

Constructing diffeomorphisms between simply connected plane domains

Kendall Atkinson* David Chien Olaf Hansen†

March 3, 2021

Abstract

Consider a simply connected domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$ that is given by a smooth function $\varphi : [a, b] \mapsto \mathbb{R}^2$. Our goal is to calculate a diffeomorphism $\Phi : \mathbb{B}_1(0) \mapsto \Omega$, $\mathbb{B}_1(0)$ the open unit disk in \mathbb{R}^2 . We present two different methods, both methods are able to handle boundaries $\partial\Omega$ that are not star shaped. The first method is based on an optimization algorithm that optimizes the curvature of the boundary and the second method is based on the physical principle of minimizing potential energy. Both methods construct first a homotopy between the boundary $\partial\mathbb{B}_1(0)$ and $\partial\Omega$ and then extend the boundary homotopy to the inside of the domains. Numerical examples show that the method is applicable to a wide variety of domains Ω .

Keywords: domain transformations, constructing diffeomorphisms, shape blending

AMS: 65D05, 49Q10

1 Introduction

Consider being given a boundary curve Γ for a simply-connected region Ω in the plane \mathbb{R}^2 . This is usually given as a function $\varphi : [a, b] \rightarrow \mathbb{R}^2$ for some interval $[a, b]$. We want to create a 1-1 function Φ from the unit disk \mathbb{B}^2 onto $\bar{\Omega}$ with $\Phi|_{\Gamma} \equiv \varphi$. In addition, we want to approximate the first derivatives of Φ and its Jacobian. The curve Γ is assumed to be smooth, with $\varphi \in C^2[a, b]$, and it is further assumed that $\varphi, \varphi', \varphi''$ can be computed explicitly.

Creating Φ is a several step process. We begin by finding the arc length parameterization of Γ . For a given n , an equal subdivision of Γ into n subintervals

*University of Iowa (kendall-atkinson@uiowa.edu)

†California State University San Marcos ({chien,ohansen}@csusm.edu)

of equal arc length is created. From it, we create a cubic spline interpolant of Γ , using the nodes associated with the n subdivisions of Γ . This is discussed in Section 2. We then vary this spline function to reduce its variation from a circle of the same arc length. We propose two different approaches to do this, to be discussed in Sections 3 and 4. These methods produce a sequence of boundaries, starting with Γ and going to a circle C of equal arc length. Using the reverse sequence of these boundaries, a mapping from the unit disk is created by solving a suitable differential equation; cf. Section 5.

This problem was explored earlier in [1, Chapter 3] and [2], but it was restricted mainly to regions Ω that are starlike with respect to some point in the region. Methods based on interpolation of the boundary function φ were studied. In addition, an iterative method was introduced to improve choices of Φ . The methods proposed in this paper can also be used for regions that are not starlike. For other papers concerned with this problem, see [9, 10, 11]. In [4] the optimization of geometries with respect to certain criteria are discussed. This is kind of an opposite problem to the one presented here as the final shape (a circle) is known, but still a transformation has to be constructed.

In the book [1], boundary value problems for partial differential equations are solved over regions that can be transformed to a unit disk. This requires being able to carry out such a transformation, and the present paper shows how to do this. In a further paper, the ideas of this paper will be applied to the solution of partial differential equations.

2 Boundary preliminaries

Let the boundary Γ be given by $\varphi \in C[a, b]$. To calculate its circumference, apply numerical integration to

$$(1) \quad L \equiv \int_a^b |\varphi'(u)| du.$$

The trapezoidal rule is a logical choice because of the periodicity of the integrand over the interval $[a, b]$.

Let σ denote the arc length as a function of the given parameterization variable τ :

$$\sigma(\tau) \equiv \int_a^\tau |\varphi'(u)| du, \quad a \leq \tau \leq b.$$

Let $\psi \in C^2[0, L]$ denote the inverse of σ . It satisfies

$$t = \sigma(\psi(t)) = \int_a^{\psi(t)} |\varphi'(u)| du, \quad 0 \leq t \leq L.$$

Differentiating,

$$1 = |\varphi'(\psi(t))| \psi'(t).$$

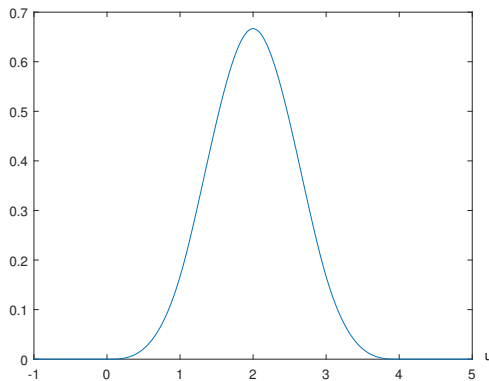


Figure 1: Graph of $B(u)$

Thus $\psi(t)$ is the solution of the initial value problem

$$\psi'(t) = \frac{1}{|\varphi'(\psi(t))|}, \quad 0 \leq t \leq L,$$

with $\psi(0) = 0$. An ode solver (say `ode113` or `rkf45` within MATLAB) is used to evaluate the solution at n evenly spaced points, with $h = L/n$ the arc length between adjacent evenly spaced nodes. Denote these nodes by x_0, x_1, \dots, x_n , with $x_n = x_0$, and let $\{t_0, t_1, \dots, t_n\}$ denote the corresponding arc lengths from x_0 of the nodes. These are extended periodically, e.g. $x_j = x_{j \pm n}$; and let $\{t_j\}$ also be extended periodically, e.g. $t_{j \pm n} = t_j \pm L$.

We replace the parameterization of Γ with a periodic cubic spline that interpolates Γ at the nodes. Denote this cubic spline by $\eta \in C_p^2[0, L]$, a periodic C^2 function on $(-\infty, \infty)$,

$$\eta(t_i) = x_i, \quad i = 0, 1, \dots, n.$$

The equal spacing of the arc length parameters $\{t_i\}$ leads to a simple representation for η .

Introduce the cubic B-spline

$$B(u) = \frac{1}{6} \left\{ (4-u)_+^3 - 4(3-u)_+^3 + 6(2-u)_+^3 - 4(1-u)_+^3 + (0-u)_+^3 \right\}$$

where

$$(y)_+^3 = \begin{cases} y^3, & y \geq 0, \\ 0, & y < 0. \end{cases}$$

See Figure 1. The support of $B(t)$ is $(0, 4)$. Introduce

$$B(x, j; h) = B((x - t_j)/h), \quad -\infty < x < \infty, \quad j = 0, \pm 1, \pm 2, \dots$$

The support of $B(x, j, h)$ is (t_j, t_{j+4}) .

We want a function

$$(2) \quad p(x) = \sum_{j=-\infty}^{\infty} \alpha_j B(x, j; h).$$

To make this periodic with period $L = nh$, We make the coefficients $\{\alpha_j\}$ periodic in j with period n ,

$$\alpha_{j+n} = \alpha_j, \quad j = 0, \pm 1, \pm 2, \dots$$

Consider interpolation of a periodic function $f(t)$ at the nodes $\{t_j\}$, with L an integer multiple of the period of f . We must solve the linear system

$$(3) \quad \sum_{j=-\infty}^{\infty} \alpha_j B(t_i, j; h) = f(t_i), \quad i = 0, \pm 1, \pm 2, \dots$$

We need consider only the case of $i = 1, \dots, n$ because of periodicity. The resulting linear system for $\{\alpha_1, \dots, \alpha_n\}$ is diagonally dominant with diagonal elements $\frac{2}{3}$ and the off-diagonal elements in each equation summing to $\frac{1}{3}$. The matrix of coefficients is

$$\frac{1}{6} \begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots \\ \vdots & & & \ddots & & \\ 0 & \dots & & 0 & 1 & 4 \\ 1 & 0 & \dots & 0 & 1 & 4 \end{bmatrix}$$

This proves the existence of a unique solution $p(x; h)$ to the interpolation problem, and it will be periodic. It will also be a cubic spline function with knots $\{t_0, t_1, \dots, t_n\}$. The cost of solving the system (3) is $\mathcal{O}(n)$.

This spline interpolation is applied to both components of the parameterization $\varphi(\psi(t))$, $0 \leq t \leq L$.

3 Moving toward a circle by using optimization

Before we discuss two different methods to connect the boundary Γ to a circle C with the same arc length in the current and following section we like to remark, why we did not choose to use the 'obvious' way of connecting the two curves. By the obvious way we denote the following homotopy: Assume that $p(x)$, $x \in [0, L]$ is an arclength parametrization of Γ and $q(x)$, $x \in [0, L]$, is an arclength parametrization of the circle C . Then one can define the homotopy $H(t, x)$, $(t, x) \in [0, 1] \times [0, L]$, by

$$H(t, x) = (1 - t)p(x) + tq(x)$$

It turns out that this approach has two problems. First, it is not clear that the intermediate curves $H(t, \cdot)$, $0 < t < 1$, are simple closed curves that enclose an open domain. Second, when we used this transformation to construct a mapping of the disk to Ω the resulting mapping for the enclosed region, see Section 5, seem to show a wide variation of the Jacobian. Therefore we looked at alternative methods to map C to Γ and two methods that we discovered are presented below.

Consider perturbing the coefficients in (2), producing a new spline formula:

$$(4) \quad p(x; \delta) = \sum_{j=-\infty}^{\infty} (\alpha_j + \delta_j) B(x, j; h)$$

with the $\{\delta_j\}$ periodic in j with period n . Proceed by making the curvature closer to that of a circle with circumference L , which has constant curvature

$$(5) \quad \frac{2\pi}{L}.$$

Calculate the curvature of the modified curve at each x_j as

$$\kappa(x_j; \delta) = \frac{p'(x_j; \delta)_1 p''(x_j; \delta)_2 - p'(x_j; \delta)_2 p''(x_j; \delta)_1}{\left(p'(x_j; \delta)_1^2 + p'(x_j; \delta)_2^2\right)^{3/2}}$$

Note the use of subscripts for the 2-vector $p(x; \delta)$. The distance between nodes is maintained by requiring

$$(6) \quad |p(x_{j-1}; \delta_{j-1}) - p(x_j; \delta_j)| = h, \quad j = 1, \dots, n.$$

This places n constraints on the $2n$ real variables $\{\delta_j\}$. To restrict the growth of δ , size restrictions are placed on $\{\delta_1, \dots, \delta_n\}$,

$$(7) \quad |\delta_j| \leq A, \quad j = 1, \dots, n$$

for some $A > 0$. Finally, the restriction

$$(8) \quad \delta_0 = 0$$

is used to fix the curve with respect to the node x_0 .

To make the curvature closer to that of (5), introduce the function

$$(9) \quad f(\delta) = \sqrt{\sum_{j=1}^n w_j \left(\kappa(x_j; \delta) - \frac{2\pi}{L} \right)^2}$$

with $\{w_j\}$ positive weights summing to 1. We use both equal weights and weights proportional to the curvature of the original curve Γ . The choice of δ in (4) comes from optimizing $f(\delta)$ subject to the constraints (6)-(8). In our examples, we have used `fmincon` from the MATLAB optimization library. For

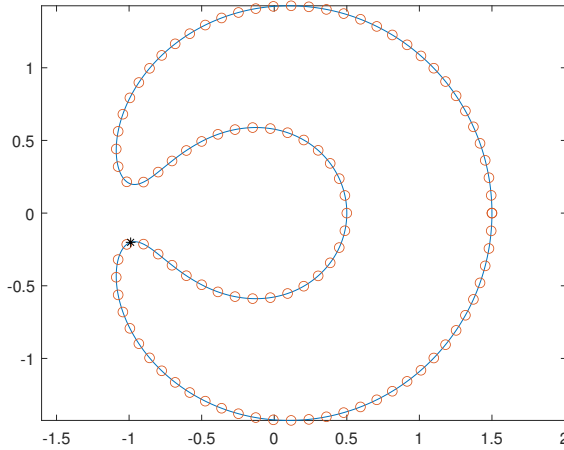


Figure 2: Pacman $\varphi(u)$ over $[0, 2\pi]$. '*' denotes maximum curvature.

the weights $\{w_j\}$, we alternate between equal weights and weights proportional to the absolute value of the curvature on the original curve at the nodes $\{x_j\}$. This alternation has led experimentally to a more robust iteration. Using just one choice of $\{w_j\}$ also converges, although not as rapid.

As will be explained further in Section 5, we need a sequence of changes in Γ with an 'easy change' from each member of the sequence to the next element of the sequence. We control this in part by our choice of the constant A in (7). However, too small a choice for A often slows the iteration a great deal.

Example 1 Consider the 'pacman' boundary given by

$$\varphi(u) = (1 + a \cos(u)) \begin{bmatrix} \cos(f \sin u) \\ \sin(f \sin u) \end{bmatrix}, \quad 0 \leq u \leq 2\pi.$$

with $a = 0.5$ and $f = \pi - 0.2$. It is shown in Figure 2 with $n = 100$ nodes created using subdivisions according to arc length, denoted by \circ . The curvature with respect to arc length is shown in Figure 3. The arc length (1) was approximated using the trapezoidal rule with $m = 300$ subdivisions; the arc length is approximately $L \doteq 12.2356817$. The maximum deviation of the interpolatory spline from the true curve was 0.00463, although it was much less at most points. The deviation was greatest at the points of maximum curvature. The region interior to Γ is not starlike with respect to any point, and therefore, the ideas in [2] are not applicable.

Apply the optimization method associated with (6)-(9), and use $A = 1.0$ for the bound on the perturbations in δ . The iterates are shown in Figure 4. The equally spaced points on the final circle are shown with \circ . The iterations have

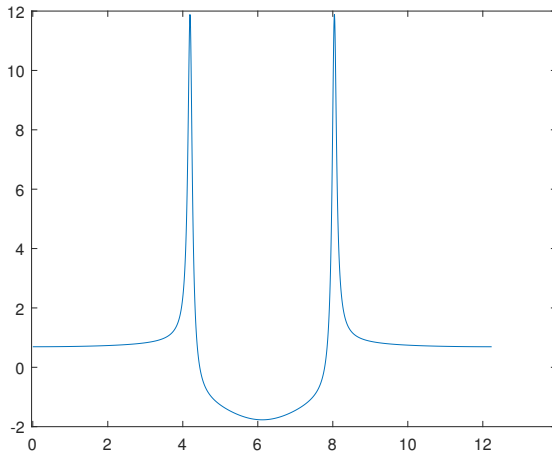


Figure 3: Curvature of pacman curve

the repeating pattern of red, green, and blue. The final iterate is a circle of approximate radius

$$R = \frac{L}{2\pi} \doteq \frac{12.2356817}{2\pi} \doteq 1.94737.$$

The curvature of the final circle should be $1/R \doteq 0.51351$, and the curvature of the final circular curve in Figure 4 varies over $[0.5125, 0.5145]$.

Example 2 Consider the boundary Γ given by

$$(10) \quad \varphi(u) = \begin{bmatrix} (5 + \sin u - 1.5 \sin(3u) - \cos(5u)) \cos u \\ (5 + \sin u + \sin(3u) - 3 \cos(5u)) \sin u \end{bmatrix}, \quad 0 \leq u \leq 2\pi.$$

It is shown in Figure 5 with $n = 100$ nodes created using subdivisions according to arc length, denoted by \circ . It will be referred to as an amoeba boundary. The arc length (1) was approximated using the trapezoidal rule with $m = 300$ subdivisions; the arc length is approximately $L \doteq 57.253363$. The curvature with respect to arc length is shown in Figure 6. The maximum deviation of the interpolatory spline from the true curve was 0.063, although it was much less at most points. The deviation was greatest at the points of maximum curvature. The region interior to Γ is not starlike with respect to any point, and therefore, the ideas in [2] are not applicable.

Apply the optimization method associated with (6)-(9). Equal weights $\{w_j\}$ are used in (9). Use $A = 2.0$ for the bound on the perturbations in δ and $n = 100$. The iterates are shown in Figure 7. The iterations have the repeating

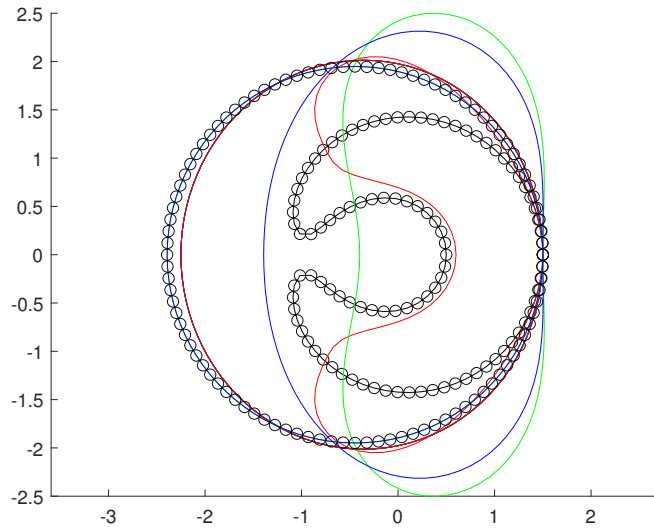


Figure 4: Iterations for pacman curve

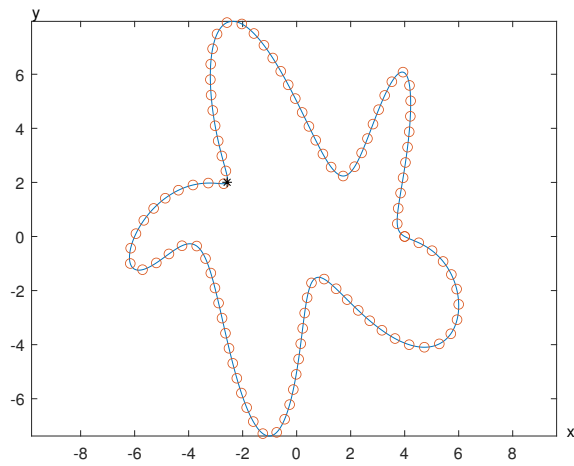


Figure 5: Amoeba $\varphi(u)$ over $[0, 2\pi]$. '*' denotes maximum curvature.

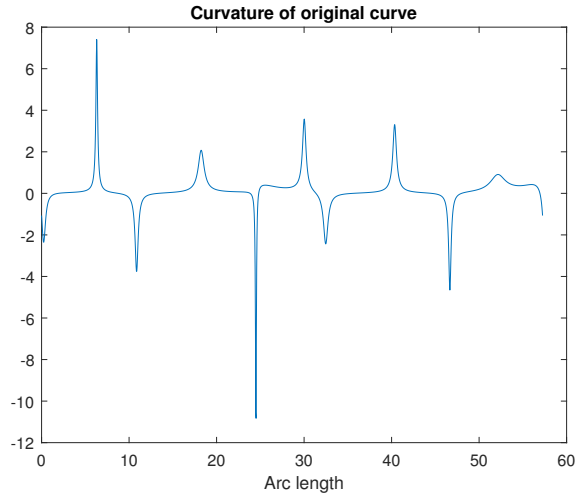


Figure 6: Curvature of (10) with respect to arc length.

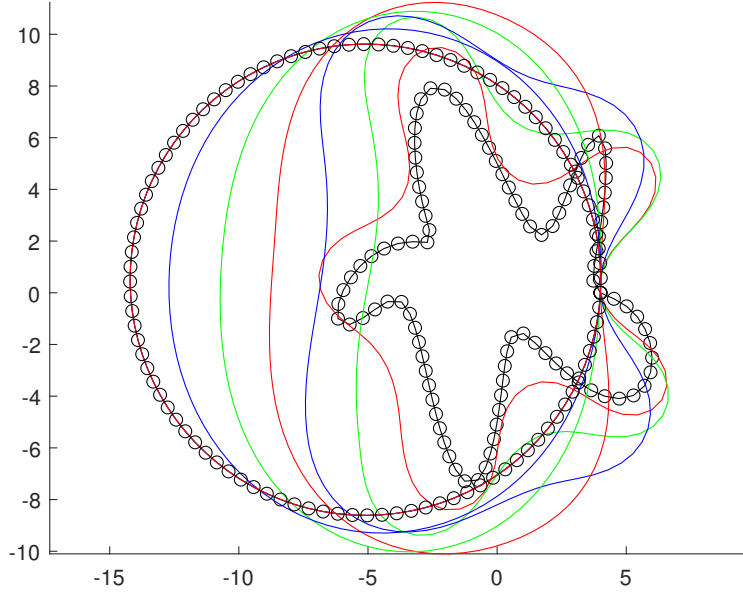


Figure 7: Iterations due to minimizing (9).

pattern of red, green, and blue. The final iterate is a circle of approximate radius

$$R = \frac{L}{2\pi} \doteq \frac{57.253363}{2\pi} \doteq 9.11216.$$

The curvature of the final circle should be $1/R \doteq 0.109744$, and the curvature of the final circular curve in Figure 7 varies over $[0.10960, 0.10983]$.

4 Moving toward a circle by minimizing energy

As in the previous Section 3 we assume that the boundary Γ is given by a function $p(x)$, see (2), and $0 \leq x \leq L$ is the arc length parameter. So the points $p_i = p(iL/n)$ have an approximately equal spacing

$$|p_i - p_{i+1}| \approx \frac{L}{n}.$$

In practice this condition is better satisfied if n is increased, but we will assume

$$|p_i - p_{i+1}| = \frac{L}{n} \equiv \Delta.$$

in the following to simplify the presentation of the algorithm. Figure 2 shows an example of the pacman boundary Γ and p_i , $i = 1, \dots, 100$.

The idea for our algorithm comes from a physical experiment. Assume that we have a flexible closed conductor loop in the shape of Γ that is charged sufficiently high, then the loop will unfold and take the shape of a circle to minimize the potential energy. A discrete version of this experiment is that we have metallic pearls attached equidistantly along a string in the shape of Γ and the pearls all carry the same electrical charge. We call these pearls charges in the following. The connection between two charges is rigid, so the distance between the charges is fixed. If the charges can move and the charge is sufficiently high the charges will move into the positions of a regular n -gon to minimize the energy.

The algebraic differential equation (DAE) for this situation is the following:

$$(11) \quad p_i''(t) = \sum_{j=1, j \neq i}^n \frac{1}{|p_i(t) - p_j(t)|^3} (p_i(t) - p_j(t)),$$

$$i = 1, \dots, n, \quad t > 0$$

$$(12) \quad \Delta^2 = \begin{cases} |p_i - p_{i+1}|^2, & i = 1, \dots, n-1 \\ |p_n - p_1|^2. \end{cases}$$

Here we assume that the position p_i of the charge i depends on time $t \geq 0$ and we have set all physical constants equal to 1. The initial conditions $p_i(0)$ are given by their initial position along Γ , as in the above Figure 2.

There are two problems with the above formulation. First equation (11) conserves energy, so if the charges p_i move towards their position along the regular n -gon they have an excess of kinetic energy and will be oscillating around their position in the n -gon. To remove energy from the above system we will include a friction term that is defined by a constant $\alpha > 0$ that will determine how fast the charges will loose their kinetic energy.

Second the final position of the n -gon is determined by the differential equation (11), but once the charges are in the final position the energy is not changed if we move the n -gon or rotate it. Numerically we will need to ensure (12) by using Newton's method. Our numerical experiments showed that eliminating these two degrees of freedom (straight movement and rotation) simplifies the numerical implementation. We remove these two degrees of freedom by fixing the positions of $p_1(t)$ (no straight movements anymore) and $p_n(t)$ (no rotational freedom anymore). Instead of p_1 and p_n we could have chosen any other neighboring pair of charges.

These two modifications lead to our final DAE

$$(13) \quad p_i''(t) = \sum_{j=1, j \neq i}^n \frac{1}{|p_i(t) - p_j(t)|^3} (p_i(t) - p_j(t)) - \alpha p_i'(t),$$

$$i = 2, \dots, n-1, t > 0$$

$$\Delta^2 = |p_i - p_{i+1}|^2, i = 1, \dots, n-1$$

To simplify the notation we introduce a vector of all movable charge positions (from now on we consider p_1 and p_n as constants):

$$P(t) = (p_2(t), \dots, p_{n-1}(t))^T \in \mathbb{R}^{2(n-2)}$$

and the function

$$g(P(t)) = (|p_1 - p_2(t)|^2 - \Delta^2, |p_2(t) - p_3(t)|^2 - \Delta^2, \dots, |p_{n-1}(t) - p_n|^2 - \Delta^2)^T \in \mathbb{R}^{n-1}$$

Now the above differential algebraic system can be written as

$$P''(t) = F(P(t), P'(t)), t > 0$$

$$g(P(t)) = 0 \in \mathbb{R}^n$$

with initial condition $P(0) = P_0 = (p_2, \dots, p_{n-1})$ and F defined according to (13).

We use Euler-Lagrange formulation, see [7, VI.5] and add constraint forces that will enforce $g(P(t)) = 0$:

$$P''(t) = F(P(t), P'(t)) - G^T(P(t))\lambda(t), t > 0 \quad g(P(t)) = 0 \in \mathbb{R}^{n-2}$$

where $G(P(t)) = (\frac{\partial g_i(P(t))}{\partial p_j})_{i,j}$ is the Jacobi matrix and $\lambda = (\lambda_1(t), \dots, \lambda_{n-1}(t))$ are Lagrange multipliers.

Lastly we transform the above system in a system of order 1, by introducing the velocity vector $Q(t)$:

$$Q(t) = P'(t)$$

Then the above second order differential algebraic system can be written as

$$(14) \quad P'(t) = Q(t)$$

$$(15) \quad \begin{aligned} Q'(t) &= F(P(t), Q(t)) - G^T(P(t))\lambda(t), \quad t > 0 \\ g(P(t)) &= 0 \end{aligned}$$

This is DAE of index 3, see [7]. Before we start to solve numerically this equation, there are two questions: What value we use for α and on which time interval $[0, T]$ we solve the DAE (14), (15).

In our experiments we found that a range of $\alpha \in [1, 20]$ leads to a fast convergence of $P(t)$ to a regular n -gon. In general a larger α means it takes longer to reach the n -gon position, but a larger α also leads to a smoother movement of the charges towards their final position.

To determine T we monitor how far away from a circle the charges $P(t)$ are. Once we find a circle such that all points in $P(t)$ are within $\varepsilon > 0$ of the circle we stop the time integration. In our experiments we use $\varepsilon = 0.01$. We have implemented this in the following way. Given a set of points $p_i, i = 1, \dots, n$, we calculate the center

$$C = \frac{1}{n} \sum_{i=1}^n p_i$$

of the points and an average radius

$$R = \frac{1}{n} \sum_{i=1}^n |p_i - C|$$

We stop the time integration once

$$\max_{i=1}^n ||p_i - C| - R| \leq \varepsilon$$

To solve the DAE (14), (15) we use a backward difference method (BDF of order 2, see [6, III.6]) with a Newton method to solve the nonlinear equation system in each step. The stepwidth is variable such that the estimated local integration error is smaller than a given tolerance τ , $\tau = 0.001$ in our experiments. At the end of the numerical integration we have a sequence $\tilde{P}(t_i), i = 0, \dots, N_T$, $t_0 = 0, t_{N_T} = T$.

Then we get a numerical approximation $\tilde{P}(t), t \in [0, T]$, for $P(t)$. Here we can use again spline interpolation or piecewise linear interpolation to extend $\tilde{P}(t_i)$ to the interval $[0, T]$. As in Section 3 the final movement from the circle $\tilde{P}(T)$ to Γ that corresponds to $\tilde{P}(0)$ is given by time reversal $\tilde{P}(T-t), t \in [0, T]$.

Figure 8 and 9 show the trajectories of $\tilde{P}(t)$, that move the points from Γ in Figure 2 to a circle C , here the the friction parameter are $\alpha = 5$ in Figure 8

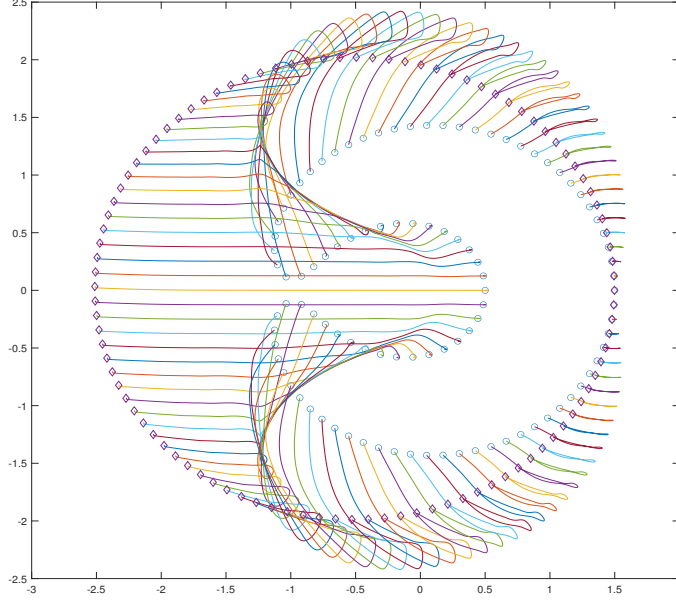


Figure 8: The trajectories of 100 points moving from the pacman curve to a circle, friction coefficient $\alpha = 5$.

and $\alpha = 15$ in Figure 9. For a fixed t , $0 < t < T$, we have a sequence of points in $\tilde{P}(T - t)$ that represent an intermediate curve between a circle, $t = 0$, and Γ , $t = T$. By using spline interpolation we can expand the points at $T - t$ to a continuous curve. Together with the above mention interpolation in time we have found a homotopy between a circle and Γ which we will denote by

$$H(t, x), (t, x) \in [0, T] \times [0, L]$$

Where for a constant iL/n , $H(\cdot, iL/n)$, is given by, for example, piecewise linear interpolation in time of the values of $\tilde{P}(t_i)$, and for a fixed t , $H(t, \cdot)$ is given by periodic spline interpolation of the $H(t, iL/n)$, $i = 0, \dots, n$.

The numerical function H depends on n , the discretization of Γ , α , ϵ , and τ , but we will in general not indicate this dependence.

Similar to this section the optimization method described in Section 3 also leads to a homotopy $H(t, x)$. In the previous Section 3 the optimization function produced a sequence of boundary curves $\tilde{\Gamma}_0$ to $\tilde{\Gamma}_N$ (in opposite order). Each curve $\tilde{\Gamma}_i$ is given by a spline function $p_i(x)$ with arclength parameter $x \in [0, L]$. Remember $\tilde{\Gamma}_0$ is a circle and $\tilde{\Gamma}_N = \partial\Omega$. Typical N is much smaller than

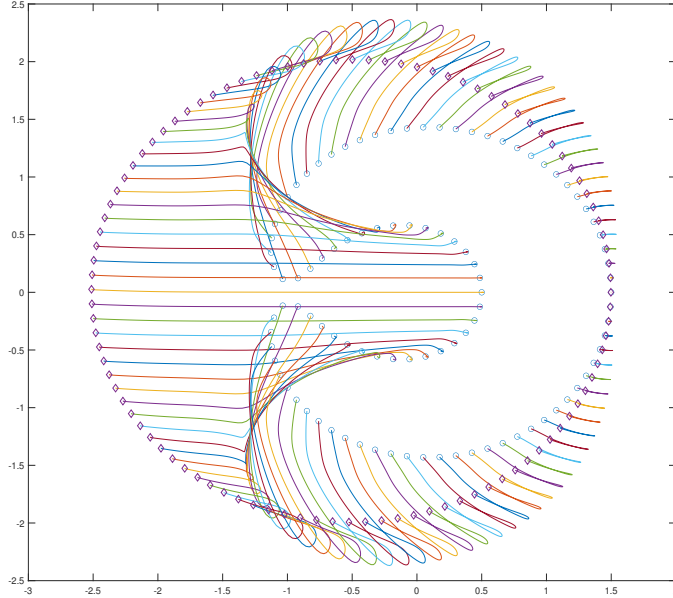


Figure 9: The trajectories of 100 points moving from the pacman curve to a circle, friction coefficient $\alpha = 15$.

the number N_T of intermediate curves of the current section. By choosing A , see 7, small enough we can assume that all curves $\tilde{\Gamma}_i$, $i = 1, \dots, N$, are simple and can be connected without self intersection. The connecting curves can be, for example, given by straight lines or splines. We introduce an artificial time t , $0 \leq t \leq N$ and this defines a homotopy

$$\tilde{H}(t, x), (t, x) \in [0, N] \times [0, L]$$

between $\tilde{\Gamma}_0$ and $\partial\Omega = \tilde{\Gamma}_N$. We will denote \tilde{H} also by H in the following sections and write T instead of N .

In conclusion, we have now described two algorithms that lead to a homotopy between a circle C of circumference L and a given curve Γ of length L . These homotopies H will be the foundation for the algorithm in Section 5 where we will construct a mapping between the disk with boundary C and the domain Ω that is bounded by Γ .

5 Mapping the disk to Ω

Following Section 3 and 4 we have now a homotopy $H(t, x)$ that connects the circle

$$\Gamma_0 = \{H(0, x) \mid 0 \leq x \leq L\}$$

and $\Gamma = \partial\Omega$, with x denoting arc length. Let \mathbb{D} denote the open disk whose boundary is the circle associated with Γ_0 , as obtained in the preceding two sections. It is straightforward to map the unit disk \mathbb{B}^2 to the disk \mathbb{D} with a smooth bijective function Φ_0 . So, once we have constructed a smooth bijective function Φ_1 from \mathbb{D} to Ω as a continuous extension of the homotopy H we have found the desired mapping

$$\Phi : \mathbb{B}^2 \rightarrow \overline{\Omega}$$

as the composition $\Phi = \Phi_1 \circ \Phi_0$. The function H defines a sequence of boundary curves

$$\Gamma_t = \{H(t, x) \mid 0 \leq x \leq L\}, \quad 0 \leq t \leq T,$$

with $\Gamma_T = \Gamma = \partial\Omega$. We denote the interior of Γ_t by \mathbb{D}_t , so $\mathbb{D}_0 = \mathbb{D}$ and $\mathbb{D}_T = \Omega$.

For a fixed $x \in [0, L]$ the function $\psi(t) = H(t, x)$ defines a trajectory that start at a point $\alpha = H(0, x) \in \Gamma_0$ and ends at a point $\beta = H(T, x) \in \Gamma_T$. The function $\psi(t)$ can be considered to be the solution of the initial value problem.

$$\begin{aligned} \dot{\psi}(t) &= V(t, \psi(t)), \quad 0 < t \leq T, \\ \psi(0) &= \alpha \in \Gamma_0 \end{aligned}$$

with the vector field

$$V(t, \xi) = \frac{\partial H(t, \mathbf{x}_t(\xi))}{\partial t}$$

given along Γ_t , $t \in [0, T]$. Here $\mathbf{x}_t(\xi)$ is the function that assigns to a $\xi \in \Gamma_t$ the arc length value x with $\xi = H(t, x)$. Figure 8 and 9 illustrate some of the $\psi(t)$ (various initial values α) for the pacman curve and Figure 4 illustrates a sequence of \mathbb{D}_t . To indicate the dependence of $\psi(t)$ on the starting point $\alpha = H(0, x)$ we will use the notation $\psi_\alpha(t)$ from now on.

We follow the approach of [5, Section 5, Th. 5.13] and continuously extend the vector field V to a vector field \mathcal{V} on \mathbb{D}_t by ‘ k -transfinite barycentric interpolation’,

$$(16) \quad \mathcal{V}(t, \xi) = \frac{1}{\int_{\Gamma_t} \frac{1}{|\eta - \xi|^k} ds_\eta} \int_{\Gamma_t} \frac{V(t, \eta)}{|\eta - \xi|^k} ds_\eta, \quad \xi \in D_t, \quad 0 \leq t \leq L,$$

where $k > 1$ is a parameter. It is shown in [5] that \mathcal{V} is continuous on \mathbb{R}^2 and that $\mathcal{V}(t, \xi) = V(t, \xi)$ for $\xi \in \Gamma_t$, see [5, Theorem 3.5].

This vector field allows us to define trajectories $\psi_\alpha(t)$ to all $\alpha \in \mathbb{D}_0$ by defining $\psi_\alpha(t)$ as the solution of initial value problem

$$(17) \quad \dot{\psi}_\alpha(t) = \mathcal{V}(t, \psi_\alpha(t)), \quad 0 < t \leq T,$$

$$(18) \quad \psi_\alpha(0) = \alpha \in \mathbb{D}_0$$

This defines a smooth mapping $\Phi_1(\alpha) = \psi_\alpha(T)$ that maps \mathbb{D}_0 to $\mathbb{D}_T = \Omega$. By our construction Φ_1 maps Γ_0 1-1 into $\Gamma = \partial\Omega$, because the solutions to the initial value problems are (17),(18) are unique.

As $\Gamma_T = \partial\Omega$ is the boundary of Ω and

$$H(0, x) \xrightarrow{\Phi_1} H(T, x), \quad x \in [0, L],$$

maps the boundary of the disk \mathbb{D}_0 to the boundary of $\Omega = \mathbb{D}_T$ the mapping degree, see [3], $d(\Phi_1, \mathbb{D}_0, y) = 1$ for every $y \in \Omega = \mathbb{D}_T$. So, the mapping $\Phi_1 : \mathbb{D}_0 \mapsto \mathbb{D}_T$ is surjective. This shows that Φ_1 is a bijective mapping from the closed disk $\overline{\mathbb{D}_0}$ to $\overline{\mathbb{D}_T} = \overline{\Omega}$.

Because our goal is to use the mapping Φ_1 for the solution of boundary value problems on Ω , see [1], we will also need to approximate the Jacobi matrix $J(\alpha) = (J_{i,j}(\alpha))_{i,j=1,2}$ with

$$J_{i,j}(\alpha) = \frac{\partial \Phi_{1,i}}{\partial x_j}(\alpha)$$

where we denote two components of Φ_1 by $\Phi_{1,1}$ and $\Phi_{1,2}$. The Jacobi matrix $J(\alpha)$ is given by $J(\alpha) = Y_\alpha(T)$. Here $Y_\alpha(t)$ is the solution of the initial value problem

$$(19) \quad \dot{Y}_\alpha(t) = DV(t, \psi_\alpha(t)) \cdot Y_\alpha(t), \quad 0 < t \leq T,$$

$$(20) \quad Y_\alpha(0) = I_{2,2}$$

with $I_{2,2}$ the identity matrix and DV the Jacobi matrix of \mathcal{V} that is determined by the derivatives of the boundary integral in (16). See [8, Section 3.1] for a derivation of (19), (20).

To approximate Φ_1 numerically we can integrate the initial value problems (17), (18) and (19), (20) simultaneously for a set of points $\alpha \in \mathbb{D}_0$.

To numerically approximate $\Phi_1(\alpha)$ and $J(\alpha)$ we can use any method to numerically integrate the initial value problems (17), (18) and (19), (20). We have initially used an explicit Euler method and all examples in the following section are generated with the explicit Euler method.

6 Numerical examples

In this last section we illustrate our algorithm by presenting the images of a grid on the unit disk in the corresponding domains Ω .

On the unit disk we consider grids as shown in Figure 10, it has 82 radial lines and 11 circumferential lines.

For every grid point α in Figure 10 we calculate a numerical approximation $\tilde{\psi}_{\alpha,h,n}$ to the trajectory ψ_α given by (17), (18). We have used $k = 2$ in (16) that defines the right hand side of (17) for the examples in this section. As mentioned in Section 5 the numerical solutions are calculated by using an explicit Euler method with step width h . The resulting images of the grid for the two domains

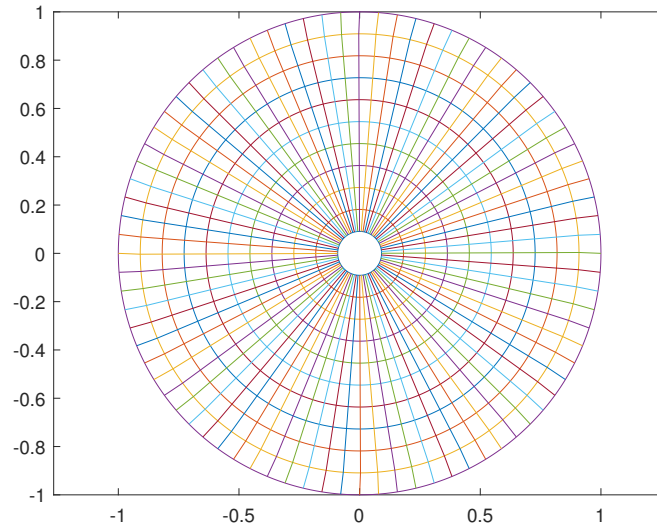


Figure 10: Grid on unit disk.

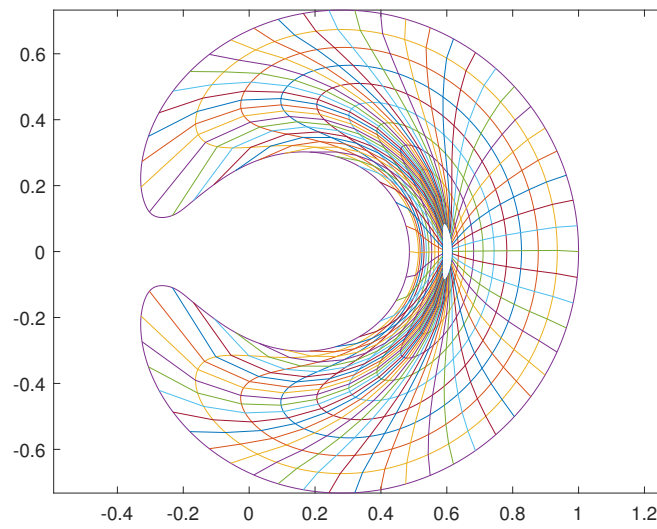


Figure 11: Grid on pacman.

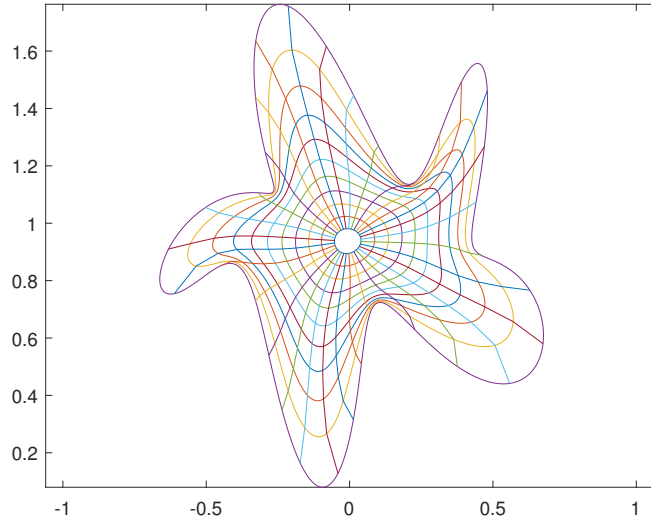


Figure 12: Grid on amoeba.

shown in Figures 2 and 5 are given in Figures 11 and 12. Here the image points $\tilde{\psi}_{\alpha,h,n}(T)$ are connected to mark their position and illustrate the deformation.

The stepwidth h for the calculation of $\tilde{\psi}_{\alpha,h,n}$ was determined experimentally. It was chosen small enough to guarantee that the image of the grids did not overlap. In a similar way the parameter $n \in \mathbb{N}$ was determined in order to approximate the vector field \mathcal{V} in equation (17) sufficiently well. We used a trapezoidal rule for both integral terms

$$\begin{aligned}
 \mathcal{V}(t, \xi) &\approx \mathcal{V}_n(t, \xi) \\
 (21) \quad &= \frac{1}{\frac{L}{n} \sum_{i=0}^{n-1} \frac{1}{|H(t, iL/n) - \xi|^k}} \frac{L}{n} \sum_{i=0}^{n-1} \frac{H(t, iL/n)}{|H(t, iL/n) - \xi|^k}
 \end{aligned}$$

Remember that L is the length of the original boundary Γ of Ω and that the parametrizations $H(t, \cdot)$ are all parameterized with respect to arclength. Formula (21) implies that numerical trajectory $\tilde{\psi}_{\alpha,h,n}$ is the result of applying the explicit Euler method to the initial value problem

$$\begin{aligned}
 \dot{\psi}_{\alpha}(t) &= \frac{\sum_{i=0}^{n-1} \frac{H(t, iL/n)}{|H(t, iL/n) - \psi_{\alpha}(t)|^k}}{\sum_{i=0}^{n-1} \frac{1}{|H(t, iL/n) - \psi_{\alpha}(t)|^k}}, \quad 0 < t \leq T, \\
 \psi_{\alpha}(0) &= \alpha \in \mathbb{D}_0
 \end{aligned}$$

Note that on a computer with parallel processors, these equations can be solved independently for different choices of grid points α .

For the application of our method to the solution of boundary value problems on Ω we will need to investigate the accuracy of $\tilde{\psi}_{\alpha,h,n}$ as a function of h and n further. This is the subject of a forthcoming paper.

References

- [1] K. ATKINSON, D. CHIEN, AND O. HANSEN, *Spectral Methods Using Multivariate Polynomials On The Unit Ball*, CRC Press, Boca Raton, USA, 2019.
- [2] K. ATKINSON, AND O. HANSEN, *Creating domain mappings*, Electronic Transactions on Numerical Analysis, 39 (2012), pp. 202-230.
- [3] K. DEIMLING, *Nonlinear Functional Analysis*, Springer-Verlag, Berlin Heidelberg New York Tokyo, 1985.
- [4] M.C. DELFOUR, J.-P. ZOLÉSIO, *Shapes and Geometries*, SIAM Advances in Control and Design, SIAM, Philadelphia, USA, 2011.
- [5] M.C. DELFOUR, A. GARON, *Transfinite Interpolations for Free and Moving Boundary Problems*, Pure and Applied Functional Analysis, 4 (2019), pp. 765-801.
- [6] E. HAIRER, S.P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I*, Springer-Verlag, Berlin Heidelberg, Germany, 1987.
- [7] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II*, Springer-Verlag, Berlin Heidelberg New York Tokyo, USA, 1991.
- [8] E. HILLE, *Lecture on Ordinary Differential Equations*, Addison–Wesley Publishing Company, London, Great Britain, 1969.
- [9] M. SABA, T. SCHNEIDER, K. HORMANN, R. SCATENI, *Curvature-based blending of closed planar curves*, 5 (2014), pp. 263-272.
- [10] T.W. SEDERBERG, P. GAO, G. WANG, H. MU, *2-D Shape Blending: An Intrinsic Solution to the Vertex Path Problem*, ACM SIGGRAPH Computer Graphics, (1993), pp. 15-18.
- [11] T. SURAZHSKY, G. ELBER, *Metamorphosis of Plan Parametric Curves via Curvature Interpolation*, International Journal of Shape Modeling, 2 (2002), pp. 201–216.