ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Reconstruction of domains with algebraic boundaries from generalized polarization tensors

H. Ammari and M. Putinar and A. Steenkamp and F. Triki

Research Report No. 2019-25
May 2019

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding SNF: 200021--172483

RECONSTRUCTION OF DOMAINS WITH ALGEBRAIC BOUNDARIES FROM GENERALIZED POLARIZATION TENSORS

HABIB AMMARI, MIHAI PUTINAR, ANDRIES STEENKAMP, AND FAOUZI TRIKI

Abstract. This paper aims at showing the stability of the recovery of a smooth planar domain with a real algebraic boundary from a finite number of its generalized polarization tensors. It is a follow-up of the work [H. Ammari et al., Math. Annalen, 2018], where it is proved that the minimal polynomial with real coefficients vanishing on the boundary can be identified as the generator of a one dimensional kernel of a matrix whose entries are obtained from a finite number of generalized polarization tensors. The recovery procedure is implemented without any assumption on the regularity of the domain to be reconstructed and its performance and limitations are illustrated.

Contents

1. Introduction

Let D be a bounded connected Lipschitz domain in $\mathbf{R}^2$, and assume that its boundary ∂D contains the origin. Let Υ be the conductivity distribution in $\mathbf{R}^2$ given by

$$\Upsilon = k\chi(D) + \chi(\mathbf{R}^2 \setminus \overline{D}),$$

where χ denotes the indicator function, and k is a fixed constant in $(0,1) \cup (1,+\infty)$. Let $u_0(x) = \frac{1}{2\pi}\ln|x|$, be the fundamental solution to the Laplacian in $\mathbf{R}^2$. For a given position z in $\mathbf{R}^2$, we consider the following conductivity equation

$$\begin{cases} \nabla \cdot \Upsilon \nabla u(x,z) = \delta_z(x) & \text{in } \mathbf{R}^2, \\ u(x,z) - u_0(x,z) = O(|x|^{-1}) & \text{as } |x| \to \infty, \end{cases}$$

where δ_z is the Dirac function at z and $u_0(x,z) := u_0(x-z)$. The system (1) has a unique solution u which is the total voltage potential generated by the point source placed at z [7]. The function $-\nabla u_0(x,z)$ represents the background electric field while $u(x,z) - u_0(x,z)$ is the perturbation of the voltage potential

Date: May 2, 2019.

1991 *Mathematics Subject Classification.* Primary: 35R30, 35C20.

Key words and phrases. inverse problems, generalized polarization tensors, algebraic domains, shape classification.

This work was supported in part by the grant ANR-17-CE40-0029 of the French National Research Agency ANR (project MultiOnde), the LabEx PERSYVAL-Lab (ANR-11-LABX- 0025-01), and the Swiss National Science Foundation grant number 200021–172483.

due to presence of the inclusion D. Then, the far-field perturbation of the voltage potential due to the presence of D is given by [7]

$$u(x,z) - u_0(x,z) = \sum_{|\alpha|,|\beta|=1}^{\infty} \frac{(-1)^{|\alpha|+|\beta|}}{\alpha!\beta!} \partial^\alpha u_0(x) \mathbf{M}_{\alpha\beta} \partial^\beta u_0(z) \qquad \text{as } |x| \to +\infty, \tag{1}$$

where and throughout this paper, we use the conventional notation:

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2}, \quad \alpha = (\alpha_1, \alpha_2) \in \mathbf{N}^2, \quad \text{and } |\alpha| = \alpha_1 + \alpha_2.$$

We also use the graded lexicographic order: α, $\beta \in \mathbf{N}^2$ verifies $\alpha \leq \beta$ if $|\alpha| < |\beta|$, or, if $|\alpha| = |\beta|$, then $\alpha_1 \leq \beta_1$ or $\alpha_1 = \beta_1$ and $\alpha_2 \leq \beta_2$.

The quantities $\mathbf{M}_{\alpha\beta}$ that appear naturally in the multi-polar asymptotic expansion (1), are called Generalized Polarization Tensors (GPTs). We emphasize that GPTs are not dependent on the positions x and z. In fact they only depend on the inclusion D and the conductivity ratio $1/k$ or conductivity contrast $\lambda := \frac{k+1}{2(k-1)}$. For a fixed contrast λ, the GPTs are indeed geometric quantities associated with the shape of the domain D such as eigenvalues, capacities, and moments. The notion of GPTs has been used in diverse fields of academic research as well as of engineering applications such as the theories of composites, inverse problems, bio-medical imaging, bio-sensing, nano-sensing, and electro-sensing [5, 9–13, 15, 26].

From the asymptotic expansion (1), we deduce that the knowledge of all the GPTs is equivalent to knowing the far-field responses of the inclusion for all harmonic excitations. It is well known that in that case the inverse problem of recovering (λ, D) has a unique solution [6], and a number of algorithms have been proposed for its numerical treatment [3, 4, 7, 8]. However, in applications, the GPTs are usually only measured with finite accuracy and only a finite number of them can be determined from noisy data. Hence, studying the well-posedness of the inverse problem when only a finite number of GPTs are available is of importance.

The purpose of this paper is to evaluate how much information one can get from the knowledge of a finite number of these GPTs. Precisely, assuming that the domain has an algebraic boundary, we are interested in the inverse problem of recovering its position, its shape and the contrast for given a finite number of its GPTs. Recently the uniqueness to this inverse problem was established by the same authors [1]. Our goal in the present paper is twofold: (i) to quantify the stability of the inversion and (ii) to implement the inversion procedure and apply it to much more general cases than those discussed in [1]. In particular, we show here how to recover the true domain (with possibly nonsmooth boundary) from the recovered polynomial level set even in the case where several candidate domains have the same polynomial level set. In doing so, we resolve key numerical issues which include handling of bifurcation points, segmentation points, and arc sets. It is worth emphasizing that the stability estimates proved in this paper holds for algebraic domains with smooth boundaries. Their generalization to the nonsmooth case is technically quite challenging.

The paper is organized as follows. In Section 2, we introduce the class under consideration of domains with algebraic boundaries. Stability issues are studied in Section 3. The main stability estimates are given in Theorem 3.2. Section 4 is devoted to the presentation of our new numerical algorithm which is designed to recover algebraic domains from finite numbers of their associated GPTs. It is worth mentioning that based on the density with respect to Hausdorff distance of algebraic domains among all bounded domains, the proposed algorithm can be extended via approximation beyond its natural context. This observation has already turned algebraic curves into an efficient tool for describing shapes and reconstructing them from their associated moments [19–23, 25].

2. Real algebraic domains

In this section, we introduce the class of bounded open subsets in $\mathbf{R}^2$ with real algebraic boundaries. We recall the following definition.

Definition 2.1. *An open set G in $\mathbf{R}^2$ is called real algebraic (or simply algebraic) if there exists a finite number of real coefficient polynomials $g_i(x), i = 1, \cdots, m$, such that*

$$\partial G \subset V := \{x \in \mathbf{R}^2 \ : g_1(x) = \cdots = g_m(x) = 0\}.$$

The ellipse is a simple example of an algebraic domain, since its general boundary coincides with the zero set of the quadratic polynomial function

$$g(x)=\sum_{|\alpha|\leq 2} g_\alpha x^\alpha$$

for given real coefficients $(g_\alpha)_{|\alpha|\leq 2}$ and proper signs in the top degree part.

We further denote by $\mathcal{G}$ the collection of bounded algebraic domains. It is well-known that the differential structure of the boundary ∂G consists of algebraic arcs joining finitely many singular points, see for instance [16].

As mentioned in [1], since the connectedness of the respective sets is not accessible by the linear algebra tools we developed for reconstructing an algebraic domain from a finite number of its generalized polarization tensors, we drop such a constraint here. Nevertheless, we call "domains" all elements $G\in\mathcal{G}$.

Following [23] we consider a particular class of algebraic domains which are better adapted to the uniqueness and stability of our inverse shape problem. Let

$$\mathcal{G}^* := \left\{G\in\mathcal{G} : G=\operatorname{int}\overline{G}\right\}. \tag{2}$$

An element of $\mathcal{G}^*$ is called an *admissible domain*, although it may not be connected.

The assumption that $G=\operatorname{int}\overline{G}$ implies that G contains no slits or ∂G does not have isolated points. If $G\in\mathcal{G}^*$, the algebraic dimension of ∂G is one, and the ideal associated to it is principal. To be more precise, ∂G is contained in a finite union of irreducible algebraic sets $X_j,\ j\in J$, of dimension one each. The reduced ideal associated to every X_j is principal:

$$I(X_j)=(P_j),\quad j\in J;$$

see, for instance, [14, Theorem 4.5.1]. We assume that each P_j is indefinite, i.e., it changes sign when crossing X_j. Therefore, one can consider the polynomial $g=\prod_{j\in J}P_j$, vanishing of the first-order on ∂G, that is $|\nabla g|\neq 0$ on the regular locus of ∂G. According to the real version of Study's lemma (cf. [16, Theorem 12]) every polynomial vanishing on ∂G is a multiple of g, that is $I(\partial G)=(g)$. We define the degree of ∂G as the degree of the generator g of the ideal $I(\partial G)$. For a thorough discussion of the reduced ideal of a real algebraic surface in $\mathbf{R}^d$, we refer the reader to [17].

Throughout this paper, we denote by $g(x)$ the single polynomial vanishing on ∂G which is the generator of $I(\partial G)$ and satisfying the following normalization condition $g_{\alpha^*}=1$, where $\alpha^*=\max_{g_\alpha\neq 0}\alpha$. We further assume that $G\in\mathcal{G}^*$.

3. Uniqueness and stability estimates

In this section, we first recall the uniqueness result obtained in [1] and then derive stability estimates for the inversion procedure for smooth algebraic domains.

3.1. Uniqueness.

Let $\mathbf{R}[x]$ be the ring of polynomials in the variables $x=(x_1,x_2)$ and let $\mathbf{R}_n[x]$ be the vector space of polynomials of degree at most n (whose dimension is $r_n=(n+1)(n+2)/2$). Any polynomial function $p(x)\in\mathbf{R}_n[x]$ has a unique expansion in the canonical basis $x^\alpha, |\alpha|\leq n$ of $\mathbf{R}_n[x]$, that is,

$$p(x)=\sum_{|\alpha|\leq n} p_\alpha x^\alpha$$

for some vector coefficients $\mathbf{p}=(p_\alpha)\in\mathbf{R}_{r_n}$. The following results are established in [1].

Theorem 3.1. *Let $G\in\mathcal{G}^*$ with ∂G Lipschitz of degree d, and let $g(x)=\sum_{|\alpha|\leq d} g_\alpha x^\alpha$, be a polynomial function that vanishes of the first-order on ∂G, satisfying $I(\partial G)=(g)$, $g_{\alpha^*}=1$, and $g(0)=0$, where $\alpha^*=\max_{g_\alpha\neq 0}\alpha$. Then, there exists a discrete set $\Sigma\subset\mathbf{C}_0:=\mathbf{C}\setminus[-1/2,1,2]$, such that for any fixed $\lambda\in\mathbf{C}_0\setminus\Sigma$, $\mathbf{g}=(g_\alpha)\in\mathbf{R}_{r_d}$ is the unique solution to the following normalized linear system:*

$$\mathbf{p}=(p_\alpha)\in\mathbf{R}_{r_d};\quad \sum_{|\beta|\leq d}\mathbf{M}_{\alpha\beta}(\lambda,G)p_\beta=0 \ \textit{ for } |\alpha|\leq 2d;\ \ p_{\alpha^*}=1,\ \ \alpha^*=\max_{p_\alpha\neq 0}\alpha. \tag{3}$$

Corollary 3.1. *Let $G, \widetilde{G} \in \mathcal{G}^*$ be Lipschitz of degree d. Let g and $\widetilde{g}$ be two polynomials that vanish respectively of the first order on ∂G and on $\partial \widetilde{G}$ satisfying $I(\partial G) = (g)$ and $I(\partial \widetilde{G}) = (\widetilde{g})$. Assume that $g(0) = \widetilde{g}(0) = 0$ and $\|\nabla g\|, \|\nabla \widetilde{g}\| > 0$ on respectively ∂G and $\partial \widetilde{G}$. Moreover, assume that G is the unique element of $\mathcal{G}^*$ containing 0 such that $\partial G \subset \{g = 0\} \cup B_r(0)$, where $B_r(0)$ is the disk of center 0 and radius r large enough. Let λ and $\widetilde{\lambda}$ be fixed in $\mathbb{C}_0$ such that $\lambda \notin \Sigma$, $\widetilde{\lambda} \notin \widetilde{\Sigma}$, where the sets $\Sigma(\partial G)$ and $\widetilde{\Sigma} = \Sigma(\partial \widetilde{G})$ are as defined in Theorem 3.1. Then, the following uniqueness result holds:*

$$(4) \qquad (\mathbf{M}_{\alpha\beta}(G, \lambda))_{|\alpha| \leq 2d, 0 < |\beta| \leq d} = (\mathbf{M}_{\alpha\beta}(\widetilde{G}, \widetilde{\lambda}))_{|\alpha| \leq 2d, 0 < |\beta| \leq d} \quad \textit{iff} \quad G = \widetilde{G} \ \textit{ and } \ \lambda = \widetilde{\lambda}.$$

Proof. The result is a direct consequence of Theorem 3.1. Since the generalized polarization tensors coincide, and $\lambda \notin \Sigma$, we can deduce from Theorem 3.1 that $g = \widetilde{g}$. The fact that $g(0) = \widetilde{g}(0) = 0$ and $\|\nabla g\|, \|\nabla \widetilde{g}\| > 0$ on respectively ∂G and $\partial \widetilde{G}$ implies that $G = \widetilde{G}$. A straightforward calculation shows then that $\lambda = \widetilde{\lambda}$, which finishes the proof. □

3.2. **Stability estimates.** In this section we derive, under some regularity assumption, stability estimates for the considered inverse problem. For fixed integer $d > 0$, and constants $R > 0$, $M_0 > 0$, $\kappa > 0$, define a reduced set of algebraic domains $\mathcal{G}_0^*$ by

$$(5) \quad \mathcal{G}_0^* := \left\{ G \in \mathcal{G}_0^* : G \subset B_R(0),\ I(\partial G) = (g),\ g(0) = 0,\ \deg(g) = d,\ \|\mathbf{g}\| \leq M_0,\ \min_{\partial G} \|\nabla g\| \geq \kappa \right\},$$

where deg denotes the degree. It is not difficult to show that there exists a constant $M > M_0$, that only depends on $\mathcal{G}_0^*$, such that

$$(6) \qquad |g|,\ \|\nabla g\|,\ \|H(g)\| \leq M \quad \text{on } B_R(0)$$

for all g satisfying $I(\partial G) = (g)$, where $G \in \mathcal{G}_0^*$ and $H(g)$ is the Hessian matrix of g.

Let K_1 and K_2 be two compact sets in $\mathbf{R}^2$. Recall that the Hausdorff distance between K_1 and K_2 is defined by

$$\mathbf{d_H}(K_1, K_2) = \max \left\{ \sup_{x \in K_1} \mathbf{d}(x, K_2),\ \sup_{x \in K_2} \mathbf{d}(x, K_1) \right\},$$

where $\mathbf{d}(x, K_i) = \inf_{y \in K_i} \|x - y\|$, $i = 1, 2$. Let $\|\ \|$ denote the Euclidean norm of tensors.

Theorem 3.2. *Let $G \in \mathcal{G}_0^*, \widetilde{G} \in \mathcal{G}_0^*$ with respectively ∂G and $\partial \widetilde{G}$. Let $\delta > 0$ be a fixed constant and $\lambda_0 \in \mathbf{R}$ satisfying $B_\delta(\lambda_0) \Subset \mathbf{C} \cap \left\{ |\lambda| > \frac{1}{2} \right\}$. Then there exists $\lambda^* \in (\lambda_0 - \delta, \lambda_0 + \delta)$, constants $\eta = \eta(\lambda_0, \delta, \mathcal{G}_0^*) \in (0, 1)$, and $C = C(\lambda_0, \delta, \mathcal{G}_0^*) > 0$, such that if*

$$\sum_{|\alpha| \leq 2d, 0 < |\beta| \leq d} \left\| \mathbf{M}_{\alpha\beta}(\lambda^*, G) - \mathbf{M}_{\alpha\beta}(\lambda^*, \widetilde{G}) \right\|^2 = \varepsilon^2 < 1,$$

then the following stability result holds:

$$(7) \qquad \mathbf{d_H}(\partial G, \partial \widetilde{G}) \leq C \varepsilon^\eta.$$

In order to prove Theorem 3.2, we need to show several intermediate results. Let $g(x) = \sum_{|\alpha| \leq d} g_\alpha x^\alpha$ and $\widetilde{g}(x) = \sum_{|\alpha| \leq d} \widetilde{g}_\alpha x^\alpha$ be respectively polynomial functions that vanish respectively of the first-order on ∂G and $\partial \widetilde{G}$ satisfying $I(\partial G) = (g)$, $g_{\alpha^*} = 1$, $g(0) = 0$, and $I(\partial \widetilde{G}) = (\widetilde{g})$, $\widetilde{g}_{\alpha^*} = 1$, $\widetilde{g}(0) = 0$.

Further, we shall use standard notation concerning Sobolev spaces. For a density $\phi \in H^{-1/2}(\partial G)$, define the Neumann-Poincaré operator: $\mathcal{K}_G^* : H^{-1/2}(\partial G) \to H^{-1/2}(\partial G)$, by

$$\mathcal{K}_G^*[\phi](x) = \frac{1}{2\pi} \text{p.v.} \int_{\partial G} \frac{\langle x - y, \nu_G(x) \rangle}{\|x - y\|^2} \phi(y)\, d\sigma(y), \quad x \in \partial G,$$

where p.v. denotes the principal value, $\nu_G(x)$ is the outward unit normal to ∂G at $x \in \partial G$, $\langle\ ,\ \rangle$ denotes the scalar product in $\mathbf{R}^2$, and $\|\ \|$ denotes the Euclidean norm in $\mathbf{R}^2$.

The following lemma characterizes the resolvent set $\rho(\mathcal{K}_G^*)$ of the operator $\mathcal{K}_G^*$, see, for instance, [7] and [18].

Lemma 3.1. *We have* $\mathbb{C} \setminus (-1/2, 1/2] \subset \rho(\mathcal{K}_G^*)$. *Moreover, if* $|\lambda| \geq 1/2$, *then* $(\lambda I - \mathcal{K}_G^*)$ *is invertible on* $H_0^{-1/2}(\partial G) := \{f \in H^{-1/2}(\partial G) : \langle f, 1\rangle_{-1/2,1/2} = 0\}$. *Here,* $\langle\ ,\ \rangle_{-1/2,1/2}$ *denotes the duality pairing between* $H^{-1/2}(\partial G)$ *and* $H^{1/2}(\partial G)$.

For $|\lambda| > 1/2$ and a multi-index $\alpha = (\alpha_1, \alpha_2) \in \mathbf{N}^2$, define ϕ_α by

$$\phi_\alpha(y) := (\lambda I - \mathcal{K}_G^*)^{-1}\left[\nu_G(x) \cdot \nabla x^\alpha\right](y), \quad y \in \partial G.$$

The GPTs $\mathbf{M}_{\alpha\beta}$ for $\alpha, \beta \in \mathbf{N}^2$ ($|\alpha|, |\beta| \geq 1$), associated with the contrast λ and the domain G can be rewritten as [7]

$$\mathbf{M}_{\alpha\beta}(\lambda, G) := \int_{\partial G} y^\beta \phi_\alpha(y)\, d\sigma(y). \tag{8}$$

Denote by $\mathbf{C}_\star := \mathbf{C} \setminus (-\infty, -2] \cup [2, +\infty)$, and let $\mu = \lambda^{-1} \in \mathbb{C}_\star$. Define respectively $\mathbb{M}(\mu)$ and $\widetilde{\mathbb{M}}(\mu)$ to be the rectangular matrices with coefficients:

$$\mathbb{M}_{\alpha\beta}(\mu) := \int_{\partial G} (I - \mu\mathcal{K}_G^*)^{-1}\left[\nu_G(x) \cdot \nabla x^\alpha\right] y^\beta\, d\sigma(y), \tag{9}$$

$$\widetilde{\mathbb{M}}_{\alpha\beta}(\mu) := \int_{\partial\widetilde{G}} (I - \mu\mathcal{K}_{\widetilde{G}}^*)^{-1}\left[\nu_{\widetilde{G}}(x) \cdot \nabla x^\alpha\right] y^\beta\, d\sigma(y). \tag{10}$$

Note that $\mathbf{M}_{\alpha\beta}(\lambda, G) = \lambda\mathbb{M}_{\alpha\beta}(1/\lambda)$ and $\mathbf{M}_{\alpha\beta}(\lambda, \widetilde{G}) = \lambda\widetilde{\mathbb{M}}_{\alpha\beta}(1/\lambda)$.
Recall the following result from [1].

Lemma 3.2. *The functions* $\mu \to \mathbb{M}(\mu), \widetilde{\mathbb{M}}(\mu) \in \mathcal{L}\left(\mathbf{R}^{r_d}, \mathbf{R}^{r_{2d}}\right)$ *are holomorphic matrix-valued on* $\mathbf{C}_\star$. *In addition, ker*$(\mathbb{M}(0)) = \{c\mathbf{g};\ \ c \in \mathbf{R}\}$ *and ker*$(\widetilde{\mathbb{M}}(0)) = \{c\widetilde{\mathbf{g}};\ \ c \in \mathbf{R}\}$.

The proof of Theorem 3.2 has two main steps. In the first step, using the normalized linear system (3), we estimate $\mathbf{g} - \widetilde{\mathbf{g}}$ in terms of $\mathbb{M}(0) - \widetilde{\mathbb{M}}(0)$. The second step consists in applying the unique continuation of holomorphic functions on $\mathbb{M}(\mu) - \widetilde{\mathbb{M}}(\mu)$ to "propagate the information" from 0 to $\mu = \lambda^{-1}$.

Let

$$F(\mu) := \sum_{|\alpha| \leq 2d, 0 < |\beta| \leq d} \left\|\mathbb{M}_{\alpha\beta}(\mu) - \widetilde{\mathbb{M}}_{\alpha\beta}(\mu)\right\|^2.$$

We remark that $F(\mu)$ is a real positive function on $\mathbf{C}_\star \cap \mathbf{R}$. We deduce from Lemma 3.2 that $F(\mu)$ is holomorphic on $\mathbf{C}_\star$ and that $F(0) = 0$ implies $\mathbf{g} = \widetilde{\mathbf{g}}$. We next estimate how much $\mathbf{g}$ is close to $\widetilde{\mathbf{g}}$ when $F(0)$ is very small.

Proposition 3.1. *Let the constants* κ *and* M *be defined by (5) and (6), respectively. Let* $\varepsilon_0 = \frac{\kappa^5}{65M^4}$ *and* $C = 64\frac{M^4}{\kappa^5}$. *Assume that* $F(0) \leq \varepsilon_0^2$. *Then the following inequality holds:*

$$\mathbf{d_H}^2(\partial G, \partial\widetilde{G}) \leq CF^{1/2}(0). \tag{11}$$

In order to prove Proposition 3.1 we need the following three lemmas.

Lemma 3.3. *We have*

$$\|g - \widetilde{g}\|_{L^2(\partial G)}^2 + \|g - \widetilde{g}\|_{L^2(\partial\widetilde{G})}^2 \leq 2\kappa^{-1}M^2F^{1/2}(0). \tag{12}$$

Proof. From the definition of the matrices $\mathbb{M}(0)$ and $\widetilde{\mathbb{M}}(0)$, we have

$$\mathbf{q}^t\left(\mathbb{M}(0)-\widetilde{\mathbb{M}}(0)\right)\mathbf{p}=\int_{\partial G}\nu_G(y)\cdot\nabla q(y)p(y)\,d\sigma(y)-\int_{\partial\widetilde{G}}\nu_{\widetilde{G}}(y)\cdot\nabla q(y)p(y)\,d\sigma(y),\quad\forall p\in\mathbf{R}_d[x],\,q\in\mathbf{R}_{2d}[x],\tag{13}$$

where the superscript t denotes the transpose.

Since g and $\widetilde{g}$ are in $\mathcal{G}^*$ defined by (2) and they respectively generate the ideals associated to ∂G and $\partial\widetilde{G}$, we have $\nu_G(x)=\frac{\nabla g(x)}{\|\nabla g(x)\|}$, $x\in\partial G$ and $\nu_{\widetilde{G}}(x)=\frac{\nabla\widetilde{g}(x)}{\|\nabla\widetilde{g}(x)\|}$, $x\in\partial\widetilde{G}$. Then (13) becomes

$$\mathbf{q}^t\left(\mathbb{M}(0)-\widetilde{\mathbb{M}}(0)\right)\mathbf{p}=\int_{\partial G}\frac{\nabla g}{\|\nabla g\|}\cdot\nabla q(y)p(y)\,d\sigma(y)-\int_{\partial\widetilde{G}}\frac{\nabla\widetilde{g}}{\|\nabla\widetilde{g}\|}\cdot\nabla q(y)p(y)\,d\sigma(y),\forall p\in\mathbf{R}_d[x],\,q\in\mathbf{R}_{2d}[x].$$

By taking $q(x)=\widetilde{g}(x)g(x)$, $p(x)=g(x)+\widetilde{g}(x)$, and considering the fact that $g(x)$ and $\widetilde{g}(x)$ respectively vanish on ∂G and on $\partial\widetilde{G}$, one finds that

$$(\widetilde{\mathbf{g}}\mathbf{g})^t\left(\mathbb{M}(0)-\widetilde{\mathbb{M}}(0)\right)(\mathbf{g}+\widetilde{\mathbf{g}})=\int_{\partial G}\|\nabla g\|\left(g-\widetilde{g}\right)^2\,d\sigma+\int_{\partial\widetilde{G}}\|\nabla\widetilde{g}\|\left(g-\widetilde{g}\right)^2\,d\sigma,$$

which in turn implies that

$$\int_{\partial G}\|\nabla g\|\left(g-\widetilde{g}\right)^2\,d\sigma+\int_{\partial\widetilde{G}}\|\nabla\widetilde{g}\|\left(g-\widetilde{g}\right)^2\,d\sigma\leq 2\left(\|\mathbf{g}\|^2+\|\widetilde{\mathbf{g}}\|^2\right)F^{1/2}(0).$$

Hence, (12) holds. □

For $r>0$ small, let $\mathscr{O}_r\subset\mathbf{R}$ being the tubular domain along ∂G, defined by

$$\mathscr{O}_r:=\left\{y+s\nu_G(y);\,y\in\partial G,\,s\in(-r,r)\right\}.$$

Lemma 3.4. *Assume that $0<r\leq\frac{\kappa}{M}$. Then*

$$|g(x)|\geq\frac{\kappa}{2}r,\quad\forall x\in\partial\mathscr{O}_r.\tag{14}$$

Proof. Let $x=y+\pm r\nu_G(y)\in\mathscr{O}_r$, for some $y\in\partial G$ be fixed. From the regularity of g, it follows that the function $s\to g(y\pm s\nu_G(y))$ is C^2 and satisfies the following Taylor expansion of order two at zero:

$$g(y\pm r\nu_G(y))=\pm\nabla g(y)\cdot\nu_G(y)r+\frac{r^2}{2}\nu_G^t(y)H(g)(y\pm s_0\nu_G(y))\nu_G(y),$$

where $H(g)(y)$ is the Hessian matrix of g at y, and s_0 is some constant in between 0 and $\pm r$. Recalling that $\nu_G(x)=\frac{\nabla g(x)}{\|\nabla g(x)\|}$, we therefore obtain that

$$|g(x)|\geq\kappa r-M\frac{r^2}{2},$$

which finishes the proof.

□

The proof of Lemma 3.4 shows that if the zero level set of g is isolated, that is, $\|\nabla g\|\neq 0$ on ∂G, then the polynomial g behaves as a weighted signed distance function to the boundary ∂G in the small tubular neighborhood domain $\mathscr{O}_r$.

Lemma 3.5. *Let $r^*=\frac{\kappa}{M}$ and $\varepsilon_0=\frac{\kappa^5}{65M^4}$. Assume that $F(0)\leq\varepsilon_0^2$. Then*

$$\partial\widetilde{G}\subset\mathscr{O}_{r^*}.\tag{15}$$

Proof. Let $\widetilde{x}(t)$ be the parametric representation of the boundary $\partial\widetilde{G}$ ($\partial\widetilde{G} = \{\widetilde{x}(t),\ t \in \mathbf{R}_+\}$) satisfying

$$\frac{d\widetilde{x}}{dt}(t) = J\nabla\widetilde{g}(\widetilde{x}(t)), \quad t > 0, \ \text{ and } \ \widetilde{x}(0) = 0, \tag{16}$$

where J is the counter-clockwise rotation matrix by $\pi/2$. Since $\widetilde{g}$ is smooth, $\widetilde{x}(t)$ is the unique solution to the system (16), which is in addition of class C^1 and is periodic on $\mathbf{R}_+$.

Now we shall prove that $\widetilde{x}(t)$ lies indeed in $\mathscr{O}_{r^*}$, for all $t \in \mathbf{R}_+$. Assume that $\partial\widetilde{G}$ is not entirely included in $\mathscr{O}_{r^*}$, and define

$$t_0 = \sup\{t \in \mathbf{R}_+ : \ \widetilde{x}(t) \in \mathscr{O}_{r^*}\}.$$

Since $0 \in \partial G$, $t_0 > 0$ is well defined, is finite, and verifies $\widetilde{x}(t_0) \in \partial\mathscr{O}_{r^*}$. Lemma 3.4 then implies that

$$|g(\widetilde{x}(t_0))| \geq \frac{\kappa}{2}. \tag{17}$$

In view of the regularity of g and since $\widetilde{x}$ verifies (16), we have

$$|g(\widetilde{x}(t)) - g(\widetilde{x}(s))| \leq M^2|t-s|, \quad \forall s,t \in \mathbf{R}_+. \tag{18}$$

Combining inequalities (17) and (18), we obtain that

$$|g(\widetilde{x}(t))| \geq \frac{\kappa}{4}$$

for all t satisfying $|t - t_0| \leq \frac{\kappa}{4M^2}$. Whence

$$\|g - \widetilde{g}\|^2_{L^2(\partial\widetilde{G})} \geq \int_{t_0-\frac{\kappa}{4M^2}}^{t_0+\frac{\kappa}{4M^2}} |g(\widetilde{x}(t))|^2\|\nabla\widetilde{g}(\widetilde{x}(t))\|dt \geq \frac{\kappa^4}{32M^2}.$$

This together with (12) entail

$$F^{1/2}(0) \geq \frac{\kappa^5}{64M^4},$$

which is in contradiction with the fact that $F(0) \leq \varepsilon_0^2$. Then the inclusion (15) is satisfied.

□

Proof of Proposition 3.1. Now, we are ready to prove Proposition 3.1. We further assume that $F(0) \leq \varepsilon_0^2$. Let $\widetilde{x}(t)$ be defined by (16). Since $\partial\widetilde{G} \subset \mathscr{O}_{r^*}$, for each $t > 0$, there exists $r(t) \in (0, r^*)$ and $y(t) \in \partial G$, such that $x(t) = y(t) \pm r(t)\nu_G(y(t))$. Noting that $x(t) \in \mathscr{O}_{r(t)}$, we get from Lemma 3.4 the following estimate:

$$|g(\widetilde{x}(t))| \geq \frac{\kappa}{2}r(t).$$

Following the same arguments as those in the proof of Lemma 3.5, we get

$$|g(\widetilde{x}(s))| \geq \frac{\kappa}{4}r(t)$$

for all s satisfying $|t - s| \leq \frac{\kappa}{4M^2}$. Whence

$$\|g - \widetilde{g}\|^2_{L^2(\partial\widetilde{G})} \geq \int_{t_0-\frac{\kappa}{4M^2}}^{t_0+\frac{\kappa}{4M^2}} |g(\widetilde{x}(t))|^2\|\nabla\widetilde{g}(\widetilde{x}(t))\|dt \geq \frac{\kappa^4}{32M^2}r^2(t)$$

for all $t \in \mathbf{R}_+$. Then

$$\frac{\kappa^4}{32M^2}\mathbf{d}^2(\widetilde{x}(t), \partial G) \leq \frac{\kappa^4}{32M^2}r^2(t) \leq \|g - \widetilde{g}\|^2_{L^2(\partial\widetilde{G})}$$

for all $t \in \mathbf{R}_+$, which implies

$$\frac{\kappa^4}{32M^2}\sup_{x\in\partial\widetilde{G}}\mathbf{d}^2(x, \partial G) \leq \|g - \widetilde{g}\|^2_{L^2(\partial\widetilde{G})}. \tag{19}$$

Repeating the same steps by interchanging G and $\widetilde{G}$, we also get

$$\frac{\kappa^4}{32M^2}\sup_{x\in\partial G}\mathbf{d}^2(x, \partial\widetilde{G}) \leq \|g - \widetilde{g}\|^2_{L^2(\partial G)}. \tag{20}$$

Finally, combining inequalities (19), (20), and (12), we obtain the final result of Proposition 3.1. □

The second step in proving Theorem 3.2 consists in showing the following proposition.

Proposition 3.2. *Let $\delta > 0$ be a fixed constant and $\lambda_0 \in \mathbf{R}$ satisfying $B_\delta(\lambda_0) \Subset \mathbf{C} \cap \{|\lambda| > \frac{1}{2}\}$. Then, there exist constants $\theta = \theta(\lambda_0, \delta) > 0$ and $C = C(\lambda_0, \delta) > 0$ such that*

$$F(0) \leq C \left\| \lambda \mapsto F(\frac{1}{\lambda}) \right\|^{\theta}_{L^\infty((\lambda_0-\delta, \lambda_0+\delta))}. \tag{21}$$

Proof. Let $\omega \Subset B_2(0)$ be the image of $(\lambda_0 - \delta, \lambda_0 + \delta)$ by the complex function $\lambda \mapsto 1/\lambda$. Then there exists a constant $r_0 \in (0, 2)$ such that $\omega \Subset B_{r_0}(0)$. Denote by $M_1 = \|\mu \mapsto F(\mu)\|_{L^\infty(B_{r_0}(0))}$, and let w be the harmonic measure satisfying

$$\begin{cases} \Delta w &= & 0 & \text{in } B_{r_0}(0) \setminus \overline{\omega}, \\ w &= & 0 & \text{on } \partial B_{r_0}(0), \\ w &= & 1 & \text{on } \partial \omega. \end{cases}$$

Since $\mu \mapsto F(\mu)$ is holomorphic on $B_{r_0}(0)$, the function $\mu \mapsto \log |F(\mu)|$ is subharmonic, and we can deduce from the Two constants Theorem [24] the following inequality:

$$F(\mu) \leq M_1^{1-w(\mu)} \|\mu \mapsto F(\mu)\|^{w(\mu)}_{L^\infty(\omega)}.$$

Then by taking $\theta = w(0)$, and $C = M_1^{1-w(0)}$, we obtain the result. □

Proof of Theorem 3.2. Finally, we are now in a position to prove Theorem 3.2. Let $\lambda^* \in (\lambda_0 - \delta, \lambda_0 + \delta)$. By combining estimates (11) and (21) together with the fact that $\mathbf{M}_{\alpha\beta} = \lambda \mathbb{M}_{\alpha\beta}$, we finally obtain the desired stability result stated in Theorem 3.2. □

4. Algorithm description and numerical examples

4.1. **Algorithm.** Before we can dive into the algorithm for recovering algebraic domains from finitely many of their GPTs we must first define a processed form of the GPTs that will form our starting point. In [1, Algorithm 6.2] the GPTs $(\mathbf{M}_{\alpha\beta})_{|\alpha|\leq 2d, |\beta| \leq d}$ are flattened out into a linear system. We define one such system explicitly here. For doing so, we use the notation $\mathbf{M}_{\alpha\beta} = \mathbf{M}_{[\alpha_1,\alpha_2],[\beta_1,\beta_2]}$, where $\alpha = (\alpha_1, \alpha_2)$ and $\beta = (\beta_1, \beta_2)$.

Definition 4.1. *The GPT tessera of order (m,n) is given by*

$$\widetilde{\boldsymbol{M}}_{m,n} := \begin{bmatrix} \boldsymbol{M}_{[m,0],[n,0]}(\lambda, G) & \boldsymbol{M}_{[m,0],[n-1,1]}(\lambda, G) & \cdots & \boldsymbol{M}_{[m,0],[1,n-1]}(\lambda, G) & \boldsymbol{M}_{[m,0],[0,n]}(\lambda, G) \\ \boldsymbol{M}_{[m-1,1],[n,0]}(\lambda, G) & \boldsymbol{M}_{[m-1,1],[n-1,1]}(\lambda, G) & \cdots & \boldsymbol{M}_{[m-1,1],[1,n-1]}(\lambda, G) & \boldsymbol{M}_{[m-1,1],[0,n]}(\lambda, G) \\ \vdots & \vdots & \ddots & \vdots & \\ \boldsymbol{M}_{[1,m-1],[n,0]}(\lambda, G) & \boldsymbol{M}_{[1,m-1],[n-1,1]}(\lambda, G) & \cdots & \boldsymbol{M}_{[1,m-1],[1,n-1]}(\lambda, G) & \boldsymbol{M}_{[1,m-1],[0,n]}(\lambda, G) \\ \boldsymbol{M}_{[0,m],[n,0]}(\lambda, G) & \boldsymbol{M}_{[0,m],[n-1,1]}(\lambda, G) & \cdots & \boldsymbol{M}_{[0,m],[1,n-1]}(\lambda, G) & \boldsymbol{M}_{[0,m],[0,n]}(\lambda, G) \end{bmatrix}.$$

Definition 4.2. *The Tesselated GPT (TGPT) of order (d) is given by*

$$\boldsymbol{TGPT}_{2d,d} := \begin{bmatrix} \widetilde{\boldsymbol{M}}_{1,1} & \widetilde{\boldsymbol{M}}_{1,2} & \cdots & \widetilde{\boldsymbol{M}}_{1,d} \\ \widetilde{\boldsymbol{M}}_{2,1} & \widetilde{\boldsymbol{M}}_{2,2} & \cdots & \widetilde{\boldsymbol{M}}_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{\boldsymbol{M}}_{2d,1} & \widetilde{\boldsymbol{M}}_{2d,2} & \cdots & \widetilde{\boldsymbol{M}}_{2d,d} \end{bmatrix}.$$

Our algorithm has in total nine steps. The detail of each step is given algorithmically below with an accompanying description and diagrams. The main steps consist in first recovering the polynomial level set from the given GPTs, then then reconstructing the domain candidates and finally selecting one of the domain candidates in order to minimise the discrepancy between its GPTs and those of the true domain. Our algorithm goes far beyond the stability estimates established in the previous section. Here there is no

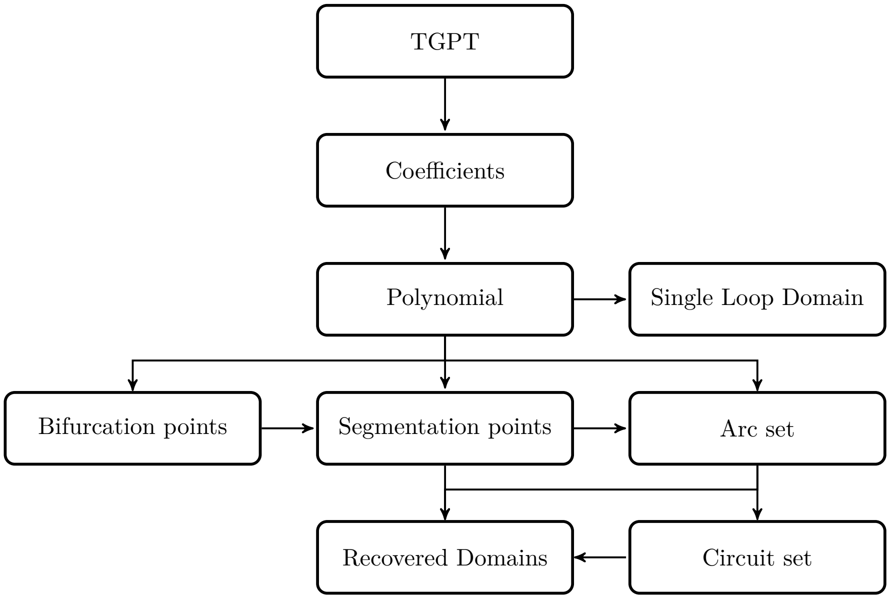

Figure 4.1. This diagram shows in broad terms the process we take to recover a domain from an associated TGPT.

need to assume that the curve to be recovered is smooth. Nevertheless, in order to reconstruct the domain candidates, several issues need to be carefully resolved. These include bifurcation points, segmentation points, and arc sets.

There are also tuning parameters scattered throughout the various processes and for the most part they are fixed. These tuning parameters should not distract from the otherwise straightforward process.

Algorithm 4.2 Check for a Loop

```
procedure checkLoop($P(x, y), t_{step}, tol_{po}, tol$)
    $\{[x_p, y_p] : p \in 1, .., N\} \leftarrow$ TraceLvlSet($P(x, y), [0, 0], [0, 0], t_{step}, tol_{po}$)
    $\overline{D} := \{[x_p, y_p] : p \in 1, .., N\}$
    if $\|[x_0, y_0] - [x_N, y_N]\| < tol$ then
        return $\overline{D}$
    else
        return $\overline{D} = \emptyset$
    end if
end procedure
```

Description: The purpose of this step is to confirm if the recovered polynomial level set is not already a smooth Jordan curve. If this is the case, the rest of the algorithm is unnecessary and inapplicable. To confirm, we trace out the level set using Algorithm 4.3 with the origin as an initial point and terminal points. Minor technicalities are involved in order to make sure that the procedure does not stop exactly where it begins.

Algorithm 4.3 Polynomial Level set trace

```
1:  procedure TraceLvlSet($P(x, y), p_0, T, dir = 1, t_{step}, tol_{po}$)
2:      $H(x, y) := [-\partial_y P(x, y), \partial_x P(x, y)]$
3:      $[x_0, y_0] = p_0$ ; $t_0 = 0$
4:      while End-condition = false do
5:          $[x_n, y_n] = dir \cdot H(x_{n-1}, y_{n-1})t_{n-1} + [x_{n-1}, y_{n-1}]$
6:          $t_n = t_{n-1} + t_{step}$
7:          if $\min_{\tau \in T} ||[x_n, y_n] - \tau|| < tol_{po}$ then
8:              End-condition = true
9:          end if
10:     end while
11:     $N := argmin_{n\in\mathbb{N}}||[X_n, Y_n] - \tau|| : \tau \in T$
12:     return $\{[X_n, Y_n]\}_{n\in\{0,1,...,N\}}$
13: end procedure
```

Description: The core notion of this procedure is the following two steps. Firstly define an equation of motion from the polynomial. Secondly use this equation to move along the level set starting from a known point on the level set. The equation of motion is given in line 5 and uses function $H(x, y)$ which is the gradient of $P(x, y)$ rotated by $\pi/2$. $H(x, y)$ is called the Hamiltonian and is tangent to the level set for points (x, y) on the level set. The tracing out is done by a Runga-Kutta algorithm. The stop condition is defined by a set T. The stop condition is hence that the traced level set reaches a specified proximity to a point in T. The set T can consist of a single or several points.

Algorithm 4.1 Recover Domain

```
procedure recDom($\mathbf{TGPT}_{2d,d}, \lambda$)
    $\overline{g} \leftarrow$Algorithm 6.2.($TGPT, \lambda$)
    $P(x, y) := \Sigma^n_{i,j} g_{i,j} x^i y^j \leftarrow \overline{g}$
    $D' \leftarrow$ checkLoop($P(x, y), t_{step}, tol_{po}, tol$)
    if $D' \neq \emptyset$ then
        return $D'$
    end if
    $B \leftarrow$ GetBifurcationPoints($P(x, y), a, b, tol_{bif}$)
    $S \leftarrow$ GetSegmentationPoints($P(x, y), B, r_{ini}, r_{step}, N$)
    $E \leftarrow$ FindArcs($P(x, y), S, Bound$)
    $C \leftarrow$ FindCircuits($E, B, S$)
    $\mathcal{D} \leftarrow$ ConstructDomains($P(x, y), S, C, t_{step}, tol$)
    $\overline{D} \leftarrow$ rankDomains($\mathcal{D}, \mathbf{TGPT}_{2,1}$)
    return $\overline{D}$
end procedure
```

Description: This is the wrapper that calls the individual procedures that constitute the algorithm. It is included as to see the sequence of steps. The assumed starting point of the algorithm is $\mathbf{TGPT}_{2d,d}$. The TGPT is obtained from [1, Algorithm 6.1]. We now go into the details of each step.

Algorithm 4.4 Bifurcation points

```
procedure GetBifurcationPoints($P(x,y), a, b, tol_{bif}$)
    $F(x,y) := [P(x,y), \partial_x P(x,y), \partial_y P(x,y)]$
    $B_{pre} := argmin_{(x,y)\in[a,b]^2} F(x,y)$
    $B \leftarrow$Cluster points in $B_{pre}$ that have distance $< tol_{bif}$
    return $B = \{[b_x^{(i)}, b_y^{(i)}]\}_{i\in M}$
end procedure
```

Description: The recovered polynomial level set consists of finitely many smooth arcs. These arcs meet at what is called bifurcation points. Bifurcation points are easily found by minimizing $P(x,y)$ and its derivatives. The order of derivatives dependent on the number of arcs meeting. For our purposes it was sufficient to only minimize the first. Two things to note, a, b specify a box within which there is searched and tol_{bif} is the threshold for the minimization. The code used would automatically increase tol_{bif} until at least two bifurcation points were found.

Algorithm 4.5 Segmentation points

```
procedure GetSegmentationPoints($P(x,y), B, r_{ini}, r_{step}, N$)
    for $b_i = [b_x^{(i)}, b_y^{(i)}] \ : \ i \in M$ do
        $\overline{s}_i = \{\}$
        $r = r_{ini}$
        while $|\overline{s}_i| < 4$ do
            $[x_j^{(i)}, y_j^{(i)}] = b_i + [r\cos(\theta_j) \ , \ r\sin(\theta_j)] \ \forall \ \theta_j := 2\pi\frac{j}{N}; \ j \in \{1, 2, ..., N\}$
            if $|\{j : P(x_j^{(i)}, y_j^{(i)}) = 0\}| > 3$ then
                $\overline{s}_i := \{[x_k^{(i)}, y_k^{(i)}] : P(x_k^{(i)}, y_k^{(i)}) = 0 \ , \ k \in M_i\}$
            else
                $r = r + r_{step}$
            end if
        end while
        $S \leftarrow \overline{s}_i$
    end for
    return $S = \{[x_k^{(i)}, y_k^{(i)}] : k \in M_i \ , \ i \in M\}$
end procedure
```

Description: As seen in Figure 4.3 the bifurcation points seldom lie on the level set. We now seek the nearest points on the level set to a fixed bifurcation point. These segmentation points define the end points of arcs in the level set. Note the double index notation, which is useful for defining arcs. The parameter N determines the fineness of the minimization and was fixed at 1000 and left at that.

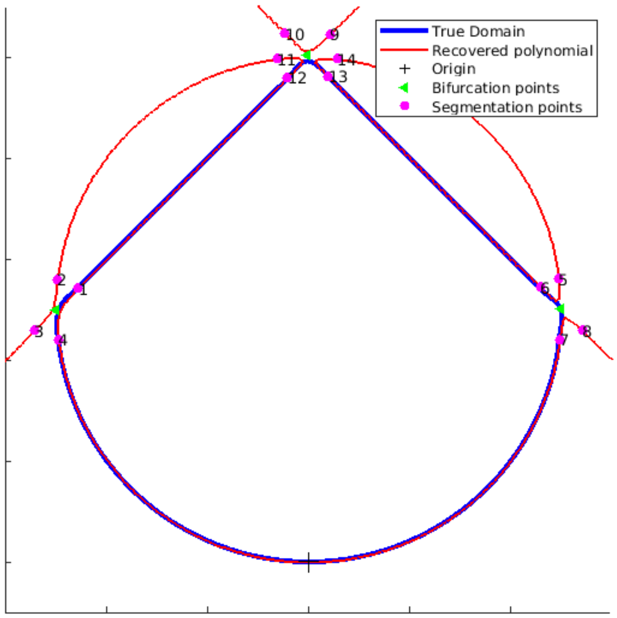

FIGURE 4.2. Recovered polynomial level set with bifurcation and segmentation points.

Algorithm 4.6 Arcs

```
procedure FINDARCS($P(x,y), S, Bound$)
    TrivArcs $:= \{(ik, il, 0) : k \neq l \wedge k, l \in M_i\ ,\ i \in M\}$
    for $[x_k^{(i)}, y_k^{(i)}] \in S$ do
        $\{[x_p, y_p] : p \in \{1, .., N\}\} =$TRACELVLSET$(P(x,y), s, S \setminus \{s\}, 1, t_{step})$
        $[x_l^{(j)}, y_l^{(j)}] =: \widehat{s} = argmin_{s' \in S \setminus \{s\}} ||[x_N, y_N] - s'||$
        if $e_i \in$ TrivArcs or $||[x_N, y_N]|| > Bound$ then
            Skip to next segmentation point.
        else
            $e_{ik} := (ik, jl, 1)$ Positive direction
            $e_{-ik} := (jl, ik, 2)$ Negative direction
            $E \leftarrow \{e_{ik}, e_{-ik}\}$
        end if
    end for
    $E \leftarrow$ TrivArcs
    return $E$
end procedure
```

Description: The task of this procedure is to find which pairs of segmentation points are connected through the level set. The connection is described as an ordered triple. The first two entries are the indices of the segmentation end points. The third entry is the direction of motion along the level set. Zero is used when the arc does not lie on the level set, see Figure 4.3. The parameter *Bound* here is just to ensure arcs do note race off to infinity.

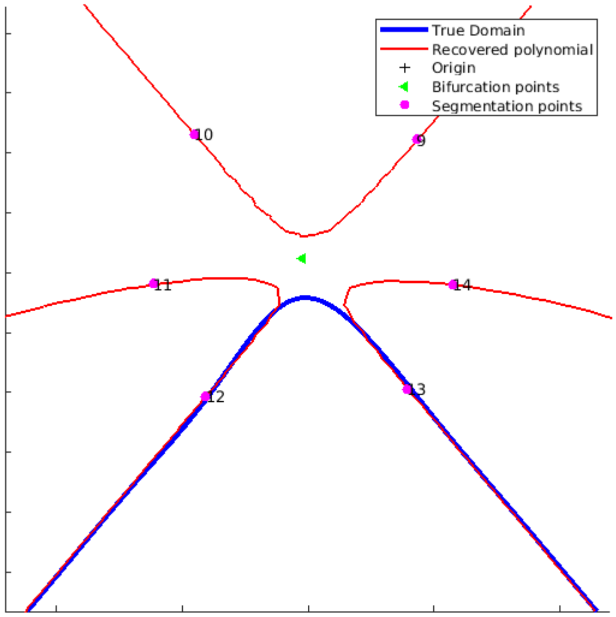

Figure 4.3. Close-up of the top bifurcation point.

Algorithm 4.7 Circuits

```
procedure FindCircuits($E, B, S$)
    $C' \leftarrow$ Algorithm from [2]
    for $c \in C'$ do
        Remove $c$ if $|c| < 4$
        Remove $c$ if $|c| > 2|B|$
        Remove $c$ if it does not containing the origin.
        Remove $c$ if it visits the same bifurcation point twice.
        Remove $c$ if it is a variation of a previous circuit.
    end for
    The result is $C \subset C'$
    return $C$
end procedure
```

Description: This procedure has the goal of making circuits from the previously obtained directed arcs. For clarity this procedure is represented in a more simplified way than the others. The first step is to use a well-known algorithm like the one in [2] in order to find all elementary circuits. Other algorithms are also viable as the arc set is quite small. These circuits represent domain candidates. To reduce the number of candidates, we incorporate some information on the domain. This information takes the form of constraints on the size, inclusion of the origin and internal bifurcation points.

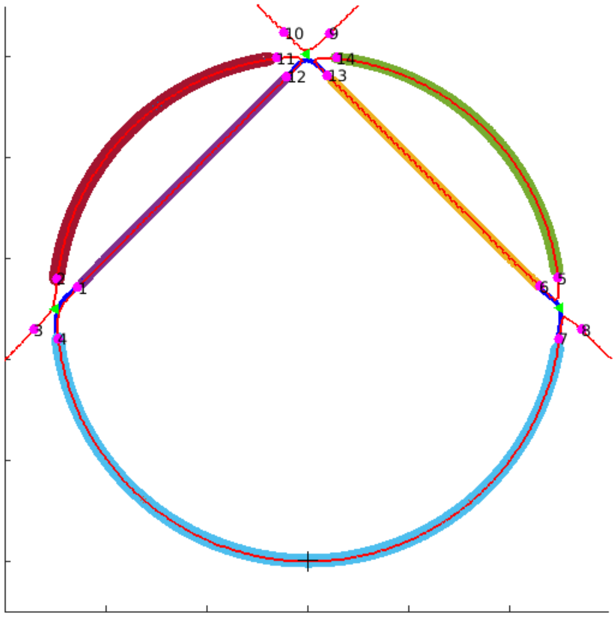

FIGURE 4.4. The various allowed arcs recovered from the level set, displayed each in a unique colour.

Algorithm 4.8 Constructed Candidate Domains

```
procedure ConstructDomains($P(x,y), S, C, t_{step}, tol$)
    for $c \in C$ do
        for $(ik, jl, dir \neq 0) := e \in c$ do
            $s := [x_k^{(i)}, y_k^{(i)}]$
            $s' := [x_l^{(j)}, y_l^{(j)}]$
            $\{[x_j^{(e)}, y_j^{(e)}] : j \in [N_e]\} = \textsc{TraceLvlSet}(P(x,y), s, s', dir, t_{step})$
        end for
        for $(ik, jl, dir = 0) := e \in c$ do
            $\{[x_j^{(e)}, y_j^{(e)}] : j \in [N_e]\} \leftarrow$ interpolate from other arcs.
        end for
        $D^{(c)} := \bigcup_{e \in c} \{[x_j^{(e)}, y_j^{(e)}] : j \in [N_e]\}$
    end for
    $\mathcal{D} := \{D^{(c)} \ : \ c \in C\}$
    return $\mathcal{D}$
end procedure
```

Description: This procedure is used to convert a circuit into a set of boundary points. The circuit can be thought of as a blue print for the domain candidate. This is because the circuit defines the sequence of arcs that constitute a domain. Hence the procedure traces out these arcs using the respective segmentation points as start and stop points. The gaps in between arcs obtained from tracing the level set are filled via interpolation from the existing arcs. The result is a set of domains in the form of a set of boundary points.

Algorithm 4.9 Rank Domains

```
procedure rankDomains($\mathcal{D}, \mathbf{TGPT}_{2,1}$)
    for $D^{(c)} \in \mathcal{D}$ do
        Export $D^{(c)}$ as $image_c.png$
        Read $image_c.png$ as a curve (see https://github.com/yanncalec/SIES)
        Compute $\mathbf{TGPT}^{(c)}_{2,1}$.
        $\overline{c} = argmin_c \frac{\|\mathbf{TGPT}^{(l)}_{2,c} - \mathbf{TGPT}_{2,c}\|}{\|\mathbf{TGPT}_{2,c}\|}$
        $\overline{D} \leftarrow D_{\overline{c}}$
    end for
    return $\overline{D}$
end procedure
```

Description: This final procedure is to discern which of the finite set of domain candidates most closely resembles the true domain. The resemblance is determined by the first order TGPT. The reason for the export step is that it sub-samples the domain candidate. Otherwise the recovered domain contains far too many points to be numerically stable.

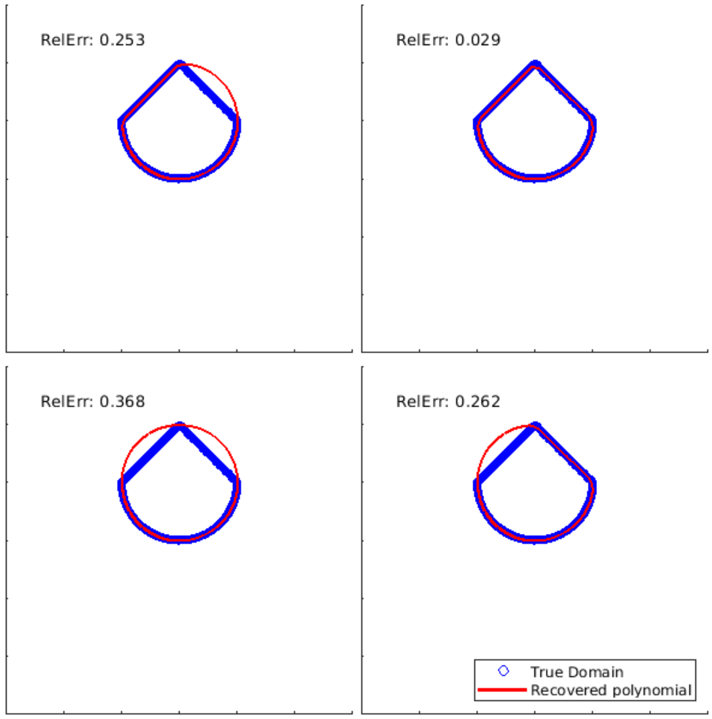

FIGURE 4.5. Recovered domains compared to the true domain.

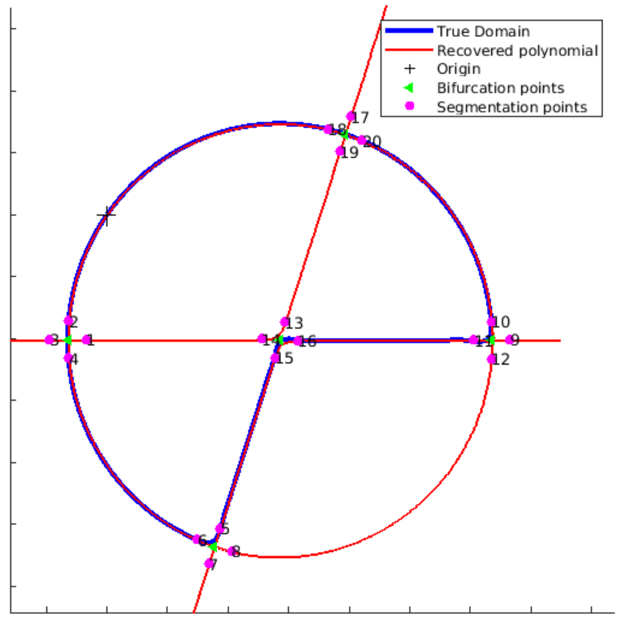

Figure 4.6. Figure of a disk with a sector missing.

4.2. **Examples.** In this section, we apply the algorithm described in the previous subsection to a few examples. We demonstrate its performance by means of a well chosen examples. We also show where the algorithm fails.

In the first example, Figure 4.7 present the possible seven domain candidates corresponding to a disk with a sector missing shown in Figure 4.6. The true domain is recovered by Algorithm 4.9. Here, it corresponds to the one with relative error 0.021.

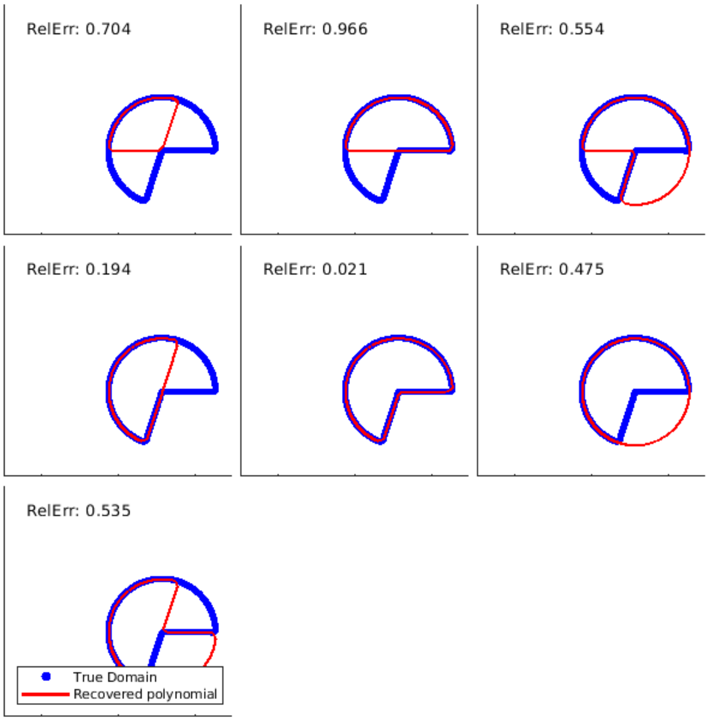

FIGURE 4.7. Figure of viable domain candidates.

In the second example, we consider domains with the same recovered level set. These are discerned from each other by using some boundary information and matching the associated TGPTs. Figures 4.9, 4.10, and 4.11 show three of the six distinct domains. We call these domains "conjoined circles", "crescent" and "intersection of circles" respectively to indicate the shape. All of these shape have the same level set namely two overlapping circles as seen in Figure 4.8. Among the candidates of the conjoined circles the best candidate was found to have relative error 0.01, see Figure 4.9. Among the candidates of the crescent the best candidate was found to have relative error 0.053, see Figure 4.10. And among the candidates of the intersection of circles shape the best candidate was found to have relative error 0.044, see Figures 4.11.

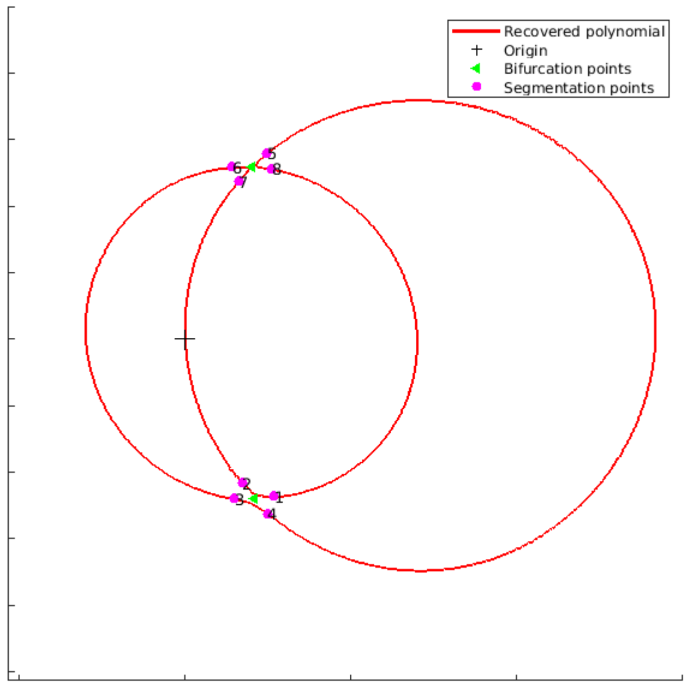

Figure 4.8. The level set of two overlapping circles gives rise to six distinct domains.

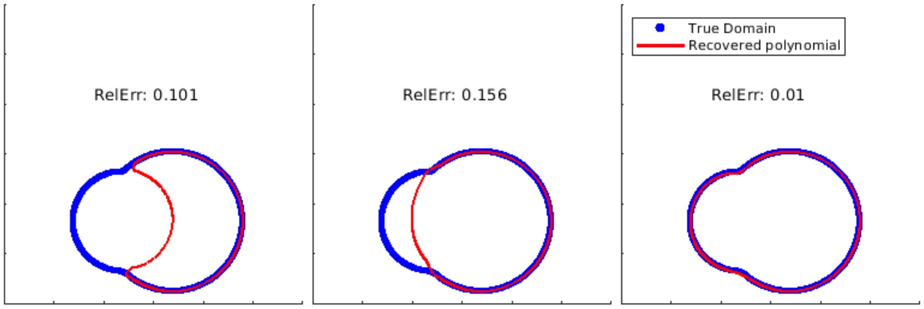

FIGURE 4.9. Conjoined circles.

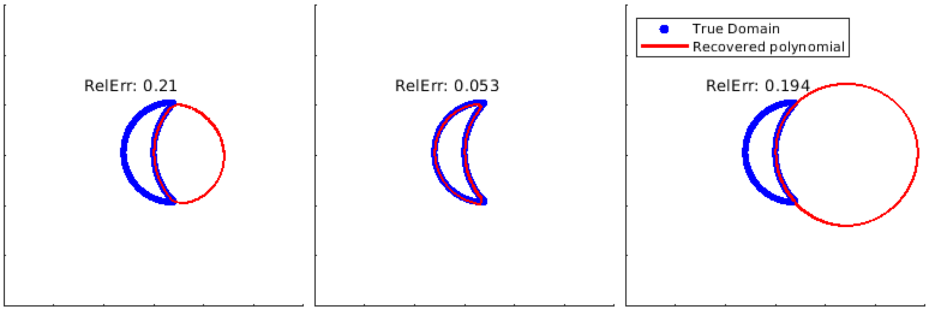

FIGURE 4.10. Crescent.

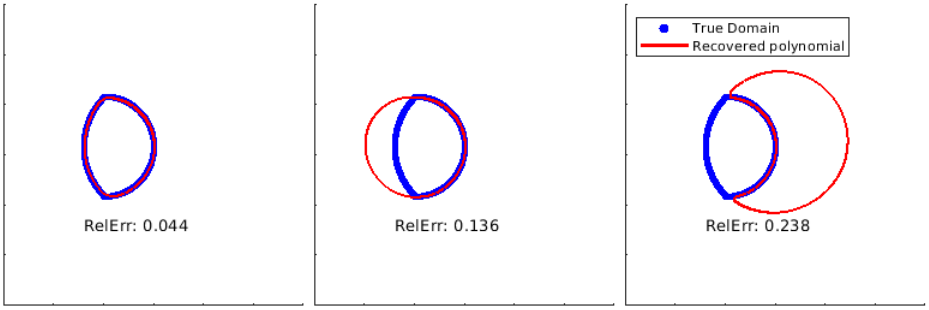

FIGURE 4.11. Intersection of circles.

In the third example, we present in Figure 4.12 a square with sinusoidal sides and its recovery from a single domain candidate.

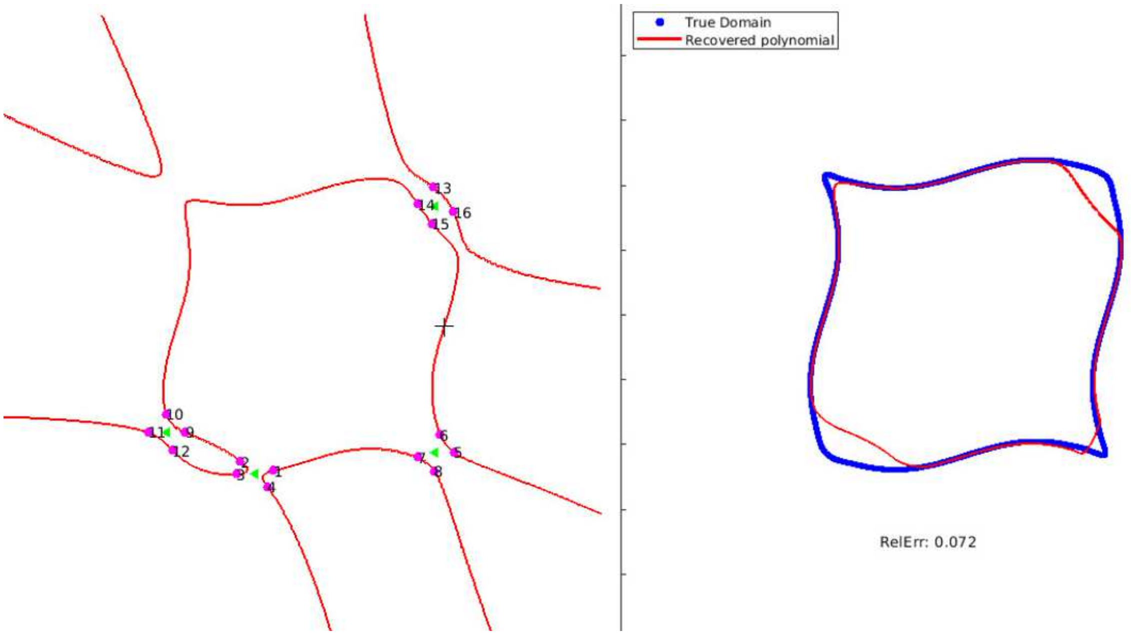

FIGURE 4.12. Domain recovery with a single candidate.

Finally, we show in Figure 4.13 that sometimes the recovered polynomial simply does not give the right domain. The true domain is in blue while the level set of the reconstructed polynomial from the GPTs is in red. This failure to recover the level set could stem from several reasons. The first reason is that higher degree domains are more unstable due to the higher powers taken in computing their GPTs. The second reason is that the proximity of the origin to a bifurcation point could cause instability. This however is still under investigation. We invite the reader to play around with the algorithm which is open source and available at https://github.com/JAndriesJ/ASPT.

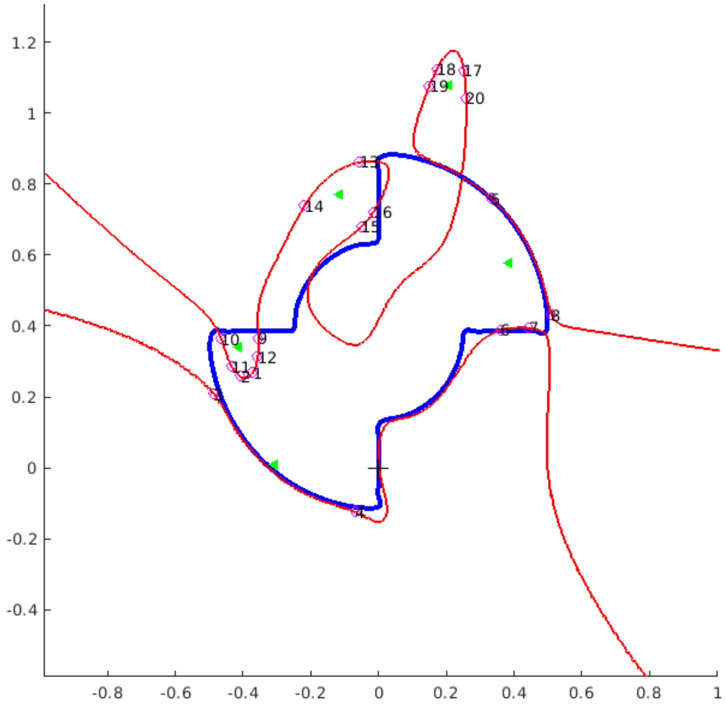

FIGURE 4.13. Failed polynomial recovery.

REFERENCES

[1] AMMARI, H., PUTINAR, M., STEENKAMP, A., TRIKI, F. *Identification of an algebraic domain in two dimensions from a finite number of its generalized polarization tensors. Math. Ann. (2018). https://doi.org/10.1007/s00208-018-1780-y* 2, 3, 5, 8, 10

[2] D.B. JOHNSON, *Finding all the elementary circuits of a directed graph*, SIAM J. Compt., 4 (1975), 77–84. 13

[3] H. AMMARI, T. BOULIER, J. GARNIER, W. JING, H. KANG AND H. WANG, *Target identification using dictionary matching of generalized polarization tensors,* Found. Comput. Math., 14 (2014), no. 1, 27–62. 2

[4] H. AMMARI, J. GARNIER, H. KANG, M. LIM, AND S. YU, *Generalized polarization tensors for shape description,* Numer. Math., 126 (2014), no. 2, 199–224. 2

[5] H. AMMARI, H. KANG, H. LEE, AND M. LIM, *Enhancement of near cloaking using generalized polarization tensors vanishing structures: Part I: The conductivity problem*, Comm. Math. Phys., 317 (2013), no. 1, 253–266. 2

[6] H. AMMARI AND H. KANG, *Properties of the generalized polarization tensors,* SIAM J. Multiscale Modeling and Simulation, 1 (2003), 335–348. 2

[7] H. AMMARI AND H. KANG, *Polarization and moment tensors with applications to inverse problems and effective medium theory*, Applied Mathematical Sciences, Vol. 162, Springer-Verlag, New York, 2007. 1, 2, 5

[8] H. AMMARI AND H. KANG, *Expansion methods,* Handbook of Mathematical Methods of Imaging, 447-499, Springer, 2011. 2

[9] H. AMMARI, H. KANG, M. LIM, AND H. ZRIBI, *The generalized polarization tensors for resolved imaging. Part I: Shape reconstruction of a conductivity inclusion,* Math. Comp., 81 (2012), 367–386. 2

[10] H. AMMARI, H. KANG, AND K. TOUIBI, *Boundary layer techniques for deriving the effective properties of composite materials*, Asymptot. Anal. 41 (2005), no. 2, 119–140. 2

[11] H. AMMARI, M. PUTINAR, M. RUIZ, S. YU, AND H. ZHANG, Shape reconstruction of nanoparticles from their associated plasmonic resonances, J. Math. Pures et Appl., 122 (2019), 23–48. 2

[12] H. Ammari, M. Ruiz, S. Yu, and H. Zhang, *Reconstructing fine details of small objects by using plasmonic spectroscopic data*, SIAM J. Imaging Sci. 11 (2018), no. 1, 1–23. 2

[13] H. Ammari, M. Ruiz, S. Yu, and H. Zhang, *Reconstructing fine details of small objects by using plasmonic spectroscopic data. Part II: The strong interaction regime*, SIAM J. Imaging Sci. 11 (2018), no. 3, 1931–1953. 2

[14] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*, Springer, Berlin, 1998. 3

[15] E. Bonnetier, C. Dapogny, and F. Triki, *Homogenization of the eigenvalues of the Neumann-Poincaré operator.* Preprint (2017). 2

[16] M. J. de la Puente *Real Plane Alegbraic Curves,* Expo. Mathematicae, 20 (2002), 291–314. 3

[17] D. Dubois and G. Efroymson, *Algebraic theory of real varieties*, in vol. *Studies and Essays presented to Yu-Why Chen on his 60-th birthday*, Taiwan University, 1970, pp. 107-135. 3

[18] E. Fabes, M. Sand, and J.K. Seo, *The spectral radius of the classical layer potentials on convex domains.* Partial differential equations with minimal smoothness and applications (Chicago, IL, 1990), 129–137, IMA Vol. Math. Appl., 42, Springer, New York, 1992. 5

[19] M. Fatemi, A. Amini, and M. Vetterli, *Sampling and reconstruction of shapes with algebraic boundaries,* IEEE Trans. Signal Process, 64 (2016), no. 22, 5807–5818. 2

[20] B. Gustafsson, C. He, P. Milanfar, and M. Putinar, *Reconstructing planar domains from their moments,* Inverse Problems, 16 (2000), 1053–1070. 2

[21] B. Gustafsson and M. Putinar, *Hyponormal Quantization of Planar Domains*, Lect. Notes Math. vol. 2199, Springer, Cham, 2017. 2

[22] M.-K. Hu, *Visual pattern recognition by moment invariants,* IRE Trans. Inform. Theory, 8 (1962), 179–187. 2

[23] J-B Lasserre, and M. Putinar, *Algebraic-exponential data recovery from moments,* Discrete & Computational Geometry, 54 (2015), 993–1012. 2, 3

[24] R. Nevanilinna, *Analytic functions*, Springer (1970) (Translated from German) 8

[25] G. Taubin, F. Cukierman, S. Sulliven, J. Ponce, and D.J. Kriegman, *Parametrized families of polynomials for bounded algebraic curve and surface fitting,* IEEE Trans. Pattern Anal. Mach. Intellig., 16 (1994), 287–303. 2

[26] F. Triki and M. Vauthrin, *Mathematical modeling of the Photoacoustic effect generated by the heating of metallic nanoparticles,* Quart. Appl. Math. 76 (2018), no. 4, 673–698. 2

Habib Ammari, Department of Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland
E-mail address: habib.ammari@math.ethz.ch

Mihai Putinar, Department of Mathematics, University of California at Santa Barbara, Santa Barbara, CA 93106-3080, USA, and School of Mathematics & Statistics, Newcastle University Newcastle upon Tyne, NE1 7RU, United Kingdom
E-mail address: mputinar@math.ucsb.edu

Andries Steenkamp, Department of Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland
E-mail address: andriess@student.ethz.ch

Faouzi Triki, Laboratoire Jean Kuntzmann, UMR CNRS 5224, Université Grenoble-Alpes, 700 Avenue Centrale, 38401 Saint-Martin-d'Hères, France
E-mail address: faouzi.triki@univ-grenoble-alpes.fr