

# CONSTRUCTING DIFFEOMORPHISMS BETWEEN SIMPLY CONNECTED PLANE DOMAINS - PART 2

KENDALL ATKINSON\*, DAVID CHIEN†, AND OLAF HANSEN‡

**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 polynomial  $P^{(n)} : \mathbb{B}^2 \mapsto \Omega$  of maximum degree  $n$  such that  $P^{(n)}$  is a diffeomorphism. Here  $\mathbb{B}^2$  is the open unit disk in  $\mathbb{R}^2$  and  $n$  has to be chosen suitable large. The polynomial mapping  $P^{(n)}$  is given as the  $L^2$  projection of a mapping  $\Phi$  that is only known for a discrete set of points in  $\mathbb{B}^2$ . The construction of  $\Phi$  was given in our previous article [2]. Using  $P^{(n)}$  we can transform boundary value problems from  $\Omega$  to  $\mathbb{B}^2$  using a Galerkin method, see [1]. In §5 we give numerical examples demonstrating the use of  $P^{(n)}$  to solve Dirichlet problems for two example regions  $\Omega$ .

**Key words.** domain mapping, multivariate polynomial, constrained minimization, nonlinear iteration

**AMS subject classification.** 65D05, 49Q10

**Keywords:** domain transformations, constructing a diffeomorphism, multivariate polynomials, Dirichlet problem

**AMS:** 65D05, 49Q10

**1. Introduction.** Consider being given a boundary curve  $\Gamma$  for a simply connected region  $\Omega$  in the plane  $\mathbb{R}^2$ . This is usually given as a function  $\varphi : [a, b] \rightarrow \mathbb{R}^2$  for some interval  $[a, b]$ . For convenience we generally choose  $[a, b] = [0, 2\pi]$ . We want to create a 1-1 function  $\Phi$  from the closed unit disk  $\mathbb{B}^2$  onto  $\overline{\Omega}$  with  $\Phi|_{\mathbb{S}^1} \equiv \varphi$ ,  $\mathbb{S}^1 = \partial\mathbb{B}^2$ . The curve  $\Gamma$  is assumed to be smooth, with  $\varphi \in C^2[0, 2\pi]$ , and it is further assumed that  $\varphi, \varphi', \varphi''$  can be computed explicitly.

Creating  $\Phi$  is a several step process which was described in [2]. We begin by finding the arc length parameterization of  $\Gamma$ , call it  $L$ . For a given  $\rho$ , an equal subdivision of  $\Gamma$  into  $\rho$  subintervals of equal arc length  $L/\rho$  is created. From it, we create a cubic spline interpolant of  $\Gamma$ , using the nodes associated with the  $\rho$  subdivisions of  $\Gamma$ . We then vary this spline function to reduce its variation to a circle of the same arc length. We proposed two different approaches to do this. These methods produce a sequence of boundaries, starting with  $\Gamma$  and going to a circle  $C$  of equal arc length  $L$ . Using the reverse sequence of these boundaries, a mapping from the unit disk is created by solving a suitable differential equation.

This problem was explored in [1, Chapter 3] and [3], but it was restricted mainly to regions  $\Omega$  that are starlike with respect to some point inside the region. Methods based on interpolation of the boundary function  $\varphi$  were studied. In addition, an iterative method was introduced in [3] to improve choices of  $\Phi$ . The methods proposed in [2] can also be used for regions that are not starlike. For other papers concerned with this problem, see [8, 9, 10].

The transformation produced in [2] produces transformed values  $\Phi(p_i)$  for given nodes  $\{p_i\} \subseteq \mathbb{B}^2$ . A more convenient form of a transformation is to have a polynomial

---

\*Depts of Mathematics & Computer Science, University of Iowa, Iowa City, IA 52242

†Dept of Mathematics, California State University at San Marcos, San Marcos, CA 92096

‡Dept of Mathematics, California State University at San Marcos, San Marcos, CA 92096

mapping defined on  $\mathbb{B}^2$ , useful for the numerical methods given in [1] for solving partial differential equations. In this paper we consider the construction of such a polynomial. Examples are given later in this paper for more complicated regions that are not starlike.

**2. General Schema.** Denote the set of polynomials in 2 variables of degree at most  $n$  by  $\Pi_n$ . Its dimension is

$$N_n = \frac{1}{2}(n+1)(n+2).$$

Let  $\{\varphi_i : i = 1, \dots, N_n\}$  denote an orthonormal basis for  $\Pi_n$ . We use the ‘Xu polynomials’ as they can be evaluated rapidly.

The Xu polynomials of degree  $n$  are given by

$$(2.1) \quad \varphi_{n,k}(x) = \frac{1}{h_{n,k}} C_{n-k}^{k+1}(x_1) (1-x_1^2)^{\frac{k}{2}} C_k^{\frac{1}{2}}\left(\frac{x_2}{\sqrt{1-x_1^2}}\right), \quad x \in \mathbb{B}^2,$$

for  $n = 0, 1, \dots$  and  $k = 0, 1, \dots, n$ . The functions  $\{C_n^\lambda(t)\}$  are ‘Gegenbauer polynomials’. They can be obtained using the following generating function:

$$(1-2rt+r^2)^{-\lambda} = \sum_{n=0}^{\infty} C_n^\lambda(t) r^n, \quad |r| < 1, \quad |t| \leq 1,$$

and their triple recursion relation is given by

$$(2.2) \quad C_{n+1}^\lambda(t) = \frac{2(n+\lambda)}{n+1} t C_n^\lambda(t) - \frac{n+2\lambda-1}{n+1} C_{n-1}^\lambda(t), \quad n \geq 1.$$

The initial cases are

$$C_0^\lambda(t) \equiv 1, \quad C_1^\lambda(t) = 2\lambda t, \quad C_2^\lambda(t) = \lambda(2(\lambda+1)t^2 - 1).$$

A general polynomial of degree  $n$  can be written as

$$(2.3) \quad p(x) = \sum_{j=0}^n \sum_{k=0}^j \alpha_{j,k} \varphi_{j,k}(x)$$

This can be evaluated in

$$(2.4) \quad 3n^2 + 9n - 4$$

arithmetic evaluations, approximately 6 times the dimension of  $\Pi_n$ . For details on this, see [1, §2.3].

Let  $\Phi$  denote an original transformation from  $\mathbb{B}^2$  onto  $\overline{\Omega}$ . For a given degree  $n$ , denote a polynomial approximation of  $\Phi$  by

$$(2.5) \quad P^{(n)}(x) = \sum_{j=1}^{N_n} \alpha_j \varphi_j(x) \approx \Phi(x).$$

Here the  $\varphi_j$  are a linear reordering of the polynomials  $\varphi_{j,k}$  in (2.3). The coefficients  $\alpha_j \in \mathbb{R}^2$ . We choose these coefficients by minimizing the least squares approximation error

$$(2.6) \quad E_n(x) = \int_{\mathbb{B}^2} \left\{ \left| P_1^{(n)}(x) - \Phi_1(x) \right|^2 + \left| P_2^{(n)}(x) - \Phi_2(x) \right|^2 \right\} dx$$

In this,  $P_1^{(n)}$  and  $P_2^{(n)}$  denote the two components of  $P^{(n)}$ , and similarly for  $\Phi_1$  and  $\Phi_2$ . This leads to the linear system

$$(2.7) \quad \sum_{j=1}^{N_n} \alpha_j^{(i)} (\varphi_k, \varphi_j) = (\Phi_i, \varphi_k), \quad k = 1, \dots, N_n, \quad i = 1, 2$$

where  $\alpha_j = [\alpha_j^{(1)}, \alpha_j^{(2)}]^T$ . Due to the orthonormality of  $\{\varphi_j\}$  over  $\mathbb{B}^2$ , we obtain from (2.7) that

$$(2.8) \quad \alpha_k = (\Phi, \varphi_k), \quad k = 1, \dots, N_n.$$

The resulting polynomial  $P^{(n)}$  is simply the truncation of the orthonormal expansion

$$\Phi(x) = \sum_{j=1}^{\infty} (\Phi, \varphi_j) \varphi_j(x), \quad x \in \mathbb{B}^2$$

to terms of degree  $\leq n$ .

The integrals in (2.7) are approximated with a quadrature over  $\mathbb{B}^2$  :

$$\begin{aligned} (\varphi_k, \varphi_j) &\approx (\varphi_k, \varphi_j)_\eta = I_\eta(\varphi_k \cdot \varphi_j), \\ (\Phi_i, \varphi_k) &\approx (\Phi_i, \varphi_k)_\eta = I_\eta(\Phi_i \cdot \varphi_k), \end{aligned}$$

for  $i = 1, 2$ . The quadrature formula we use is the well-known Gauss-Legendre formula

$$(2.9) \quad I(g) \equiv \int_{\mathbb{B}^2} g(x) dx \approx I_\eta(g) \equiv \frac{2\pi}{2\eta+1} \sum_{l=0}^{\eta} \sum_{m=0}^{2\eta} \omega_l r_l \tilde{g}\left(r_l, \frac{2\pi m}{2\eta+1}\right),$$

in which  $\tilde{g}(r, \theta) \equiv g(r \cos \theta, r \sin \theta)$ . The formula uses the trapezoidal rule with  $2\eta+1$  subdivisions for the integration over  $[0, 2\pi]$  in the azimuthal variable  $\theta$ . The numbers  $r_l$  and  $\omega_l$  are, respectively, the nodes and weights of the  $(\eta+1)$ -point Gauss-Legendre quadrature formula on  $[0, 1]$ . This quadrature over  $\mathbb{B}^2$  is exact for all polynomials  $g \in \Pi_{2\eta}$ , see [1, §2.6]. Using this in (2.7), we would then get  $(\varphi_k, \varphi_j)_\eta = \delta_{j,k}$ . For the numerical examples in §4 and §5 we used  $\eta = 80$ .

To approximate (2.8), we need to evaluate  $\Phi(x)$  at the  $M_\eta \equiv (\eta+1)(2\eta+1)$  nodes used in (2.9). Denote these nodes by  $\{p_j : j = 1, \dots, M_\eta\}$ . We use the codes from [2] to do this evaluation of  $\{\Phi(p_j)\}$ . A more detailed explanation is given in the following §3. After calculating the coefficients  $\{\alpha_j\}$  of  $P$ , we need to evaluate its accuracy in reproducing the boundary  $\Gamma$ . We also want to insure it is 1-1 over  $\mathbb{B}^2$ , and to aid in this we evaluate the Jacobian of  $P^{(n)}$  over  $\mathbb{B}^2$ , checking that it is nonzero everywhere. These calculations are illustrated in §4.

**3. Evaluating  $\Phi$ .** Here we describe briefly the construction of the mapping  $\Phi : \mathbb{B}^2 \mapsto \Omega$ . A more detailed description can be found in [2].

As mentioned in the introduction we start with a sequence of  $\rho$  points  $\omega_{0,j} = \varphi(u_j)$ ,  $j = 0, \dots, \rho$ ,  $0 = u_0 < u_1 < \dots < u_\rho = 2\pi$ , where the  $u_j$  are chosen such that the arclength along  $\varphi$  between the points  $\omega_{0,j}$  is approximately equal to  $L/\rho$ .  $L$  the length of the curve  $\Gamma$ , see introduction.

In [2] we describe two methods to iteratively deform the points  $(\omega_{0,j})_{j=0 \dots \rho}$  into a sequence of points  $(\omega_{k,j})_{j=0 \dots \rho}$ ,  $k = 1, \dots, K$ . The idea is that each set of points

TABLE 3.1  
Boundary errors of  $\tilde{\Gamma}$  for various  $\rho$  values

$\rho$	Max. Distance (amoeba)	Max. Distance (pacman)
100	$4.345 \times 10^{-2}$	$1.563 \times 10^{-3}$
200	$3.012 \times 10^{-3}$	$4.323 \times 10^{-4}$
300	$1.733 \times 10^{-3}$	$9.919 \times 10^{-5}$
400	$6.612 \times 10^{-4}$	$2.292 \times 10^{-5}$

$(\omega_{k,j})_{j=0\dots\rho}$  is located along the boundary of a domain  $\Omega_k$  that develops from the original domain  $\Omega_0 = \Omega$  to a disk  $\Omega_K$ :

$$\Gamma \ni (\omega_{0,j})_{j=0\dots\rho} \rightarrow (\omega_{1,j})_{j=0\dots\rho} \rightarrow \dots \rightarrow (\omega_{K,j})_{j=0\dots\rho} \in C$$

The final set of points  $(\omega_{K,j})_{j=0\dots\rho}$  is located along the boundary of a circle  $C$ . During this process the arclength between the points  $(\omega_{k,j})_{j=0\dots\rho}$  is kept approximately constant. To simplify our presentation we will assume that this circle  $C$  is the unit circle  $C = \mathbb{S}^1$ . In our programs we use an additional affine linear transformation to transform  $C$  to  $\mathbb{S}^1$ .

Now we define a tensor product spline  $\Psi(\tau, \sigma)$  with  $(\tau, \sigma) \in [0, K] \times [0, 2\pi]$  by the interpolation condition

$$(3.1) \quad \Psi(k, \frac{2\pi}{\rho}j) = \omega_{K-k,j}, \quad k = 0, \dots, K, \quad j = 0, \dots, \rho.$$

See [4, Chapter XVII] for the definition of tensor product splines. We use cubic periodic splines in the  $\sigma$  variable and cubic splines with a not-a-knot condition for the  $\tau$  variable.

In (3.1) one sees that we reverse the  $k$  variable so

$$\tilde{\mathbb{S}}^1 := \{\Psi(0, \sigma) \mid 0 \leq \sigma \leq 2\pi\} \approx \mathbb{S}^1$$

is a spline approximation to the unit circle and

$$\tilde{\Gamma} := \{\Psi(K, \sigma) \mid 0 \leq \sigma \leq 2\pi\} \approx \Gamma$$

is a spline approximation to the boundary  $\Gamma$  of  $\Omega$ . Table 3.1 shows how close  $\tilde{\Gamma}$  is to  $\Gamma$ , by giving the maximum distance between  $\tilde{\Gamma}$  and  $\Gamma$  for various  $\rho$  values of the spline  $\Psi(K, \cdot)$ .

One can think of  $Z = \{\Psi(\tau, \sigma) \mid 0 \leq \tau \leq K, 0 \leq \sigma \leq 2\pi\}$  as a cylinder that connects  $\mathbb{S}^1$  with  $\Gamma$ . The spline  $\Psi$  defines our mapping  $\Phi$  on the boundary  $\mathbb{S}^1$ :

$$\Phi(\Psi(0, \sigma)) = \Psi(K, \sigma), \quad \sigma \in [0, 2\pi]$$

This formula also shows that we actually don't map  $\mathbb{S}^1$  onto  $\Gamma$  but the approximation  $\tilde{\mathbb{S}}^1$  onto the approximation  $\tilde{\Gamma}$ .

The next step is to extend this mapping in a smooth way to the interior of  $\tilde{\mathbb{S}}^1$  in such a way that the mapping is bijective onto the interior  $\tilde{\Omega}$  of  $\tilde{\Gamma}$ .

For every  $\sigma \in [0, 2\pi]$  the mapping

$$\tau \mapsto u_\sigma(\tau) = \Psi(\tau, \sigma), \quad \tau \in [0, K],$$

is a trajectory that connects  $\Psi(0, \sigma) \in \tilde{\mathbb{S}}^1$  with  $\Psi(K, \sigma) \in \tilde{\Gamma}$  along the cylinder  $Z$ . The function  $u_\sigma$  defines a vector field  $\nu$  along the cylinder  $Z$ :

$$\nu(\tau, \sigma) = \frac{d}{d\tau} u_\sigma(\tau)$$

This vector field can be smoothly extended to the inside of the cylinder  $Z$  by using a technique, called transfinite interpolation, see [6, 5]. We call the resulting vector field  $\hat{\nu}(\tau, x)$ , which is now defined inside the cylinder  $Z$ .

This vector field defines a dynamical system that maps the interior of  $\tilde{\mathbb{S}}^1$  to  $\tilde{\Omega}$ . For each of the quadrature points  $\{p_j : j = 1, \dots, M_n\}$  we solve numerically the initial value problem

$$\begin{aligned} u_j(0) &= p_j \\ u_j'(\tau) &= \hat{\nu}(\tau, u_j(\tau)), \quad 0 < \tau \leq K \end{aligned}$$

and define

$$\Phi(p_j) = u_j(K)$$

(actually the numerical approximation to  $u_j(K)$ ). As reported in [2] we use Runge–Kutta codes with step size control for this approximation.

Because the spline boundary  $\tilde{\mathbb{S}}^1$  is not equal to the boundary of the unit circle  $\mathbb{S}^1$  it might happen that some quadrature points  $p_j$  are outside of  $\tilde{\mathbb{S}}^1$ . Especially, because the Gauss–Legendre points are concentrated towards the boundary points 0 and 1. For this reason we modify the quadrature points  $\{p_j : j = 1, \dots, M_n\}$  by multiplying them with the constant to guarantee that all points are inside of  $\tilde{\mathbb{S}}^1$ . The disadvantage of this approach is that the Xu–polynomials are only approximately orthonormal. Then the solution of system (2.7) may be better. This effect is reduced for larger values of  $\rho$  when  $\Psi(0, \sigma)$  approximates the unit circle better. In none of the examples that are presented in the next section did we need to modify the quadrature points.

**4. Numerical Examples.** We illustrate our ideas with two curves - a ‘pacman’ curve and an ‘amoeba’ curve. Use the parametrization interval  $0 \leq t \leq 2\pi$ .

*Amoeba:* Define

$$(4.1) \quad x = \begin{bmatrix} (5 + \sin t - 1.5 \sin(3t) - \cos(5t)) \cos t \\ (5 + \sin t + 1.5 \sin(3t) - 2 \cos(5t)) \sin t \end{bmatrix}$$

See Figure 4.1.

*Pacman:* Define

$$(4.2) \quad x = \begin{bmatrix} (1 + h \cos t) \cos(f \sin t) \\ (1 + h \cos t) \sin(f \sin t) \end{bmatrix}$$

with  $f = \pi - 0.2$ ,  $h = 0.5$ . See Figure 4.2.

**4.1. Calculations for the amoeba.** In the formula (2.9), let  $\eta = 80$ . This leads to  $M_{80} = 81 \times 161$  points  $p_j$  in  $\mathbb{B}^2$ ,  $Q_{80} \equiv \{p_j : j = 1, \dots, M_{80}\}$ . The method in §3 produces the transformation  $\Phi$  of  $\mathbb{B}^2$  to  $\Omega$  at these points. The transformation uses the cubic spline interpolation of the curve  $\Gamma$ . We use  $\rho = 100, 200, 300, 400$  equal arclength subdivisions of the curve  $\Gamma$  for defining the cubic spline interpolant.

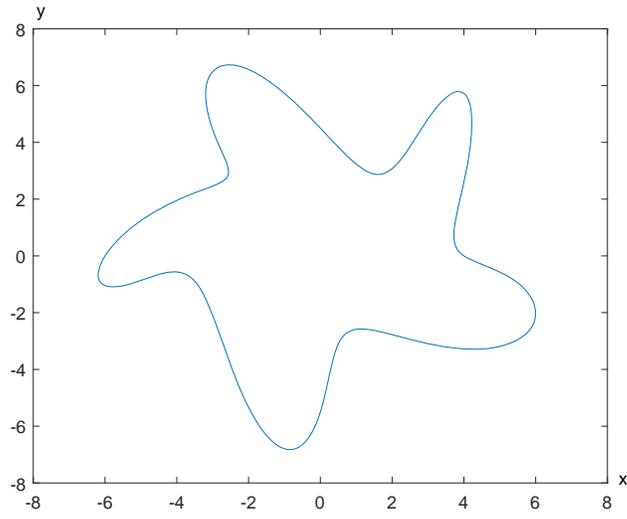


FIG. 4.1. *Amoeba (4.1)*

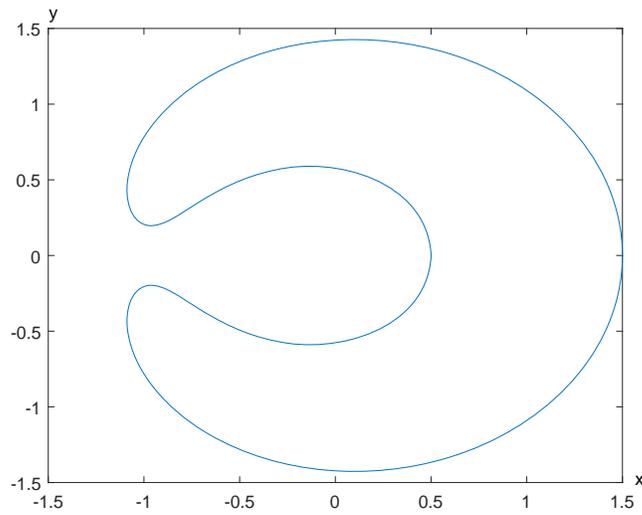


FIG. 4.2. *Pacman (4.2)*

With this transformation, we look at polynomials  $P^{(n)}(x)$  of degrees  $n = 32, 48, 64$ , produced using  $\Phi$  evaluated at the nodes in  $Q_{80}$ , using these values to evaluate the needed coefficients (2.8) with the quadrature formula (2.9). In Table 4.1  $BE$  denotes the maximum error in  $P^{(n)}(x)$  as  $x$  varies over  $\mathbb{S}^1$  as when compared to the true boundary  $\Gamma$ . The point of maximum deviation occurs at a point of maximum curvature. Figures 4.3 and 4.4 show more detail for  $n = 64$  and  $\rho = 400$ . The true curve in Figure 4.3 is shown in red and the approximating curve is shown in a black dotted line. As can be seen, the two curves are essentially identical. The point of maximum

TABLE 4.1  
Boundary errors

$\rho$	n	BE (amoeba)	BE (pacman)
100	32	0.0778	0.0149
	48	0.0493	0.00488
	64	0.0465	0.00187
200	32	0.0615	0.0146
	48	0.0261	0.00492
	64	0.0127	0.00185
300	32	0.0602	0.0139
	48	0.0248	0.00466
	64	0.0116	0.00177
400	32	0.0593	0.0123
	48	0.0233	0.00411
	64	0.00892	0.00154

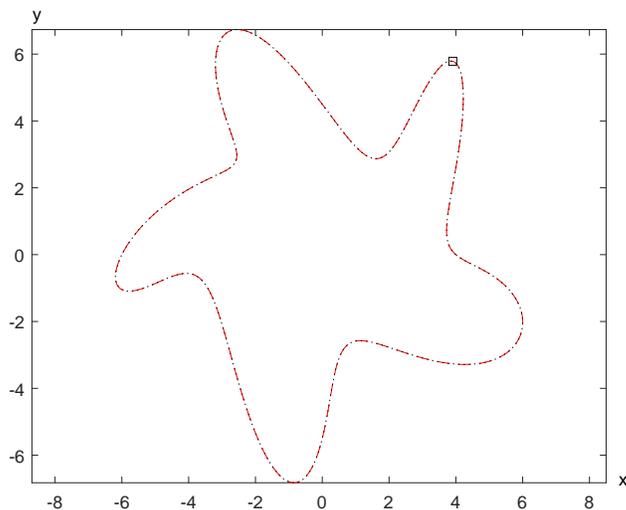


FIG. 4.3. Showing the point of maximum deviation of the degree 64 approximating polynomial to the amoeba with  $\rho = 400$ .

error is denoted by a square  $\square$ . The deviation over  $[0, 2\pi]$  is shown in Figure 4.4. See Figure 4.5 for a graph of the Jacobian.

**4.2. Calculations for the pacman.** The calculations are the same as described for the amoeba. Again, the figures are for degree 64 and  $\rho = 400$  subdivisions of the boundary. Graphs analogous to those given for the amoeba region are shown in Figures 4.6, 4.7, and 4.8.

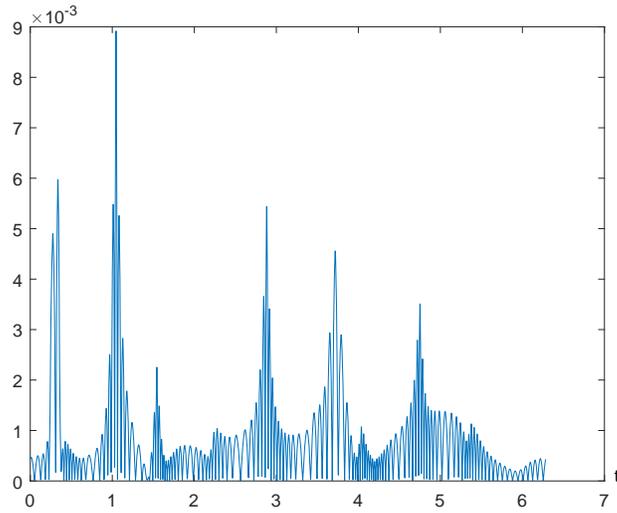


FIG. 4.4. Deviation of approximate boundary from true amoeba boundary for degree  $n = 64$  and  $\rho = 400$

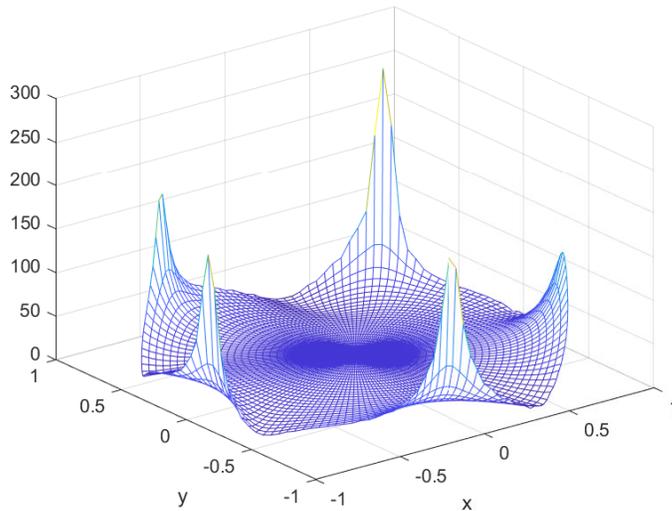


FIG. 4.5. Jacobian for amoeba polynomial transformation of degree 64.

**5. Application to Solving A Dirichlet Problem.** We apply the above ideas to solving the standard Dirichlet problem

$$(5.1) \quad \begin{aligned} -\Delta u(s, t) + d(s, t)u(s, t) &= f(s, t), & (s, t) \in \Omega, \\ u(s, t) &= 0, & (s, t) \in \Gamma. \end{aligned}$$

We use polynomials such as those illustrated in §4. As is illustrated in Table 4.1, the polynomial mapping does not precisely map  $\mathbb{B}^2$  onto  $\Omega$ . The table shows there is a

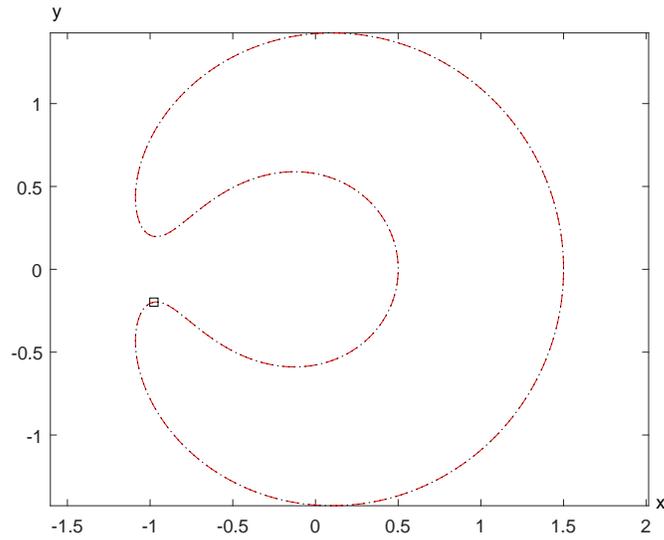


FIG. 4.6. Showing the point of maximum deviation of the degree 64 approximating polynomial to the pacman with  $\rho = 400$ .

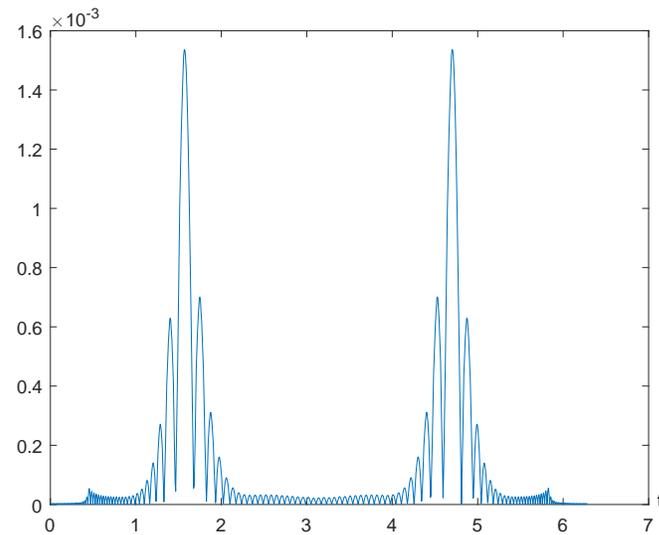


FIG. 4.7. Deviation of approximate boundary from true pacman boundary for  $n = 64$  and  $\rho = 400$ .

small error in reproducing the boundary  $\Gamma$ . For a general analysis of how this affects the accuracy in solving (5.1), see the study [7]. In short and over-simplifying, if the boundary  $\Gamma$  is perturbed by an amount  $\varepsilon$ , then the solution of (5.1) is perturbed by  $\mathcal{O}(\varepsilon)$ . The numerical method is Galerkin's method as described in [1, Chaps. 1,4]. We give a numerical example over each region.

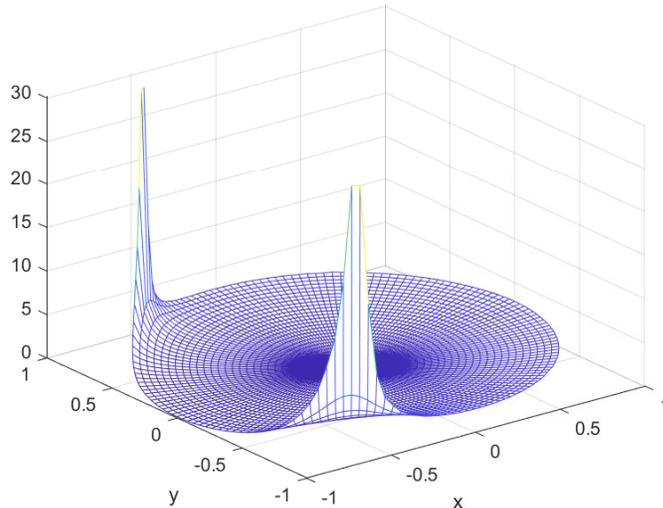


FIG. 4.8. *Jacobian for pacman polynomial transformation of degree 64.*

**5.1. Example for amoeba region.** For  $\Omega$  the amoeba region, we solve

$$(5.2) \quad \begin{aligned} -\Delta u(s, t) + e^{s-t} u(s, t) &= f(s, t), & (s, t) \in \Omega, \\ u(s, t) &= 0, & (s, t) \in \partial\Omega. \end{aligned}$$

The function  $f$  is so chosen that the true solution is

$$(5.3) \quad u(s, t) = (1 - x^2 - y^2) e^{0.15t} \cos(0.15\pi s),$$

where  $(s, t) \in \Omega$  corresponds to  $(x, y) \in \mathbb{B}^2$  under our polynomial transformation of degree  $n = 64$ . The polynomial transformation is derived from a spline interpolation of the boundary with  $\rho = 400$  subdivisions. The order of the associated Galerkin linear system is  $N_n$ . The solution (5.3) is shown in Figure 5.1. The maximum errors are shown in Figure 5.2. The straight line is a linear least squares fit to the data  $[(n, \log me) : n = 6, \dots, 35]$  where  $me$  denotes the maximum error associated with each degree  $n$ . Empirically the maximum error behaves like  $\mathcal{O}(n^{-2.94})$  or  $\exp(-0.165n)$ , the latter indicating an exponential rate of convergence to zero of the error. The condition number of the associated Galerkin linear system is experimentally  $\mathcal{O}(n^{1.02})$ .

**5.2. Example for pacman region.** The problem being solved is again (5.2), but now the function  $f$  is so chosen that the true solution is

$$(5.4) \quad u(s, t) = (1 - x^2 - y^2) e^{0.5t} \cos(0.5\pi s),$$

where  $(s, t) \in \Omega$  corresponds to  $(x, y) \in \mathbb{B}^2$ . The polynomial transformation of degree  $n = 64$  was constructed for the pacman region in the same manner as previously for the amoeba region. The solution (5.4) is shown in Figure 5.3. The maximum errors are shown in Figure 5.4. Empirically the maximum error behaves like  $\mathcal{O}(n^{-5})$  or  $\exp(-0.25n)$ , the latter indicating an exponential rate of convergence to zero of the error. The condition number of the associated Galerkin linear system is experimentally  $\mathcal{O}(n^{3.4})$ . It is unknown why it is so much worse than for the amoeba region.

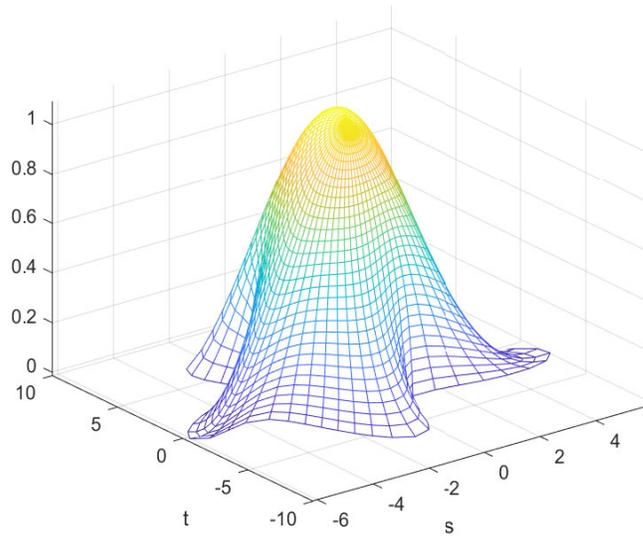


FIG. 5.1. *Solution (5.3) over amoeba region.*

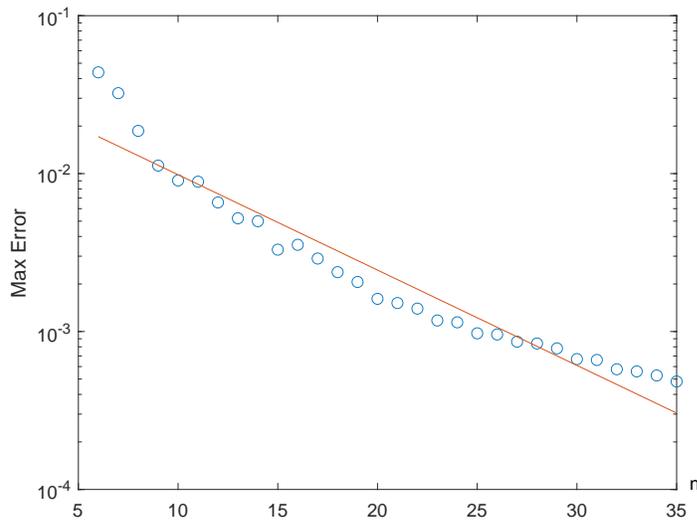


FIG. 5.2. *Maximum error when solving (5.2) over amoeba for varying Galerkin degree*

**5.3. Example for pacman region, polynomial solution.** The problem being solved is again (5.2), but now the function  $f$  is chosen that the true solution is

$$(5.5) \quad u(s, t) = (1 - x^2 - y^2)(s + t),$$

The polynomial transformation  $\Phi$  of degree  $n = 16$  was used for the pacman region here for demonstration purposes. This implies that  $s(x, y)$  and  $t(x, y)$  are polynomials of degree 16 on the unit disk. If we ensure that the quadrature error for the discrete

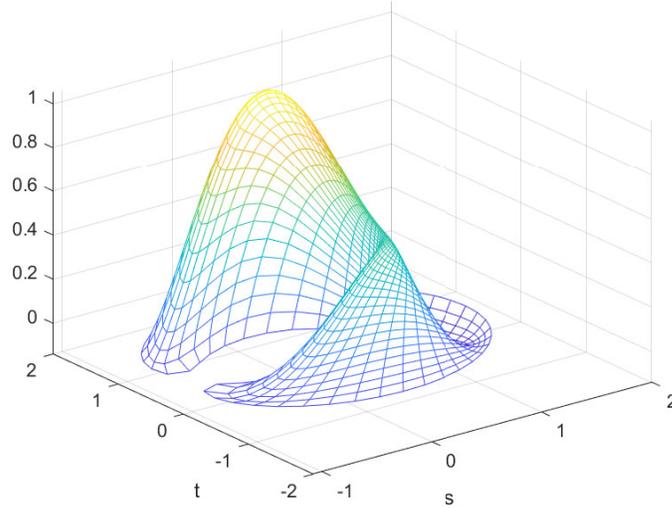


FIG. 5.3. *Solution (5.4) over pacman region.*

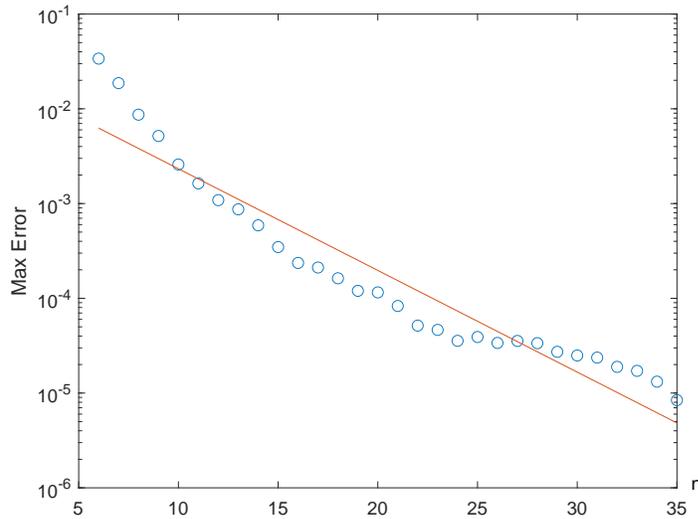


FIG. 5.4. *Maximum error when solving (5.2) over the pacman region for varying Galerkin degree  $n$*

Galerkin equation system does not dominate the approximation then we expect a sharp drop off of the error for approximation degree 16. If we would be able to calculate the discrete Galerkin system exactly the error should be in the rounding error range. But as one can see in [1, Section 4.2.1] the calculation of the Galerkin matrix involves the integration of the inverse of the determinant of the Jacobian matrix of the transformation mapping  $\Phi$ . This can usually only be done with the

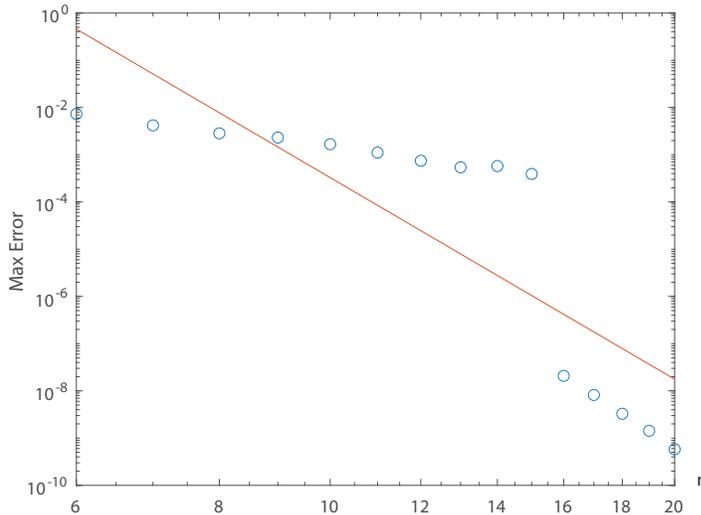


FIG. 5.5. Maximum error when solving (5.2) with right hand side (5.5) over the pacman region for varying Galerkin degree  $n$

help of numerical integration.

The maximum errors are shown in Figure 5.5. We see the drop off of the error for degree 16 and any further decrease of the error is due to the coupling of  $\eta$  in the Gauss–Legendre-trapezoidal formula (2.9) to the approximation polynomial degree. The order  $\eta$  of the Gauss–Legendre-trapezoidal formula in this example was 4 times the order of the maximum polynomial degree of the trial functions.

#### 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, D. Chien, and O. Hansen, *Constructing diffeomorphisms between simply connected plane domains*, *Electronic Transactions on Numerical Analysis*, 55 (2022), pp. 671–686.
- [3] K. Atkinson, and O. Hansen, *Creating domain mappings*, *Electronic Transactions on Numerical Analysis*, 39 (2012), pp. 202–230.
- [4] C. de Boor, *A Practical Guide to Splines*, Springer, New York Heidelberg Berlin, USA, 1978.
- [5] M.C. Delfour, J.-P. Zolésio, *Shapes and Geometries*, SIAM Advances in Control and Design, SIAM, Philadelphia, USA, 2011.
- [6] M.C. Delfour, A. Garon, *Transfinite Interpolations for Free and Moving Boundary Problems*, *Pure and Applied Functional Analysis*, 4 (2019), pp. 765–801.
- [7] D. Henry, *Perturbation of the Boundary in Boundary-Value Problems of Partial Differential Equations*, Cambridge University Press, 2005.
- [8] M. Saba, T. Schneider, K. Hormann, R. Scateni, *Curvature-based blending of closed planar curves*, *Graphical Models*, 5 (2014), pp. 263–272.
- [9] 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.
- [10] T. Surazhsky, G. Elber, *Metamorphosis of Plan Parametric Curves via Curvature Interpolation*, *International Journal of Shape Modeling*, 2 (2002), pp. 201–216.