

A Fast Local Level Set Method for Inverse Gravimetry

Victor Isakov¹, Shingyu Leung² and Jianliang Qian^{3,*}

¹ *Department of Mathematics and Statistics, Wichita State University, Wichita, Kansas, USA.*

² *Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR.*

³ *Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA.*

Received 10 July 2010; Accepted (in revised version) 2 December 2010

Available online 24 June 2011

Abstract. We propose a fast local level set method for the inverse problem of gravimetry. The theoretical foundation for our approach is based on the following uniqueness result: if an open set D is star-shaped or x_3 -convex with respect to its center of gravity, then its exterior potential uniquely determines the open set D . To achieve this purpose constructively, the first challenge is how to parametrize this open set D as its boundary may have a variety of possible shapes. To describe those different shapes we propose to use a level-set function to parametrize the unknown boundary of this open set. The second challenge is how to deal with the issue of partial data as gravimetric measurements are only made on a part of a given reference domain Ω . To overcome this difficulty, we propose a linear numerical continuation approach based on the single layer representation to find potentials on the boundary of some artificial domain containing the unknown set D . The third challenge is how to speed up the level set inversion process. Based on some features of the underlying inverse gravimetry problem such as the potential density being constant inside the unknown domain, we propose a novel numerical approach which is able to take advantage of these features so that the computational speed is accelerated by an order of magnitude. We carry out numerical experiments for both two- and three-dimensional cases to demonstrate the effectiveness of the new algorithm.

AMS subject classifications: 52B10, 65D18, 68U05, 68U07

Key words: Level set methods, inverse gravimetry, fast algorithms, numerical continuation.

*Corresponding author. *Email addresses:* victor.isakov@wichita.edu (V. Isakov), masyleung@ust.hk (S. Leung), qian@math.msu.edu (J. Qian)

1 Introduction

Let Ω be a domain in \mathcal{R}^n with connected $\mathcal{R}^n \setminus \bar{\Omega}$ and let $U(\cdot; \mu)$ be the potential of a measure μ with respect to the kernel for the Laplacian operator. The inverse problem of potential theory is formulated as follows [8]: find a measure μ with support contained in Ω from its potential $U(\cdot; \mu)$ in $\mathcal{R}^n \setminus \Omega$. As stated, the solution to this problem is notoriously non unique. In gravimetry it is reasonable to assume that the measure μ is a volume mass distribution with density f on an open set D (an open bounded subset of Ω); moreover, in many realistic situations f is a known constant. Then, under some geometrical assumptions on the open set D one can claim its uniqueness for the inverse problem.

As known, the inverse gravimetry problem is severely ill-posed [8], and many regularization techniques have been proposed to obtain a conditionally well-posed problem so as to solve this important inverse problem numerically. Most recently, the inverse gravimetry problem with an unknown density f have been tackled in [1] by using the total-variation regularization. Although the numerical reconstruction in [1] is successful for some examples, the proposed methods in [1] may run into difficulties due to the presence of many local minima of the to-be-minimized non convex functional. We propose to tackle the inverse gravimetry problem with density $f = 1$ on an unknown open set D by using the level set method. The theoretical foundation for our approach is the uniqueness result presented in [8]: namely, if D is star-shaped with respect to its center of gravity or x_3 -convex, then the exterior potential uniquely determines D . To achieve our purpose, the first challenge is how to represent D . We propose to consider ∂D as the zero-level set of a level-set function to be found. The level set method [13] is a powerful approach for interface or shape-optimization problems, which can take care of interface merging and topological changes automatically. The second challenge is how to deal with the issue of partial data as the potential measurement is only made on a part of the whole boundary of the computational domain. To overcome this difficulty, inspired by ideas in [6] we propose a numerical continuation approach to obtain fictitious measurements of the potential due to the unknown domain. The third challenge is how to speed up the level set inversion process. Based on the features of the underlying inverse gravimetry problem, we propose a novel approach to speed up the computational process by an order of magnitude, so that we can carry out numerical experiments for both two- and three-dimensional cases.

1.1 Related work

For inverse (obstacle) problems the level set method has been first used by Santosa [16]. Later on, there are many efforts to analyze this method and extend this beautiful idea to a variety of inverse problems; see [2, 3, 7, 11, 19] and references therein.

The closest to our work is the recent interesting paper [18] in which a method based on multiple level sets has been proposed for piecewise constant volume density recon-

struction in highly ill-posed inverse potential problems. In [18] the authors assumed (zero) Dirichlet boundary condition on $\partial\Omega$, while ours does not make such an assumption. Our formulation seems to be more suitable for geophysics. Although in [18] they tested partial surface data, they did not use the numerical continuation approach to obtain complete fictitious measurements on the boundary of a certain domain enclosing the unknown target. In addition, they did not use the local level set method as we are using in this work. Thus, our approach is different from theirs.

The rest of the paper is organized as follows. In Section 2, we introduce the inverse problem of gravimetry and level set methods. In Section 3, we propose the numerical method, including a fast local level-set implementation of the proposed approach. In Section 4, we present the algorithm for carrying out numerical continuation based on partial measurements on the surface. In Section 5 we give a variety of examples to demonstrate the performance of the proposed method, including two- and three-dimensional cases.

2 Inverse problem of gravimetry and level set methods

2.1 Inverse gravimetry

The gravity force generated by mass distribution μ in a domain $\Omega \subset \mathcal{R}^n$ is

$$\nabla_x u(x; \mu) = \int_{\Omega} \nabla_x K(x, y) d\mu(y), \quad (2.1)$$

where

$$K(x, y) = \frac{1}{4\pi|x-y|} \quad \text{when } n=3, \quad \text{and} \quad -\frac{1}{2\pi} \ln|x-y| \quad \text{when } n=2.$$

We will assume that μ is zero outside \bar{D} , where D is some bounded open set, $\bar{D} \subset \Omega$, and Ω is a given open set. In most geophysical applications, Ω is a half-space or a ball. Typical mass distributions are 1) $\mu = f dm$ (volume mass distribution of density $f \in L_1$, dm is the Lebesgue measure), 2) $\mu = g d\Gamma$ (single layer distribution of density $g \in L_1(\Gamma)$ over a Lipschitz surface Γ), and 3) $\mu = m_1 \delta(-x(1)) + \dots + m_k \delta(-x(k))$ (point masses m_j at points $x(j)$).

The inverse problem of gravimetry is to find μ given $\nabla u(x; \mu)$ on $\Gamma_0 \subset \partial\Omega$. We will assume that μ is a volume distribution, in particular, $\mu = \chi_D dy$, which is the most geophysically realistic assumption. Then we can state the inverse problem of gravimetry as follows: given the gravity data g on Γ_0 , solve the equation

$$\nabla u(\cdot; \chi_D)|_{\Gamma_0} = \mathbf{g}, \quad (2.2)$$

to find D , where the data $\mathbf{g} \in L^2(\Gamma_0)$ is given on a part of the boundary: $\Gamma_0 \subset \partial\Omega$.

We have the following lemma from [8] in terms of continuation.

Lemma 2.1. [8] *Let Ω be a convex domain with analytic (regular) boundary, and Γ_0 be a non empty hyper surface contained in $\partial\Omega_0$. If measures μ_j are non negative and $|\nabla U_1| = |\nabla U_2|$ on Γ_0 , then $U_1 = U_2$ on $\mathcal{R}^n \setminus \Omega$. Here $U_j = U(\cdot; \mu_j)$, $j = 1, 2$.*

A domain Ω in \mathcal{R}^n is said to be star-shaped with respect to a point a is $a \in \Omega$ and the intersection of any ray originated at a with Ω is an interval. An open set Ω is said to be convex in x_1 if the intersection of any straight line parallel to the x_1 -axis with Ω is an interval [8]. We have the following uniqueness theorem from [8].

Theorem 2.1. [8] *Suppose that either 1) D_1 and D_2 are star-shaped with respect to their centers of gravity or 2) D_1 and D_2 are convex in x_3 . If $U(\cdot; \chi_{D_1}) = U(\cdot; \chi_{D_2})$ on $\mathcal{R}^n \setminus \Omega$, then $D_1 = D_2$.*

To make use of the uniqueness theorem, we first assume that the measurement is made on the whole boundary, $\Gamma_0 = \partial\Omega$. Later on, based on the continuation lemma, we will use numerical continuation to continue the partial measurement towards the unknown domain so that we can construct a set of gravity data on the whole boundary of a certain artificial domain enclosing the unknown D .

2.2 The level set method

To parameterize the unknown domain D , we will introduce a level set function ϕ^* which is Lipschitz continuous satisfying

$$\phi^*(x) > 0, \quad \text{for } x \in D, \tag{2.3a}$$

$$\phi^*(x) = 0, \quad \text{for } x \in \partial D, \tag{2.3b}$$

$$\phi^*(x) < 0, \quad \text{for } x \in \bar{D}^c. \tag{2.3c}$$

Then we can define the following operator according to the gravity force relation:

$$A(\phi^*(x)) = \nabla_x u(x; \chi_D)|_{\Gamma_0}, \tag{2.4a}$$

$$\nabla_x u(x; \chi_D) = \int_D \nabla_x K(x, y) dy = \int_{\Omega} \nabla_x K(x, y) H(\phi^*(y)) dy, \tag{2.4b}$$

where H is the Heaviside function.

Consequently, the inverse problem is stated as finding D represented by ϕ^* such that $A(\phi^*) = \mathbf{g}$. Furthermore, we will solve the following minimization problem to find ϕ^* and thus D :

$$\min F(\phi) = \min \|A(\phi) - \mathbf{g}\|_{L^2(\Gamma_0)}^2. \tag{2.5}$$

To that end we compute the Frechet derivative of the functional $F(\phi)$ by using a variational approach [19]:

$$F(\phi + h) - F(\phi) = \left\langle \frac{\partial F}{\partial \phi}, h \right\rangle + \mathcal{O}(h^2). \tag{2.6}$$

To simplify the notation, we introduce

$$G(x; \phi(\cdot)) = \int_{\Omega_z} \nabla_x K(x, z) H(\phi(z)) dz - \mathbf{g}(x). \quad (2.7)$$

Then we have

$$\begin{aligned} F(\phi+h) - F(\phi) &= \int_{\Gamma_0} G^T(x; (\phi+h)(\cdot)) G(x; (\phi+h)(\cdot)) dx - F(\phi) \\ &= \int_{\Gamma_0} 2G^T(x; \phi) \left(\int_{\Omega_y} \nabla_x K(x, y) H'(\phi(y)) h(y) dy \right) dx + \mathcal{O}(h^2) \\ &= \int_{\Omega_y} \left(\int_{\Gamma_0} 2G^T(x; \phi) \nabla_x K(x, y) dx \right) H'(\phi(y)) h(y) dy + \mathcal{O}(h^2). \end{aligned} \quad (2.8)$$

Hence the Frechet derivative of the functional F is:

$$\frac{\partial F}{\partial \phi} = \int_{\Gamma_0} 2G^T(x; \phi) \nabla_x K(x, y) dx \delta(\phi(y)). \quad (2.9)$$

The necessary condition for ϕ to be a minimizer is that

$$0 = \frac{\partial F}{\partial \phi} = \int_{\Gamma_0} 2G(x; \phi)^T \nabla_x K(x, y) dx \delta(\phi(y)), \quad (2.10a)$$

$$0 = \frac{1}{|\nabla \phi|} \frac{\partial \phi}{\partial n}, \quad \text{on } \partial \Omega, \quad (2.10b)$$

where we have imposed the natural boundary condition for ϕ on the boundary Ω . If we choose $h = -\frac{\partial F}{\partial \phi}$, which is the gradient descent direction, then the functional is decreasing along the negative gradient direction. Thus we will evolve the following gradient flow to the steady state to compute the minimizer:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial F}{\partial \phi}, \quad (2.11a)$$

$$\frac{1}{|\nabla \phi|} \frac{\partial \phi}{\partial n} = 0, \quad \text{on } \Gamma_0, \quad (2.11b)$$

where $\phi = \phi(x, t)$ with t being the artificial evolution time. We will take $\phi^*(x) = \phi(x, \infty)$ and $\partial D = \{x: \phi^*(x) = 0\}$ which is the zero level set of ϕ^* .

3 Numerical methods

3.1 The algorithm

In this section, we first summarize the algorithm and then describe the numerical procedures in solving the level set evolution.

Algorithm:

1. Initialization. Initialize the level set function ϕ .
2. Compute the mismatch $G(x)$ along the boundary Γ_0 according to (2.7).
3. Compute the level set derivative of the energy according to (2.10).
4. Evolution. Evolve the level set function according to the gradient flow (2.11).
5. Reinitialization. Reinitialize the level set function to maintain the signed distance property.
6. Repeat 2-5 until it converges.

The first component is to compute the mismatch $G(x;\phi)$ between the measurements $\mathbf{g}(x)$ and the numerical solution, given by (2.7) along the boundary Γ_0 . The Heaviside function $H(\phi)$ in the integrand is numerically approximated by the following ϵ -Heaviside function $H_\epsilon(\phi)$,

$$H_\epsilon(\phi) = \begin{cases} 0, & \phi < -\epsilon, \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right), & -\epsilon \leq \phi \leq \epsilon, \\ 1, & \epsilon < \phi. \end{cases} \quad (3.1)$$

The integral is then computed by applying the simple midpoint quadrature rule on an underlying uniform Cartesian mesh. Numerically, assuming that there are N mesh points in each direction, the computational complexity of determining the mismatch $G(x;\phi)$ for all $x \in \partial\Omega$ is $\mathcal{O}(N^3)$ for a 2-D case and $\mathcal{O}(N^5)$ for a 3-D case.

Once we have the mismatch along the boundary, we back-propagate it into the computational domain Ω by evaluating the convolution (2.10). This essentially provides the normal velocity v_n to the level sets. With N the number of grid points in each direction, we have the computational complexity of this part given by $\mathcal{O}(N^3)$ for a 2-D case and $\mathcal{O}(N^5)$ for a 3-D case. To speed up the algorithm, we can replace this full level set implementation by the local level set method [14]. Since we only care about the location of the zero level set, we modify the level set function only in a neighborhood of the zero level set. To update this computational tube and maintain the signed distance property of the level set function, we also solve the following reinitialization equation for several τ steps [12, 15],

$$\tilde{\phi}_\tau + S_\epsilon(\phi)(|\nabla\tilde{\phi}| - 1) = 0 \quad (3.2)$$

with the initial condition

$$\tilde{\phi}(\tau=0) = \phi, \quad (3.3)$$

where S_ϵ is the ϵ -signum function given by $S_\epsilon(\phi) = 2H_\epsilon(\phi) - 1$.

Both the gradient flow (2.11) and the reinitialization equation are Hamilton-Jacobi equations. We apply the TVD RK3 [17] in the time direction and the WENO5 [9] for the spatial derivative. This gives a high-order accurate solution to the time evolution. The CFL condition for solving the Hamilton-Jacobi equation is given by $dt < dx / \max(v_n) =$

$\mathcal{O}(1/N)$. We may also use other fast methods to solve the above Hamilton-Jacobi equations [4, 5, 10]

3.2 Fast implementation for computing the mismatch term (2.7)

As discussed in the last subsection, one of the most computationally expensive step in the algorithm is to evaluate the integral in equation (2.7) for any $\mathbf{x} \in \Gamma_0$. We propose an $\mathcal{O}(N^{2(n-1)})$ algorithm for computing this integral by taking advantage of the fact that the integrand is in fact zero for any $\mathbf{z} \in \Omega \setminus D$.

3.2.1 Two-dimensional cases

For simplicity, we first consider the following two dimensional case where the kernel $K(x, z) = -\ln|x - z|/(2\pi)$. Changing the variable from the Cartesian coordinates to the polar coordinates centered at the point $\mathbf{x} \in \Gamma_0$, we have

$$\int_{\Omega_z} \nabla_x K(x, z) H(\phi(z)) dz = \frac{1}{2\pi} \int_{\{(r, \theta): H(\phi) \geq 0\}} \frac{r \cdot (\cos \theta, \sin \theta)}{r^2} r dr d\theta \quad (3.4a)$$

$$= \frac{1}{2\pi} \int_{\{(r, \theta): H(\phi) \geq 0\}} (\cos \theta, \sin \theta) dr d\theta. \quad (3.4b)$$

To numerically evaluate this integral, we will need two ingredients. The first one is an efficient Cartesian-to-polar mapping of the level set function in (3.4a) which determines the domain $\{(r, \theta): H(\phi) \geq 0\}$ for the integration. The second one is an efficient quadrature to sum up the integral (3.4b) in this polar coordinate system.

An efficient quadrature is readily available for the integral (3.4b) if for each θ_k , we have an explicit representation of the set $\{(r, \theta_k): H(\phi(r, \theta_k)) \geq 0\}$, i.e. the interval(s) $[r_{k, \min}^i, r_{k, \max}^i]$ ($i = 1, \dots, i_k$), in which the level set function ϕ is positive. Then the computational complexity of this integral will be $\mathcal{O}(N)$ for each \mathbf{x} if we discretize the θ -domain using N grid points.

We propose the following strategy to determine the Cartesian-to-polar mapping which will give $[r_{k, \min}^i, r_{k, \max}^i]$ ($i = 1, \dots, i_k$) for each θ_k . The first step is a pre-processing step to convert the implicit level set representation into an explicit representation. We represent the boundary of the target by connected segments $[p_m, p_{m+1}]$ for $m = 0, 1, \dots, M = \mathcal{O}(N)$ with $p_0 = p_{M+1}$; this yields a polygon approximation to the boundary of the target.

An easy way to determine such explicit representation is to use the MATLAB function `contourc`. For the most trivial implementation without using this implementation, one can search all $\mathcal{O}(N^2)$ cells and then determine this explicit representation. Even though the computational complexity of this trivial implementation is $\mathcal{O}(N^2)$, this pre-processing step is only done once and will not increase the computational complexity of the overall process.

Next, given $\mathbf{x} \in \Gamma_0$, we determine for each segment $[p_m, p_{m+1}]$ if it intersects with any of the polar axis: $\theta = \theta_k$ ($k = 1, \dots, N$). This can be done by checking if $\lfloor \theta(p_m) / \Delta\theta \rfloor =$

$\lfloor \theta(p_{m+1}) / \Delta\theta \rfloor$, where $\lfloor \cdot \rfloor$ is the integer part of the argument, $\theta(p_m) = \text{Arg}(p_m - \mathbf{x})$ is the polar angle of the point p_m represented in the polar coordinate system centered at the point \mathbf{x} , and $\Delta\theta = 2\pi / (N - 1)$ is the mesh size in the θ -direction. If those two integers are different, then it implies that the segment intersects with a certain polar axis $\theta = \theta_k$, and we determine the location where they intersect; this yields an approximation to $r_{k,\min}^i$ or $r_{k,\max}^i$ ($i = 1, \dots, i_k$), for $k = \lfloor \theta(p_{m+1}) / \Delta\theta \rfloor$.

Therefore the whole algorithm to compute the integral in (2.7) for all $\mathbf{x} \in \Gamma_0$ consists of carrying out a pre-processing step ($\mathcal{O}(N^2)$ operations), determining intervals for integration (N points on the boundary Γ_0 and $\mathcal{O}(N)$ operations each), and carrying out a trapezoidal summation (N points on the boundary Γ_0 and $\mathcal{O}(N)$ operations each). This yields an $\mathcal{O}(N^2)$ algorithm for computing (2.7).

3.2.2 Three-dimensional cases

For the three dimensional case, we apply the spherical coordinates (ρ, ψ, θ) with the origin at \mathbf{x} , and the integral in (2.7) becomes

$$\int_{\Omega} \nabla_x K(x, z) H(\phi(z)) dz = \frac{1}{4\pi} \int_{\{H(\phi) \geq 0\}} (\sin^2 \psi \cos \theta, \sin^2 \psi \sin \theta, \sin \psi \cos \phi) d\rho d\psi d\theta. \quad (3.5)$$

We can easily generalize the above two dimensional Cartesian-to-polar mapping to the three dimension. The pre-processing step is to obtain an explicit representation of the interface by patches of triangles. This surface triangulation can be done by using the command `isosurface` in MATLAB. Other packages can also be used and the computational complexity of this step can be at worst of $\mathcal{O}(N^4)$ without altering the total computational complexity of computing the integral everywhere on Γ_0 , where N is the number of grid points in each dimension. The output from this pre-processing step is $\mathcal{O}(N^2)$ triangles with vertices \mathbf{v}_i which approximate the zero level set of ϕ .

For each point \mathbf{x} on the boundary Γ_0 , we first construct a new coordinate system to simplify our later computations. We translate the Cartesian coordinate system to be centered at the point \mathbf{x} ; then we construct the Householder reflector,

$$H = I - 2 \frac{\tilde{\mathbf{v}} \tilde{\mathbf{v}}^T}{\tilde{\mathbf{v}}^T \tilde{\mathbf{v}}}, \quad (3.6)$$

where $\tilde{\mathbf{v}} = \|\tilde{\mathbf{v}}\| e_2 - \tilde{\mathbf{v}}$ and $\tilde{\mathbf{v}} = \sum_i \mathbf{v}_i / \sum_i 1$ is the centroid of the vertices from the triangulation. With this reflector, we obtain a Cartesian coordinate system so that \mathbf{x} is now at the origin and the \tilde{y} -axis, the new y -axis, points at the centroid of the vertices.

The next step in computing the integral (3.5) is to discretize the $\theta - \psi$ space and then to determine $r_{i,j,\min}$ and $r_{i,j,\max}$ for each $\theta = \theta_i$ and $\psi = \psi_j$. First we apply the spherical coordinates and project each of the above triangles onto the $\theta - \psi$ space. Now if the observation point $\mathbf{x} \in \Gamma_0$ lies outside the zero level set, the translation and reflection we proposed above will guarantee that all projected triangles will give $\theta \in (0, \pi)$ and $\psi \in (0, \pi)$ rather than $\theta \in [0, 2\pi)$ in the original spherical coordinate system. This simplifies our later

implementation since we need not worry about the 2π periodicity in the representation using the spherical coordinates. The projection of a triangulated surface in \mathbb{R}^3 onto the $\theta-\psi$ space is still a triangle. If this projected triangle contains any of $(\theta_{i'}, \psi_{j'})$, we then determine the distance between \mathbf{x} and the triangulated surface in the direction $(\theta_{i'}, \psi_{j'})$. This would give an approximation to $r_{i',j',\min}$ or $r_{i',j',\max}$ such that $\phi(\theta_{i'}, \psi_{j'}, r) = 0$. For each observation point \mathbf{x} on the boundary Γ_0 , the computational complexity to determine all $r_{i,j}$ is therefore of $\mathcal{O}(N^2)$, the same order of the number of triangulated surfaces. Once we sweep through all triangles, we can approximate the integral (2.7) evaluated at $\mathbf{x} \in \Gamma_0$ with an extra $\mathcal{O}(N^2)$ operations using the Trapezoidal rule in the $\theta-\psi$ space. Therefore, the overall computational complexity of evaluating (2.7) everywhere on the boundary is $\mathcal{O}(N^4)$, which is one order of magnitude lower than the full implementation in Section 3.1.

4 Partial measurements and numerical continuation

In the above discussions, we have assumed that the measurements are given everywhere on the boundary of the domain. However, the derivation carries through directly to the situation where measurements are made only on a portion of $\Gamma_0 = \partial\Omega$. Suppose that we have only partial data on γ along the boundary Γ_0 ; see marked points in Fig. 1. Nevertheless, the uniqueness theorem guarantees uniqueness only when we have the full measurement on the whole boundary of $\Gamma_0 = \partial\Omega$. To overcome this difficulty, we use Lemma 2.1 to carry out an analytical continuation numerically.

Referring to Fig. 1, we first represent the partial measurement due to the unknown domain as an equivalent single-layer potential due to an unknown surface density on the

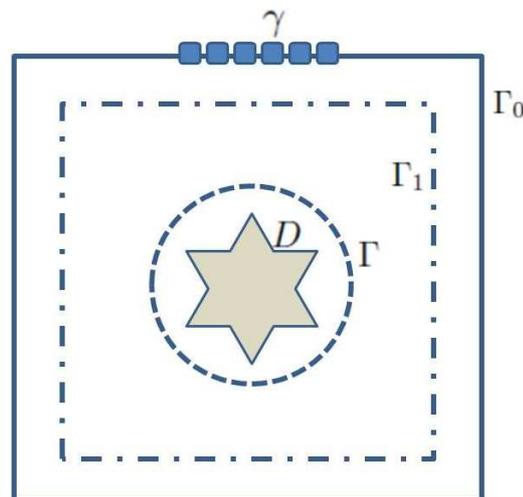


Figure 1: Illustration of numerical continuation.

surface Γ ; then we try to reconstruct the surface density on Γ from the partial measurement on $\gamma \subset \Gamma_0 = \partial\Omega$; finally we use the reconstructed surface density on Γ to obtain the “fictitious” full measurement on Γ_1 due to the unknown domain. Then the inversion will be based on the data on Γ_1 . Such a numerical continuation has been used in [6].

According to the single layer potential theory for the Laplacian operator, we have for any x outside the domain enclosed by Γ ,

$$u(x) = \int_{\Gamma} f(y)K(x,y)d\Gamma, \quad (4.1)$$

where $f(y)$ is the potential density to be determined due to the unknown target. To determine $f(y)$ we use the measurement data on the top boundary for $x \in \gamma \subset \Gamma_0$. However, this will be an extremely ill-posed problem because it is a Fredholm integral equation of the first kind. We use the Singular Value Decomposition (SVD) to solve this equation to obtain f at some discrete points. Once we have f at those discrete points, we can use the expression to compute u on Γ_1 outside the domain enclosed by Γ .

Numerically, we discretize Eq. (4.1) using

$$\nabla_x u(x_i) = h \sum_j f_j \nabla_x K(x_i, y_j). \quad (4.2)$$

Here $x_i \in \gamma$ are locations where we have measurements, $y_j \in \Gamma$, and the mesh size $h = \Delta x = \Delta y$. This implies that the system of linear equations $\mathcal{K}\mathbf{f} = \mathbf{u}$, where the matrix \mathcal{K} is under-determined. We apply SVD to compute the least-squares solution to this system. We decompose $\mathcal{K} = USV^T$, where S is a diagonal matrix, and U and V are unitary. To regularize the computations, we construct the pseudo-inverse for the diagonal matrix S to obtain $T = [t_{i,j}]$ by

$$t_{i,i} = \begin{cases} 1/s_{i,j}, & \text{if } s_{i,j} > \epsilon_{SVD}, \\ 0, & \text{otherwise.} \end{cases}$$

Then the least-squares solution is obtained by $\mathbf{f} = VTU^T\mathbf{u}$. Once we have f_j , we can determine $u(x_k)$ for $x_k \in \Gamma_1$ and use these data as measurements for the level set algorithm as designed above.

5 Numerical examples

Unless otherwise specified, we have discretized the two-dimensional computational domain $\Omega_2 = [0,1]^2$ using 65 mesh points in each direction and the three-dimensional domain $\Omega_3 = [0,1]^3$ using 33^3 mesh points. The initial zero level set is a circle centered at $\mathbf{x}_c = (x_c, y_c) = (0.5, 0.5)$ with radius $r = 0.45$ for two dimensional cases and is a sphere centered at $\mathbf{x}_c = (x_c, y_c, z_c) = (0.5, 0.5, 0.5)$ with the same radius $r = 0.45$ for the three-dimensional cases. The corresponding level set function can be easily assigned at each grid location by

$$\phi(\mathbf{x}) = r - \|\mathbf{x} - \mathbf{x}_c\|_2. \quad (5.1)$$

Such an initial guess is based on a rough estimate where the target is located so that the initial zero level set encloses the target; otherwise, the level set method may fail. As long as the initial zero level set encloses the target, the level set method is not sensitive to the specific location of the zero level set.

5.1 Two-dimensional cases

5.1.1 Ellipsoidal target

In this computation test, we use the following benchmark example in which the gravity force of D is given by a simple formula. Let D be the ellipse $a_1^{-2}x_1^2 + a_2^{-2}x_2^2 < 1$, where $0 < a_2 \leq a_1$ are to be chosen. As shown in [8], page 99,

$$\partial u(x; \chi_D) = -\frac{a_1 a_2}{2(x + \sqrt{x^2 - e^2})} = -\frac{a_1 a_2}{2e^2}(x - \sqrt{x^2 - e^2}), x \in \mathbf{R}^2 \setminus D, \quad (5.2)$$

where we complexify x as $x_1 + ix_2$ so that $\partial = \frac{1}{2}(\partial_1 - i\partial_2)$, and $e = \sqrt{a_1^2 - a_2^2}$. Thus we identify \mathbf{R}^2 with \mathbf{C} , and use multiplication in \mathbf{C} .

We use the branch $\sqrt{z^2 - e^2}$ of the many-valued function on $\mathbf{C} \setminus [-e, e]$ fixed by the requirement $\sqrt{e^2} = e$. The real-valued form of this formula would be

$$\nabla u(x; \chi_D) = -\frac{a_1 a_2}{e^2} \left(x_1 - y_1, -x_2 + \frac{x_1 x_2}{y_1} \right), \quad (5.3)$$

where y_1, y_2 are (signed) solutions to the equation $\sqrt{x^2 - e^2} = y_1 + iy_2$ so that $\text{sign}(x_1) = \text{sign}(y_1)$, or

$$x_1^2 - x_2^2 - e^2 = y_1^2 - y_2^2, \quad x_1 x_2 = y_1 y_2.$$

By using the branch condition $\sqrt{z^2} = z$, we have

$$y_1 = \text{sign}(x_1) \sqrt{\frac{1}{2} \left(x_1^2 - x_2^2 - e^2 + \sqrt{(x_1^2 - x_2^2 - e^2)^2 + 4x_1^2 x_2^2} \right)}. \quad (5.4)$$

In the following, we take $a_1 = 0.3$ and $a_2 = 0.15$.

First, we assume that data collection is carried out along the whole boundary of the computational domain; namely, we have a complete set of data measurement along the whole boundary. To test robustness and stability of the level set algorithm, we will use measurement with different levels of noise: 0% noise (clean measurement), 10% noise and 20% noise.

Fig. 2 shows the results in the case of measurement with 0% noise, where the complete, clean measurement is given by the analytical expression (5.3). The zero level set at various stages of evolution are shown in Fig. 2, where the elliptic target is plotted in red centered at $(x_1, x_2) = (0.5, 0.5)$ and the initial level set is drawn in blue. As we can see, the steady state solution to the level set evolution matches with the target very well.

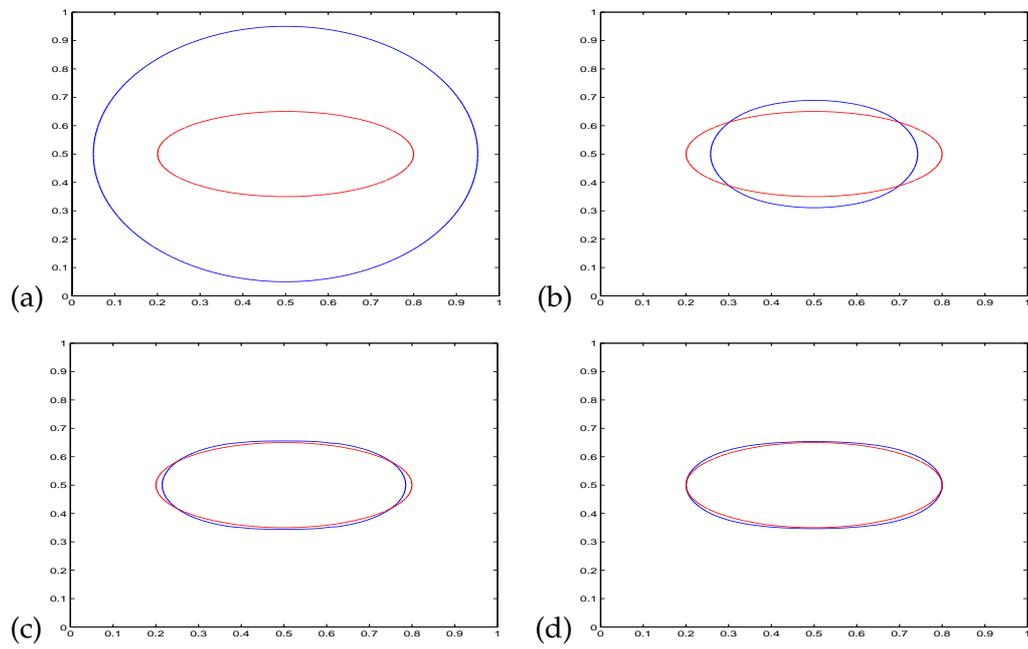


Figure 2: (Ellipsoidal Target) Complete clean measurements.

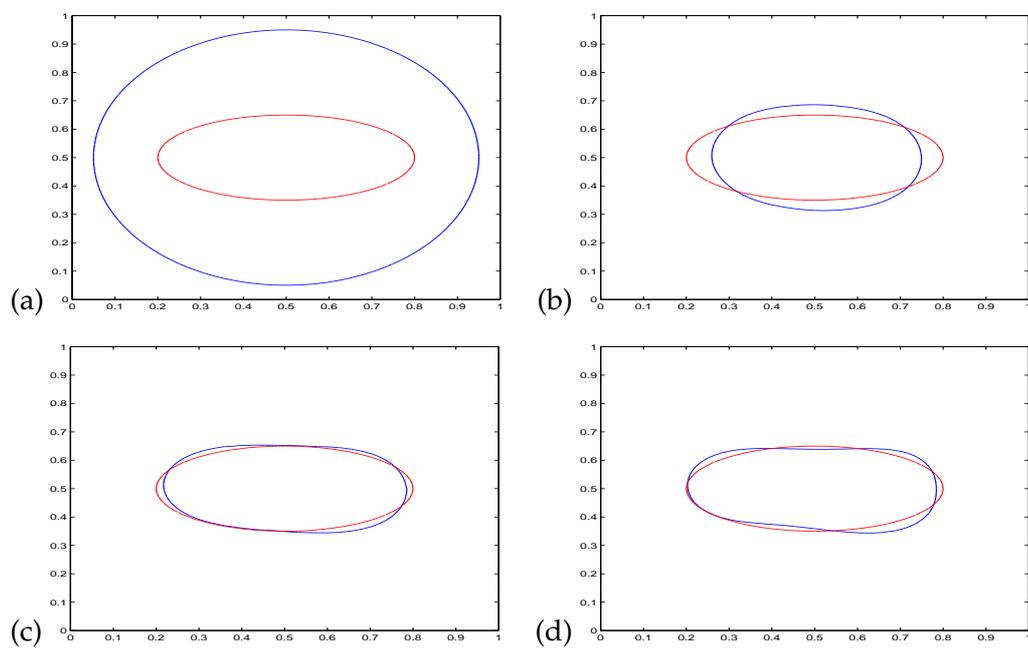


Figure 3: (Ellipsoidal Target) Complete measurements with 10% noise.

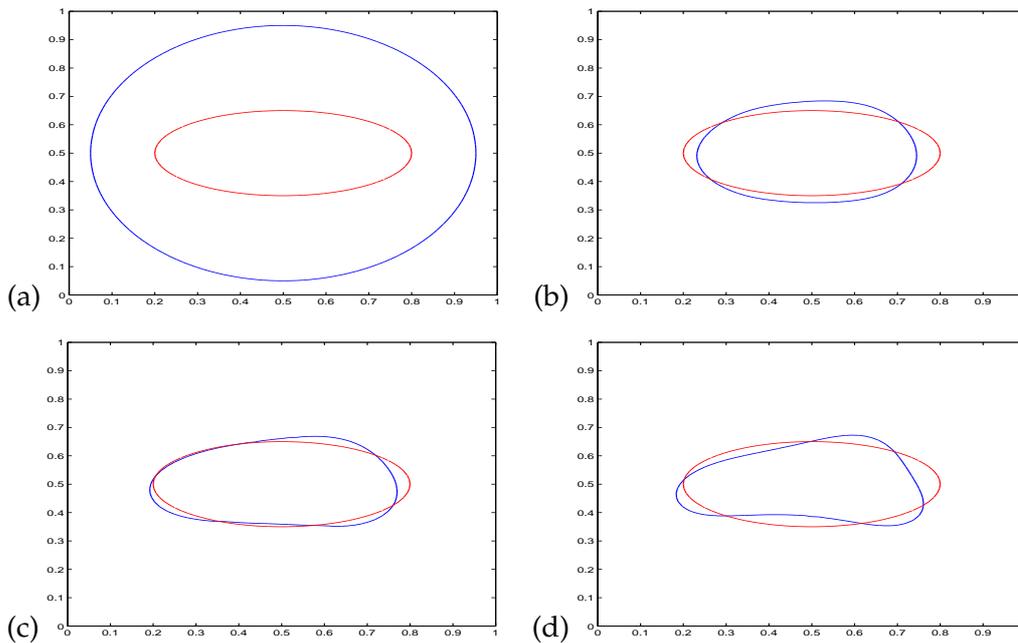


Figure 4: (Ellipsoidal Target) Complete measurements with 20% noise.

Fig. 3 shows the results in the case of measurement with 10% noise, where the complete, clean measurement is given by the analytical expression (5.3). The zero level set at various stages of evolution are shown in Fig. 3, where the elliptic target is plotted in red centered at $(x_1, x_2) = (0.5, 0.5)$ and the initial level set is drawn in blue. As we can see, the steady state solution to the level set evolution matches with the target reasonably well.

Fig. 4 shows the results in the case of measurement with 20% noise, where the complete, clean measurement is given by the analytical expression (5.3). The zero level set at various stages of evolution are shown in Fig. 4, where the elliptic target is plotted in red centered at $(x_1, x_2) = (0.5, 0.5)$ and the initial level set is drawn in blue. As we can see, the steady state solution to the level set evolution matches with the target with reasonably accuracy, given the high-level noise.

Next we assume that data collection is only carried out along a part of the boundary; namely, we only have partial measurement along the boundary. We consider two possible strategies to use this partial measurement. One strategy is to use this partial measurement directly in the inversion process, as our level set formulation allows us to do so with slight change in terms of implementation. The other strategy is to carry out a numerical continuation by using the partial measurement as shown in Section 4 so as to obtain a complete set of data along the whole boundary of a fictitious domain which encloses the unknown domain.

We apply both strategies to two sets of partial measurements: 1) the measurement is made on $\{(x, y) : x \in [0.25, 0.75], y = 1\}$; 2) the measurement is made on $\{(x, y) : x \in$

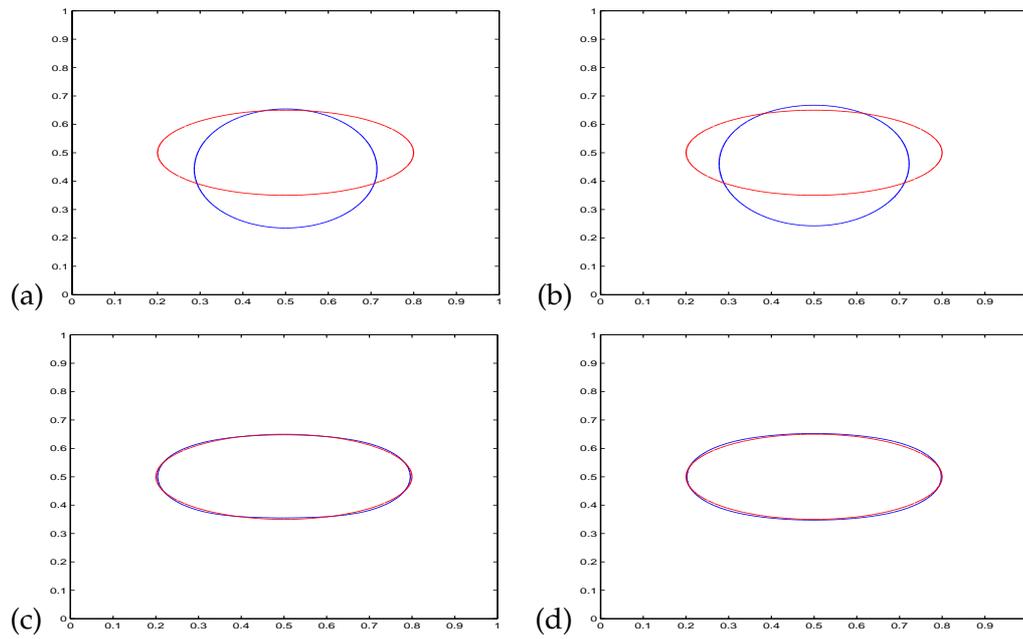


Figure 5: (Ellipsoidal Target) (a) The solution using measurements directly on $\{(x,y) : x \in [0.25,0.75], y = 1\}$. (b) The solution using measurements directly on $x \in [0,1]$ and $y = 1$. (c) The solution using measurements from numerical continuation using data on $\{(x,y) : x \in [0.25,0.75], y = 1\}$. (d) The solution using measurements from numerical continuation using data on $\{(x,y) : x \in [0,1], y = 1\}$.

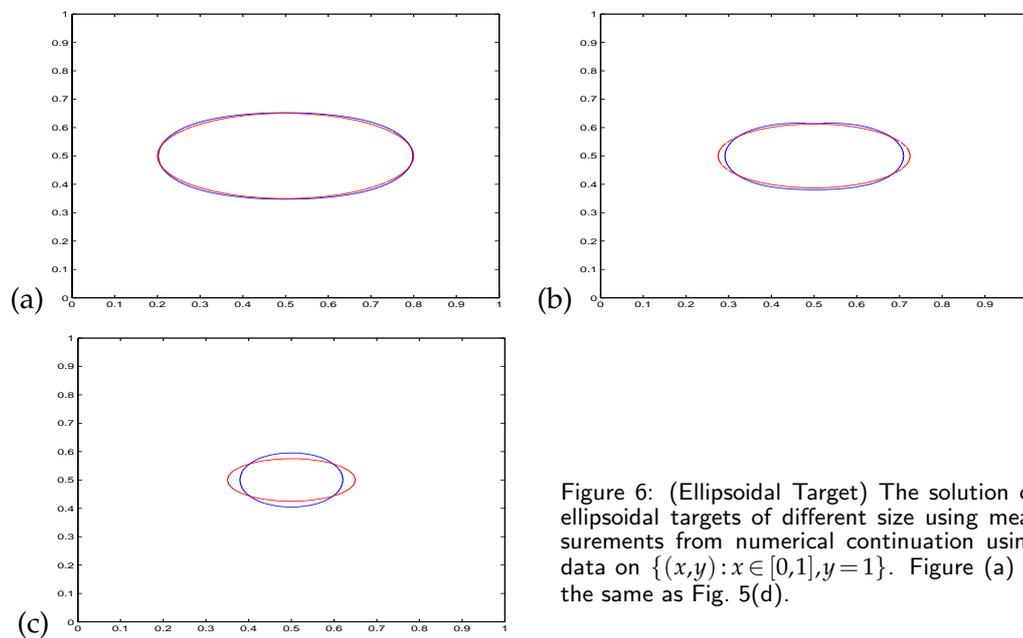


Figure 6: (Ellipsoidal Target) The solution of ellipsoidal targets of different size using measurements from numerical continuation using data on $\{(x,y) : x \in [0,1], y = 1\}$. Figure (a) is the same as Fig. 5(d).

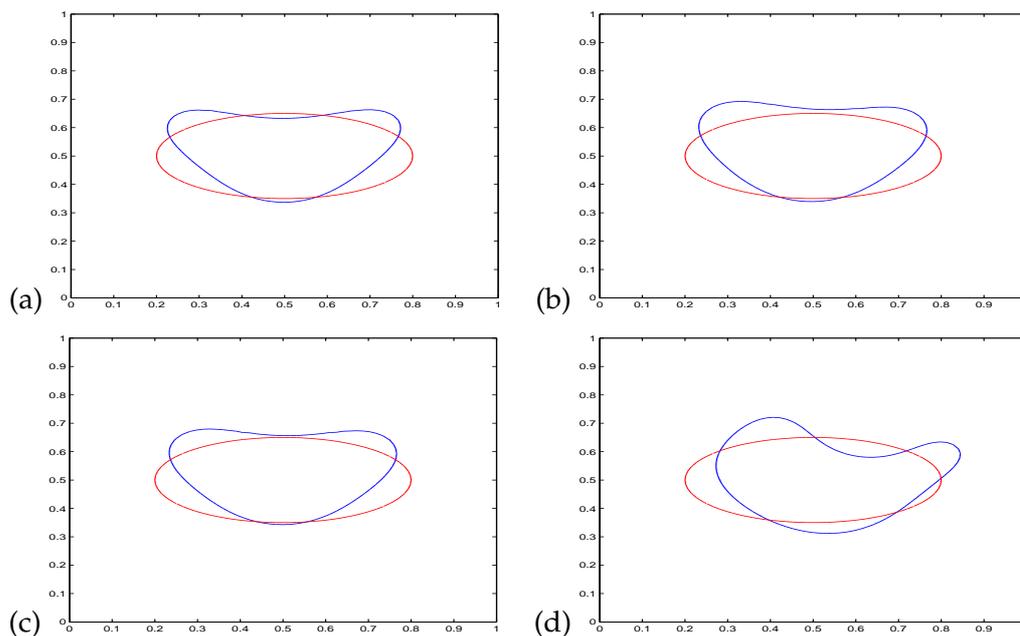


Figure 7: (Ellipsoidal Target) (a) The solution using measurements with 1% noise from numerical continuation using data on $\{(x,y):x \in [0,1],y=1\}$. (b) The solution using measurements with 5% noise from numerical continuation using data on $\{(x,y):x \in [0,1],y=1\}$. (c) The solution using measurements with 10% noise from numerical continuation using data on $\{(x,y):x \in [0,1],y=1\}$. (d) The solution using measurements with 20% noise from numerical continuation using data on $\{(x,y):x \in [0,1],y=1\}$.

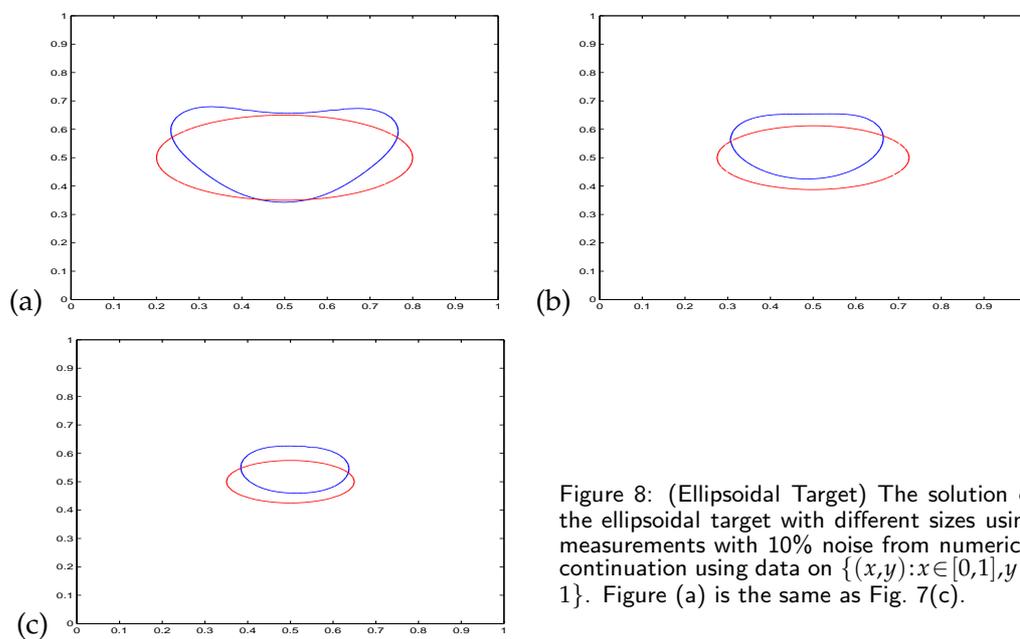


Figure 8: (Ellipsoidal Target) The solution of the ellipsoidal target with different sizes using measurements with 10% noise from numerical continuation using data on $\{(x,y):x \in [0,1],y=1\}$. Figure (a) is the same as Fig. 7(c).

$[0,1], y=1\}$. In the numerical continuation, the threshold ϵ_{SVD} in the SVD regularization is $10^{-10} \max_i \sigma_i$, where σ_i are the singular values of the matrix \mathcal{K} ; the interface Γ used in numerical continuation as shown in Fig. 1 is chosen to be $\sqrt{(x-0.5)^2+(y-0.5)^2}=0.375$, which is further discretized with $4(N-1)$ points (N the number of grid points of the underlying Cartesian mesh in each direction), and the interface Γ_1 in Fig. 1 is chosen to be $\max(|x-0.5|, |y-0.5|)=0.4375$. Fig. 5 shows the corresponding results. As we can see, the strategy based on numerical continuation makes significant differences in the inversion results in comparison to the strategy based on directly using the partial measurement.

To further test the effectiveness of numerical continuation and the resolution of the level set formulation, we decrease the size of the unknown target in the setting of the partial measurement on the boundary where $\{(x,y):x \in [0,1], y=1\}$. Fig. 6 shows the results, where the computational setup in terms of numerical continuation is the same as in Fig. 5; namely, the interface Γ as shown in Fig. 1 is chosen to be $\sqrt{(x-0.5)^2+(y-0.5)^2}=0.375$, which is further discretized with $4(N-1)$ points (N the number of grid points of the underlying Cartesian mesh in each direction), and the interface Γ_1 in Fig. 1 is chosen to be $\max(|x-0.5|, |y-0.5|)=0.4375$; the threshold ϵ_{SVD} in the SVD regularization is $10^{-10} \max_i \sigma_i$, where σ_i are the singular values of the matrix \mathcal{K} . As we can see, the resolution degrades as the target becomes smaller.

We also consider partial measurement with different levels of noise using the strategy of numerical continuation. The partial measurement is made on the boundary where $\{(x,y):x \in [0,1], y=1\}$, and we add 1%, 5%, 10% and 20% noise to the partial measurement. Afterwards, we apply the numerical continuation procedure to the polluted data, and the setup for numerical continuation is the same as in Fig. 5; namely, the interface Γ is chosen to be $\sqrt{(x-0.5)^2+(y-0.5)^2}=0.375$, the interface Γ_1 is chosen to be $\max(|x-0.5|, |y-0.5|)=0.4375$, and the threshold ϵ_{SVD} in the SVD regularization is $10^{-1} \max_i \sigma_i$, where σ_i are the singular values of the matrix \mathcal{K} . Fig. 7 shows the results. As we can see, increasing level of noise does degrade the resolution, but we are still able to capture the target with reasonable accuracy.

To further test our algorithm based on numerical continuation, we also decrease the size of the unknown target in the setting of polluted data. The partial measurement is made on the boundary where $\{(x,y):x \in [0,1], y=1\}$, and we add 10% noise to the partial measurement. Afterwards, we apply the numerical continuation procedure to the polluted data. Fig. 8 shows the results, where the computational setup in terms of numerical continuation is almost the same as in Fig. 5, but the threshold ϵ_{SVD} in the SVD regularization is $10^{-1} \max_i \sigma_i$, where σ_i are the singular values of the matrix \mathcal{K} . As we can see, in this case the resolution degrades as the target becomes smaller.

5.1.2 Disjoined targets

A very nice property of the level set method is the ability to naturally model the topological change in any interface evolution. By using disjoined targets, we will demonstrate the advantage of formulating the inverse problem using the level set formulation. The unknown domain consists of two disks centered at $(0.5 \pm 0.25, 0.5)$. To obtain full or par-

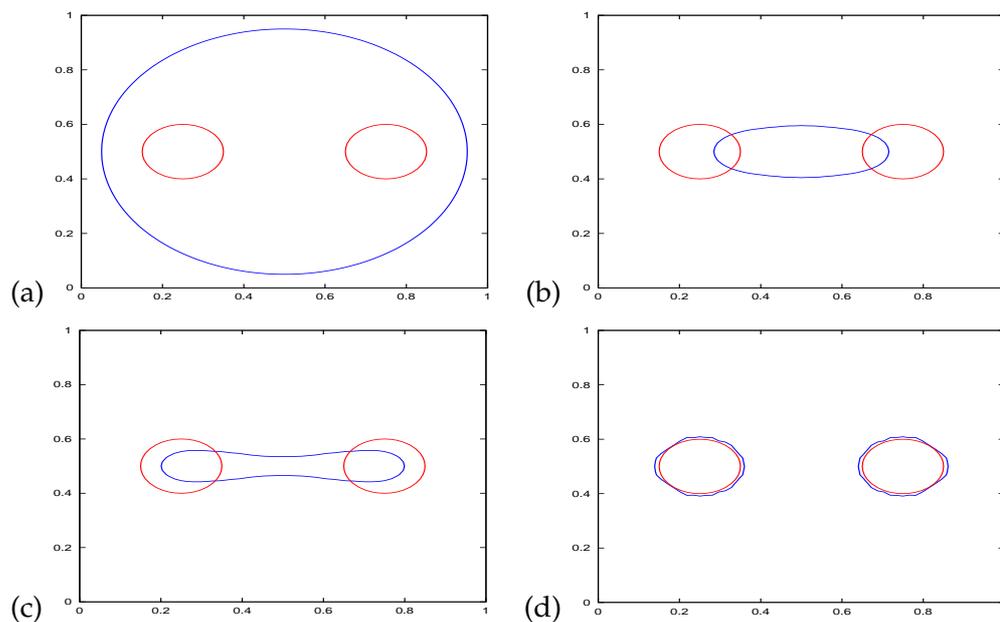


Figure 9: (Two disks) Complete clean measurements.

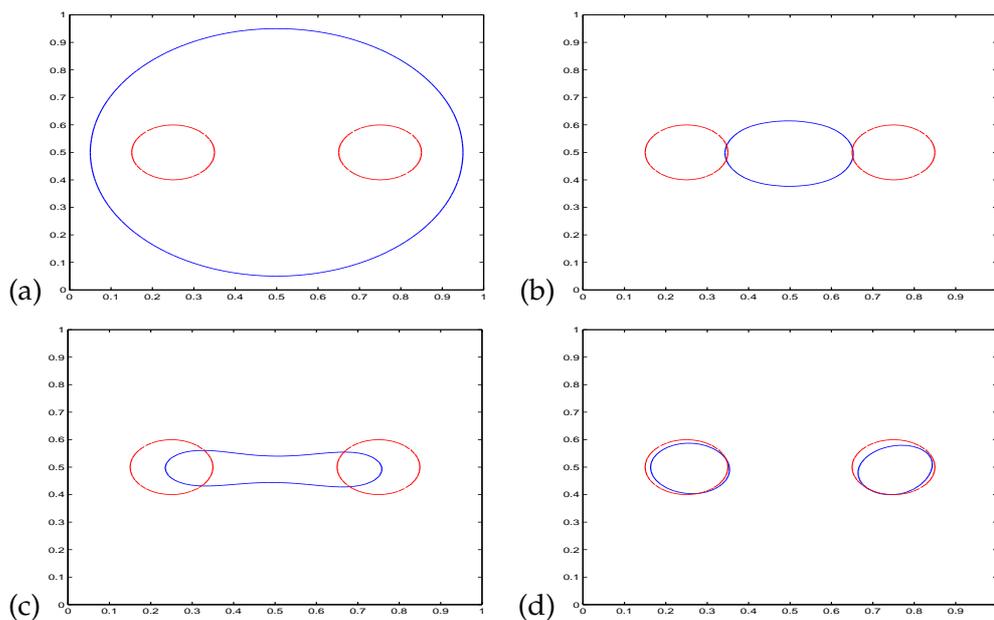


Figure 10: (Two disks) Complete measurements with 10% noise.

tial measurement $\mathbf{g}(x)$, we will use formula (2.4b) with the exact ϕ^* such that $H(\phi^*)$ is the characteristic function of the two disks.

First, we assume that data collection is carried out along the whole boundary of the

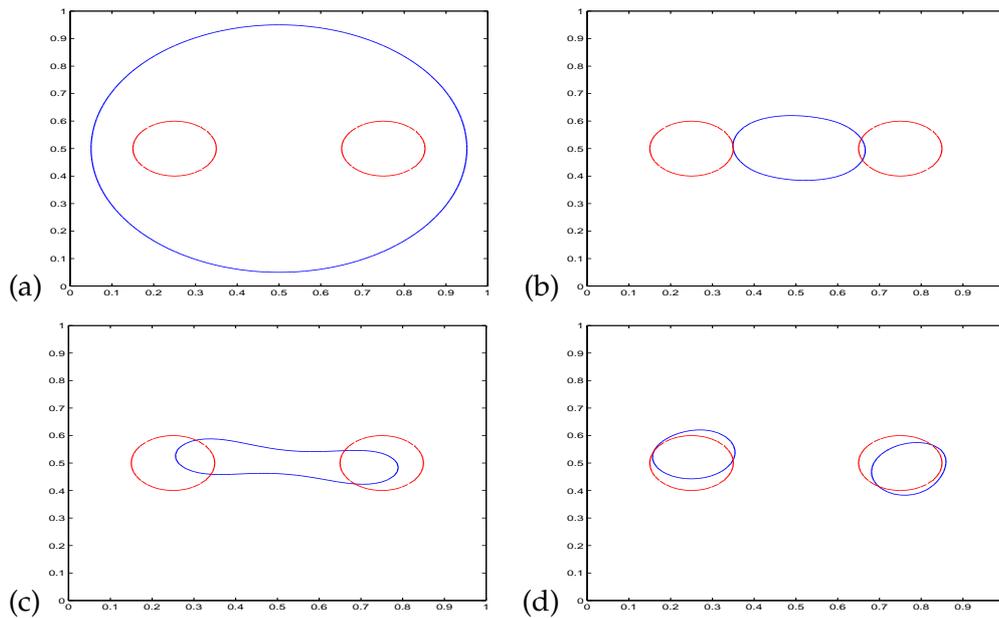


Figure 11: (Two disks) Complete measurements with 20% noise.

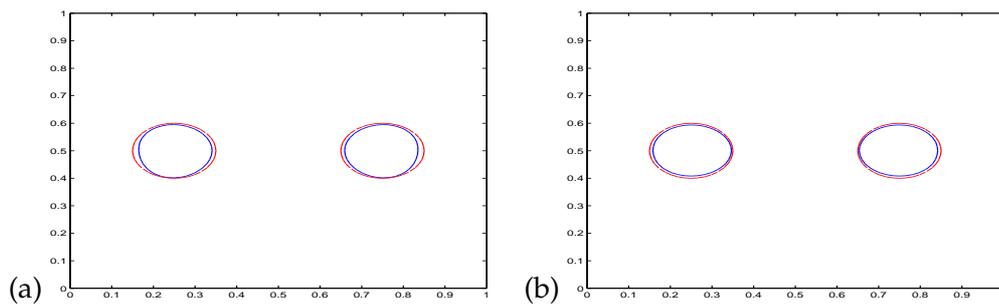


Figure 12: (Two disks) (a) The solution using measurements from numerical continuation using data on $\{(x,y): x \in [0.25, 0.75], y=1\}$. (b) The solution using measurements from numerical continuation using data on $\{(x,y): x \in [0, 1], y=1\}$.

computational domain; namely, we have a complete set of data measurement along the whole boundary. To test robustness and stability of the level set algorithm, we will use measurement with different levels of noise: 0% noise (clean measurement), 10% noise and 20% noise.

Fig. 9 shows the results in the case of measurement with 0% noise. The zero level set at various stages of evolution are shown in Fig. 9, where the two-disk target is plotted in red and the initial level set is drawn in blue. As we can see, the steady state solution to the level set evolution matches with the target very well, and the initial one single zero-level curve is split into two zero-level curves as time evolves.

Fig. 10 shows the results in the case of measurement with 10% noise. The zero level set

at various stages of evolution are shown in Fig. 10, where the two-disk target is plotted in red and the initial level set is drawn in blue. As we can see, the steady state solution to the level set evolution matches with the target reasonably well, and the initial one single zero-level curve is split into two zero-level curves as time evolves.

Fig. 11 shows the results in the case of measurement with 20% noise. The zero level set at various stages of evolution are shown in Fig. 11, where the two-disk target is plotted in red and the initial level set is drawn in blue. As we can see, the steady state solution to the level set evolution matches with the target well, and the initial one single zero-level curve is split into two zero-level curves as time evolves.

Next we assume that data collection is only carried out along a part of the boundary; namely, we only have partial measurement along the boundary. Then we carry out numerical continuation by using the partial measurement as shown in Section 4 so as to obtain a complete set of data along the whole boundary of a fictitious domain which encloses the unknown domain. We apply this strategy to two sets of partial measurements: 1) the measurement is made on $\{(x,y) : x \in [0.25,0.75], y = 1\}$; 2) the measurement is made on $\{(x,y) : x \in [0,1], y = 1\}$. In numerical continuation, the threshold ϵ_{SVD} in the SVD regularization is $10^{-10} \max_i \sigma_i$, where σ_i are the singular values of the matrix \mathcal{K} ; the interface Γ used in numerical continuation as shown in Fig. 1 is chosen to be $\sqrt{(x-0.5)^2 + (y-0.5)^2} = 0.375$, which is further discretized with $4(N-1)$ points (N the number of grid points of the underlying Cartesian mesh in each direction), and the interface Γ_1 in Fig. 1 is chosen to be $\max(|x-0.5|, |y-0.5|) = 0.4375$. Fig. 12 shows the corresponding results. As we can see, the strategy based on numerical continuation yields inversion results which matches with the target disks very well.

5.1.3 A star-shaped target

Next we consider a star-shaped unknown domain. Fig. 13 shows the solutions with complete measurements, and partial measurements on the boundary of $\{(x,y) : x \in [0.25,0.75], y = 1\}$, respectively. As we can see from Fig. 13, the algorithm yields reasonable solutions near the region where the measurements are made.

5.2 Three-dimensional cases

The level set approach can be easily extended to higher dimensions.

5.2.1 3-D Ellipsoidal target

We consider an ellipsoid target in three dimension and the measurement is made on the whole boundary of the cube $[0,1]^3$. Fig. 14 shows our solutions at various iteration steps (a-e). It converges to the exact solution very well after very few iterations (30 iterations) and with relatively coarse mesh (33 grid points in each direction).

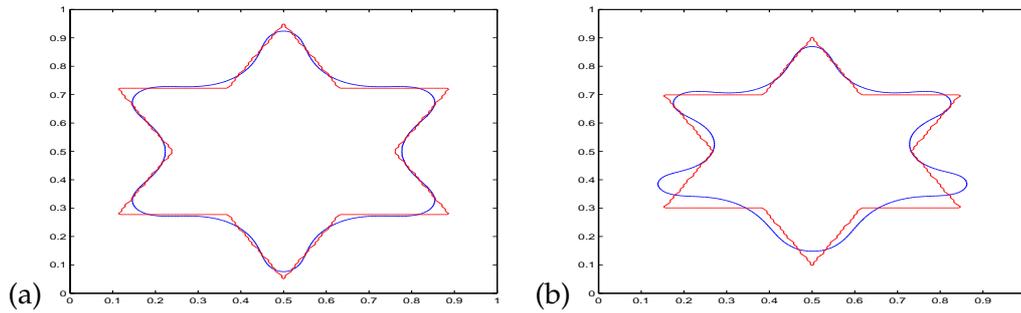


Figure 13: (Star-shaped target) (a) The solution using 129 grid points with complete measurements. (b) The solution based on measurements from numerical continuation using data on $\{(x,y): x \in [0.25,0.75], y=1\}$, 129 grid points in each direction and 512 points to discretize Γ_1 .

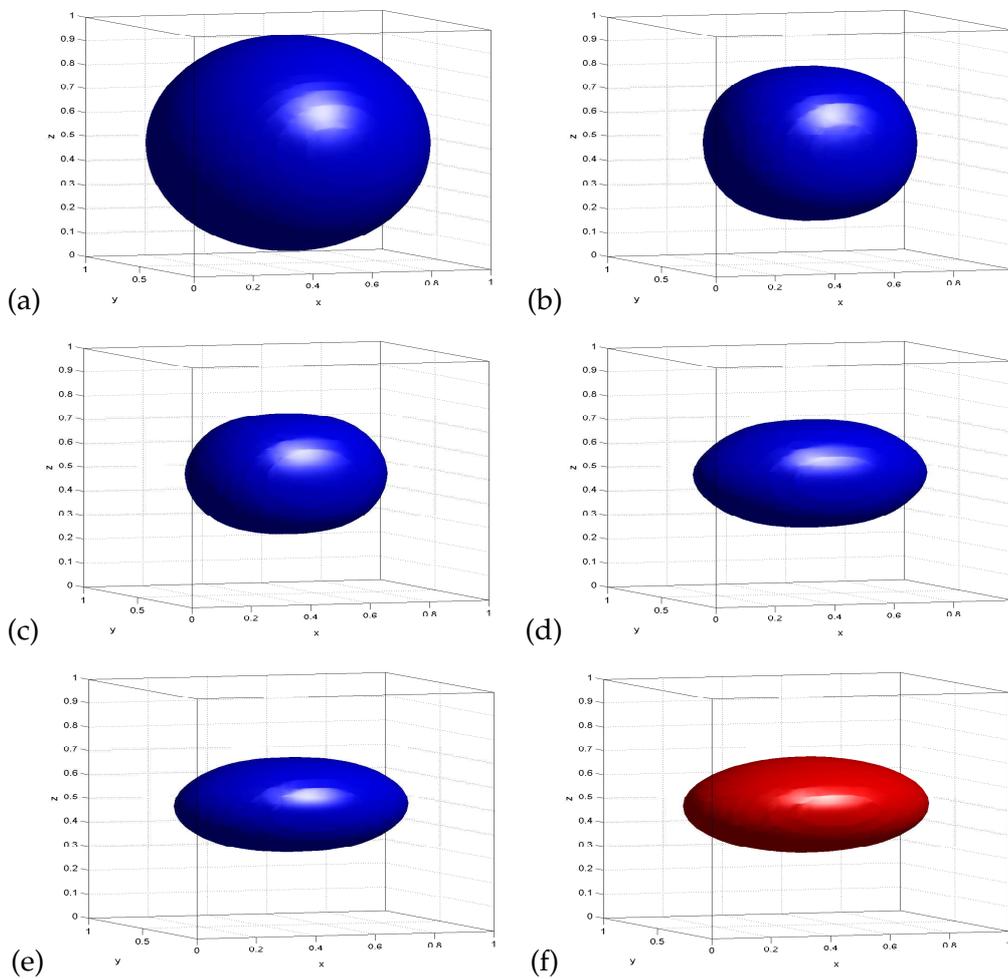


Figure 14: (Ellipsoidal Target in 3D) The solutions at various iterations using measurements on $(x,y,z) \in \partial[0,1]^3$, 33 grid points in each direction. The exact solution is plotted in (f).

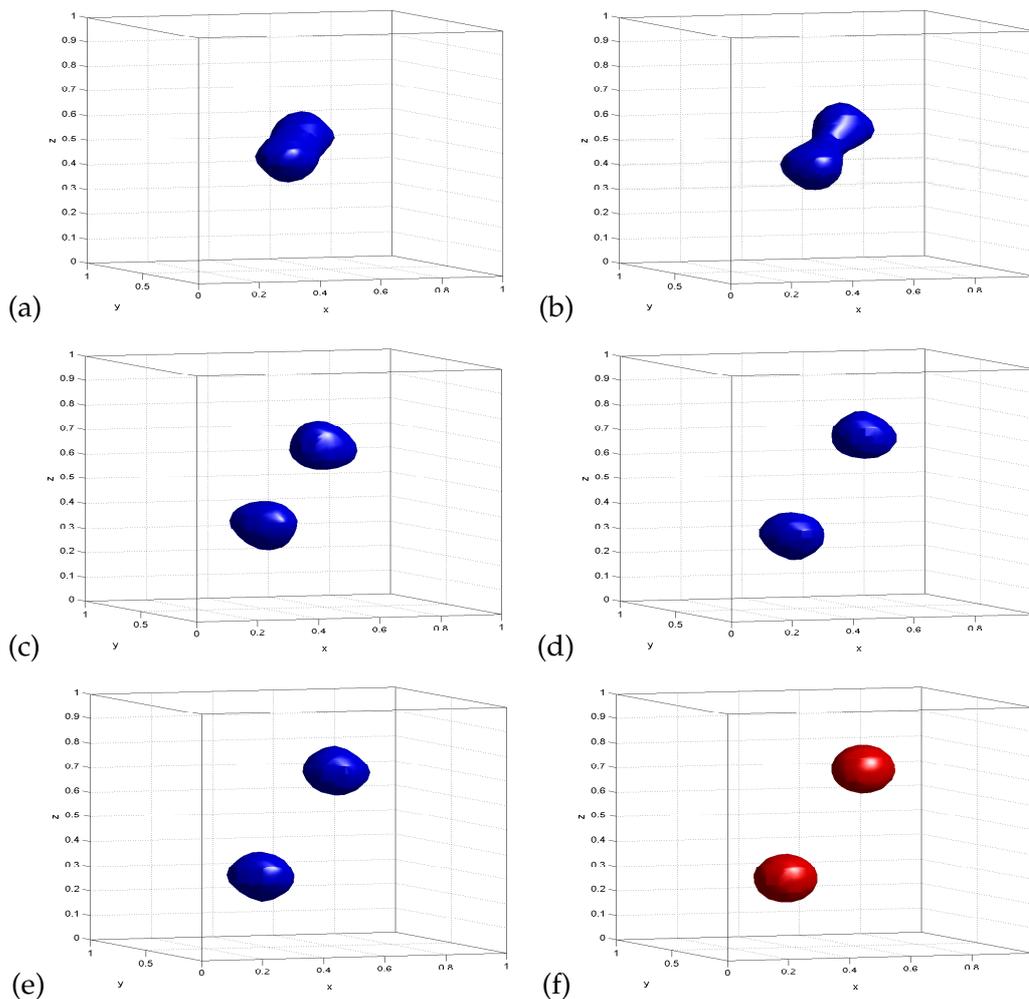


Figure 15: (Two spheres in 3D) The solutions at various iterations using measurements on $(x,y,z) \in \partial[0,1]^3$, 33 grid points in each direction. The exact solution is plotted in (f).

5.2.2 Two spheres

We consider a case in which the target consists of two spheres. To validate the algorithm, we will consider three different sets of data measurements: the complete measurement on the whole boundary of the unit cube $[0,1]^3$, the fictitious measurement based on numerical continuation of the partial data given on the surface $z = 1$, and the fictitious measurement based on numerical continuation of the partial data of 5% noise given on the surface $z = 1$.

The numerical continuation is carried out in the following way as illustrated in Fig. 1: the interface Γ is chosen to be $\sqrt{(x-0.5)^2+(y-0.5)^2+(z-0.5)^2} = 0.375$, which is discretized by $6(N-1)^2$ points with N the number of grid points of the underlying Cartesian

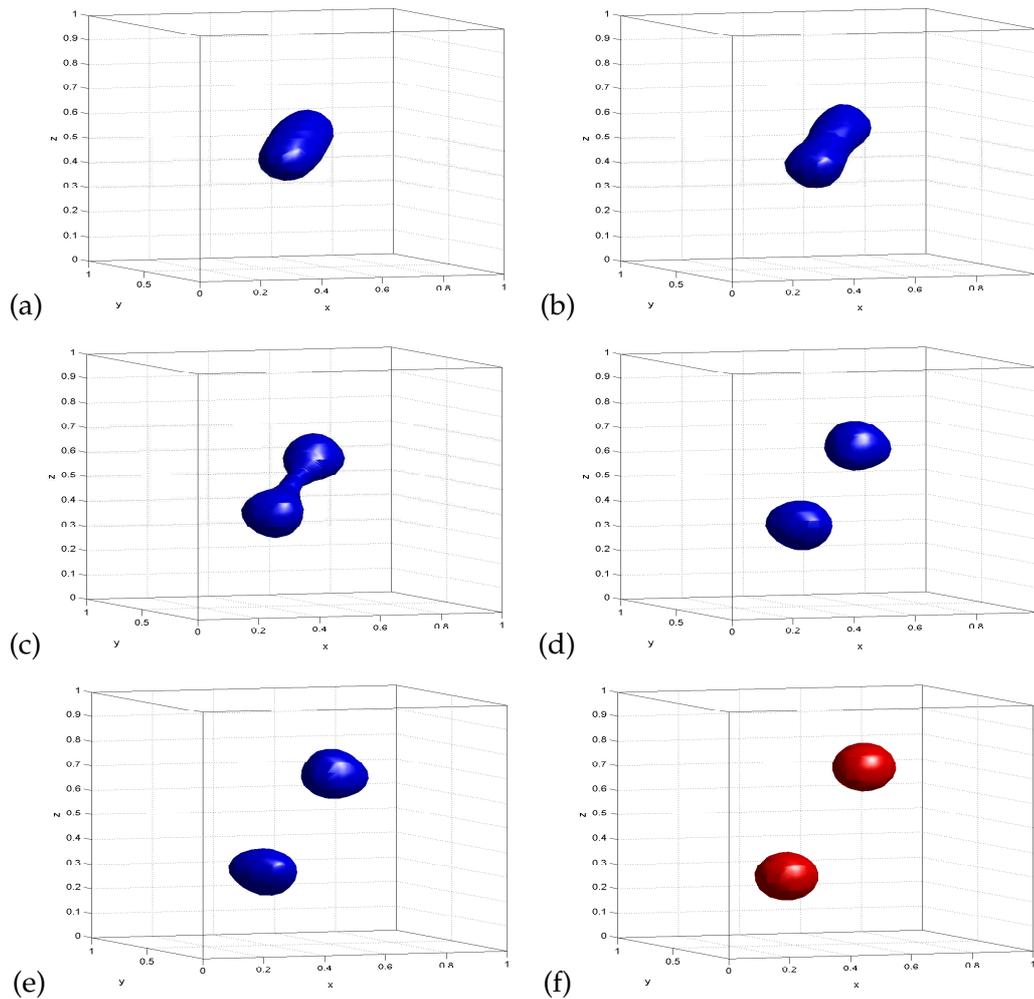


Figure 16: (Two spheres in 3D) The solution using measurements from numerical continuation using data on the surface $\{(x,y,z) : (x,y) \in [0,1]^2, z=1\}$, 33 grid points in each direction and 100^2 points to discretize $\Gamma_1 = \{(\theta, \psi) \in [0, 2\pi) \times [0, \pi]\}$. Solutions at various iterations. The exact solution is plotted in (f).

mesh in each direction, and Γ_1 is chosen to be $\max(|x-0.5|, |y-0.5|, |z-0.5|) = 0.4375$.

Fig. 15 shows the results based on the complete measurements on the whole boundary of the unit cube. As we can see, the inversion result matches with the exact solution very well.

Fig. 16 shows the results based on numerical continuation using the data on the surface $\{(x,y,z) : (x,y) \in [0,1]^2, z=1\}$. As we can see, the inversion result matches with the exact solution pretty well.

Fig. 17 shows the results based on numerical continuation using the data of 5% noise on the surface $\{(x,y,z) : (x,y) \in [0,1]^2, z=1\}$. As we can see, the inversion result matches with the exact solution in a decent way.

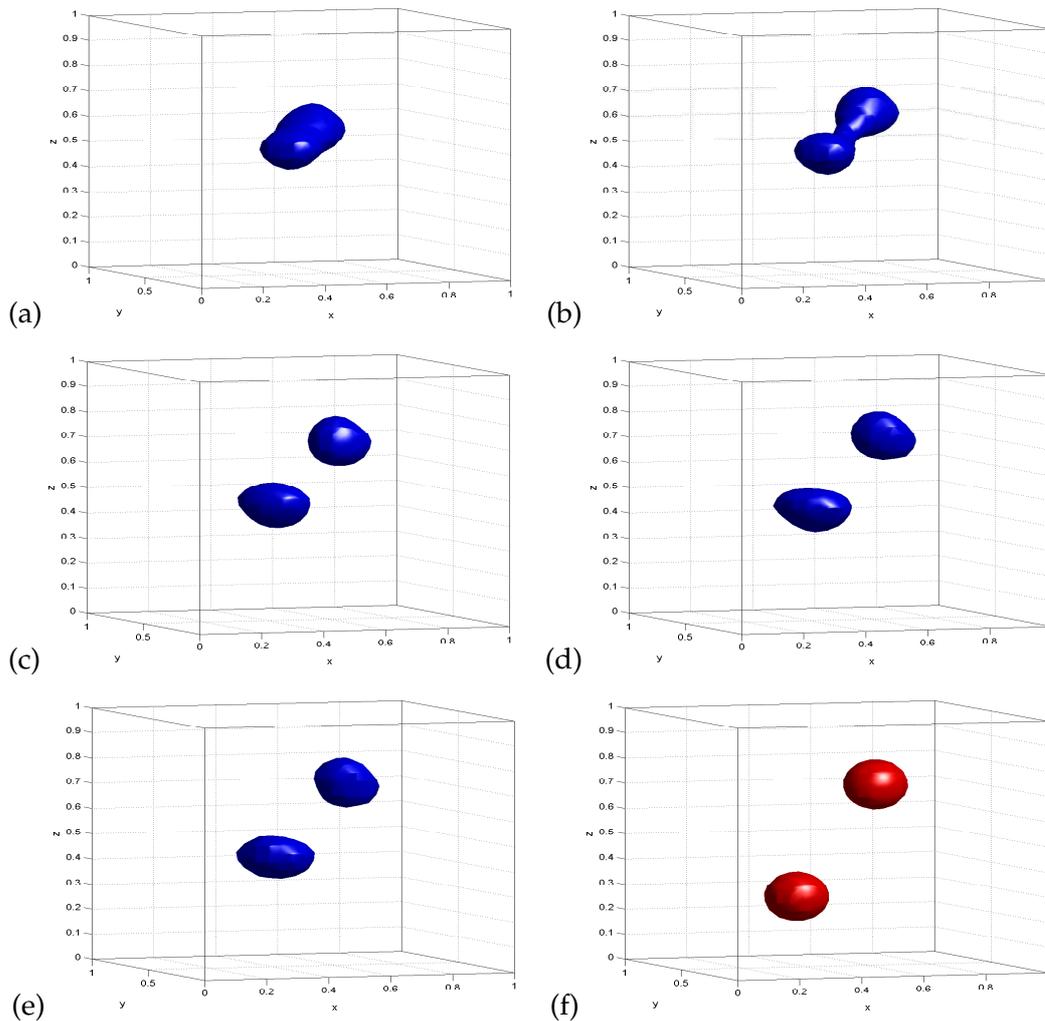


Figure 17: (Two spheres in 3D) The solution using measurements from numerical continuation using 5% noisy data on the surface $\{(x,y,z):(x,y) \in [0,1]^2, z=1\}$, 33 grid points in each direction and 100^2 points to discretize $\Gamma_1 = \{(\theta, \psi) \in [0, 2\pi] \times [0, \pi]\}$. Solutions at various iterations. The exact solution is plotted in (f).

5.2.3 Two cubes

We consider a case in which the target consists of two cubes. To validate the algorithm, we will consider three different sets of data measurements: the complete measurement on the whole boundary of the unit cube $[0,1]^3$, the fictitious measurement based on numerical continuation of the partial data given on the surface $z=1$, and the fictitious measurement based on numerical continuation of the partial data of 5% noise given on the surface $z=1$.

The numerical continuation is carried out in the following way as illustrated in Fig. 1: the interface Γ is chosen to be $\sqrt{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2} = 0.375$, which is dis-

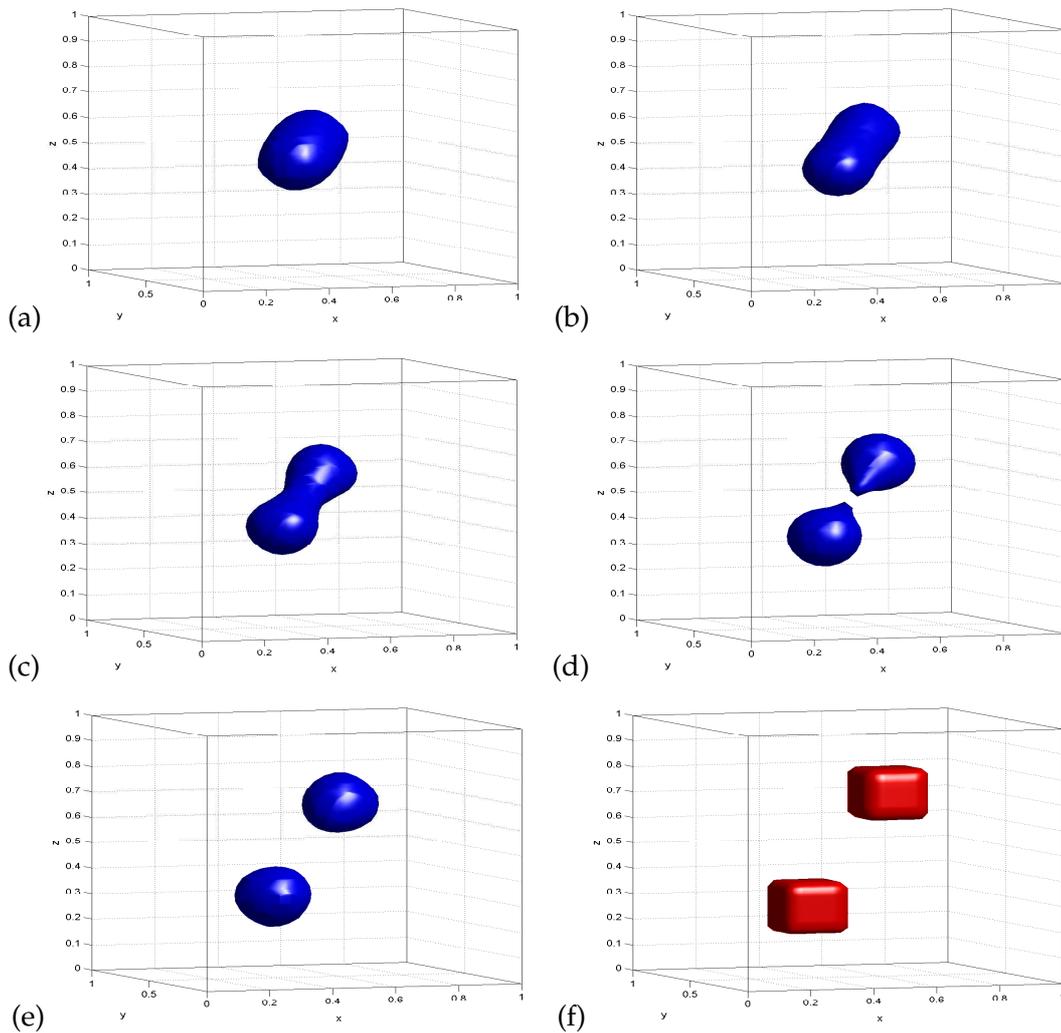


Figure 18: (Two cubes in 3D) The solutions at various iterations using measurements on $(x,y,z) \in \partial[0,1]^3$, 33 grid points in each direction. The exact solution is plotted in (f).

cretized by $6(N-1)^2$ points with N the number of grid points of the underlying Cartesian mesh in each direction, and Γ_1 is chosen to be $\max(|x-0.5|, |y-0.5|, |z-0.5|) = 0.4375$.

Fig. 18 shows the results based on the complete measurements on the whole boundary of the unit cube. As we can see, the inversion result matches with the exact solution very well.

Fig. 19 shows the results based on numerical continuation using the data on the surface $\{(x,y,z) : (x,y) \in [0,1]^2, z=1\}$. As we can see, the inversion result matches with the exact solution pretty well.

Fig. 20 shows the results based on numerical continuation using the data of 5% noise

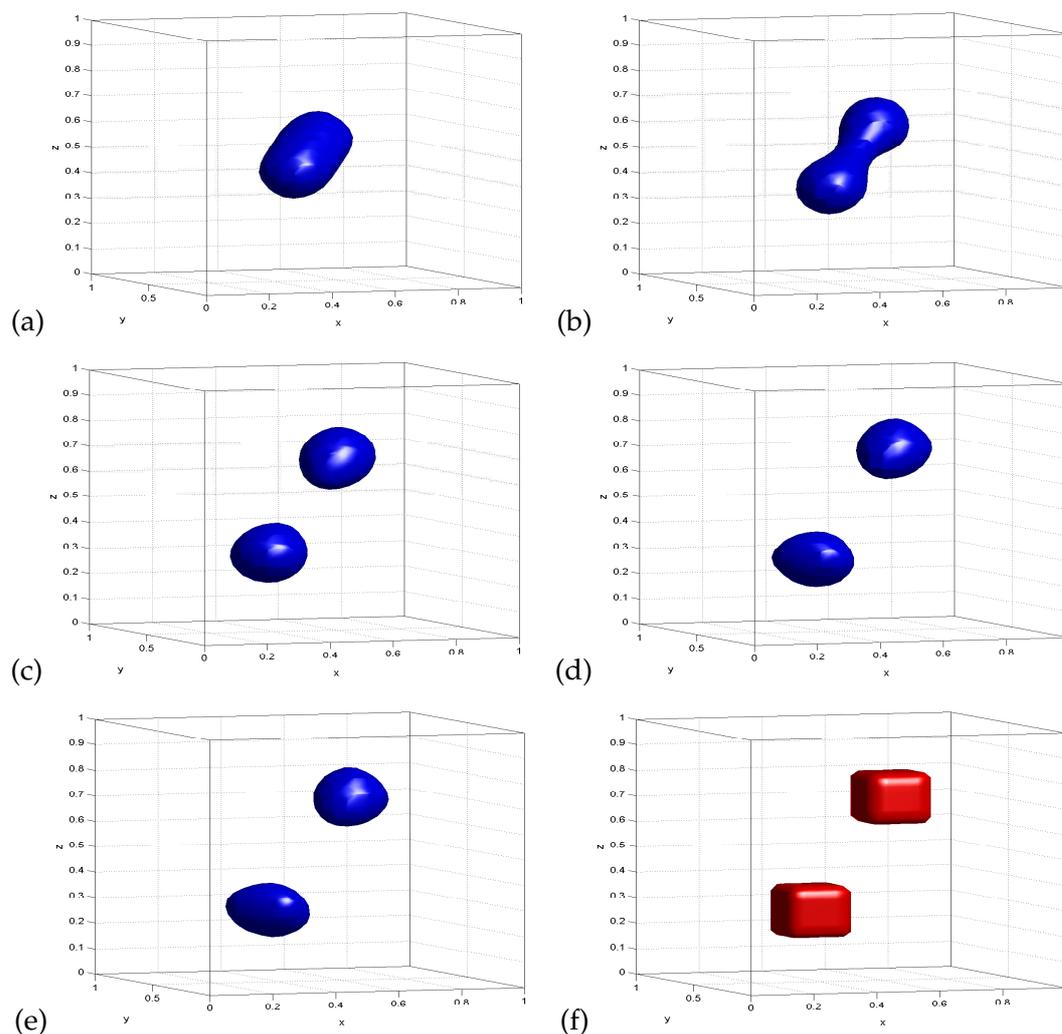


Figure 19: (Two cubes in 3D) The solutions at various iterations using measurements from numerical continuation using data on the surface $\{(x,y,z):(x,y) \in [0,1]^2, z=1\}$, 33 grid points in each direction and 100^2 points to discretize $\Gamma_1 = \{(\theta, \psi) \in [0, 2\pi] \times [0, \pi]\}$. The exact solution is plotted in (f).

on the surface $\{(x,y,z):(x,y) \in [0,1]^2, z=1\}$. As we can see, the inversion result matches with the exact solution in a decent way.

6 Conclusion

We proposed a fast local level set method for the inverse problem of gravimetry to recover open sets from their exterior volume potentials. To achieve this purpose, we have used the level set method, numerical continuation, and a novel approach for evaluating

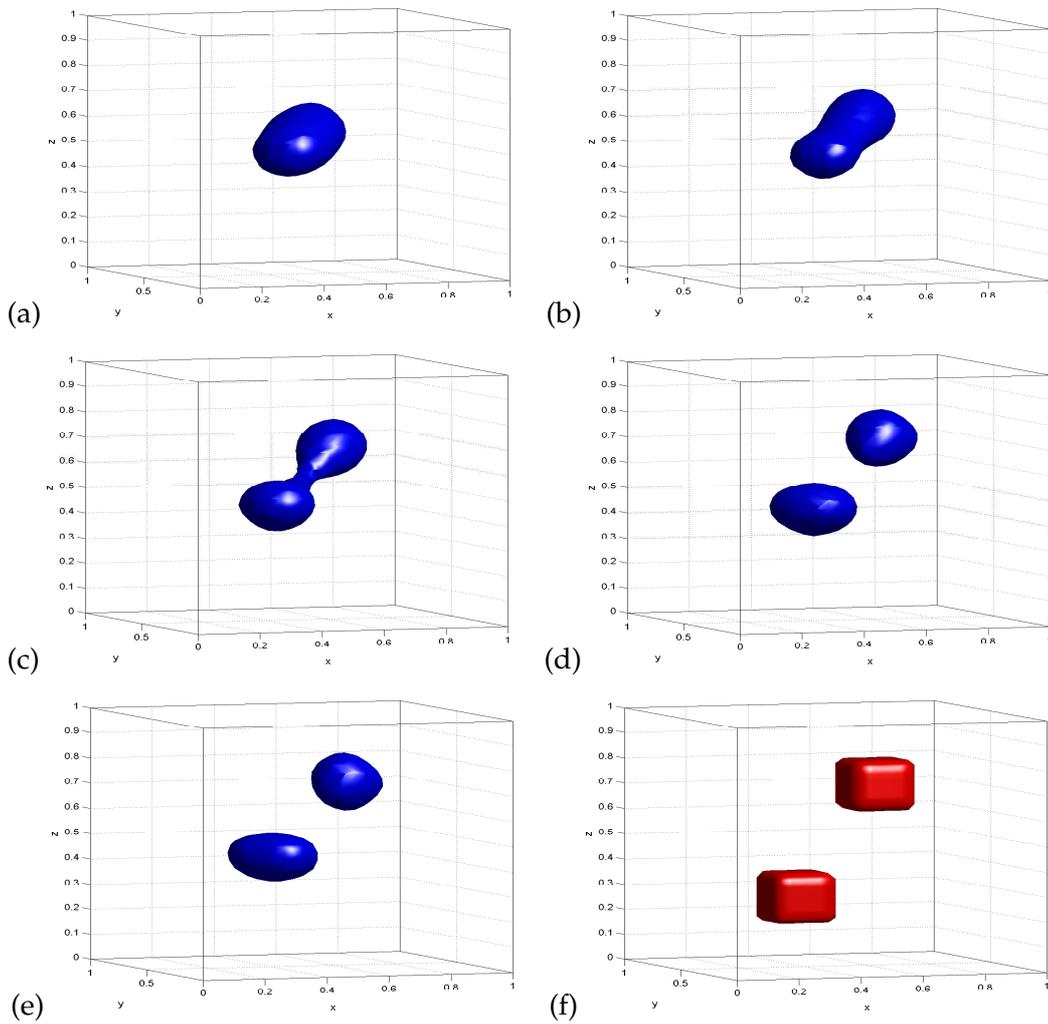


Figure 20: (Two cubes in 3D) The solutions at various iterations using measurements from numerical continuation using data of 5% noise on the surface $\{(x,y,z):(x,y) \in [0,1]^2, z=1\}$, 33 grid points in each direction and 100^2 points to discretize $\Gamma_1 = \{(\theta, \psi) \in [0, 2\pi) \times [0, \pi]\}$. The exact solution is plotted in (f).

an integral associated with the inversion process. We have carried out numerical experiments for both 2-D and 3-D cases to demonstrate the effectiveness of the new algorithm.

Acknowledgments

The research of Victor Isakov and of Jianliang Qian is supported by an NGA NURI grant. Leung’s research is partially supported by the RGC under Grant DAG09/10.SC02 and GRF602210. Qian’s research is also partially supported by NSF 0810104.

References

- [1] H. Bertete-Aguirre, E. Cherkaev, and M. Oristaglio. Non-smooth gravity problem with total variation penalization functional. *Geophys. J. Int.*, 149:499–507, 2002.
- [2] M. Burger. A level set method for inverse problems. *Inverse Problems*, 17:1327–1356, 2001.
- [3] M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. *European J. Appl. Math.*, 16:263–301, 2005.
- [4] T. Cecil, S. J. Osher, and J. Qian. Simplex free adaptive tree fast sweeping and evolution methods for solving level set equations in arbitrary dimension. *J. Comp. Phys.*, 213:458–473, 2006.
- [5] T. Cecil, J. Qian, and S. J. Osher. Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions. *J. Comp. Phys.*, 196:327–347, 2004.
- [6] T. DeLillo, V. Isakov, N. Valdivia, and L. Wang. The detection of surface vibrations from interior acoustical pressure. *Inverse Problems*, 19:507–524, 2003.
- [7] O. Dorn and D. Lesselier. Level set methods for inverse scattering. *Inverse Problems*, 22:R67–R131, 2006.
- [8] V. Isakov. *Inverse Source Problems*. American Mathematical Society, Providence, Rhode Island, 1990.
- [9] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.
- [10] S. Leung, J. Qian, and S. J. Osher. A level set method for three dimensional paraxial geometrical optics with multiple sources. *Comm. Math. Sci.*, 2:657–686, 2004.
- [11] A. Litman, D. Lesselier, and F. Santosa. Reconstruction of a 2-D binary obstacle by controlled evolution of a level-set. *Inverse Problems*, 14:685–706, 1998.
- [12] S. J. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [13] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
- [14] D. Peng, B. Merriman, S. Osher, H. K. Zhao, and M. Kang. A pde-based fast local level set method. *J. Comput. Phys.*, 155:410–438, 1999.
- [15] J. Qian and S. Leung. A level set method for paraxial multivalued traveltimes. *J. Comp. Phys.*, 197:711–736, 2004.
- [16] F. Santosa. A level-set approach for inverse problems involving obstacles. *Control, Optimizat. Calculus Variat.*, 1:17–33, 1996.
- [17] C. W. Shu and S. J. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [18] K. van den Doel, U. Ascher, and A. Leitao. Multiple level sets for piecewise constant surface reconstruction in highly ill-posed problems. *J. Sci. Comput.*, 43:44–66, 2010.
- [19] H.-K. Zhao, T. Chan, B. Merriman, and S. J. Osher. A variational level set approach for multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.