Stability of helical equilibria

Pauline Rüegg-Reymond
Supervised by: Prof. John H. Maddocks and Ludovica Cotta-Ramusino

January 12, 2012

1 Introduction

This project intends to provide a new tool for the study of helical rods. Indeed if we have now analytical expressions for equilibria of such rods, it is less easy to find explicit stability conditions for these equilibria. The goal is to write a program with Matlab to compute numerically the stability of helical equilibria.

Concepts of equilibrium and stability of helices are related to the energy of rods. A rod is at equilibrium if it is an extremum of the energy and is stable if it minimizes it.

2 Notations and Elements of Theory about Rods

A physical rod is mathematically defined by a curve $\mathbf{r} : I \subset \mathbb{R} \to \mathbb{R}^3$ and a frame $\{\mathbf{d_1}(s), \mathbf{d_2}(s), \mathbf{d_3}(s)\}$ which will be most of the time written in a matrix form

$$\mathbf{R}(s) = \begin{bmatrix} | & | & | \\ \mathbf{d_1}(s) & \mathbf{d_2}(s) & \mathbf{d_3}(s) \\ | & | & | \end{bmatrix}.$$

This frame represents the orientation of the material cross-section of the rod and will be referred to as the *moving frame* because it depends on s while the canonical frame $\{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\}$ will be referred to as the *fixed frame*. In this section, $\mathbf{a}$ will denote a variable (matrix or vector) expressed in the fixed frame and $\tilde{\mathbf{a}}$ its expression in the moving frame. For a vector $\mathbf{a} \in \mathbb{R}^3$, we have $\tilde{\mathbf{a}} = \mathbf{R}^{-1}\mathbf{a}$ and for a matrix $\mathbf{A} \in M^3$, $\tilde{\mathbf{A}} = \mathbf{R}^{-1}\mathbf{A}\mathbf{R}$, or when $\mathbf{R}$ is orthogonal, $\tilde{\mathbf{a}} = \mathbf{R}^T\mathbf{a}$ and $\tilde{\mathbf{A}} = \mathbf{R}^T\mathbf{A}\mathbf{R}$.

We will often take $\mathbf{d_3}(s) = \mathbf{r}(s)$ and $\mathbf{d_1}$ and $\mathbf{d_2}$ two vectors in the normal plane such that $\{\mathbf{d_i}\}$ is orthonormal and right-handed. From now on, this will be generally assumed.

The *strains* are the functions $\tilde{\mathbf{u}}, \tilde{\mathbf{v}} : I \to \mathbb{R}^3$ defined by

$$\tilde{\mathbf{v}}(s) = \mathbf{R}^T\mathbf{r}'(s) \tag{1}$$

and

$$\mathbf{d_i}'(s) = \tilde{\mathbf{u}}(s) \times \mathbf{d_i}(s) \tag{2}$$

for $i = 1, 2, 3$. $\tilde{\mathbf{u}}$ is called the *Darboux vector*. Introducing the notation

$$\mathbf{u}^\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u1 & 0 \end{bmatrix}$$

we can rewrite (2) $\mathbf{R}' = \mathbf{R}\tilde{\mathbf{u}}^\times$ or $\tilde{\mathbf{u}}^\times = \mathbf{R}^T\mathbf{R}'$.

We remark here that a rod can be equivalently given by its curve and intrinsic frame $\mathbf{r}$ and $\mathbf{R}$ and by its strains $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$.

The initial configuration of the rod is represented by intrinsic strains $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$. This initial or reference configuration can be thought of as the configuration of the rod in the absence of external forces.

$\mathbf{m}(s)$ and $\mathbf{n}(s)$ are respectively the resultant moment and the resultant force applied on the rod $(\mathbf{r}, \mathbf{R})$ at the point s. They are called the *stresses*.

Assuming the rod is uniform and hyperelastic, there exists a convex and coercive strain-energy density function $W : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ such that $W(\mathbf{0}, \mathbf{0}) = 0$. This function leads to the *constitutive relations* between stresses and strains :

$$\begin{aligned} \mathbf{m} &= \frac{\partial}{\partial \mathbf{u}} W(\mathbf{u} - \hat{\mathbf{u}}, \mathbf{v} - \hat{\mathbf{v}}) \\ \mathbf{n} &= \frac{\partial}{\partial \mathbf{v}} W(\mathbf{u} - \hat{\mathbf{u}}, \mathbf{v} - \hat{\mathbf{v}}) \end{aligned} \tag{3}$$

In the following, the srain-energy density will be quadratic, i.e.

$$W(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \begin{bmatrix} \mathbf{v}^T & \mathbf{u}^T \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix}.$$

$\mathbf{P} \in M^6$ is called the *stiffness matrix* and is of the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{K} \end{bmatrix}$$

with $\mathbf{K}$, $\mathbf{A}$ and $\mathbf{G} \in M^3$ such that $\mathbf{P}$, $\mathbf{K}$ and $\mathbf{A}$ are symmetric and positive definite. Thus the constitutive relations (3) are :

$$\begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{v} - \hat{\mathbf{v}} \\ \mathbf{u} - \hat{\mathbf{u}} \end{bmatrix} \tag{4}$$

The energy of a rod $q(s) = (\mathbf{r}(s), \mathbf{R}(s))$ where $s \in [0, L]$ is defined by the functional

$$E[q(s)] = \int_0^L W(\mathbf{u}(s) - \hat{\mathbf{u}}(s), \mathbf{v}(s) - \hat{\mathbf{v}}(s)) ds. \tag{5}$$

3 First Variation

A rod $\bar{q}(s)$ is at *equilibrium* if its energy $E[q(s)]$ has an extremum at $q(s) = \bar{q}(s)$.

Definition. *Let $J[y]$ be a functional defined on some normed linear space and let*

$$\Delta J[h] = J[y + h] - J[y]$$

*be its **increment**. If y is fixed, $\Delta J[h]$ is a functional of h, generally a nonlinear one. If $\Delta J[h] = \phi[h] + \epsilon \|h\|$ where $\phi[h]$ is a linear functional and $\epsilon \xrightarrow[\|h\| \to 0]{} 0$, then the functional $J[y]$ is said to be **differentiable** and $\phi[h]$ is called the **variation** of $J[y]$ and is denoted by $\delta J[h]$. The variation of a differentiable functional is unique.*

Theorem. *Let $J[y]$ be a differentiable functional. A necessary condition for $J[y]$ to have an extremum for $y = \bar{y}$ is that its variation vanishes for $y = \bar{y}$, i.e. that $\delta J[h] = 0$ for $y = \bar{y}$ and all admissible h.*

The first order variation of the energy is

$$\delta E = \frac{d}{d\alpha}|_{\alpha=0} E[q(s) + \alpha \delta q(s)] = \int_0^L \left[\frac{\partial W}{\partial \mathbf{v}} \cdot \delta \tilde{\mathbf{v}} + \frac{\partial W}{\partial \mathbf{u}} \cdot \delta \tilde{\mathbf{u}} \right] ds. \tag{6}$$

Replacing $\delta \tilde{\mathbf{v}} = \mathbf{R}^T (\delta \mathbf{r}' + \mathbf{r}'^{\times} \delta \boldsymbol{\Theta})$ and $\delta \tilde{\mathbf{u}} = \mathbf{R}^T \delta \boldsymbol{\Theta}'$, we get

$$\delta E = -\int_0^L (\mathbf{n}' \cdot \delta \mathbf{r} + (\mathbf{m}' + \mathbf{r}' \times \mathbf{n}) \cdot \delta \boldsymbol{\Theta}) ds$$

which leads to the *balance laws* :

$$\begin{aligned} \mathbf{n}' &= 0 \\ \mathbf{m}' + \mathbf{r}' \times \mathbf{n} &= 0 \end{aligned} \tag{7}$$

at equilibrium. In the moving frame, the balance laws are

$$\begin{aligned} \tilde{\mathbf{n}}' + \tilde{\mathbf{u}} \times \tilde{\mathbf{n}} &= 0 \\ \tilde{\mathbf{m}}' + \tilde{\mathbf{u}} \times \tilde{\mathbf{m}} + \tilde{\mathbf{v}} \times \tilde{\mathbf{n}} &= 0 \end{aligned} \tag{8}$$

4 Helical Rods

Going further, it can be shown that at relative equilibria for a non-isotropic rod, $\tilde{\mathbf{u}}$, $\tilde{\mathbf{v}}$, $\tilde{\mathbf{m}}$ and $\tilde{\mathbf{n}}$ are constant and

$$\begin{aligned} \tilde{\mathbf{m}} &= \mu_1 \tilde{\mathbf{u}} + \mu_2 \tilde{\mathbf{v}} \\ \tilde{\mathbf{n}} &= \mu_2 \tilde{\mathbf{u}} \end{aligned} \tag{9}$$

The fact that the components of the strains in the moving frame are constant implies that the rod is a helix.

A helix is a curve with constant curvature $\kappa > 0$ and torsion τ. It can be parametrized by the arc-length form $\mathbf{r}(s) = (r\cos(\eta_1 s), r\sin(\eta_1 s), p\eta_1 s)$ where the radius r and the pitch p are related to curvature and torsion by

$$\begin{aligned} r &= \frac{\kappa}{\kappa^2 + \tau^2} \\ p &= \frac{\tau}{\kappa^2 + \tau^2} \end{aligned} \tag{10}$$

and

$$\eta_1 = \frac{1}{\sqrt{r^2 + p^2}} \in]0, \infty[\,. \tag{11}$$

If $\hat{\mathbf{v}} = (0,0,1)^T$, $\tilde{\mathbf{u}}$ is related to curvature, torsion and register ϕ by

$$\begin{aligned} \tilde{u}_1 &= \kappa \sin\phi \\ \tilde{u}_2 &= \kappa \cos\phi \\ \tilde{u}_3 &= \tau + \phi'. \end{aligned} \tag{12}$$

Since $\tilde{\mathbf{u}}$ is constant, we obtain

$$\tilde{\mathbf{u}} \cdot \tilde{\mathbf{u}} = \eta_1^2. \tag{13}$$

5 Second Variation

Now we know the conditions for a rod to be at equilibrium, we want to study what are the conditions for an equilibrium to be stable, i.e. for the energy of this rod to be minimal.

Theorem. *A necessary conditional for the functional $J[y]$ to have a minimum for $y = \bar{y}$ is that $\delta^2 J[h] \geqslant 0$ for $y = \bar{y}$ and all admissible h.*

Setting $\mathbf{h} = \begin{bmatrix} \delta\mathbf{r} \\ \delta\boldsymbol{\Theta} \end{bmatrix}$, the second order variation of the energy is

$$\begin{aligned} \delta^2 E &= \frac{d^2}{d\alpha^2}|_{\alpha=0} E\left[q(s) + \alpha\delta q(s)\right] \\ &= \int_0^L \left[\mathbf{h}'^T \mathbf{P}\mathbf{h}' + \mathbf{h}^T \mathbf{Q}\mathbf{h} + 2\mathbf{h}'^T \mathbf{C}\mathbf{h}\right] ds \end{aligned} \tag{14}$$

where

$$\begin{aligned}
\mathbf{P} &= \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{K} \end{bmatrix} \\
\mathbf{Q} &= \begin{bmatrix} 0 & 0 \\ 0 & -\mathbf{r'}^{\times}\mathbf{A}\mathbf{r'}^{\times} + \frac{1}{2}(\mathbf{n}^{\times}\mathbf{r'}^{\times} + \mathbf{r'}^{\times}\mathbf{n}^{\times}) \end{bmatrix} \\
\mathbf{C} &= \begin{bmatrix} 0 & \mathbf{A} - \mathbf{r'}^{\times} \\ 0 & \mathbf{G}^T - \frac{1}{2}\mathbf{m}^{\times} \end{bmatrix}.
\end{aligned} \tag{15}$$

Using the variable $\tilde{\mathbf{h}} = \bar{\mathbf{R}}^T\mathbf{h}$ in the moving frame, where $\bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix}$, we obtain a similar expression for the second variation :

$$\delta^2 E\left[\tilde{\mathbf{h}}\right] = \int_0^L \left[\tilde{\mathbf{h}}'^T\breve{\mathbf{P}}\tilde{\mathbf{h}}' + \tilde{\mathbf{h}}^T\breve{\mathbf{Q}}\tilde{\mathbf{h}} + 2\tilde{\mathbf{h}}'^T\breve{\mathbf{C}}\tilde{\mathbf{h}}\right] ds \tag{16}$$

whith

$$\begin{aligned}
\breve{\mathbf{P}} &= \tilde{\mathbf{P}} \\
\breve{\mathbf{Q}} &= \tilde{\mathbf{Q}} - \bar{\mathbf{u}}^{\times}\tilde{\mathbf{C}} + \tilde{\mathbf{C}}^T\bar{\mathbf{u}}^{\times} - \bar{\mathbf{u}}^{\times}\tilde{\mathbf{P}}\bar{\mathbf{u}}^{\times} \\
\breve{\mathbf{C}} &= \tilde{\mathbf{P}}\bar{\mathbf{u}}^{\times} + \tilde{\mathbf{C}}
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
\tilde{\mathbf{P}} &= \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^T & \tilde{\mathbf{K}} \end{bmatrix} \\
\tilde{\mathbf{Q}} &= \begin{bmatrix} 0 & 0 \\ 0 & -\tilde{\mathbf{v}}^{\times}\tilde{\mathbf{A}}\tilde{\mathbf{v}}^{\times} + \frac{1}{2}(\tilde{\mathbf{n}}^{\times}\tilde{\mathbf{v}}^{\times} + \tilde{\mathbf{v}}^{\times}\tilde{\mathbf{n}}^{\times}) \end{bmatrix} \\
\tilde{\mathbf{C}} &= \begin{bmatrix} 0 & \tilde{\mathbf{A}} - \tilde{\mathbf{v}}^{\times} \\ 0 & \tilde{\mathbf{G}}^T - \frac{1}{2}\tilde{\mathbf{m}}^{\times} \end{bmatrix}
\end{aligned} \tag{18}$$

are the expressions for $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{C}$ in the moving frame and $\bar{\mathbf{u}}^{\times} = \begin{bmatrix} \tilde{\mathbf{u}} & 0 \\ 0 & \tilde{\mathbf{u}} \end{bmatrix}$.

6 Jacobi Equation and Conjugate Points

From now on, no different notation for variables expressed in the fixed or moving frame will be used anymore. Everything in this section is valid for both, but in the following all the variables will be assumed to be expressed in the moving frame.

The *Jacobi equation* of the energy is

$$(\mathbf{P}\mathbf{h}' + \mathbf{C}\mathbf{h})' - \mathbf{C}^T\mathbf{h}' - \mathbf{Q}\mathbf{h} = 0 \tag{19}$$

where $\mathbf{h}$, $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{C}$ represent either the coefficients of (14) in the fixed or (16) in the moving frame.

Definition. *The point $\bar{s}$ is said to be* ***conjugate*** *to 0 if the Jacobi equation (19) has a non identically null solution $\mathbf{h}(s)$ vanishing for $s = 0$ ans $s = \bar{s}$.*

Theorem. *If $\mathbf{P}$ is positive definite, then $\delta^2 E$ is positive definite $\forall \mathbf{h}$ such that $\mathbf{h}(0) = \mathbf{h}(L) = 0$ $\Leftrightarrow$ there is no conjugate point to 0 in $[0, L]$.*

Since we choose $\mathbf{P}$ to be symmetric and poisitive definite, we know that a rod at equilibrium is stable if its length L is smaller than its first conjugate point $\bar{s}$.

Setting the momentum $\mu = \mathbf{P}\mathbf{h}' + \mathbf{C}\mathbf{h}$, we can rewrite (19) in the Hamiltonian form

$$\begin{bmatrix}\mathbf{h}\\ \mu\end{bmatrix}' = \underbrace{\begin{bmatrix}-\mathbf{P}^{-1}\mathbf{C} & \mathbf{P}^{-1}\\ \mathbf{Q}-\mathbf{C}^T\mathbf{P}^{-1}\mathbf{C} & \mathbf{C}^T\mathbf{P}^{-1}\end{bmatrix}}_{=:\mathbf{U}} \underbrace{\begin{bmatrix}\mathbf{h}\\ \mu\end{bmatrix}}_{=:\mathbf{z}}. \tag{20}$$

The coefficients of $\mathbf{U}$ are constant in our case, so every $\mathbf{z}$ satisfying the twelve by twelve ordinary differential linear system $\mathbf{z}' = \mathbf{U}\mathbf{z}$ is a linear combination of $\mathbf{z}_1, ..., \mathbf{z}_{12}$ where $\mathbf{z}_j$ is the solution of (20) such that $\mathbf{z}_j(0) = \mathbf{e}_j \ \forall j = 1, .., 12$.

Since we are only interested in solutions for which $\mathbf{h}(0) = 0$, we compute $\mathbf{h}_j$ for $j = 1, ..., 6$, where $\mathbf{z}_j = \begin{bmatrix}\mathbf{h}_j\\ \mu_j\end{bmatrix}$ is the solution of (20) such that $\mathbf{h}_j(0) = 0$ and $\mu_j = \mathbf{e}_j$.

Then $\bar{s}$ is the first conjugate point to 0 if there exist $\alpha_1, ..., \alpha_6 \neq 0$ such that $\mathbf{h}(s) := \sum_{j=0}^{6} \alpha_j \mathbf{h}_j(s) \neq 0 \ \forall 0 < s < \bar{s}$ and $\mathbf{h}(\bar{s}) = 0$.

The existence of the α_j's is equivalent to the fact that the determinant of the *Jacobi fields*

$$\mathbf{H}(s) = \begin{bmatrix} | & | & | & | & | & |\\ \mathbf{h}_1(s) & \mathbf{h}_2(s) & \mathbf{h}_3(s) & \mathbf{h}_4(s) & \mathbf{h}_5(s) & \mathbf{h}_6(s)\\ | & | & | & | & | & |\end{bmatrix}$$

is non null $\forall 0 < s < \bar{s}$ and $\det \mathbf{H}(\bar{s}) = 0$. We look therefore for the first $\bar{s} > 0$ such that $\det \mathbf{H}(\bar{s}) = 0$.

7 Circular Rods

The stiffness matrix is of diagonal form, i.e. $\mathbf{P} = \begin{bmatrix}\mathbf{A} & 0\\ 0 & \mathbf{K}\end{bmatrix}$ with

$$\mathbf{A} = \begin{pmatrix} A_1 & 0 & 0\\ 0 & A_2 & 0\\ 0 & 0 & A_3\end{pmatrix}$$

and

$$\mathbf{K} = \begin{pmatrix} K_1 & 0 & 0\\ 0 & K_2 & 0\\ 0 & 0 & K_3\end{pmatrix},$$

if there is no coupling between the deformations.

The intrinsic strains $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ of an intrinsically straight and untwisted rod are $\hat{\mathbf{u}} = (0, 0, 0)^T$ and $\hat{\mathbf{v}} = (0, 0, 1)^T$.

$\mathbf{u} = (c, 0, 0)^T$ and $\mathbf{v} = (0, 0, 1)^T$, where $c = \frac{2\pi}{L}$, is an helical equilibrium for such a stiffness matrix and such intrinsic strains. This corresponds to an untwisted circular rod of length L. Choosing $K_1 < K_2$ ensures that the circle lies in the plane $span\{\mathbf{e}_2, \mathbf{e}_3\}$.

What we are interested in is to know whether the circle is stable until it closes or if there is a conjugate point $\bar{s} < L$. The result was already computed analytically and the purpose of the first program I did was to find the same result numerically. I did two versions : one with decoupling between out-of-plane[1] and in-plane[2] fluctuations as in the analytical computations of Ludovica Cotta-Ramusino [3] and the other without[3].

Both versions take as an input the components K_1, K_2, K_3, A_1, A_2, A_3 of $\mathbf{P}$ and the length L of the rod. The output is the first conjugate point to 0, i.e. the maximum length the rod can have, remaining stable.

[1] Function *stabCircle_offplane*, Appendix A.1
[2] Function *stabCircle_inplane* , Appendix A.2
[3] Function *stabCircle2*, Appendix B

7.1 Version with Decoupling

For the decoupled case, we have $\mathbf{h}_p = \mathbf{R}^T \left(\delta r_2, \delta r_3, \delta\Theta_1\right)^T$ for the in-plane fluctuations and $\mathbf{h}_{op} = \mathbf{R}^T \left(\delta r_1, \delta\Theta_2, \delta\Theta_3\right)^T$ for the out-of-plane fluctuations. So we have two six by six systems to solve : $\mathbf{z}' = \mathbf{U}_p\mathbf{z}$ and $\mathbf{z}' = \mathbf{U}_{op}\mathbf{z}$ where :

$$\mathbf{U}_p = \begin{bmatrix} 0 & c & -1 & \frac{1}{A_2} & 0 & 0 \\ -c & 0 & 0 & 0 & \frac{1}{A_3} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{K_1} \\ 0 & 0 & 0 & 0 & c & 0 \\ 0 & 0 & 0 & -c & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{U}_{op} = \begin{bmatrix} 0 & 1 & 0 & \frac{1}{A_1} & 0 & 0 \\ 0 & 0 & \frac{c(2K_2-K_1)}{2K_2} & 0 & \frac{1}{K_2} & 0 \\ 0 & -\frac{c(2K_3-K_1)}{2K_3} & 0 & 0 & 0 & \frac{1}{K_3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c^2\frac{K_1^2}{4K_3} & 0 & -1 & 0 & \frac{c(2K_3-K_1)}{2K_3} \\ 0 & 0 & -c^2\frac{K_1^2}{4K_2} & 0 & -\frac{c(2K_2-K_1)}{2K_2} & 0 \end{bmatrix}.$$

The determinant of the Jacobi fields is $\det \mathbf{H}(s) = \det \mathbf{H}_p(s) \det \mathbf{H}_{op}$.

7.2 Analytical Results

For $K_3 < K_1 < K_2$, the determinant of the out-of-plane Jacobi fields at $s = L$ is

$$\det \mathbf{H}_{op}(L) = 2L^5(\cosh \rho L - 1)\frac{B}{(2\pi)^4 K_1^2 (K_1 - K_3)}$$

where

$$B = 1 + \left(\frac{2\pi}{L}\right)^2 \frac{K_3 - K_1}{A_1}$$

and

$$\rho = \left(\frac{2\pi}{L}\right)\sqrt{\frac{(K_1 - K_3)\,(K_2 - K_1)}{K_2 K_3}}.$$

So for

$$L = \bar{L} = 2\pi\sqrt{\frac{K_1 - K_3}{A_1}},$$

we have $B = 0$ which implies that the solution

$$\mathbf{z}(s) := \mathbf{h}_3(s) - \frac{A_1}{K_3}\sqrt{\frac{K_1 - K_3}{A_1}}\mathbf{h}_1(s)$$

satisfies $\mathbf{z}(0) = \mathbf{z}(\bar{L}) = 0$. So there is a conjugate point.
For $K_3 > K_1$, we have

$$\det \mathbf{H}_{op}(L) = 2L^5(1 - \cos \lambda L)\frac{B}{(2\pi)^4 K_1^2 (K_3 - K_1)}$$

where

$$\lambda = \frac{2\pi}{L}\sqrt{\frac{(K_3 - K_1)\,(K_2 - K_1)}{K_2 K_3}}$$

and for $K_1 = K_3$,

$$\det \mathbf{H}_{op}(L) = \frac{L^5 (K_2 - K_1)}{(2\pi)^2 K_1^3 K_2}.$$

These expression do not vanish for any L.

The determinant of the implane Jacobi fields is

$$\det \mathbf{H}_p = \frac{W^2 L^7}{4 (2\pi)^4 K_1^3}$$

where

$$W = 1 + \left(\frac{2\pi}{L}\right)^2 K_1 \left(\frac{1}{A_2} + \frac{1}{A_3}\right)$$

and never vanishes as well.

7.3 Version without Decoupling

The $\mathbf{Q}$ and $\mathbf{C}$ matrices without decoupling are

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c^2 A_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & c^2 A_2 & -cA_2 & 0 & 0 \\ 0 & 0 & -cA_2 & A_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & c^2(K_3 - K_1) + A_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & c^2(K_2 - K_1) \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & -A_1 & 0 \\ 0 & 0 & -cA_2 & A_2 & 0 & 0 \\ 0 & cA_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{c}{2}(K_1 - 2K_2) \\ 0 & 0 & 0 & 0 & \frac{c}{2}(2K_3 - K_1) & 0 \end{bmatrix}$$

and $\mathbf{U}$ is a twelve by twelve matrix computed as in (20).

8 Unshearable and Inextensible Rods

For inextensible and unshearable rods, $\mathbf{v} \equiv \hat{\mathbf{v}} = (0, 0, 1)^T$ and the stiffness matrix $\mathbf{P}$ takes the form

$$\mathbf{P} = \lim_{\omega \to 0} \begin{bmatrix} \mathbf{A}/\omega^2 & \mathbf{G}/\omega \\ \mathbf{G}^T/\omega & \mathbf{K} \end{bmatrix}. \tag{21}$$

We then have

$$\begin{aligned} \mathbf{P}^{-1} &= \lim_{\omega \to 0} \begin{bmatrix} \omega^2 \mathbf{A}^{-1} \left[\mathbb{I} + \mathbf{G}(\mathbf{K} - \mathbf{G}^T \mathbf{A}^{-1} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{A}^{-1}\right] & -\omega \mathbf{A}^{-1} \mathbf{G} (\mathbf{K} - \mathbf{G}^T \mathbf{A}^{-1} \mathbf{G})^{-1} \\ -\omega (\mathbf{K} - \mathbf{G}^T \mathbf{A}^{-1} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{A}^{-1} & (\mathbf{K} - \mathbf{G}^T \mathbf{A}^{-1} \mathbf{G})^{-1} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & (\mathbf{K} - \mathbf{G}^T \mathbf{A}^{-1} \mathbf{G})^{-1} \end{bmatrix} =: \begin{bmatrix} 0 & 0 \\ 0 & \breve{\mathbf{K}}^{-1} \end{bmatrix} \end{aligned}$$

where $\breve{\mathbf{K}}$ denotes the Schur complement, i.e. $\breve{\mathbf{K}} = \mathbf{K} - \mathbf{G}^T \mathbf{A}^{-1} \mathbf{G}$.

In the following, $\mathbf{K}$, $\mathbf{A}$ and $\mathbf{G}$ won't be mentioned anymore and $\mathbf{K}$ will denote the Schur complement previously denoted $\breve{\mathbf{K}}$.

For physical reasons, $\mathbf{K}$ will be assumed to be of the form

$$\mathbf{K} = \begin{bmatrix} K_1 & 0 & K_{13} \\ 0 & K_2 & K_{23} \\ K_{13} & K_{23} & K_3 \end{bmatrix}$$

with $K_1 \leqslant K_2$.

The quadratic strain-energy density is approximated by a function of $\mathbf{u}$ only: $W(\mathbf{u}-\hat{\mathbf{u}}) = \frac{1}{2}(\mathbf{u}-\hat{\mathbf{u}})^T \mathbf{K}(\mathbf{u}-\hat{\mathbf{u}})$ and we have

$$\mathbf{m} = \mathbf{K}(\mathbf{u}-\hat{\mathbf{u}}). \tag{22}$$

For a given constant $\hat{\mathbf{u}}$ and a given $\mathbf{K}$, it can be shown that every equilibria is an helix described by its constant Darboux vector $\mathbf{u}$ lying on the hyperboloid

$$(u_1 - a - bu_3)(u_2 - c - du_3) = (a + bu_3)(c + du_3) \tag{23}$$

where

$$\begin{aligned} a &= \frac{K_1\hat{u}_1 + K_{13}\hat{u}_3}{K_1 - K_2}, \\ b &= \frac{K_{13}}{K_2 - K_1}, \\ c &= \frac{K_2\hat{u}_2 + K_{23}\hat{u}_3}{K_2 - K_1}, \\ d &= \frac{K_{23}}{K_1 - K_2}. \end{aligned}$$

Knowing $\mathbf{u}$ on the hyperboloid (23), we can compute $\mathbf{m}$ and $\mathbf{n}$ with the constitutive relation (22) and the result (9). $\mathbf{U}$ can be found using (20) with $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{C}$ defined in (17) and the components of (18) replaced by the components of (21) :

$$\mathbf{U} = \begin{bmatrix} -\mathbf{u}^\times & -\mathbf{v}^\times & 0 & 0 \\ 0 & \frac{1}{2}(\mathbf{K}^{-1}\mathbf{m}^\times) - \mathbf{u}^\times & 0 & \mathbf{K}^{-1} \\ 0 & 0 & -\mathbf{u}^\times & 0 \\ 0 & \frac{1}{4}\mathbf{m}^\times\mathbf{K}^{-1}\mathbf{m}^\times - \frac{1}{2}(\mathbf{v}^\times\mathbf{n}^\times + \mathbf{n}^\times\mathbf{v}^\times) & -\mathbf{v}^\times & \frac{1}{2}\mathbf{m}^\times\mathbf{K}^{-1} - \mathbf{u}^\times \end{bmatrix}. \tag{24}$$

So basically, the function computing the first conjugate point of a helix should take as inputs: K_1, K_2, K_3, K_{13} and K_{23} such that the matrix $\mathbf{K}$ is positive definite, $\hat{\mathbf{u}}$ and $\mathbf{u}$ such that the corresponding helix is an equilibrium, i.e. $\mathbf{u}$ on the hyperboloid (23). Assuming we know such a $\mathbf{u}$, which we will discuss later, we still have an issue: Matlab's ODE solvers need to know on which interval they have to integrate the equation.

Therefore, we are presently only able to know if a rod is stable until a chosen length L or until where it is stable if this is not the case. I used mainly $L = 10$ for the computations because this proved to give accurate enough results.

8.1 Finding Equilibria

In a first place, I was not given points on the hyperboloid (23) and had to find such points before computing the stability of the corresponding helix. Therefore, the radius r and the pitch p of the helix are given as an input of the function. Every $\mathbf{u}$ of the hyperboloid corresponding with a helix with the given pitch and the given radius (there exist at most four such helices) is computed in the following way :

(11) gives η_1 and

$$u_3 = \tau = \frac{p}{r^2 + p^2}.$$

Setting now $x = a + bu_3$ and $y = c + du_3$, we can rewrite (23) : $(u_1 - x)(u_2 - y) = xy$ which allows us to express u_2 in function of u_1 :

$$u_2 = y + \frac{xy}{u_1 - x}.$$

And since $\mathbf{u}\cdot\mathbf{u} = u_1^2 + u_2^2 + u_3^2 = \eta_1^2$, we have now an equation in u_1 to solve:

$$\eta_1^2 = u_1^2 + \left(y + \frac{xy}{u_1 - x}\right)^2 + u_3^2.$$

This is equivalent to finding the real roots of the fourth degree polynomial in u_1 :

$$u_1^4 - 2xu_1^3 + \left(x^2 + y^2 - u_3^2 - \eta_1^2\right) u_1^2 + 2x\left(2y^2 - \eta_1^2 - u_3^2\right) u_1 - x^2\left(y^2 + \eta_1^2 + u_3^2\right) \qquad (25)$$

This is done by the Matlab function *equ_unsh_inext*[4] and the stability is computed by the function *stab_unsh_inext*[5]

8.2 Stability of Mesh Points of the Hyperboloid

The final part of the work consisted in using together the results Etienne Favre, Giacomo Rosilho de Souza and I had found. During their work, they computed a mesh of the hyperboloid (23) in order to visualize it. Provided these points, there is no longer need to compute equilibria for given pitch and radius and the input of the function[6] is simply a $\mathbf{u}$ of the mesh.

The parameters we used were $K_1 = 1$, $K_2 = 1.5$, $K_3 = 1.2$, $K_{13} = K_{23} = 0.5$, $\hat{\mathbf{u}} = (1, 0, 0)^T$ and $L = 10$. The first trial was on a 10×10 points mesh and took a few second to compute. A mesh with 100×100 points takes about forty-five minutes to compute on my personal laptop and gives a result fine enough to see what happens. To have a smoother view of the hyperboloid, we decided to use a mesh of 300×300 points.

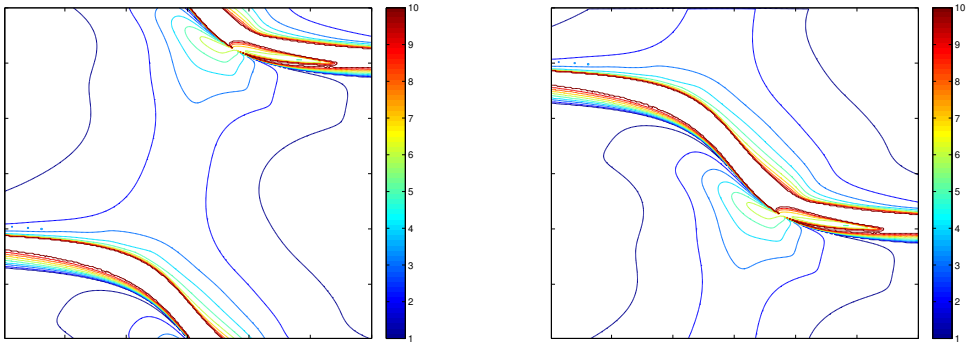

Figure 1: Stability of the mesh points of the hyperboloid cut lengthwise at two different longitudes and unwrapped. The top-edge joins the bottom-edge of each picture. The vertical middle represents the section of the hyperboloid having the smallest diameter.

8.3 Comparison with the Circular Case

To ensure that the results are good, we can compare the unshearable and inextensible case on circular rods with the circular case. This needs a slight modification of the $\mathbf{U}$ matrix : taking the decoupled case, the $\frac{1}{A_j}$ are replaced by 0 in $\mathbf{U}_p$ and $\mathbf{U}_{op}$.

Then, using the function *stab_unsh_inext2* with chosen K_1, K_2, K_3 and L and $K_{13} = K_{23} = 0$, $\hat{\mathbf{u}} = (0, 0, 0)^T$ and $\mathbf{u} = \left(\frac{2\pi}{L}, 0, 0\right)^T$ and the function *stabCircleUnshInext*[7] with the same K_1, K_2, K_3 and L should provide the same result. Since this was the case for every set of values I tried, we can assume that the functions *stab_unsh_inext* and *stab_unsh_inext2* give the desired output.

[4] Appendix C.1
[5] Appendix C.2
[6] Function *stab_unsh_inext2*, Appendix D.1
[7] Appendix E

9 Conclusion

Interesting results and questions arose from this project. We still don't know how to test if a helix is unconditionally stable though we have some clues. However, we now have a good tool to test the stability of a helix up to a certain length which is already a great step forward.

A Decoupled Circular Case

A.1 Out of the Plane Fluctuations

```
function [ CP ] = stabCircle_offplane( K1,K2,K3,A1,L )
%STABCIRCLE_OFFPLANE computes the first conjugate point of the out-of-plane
%fluctuations of a circular rod of length L with diagonal stiffness matrix
%given by K1,K2,K3,A1,A2,A3
%   The strains correspond to
%   u = [c;0;0] and v =[0;0;1]
%   The strains in the reference state correspond to
%   u_ref = [0;0;0] and v_ref = [0;0;1]
%   The function also plots the determinant of the Jacobi fields of the
%   out-of-plane fluctuations in [0,L]
%   If there is no conjugate point in [0,L], the output is L.
%   We should have K2>K1 and the matrix K positive definite.

c = 2*pi/L;

%We declare U as a global variable in order to use it to define the
%function odefun that we need to solve the ode
global U;
U =[0 1 0 1/A1 0 0;...
    0 0 c*(2*K2-K1)/(2*K2) 0 1/K2 0;...
    0 -c*(2*K3-K1)/(2*K3) 0 0 0 1/K3;...
    0 0 0 0 0 0;...
    0 -c*c*K1*K1/(4*K3) 0 -1 0 c*(2*K3-K1)/(2*K3);...
    0 0 -c*c*K1*K1/(4*K2) 0 -c*(2*K2-K1)/(2*K2) 0];

N = 1000;%numbers of points -1 at which the determinant will be computed
tspan = 0:L/N:L;

%@odefun calls the function built with the global variable U that
%corresponds to the ode z'=Uz
[t,h1op] = ode45(@odefun,tspan,[0;0;0;1;0;0]);
[~,h2op] = ode45(@odefun,tspan,[0;0;0;0;1;0]);
[~,h3op] = ode45(@odefun,tspan,[0;0;0;0;0;1]);

clear U;

h1op = h1op(:,1:3);
h2op = h2op(:,1:3);
h3op = h3op(:,1:3);

D = zeros(1,N+1);
for i = 1:N+1
    D(i) = det([h1op(i,:); h2op(i,:); h3op(i,:)]);
end
figure;
plot(t,D);

%choice for the tolerance
zero = 10e-3;

%if D is null only in some points, there is a conjugate point
if ~all(abs(D(2:N+1)) >= zero) && ~all(abs(D(2:N+1)) < zero)
    %disp('1');
    CPtmp = find(abs(D(2:N+1)) < zero, 1, 'first');
    %if the previous point is also zero, this is not a conjugate point
    if abs(D(CPtmp)) >= zero
```

```
57          CP = length/N*CPtmp;
58      end
59      clear CPtmp;
60  end
61
62  %if D crosses the horizontal ax, there is a conjugate point
63  if ~all(D>=0) && ~all(D<=0)
64      %disp('2');
65      if D(2)>0
66          CPtmp = find(D<0,1,'first');
67      else
68          CPtmp = find(D>0,1,'first');
69      end
70      CPtmp = -D(CPtmp-1)/(D(CPtmp)-D(CPtmp-1)) + CPtmp-1;
71      CPtmp = (CPtmp-1)*length/N;
72      if exist('CP','var')
73          CP = min(CP,CPtmp);
74      else
75          CP = CPtmp;
76      end
77
78  %if D is nonzero (but in 0) we don't have any conjugate point
79  elseif (all(D>=0) || all(D<=0)) && ~all(abs(D(2:N+1)) < zero)
80      %disp('3');
81      CP = length;
82
83  %if D is identically null, we can't say anything about the existence of
84  %conjugate points
85  elseif all(abs(D) < zero)
86      %disp('4');
87      CP = 0;
88  end
89
90  end
```

A.2 In Plane Fluctuations

```
1   function [ CP ] = stabCircle_inplane( K1,A2,A3,L )
2   %STABCIRCLE_INPLANE computes the first conjugate point of the inplane
3   %fluctuations of a circular rod of length L with diagonal stiffness matrix
4   %given by K1,K2,K3,A1,A2,A3
5   %   The strains correspond to
6   %   u = [c;0;0] and v =[0;0;1]
7   %   The strains in the reference state correspond to
8   %   u_ref = [0;0;0] and v_ref = [0;0;1]
9   %   The function also plots the determinant of the Jacobi fields of the
10  %   inplane fluctuations in [0,L]
11  %   If there is no conjugate point in [0,L], the output is L.
12  %   We should have K2>K1 and the matrix K positive definite.
13
14  c = 2*pi/L;
15
16  %We declare U as a global variable in order to use it to define the
17  %function odefun that we need to solve the ode
18  global U;
19  U =[0 c -1 1/A2 0 0;...
20      -c 0 0 0 1/A3 0;...
21      0 0 0 0 0 1/K1;...
22      0 0 0 0 c 0;...
23      0 0 0 -c 0 0;...
24      0 0 0 1 0 0];
25
26  N = 1000;%numbers of points -1 at which the determinant will be computed
27  tspan = 0:L/N:L;
28
29  %@odefun calls the function built with the global variable U that
30  %corresponds to the ode z'=Uz
31  [t,h1p] = ode45(@odefun,tspan,[0;0;0;1;0;0]);
32  [~,h2p] = ode45(@odefun,tspan,[0;0;0;0;1;0]);
33  [~,h3p] = ode45(@odefun,tspan,[0;0;0;0;0;1]);
```

```
34
35
36 clear U;
37
38 h1p = h1p(:,1:3);
39 h2p = h2p(:,1:3);
40 h3p = h3p(:,1:3);
41
42 D = zeros(1,N+1);
43 for i = 1:N+1
44     D(i) = det([h1p(i,:); h2p(i,:); h3p(i,:)]);
45 end
46 figure;
47 plot(t,D);
48
49 %choice for the tolerance
50 zero = 10e-3;
51
52 %if D is null only in some points, there is a conjugate point
53 if ~all(abs(D(2:N+1)) >= zero) && ~all(abs(D(2:N+1)) < zero)
54     %disp('1');
55     CPtmp = find(abs(D(2:N+1)) < zero, 1, 'first');
56     %if the previous point is also zero, this is not a conjugate point
57     if abs(D(CPtmp)) >= zero
58         CP = L/N*CPtmp;
59     end
60     clear CPtmp;
61 end
62
63 %if D crosses the horizontal ax, there is a conjugate point
64 if ~all(D>=0) && ~all(D<=0)
65     %disp('2');
66     if D(2)>0
67         CPtmp = find(D<0,1,'first');
68     else
69         CPtmp = find(D>0,1,'first');
70     end
71     CPtmp = -D(CPtmp-1)/(D(CPtmp)-D(CPtmp-1)) + CPtmp-1;
72     CPtmp = (CPtmp-1)*L/N;
73     if exist('CP','var')
74         CP = min(CP,CPtmp);
75     else
76         CP = CPtmp;
77     end
78
79 %if D is nonzero (but in 0) we don't have any conjugate point
80 elseif (all(D>=0) || all(D<=0)) && ~all(abs(D(2:N+1)) < zero)
81     %disp('3');
82     CP = L;
83
84 %if D is identically null, we can't say anything about the existence of
85 %conjugate points
86 elseif all(abs(D) < zero)
87     %disp('4');
88     CP = 0;
89 end
90
91 end
```

B Circular Case without Decoupling

```
1 function [ CP ] = stabCircle2( K1,K2,K3,A1,A2,A3,L )
2 %STABCIRCLE2 computes the first conjugate point in [0,L] of a circular rod
3 %of length L with diagonal stiffness matrix given by K1,K2,K3,A1,A2,A3. It
4 %returns L if there is no conjugate point
5 %   The rod is given by the strains
6 %   u = [c;0;0] and v = [0;0;1]
7 %   its reference state is given by
8 %   u_ref = [0;0;0] and v_ref = [0;0;1]
```

```
%   the determinant of the Jacobi fields is plotted.
%   We should have K2>K1 and the matrix K positive definite.

c = 2*pi/L;

P = diag([A1, A2, A3, K1, K2, K3]);
Q = [0  0  0  0  0  0;...
     0  c*c*A3  0  0  0  0;...
     0  0  c*c*A2 -c*A2  0  0;...
     0  0 -c*A2  A2  0  0;...
     0  0  0  0  c*c*(K3-K1)+A1  0;...
     0  0  0  0  0  c*c*(K2-K1)];
C = [0  0  0  0 -A1  0;...
     0  0 -c*A2  A2  0  0;...
     0  c*A3  0  0  0  0;...
     0  0  0  0  0  0;
     0  0  0  0  0  c/2*(K1-2*K2);...
     0  0  0  0  c/2*(2*K3-K1)  0];

invP = inv(P);

%We declare U as a global variable in order to use it to define the
%function odefun that we need to solve the ode
global U;
U =[-invP*C invP; Q-(C'*invP*C) C'*invP];

N = 100;%numbers of points -1 at which the determinant will be computed
tspan = 0:L/N:L;

%@odefun calls the function built with the global variable U that
%corresponds to the ode z'=Uz
[t,h1] = ode45(@odefun,tspan,[0;0;0;0;0;0;1;0;0;0;0;0]);
[~,h2] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;1;0;0;0;0]);
[~,h3] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;1;0;0;0]);
[~,h4] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;1;0;0]);
[~,h5] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;0;1;0]);
[~,h6] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;0;0;1]);

clear U;

h1 = h1(:,1:6);
h2 = h2(:,1:6);
h3 = h3(:,1:6);
h4 = h4(:,1:6);
h5 = h5(:,1:6);
h6 = h6(:,1:6);

D = zeros(1,N+1);
for i = 1:N+1
    D(i) = det([h1(i,:); h2(i,:); h3(i,:); h4(i,:); h5(i,:); h6(i,:)]);
end
figure;
plot(t,D,t,0*t);

%choice for the tolerance
zero = 10e-3;

%if D is null only in some points, there is a conjugate point
if ~all(abs(D(2:N+1)) >= zero) && ~all(abs(D(2:N+1)) < zero)
    %disp('1');
    CPtmp = find(abs(D(2:N+1)) < zero, 1, 'first');
    %if the previous point is also zero, this is not a conjugate point
    if abs(D(CPtmp)) >= zero
        CP = length/N*CPtmp;
    end
    clear CPtmp;
end

%if D crosses the horizontal ax, there is a conjugate point
if ~all(D>=0) && ~all(D<=0)
```

```
    %disp('2');
    if D(2)>0
        CPtmp = find(D<0,1,'first');
    else
        CPtmp = find(D>0,1,'first');
    end
    CPtmp = -D(CPtmp-1)/(D(CPtmp)-D(CPtmp-1)) + CPtmp-1;
    CPtmp = (CPtmp-1)*length/N;
    if exist('CP','var')
        CP = min(CP,CPtmp);
    else
        CP = CPtmp;
    end

%if D is nonzero (but in 0) we don't have any conjugate point
elseif (all(D>=0) || all(D<=0)) && ~all(abs(D(2:N+1)) < zero)
    %disp('3');
    CP = length;

%if D is identically null, we can't say anything about the existence of
%conjugate points
elseif all(abs(D) < zero)
    %disp('4');
    CP = 0;
end

end
```

C Unshearable Inextensible Case without Mesh

C.1 Equilibria

```
function [ u ] = equ_unsh_inext( K1,K2,K13,K23,u_ref,pitch,rad )
%EQU_UNSH_INEXT computes every unshearable inextensible helical rod at
%equilibrium with given pitch and radius, reference state given by the
%strain u_ref and symmetric stiffness matrix given by K1,K2,K3,K13,K23.
%The output is a matrix whose columns are the u-points i.e. each column is
%u1,u2,u3 for one equilibrium.
%   v_ref = [0;0;1]
%   Equilibrium conditions don't depend on K3.
%   We should have K2>K1 and K positive definite for some K3.

if K1 >= K2
    error('K1 must be smaller than K2');
end

tau = pitch/(rad*rad + pitch*pitch);
eta1 = 1/sqrt(rad*rad + pitch*pitch);

%constraints on u : <u,u> = eta1^2 and
u3 = tau;

a = (K1*u_ref(1) + K13*u_ref(3))/(K1-K2);
b = K13/(K2-K1);
c = (K2*u_ref(2) + K23*u_ref(3))/(K2-K1);
d = K23/(K1-K2);
x = a + b*u3;
y = c + d*u3;

%u1 is equal to the roots of the polynomial with the following coefficients
u1cpx = roots([1,...
    -2*x, x*x+y*y-u3*u3-eta1*eta1,...
    2*x*(2*y*y-eta1*eta1-u3*u3),...
    -x*x*(y*y+eta1*eta1+u3*u3)]);
j = 0;
for i = 1:length(u1cpx)
    if isreal(u1cpx(i))
        tmp = y + x*y/(u1cpx(i)-x);
        if isreal(tmp)
```

```
            j = j+1;
            u2(j) = tmp;
            u1(j) = u1cpx(i);
        end
    end
end

u = zeros(3,j);
for i = 1:j
    u(:,i) = [u1(i);u2(i);u3];
end

end
```

C.2 Stability

```
function [ cp ] = stab_unsh_inext( K1,K2,K3,K13,K23,u_ref,pitch,rad,L )
%STAB_UNSH_INEXT computes the first conjugate point in [0,L] of every
%helical rod at equilibrium with given pitch and radius and with reference
%state given by u_ref and symmetric positive definite stiffness matrix
%given by K1,K2,K3,K13,K23
%The output is a vector of the length equals to the number of equilibria
%   v_ref = [0;0;1];
%   u_ref is constant => the center line of the ref state is helical
%   The function also plots the determinant of the Jacobi fields in [0,L].
%   If there is no conjugate point in [0,L], the output is L
%   We should have K2>K1 and the matrix K positive definite.

K = [K1 0 K13;0 K2 K23;K13 K23 K3];

%computation of equilibria for the given parameters
us = equ_unsh_inext(K1,K2,K3,K13,K23,u_ref,pitch,rad);

N = 100;%numbers of points -1 at which the determinant will be computed
tspan = 0:L/N:L;
cp = zeros(1,size(us,2));

for i = 1:size(us,2)
    u = us(:,i);
    if u(1) == 0 && u(2) == 0
        break;
    end
    %m and n are constant and n parallel to u
    m = K*(u-u_ref);
    if u(1) == 0
        mu1 = m(2)/u(2);
    else
        mu1 = m(1)/u(1);
    end
    n = (m(3) - mu1*u(3))*u;
    v = [0;0;1];

    ux = vectCross(u);
    vx = vectCross(v);
    mx = vectCross(m);
    nx = vectCross(n);

    global U;
    U = [-ux, -vx, zeros(3,6);...
        zeros(3,3), 1/2*(K\mx)-ux, zeros(3,3), inv(K);...
        zeros(3,6), -ux, zeros(3,3);...
        zeros(3,3), 1/4*mx/K*mx-1/2*(vx*nx+nx*vx), -vx, 1/2*mx/K-ux];

    %@odefun calls the function built with the global variable U that
    %corresponds to the ode z'=Uz
    [t,z1] = ode45(@odefun,tspan,[0;0;0;0;0;0;1;0;0;0;0;0]);
    [~,z2] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;1;0;0;0;0]);
    [~,z3] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;1;0;0;0]);
    [~,z4] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;1;0;0]);
    [~,z5] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;0;1;0]);
```

```
    [~,z6] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;0;0;0;1]);

    h1 = z1(:,1:6);
    h2 = z2(:,1:6);
    h3 = z3(:,1:6);
    h4 = z4(:,1:6);
    h5 = z5(:,1:6);
    h6 = z6(:,1:6);

    clear U;

    D = zeros(1,N+1);
    for j = 1:N+1
        D(j) = det([h1(j,:); h2(j,:); h3(j,:); h4(j,:); h5(j,:); h6(j,:)]);
    end
    figure;
    plot(t,D,t,0*t);

    %choice for the tolerance
    zero = 10e-3;

    %if D is null only in some points, there is a conjugate point
    if ~all(abs(D(2:N+1)) >= zero) && ~all(abs(D(2:N+1)) < zero)
        %disp('1');
        CPtmp = find(abs(D(2:N+1)) < zero, 1, 'first');
        %if the previous point is also zero, this is not a conjugate point
        if abs(D(CPtmp)) >= zero
            CP = L/N*CPtmp;
        end
        clear CPtmp;
    end

    %if D crosses the horizontal ax, there is a conjugate point
    if ~all(D>=0) && ~all(D<=0)
        %disp('2');
        if D(2)>0
            CPtmp = find(D<0,1,'first');
        else
            CPtmp = find(D>0,1,'first');
        end
        CPtmp = -D(CPtmp-1)/(D(CPtmp)-D(CPtmp-1)) + CPtmp-1;
        CPtmp = (CPtmp-1)*L/N;
        if exist('CP','var')
            CP = min(CP,CPtmp);
        else
            CP = CPtmp;
        end

    %if D is nonzero (but in 0) we don't have any conjugate point
    elseif (all(D>=0) || all(D<=0)) && ~all(abs(D(2:N+1)) < zero)
        %disp('3');
        CP = L;

    %if D is identically null, we can't say anything about the existence of
    %conjugate points
    elseif all(abs(D) < zero)
        %disp('4');
        CP = 0;
    end
end

end
```

D Unshearable Inextensible Case with Mesh

D.1 Stability

```
function [ CP ] = stab_unsh_inext2( K1,K2,K3,K13,K23,u_ref,u,L )
%STAB_UNSH_INEXT2 computes the first conjugate point in [0,L] of the
```

```
%unshearable inextensible rod at equilibrium given by the strain u, having
%a reference state given by u_ref and a symmetric positive definite
%stiffness matrix given by K1,K2,K3,K13,K23.
%    v_ref = [0;0;1];
%    u_ref is constant => the center line of the ref state is helical
%    If there is no conjugate point in [0,L], the output is L.
%   We should have K2>K1 and the matrix K positive definite.

K = [K1 0 K13;0 K2 K23;K13 K23 K3];
N = 1000;%numbers of points -1 at which the determinant will be computed
tspan = 0:L/N:L;

%m and n are constant and n parallel to u
m = K*(u-u_ref);
if u(1) == 0
    mu1 = m(2)/u(2);
else
    mu1 = m(1)/u(1);
end
mu2 = (m(3) - mu1*u(3));
n = mu2*u;
v = [0;0;1];

ux = vectCross(u);
vx = vectCross(v);
mx = vectCross(m);
nx = vectCross(n);

invK = inv(K);
global U;
U = [-ux, -vx, zeros(3,6);...
     zeros(3,3), 1/2*(invK*mx)-ux, zeros(3,3), invK;...
     zeros(3,6), -ux, zeros(3,3);...
     zeros(3,3), 1/4*mx*invK*mx-1/2*(vx*nx+nx*vx), -vx, 1/2*mx*invK-ux];

%@odefun calls the function built with the global variable U that
%corresponds to the ode z'=Uz
[t,z1] = ode45(@odefun,tspan,[0;0;0;0;0;0;1;0;0;0;0;0]);
[~,z2] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;1;0;0;0;0]);
[~,z3] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;1;0;0;0]);
[~,z4] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;1;0;0]);
[~,z5] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;0;1;0]);
[~,z6] = ode45(@odefun,tspan,[0;0;0;0;0;0;0;0;0;0;0;1]);

h1 = z1(:,1:6);
h2 = z2(:,1:6);
h3 = z3(:,1:6);
h4 = z4(:,1:6);
h5 = z5(:,1:6);
h6 = z6(:,1:6);

D = zeros(1,N+1);
for j = 1:N+1
    D(j) = det([h1(j,:); h2(j,:); h3(j,:); h4(j,:); h5(j,:); h6(j,:)]);
end
clear U;
% figure;
% plot(t,D);

%choice for the tolerance
zero = 10e-3;

%if D is null only in some points, there is a conjugate point
if ~all(abs(D(2:N+1)) >= zero) && ~all(abs(D(2:N+1)) < zero)
    %disp('1');
    CPtmp = find(abs(D(2:N+1)) < zero, 1, 'first');
    %if the previous point is also zero, this is not a conjugate point
    if abs(D(CPtmp)) >= zero
        CP = L/N*CPtmp;
    end
```

```
    clear CPtmp;
end

%if D crosses the horizontal ax, there is a conjugate point
if ~all(D>=0) && ~all(D<=0)
    %disp('2');
    if D(2)>0
        CPtmp = find(D<0,1,'first');
    else
        CPtmp = find(D>0,1,'first');
    end
    CPtmp = -D(CPtmp-1)/(D(CPtmp)-D(CPtmp-1)) + CPtmp-1;
    CPtmp = (CPtmp-1)*L/N;
    if exist('CP','var')
        CP = min(CP,CPtmp);
    else
        CP = CPtmp;
    end

%if D is nonzero (but in 0) we don't have any conjugate point
elseif (all(D>=0) || all(D<=0)) && ~all(abs(D(2:N+1)) < zero)
    %disp('3');
    CP = L;

%if D is identically null, we can't say anything about the existence of
%conjugate points
elseif all(abs(D) < zero)
    %disp('4');
    CP = 0;
end

end
```

E Circular Inextensible Unshearable Case

```
function [ CP ] = stabCircleUnshInext( K1,K2,K3,L )
%STABCIRCLEUNSHINEXT computes the first conjugate point in [0,L] of a
%cirular unshearable inextensible rod of length L at equilibrium with a
%diagonal stiffness matrix given by K1,K2,K3.
%   The rod is given by the strains:
%   u = [c;0;0] and v = [0;0;1]
%   Its reference state is given by
%   u_ref = [0;0;0] and v_ref = [0;0;1]
%   The function also plots the determinant of the Jacobi fields in [0,L]
%   If there is no conjugate point in [0,L], the output is L.
%   We should have K2>K1 and the matrix K positive definite.

c = 2*pi/L;

N = 100;%numbers of points -1 at which the determinant will be computed
tspan = 0:L/N:L;

global U;
U =[0 1 0 0 0 0;...
    0 0 c*(2*K2-K1)/(2*K2) 0 1/K2 0;...
    0 -c*(2*K3-K1)/(2*K3) 0 0 0 1/K3;...
    0 0 0 0 0 0;...
    0 -c*c*K1*K1/(4*K3) 0 -1 0 c*(2*K3-K1)/(2*K3);...
    0 0 -c*c*K1*K1/(4*K2) 0 -c*(2*K2-K1)/(2*K2) 0];

%@odefun calls the function built with the global variable U that
%corresponds to the ode z'=Uz
[~,h1op] = ode45(@odefun,tspan,[0;0;0;1;0;0]);
[~,h2op] = ode45(@odefun,tspan,[0;0;0;0;1;0]);
[~,h3op] = ode45(@odefun,tspan,[0;0;0;0;0;1]);

clear U;
global U;

```

```
U =[0  c  -1  0  0   0;...
    -c  0  0  0  0   0;...
    0  0  0  0  0  1/K1;...
    0  0  0  0  c   0;...
    0  0  0 -c  0   0;...
    0  0  0  1  0   0];

%@odefun calls the function built with the global variable U that
%corresponds to the ode z'=Uz
[t,h1p] = ode45(@odefun,tspan,[0;0;0;1;0;0]);
[~,h2p] = ode45(@odefun,tspan,[0;0;0;0;1;0]);
[~,h3p] = ode45(@odefun,tspan,[0;0;0;0;0;1]);

clear U;

h1p = h1p(:,1:3);
h2p = h2p(:,1:3);
h3p = h3p(:,1:3);
h1op = h1op(:,1:3);
h2op = h2op(:,1:3);
h3op = h3op(:,1:3);

D = zeros(1,N+1);
for i = 1:N+1
    %D(i) = det([h1(i,:); h2(i,:); h3(i,:); h4(i,:); h5(i,:); h6(i,:)]);
    D(i) = det([h1p(i,:);h2p(i,:);h3p(i,:)])*det([h1op(i,:);h2op(i,:);h3op(i,:)]);
end
figure;
plot(t,D,t,0*t);

%choice for the tolerance
zero = 10e-3;

%if D is null only in some points, there is a conjugate point
if ~all(abs(D(2:N+1)) >= zero) && ~all(abs(D(2:N+1)) < zero)
    %disp('1');
    CPtmp = find(abs(D(2:N+1)) < zero, 1, 'first');
    %if the previous point is also zero, this is not a conjugate point
    if abs(D(CPtmp)) >= zero
        CP = L/N*CPtmp;
    end
    clear CPtmp;
end

%if D crosses the horizontal ax, there is a conjugate point
if ~all(D>=0) && ~all(D<=0)
    %disp('2');
    if D(2)>0
        CPtmp = find(D<0,1,'first');
    else
        CPtmp = find(D>0,1,'first');
    end
    CPtmp = -D(CPtmp-1)/(D(CPtmp)-D(CPtmp-1)) + CPtmp-1;
    CPtmp = (CPtmp-1)*L/N;
    if exist('CP','var')
        CP = min(CP,CPtmp);
    else
        CP = CPtmp;
    end

%if D is nonzero (but in 0) we don't have any conjugate point
elseif (all(D>=0) || all(D<=0)) && ~all(abs(D(2:N+1)) < zero)
    %disp('3');
    CP = L;

%if D is identically null, we can't say anything about the existence of
%conjugate points
elseif all(abs(D) < zero)
    %disp('4');
    CP = 0;
```

```
105 end
106
107 end
```

References

[1] N. Chouaieb, A. Goriely, and J. H. Maddocks. Helices. *PNAS*, 103(25):9398–9403, jun 2006.

[2] N. Chouaieb and J. H. Maddocks. Kirchhoff's problem of helical equilibria of uniform rods. *Journal of Elasticity*, 77:221–247, dec 2004.

[3] L. Cotta-Ramusino. *A path-integral formalism of DNA looping probability*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2008.

[4] L. Cotta-Ramusino and J. H. Maddocks. Looping probabilities of elastic chains: A path integral approach. *Physical Review*, 82(5):051924–1 – 051924–15, nov 2010.

[5] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Richard A. Silverman, 1963.

[6] G. Iffland and B. Zwahlen. Equations différentielles ordinaires. Ecole Polytechnique Fédérale de Lausanne, 1998. Notes rédigées d'après plusieurs cours de C.A. Stuart et B.Zwahlen.