ISSN 0005-1179, Automation and Remote Control, 2009, Vol. 70, No. 4, pp. 633–643. © Pleiades Publishing, Ltd., 2009.
Original Russian Text © A.A. Ardentov, Yu.L. Sachkov, 2009, published in Avtomatika i Telemekhanika, 2009, No. 4, pp. 78–88.

SYSTEMS WITH LUMPED PARAMETERS

Solution to Euler's Elastic Problem[1]

A. A. Ardentov and Yu. L. Sachkov

Program Systems Institute, Russian Academy of Sciences, Pereslavl-Zalesskii, Russia

October 1, 2008

Abstract—Euler's problem on stationary configurations of elastic rod with fixed endpoints and tangents at the endpoints is considered. The corresponding optimal control problem is reduced to several systems of algebraic equations in Jacobi's functions. An algorithm and software for solving the optimal control problem are constructed.

PACS number: 02.30.Yy

DOI: 10.1134/S0005117909040092

1. EULER'S ELASTIC PROBLEM: HISTORY, STATEMENT, APPLICATIONS

In 1744 Leonhard Euler considered the following problem on stationary configurations of elastic rod [1]. Given an elastic rod in the plane with fixed endpoints and tangents at the endpoints, one should find possible profiles of the rod with given boundary conditions. Euler derived differential equations for stationary configurations of a rod and described their possible qualitative types. These configurations are called Euler's elasticae.

In this work we reduce Euler's problem to solving certain set of algebraic equations in Jacobi's functions obtained at the basis of works [2–5]. An algorithm of finding a numerical solution is constructed. A software in Mathematica [6] is constructed for solving Euler's problem, results of work of this software are presented. Perspectives of a parallel algorithm and its supercomputer realization are discussed.

The problem on configuration of elastic rod has a rich history related to Jacob Bernoulli (1691) [7], Daniel Bernoulli (1742) [8], Leonhard Euler (1744) [1], Max Born (1906) [9]. A detailed description of this history is presented in [10–12].

This work is based on detailed study of the elastic problem performed in [2, 4, 5] via methods of geometric control theory [13].

Euler elasticae and their natural generalisations have important applications in mechanics, engineering, control theory, approximation theory, molecular biology, nanotechnologies [14–27]. Thus a complete solution to Euler's problem presented in this work seems reasonable.

We recall the exact statement of elastic problem. Consider a homogeneous elastic rod of fixed length $l > 0$ in a two-dimensional plane $\mathbb{R}^2$. Choose any points $a_0,\ a_1 \in \mathbb{R}^2$ and arbitrary unit tangent vectors at these points $v_i \in T_{a_i}\mathbb{R}^2$, $|v_i| = 1$, $i = 0, 1$. One should find the form of the rod $\gamma : [0, t_1] \to \mathbb{R}^2$, starting from the point a_0 and coming to the point a_1 with the corresponding tangent vectors v_0 and v_1 (see Fig. 1).

[1] This work was supported by the Russian Foundation for Basic Research, project no. 06-01-00330.

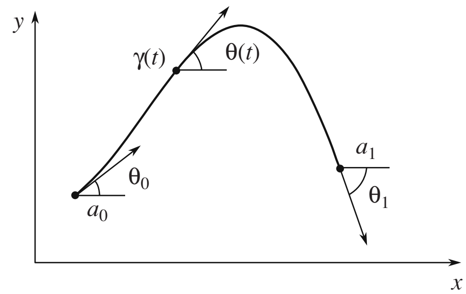

Fig. 1. Statement of Euler's problem.

The elastic problem is formalised as the following optimal control problem (see [2, 28]):

$$\dot{x} = \cos\theta, \qquad \dot{y} = \sin\theta, \qquad \dot{\theta} = u, \tag{1}$$

$$q = (x, y, \theta) \in M = \mathbb{R}^2_{x,y} \times S^1_\theta, \qquad u \in \mathbb{R}, \tag{2}$$

$$q(0) = q_0 = (0, 0, 0), \qquad q(1) = q_1 = (x_1, y_1, \theta_1), \tag{3}$$

$$J = \frac{1}{2}\int_0^1 u^2(t)\, dt \to \min, \tag{4}$$

$$u(\cdot) \in L_2[0, 1], \qquad q(\cdot) \in AC[0, 1], \tag{5}$$

without loss of generality we assume that the elastic rod has a unit length. Integral (4) has the sense of elastic energy of the rod: we look for a rod of minimum elastic energy under fixed boundary conditions (3).

As shown in [2], for this problem the reachable set from the point $q_0 = (0, 0, 0)$ for time 1 has the form $\mathcal{A} = \{(x, y, \theta) \in M \mid x^2 + y^2 < 1 \text{ or } (x, y, \theta) = (1, 0, 0)\}$.

2. REDUCTION OF EULER'S PROBLEM TO SYSTEMS OF EQUATIONS

We describe reduction of the optimal control problem (1)–(5) to solving several systems of equations in Jacobi's functions.

2.1. Exponential Mapping

Via Pontryagin maximum principle [29], extremal trajectories in problem (1)–(5) are parametrised by Jacobi's functions [30]. The family of extremal trajectories is parametrised by points (β, c) of the phase cylinder $C = S^1_\beta \times \mathbb{R}_c$ of the mathematical pendulum $\dot{\beta} = c$, $\dot{c} = -\sin\beta$. The total energy of the pendulum $E = \frac{c^2}{2} - \cos\beta$ defines decomposition of the cylinder $C = \bigcup_{i=1}^{5} C_i$ (see Fig. 2), where

$$C_1 = \{(\beta, c) \in C \mid E \in (-1, 1)\}, \tag{6}$$

$$C_2 = \{(\beta, c) \in C \mid E \in (1, +\infty)\}, \tag{7}$$

$$C_3 = \{(\beta, c) \in C \mid E = 1,\ \beta \neq \pi\}, \tag{8}$$

$$C_4 = \{(\beta, c) \in C \mid E = -1\}, \tag{9}$$

$$]C_5 = \{(\beta, c) \in C \mid E = 1,\ \beta = \pi\}. \tag{10}$$

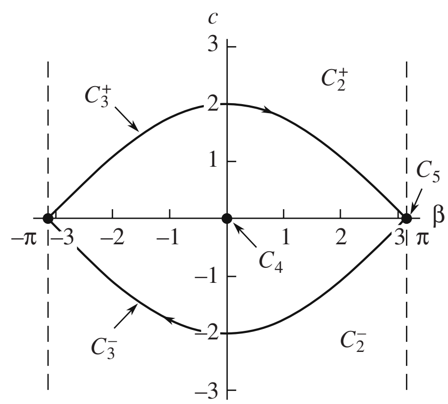

Fig. 2. Decomposition of the phase cylinder of pendulum.

In [4, 31] we obtained an explicit parametrisation of extremal trajectories in Euler's problem by Jacobi's functions cn, sn, dn, E [30]:

$$x = x(\lambda, t), \quad y = y(\lambda, t), \quad \theta = \theta(\lambda, t), \qquad \lambda \in N = T^*_{q_0} M \simeq \mathbb{R}^3, \quad t \in \mathbb{R}. \tag{11}$$

Here and below λ denotes the triple of adjoint variables of Pontryagin maximum principle; $N \simeq \mathbb{R}^3$ is the space of conjugate variables.

Consider the exponential mapping for the unit time:

$$\operatorname{Exp} : N \to M, \qquad \operatorname{Exp}(\lambda) = q(\lambda, 1) = (x(\lambda, 1), y(\lambda, 1), z(\lambda, 1)).$$

This mapping transforms the vector of adjoint variables $\lambda \in N \simeq \mathbb{R}^3$ into the endpoint of the corresponding extremal trajectory at the instant $t = 1$. Explicit formulas for the functions (x, y, θ) for the domains C_1, C_2 have the following form (here we use special elliptic coordinates k, φ, ψ, that rectify the equation of pendulum [4, 14, 31]).

If $(\beta, c) \in C_1$, then

$$\sin \frac{\theta_t}{2} = k \operatorname{dn}(\sqrt{r}\varphi) \operatorname{sn}(\sqrt{r}\varphi_t) - k \operatorname{sn}(\sqrt{r}\varphi) \operatorname{dn}(\sqrt{r}\varphi_t), \tag{12}$$

$$\cos \frac{\theta_t}{2} = \operatorname{dn}(\sqrt{r}\varphi) \operatorname{dn}(\sqrt{r}\varphi_t) + k^2 \operatorname{sn}(\sqrt{r}\varphi) \operatorname{sn}(\sqrt{r}\varphi_t), \tag{13}$$

$$\begin{aligned} x_t = {} & \frac{2}{\sqrt{r}} \operatorname{dn}^2(\sqrt{r}\varphi)(\operatorname{E}(\sqrt{r}\varphi_t) - \operatorname{E}(\sqrt{r}\varphi)) \\ & + \frac{4k^2}{\sqrt{r}} \operatorname{dn}(\sqrt{r}\varphi) \operatorname{sn}(\sqrt{r}\varphi)(\operatorname{cn}(\sqrt{r}\varphi) - \operatorname{cn}(\sqrt{r}\varphi_t)) \\ & + \frac{2k^2}{\sqrt{r}} \operatorname{sn}^2(\sqrt{r}\varphi)(\sqrt{r}t + \operatorname{E}(\sqrt{r}\varphi) - \operatorname{E}(\sqrt{r}\varphi_t)) - t, \end{aligned} \tag{14}$$

$$\begin{aligned} y_t = {} & \frac{2k}{\sqrt{r}}(2 \operatorname{dn}^2(\sqrt{r}\varphi) - 1)(\operatorname{cn}(\sqrt{r}\varphi) - \operatorname{cn}(\sqrt{r}\varphi_t)) \\ & - \frac{2k}{\sqrt{r}} \operatorname{sn}(\sqrt{r}\varphi) \operatorname{dn}(\sqrt{r}\varphi)(2(\operatorname{E}(\sqrt{r}\varphi_t) - \operatorname{E}(\sqrt{r}\varphi)) - \sqrt{r}t). \end{aligned} \tag{15}$$

If $(\beta, c) \in C_2$, then

$$\sin\frac{\theta_t}{2} = \pm(\operatorname{cn}(\sqrt{r}\psi)\operatorname{sn}(\sqrt{r}\psi_t) - \operatorname{sn}(\sqrt{r}\psi)\operatorname{cn}(\sqrt{r}\psi_t)), \tag{16}$$

$$\cos\frac{\theta_t}{2} = \operatorname{cn}(\sqrt{r}\psi)\operatorname{cn}(\sqrt{r}\psi_t) + \operatorname{sn}(\sqrt{r}\psi)\operatorname{sn}(\sqrt{r}\psi_t), \tag{17}$$

$$x_t = \frac{1}{\sqrt{r}}(1 - 2\operatorname{sn}^2(\sqrt{r}\psi))\left(\frac{2}{k}(\operatorname{E}(\sqrt{r}\psi_t) - \operatorname{E}(\sqrt{r}\psi)) - \frac{2-k^2}{k^2}\sqrt{r}t\right) + \frac{4}{k\sqrt{r}}\operatorname{cn}(\sqrt{r}\psi)\operatorname{sn}(\sqrt{r}\psi)(\operatorname{dn}(\sqrt{r}\psi) - \operatorname{dn}(\sqrt{r}\psi_t)), \tag{18}$$

$$y_t = \pm\left(\frac{2}{k\sqrt{r}}(2\operatorname{cn}^2(\sqrt{r}\psi) - 1)(\operatorname{dn}(\sqrt{r}\psi) - \operatorname{dn}(\sqrt{r}\psi_t)) - \frac{2}{\sqrt{r}}\operatorname{sn}(\sqrt{r}\psi)\operatorname{cn}(\sqrt{r}\psi)\left(\frac{2}{k}(\operatorname{E}(\sqrt{r}\psi_t) - \operatorname{E}(\sqrt{r}\psi)) - \frac{2-k^2}{k^2}\sqrt{r}t\right)\right). \tag{19}$$

2.2. Optimality of Elasticae

In [4, 5] we obtained bounds on the instant of time when extremal trajectories lose optimality (i.e., cut time t_{cut}), of the form $t_{\text{cut}}(\lambda) \leqslant \mathbf{t}(\lambda)$, $\lambda \in N$, in terms of a function $\mathbf{t} : N \to (0, +\infty]$, described explicitly in [4, 5]. Thus in order to find optimal elasticae, it suffices to consider the restriction of the exponential mapping $\operatorname{Exp} : N' \to \mathcal{A}$, where $N' = \{\lambda \in N \mid \mathbf{t}(\lambda) \leqslant 1\}$.

2.3. Global Structure of Exponential Mapping

Consider the decomposition $N' = \bigcup_{i=1}^{4} L_i$, where L_i are the subsets in N' corresponding to four quadrants $\{(c, \theta) \mid \operatorname{sgn} c = \pm 1, \operatorname{sgn}\theta = \pm 1\}$ at the phase cylinder of the pendulum C (see Fig. 3). At this figure the surface $p = p_1^{\text{MAX}}$ that bounds the domains L_i from above corresponds to the equality $\mathbf{t}(\lambda) = 1$.

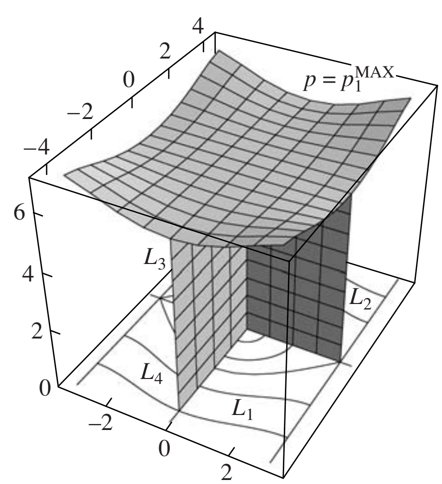

Fig. 3. Decomposition in preimage of exponential mapping.

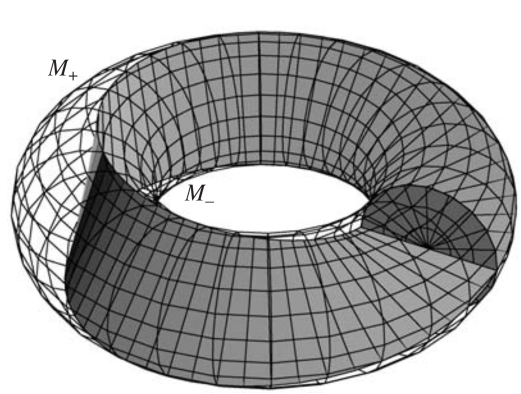

Fig. 4. Decomposition in image of exponential mapping.

Consider also the decomposition of the solid torus

$$\operatorname{int}\mathcal{A} = \{(x, y, \theta) \mid x^2 + y^2 < 1, \quad \theta \in (0, 2\pi)\}$$

into the subsets $\operatorname{int}\mathcal{A} = M_+ \cup M_- \cup M_0$ according to the sign of the function $P = x \sin\frac{\theta}{2} - y \cos\frac{\theta}{2}$:

$$\begin{aligned} M_+ &= \{(x, y, \theta) \in \operatorname{int}\mathcal{A} \mid P > 0, \quad \theta \in (0, 2\pi)\}, \\ M_- &= \{(x, y, \theta) \in \operatorname{int}\mathcal{A} \mid P < 0, \quad \theta \in (0, 2\pi)\}, \\ M_0 &= \{(x, y, \theta) \in \operatorname{int}\mathcal{A} \mid P = 0, \quad \theta \in (0, 2\pi)\} \end{aligned}$$

(see Fig. 4).

It follows from the results obtained in [2, 4, 5] that

$$\operatorname{Exp}(L_1) \subset M_-, \quad \operatorname{Exp}(L_3) \subset M_-, \quad \operatorname{Exp}(L_2) \subset M_+, \quad \operatorname{Exp}(L_4) \subset M_+,$$

moreover, each of the mappings

$$\operatorname{Exp}: L_1 \to M_-, \quad \operatorname{Exp}: L_3 \to M_-, \quad \operatorname{Exp}: L_2 \to M_+, \quad \operatorname{Exp}: L_4 \to M_+$$

is a diffeomorphism, i.e., a smooth bijection with a smooth inverse mapping.

3. ALGORITHM OF SOLUTION TO ELASTIC PROBLEM

3.1. Reduction to Systems of Equations

On the basis of the above results, we obtain the following algorithm for solving the optimal control problem (1)–(5). Choose an arbitrary terminal point $q_1 = (x_1, y_1, \theta_1) \in M$.

(1) If $q_1 \notin \mathcal{A}$, then the problem has no solution; thus we assume in the sequel that $q_1 \in \mathcal{A}$.

(2) If $q_1 = (1, 0, 0)$, then the required elastica is a line: $(x, y) = (t, 0)$. We assume in the sequel that $q_1 \neq (1, 0, 0)$.

(3) We have $q_1 \in \operatorname{int}\mathcal{A} = M_+ \cup M_- \cup M_0$. We determine, which of the sets M_+ M_-, M_0 contains the point q_1. In a generic case we can suppose that $q_1 \notin M_0$. If $q_1 \in M_-$, then we find solutions to the equations:

$$\operatorname{Exp}(\lambda_1) = q_1, \quad \lambda_1 \in L_1, \quad q_1 \in M_-, \tag{20}$$
$$\operatorname{Exp}(\lambda_2) = q_1, \quad \lambda_2 \in L_3, \quad q_1 \in M_-, \tag{21}$$

each of these equations has a unique solution. In the case $q_1 \in M_+$ we find solutions to the equations:

$$\operatorname{Exp}(\lambda_1) = q_1, \quad \lambda_1 \in L_2, \quad q_1 \in M_+, \tag{22}$$
$$\operatorname{Exp}(\lambda_2) = q_1, \quad \lambda_2 \in L_4, \quad q_1 \in M_+, \tag{23}$$

which also has a unique solution.

(4) From the values computed λ_1, λ_2 we find the corresponding controls $u_i(t)$ via formulas of [4], and compute the value of the elastic energy functional (5) for the corresponding elasticae $J_i = \int_0^1 u_i^2(t)\, dt$. The optimal one is the control $u_i(t)$ with the less value of functional J_i.

3.2. Solution to the Boundary Value Problem in the Domain L_i

Despite theoretically each of the systems (20)–(23) has a unique solution, a practical computation of this solution constitutes a nontrivial problem for the following reasons. First, the functions x, y, θ (11) parametrising the exponential mapping are given by non-elementary functions—Jacobi's functions. Second, in different domains C_i (see (6)–(10)) these functions x, y, θ are given by different formulas (see (12)–(19)). Thus the direct application of standard numerical schemes of solving systems of nonlinear equations (e.g., in Mathematica [6]) does not provide solutions to the systems (20)–(23).

The authors developed and realized in Mathematica system a special algorithm that provides a stable solution to these systems of equations and thus gives a complete solution to Euler's problem. This algorithm uses a certain non-trivial combination of standard methods of solving equations: Newton's method, the secant method, method of random choice with fixed boundaries, and the method of uniform choice.

We describe the algorithm of solving the system of Eqs. (20); the rest systems (21)–(23) are solved similarly.

The algorithm starts from an arbitrary point $q_1 \in M_-$. It follows from results of [2, 4, 5] that there exists a unique point $\lambda_1 \in L_1 = L_1^1 \cup L_1^2 \cup L_1^3$, for which the system (20) holds. Here and below

$$L_i = L_i^1 \cup L_i^2 \cup L_i^3,$$
$$L_i^1 = L_i \cap \{E < 1\}, \quad L_i^2 = L_i \cap \{E > 1\}, \quad L_i^3 = L_i \cap \{E = 1\}.$$

The set L_1^3 has measure zero, thus we can assume that $\lambda_1 \in L_1^1 \cup L_1^2$.

(1) Search of root of system (20) in domain L_1^1.

(1.1) We choose randomly two points $\lambda_1^1,\ \lambda_1^2 \in L_1^1$. We start the secant method of solving system (20) with the use of these points as initial ones, as a result we obtain a point $\widehat{\lambda_1^0}$. If $\widehat{\lambda_1^0} \in L_1^1$, then we set $\widetilde{\lambda_1^0} = \widehat{\lambda_1^0}$. If $\widehat{\lambda_1^0} \notin L_1^1$, then coordinates of the point $\widehat{\lambda_1^0}$ are normalised (via symmetries of the exponential mapping) to the segments that define the domain L_1^1. Denote this transformation ρ and set $\widetilde{\lambda_1^0} = \rho(\widehat{\lambda_1^0})$.

(1.2) We start Newton's method with the initial point $\widetilde{\lambda_1^0}$. As a result we obtain the point $\widehat{\lambda_1^1}$. Similarly to the previous point, $\widehat{\lambda_1^1}$ is transformed to $\widetilde{\lambda_1^1}$ via ρ.

(1.3) We compare the points $\widetilde{\lambda_1^0}$ and $\widetilde{\lambda_1^1}$ in respect of system (20): we compute the difference $\delta = |\operatorname{Exp}(\widetilde{\lambda_1^0}) - q_1| - |\operatorname{Exp}(\widetilde{\lambda_1^1}) - q_1|$.

(1.4) If $\delta > 0$, then we pass to item (1.2) with a shift of indices of $\widetilde{\lambda_1^i}$. Otherwise we pass to the next item.

(1.5) If the point obtained $\widetilde{\lambda_1^i}$ satisfies system (20) up to accuracy required, then the required root $\lambda_1 = \widetilde{\lambda_1^i}$ was found in the domain L_1^1, and the algorithm stops. Otherwise the root was not found in the domain L_1^1, and we pass to item (2).

(2) Search of the root of system (20) in the domain L_1^2. We fulfil items (2.1)–(2.5), completely similar to items (1.1)–(1.5), replacing the domain L_1^1 by the domain L_1^2.

(3) If the root λ_1 was found, it is returned as a result. Otherwise we pass to item (1).

3.3. Solving the Boundary Value Problem in Domain $L_i \cup L_j$

The scheme described in Subsection 3.2 works if we know, which of the domains L_i contains the required root. But in this problem we have two domains L_i and L_j (more precisely, L_1 and L_3,

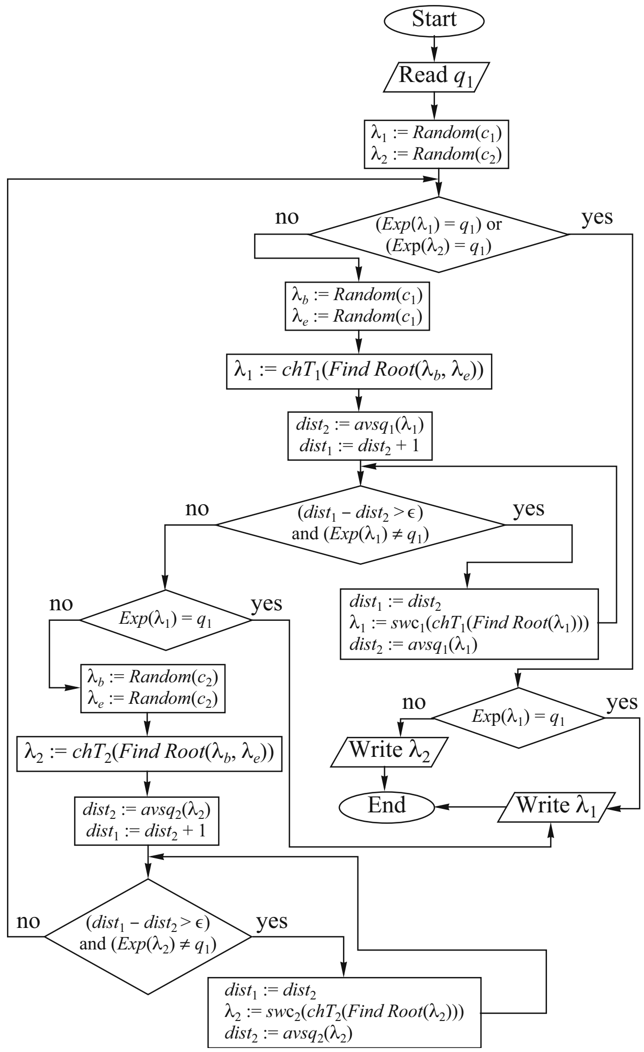

Fig. 5. Scheme of the program `FindElastic`.

or L_2 and L_4), moreover, the root is contained in one of these domains. If we start the search of the root first in one domain, and then in another sequentially, it may happen that the first domain does not contain the root, and the search will not be successful.

Thus one has to perform parallel computations. In order to realize parallel computations in a sequential way, one has to discretise computations and fulfil them in turn: first in the domain L_i, then in L_j, then once more in L_i etc. In this problem there is a natural decomposition of the algorithm into steps (the iterations of the algorithm described in Subsection 3.2).

So first we perform one iteration in the domain L_i. If the root was found, the computation terminates, and if not, we start an iteration in L_j. If the root was not found, we start once more in L_i etc, till we found the root in one of the domains.

A complete flow chart of the algorithm of solving of the elastic problem is presented at Fig. 5.

4. SOFTWARE REALIZATION

4.1. Description of the Program `FindElastic`

We developed a program `FindElastic` in Mathematica [6] that realizes the above algorithm of search of optimal elastica. The program starts from the triple $(x, y, \theta) = q_1$ and the format of the graphic file (eps, jpeg, png, gif, etc). The program returns a file of the indicated format with a plot of the elastica that solves the problem (1)–(5). In the Mathematica notebook there are computed the parameters corresponding to the optimal elastica.

The program is rather large, so we describe only the main its features.

(1) The main function is `FindElastic`, which takes a point (x, y, $cosThetaDT$, $sinThetaDT$) and an index of domain $area$ (corresponding to L_{area}) and computes a tuple (u, v, $b0$, k, r, $whichCase$, $energy$) that contains a complete information on the roots λ_1 or λ_2 of system (20)–(23) (depending on the domain $area$ where the choice was performed);

(2) The secant method and Newton's method are realized by the standard function `FindRoot`, moreover, the secant method is launched with 300 iterations, Newton's method is launched with 50 iterations, while at the last launch Newton's method works 300 iterations. Accuracy of computations for these methods is defined by the constants ag1 and wp1, which can be changed if necessary in the code of the program;

(3) The transformation ρ described in the algorithm was additionally developed while realized in software. If it transforms a result of computation of the secant method, then only one of the functions `chT1` or `chT2` is launched, depending on the parameter $area$. If it transforms a result of computation of Newton's method, then in addition to these functions, there is also launched one of the functions `swc1` or `swc2` (depending on $area$);

(4) The functions `cht1`, `cht2` perform the following transformations with coordinates of the vector $\widehat{\lambda_k^i}$: if v is the coordinate being transformed, and $vmax$ and $vmin$ are its maximum and minimum defining the required domain L_{area}^k, then the result $Mod[v, vmax - vmin] + vmin$ is returned. This notation is equivalent to the following:

$$For[, v >= vmax, v- = vmax - vmin]; \tag{24}$$

$$For[, v < vmin, v+ = vmax - vmin]. \tag{25}$$

Despite compactness of the first notation, the second notation is more reliable and visual in practice.

(5) The functions `swc1`, `swc2` perform sequential choice on a grid in the domain of choice. In other words, to a triple (u, v, k) there is put into correspondence the tuple $\{(u, v+ei\times j_1, k+ei\times j_2) \in L_{area}^k \,|\, j_1, j_2 \in \mathbb{Z}\}$, where ei is a constant, step of the grid. From this set there is chosen a point, for which the distance $|\,\mathrm{Exp}(u, v + ei \times j_1, k + ei \times j_2) - q_1|$ is minimum;

(6) The distance $|\,\mathrm{Exp}(\cdot) - q_1|$ is computed via the functions `avsq1` and `avsq2`;

(7) The functions `inR1`, `inR2` check whether a point belongs to the corresponding domain;

(8) The functions `switchArea1`, `switchArea2` transform the input point (x, y, $cosThetaDT$, $sinThetaDT$) so that we could apply the same formulas for exponential mapping (on the basis of its symmetries);

(9) The function `switchSearch` transforms current boundaries to boundaries of a given domain L_{area}^k.

4.2. Results of Work of Software

The program `FindElastic` was thoroughly tested at the supercomputer skif.botik.ru [32]. Here are results of a typical testing:

- Number of correct roots found: 2500 of 2500.
- Average number of attempts: 2.6144.
- Maximum number of attempts: 908.
- Time spent: 46 h 7 min 45.8 s.

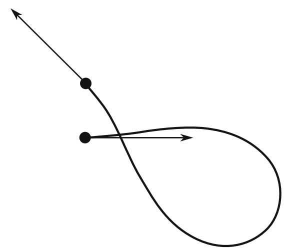

Fig. 6. Optimal elastica.

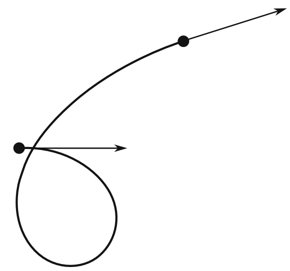

Fig. 7. Optimal elastica.

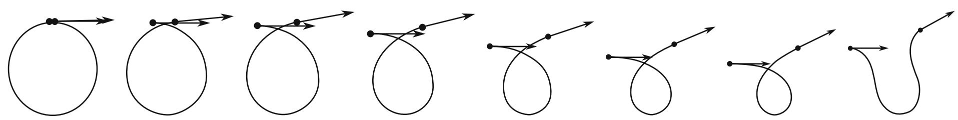

Fig. 8. Shots of animation with optimal elastica.

• Average time for a test: 1 min 6.42632 s.

Figures 6 and 7 present examples of optimal elasticae computed by the program `FindElastic`.

We developed also a program that constructs animation of a family of optimal elasticae corresponding to changing boundary conditions. Examples of a sequence of figures in such animation are shown at Fig. 8.

The animations and other results of work of the programs related to elasticae are presented in Internet [33].

In [34] was described application of the program `FindElastic` for construction of the cut locus in Euler's problem (i.e., the set of points where elasticae lose optimality).

4.3. Parallel Version of the Program

In the future we plan to develop parallel versions of the above programs in the system gridMathematica—a parallel version of the system Mathematica. Here are some possible ways of development of the parallel programs:

(1) one often has to compute several solutions to the problem (e.g., to construct animation), then one has a natural approach to parallelism—parallelism w.r.t. data;

(2) if necessary, it is possible to accelerate computation of the main function `findElastic`, then it suffices to launch the function on several nodes; the result is the first successful termination of the program, all the rest launches are stopped. Notice that in item (1) one can expect acceleration close to linear, but in this item acceleration will be much less;

(3) one can also distribute search among several nodes. In other words, the space of search can be cut into several parts, and each of the parts can be computed at a separate node.

Notice that item (2) can be essentially developed by choosing particular configuration of choice for each node. For example, we can choose to apply a larger number of iterations for the secant method at a certain node, and choose a larger number for Newton's method at another node etc. A similar modification can be applied to item (3), where the space can be divided into subsets in various ways. In item (1) one can send two or three problems to a node etc.

Application of a parallel program can be useful for the sequential version. If the state space is divided into a grid with small subdomains, one can use results of the preceding computation as an initial value for the next computation (by continuity).

5. CONCLUSION

Application of geometric control theory methods proved effective for complete theoretical study of Euler's elastic problem.

The algorithm and software developed at the basis of these results give a complete practical solution to Euler's problem.

This approach will be applied to several other control problems: sub-Riemannian problem on the group of motions of a plane, the problem on rolling a sphere on a plane [22], the nilpotent sub-Riemannian problem with the growth vector (2,3,5) [14–17], sub-Riemannian problem on Engel's group.

REFERENCES

1. Euler, L., *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes, sive solutio problematis isoperimitrici latissimo sensu accepti*, Lausanne: Bousquet, 1744.
2. Sachkov, Yu.L., Optimality of Euler's Elasticae, *Dokl. Ross. Akad. Nauk*, 2007, vol. 417, no. 1, pp. 23–25.
3. Sachkov, Yu.L., Controllability and Optimal Control for Invariant Systems on Lie Groups and Homogeneous Spaces, *Doctoral (Phys.-Math.) Dissertation*, Moscow: Inst. Mat., Ross. Akad. Nauk, 2008.
4. Sachkov, Yu.L., Maxwell Strata in Euler's Elastic Problem, *J. Dynam. Control Syst.*, 2008, vol. 14, no. 2, pp. 169–234.
5. Sachkov, Yu.L., Conjugate Points in Euler's Elastic Problem, *J. Dynam. Control Syst.*, 2008, vol. 14, no. 3, pp. 409–439.
6. Wolfram, S., *Mathematica: A System for Doing Mathematics by Computer*, Reading: Addison-Wesley, 1991.
7. Bernoulli, J., Véritable hypothèse de la résistance des solides, avec la demonstration de la corbure des corps qui font ressort, in *Collected Works*, Geneva, 1744, vol. 2.
8. Bernoulli, D., 26th Letter to L. Euler (October, 1742), in Fuss, P.H., *Correspondance mathématique et physique*, St. Petersburg, 1843, vol. 2.
9. Born, M., Stabilität der elastischen Linie in Ebene und Raum, in *Preisschrift und Dissertation*, Göttingen: Dieterichsche Universitäts-Buchdruckerei Göttingen, 1963, vol. 1, pp. 5–101.
10. Love, A.E.H., *A Treatise on the Mathematical Theory of Elasticity*, New York: Dover, 1927.
11. Truesdell, C., The Influence of Elasticity on Analysis: The Classic Heritage, *Bull. AMS*, 1983, vol. 9, no. 3, pp. 293–310.
12. Timoshenko, S., *History of Strength of Materials*, New York: McGraw-Hill, 1953.
13. Agrachev, A.A. and Sachkov, Yu.L., *Control Theory from the Geometric Viewpoint*, Berlin: Springer-Verlag, 2004.
14. Sachkov, Yu.L., Exponential Mapping in Generalized Dido Problem, *Mat. Sb.*, 2003, vol. 194, no. 9, pp. 63–90. Translated into English in *Sbornik: Mathematics*, 2003, vol. 194.
15. Sachkov, Yu.L., Discrete Symmetries in the Generalized Dido Problem, *Mat. Sb.*, 2006, vol. 197, no. 2, pp. 95–116. Translated into English in *Sbornik: Mathematics*, 2006, vol. 197, no. 2, pp. 235–257.
16. Sachkov, Yu.L., The Maxwell Set in the Generalized Dido Problem, *Mat. Sb.*, 2006, vol. 197, no. 4, pp. 123–150. Translated into English in *Sbornik: Mathematics*, 2006, vol. 197, no. 4, pp. 595–621.
17. Sachkov, Yu.L., Complete Description of the Maxwell Strata in the Generalized Dido Problem, *Mat. Sb.*, 2006, vol. 197, no. 6, pp. 111–160. Translated into English in *Sbornik: Mathematics*, 2006, vol. 197, no. 6, pp. 901–950.
18. Birkhoff, G. and de Boor, C.R., Piecewise Polynomial Interpolation and Approximation, in *Proc. Symp. Approxim. Functions*, Amsterdam: Elsevier, 1965, pp. 164–190.

19. Golumb, M. and Jerome, J., Equilibria of the Curvature Functional and Manifolds of Non-linear Interpolating Spline Curves, *SIAM J. Math. Anal.*, 1982, vol. 13, pp. 421–458.

20. Jerome, J.W., Minimization Problems and Linear and Nonlinear Spline Functions, I: Existence, *SIAM J. Numer. Anal.*, 1973, vol. 10, pp. 808–819.

21. Jerome, J.W., Smooth Interpolating Curves of Prescribed Length and Minimum Curvature, *Proc. AMS*, 1975, vol. 51, pp. 62–66.

22. Jurdjevic, V., The Geometry of the Ball-plate Problem, *Arch. Rat. Mech. Anal.*, 1993, vol. 124, pp. 305–328.

23. Jurdjevic, V., Non-Euclidean Elastica, *Am. J. Math.*, 1995, vol. 117, pp. 93–125.

24. Linnér, A., Unified Representations of Non-linear Splines, *J. Approx. Theory*, 1996, vol. 84, pp. 315–350.

25. Manning, R.S., Maddocks, J.H., and Kahn, J.D., A Continuum Rod Model of Sequence-dependent DNA Structure, *J. Chem. Phys.*, 1996, vol. 105, pp. 5626–5646.

26. Manning, R.S., Rogers, K.A., and Maddocks, J.H., Isoperimetric Conjugate Points with Application to the Stability of DNA Minicircles, *Proc. Royal Soc. London, Series A*, 1998, vol. 454, pp. 3047–3074.

27. Mumford, D., Elastica and Computer Vision, in *Algebraic Geometry and Its Applications*, Bajaj, C.L., Ed., New York: Springer-Verlag, 1994, pp. 491–506.

28. Jurdjevic, V., *Geometric Control Theory*, Cambridge: Cambridge Univ. Press, 1997.

29. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., and Mishchenko, E.F., *The Mathematical Theory of Optimal Processes*, New York: Wiley Interscience, 1962.

30. Whittaker, E.T. and Watson, G.N., *A Course of Modern Analysis: An Introduction to the General Theory of Infinite Processes and of Analytic Functions; with an Account of Principal Transcendental Functions*, Cambridge: Cambridge Univ. Press, 1996.

31. Ardentov, A.A., Exremal Curves in Euler's Elastic Problem, in *Proc. Conf. on Program Systems: Theory and Applications*, Pereslavl-Zalesskii, 2006, Moscow: Fizmatlit, 2006, vol. 2, pp. 23–37.

32. http://skif.pereslavl.ru/skif/.

33. http://www.botik.ru/PSI/CPRC/sachkov/GROUP/group.html.

34. Ardentov, A.A., Cut Locus in Euler's Elastic Problem, in *Proc. Conf. on Program Systems: Theory and Applications*, Pereslavl-Zalesskii, 2008, vol. 2, pp. 58–66.

This paper was recommended for publication by A.P. Kurdyukov, a member of the Editorial Board