R. Hiptmair
L. Scarabosio
C. Urzua Torres

Spring Term 2015

Numerical Methods for Partial Differential Equations

ETH Zürich
D-MATH

Homework Problem Sheet 2

Problem 2.1 A Curve with Tension

In [NPDE, Section 1.3] we examined the energy minimization problem for an elastic string in an external force field. We faced a minimization problem on a space of curves. A similar problem is investigated in this task in order to practice the derivation of variational equations and of the associated two-point boundary value problems.

Given a curve $\mathbf{u} \in V := (\mathcal{C}^1_{\mathrm{pw}}([0,1]))^2$, its tension energy is proportional to the square of its length:

$$J_{\mathrm{tens}}(\mathbf{u}) := \left(\int_0^1 \|\mathbf{u}'(\xi)\| \, \mathrm{d}\xi \right)^2 . \tag{2.1.1}$$

On the curve acts as an external force field $\mathbf{f} = \mathbf{f}(\xi)$, which results in a potential energy contribution according to

$$J_{\mathrm{f}}(\mathbf{u}) := - \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) \, \mathrm{d}\xi \tag{2.1.2}$$

The curve seeks to attain a shape such that its total energy $J(\mathbf{u}) = J_{\mathrm{tens}}(\mathbf{u}) + J_{\mathrm{f}}(\mathbf{u})$ becomes minimal. In addition, it is pinned at its ends, that is, we have to deal with the constraints

$$\mathbf{u}(0) = \mathbf{u}_0, \qquad \mathbf{u}(1) = \mathbf{u}_1. \tag{2.1.3}$$

We follow the approach of [NPDE, Section 1.3.1] to determine which variational problem needs to be solved to compute the shape of the curve.

(2.1a) Determine the *test space* V_0, i.e. the functional space of all admissible variations $\mathbf{v}$.

HINT: Remember that any direction/variation $\mathbf{v} \in V_0$ must vanish wherever the argument function $\mathbf{u}$ is *fixed*, because the test space has to be a vector space. Look up [NPDE, Def. 1.3.16] again.

Solution: We have $V_0 = (\mathcal{C}^1_{0,\mathrm{pw}}([0,1]))^2 := \left\{ \mathbf{v} \in (\mathcal{C}^1_{\mathrm{pw}}([0,1]))^2 : \mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0} \right\}$.

(2.1b) Following [NPDE, Section 1.3.1], compute the *directional/configurational derivative* of $J_{\mathrm{f}} = J_{\mathrm{f}}(\mathbf{u})$ in direction $\mathbf{v} \in V_0$ at a generic curve $\mathbf{u}$ that is compute

$$\lim_{t \to 0} \frac{J_f(\mathbf{u} + t\mathbf{v}) - J_{\mathrm{f}}(\mathbf{u})}{t} .$$

HINT: Observe that J_{f} is a *linear* functional and that the derivative of a linear mapping is easy to compute.

Solution: We have:

$$\lim_{t\to 0}\frac{J_f(\mathbf{u}+t\mathbf{v})-J_{\mathrm{f}}(\mathbf{u})}{t} = -\lim_{t\to 0}\frac{1}{t}\int_0^1 \mathbf{f}(\xi)\cdot t\mathbf{v}(\xi)\,\mathrm{d}\xi = -\int_0^1 \mathbf{f}(\xi)\cdot\mathbf{v}(\xi)\,\mathrm{d}\xi.$$

(2.1c) Compute the *directional derivative* of $J_{\mathrm{tens}} = J_{\mathrm{tens}}(\mathbf{u})$ in direction $\mathbf{v}\in V_0$ at a generic curve shape $\mathbf{u}$, that is, compute

$$\lim_{t\to 0}\frac{J_{\mathrm{tens}}(\mathbf{u}+t\mathbf{v})-J_{\mathrm{tens}}(\mathbf{u})}{t},$$

for any $\mathbf{v}\in V$.

HINT: It is strongly recommended to study [NPDE, § 1.3.3] before tackling this problem. In particular [NPDE, Eq. (1.3.5)] will be useful. The identity $A^2-B^2=(A-B)(A+B)$ may also come handily.

Solution: The perturbation analysis for the tension energy J_{tens} requires us to compute the limit

$$\lim_{t\to 0}\frac{J_{\mathrm{tens}}(\mathbf{u}+t\mathbf{v})-J_{\mathrm{tens}}(\mathbf{u})}{t}$$

For this, we make use of the identity

$$A^2-B^2=(A-B)(A+B)$$

where

$$A=\sqrt{J_{\mathrm{tens}}(\mathbf{u}+t\mathbf{v})}=\int_0^1\|\mathbf{u}'(\xi)+t\mathbf{v}'(\xi)\|\,\mathrm{d}\xi$$

and

$$B=\sqrt{J_{\mathrm{tens}}(\mathbf{u})}=\int_0^1\|\mathbf{u}'(\xi)\|\,\mathrm{d}\xi$$

As such,

$$A-B=\int_0^1\|\mathbf{u}'(\xi)+t\mathbf{v}'(\xi)\|-\|\mathbf{u}'(\xi)\|\,\mathrm{d}\xi=$$

$$=\int_0^1\|\mathbf{u}'(\xi)\|+t\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t^2)-\|\mathbf{u}'(\xi)\|\,\mathrm{d}\xi=\int_0^1 t\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t^2)\,\mathrm{d}\xi$$

and

$$A+B=\int_0^1\|\mathbf{u}'(\xi)+t\mathbf{v}'(\xi)\|+\|\mathbf{u}'(\xi)\|\,\mathrm{d}\xi=\int_0^1 2\|\mathbf{u}'(\xi)\|+t\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t^2)\,\mathrm{d}\xi$$

Therefore

$$\begin{aligned}
\lim_{t\to 0}\frac{J_{\text{tens}}(\mathbf{u}+t\mathbf{v})-J_{\text{tens}}(\mathbf{u})}{t} &= \lim_{t\to 0}\frac{A^2-B^2}{t} = \lim_{t\to 0}\frac{(A-B)(A+B)}{t}\\
&= \lim_{t\to 0}\frac{1}{t}\left(\int_0^1 t\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t^2)\,\mathrm{d}\xi\right)\left(\int_0^1 2\|\mathbf{u}'(\xi)\|+t\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t^2)\,\mathrm{d}\xi\right)\\
&= \lim_{t\to 0}\left(\int_0^1 \frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t)\,\mathrm{d}\xi\right)\left(\int_0^1 2\|\mathbf{u}'(\xi)\|+t\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}+\mathcal{O}(t^2)\,\mathrm{d}\xi\right)\\
&= 2\int_0^1 \frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}\,\mathrm{d}\xi\underbrace{\int_0^1\|\mathbf{u}'(\xi)\|}_{=:\ell(\mathbf{u})} = 2\ell(\mathbf{u})\int_0^1 \frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}\,\mathrm{d}\xi
\end{aligned}$$

Here $\ell(\mathbf{u})$ is the length of $\mathbf{u}$. This term is not a constant as L in the slides, it rather depends on $\mathbf{u}$ itself and it is what makes the variational problem non-linear and the minimization problem non-quadratic.

(2.1d) Determine, which variational problem needs to be solved to compute the shape of the curve. Explain, why we face a non-linear variational problem.

HINT: Use the results from subproblems (2.1a), (2.1c) and (2.1b).

HINT: Do not forget to specify the trial and test spaces.

Solution: The variational problem is:

Find $\mathbf{u}\in(\mathcal{C}^1_{\text{pw}}([0,1]))^2$ such that

$$\int_0^1 2\ell(\mathbf{u})\frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}-\mathbf{f}(\xi)\cdot\mathbf{v}(\xi)\,\mathrm{d}\xi = 0$$

for any $\mathbf{v}\in(\mathcal{C}^1_{0,pw}([0,1]))^2$.

(2.1e) Assuming that $\mathbf{u}$ is sufficiently smooth, find a two-point boundary value problem, whose solution provides the shape of the curve. Take the cue from the approach in [NPDE, Section 1.3.3].

HINT: Remember to use the boundary conditions for test functions $\mathbf{v}\in V_0$.

Solution: By using integration by parts, we have that

$$2\ell(\mathbf{u})\int_0^1 \frac{\mathbf{u}'(\xi)\cdot\mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|}\,\mathrm{d}\xi = -2\ell(\mathbf{u})\int_0^1 \frac{\mathrm{d}}{\mathrm{d}\xi}\left(\frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|}\right)\cdot\mathbf{v}(\xi)\,\mathrm{d}\xi$$

where we used the fact that $\mathbf{v}(0)=\mathbf{v}(1)=\mathbf{0}$. The variational equation can be rewritten as

$$\int_0^1\left(-2\ell(\mathbf{u})\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|}\right)-\mathbf{f}(\xi)\right)\cdot\mathbf{v}(\xi)\,\mathrm{d}\xi = 0$$

which, under the necessary smoothness assumptions, results in the following two-point boundary value problem:

$$2\ell(\mathbf{u})\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|}\right) = -\mathbf{f}(\xi)$$

with the boundary conditions

$$\mathbf{u}(0)=\mathbf{u}_0,\qquad \mathbf{u}(1)=\mathbf{u}_1.$$

Problem 2.2 The Brachistochrone Problem

This task retraces all the essential considerations employed in elastic string modeling in class for a different problem from classical mechanics. The purpose of this problem is to practice all the techniques introduced in [NPDE, Section 1.2.2], [NPDE, Section 1.2.3], [NPDE, Section 1.3.1], and [NPDE, Section 1.3.3]. In addition, it involves some MATLAB implementation and calculus drill (which will not do you any harm).

The Brachistochrone Problem is a classical problem of variational calculus, already tackled by Newton and Bernoulli in the 17th century: Given two points $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^2$, such that $a_2 > b_2$, we are looking for a curve $\mathbf{u} = (u_1, u_2)^\top : [0, 1] \to \mathbb{R}^2$ connecting $\boldsymbol{a}$ to $\boldsymbol{b}$ so that a ball rolling down the curve $\mathbf{u}$ reaches $\mathbf{b}$ in minimal time.

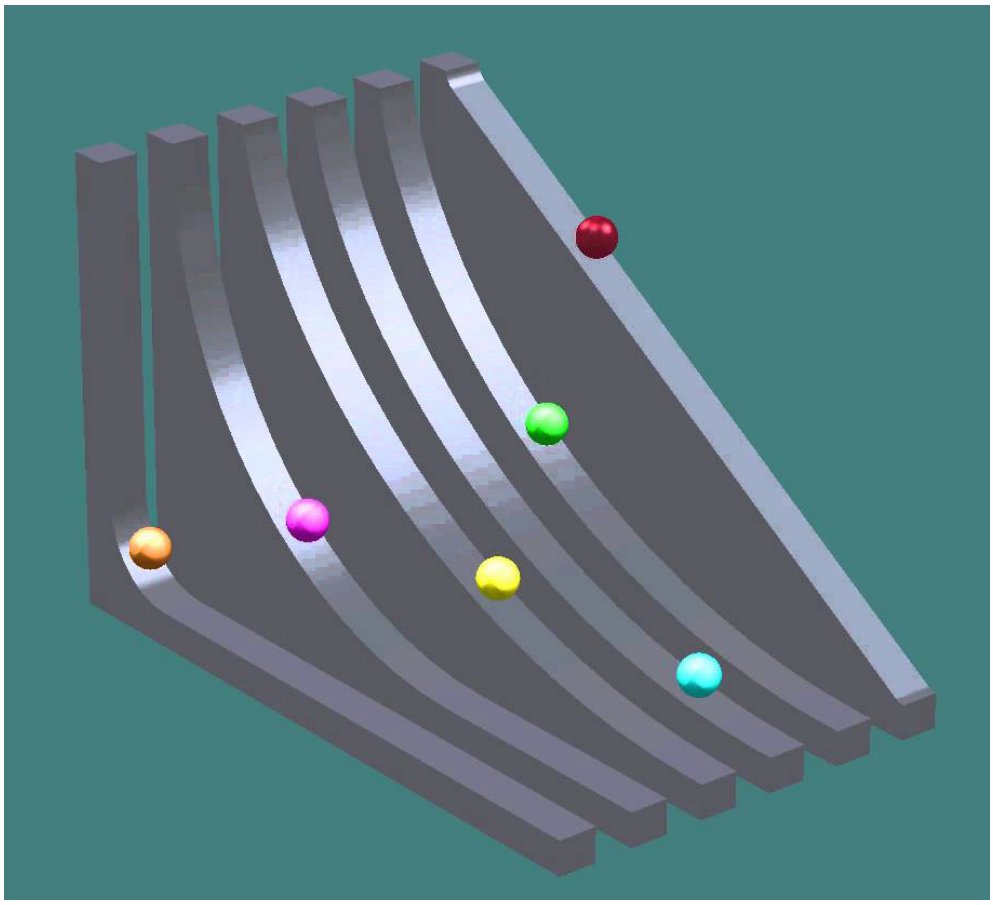

Figure 2.1: Different curves a ball may roll down. On which does it reach the bottom in shortest time?

Following our approach to the modeling of an elastic string in [NPDE, Section 1.2.2], we first consider an approximate discrete model in order to arrive at a continuous minimization problem describing the Brachistochrone Problem by a limit process.

The **discrete model** approximates the Brachistochrone curve by a polygon; let $\mathbf{u}^i$ for $i = 0, \ldots, N+1$, be points (knots) along the curve, with $\mathbf{u}^0 = \boldsymbol{a}$ and $\mathbf{u}^{N+1} = \boldsymbol{b}$, so that there are N "free" points. To find the speed v of the ball at a point $\mathbf{u}$, we assume that the ball starts at rest and appeal to conservation of total (kinetic and potential) energy. Thus, in non-dimensional form already, we

have for its speed

$$v(\mathbf{u})^2 = a_2 - u_2 \quad \Longrightarrow \quad v(\mathbf{u}) = \sqrt{a_2 - u_2};.$$

In the following, we will assume $a_2 = 0$, which can always be achieved by choosing a suitable coordinate system, so that $v(\mathbf{u}) = \sqrt{-u_2}$. Note that this, of course, requires $u_2 < 0$.

Each segment $[\mathbf{u}^i, \mathbf{u}^{i+1}]$ has length $\|\mathbf{u}^i - \mathbf{u}^{i+1}\|$, and we approximate the speed of the ball on this segment by the constant speed

$$v^i := \sqrt{-\frac{1}{2}(u_2^i + u_2^{i+1})}\,, \quad i = 0, \dots, N\,.$$

This means that the time t_i it takes the ball to cross segment $[\mathbf{u}^i, \mathbf{u}^{i+1}]$ is approximately

$$t_i(\mathbf{u}^i, \mathbf{u}^{i+1}) := \frac{\|\mathbf{u}^{i+1} - \mathbf{u}^i\|}{v^i} = \sqrt{\frac{\|\mathbf{u}^i - \mathbf{u}^{i+1}\|^2}{-\frac{1}{2}(u_2^i + u_2^{i+1})}}\,,$$

and the total time required for rolling along the curve is

$$T_N(\mathbf{u}^1, \dots, \mathbf{u}^N) = \sum_{i=0}^{N} t_i(\mathbf{u}^i, \mathbf{u}^{i+1})\,.$$

The knot positions of the optimal polygon minimize $T_N(\mathbf{u}^1, \dots, \mathbf{u}^N)$.

(2.2a) We consider the discrete model described above. Write a MATLAB function

```
time = time(u, a, b)
```

that accepts arguments $\boldsymbol{a}$ and $\boldsymbol{b}$ (2×1-vectors) and $\mathbf{u}$ (a $2N \times 1$-vector) containing $\mathbf{u}^1, \dots, \mathbf{u}^N$ stacked on top each other. It should compute and return $T_N(\mathbf{u})$.

HINT: Use the MATLAB function `reshape` to convert `u` to an $N \times 2$-matrix instead, i.e.

```
u = reshape(u, 2, N);
```

Other useful MATLAB functions are `diff` and `sum`.

Solution: See listing 2.1 for the code. It makes good use of the provided MATLAB commands to vectorize operations as far as possible.

Listing 2.1: Implementation for `time`

```
function t = time(u, a, b)

    N = length(u)/2;
    u = reshape(u, 2, N);

    du = diff([a, u, b], 1, 2);
    norm_du = sqrt(sum(du.^2, 1));

    mu = ([a(2), u(2,:)] + [u(2,:), b(2)])/2;
```

```
    speed = sqrt(-mu);

    t = sum(norm_du./speed);

end
```

In order to efficiently minimize the function `time` we will also require the gradient of T_N.

(2.2b) In this and the following sub-problems we examine the derivative of the travel time with respect to the point positions. The obtained expressions will be instrumental in the implementation of an iterative solution strategy.

Show that the gradients of t_i with respect to $\mathbf{u}^i$ and $\mathbf{u}^{i+1}$ are

$$\mathbf{grad}_{\mathbf{u}^i} t_i = -\frac{1}{(u_2^i + u_2^{i+1})^2 t_i}\left[2(u_2^i + u_2^{i+1})(\mathbf{u}^i - \mathbf{u}^{i+1}) - \|\mathbf{u}^i - \mathbf{u}^{i+1}\|^2 \begin{pmatrix}0\\1\end{pmatrix}\right]$$
$$\mathbf{grad}_{\mathbf{u}^{i+1}} t_i = -\frac{1}{(u_2^i + u_2^{i+1})^2 t_i}\left[2(u_2^i + u_2^{i+1})(\mathbf{u}^{i+1} - \mathbf{u}^i) - \|\mathbf{u}^i - \mathbf{u}^{i+1}\|^2 \begin{pmatrix}0\\1\end{pmatrix}\right].$$

HINT: You may use the fact that $\mathbf{grad}_{\mathbf{x}} \|\mathbf{x} - \mathbf{y}\|^2 = 2(\mathbf{x} - \mathbf{y})$. Otherwise, this is an exercise in the rules of differentiation (chain rule, quotient rule).

Solution: Since t_i is a square-root type expression, we use the chain rule to get

$$\mathbf{grad}_{\mathbf{u}^i} t_i = \frac{1}{2t_i} \mathbf{grad}_{\mathbf{u}^i} \frac{\|\mathbf{u}^i - \mathbf{u}^{i+1}\|^2}{-\frac{1}{2}(u_2^i + u_2^{i+1})} = -\frac{1}{t_i} \mathbf{grad}_{\mathbf{u}^i} \frac{\|\mathbf{u}^i - \mathbf{u}^{i+1}\|^2}{u_2^i + u_2^{i+1}}.$$

Now we apply the quotient rule. The derivative of the numerator is (by the hint) $2(\mathbf{u}^i - \mathbf{u}^{i+1})$, and the derivative of the denominator is $\begin{pmatrix}0\\1\end{pmatrix}$. This shows the first equation. The second is almost identical.

(2.2c) Show that the gradient of T_N regarded as a function $\mathbb{R}^{2N} \mapsto \mathbb{R}$ is the $2N$-vector

$$\mathbf{grad}_{\mathbf{u}} T_N(\mathbf{u}) = \begin{pmatrix} \mathbf{grad}_{\mathbf{u}^1} t_0 + \mathbf{grad}_{\mathbf{u}^1} t_1 \\ \mathbf{grad}_{\mathbf{u}^2} t_1 + \mathbf{grad}_{\mathbf{u}^2} t_2 \\ \vdots \\ \mathbf{grad}_{\mathbf{u}^N} t_{N-1} + \mathbf{grad}_{\mathbf{u}^N} t_N \end{pmatrix}.$$

Solution: Based on the definition for T_N we have that

$$\mathbf{grad}_{\mathbf{u}} T_N(\mathbf{u}) = \sum_{i=0}^{N} \begin{pmatrix} \mathbf{grad}_{\mathbf{u}^1} t_i \\ \mathbf{grad}_{\mathbf{u}^2} t_i \\ \vdots \\ \mathbf{grad}_{\mathbf{u}^N} t_i \end{pmatrix}.$$

Most of these contributions are actually zero, since t_i only depends on $\mathbf{u}^i$ and $\mathbf{u}^{i+1}$. In fact, each t_i only contributes twice, except t_0 and t_N which contributes only once.

(2.2d) Write a MATLAB function

```
dt = difftime(u, a, b)
```

that accepts the same arguments as the function `time` in sub-problem (2.2a), and returns the gradient as a $2N$-vector.

HINT: Use the MATLAB function `reshape` to convert `u` to an $N \times 2$-matrix instead, i.e.

```
u = reshape(u, 2, N);
```

Other useful MATLAB functions are `diff` and `sum`.

Solution: See listing 2.2 for the code.

Listing 2.2: Implementation for `difftime`

```
function dt = difftime(u, a, b)

    N = length(u)/2;
    u = reshape(u, 2, N);

    du = diff([a, u, b], 1, 2);
    norm_du_sq = sum(du.^2, 1);
    norm_du_sq_mx = [zeros(1,N+1); norm_du_sq];

    mu = ([a(2), u(2,:)] + [u(2,:), b(2)]);
    mu_mx = [mu; mu];

    R = sqrt(-norm_du_sq./mu*2);
    R_mx = [R; R];

    dt = -(-2*mu_mx(:,2:end).*du(:,2:end)...
           -norm_du_sq_mx(:,2:end))...
         ./mu_mx(:,2:end).^2./R_mx(:,2:end);
    dt = dt - (2*mu_mx(:,1:end-1).*du(:,1:end-1)...
           -norm_du_sq_mx(:,1:end-1))...
         ./mu_mx(:,1:end-1).^2./R_mx(:,1:end-1);
    dt = reshape(dt, 2*N, 1);

end
```

Our policy is to use a successive minimization-refinement algorithm. that is, we want to solve the minimization problem for T_1 using just a single free point between $\boldsymbol{a}$ and $\boldsymbol{b}$. Then, we can divide each curve segment into two by inserting its midpoint as a new knot of the polygon, and solve the minimization problem again for T_3, and so on.

(2.2e) To this end, write a MATLAB function

```
newu = refine(u, a, b)
```

that accepts the same arguments as the function `time` in sub-problem (2.2a) (with $\mathbf{u}$ of length $2N$), and returns the extended $\mathbf{u}$-vector of length $4N + 2$. The extra knots are located at the midpoint positions of the sides of the original polygon.

Solution: See listing 2.3 for the code.

Listing 2.3: Implementation for `refine`

```
function newu = refine(u, a, b)

    N = length(u)/2;
    u = reshape(u, 2, N);
    pts = [a, u, b];

    midpts = (pts(:,1:end-1) + pts(:,2:end))/2;
    newu = reshape([midpts; u, b], 4*N+4, 1);
    newu = newu(1:end-2);

end
```

(2.2f) Write a MATLAB script

`minimize.m`

that solves the Brachistochrone problem for $\boldsymbol{a} = (0, 0)^\top, \boldsymbol{b} = (6, -1)^\top$ using the successive minimization-refinement technique described earlier. Use the midpoint $(3, -0.5)$ as your initial guess for $\mathbf{u}^1$.

To solve the minimization problems in MATLAB, you may use the builtin MATLAB function `fminunc` from MATLAB's optimization toolbox in the following way:

```
options = optimset('GradObj', 'on');
u = fminunc(@minfunc, u, options);
```

Here, `minfunc` must be a function that takes *only* $\mathbf{u}$ as argument (not $\mathbf{a}$ or $\mathbf{b}$), and returns the function value and the gradient. See the provided code template for details.

HINT: If you are tired of `fminunc` writing gibberish to your console, you can provide the options `options = optimset('GradObj', 'on', 'Display', 'off');`

HINT: For more information about `fminunc`, type `help fminunc` in MATLAB.

HINT: You can use the supplied function `plot_curve` to draw your curve.

Solution: See listing 2.4 for the code.

Listing 2.4: Implementation for `minimize`

```
function minimize

    a = [0; 0];
    b = [6;-1];

    u = (a+b)/2;
```

```
7
8       options = optimset('GradObj', 'on', 'Display', 'off');
9       for i = 1:4
10          if i > 1
11              u = refine(u, a, b);
12          end
13          u = fminunc(@wrapper, u, options);
14          plot_curve(u, a, b);
15      end
16
17      function [t, dt] = wrapper(u)
18          t = time(u, a, b);
19          dt = difftime(u, a, b);
20      end
21
22  end
```

(2.2g) Now we tackle the continuum limit $N \to \infty$ for the polygon model. Please study again [NPDE, Section 1.2.3], where the corresponding considerations are pursued for the mass-spring model of an elastic string.

(2.2h) As in [NPDE, Fig. 10] assume that the points $\mathbf{u}^i$ lie on a smooth curve $\mathbf{u} : [0,1] \to \mathbb{R}^2$, i.e. that

$$\mathbf{u}^i = \mathbf{u}(\xi^i) = \mathbf{u}\left(\frac{i}{N+1}\right).$$

Show that in the limit $N \to \infty$, we get

$$T(\mathbf{u}) := \lim_{N\to\infty} T_N(\mathbf{u}^1, \ldots, \mathbf{u}^N) = \int_0^1 \frac{\|\mathbf{u}'(\xi)\|}{\sqrt{-u_2(\xi)}}\,\mathrm{d}\xi\,. \tag{2.2.1}$$

HINT: You may use [NPDE, Eq. (1.2.39)] and [NPDE, Eq. (1.3.6)].

Solution: For fixed N, we write $\mathbf{u}^i = \mathbf{u}(\xi^i)$, where $\xi^i = \frac{i}{N+1}$. We have the following approximations:

$$\left\|\mathbf{u}^i - \mathbf{u}^{i+1}\right\| = \frac{1}{N+1}(N+1)\left\|\mathbf{u}(\xi^i) - \mathbf{u}(\xi^{i+1})\right\| = \frac{1}{N+1}\left\|\mathbf{u}'(\xi^{i+\frac{1}{2}})\right\| + \mathcal{O}(N^{-3}),$$

$$\sqrt{-\frac{1}{2}(u_2^i + u_2^{i+1})} = \sqrt{-u_2(\xi^{i+\frac{1}{2}})} + \mathcal{O}(N^{-2}).$$

Thus,

$$T_N(\mathbf{u}) = \frac{1}{N+1}\sum_{i=0}^{N} \frac{\left\|\mathbf{u}'(\xi^{i+\frac{1}{2}})\right\| + \mathcal{O}(N^{-3})}{\sqrt{-u_2(\xi^{i+\frac{1}{2}})} + \mathcal{O}(N^{-2})}.$$

This is a Riemann-type sum for the integral (2.2.1).

(2.2i) Analogous to [NPDE, Section 1.3.1] derive the variational problem arising from the minimization of $T(\mathbf{u})$ from 2.2.1 over the space of curves

$$V := \left\{ \mathbf{v} \in (\mathcal{C}^1_{\mathrm{pw}}([0,1]))^2 \,\middle|\, \mathbf{v}(0) = \boldsymbol{a},\ \mathbf{v}(1) = \boldsymbol{b} \right\} .$$

HINT: Use [NPDE, Eq. (1.3.5)] and the Taylor expansion of $x \mapsto x^{-\frac{1}{2}}$:

$$\frac{1}{\sqrt{-x-th}} = \frac{1}{\sqrt{-x}} \left(1 - \tfrac{1}{2} t \, \frac{h}{x} \right) + \mathcal{O}(t^2) .$$

Solution: Using the hint we can write

$$\begin{aligned} \frac{\|\mathbf{u}' + t\mathbf{v}'\|}{\sqrt{-u_2 - tv_2}} &= \left(\|\mathbf{u}'\| + \frac{\mathbf{u}' \cdot \mathbf{v}'}{\|\mathbf{u}'\|} t + \mathcal{O}(t^2) \right) \left(\frac{1}{\sqrt{-u_2}} \left(1 - \frac{tv_2}{2u_2} \right) + \mathcal{O}(t^2) \right) \\ &= \frac{\|\mathbf{u}'\|}{\sqrt{-u_2}} + t \left(\frac{\mathbf{u}' \cdot \mathbf{v}'}{\sqrt{-u_2}\|\mathbf{u}'\|} - \frac{v_2 \|\mathbf{u}'\|}{2\sqrt{-u_2} u_2} \right) + \mathcal{O}(t^2). \end{aligned}$$

Integrating this equality from $\xi = 0$ to $\xi = 1$ gives us

$$T(\mathbf{u} + t\mathbf{v}) = T(\mathbf{u}) + t \int_0^1 \left(\frac{\mathbf{u}' \cdot \mathbf{v}'}{\sqrt{-u_2}\|\mathbf{u}'\|} - \frac{v_2 \|\mathbf{u}'\|}{2\sqrt{-u_2} u_2} \right) \mathrm{d}\xi + \mathcal{O}(t^2).$$

So

$$D_{\mathbf{v}} T(\mathbf{u}) = \lim_{t \to 0} \frac{1}{t} (T(\mathbf{u} + t\mathbf{v}) - T(\mathbf{u})) = \int_0^1 \left(\frac{\mathbf{u}' \cdot \mathbf{v}'}{\sqrt{-u_2}\|\mathbf{u}'\|} - \frac{v_2 \|\mathbf{u}'\|}{2\sqrt{-u_2} u_2} \right) \mathrm{d}\xi. \tag{2.2.2}$$

The variational problem is as follows: Find $\mathbf{u} \in V$ so that $D_{\mathbf{v}} T(\mathbf{u}) = 0$ for all $\mathbf{v} \in V_0$, where the *test space* V_0 is

$$V_0 := \left\{ \mathbf{v} \in (\mathcal{C}^1_{\mathrm{pw}}([0,1]))^2 \,\middle|\, \mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0} \right\}.$$

(2.2j) Using integration by parts as in [NPDE, Section 1.3.3] derive the differential equation (Euler-Lagrange equation) spawned by the variational problem obtained in sub-problem (2.2i).

Solution: First, note that we can rewrite 2.2.2 in the form

$$D_{\mathbf{v}} T(\mathbf{u}) = \int_0^1 \left(\frac{\mathbf{u}' \cdot \mathbf{v}'}{\sqrt{-u_2}\|\mathbf{u}'\|} - \frac{1}{2} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \frac{\|\mathbf{u}'\|}{\sqrt{-u_2} u_2} \cdot \mathbf{v} \right) \mathrm{d}\xi.$$

Using partial integration on the first term yields

$$D_{\mathbf{v}} T(\mathbf{u}) = - \int_0^1 \left(\frac{\mathrm{d}}{\mathrm{d}d\xi} \left(\frac{\mathbf{u}'}{\sqrt{-u_2}\|\mathbf{u}'\|} \right) + \frac{1}{2} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \frac{\|\mathbf{u}'\|}{\sqrt{-u_2} u_2} \right) \cdot \mathbf{v} \ \mathrm{d}\xi.$$

By [NPDE, Lemma 1.3.33], we can turn this into the ODE

$$\frac{\mathrm{d}}{\mathrm{d}\xi} \left(\frac{\mathbf{u}'}{\sqrt{-u_2}\|\mathbf{u}'\|} \right) + \frac{1}{2} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \frac{\|\mathbf{u}'\|}{\sqrt{-u_2} u_2} = 0.$$

(2.2k) Show that the cycloid curve

$$\mathbf{u}(\xi) = \begin{pmatrix} \pi\xi - \sin(\pi\xi) \\ \cos(\pi\xi) - 1 \end{pmatrix}, \quad 0 \le \xi \le 1, \tag{2.2.3}$$

satisfies the differential equation found in sub-problem (2.2j).

Solution: First, let us get rid of $\|\mathbf{u}'\|$. We have

$$\begin{aligned} \|\mathbf{u}'\|^2 &= (\pi - \pi\cos(\pi\xi))^2 + (\pi \sin \pi\xi)^2 \\ &= \pi^2(1 - 2\cos(\pi\xi) + \cos(\pi\xi)^2 + \sin(\pi\xi)^2) \\ &= \pi^2(2 - 2\cos(\pi\xi)) = -2\pi^2 u_2, \end{aligned}$$

so $\|Vu'\| = \sqrt{2}\pi\sqrt{-u_2}$. Inserting this gives the ODE

$$-\frac{1}{\sqrt{2}\pi}\frac{\mathrm{d}}{\mathrm{d}\xi}\begin{pmatrix} \mathbf{u}' \\ u_2 \end{pmatrix} + \frac{\pi}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\frac{1}{u_2} = 0.$$

We now consider the first term, component by component.

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\begin{pmatrix} u_1' \\ u_2 \end{pmatrix} = \frac{\mathrm{d}}{\mathrm{d}\xi}\begin{pmatrix} \pi - \pi\cos(\pi\xi) \\ 1 - \cos(\pi\xi) \end{pmatrix} = \frac{\mathrm{d}}{\mathrm{d}\xi}\pi = 0.$$

So we see that the ODE holds in the first component. What about the second?

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\xi}\begin{pmatrix} u_2' \\ u_2 \end{pmatrix} &= \frac{\mathrm{d}}{\mathrm{d}\xi}\begin{pmatrix} -\pi\sin(\pi\xi) \\ \cos(\pi\xi) - 1 \end{pmatrix} \\ &= \frac{-\pi^2\cos(\pi\xi)(\cos(\pi\xi) - 1) + \pi\sin(\pi\xi)(-\pi\sin(\pi\xi))}{(\cos(\pi\xi) - 1)^2} \\ &= \frac{-\pi^2\cos(\pi\xi)^2 - \pi^2\sin(\pi\xi)^2 + \pi^2\cos(\pi\xi)}{(\cos(\pi\xi) - 1)^2)} \\ &= \pi^2\frac{\cos(\pi\xi) - 1}{(\cos(\pi\xi) - 1)^2} = \frac{\pi^2}{u_2}. \end{aligned}$$

So in conclusion we get

$$-\frac{1}{\sqrt{2}\pi}\frac{\mathrm{d}}{\mathrm{d}\xi}\begin{pmatrix} \mathbf{u}' \\ u_2 \end{pmatrix} = -\frac{1}{\sqrt{2}\pi}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\frac{\pi^2}{u_2} = -\frac{\pi}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\frac{1}{u_2},$$

so the ODE holds.

(2.2l) Finally, we switch to a graph description of the curve. The considerations run parallel to those of [NPDE, Section 1.4.2], which should be read again before starting with this sub-problem.

We re-parameterize the integral (2.2.1) using the variable $x = u_1(\xi)$. For this to work, we assume that the function u_1 maps $[0, 1]$ one-to-one and onto $[a_1, b_1]$. (It is sufficient, for example, that $u_1'(\xi) > 0$ everywhere.) Then, there exists an inverse

$$\xi : [a_1, b_1] \to [0, 1]$$

so that $u_1(\xi(x)) = x$. Now, define $y(x) = u_2(\xi(x))$. Then $(x, y(x))$ is the *graph* for the curve $\mathbf{u}$ on the x-interval $[a_1, b_1]$.

Show that under this parametrization, (2.2.1) becomes

$$T(y) = \int_{a_1}^{b_1} \sqrt{\frac{1+y'(x)^2}{-y(x)}}\,\mathrm{d}x\ .$$

HINT: Use the chain rule to express $\partial_x u_2(\xi(x))$ and $\partial_x u_1(\xi(x))$, see [NPDE, Eq. (1.4.21)]. Also apply the transformation formula for integrals ("integration by substitution") [NPDE, Eq. (1.4.22)].

Solution: We write $\xi = \xi(x)$, so $\mathrm{d}\xi = \xi'(x)\,\mathrm{d}x$. Of course, $\sqrt{-u_2(\xi)} = \sqrt{-u_2(\xi(x))} = \sqrt{-y(x)}$. (2.2.1) then becomes

$$T(y) = \int_{a_1}^{b_1} \frac{\sqrt{u_1'(\xi(x))^2 + u_2'(\xi(x))^2}}{\sqrt{-y(x)}}\xi'(x)\,\mathrm{d}x$$

Pulling the $\xi'(x)$ into the square root, we see that we have an expression like

$$[u_1'(\xi(x))\xi'(x)]^2 + [u_2'(\xi(x))\xi'(x)]^2 = [\partial_x u_1(\xi(x))]^2 + [\partial_x u_2(\xi(x))]^2 = [\partial_x x]^2 + [\partial_x y]^2 = 1 + y'(x)^2.$$

This concludes the proof.

Listing 2.5: Testcalls for Problem 2.2

```
test_time = time([3;-3], [0;0], [6;-1])
test_difftime = difftime([3;-3], [0;0], [6;-1])'
test_extend = extend([3;-3], [0;0], [6;-1])'
```

Listing 2.6: Output for Testcalls for Problem 2.2

```
test_time =

    6.0136

test_difftime =

   -0.0110   -0.0735

test_refine =

    1.5000   -1.5000    3.0000   -3.0000    4.5000   -2.0000
```

Published on February 25.

To be submitted on March 4.

References

[NPDE] Lecture Slides for the course "Numerical Methods for Partial Differential Equations".SVN revision # 73870.

[1] M. Struwe. Analysis für Informatiker. Lecture notes, ETH Zürich, 2009. https://moodle-app1.net.ethz.ch/lms/mod/resource/index.php?id=145.

Last modified on March 4, 2015