A TOOL FOR RELAXING KNOTS, EMBEDDED CURVES, AND CONFINED RODS

S. BARTELS, P. FALK, P. WEYER

1.1. **Energy functional.** Given an arclength parametrized curve $u : I \to \mathbb{R}^3$ with $I = [0, L]$ we define its (total) energy as

$$E[u] = \frac{\kappa}{2} \int_I |u''(x)|^2 \,\mathrm{d}x + \varrho \mathrm{TP}[u] + \mu \mathrm{CP}[u].$$

The tangent-point functional $\mathrm{TP} : H^s(I; \mathbb{R}^3) \to \mathbb{R} \cup \{+\infty\}$ is finite on the subset of injective (embedded) curves and defined via

$$\mathrm{TP}[u] = \frac{2^{-q}}{q} \iint_{I \times I} \frac{1}{\mathbf{r}^q\big(u(y), u(x)\big)} \,\mathrm{d}x \,\mathrm{d}y,$$

where $2\mathbf{r}\big(u(x), u(y)\big) = |u(x) - u(y)| / \sin(u'(y), u(y) - u(x))$ creates a singularity in the integral for $u(x) = u(y)$ with $x \neq y$ whereas for $x \approx y$ we have that $\mathbf{r}$ approximates the inverse of the curvature of u at x, cf. Fig. 1.

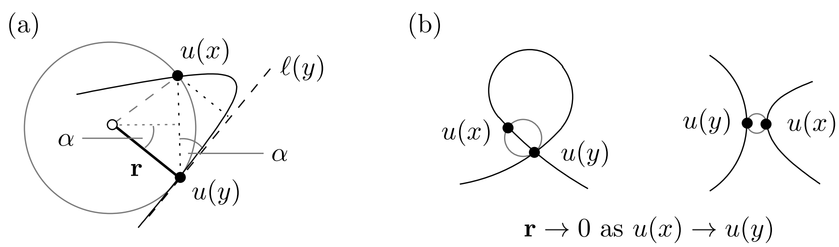

FIGURE 1. (a) Tangent-point function $\mathbf{r} = \mathbf{r}(u(y), u(x))$ for a curve u; (b) the radius $\mathbf{r}$ vanishes at self-intersections.

The confinement potential functional $\mathrm{CP} : C^0(I; \mathbb{R}^3) \to \mathbb{R}_{\geq 0}$ for a domain $D \subset \mathbb{R}^3$ with center of mass $m \in \mathbb{R}^3$ is defined as

$$\mathrm{CP}[u] = \frac{1}{2} \int_I \big(|u(x) - m|_D - 1\big)_+^2 \,\mathrm{d}x$$

for the semi-norm $|y|_D = \big(y^\top G_D \, y\big)^{1/2}$ induced by a positive semi-definite symmetric matrix G_D. It is zero if $u(x) \in D$ for all $x \in I$ and strictly positive otherwise. Configurations outside of D are penalized when the

Date: December 18, 2020.

energy is reduced by the gradient flow. We use a decomposition of the potential, i.e. the integrand of CP, into a quadratic and a concave part:

$$V_D(y) = \big(|y|_D - 1\big)_+^2 = |y|_D^2 + \begin{cases} -|y|_D^2, & \text{if } |y|_D \leq 1, \\ -2|y|_D + 1 & \text{else.} \end{cases}$$

for points $y = u(x) - m$. The second term is a concave function $V_{\rm cv} : \mathbb{R}^3 \to \mathbb{R}$. The nonnegative parameters κ, ϱ, and μ define weights for the influence of the bending energy, the self-avoidance potential, and the confinement energy. If $\kappa \gg \varrho$ we call minimizers of E *physical knots* while in the case $\varrho \gg \kappa$ we call them *mathematical knots*. The parameter ϱ defines a length-scale that determines how close a curve may be to itself. The parameter μ^{-1} determines a length-scale that describes the penetration depth of the curve in the space $\mathbb{R}^3 \backslash D$.
We are interested in computing (local) minimizers for E and evolutions defined via gradient flows for E. These tasks are directly linked since gradient flows lead to stationary configurations with low energy and conversely, gradient flows can be defined via incremental minimization problems. We will therefore consider one particular gradient flow defined by a metric related to mapping properties of the tangent-point functional TP.

1.2. H^s **gradient flow.** Given an inner product $(\cdot,\cdot)_*$ on $H^2(I;\mathbb{R}^3)$ the gradient flow for E in the class of arclength parametrized curves is defined via

$$(\partial_t u, w)_* = -\delta E[u][w].$$

The family of curves $u : [0,T] \to H^s(I;\mathbb{R}^3)$ satisfies the initial and arclength conditions

$$u(0,\cdot) = u_0, \quad |u'(t,\cdot)|^2 = 1,$$

for all $t \in [0,T]$. Moroever, we impose boundary conditions

$$L_{bc}[u(t,\cdot)] = \ell_{bc},$$

defined by a linear operator $L_{bc} : H^s(I;\mathbb{R}^3) \to \mathbb{R}^m$ and a vector $\ell_{bc} \in \mathbb{R}^m$. Examples are periodicity

$$u(t,0) = u(t,L), \quad u'(t,0) = u'(t,L),$$

or a clamped boundary condition at one end of the interval, e.g.,

$$u(t,0) = \mathfrak{p}_0, \quad u'(t,0) = \mathfrak{t}_0.$$

The test functions w satisfy the linearized conditions

$$w'(t,\cdot) \cdot u'(t,\cdot) = 0, \quad L_{bc}[w(t,\cdot)] = 0.$$

Letting $(\cdot,\cdot)$ denote the L^2 scalar product, the gradient flow thus determines a family of curves via the following problem:

$$(P)\quad \begin{cases} \text{Find } u:[0,T]\to H^s(I;\mathbb{R}^3) \text{ satisfying for all } t\in[0,T] \\ u(0,\cdot)=u_0,\ |u(t,\cdot)'|^2=1,\ L_{bc}[u(t,\cdot)]=\ell_{bc}, \\ \quad (\partial_t u,w)_* + \kappa(u'',w'') = -\varrho\,\delta\mathrm{TP}[u][w] - \mu\,\delta\mathrm{CP}[u][w] \\ \text{for all } w\in H^s(I;\mathbb{R}^3) \text{ with } w'\cdot u'(t,\cdot)=0,\ L_{bc}[w]=0. \end{cases}$$

Note that if $|u_0'|^2=1$ then the condition $|u'(t,\cdot)|^2=1$ is equivalent to the linearized condition $\partial_t u'(t,\cdot)\cdot u'(t,\cdot)=0$.

1.3. **Time-stepping scheme.** For a step size $\tau>0$ we define the backward difference quotient d_t on sequences $(u^k)_{k=0,\dots,K}$ via

$$d_t u^k = \frac{1}{\tau}\big(u^k - u^{k-1}\big).$$

We then use the semi-implicit time stepping scheme that determines iterates $(u^k)_{k=0,\dots,K}$ via the initialization $u^0=u_0$ and the recursion

$$\begin{aligned}(d_t u^k,w)_s + \kappa([u^k]'',w'') + \mu(u^k - m, G_D\,w)\\ = -\varrho\delta\mathrm{TP}[u^{k-1}][w] - \mu(\nabla V_{\mathrm{cv}}(u^{k-1}),w)\end{aligned}$$

subject to the boundary condition $L_{bc}[u^k]=\ell_{bc}$. Here, ∇V_{cv} denotes the gradient of V_{cv}. The quadratic part of the confinement energy is evaluated at step k, whereas we use the concave part of the previous step $k-1$. We assume that the initial u_0 is such that

$$L_{bc}[u_0]=\ell_{bc},\quad |u_0'|^2=1.$$

The linearized arclength condition is discretized via

$$[d_t u^k]'\cdot[u^{k-1}]'=0.$$

We abbreviate $v^k=d_t u^k$ and note that

$$u^k = u^{k-1}+\tau v^k,$$

and $L_{bc}[u^k]=\ell_{bc}$ for $k=1,2,\dots,K$ is equivalent to $L_{bc}[v^k]=0$ for $k=1,2,\dots,K$. Our time-stepping scheme reads as follows:

$$(P_\tau)\quad \begin{cases} \text{Find } (u^k)_{k=0,\dots,K}\subset H^s(I;\mathbb{R}^3) \text{ satisfying for } k=1,2,\dots,K \\ u^0=u_0,\ d_t u^k = v^k,\ [v^k]'\cdot[u^{k-1}]'=0,\ L_{bc}[v^k]=0, \\ \quad (v^k,w)_* + \tau\kappa([v^k]'',w'') + \tau\mu(v^k,G_D\,w) \\ = -\kappa([u^{k-1}]'',w'') - \varrho\,\delta\mathrm{TP}[u^{k-1}][w] - \mu\,\delta\mathrm{CP}[u^{k-1}][w] \\ \text{for all } w\in H^s(I;\mathbb{R}^3) \text{ with } [w]'\cdot[u^{k-1}]'=0,\ L_{bc}[w]=0. \end{cases}$$

The violation of the arclength condition is controlled by the step size, i.e., because of the ortogonality $[v^k]' \cdot [u^{k-1}]' = 0$ we have

$$|[u^k]'|^2 - 1 = |[u^{k-1} + \tau v^k]'|^2 - 1 = |[u^{k-1}]'|^2 - 1 + \tau^2 |[v^k]'|^2$$
$$= \cdots = |[u^0]'|^2 - 1 + \tau^2 \sum_{\ell=1}^{k} |[v^\ell]'|^2 = \mathcal{O}(\tau),$$

since $|[u^0]'|^2 = 1$ and provided that $\tau \sum_{\ell=1}^{K} |[v^\ell]'|^2$ is bounded.

1.4. **H^2-conforming finite elements.** For a partition $\mathcal{T}_h$ of $I = [0, L]$ into subintervals $I_i = [z_{i-1}, z_i]$, $i = 1, 2, \dots, N$ with with nodes $0 = z_0 < z_1 < \cdots < z_N = L$ and maximal meshsize $h = \max_{i=1,\dots,N} h_i$ for

$$h_i = z_i - z_{i-1},$$

we use the space of continuously differentiable cubic spline functions

$$\mathcal{S}^{3,1}(\mathcal{T}_h) = \big\{ v_h \in C^1(I) : v_h|_{I_i} \in \mathcal{P}_3(I_i),\ i = 1, 2, \dots, N \big\}.$$

A basis for V_h is defined with the reference interval $\widehat{I} = [0, 1]$ and the polynomials

$$\widehat{p}_{0,0}(\widehat{x}) = 1 - 3\widehat{x}^2 + 2\widehat{x}^3, \quad \widehat{p}_{0,1}(\widehat{x}) = \widehat{x} - 2\widehat{x}^2 + \widehat{x}^3,$$
$$\widehat{p}_{1,0}(\widehat{x}) = 3\widehat{x}^2 - 2\widehat{x}^3, \quad \widehat{p}_{1,1}(\widehat{x}) = -\widehat{x}^2 + \widehat{x}^3,$$

cf. Fig. 2(a).

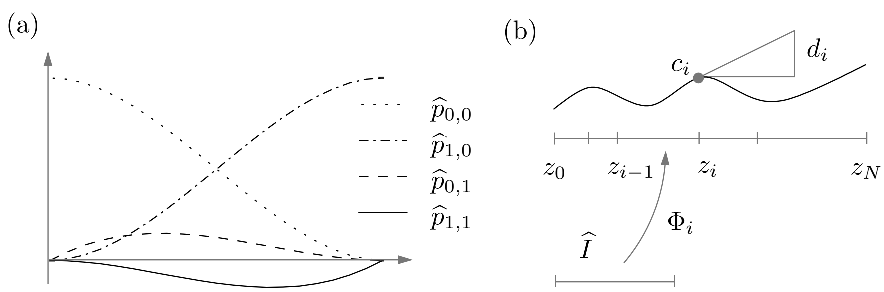

FIGURE 2. (a) Cubic basis functions on $\widehat{I} = [0, 1]$; (b) affine transformation and nodal values.

With the affine transformations

$$\Phi_i : \widehat{I} \to I_i, \quad \widehat{x} \mapsto x = z_{i-1} + \widehat{x}(z_i - z_{i-1}),$$

and the convention $I_0 = I_{N+1} = \emptyset$ we define basis functions $(p_{i,0} : i = 0, 1, \dots, N) \cup (p_{i,1} : i = 0, 1, \dots, N)$ via

$$p_{i,0} = \begin{cases} \widehat{p}_{0,0} \circ \Phi_i^{-1}, & \text{in } I_{i+1}, \\ \widehat{p}_{1,0} \circ \Phi_{i-1}^{-1}, & \text{in } I_i, \\ 0, & \text{elsewhere}, \end{cases} \qquad p_{i,1} = \begin{cases} h_i \widehat{p}_{0,1} \circ \Phi_i^{-1}, & \text{in } I_{i+1}, \\ h_{i-1} \widehat{p}_{1,1} \circ \Phi_{i-1}^{-1}, & \text{in } I_i, \\ 0, & \text{elsewhere}. \end{cases}$$

For $i, j, m, n = 0, 1$ we have the Kronecker property that

$$p_{i,m}^{(n)}(z_j) = \delta_{ij}\delta_{mn},$$

which implies that for a function $r_h \in \mathcal{S}^{3,1}(\mathcal{T}_h)$ we have

$$r_h(z) = \sum_{i=0}^{N} c_i p_{i,0}(z) + \sum_{i=0}^{N} d_i p_{i,1}(z)$$

if and only if the nodal values satisfy

$$r_h(z_i) = c_i, \quad r_h'(z_i) = d_i$$

for $i = 0, 1, \ldots, N$. Correspondingly, a piecewise cubic curve

$$u_h \in V_h = \left[\mathcal{S}^{3,1}(\mathcal{T}_h)\right]^3 \subset C^1(I; \mathbb{R}^3)$$

is defined by positions $(\mathfrak{p}_i)_{i=0,\ldots,N} \subset \mathbb{R}^3$ and tangents $(\mathfrak{t}_i)_{i=0,\ldots,N} \in \mathbb{R}^3$ via

$$u_h(z) = \sum_{i=0}^{N} \mathfrak{p}_i \, p_{i,0}(z) + \sum_{i=0}^{N} \mathfrak{t}_i \, p_{i,1}(z).$$

1.5. **Fully discrete scheme.** With the finite element space $V_h \subset C^1(I; \mathbb{R}^3)$ we define the spatially discrete time-stepping scheme as follows:

$$(P_{\tau,h}) \begin{cases} \text{Find } (u_h^k)_{k=0,\ldots,K} \subset V_h \text{ satisfying for } k = 1, 2, \ldots, K \\ u_h^0 = u_{0,h}, \ d_t u_h^k = v_h^k, \ [v_h^k]'(z_i) \cdot [u_h^{k-1}]'(z_i) = 0, \ L_{bc,h}[v_h^k] = 0, \\ \quad (v_h^k, w_h)_{*,h} + \kappa\tau([v_h^k]'', w_h'') + \mu\tau(v_h^k, G_D\, w_h)_h \\ = -\kappa([u_h^{k-1}]'', w_h'') - \varrho\, \delta\mathrm{TP}_h[u_h^{k-1}][w_h] - \mu\, \delta\mathrm{CP}_h[u_h^{k-1}][w_h] \\ \text{for all } w_h \in V_h \text{ with } [w_h]'(z_i) \cdot [u^{k-1}]'(z_i) = 0, \ L_{bc,h}[w_h] = 0. \end{cases}$$

Here, $u_{0,h} \in V_h$ is the interpolant of u_0, $L_{bc,h}$ is a discretization of the boundary conditions such that $L_{bc,h}[u_{0,h}] = \ell_{bc}$, and $(\cdot,\cdot)_{*,h}$ is an inner product on V_h. Moreover, the linearized arclength-condition is imposed only at the nodes $(z_i)_{i=0,\ldots,N}$. To guarantee good stability properties of the scheme we choose a discrete inner product that is continuous with respect to the norm in $H^s(I; \mathbb{R}^3)$, i.e., we use

$$(v_h, w_h)_{*,h} = (v_h, w_h) + h^r(v_h'', w_h''),$$

for a suitably chosen parameter r. For $r < 0$, we set h^r to zero.

1.6. **Discretized tangent-point functional.** Discretzing the functional TP requires the use of suitable quadrature and treatment of the singular part. We note that we have

$$\mathrm{TP}(u) = \frac{1}{q} \iint_{I \times I} \lambda_u(x, y)\, \mathrm{d}x\, \mathrm{d}y, \quad \lambda_u(x, y) = \frac{|u'(y) \wedge [u(x) - u(y)]|}{|u(x) - u(y)|^2}.$$

For a disjoint pair of intervals I_j, I_k and a continous function $\phi : I_j \times I_k \to \mathbb{R}$ we let $\mathcal{Q}_h \phi$ be the average of the values of ϕ at the vertices of $I_j \times I_k$, i.e.,

$$\mathcal{Q}_h \phi|_{I_j \times I_k} = \frac{1}{4}\big(\phi(z_{j-1}) + \phi(z_j) + \phi(z_{k-1}) + \phi(z_k)\big).$$

If $I_j \cap I_k \neq \emptyset$ then we set $\mathcal{Q}_h \phi|_{I_j \times I_k} = 0$. With $\varepsilon \geq 2h$ this leads to the approximation

$$\mathrm{TP}_{\varepsilon,h}[u] = \sum_{\substack{j,k=1,\ldots,N \\ I_j \times I_k \subset \mathcal{R}^c_{\varepsilon,h}}} h_j h_k \mathcal{Q}_h \lambda_u,$$

where $\mathcal{R}_{\varepsilon,h}$ is a neighbourhood of the diagonal in $I \times I$ defined with the midpoints $(m_j)_{j=1,\ldots,N}$ of the intervals $(I_j)_{j=1,\ldots,N}$ via

$$\mathcal{R}^c_{\varepsilon,h} = \bigcup \big\{ I_j \times I_k : j,k = 1,2,\ldots,N,\ |m_j - m_k| \geq \varepsilon \big\}.$$

The variation $\delta\mathrm{TP}[u][w]$ is approximated similarly. We set

$$\delta\mathrm{TP}_{\varepsilon,h}[u][w] = \mathcal{M}_{\varepsilon,h}(u;u,w) + \mathcal{M}_{\varepsilon,h}(u;w,u) - 2\mathcal{A}_{\varepsilon,h}(u;u,w),$$

with the mappings $\mathcal{M}_{\varepsilon,h}$ and $\mathcal{A}_{\varepsilon,h}$ defined by

$$\mathcal{M}_{\varepsilon,h}(u;v,w) = \iint_{\mathcal{R}^c_{\varepsilon,h}} \mathcal{Q}_h \Phi_u(x,y) \cdot \big(v'(y) \wedge (w(x) - w(y))\big) \,\mathrm{d}x\,\mathrm{d}y,$$
$$\mathcal{A}_{\varepsilon,h}(u;v,w) = \iint_{\mathcal{R}^c_{\varepsilon,h}} \mathcal{Q}_h \Psi_u(x,y) \big(v(x) - v(y)\big) \cdot \big(w(x) - w(y)\big) \,\mathrm{d}x\,\mathrm{d}y,$$

for functions Φ_u and Ψ_u given by

$$\Phi_u(x,y) = |u'(y) \wedge (u(x) - u(y))|^{q-2} \frac{u'(y) \wedge (u(x) - u(y))}{|u(x) - u(y)|^{2q}},$$
$$\Psi_u(x,y) = \frac{|u'(y) \wedge (u(x) - u(y))|^q}{|u(x) - u(y)|^{2q+2}}.$$

Note that setting $\mathcal{Q}_0 = \mathrm{id}$ we have $\delta\mathrm{TP}[u] = \delta\mathrm{TP}_{0,0}[u]$.

1.7. **Discrete confinement potential.** The confinement function CP is discretized by using mass lumping. We define the lumped L^2-product $(.,.)_h$ as

$$(u,v)_h = \int_I \mathcal{I}_{1,h}(uv) \,\mathrm{d}x.$$

This integral can be exactly calculated by using the trapezoidal quadrature. The discrete variation $\delta\mathrm{CP}_h$ can be expressed as

$$\delta\mathrm{CP}_h[u_h, w_h] = \left((|u_h - m|_D - 1)_+ \frac{G_D(u_h - m)}{|u_h - m|_D}, w_h \right)_h,$$

where the first factor represents the gradient of the potential V_D evaluated at $u_h - m$.

1.8. **Discrete reparametrization.** Given a regular curve $\widetilde{u}:\widetilde{I}=[0,\widetilde{L}]\to\mathbb{R}^3$, an equivalent arclength parametrized curve $u:I\to\mathbb{R}^3$ is given by

$$u(x)=\widetilde{u}\circ\psi^{-1}(x),\quad \psi(y)=\int_0^y|\widetilde{u}'(s)|\,\mathrm{d}s$$

for $x\in I=[0,L]$ with $L=\psi(\widetilde{L})$. This follows from noting $\psi'(y)=|\widetilde{u}'(y)|>0$ and $[\psi^{-1}]'=1/(\psi'\circ\psi^{-1})$, so that

$$u'=\big(\widetilde{u}'\circ\psi^{-1}\big)\big[\psi^{-1}\big]'=\frac{\widetilde{u}'\circ\psi^{-1}}{|\widetilde{u}'\circ\psi^{-1}|},$$

i.e., $|u'(x)|=1$ for all $x\in I$. For a partition $0=y_0<y_1<\dots<y_N=\widetilde{L}$ of $\widetilde{I}$ we set $z_i=\psi(y_i)$ and obtain the nodal values

$$u(z_i)=\widetilde{u}(y_i),\quad u'(z_i)=\frac{\widetilde{u}'(y_i)}{|\widetilde{u}'(y_i)|},$$

for $i=0,1,\dots,N$. Practically, the function ψ is evaluated via quadrature, i.e., $z_0=0$ and

$$z_i=z_{i-1}+\frac{|y_i-y_{i-1}|}{2}\big(|\widetilde{u}'(y_{i-1})|+|\widetilde{u}'(y_i)|\big),$$

for $i=1,2,\dots,N$.

1.9. **Computation of auxiliary tangents.** If only values $(u_i)_{i=0,\dots,N}\subset\mathbb{R}^3$ are given to define a discrete initial curve, we set $z_0=0$ and $u_0'=e_1$ and

$$z_i=z_{i-1}+|u_i-u_{i-1}|,\quad u_i'=\frac{u_i-u_{i-1}}{|u_i-u_{i-1}|},$$

for $i=1,2,\dots,N$. Note that the piecewise Lagrange interpolant u_h belongs to $W^{1,\infty}(I;\mathbb{R}^3)$ and is parametrized by arclength. An arclength parametrized interpolant in $H^2(I;\mathbb{R}^3)$ would require a more complicated definition. Better constructions can be obtained using central or other higher order difference quotients such as $u_i'=(4u_{i+1}-u_{i+2}-3u_i)/(2h)$.

1.10. **Data structures.** A discrete curve is defined via the arrays

$$\texttt{c4n}=\begin{bmatrix}z_0\\z_1\\\vdots\\z_N\end{bmatrix}\in\mathbb{R}^{n_C},\quad \texttt{pos}=\begin{bmatrix}\mathfrak{p}_0^\mathsf{T}\\\mathfrak{p}_1^\mathsf{T}\\\vdots\\\mathfrak{p}_N^\mathsf{T}\end{bmatrix}\in\mathbb{R}^{n_C\times 3},\quad \texttt{tang}=\begin{bmatrix}\mathfrak{t}_0^\mathsf{T}\\\mathfrak{t}_1^\mathsf{T}\\\vdots\\\mathfrak{t}_N^\mathsf{T}\end{bmatrix}\in\mathbb{R}^{n_C\times 3},$$

with $n_C=N+1$. For the derivation of the linear system of equations the arrays `pos` and `tan` are arranged in the vector $\texttt{u}$ of length $6n_C$, i.e.,

$$\texttt{u}=\big[\mathfrak{p}_{0,1}\,\mathfrak{t}_{0,1}\,\mathfrak{p}_{1,1}\,\mathfrak{t}_{1,1}\,\dots\,\mathfrak{p}_{N,1}\,\mathfrak{t}_{N,1}\,\mathfrak{p}_{0,2}\,\mathfrak{t}_{0,2}\,\dots\,\mathfrak{p}_{N,2}\,\mathfrak{t}_{N,2}\,\mathfrak{p}_{0,3}\,\mathfrak{t}_{0,3}\,\dots\,\mathfrak{p}_{N,3}\,\mathfrak{t}_{N,3}\big]^\mathsf{T}.$$

The first $2n_C=2(N+1)$ entries define the first component $u_{h,1}$ of the function u_h via

$$u_{h,1}=\sum_{i=0}^N \texttt{u}_i p_i,$$

where $p_i = p_{i/2,0}$ if i is even and $p_i = p_{(i-1)/2,1}$ if i is odd.

1.11. **Linear systems.** Every time step requires solving a linear system of equations with saddle-point structure, i.e.,

$$\begin{bmatrix} A & B^\top \\ B & \end{bmatrix} \begin{bmatrix} v \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

where $A \in \mathbb{R}^{6n_C \times 6n_C}$ is the symmetric and positive definite matrix defined by

$$\begin{aligned} A_{i+(r-1)2n_C, j+(s-1)2n_C} &= (e_r p_i, e_s q_j)_* + \kappa\tau([e_r p_i]'', [e_s q_j]'') \\ &\quad + \mu\tau(e_r p_i, G_D e_s q_j)_h \\ &= \delta_{rs}(p_i, q_j)_* + \kappa\tau\delta_{rs}([p_i]'', [q_j]'') \\ &\quad + \mu\tau\, [e_r^\top G_D e_s](p_i, q_j)_h, \end{aligned}$$

for $i, j = 1, 2, \dots, 2n_C$ and $r, s = 1, 2, 3$, assuming that the inner product $(\cdot,\cdot)_*$ is induced from a scalar variant. The right-hand side vector $b \in \mathbb{R}^{6n_C}$ is given by

$$b_{i+(r-1)2n_C} = -\kappa([u_h^{k-1}]'', [e_r p_i]'') - \varrho\delta \mathrm{TP}_h[u_h^{k-1}][e_r p_i] - \mu\delta \mathrm{CP}_h[u_h^{k-1}][e_r p_i].$$

The constraint matrix B contains the tangent vector from the previous time step and the boundary conditions, e.g., in the case of periodic boundary conditions we have

$$B = \left[\begin{array}{cccc|cccc|cccc} t^{k-1}_{0,1} & & & & t^{k-1}_{0,2} & & & & t^{k-1}_{0,3} & & & \\ & \ddots & & & & \ddots & & & & \ddots & & \\ & & & t^{k-1}_{N,1} & & & & t^{k-1}_{N,2} & & & & t^{k-1}_{N,3} \\ \hline 1 & & -1 & & & & & & & & & \\ & 1 & & -1 & & & & & & & & \\ \hline & & & & 1 & & -1 & & & & & \\ & & & & & 1 & & -1 & & & & \\ \hline & & & & & & & & 1 & & -1 & \\ & & & & & & & & & 1 & & -1 \end{array}\right]$$

Because of the identity $u_h'(z_0) = u_h'(z_N)$ in the periodic setting the first line in B can be omitted which then leads to a matrix of full rank. To avoid the use of a direct solver for the indefinite linear system we use an ADMM iteration in every time step. This determines the solution $[v, \lambda]$ of the saddle-point system

$$\min_{v \in \mathbb{R}^{6nC}} \max_{\lambda \in \mathbb{R}^\ell} \frac{1}{2} v^\top A v - b^\top v + \lambda^\top B x + \frac{\delta}{2}\|Bx\|^2$$

via the following iteration.

Algorithm 1.1 (ADMM iteration)**.** *Choose* $\lambda \in \mathbb{R}^\ell$ *and* $\delta > 0$.
(1) Solve $(A + \delta B^\top B)x = b - B^\top \lambda$.
(2) Update $\lambda = \lambda + \delta B x$.
(3) Stop if $\|Bx\| \le \varepsilon_{\mathrm{stop}}$.

The symmetric positive definite matrix $A+\delta B^{\mathsf{T}}B$ is inverted using a Cholesky factorization.

1.12. **Using KNOTEVOLVE.** The previously described ADMM algorithm is implemented in `GO` source code. The user interface is accessible under

`aam.uni-freiburg.de/knotevolve`,

which loads a self-avoiding trefoil knot without confinement by default. The initial conditions can be specified by appending the following strings to the URL.

(1) `/03-Trefoil`, `/04-Figure 8`, `/05-1 Cinquefoil`, `/05-2 Three-twist knot`, `/06-1 Stevedore`, `/06-2`, `/06-3`, `/07_001`, `...`, `/07_007`, `/08_001`, `...`, `/08_021`, `/09_001`, `...`, `/09_049`, `/10_001`, `...`, `/10_166` to load the specified knot.
(2) `/torus-p-q-n`, where `p`, `q`, and `n` are positive integers, to load a torus knot with degrees p and q ($\gcd(p,q) = 1$ is necessary!) and n discrete nodes.
(3) `/braid-t-s-x`, where `t` is a braid string (e.g. AAA or aaa for the trefoil knot, AbAb for the figure-eight knot etc.), `s` is a positive real number indicate the distance of two adjacent strands, and `x` is the distance of two crossing strands.

In addition, the URL can be amended by a list of `option=value` pairs. The list needs to be started with a question mark `?`, whereas multiple options are joined by an ampersand `&`. The following groups of options are available (default values in square brackets):

(1) For all curves:
- `Tmax=`T: maximal number of time steps [10000]
- `StepW=`τ: time step width [0.01]
- `StepWxHmax=`{`true,false`}: whether the time step width is given as a multiple of the largest spatial grid width h [true]
- `Kappa=`κ: bending rigidity [10]
- `R=`r: norm exponent for $(\partial_t u, v) + h^r([\partial_t u]'', v'')$. Please note: If $r < 0$, the term h^r is set to zero to yield the L^2 metric instead of a H^s metric. [-0.1]

(2) For loading a parametric curve: `/parametric` together with
- `eq_x="formula"&eq_y="formula"&eq_z="formula"`: strings that contain the parametric equations for the three coordinates [equations for an overhand knot]
- `eq_nC=`N: number of discrete nodes [100]
- `param_bctype=`{`0,1,2,3,4,5`}: boundary condition (0: periodic, 1: clamped/free, 2: clamped/fixed, 3: clamped/clamped with both ends possibly moving, 4: fixed/fixed, 5: free/free) [5]
- `VDiri0=`v_0`&VDiriEnd=`v_L: speed of movement in the direction of the tangents at beginning and end of curve for BC type 3 [0,0].

(3) For the tangent-point potential: `Rho=`ϱ/κ`&Ell=`$q/2$: as defined in the description of TP [10 and 1.5]
(4) For the confinement:
  - `CnfmType=`{`none,ellipsoid,cylinder-x,cylinder-y,cylinder-z,box`}: shape of the confinement domain [none]
  - `CnfmMu=`μ/κ: strength [10]
  - `CnfmRadius=`R_x`,`R_y`,`R_z: size [4,4,4]
  - `CnfmOffset=`m_x`,`m_y`,`m_z: center of mass [0,0,0]

1.13. **Exemplary** KNOTEVOLVE **calls.**

- `https://aam.uni-freiburg.de/knotevolve/03-Trefoil`
- `https://aam.uni-freiburg.de/knotevolve/torus-2-3-100?Tmax=2000`
- `https://aam.uni-freiburg.de/knotevolve/braid-AAA-0.15-0.05?Rho=0&StepW=0.1&StepWxHmax=false`
- `https://aam.uni-freiburg.de/knotevolve/parametric?eq_x=sin(6pi*x)&eq_y=cos(6pi*x)&eq_z=2x&eq_nC=50&param_bctype=5`
- `https://aam.uni-freiburg.de/knotevolve/parametric?param_bctype=3&VDiri0=-1&VDiriEnd=1`
- `https://aam.uni-freiburg.de/knotevolve/torus-1-5-101?Rho=0&CnfmType=ellipsoid&CnfmRadius=3,3,3`
- `https://aam.uni-freiburg.de/knotevolve/parametric?Rho=0&eq_x=5sin(6pi*x)&eq_y=5cos(6pi*x)&eq_z=8x-4&eq_nC=200&param_bctype=5&CnfmType=box`