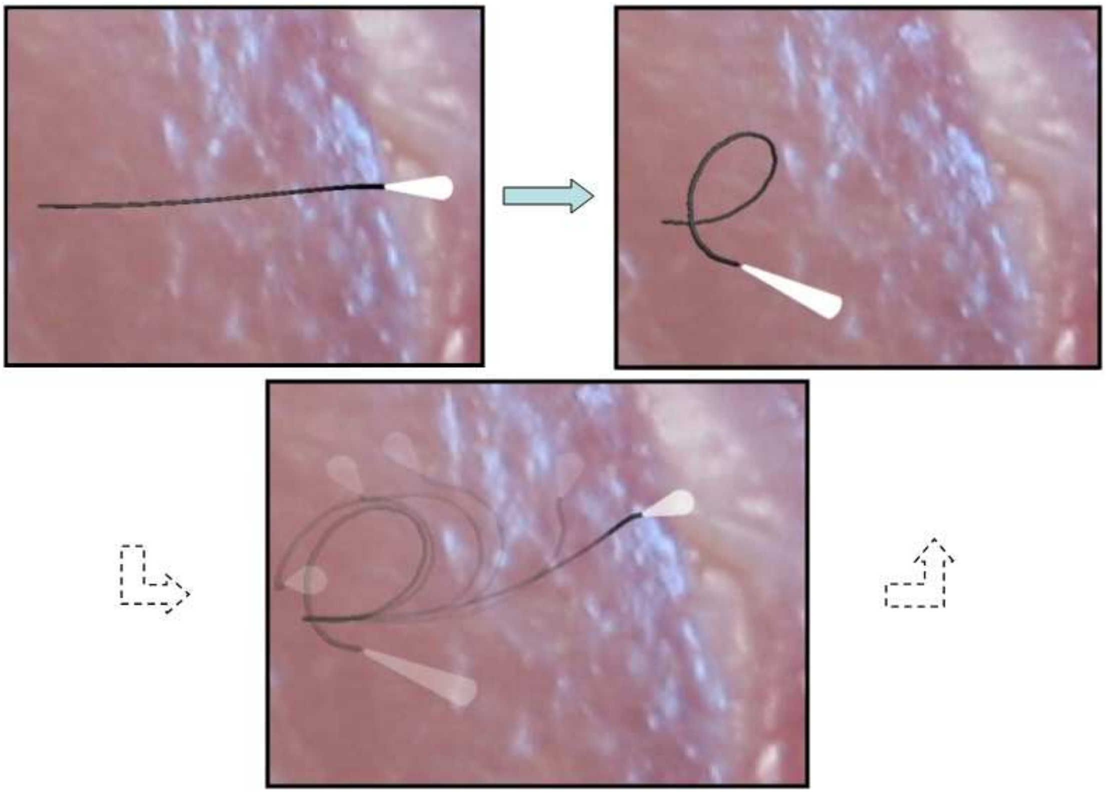


Figure 3.3: A forceps grasps the curve far to the right and the push the grasped end rightwards. Nylon mechanics forms the loop, to the right of any point where the forceps tip has gone. A ‘following’ simulation does not match the task.

3.4 Experiments and Results

We have simulated 3D thread dynamics using the scheme developed in this chapter. For the experiments, we have used a 10-segment discretization of the thread, each segment being of unit length, and used cubic spline interpolation [21] to render visually smooth thread. The user moves an end of the ‘thread’ by a haptic controller, which enforces position constraints at the end and direction constraint at the end segment, implemented both with a Delta haptic interface [18] and a desktop PHANToM [19], on a Dell workstation with Dual 2.6 XG Xeon processors. The loop formation in a thread comes from the bending of thread, unlike following the end point of thread around a loop. This is demonstrated in Fig. 3.3, where a thread fixed at one end is manipulated at the other end by a virtual forcep. The picture showing the intermediate stages of loop formation makes it clear, that a loop can be formed without following the loop ‘exactly’.

The ‘snap-through’ jumps is an essential non-linearity of a flexible nylon thread

(Fig. 1.3A). The snap-sequence is explained in Fig. 3.4. The snaps take place because of bifurcations, which means that there is change in the number of available equilibria with a small change in the system parameter, in this case the Delta or PHANToM controlled end point of thread. The snap sequence makes this phenomenon clear. The videos, illustrating this better, are available at [14].

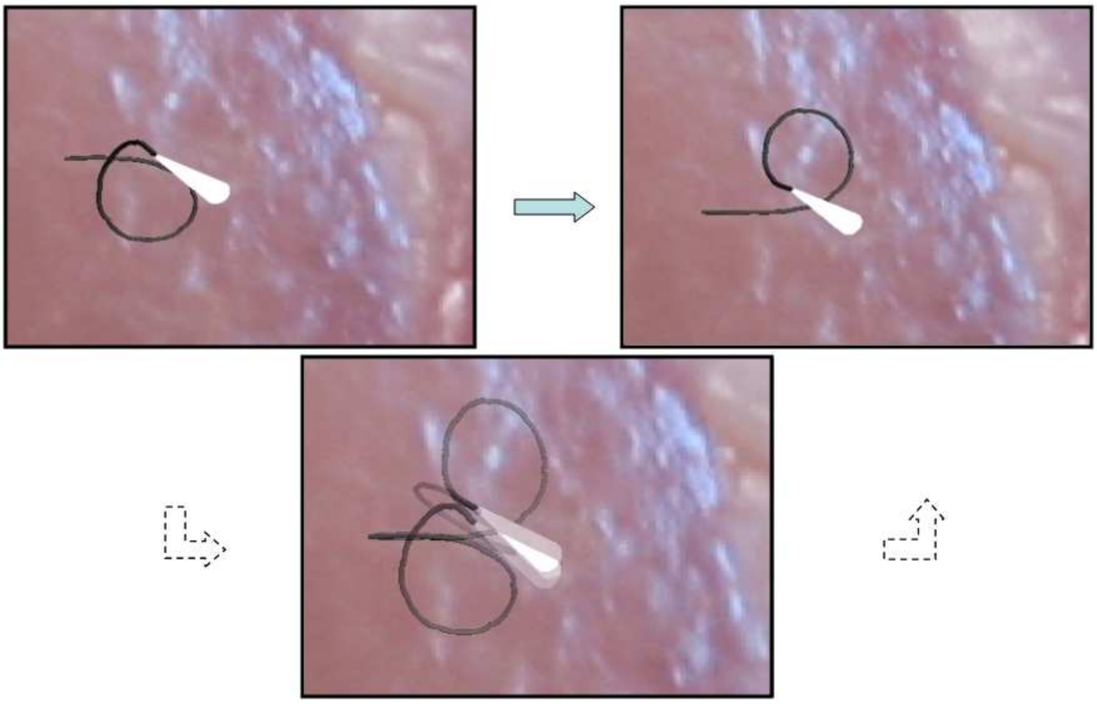


Figure 3.4: The thread encounters 'snap-through' jumps near bifurcations. Even with a small change in the forceps position, the thread undergoes a big change.

Experiments with the force feedback result in vibrations when we use the standard Newton's solution method. These vibrations are completely eliminated (Fig. 3.5) when the algorithm developed in previous chapter is implemented. The refresh rates obtained are shown in Fig. 3.6. As is usual, there is a trade-off between the haptic speed and visual realism. Better visual realism will require finer discretization of thread model, but it will increase the number of internal variables in the system leading to lower refresh rates. Where the aim is only to visualize the thread, the thread can be modelled using sufficient segments.

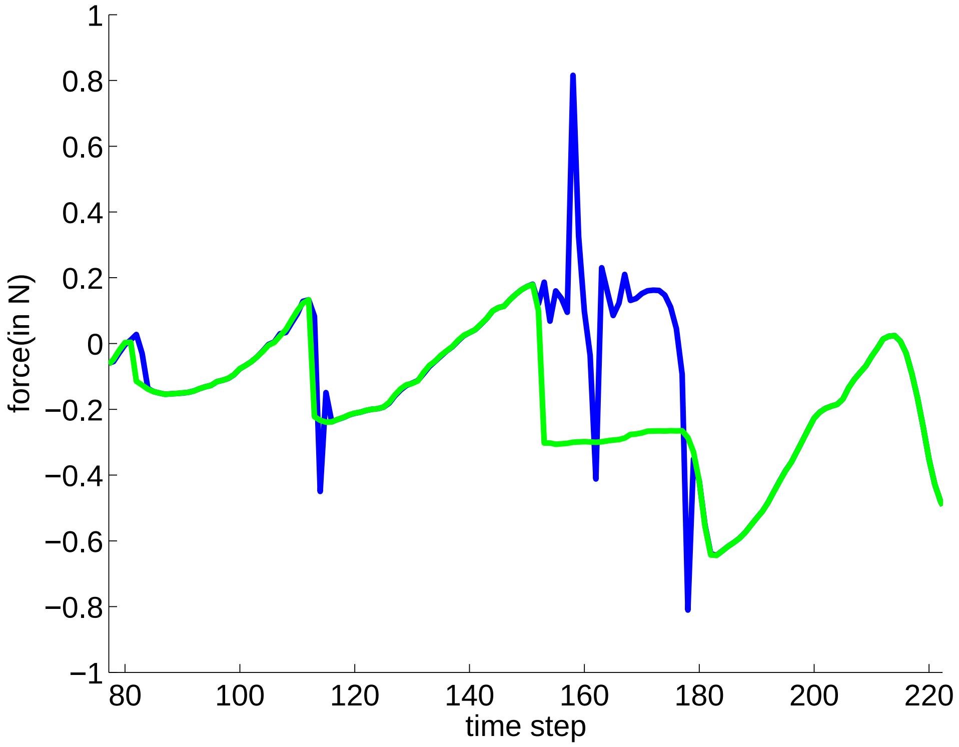


Figure 3.5: Vibrations (blue curve) are eliminated (green curve) by implementing the algorithm developed in previous chapter. The two curves are obtained by moving the grasped end of the thread along identical paths.

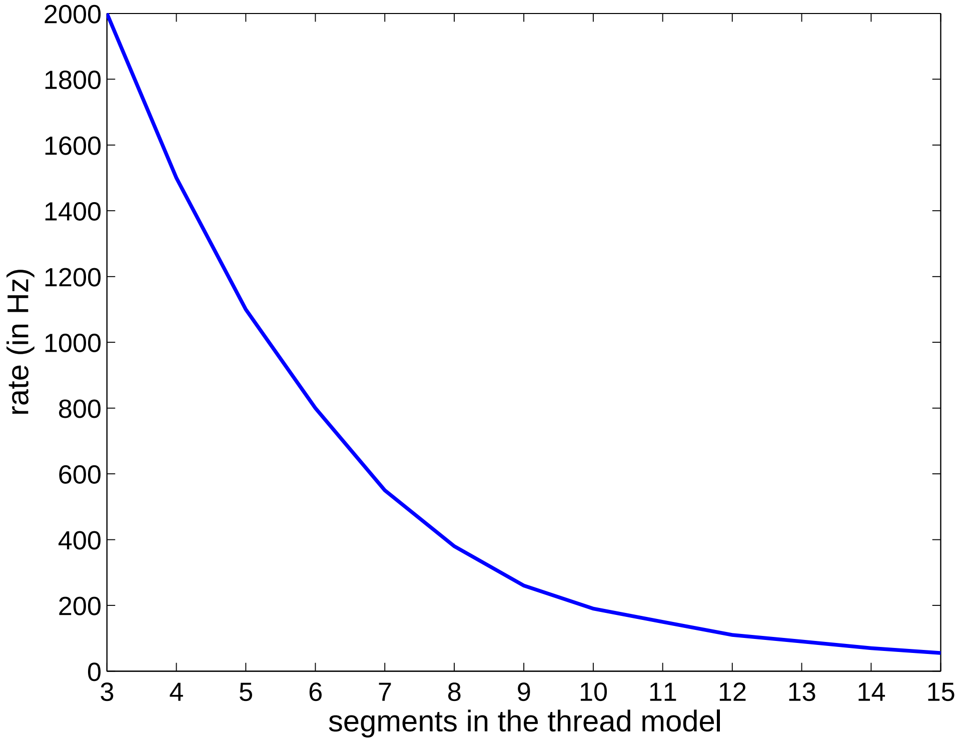


Figure 3.6: Measured numerical refresh rate as a function of the number of segments in the thread model.

3.5 Discussion

We have successfully modelled a virtual nylon thread, which demonstrates phenomenon such as large flexible deformation and 'snap-through' jumps. Using the haptic device appropriate forces can be felt, which lead to a palpable thread. The application of bifurcation handling algorithm ensures vibration free behavior of thread.

We took into account the stretching and bending energies, which was sufficient to demonstrate the advantages of energy-based method, developed here using computationally simple energy definitions. We clearly have a model which takes into account the essentials of loop formation and non-linearities of thread dynamics, which a geometric approach [1] cannot realize. However, our approach is much more computationally expensive than the latter. So, there is an inevitable trade-off.

To include twist in our model, we need to first assign a normal frame to each segment (§3.3). There is no natural,local assignment of a normal frame, so we must propagate a starting frame by minimal rotations. Measurement of twist will be by rotation of a mechanically fixed normal $\mathtt{m}$ away from this frame. But to use the vector $\mathtt{m}$ as the variable includes too many numerical degrees of freedom, since a vector takes three numbers to specify while $\mathtt{m}$ at a point p of the curve can vary only around a circle in the plane orthogonal to the curve at p. We do better to use the twist τ directly, specifying it as a scalar rate (continuous version) or step (finite version) at points along the curve. Integrating or summing it, we have a scalar angle ψ between the physically attached normal $\mathtt{m}$ and the geometrically untwisted normal field $\mathtt{n}$ along the curve. The tricky part is that $\mathtt{n}$ can only be found by propagation along the whole curve, from the (arbitrary) starting normal vector $\mathtt{n}_0$ at P_0. When we grasp the free end of the curve we are constraining the directions $\mathtt{w}_N$ and $\mathtt{m}_N$ as well as P_N; the thread must then adjust to minimize energy, balancing bending and twist. This will add a lot to the computational burden and less expensive techniques have to be developed.

Another issue that we have not discussed here is the issue of dealing with self-collision and collisions with external objects, though the former is more important. Various geometric techniques are available, and a BVH based technique is discussed in [1], [3]. [22], [23], [24], [25] and [26] discuss more about such techniques. These techniques are as computationally fast as they can be, but we work in an energy paradigm here. Using collision detection techniques for finding geometric intersection 'only' can be easily done but is not enough, as the core of our method is dynamical realism. Self-collision indeed plays an important role in loop formation and affects 'snap-through' jumps. The ideal thing for us will be to include self-collision as a constraint which we can enforce in our solution process, as we do with the end point position and direction of end segment.

Chapter 4

Conclusion

We have developed a novel technique to handle bifurcations in haptics, and a model towards a palpable virtual nylon thread. This study was prompted by work towards simulation of surgical suturing. We felt the need of simulating nylon dynamics which was not looked into carefully upto now, specifically ignoring phenomenon such as 'snap-through' jumps. We developed a computationally simple energy-based model of thread which includes stretching and bending energies. The solution process involves finding equilibrium for the thread. The jumps are successfully demonstrated and virtual thread is realistic. Jumps occur near bifurcations which means that the number of available equilibria suddenly change with slight change in system parameters. The usual numerical techniques, such as standard Newton's method, to find energy equilibria fail near bifurcations and hence result in unwanted instabilities. We showed with the example of Zeeman machine and later detailed analysis that this instability can be removed using higher derivatives for approximating energy and then solving the equilibrium equation. Haptics require very fast update rates and hence the standard Newton's method is a natural choice for the solution process. But in systems demonstrating bifurcations, which includes nylon thread, we need higher derivatives. Using bifurcation theory, we demonstrate that we can restrict the higher derivatives of energy upto fourth order. But in a system like the virtual thread model developed here, we have many variables. A typical 3D thread modelled in this thesis with 10 segments has 21 free internal variables, and finding derivatives upto fourth order in those variables is computationally very expensive and of no use whatsoever for haptic applications. We in this thesis demonstrate that we can cleverly avoid these many derivatives and look for higher derivatives along particular directions.

A full description of thread will require inclusion of twisting energy and self-collision detection, and will be dealt with in future. Nevertheless, our technique for the first time brings out the characteristic features associated with a nylon thread, which are ignored sometimes for more visual realism. The algorithm for handling bifurcations is not specific to the thread, and opens up many possibilities for the visual and haptic simulation of dynamics with bifurcations and other nonlinear dynamics.

References

[1] J. Brown, J. C. Latombe, and K. Montgomery. Real-time knot tying simulation. *The Visual Computer Journal*, 20(2-3):165–179, 2004.

[2] F Wang, T. Poston, K. M. Lim, C. L. Teo, , and E. Burdet. Multisensory learning cues using analytical collision detection between a needle and a tube. In *Proceedings of IEEE International Conference on Virtual Reality*, pages 339–346, 2004.

[3] F. Wang. *Development of a virtual reality based microsurgery trainer*. Masters thesis, National University of Singapore, December 2004.

[4] F Wang, E. Burdet, A. Dhanik, T. Poston, and C. L. Teo. Dynamic thread for real-time knot-tying. In *Proceedings of Joint eurohaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems*, 2005. accepted.

[5] R. J. Adams. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):165–474, 1999.

[6] R. J. Adams, M. R. Montreya, and B. Hannaford. Stability and performance of haptic displays: Theory and experiments. In *In Proceedings of ASME International Mechanical Engineering Congress & Exhibtion*, pages 227–234, 1998.

[7] J-C. Tsai. *Towards Guaranteed Stability in the Haptic Display of Virtual Environments*. Ph d dissertation, Northwestern University, December 1996.

[8] J. E. Colgate, P. E. Grafing, M. C. Stanley, and G. Schenkel. Implementation of stiff virtual walls in force-reflecting interfaces. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 202–208, 1993.

[9] J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. In *Proceedings of IEEE 1994 International Conference on Robotics & Automation*, pages 3205–10, 1994.

[10] B. E. Miller, J. E. Colgate, and R. A. Freeman. Computational delay and free mode environment design for haptic display. In *Proceedings of ASME Dynamic Systems and Control Division*, pages 229–236, 1999.

[11] B. E. Miller, J. E. Colgate, and R. A. Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6):712–719, 2000.

[12] E. C. Zeeman. A catastrophe machine. *Towards a Theoretical Biology 4*, pages 276–282, 1972. C H Waddington, ed.

[13] T. Poston, A. Dhanik, E. Burdet, and C. L. Teo. Haptics of buckling. In *Proceedings of Joint eurohaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems*, 2005. accepted.

[14] *http://guppy.mpe.nus.edu.sg/∼eburdet/People/ankur/video.html.*

[15] T. Poston and I. N. Stewart. *Catastrophe Theory and its Applications.* Pitman, 1978.

[16] E. Kreyszig. *Advanced Engineering Mathematics.* John Wiley, New York, 1999. 8th Edition.

[17] T. Poston and R. J. Magnus. *Infinite Dimensions and the Fold Catastrophe.* Springer-Verlag, 1979. W. Güttinger, H. Eikemeier (eds).

[18] *http://www.forcedimension.com.*

[19] *http://www.sensable.com.*

[20] J. Phillips, A. Ladd, and L. E. Kavraki. Simulated knot tying. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 841–846, 2002.

[21] I. D. Faux and M. J. Pratt. *Computational Goemetry for Design and Manufacture.* Ellis Horwood Limited, 1979.

[22] C. Sheen and J. Johnstone. On the planar intersection of natural quadrics. In *Proceedings of ACM Solid Modeling*, pages 234–244, 1991.

[23] J. R. Miller. Geometric approach to nonplanar quadric surface intersection curves. In *ACM transactions on Graphics*, volume 6, pages 274–307, 1987.

[24] S. Gottschalk, M. Lin, and D. Manocha. Obb-tree: A heirarchical structure for rapid interference detecion. In *Proceedings of ACM Siggraph*, pages 171–180, 1996.

[25] J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, 1991.

[26] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer, 2000.

Appendix A

k-determinacy

Definition *For any smooth function* $f : \mathbb{R}^n \to \mathbb{R}$*,*

f is k-determinate at 0 *if any smooth function* $f + g$*, where* g *is of order* $k + 1$ *at* 0*, can be locally expressed as* $f(y(x))$ *where* $y : \mathbb{R}^n \to \mathbb{R}^n$ *is a smooth reversible change of coordinates.*

Theorem *Let* $f : \mathbb{R} \to \mathbb{R}$ *be a smooth function, such that*

$$f(0) = Df\big|_0 = \cdots = D^{k-1}f\big|_0,$$

but

$$D^k f\big|_0 \neq 0.$$

Then there exists a smooth local change of coordinates under which f *takes the form*

$$x^k \quad (k\,odd)$$

$$\pm x^k \quad (k\,even)$$

and in the latter case the sign is that of $Df|_0$*.*

Appendix B

Splitting Lemma

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth function, with $Df(0) = 0$ whose Hessian at 0 has rank r (and corank $n - r$). Then f is equivalent, around 0, to a function of the form

$$\pm x_1^2 \pm \cdots \pm x_r^2 + \hat{f}(x_{r+1}, \ldots, x_n)$$

where

$$\hat{f} : \mathbb{R}^{n-r} \to \mathbb{R}$$

is smooth.

Proof By a linear change of coordinates $u = u(x)$ we can transform the Hessian of f at 0 into the form

$$\left[\begin{array}{cccccc|c} 1 & & & & & & \\ & \ddots & & \mathbf{0} & & & \\ & & 1 & & & & \mathbf{0} \\ & & & -1 & & & \\ & \mathbf{0} & & & \ddots & & \\ & & & & & -1 & \\ \hline & & & & & & \\ & & & \mathbf{0} & & & \mathbf{0} \\ & & & & & & \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\ \ddots\\ 1\\ -1\\ \ddots\\ -1\end{array}}\right\} r \\ \vphantom{\begin{array}{c}0\\0\\0\end{array}} \end{array}$$

The Implicit Function Theorem now allows us to express the set

$$\{u \Big| \tfrac{\partial f}{\partial u_1} = \cdots = \tfrac{\partial f}{\partial u_r} = 0\}$$

(locally) as the graph

$$\{(g_1(u_{r+1}, \ldots, u_n), \ldots, g_r(u_{r+1}, \ldots, u_n), u_{r+1}, \ldots, u_n)\}$$

of a smooth function

$$g : \mathbb{R}^{n-r} \to \mathbb{R}^r.$$

We use g to turn this graph into the $(u_{r+1}, \ldots, u_n)$-axis, by a map ϕ, easily seen to be a diffeomorphism, defined by

$$\phi(u_1, \ldots, u_n) = (u_1 + g_1(u_{r+1}, \ldots, u_n), \ldots, u_r + g_r(u_{r+1}, \ldots, u_n), u_{r+1}, \ldots, u_n).$$

(This step is crucial to the argument, as an initial tidying step to get the correct kind of dependence on $u_{r+1}, \ldots, u_n$.)

Let $F = f \circ \phi$. Locally, each function

$$F_{(u_{r+1},\ldots,u_n)} : \mathbb{R}^r \to \mathbb{R}$$

$$(u_1, \ldots, u_r) \longmapsto F(u_1, \ldots, u_r, u_{r+1}, \ldots, u_n)$$

has a Morse critical point at the origin of $\mathbb{R}^r$, though not necessarily taking the value 0 at that point. We write

$$\hat{f}(u_{r+1}, \ldots, u_n) = F(0, \ldots, 0, u_{r+1}, \ldots, u_n).$$

Now the argument appearing in the first part of the proof of the Morse Lemma (Appendix D) may be used (after generalizing Lemma 1 in Appendix D to the case where f vanishes along a multi-axis) to write

$$F(u) = \hat{f}(u_{r+1}, \ldots, u_n) + \textstyle\sum_{k,m \leq r} u_k u_m h_{km}^{(u_{r+1},\ldots,u_n)}(u_1, \ldots, u_r),$$

where for each choice of $u_{r+1}, \ldots, u_n$ the function

$$h_{km}^{(u_{r+1},\ldots,u_n)} : \mathbb{R}^r \to \mathbb{R}$$

is smooth. The remainder of the proof of Morse Lemma, applied in a $(u_{r+1}, \ldots, u_n)$-dependent fashion to this expression, reduces F to the form

$$\hat{f}(u_{r+1}, \ldots, u_n) \pm v_1^2 \pm \cdots \pm v_r^2$$

and proves the theorem. In essence, the whole *process* of reduction to standard form used to prove the Morse Lemma depends smoothly on $u_{r+1}, \ldots, u_n$, once the initial tidying has been done.

This theorem says, in a strong sense, that the behavior of a function near a degenerate critical point can be found by studying a function involving a number of variables equal to the *corank* of the Hessian. Thus, say, a critical point of a function of 2001 variables, of corank 3, requires us to study only a function of three variables. This reduction to a small number of variables is what makes Splitting Lemma so useful, and so surprising.

Appendix C

Transversality

Two subspaces U and V of $\mathbb{R}^n$ are *transverse* if they meet in a subspace whose dimensions is as small as possible. If dim $U = s$ and dim $V = t$ then this minimal dimension is

$$max(0, s + t - n).$$

For example, two planes in $\mathbb{R}^3$ are transverse if they meet in a line (or, equivalently, do not coincide), for

$$max(0, 2 + 2 - 3) = 1.$$

A 4-dimensional and a 6-dimensional subspace of $\mathbb{R}^7$ are transverse if they meet in a space of dimension

$$max(0, 4 + 6 - 7) = 3.$$

Appendix D

The Morse Lemma

Lemma 1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be smooth in a neighbourhood of 0, with $f(0) = 0$. Then in a possibly smaller neighbourhood, there exist functions $g_i : \mathbb{R}^n \to \mathbb{R}$ such that*

$$f = \textstyle\sum_{i=1}^n x_i g_i,$$

where each g_i is smooth, and $g_i(0) = \frac{\partial f}{\partial x_i}\Big|_0$.

Proof We have

$$f(x_1, \ldots, x_n) = \textstyle\int_0^1 \frac{d}{dt}(f(tx_1, \ldots, tx_n))dt = \int_0^1 \sum_{i=1}^n \frac{\partial f}{\partial x_i}\Big|_{(tx_1,\ldots,tx_n)} \cdot x_i dt$$

Hence we define

$$g_i(x_1, \ldots, x_n) = \textstyle\int_0^1 \frac{\partial f}{\partial x_i}\Big|_{(tx_1,\ldots,tx_n)} dt.$$

Differentiating partially with respect to x_i shows that $g_i(0) = \frac{\partial f}{\partial x_i}\Big|_0$

Lemma 2 *(Morse Lemma) Let u be a nondegenerate critical point of the smooth function $f : \mathbb{R}^n \to \mathbb{R}$. Then there is a local coordinate system $(y_1, \ldots, y_n)$ in a neighbourhood U of u, with $y_i(u) = 0$ for all i, such that*

$$f = f(u) - y_1^2 - \cdots - y_l^2 + y_{l+1}^2 + \cdots + y_n^2$$

for all $y \,\varepsilon\, U$.

Proof We can translate the origin to u and hence assume $u = 0$, and $f(u) = f(0) = 0$. By Lemma 1 we may then write

$$f(x) = \textstyle\sum_{j=1}^n x_j g_j(x)$$

in some neighborhood of 0. Since 0 is a critical point, we have

$$g_j(0) = \frac{\partial f}{\partial x_j}\Big|_0 = 0.$$

Hence, using Lemma 1 again, there exist smooth functions h_{ij} such that

$$g_j(x) = \textstyle\sum_{i=1}^n x_i h_{ij}(x),$$

and we can write

$$f(x) = \sum_{i,j=1}^{n} x_i x_j h_{ij}(x). \tag{D.1}$$

If we replace h_{ij} by

$$\hat{h_{ij}} = \tfrac{1}{2}\left(h_{ij} + h_{ji}\right)$$

this equation still holds, and further $\hat{h_{ij}} = \hat{h}_{ji}$.

Partially differentiating (D.1) twice, we see that

$$\frac{\partial^2 f}{\partial x_i \partial x_j}\Big|_0 = 2\hat{h_{ij}}(0),$$

and hence the matrix

$$[\hat{h_{ij}}(0)] = \left[\frac{1}{2}\frac{\partial^2 f}{\partial x_i \partial x_j}\Big|_0\right]$$

is non-singular since 0 is nondegenerate critical point.

Suppose inductively that there exist local coordinates $u_1, \ldots, u_n$ in a neighbourhood U_1 of 0 such that

$$f = \pm u_1^2 \pm \cdots \pm u_{r-1}^2 + \textstyle\sum_{i,j \geq r} u_i u_j H_{ij}(u_1, \ldots, u_n),$$

where $H_{ij} = H_{ji}$. By a linear change in the final r coordinates we may assume that $H_{rr}(0) \neq 0$. Let

$$g(u_1, \ldots, u_n) = \sqrt{\mid H_{rr}(u_1, \ldots, u_n) \mid}.$$

By the Inverse Function Theorem this is smooth in some neighbourhood U_2 of 0, contained in U_1. (This step is the main reason why the Morse Lemma holds *locally* in general). We change coordinates to $v_1, \ldots, v_n$ defined by

$$v_i = u_i \quad (i \neq r),$$

$$v_r = g(u_1, \ldots, u_n)\left(u_r + \textstyle\sum_{i>r} \frac{u_i H_{ir}(u_1,\ldots,u_n)}{H_{rr}(u_1,\ldots,u_n)}\right)$$

which, again using the Inverse Function Theorem, is a local diffeomorphism. Now

$$f(v_1, \dots, v_n) = \pm v_1^2 \pm \cdots \pm v_r^2 + \textstyle\sum_{i,j \geq r+1} v_i v_j H'_{ij}(v_1, \dots, v_n),$$

a formula like that for the u_i, but with r replaced by $r + 1$. Hence, by induction, we obtained the conclusions of the theorem.