Numerical inversion of 3D geodesic X-ray transform and its application to traveltime tomography

Eric Chung

Department of Mathematics
The Chinese University of Hong Kong

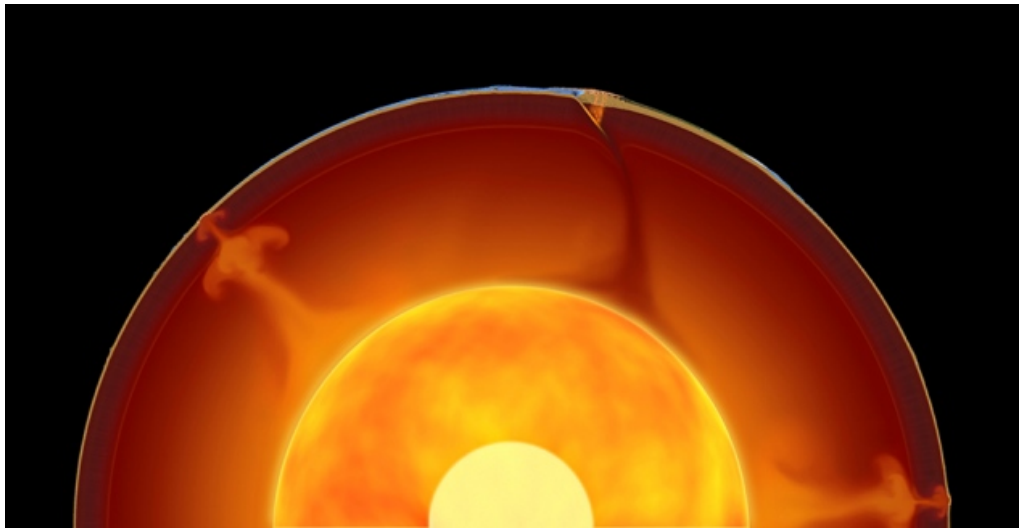
Joint work with Ivan Au Yeung and Gunther Uhlmann

Outline

1. X-ray transform
2. Numerical inversion
3. Numerical experiments
4. Traveltime tomography
5. Numerical results

Traveltime Tomography

- To image the inner structure of the Earth, we need signals that can get from there to the surface. One such signal are seismic waves.
- When there is an earthquake, a network of seismic stations around the world record the seismic wave that arrives there and in particular, time it takes the wave to get there.
- The speed of those waves depends on the structure of the Earth, and one hopes to use this information to recover the latter.



Mathematical formulation

- Assume that we have a metric $g = (g_{ij})$. We define Hamiltonian H_g by

$$H_g(x,\xi) = \frac{1}{2}\Big(\sum_{i,j=1}^{n} g^{ij}(x)\xi_i\xi_j - 1\Big)$$

  for each $x \in \Omega$ and $\xi \in \mathbb{R}^n$, where $(g_{ij})^{-1} = (g^{ij})$.

- Let $X^{(0)} = (x^{(0)}, \xi^{(0)})$ be a given initial condition, where $x^{(0)} \in \partial\Omega$ and $\xi^{(0)} \in \mathbb{R}^n$, such that the inflow condition holds,

$$H_g(x^{(0)}, \xi^{(0)}) = 1, \qquad \sum_{i,j=1}^{n} g^{ij}(x^{(0)})\xi_i^{(0)}\nu_j(x^{(0)}) < 0$$

  where $\nu(x)$ is the unit outward normal vector of $\partial\Omega$ at the point x and $\nu_j(x)$ denote the jth component of this vector.

Hamiltonian system

We define $X_g(s, X^{(0)}) = (x_g(s, X^{(0)}), \xi_g(s, X^{(0)}))$ by the solution to the hamiltonian system defined by

$$\frac{dx}{ds} = \frac{\partial H_g}{\partial \xi} \quad , \quad \frac{d\xi}{ds} = -\frac{\partial H_g}{\partial x}$$

with the initial condition

$$(x^{(0)}, \xi^{(0)}) = X^{(0)}.$$

The solution X_g defines a geodesic/ray in the phase space. The parameter s denotes travel time. Thus, we denote the set of geodesics X_g which are contained in Ω with endpoints at $\partial\Omega$ by $\mathcal{M}_\Omega$.

Case of an isotropic medium

$$g_{ij} = \frac{1}{c^2}\delta_{ij},$$

where c is a function from $\mathbb{R}^n$ to $\mathbb{R}$.

The reconstruction method

- Let f be a smooth function from Ω to $\mathbb{R}$. To determine the unknown function $f(x)$ from the geodesic X-ray data of the function, our method is based on a truncation of convergent Neumann series.
- First, we define the geodesic X-ray transform of the function f defined on Ω as the collection $(If)(X_g)$ of integrals of f along geodesic $X_g \in \mathcal{M}_\Omega$, where

$$(If)(X_g) := \int_{x(s)} f(s)\, ds,$$

  where $X_g(s) = (x(s), \xi(s))$.
- We note that $(If)(X_g)$ this is the measurement data, and we use this data to reconstruct $f(x)$.

- Let Λ be the adjoint of the operator I. Then Uhlmann and Vasy show that there is an operator R such that

$$R\Lambda(If) = f - Kf$$

- The error operator K is small in the sense of $||K|| < 1$ for an appropriate norm.

Neumann series

The unknown function f can be represented by the following convergent Neumann series

$$f = \sum_{n=0}^{\infty} K^n R\Lambda(If) \tag{1}$$

Remark

- The operator R is related to the inverse of the operator $\Lambda \circ I$.
- Results from Uhlmann suggest that the unknown function f can be reconstructed locally in a layer by layer fashion.

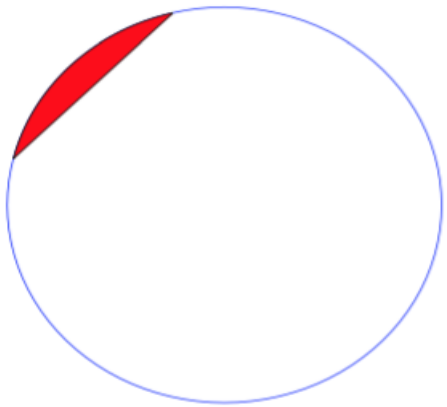
Layer by layer computations

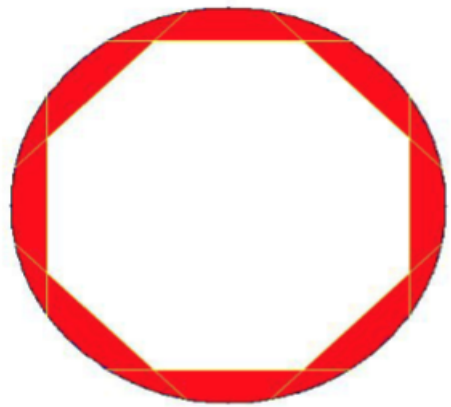
One can first reconstruct the unknown function f using (1) in small neighborhoods near the boundary of the domain, and then repeat the procedure in the next inner layer of the domain, and so on.

- The challenges in the numerical computations of the unknown function f using the above representation:
  - computing the operators Λ and R
  - implementating the layer stripping algorithm

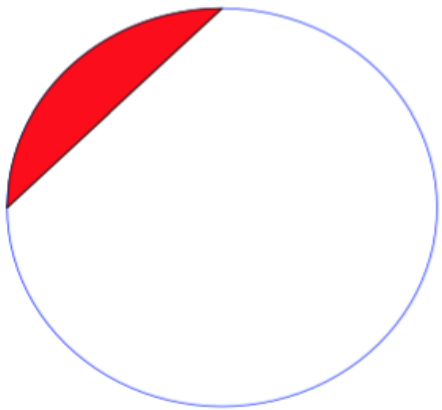
An illustration

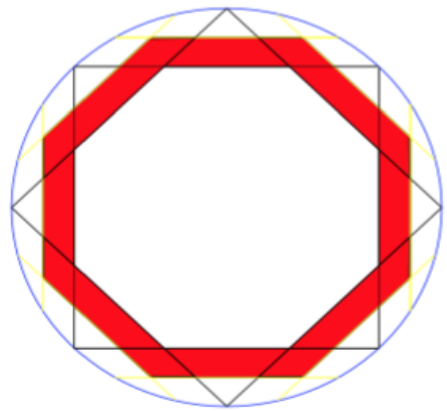
First layer:





Second layer:





Numerical procedure

- Let Z be a set of grid points, denoted as $\{z_i\}$, in the domain Ω.
- We will determine the values of the unknown function $f(x)$ at these grid points using the given data set $\{(If)(X_g)\}$.
- For a given point $x \in \Omega$, we define $\mathcal{M}_\Omega(x)$ as the set of all geodesics passing through the point x. We use the notation $|\mathcal{M}_\Omega(x)|$ to represent the number of elements in the set $\mathcal{M}_\Omega(x)$.
- Numerically, we can define the action of the operator $\Lambda \circ I$ as the average of the line integrals (back-projection) by

$$\Lambda(If) := \frac{\sum_{j=1}^{|\mathcal{M}_\Omega(x)|} (If)(X_g^j)}{|\mathcal{M}_\Omega(x)|} \tag{2}$$

where we use the notation $\mathcal{M}_\Omega(x) = \{X_g^j\}$, $j = 1, 2, \cdots, |\mathcal{M}_\Omega(x)|$.

Remark

Recall that

$$\Lambda(If) := \frac{\sum_{j=1}^{|\mathcal{M}_\Omega(x)|} (If)(X_g^j)}{|\mathcal{M}_\Omega(x)|}$$

- Notice that the above formula defines a function with domain Ω using the given geodesic X-ray transform data $(If)(X_g)$.
- For standard X-ray transform, the above operator Λ gives a good approximation to the unknown function $f(x)$.
- For geodesic X-ray transform, this operator provides an initial approximation, which is the initial term in a convergent Neumann series representation of $f(x)$.

Neumann series

- We will use an operator A to model the action of the operator R presented above.
- We will construct an operator A such that

$$(A^*A)^{-1}\Lambda \circ I = Id - K, \tag{3}$$

where K is an error operator with $||K|| < 1$ for some appropriate norm, and Id is the identity operator.

- The operator A^* is the adjoint operator of A.

Neumann series

Using the above, we can write the following Neumann series

$$f = \sum_{n=0}^{\infty} K^n (A^*A)^{-1}\Lambda(If)$$

- Note, the inverse of A^*A is an approximate inverse of the operator $\Lambda \circ I$.

Numerical implementations

Recall the Neumann series

$$f = \sum_{n=0}^{\infty} K^n (A^* A)^{-1} \Lambda(If)$$

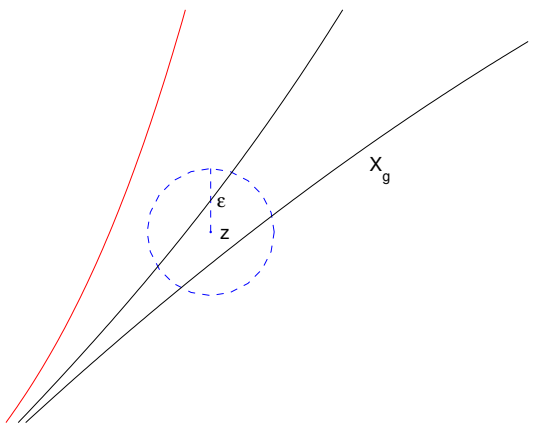
- In the previous section, we presented a general overview of our numerical procedure and its corresponding theoretical motivations.
- There are three important ingredients in our numerical algorithm. They are
  1. Given the data $\{(If)(X_g)\}$, compute $\Lambda(If)$.
  2. Compute the action of $\Lambda \circ I$.
  3. Compute the action of $K := Id - (A^* A)^{-1}(\Lambda \circ I)$.

Implementation details

- Use the following formula for the computation of $\Lambda(If)$ using the given data set $\{(If)(X_g)\}$, where

$$\Lambda(If) := \frac{\sum_{j=1}^{|\mathcal{M}_\Omega(x)|} (If)(X_g^j)}{|\mathcal{M}_\Omega(x)|}, \qquad (4)$$

- We will only recover the unknown function f on the set of points Z, the output $\Lambda(If)$ is also defined only on the same set of grid points Z.
- But, in general, cannot find the geodesic that is passing through a given point $z_i \in Z$.
- We will choose a small neighborhood of z_i and find all the geodesics passing through the ϵ-neighborhood of z_i.
- Then, apply the above formula (4) using this set of geodesics for each z_i.



Implementation details

- Next, we will discuss the action of the operator $\Lambda \circ I$.
- For a given function f whose values are defined only on the set of grid points Z, we will evaluate If.
- We need to compute the integral of f on a geodesic X_g.
- Solve the ray equation in the phase space starting from a particular initial point $X^{(0)}$ by the 4th order Runge-Kutta Method.
- Obtain a set of points $\{x(s_i)\}$ defining the geodesic in the physical space.

Implementation details

- Use a version of the trapezoidal rule to compute the line integral $(If)(X_g)$ of the function f along the geodesic X_g.

$$(If)(X_g) \approx \sum_i f(x(s_i))\,(x'(s_i))\,(s_i - s_{i-1}).$$

- $(s_i - s_{i-1})$ is the step size used in the 4th order Runge-Kutta Method.
- $x'(s_i)$ can be computed using $x' = \dfrac{\partial H_g}{\partial \xi}$.
- The term $f(x(s_i))$ is not well defined since f is only defined on the set of grid points Z and the point $x(s_i)$ may not be one of the grid points. To overcome this issue, we replace $f(x(s_i))$ by $\widehat{f}(x(s_i))$, which is the linear interpolation of f using the grid points near $x(s_i)$.

Action of If

The formula to compute the action of If:

$$(If)(X_g) \approx \sum_i \widehat{f}(x(s_i))\,(x'(s_i))\,(s_i - s_{i-1}). \tag{5}$$

Coarse and fine grid

- Finally, we discuss the action of the error operator K.
- Compute the operator $(A^*A)^{-1}$, which approximates the operator R.
- The operator A is essentially integrals along geodesics, and the operator A^* performs average of line integrals passing through a given point. Notice that, the action of A is similar to that of I, and the action of A^* is similar to the action of Λ.
- In order to obtain a good approximation to the operator R and hence a good reconstructed f, we will perform the action of A and A* on a finer grid Z_f, which is a refinement of the grid Z.
- Without finer grid, it's impossible to compute the terms in the Neumann series.
- The following is the example of 2D grids:

$$\begin{array}{ccccc} \cdot & \times & \cdot & \times & \cdot \\ \times & \cdot & \times & \cdot & \times \\ \cdot & \times & \cdot & \times & \cdot \end{array}$$

All the $\{\times\}$ form the coarse grid Z and all $\{\times, \cdot\}$ form the fine grid Z_f. (There are other choices)

Reconstruction formula

- To complete the steps, we need a projection operator P, which maps functions defined on the finer grid Z_f to functions defined on the grid Z.
- The operator P^* maps functions defined on the grid Z to functions defined on the finer grid Z_f.
- Now, we can write down the reconstruction formula:

$$f = \sum_{n=0}^{\infty} K^n P(A^* A)^{-1} P^* \Lambda(If)$$

where

$$K = Id - P(A^* A)^{-1} P^* (\Lambda \circ I).$$

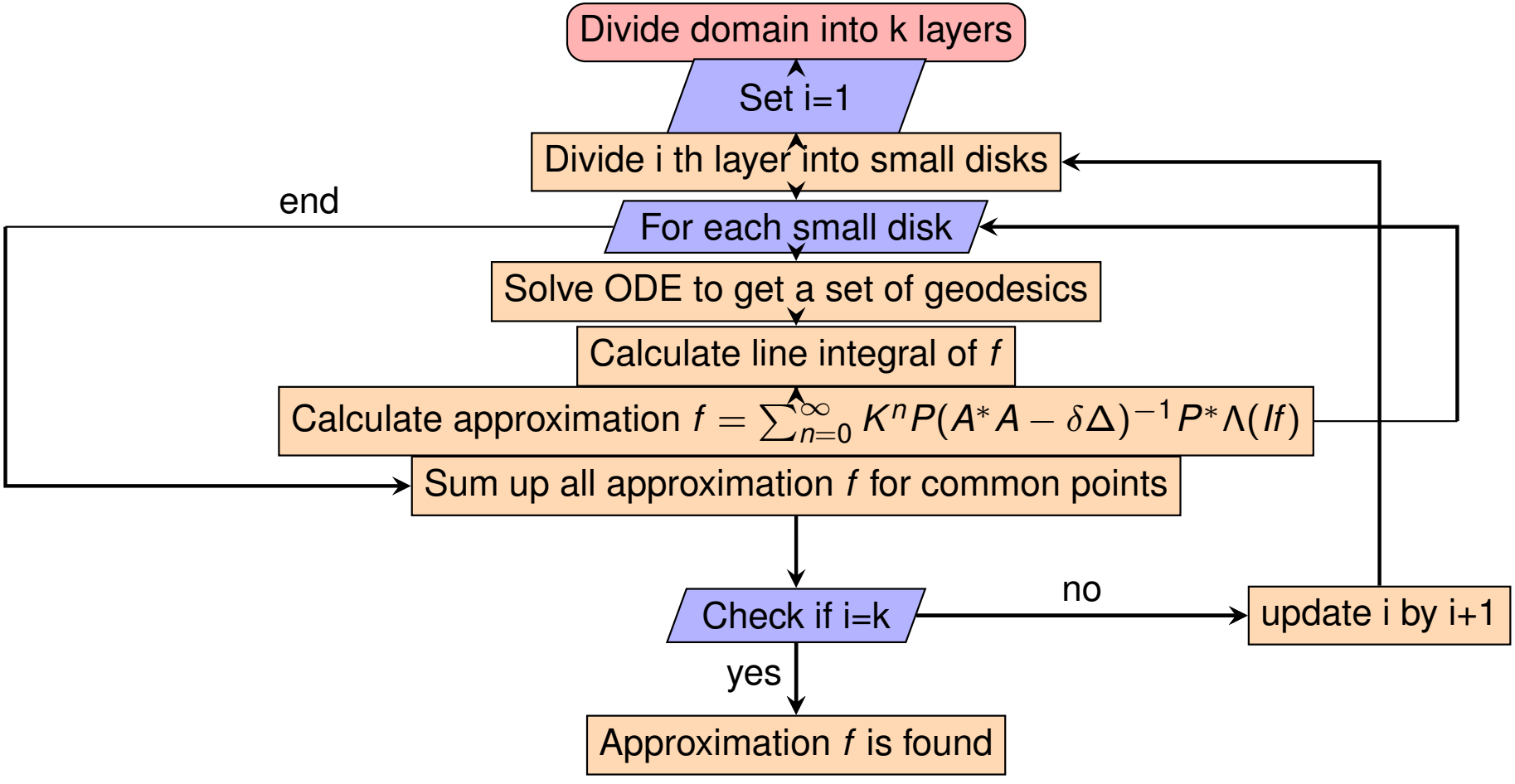
Reconstruction formula

To regularize the problem, we will replace the above sum by

$$f = \sum_{n=0}^{\infty} K^n P(A^* A - \delta\Delta)^{-1} P^* \Lambda(If) \tag{6}$$

where $\delta > 0$ is a regularization parameter and Δ is the Laplace operator.

Layer stripping algorithm for 3D model



Accuracy tests

- We compared results for different test functions. The fine rectangular grid used is chosen with grid size $h = 0.02$. We assume that the speed c is chosen as

$$c(x, y, z) = 1 + 0.3 \times \cos(\sqrt{(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2}).$$

- There were five test cases for this experiment:
  $f_1 = 0.01 + \sin(2\pi(x + y + z)/10)$,
  $f_2 = 0.01 + \sin(2\pi(x + y)/10) + \cos(2\pi z/20)$,
  $f_3 = x + y^2 + z^2/2$,
  $f_4 = 1 + 6x + 4y + 9z + \sin(2\pi(x + z)) + \cos(2\pi y)$,
  $f_5 = x + e^{y+z/2}$.

Accuracy tests(cont.)

relative error	f_1	f_2	f_3	f_4	f_5
n=0	47.28%	47.45%	46.83%	46.97%	47.18%
n=1	23.89%	24.10%	23.43%	23.61%	23.76%
n=2	13.09%	13.26%	13.03%	13.23%	13.00%
n=3	8.53%	8.52%	9.32%	9.48%	8.53%
n=4	6.99%	6.74%	8.71%	8.80%	7.14%

Table: Relative errors for the functions f_i, $i = 1, \cdots, 5$.

Accuracy tests with noisy data

- Next, we show some numerical tests using noise contaminated data $\Lambda^\epsilon g$. We use the same speed

$$c(x, y, z) = 1 + 0.3 \times \cos(\sqrt{(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2})$$

- reconstructing the function

$$g(x, y, z) = 0.01 \sin(2\pi(x + y + z)/10).$$

- The measurement data has been contaminated by uniformly distributed noise ϵ,

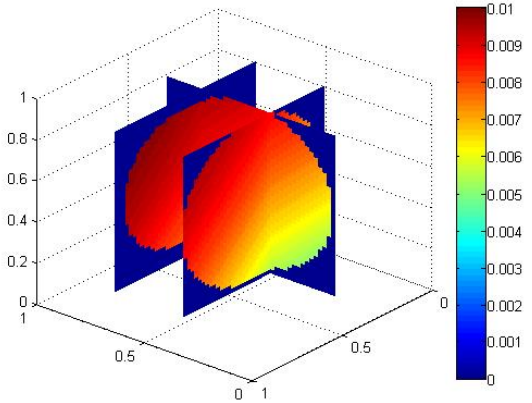
$$\Lambda^\epsilon g := \Lambda g + \epsilon$$

with relative error $|\epsilon|/|\Lambda g| = 0.05$ (i.e. 5% noise), where ϵ is a random function.

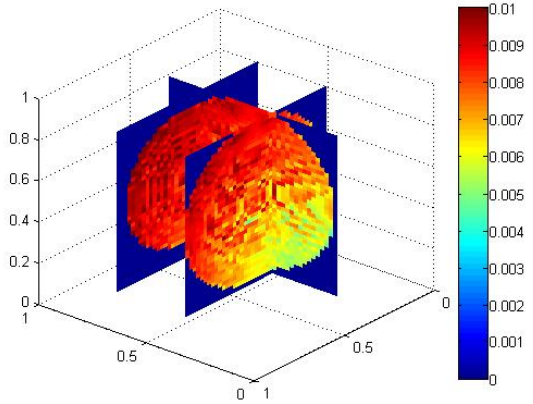
relative error	g_1	g_2
n=5	6.72%	8.50%

Table: Tables of relative errors of test functions reconstructing from exact data g_1 and noisy data g_2

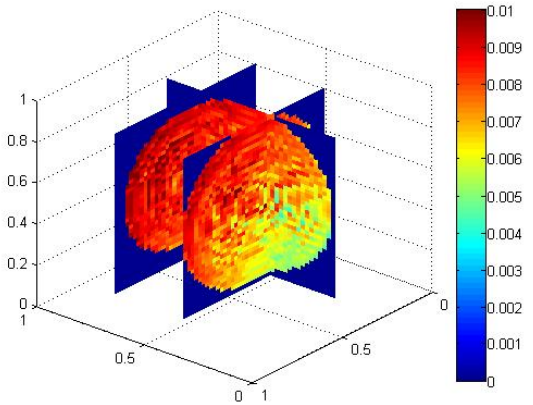
Accuracy tests with noisy data (cont.)



: exact solution



: approximate solution for g_1



: approximate solution for g_2

Experiments with different speeds

- We compare the numerical tests with different speeds. We use grid size $h = 0.02$ as above and perform this test by considering function

$$f(x, y, z) = 0.01 \sin(2\pi(x + y + z)/10)$$

- The first speed to test is defined as

$$c_1(x, y, z) = 1 + 0.2 \sin(3\pi x) \sin(\pi y) \sin(2\pi z).$$

- The second and the third tests are related to the well-known benchmark problem: the Marmousi model. In our simulations, we take two spherical sections of the Marmousi model, called c_2 and c_3.

Experiments with different speeds(cont.)

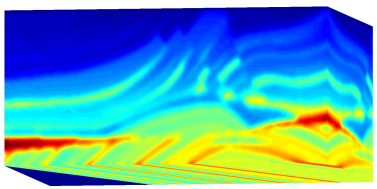


Figure: The 3D Marmousi model.

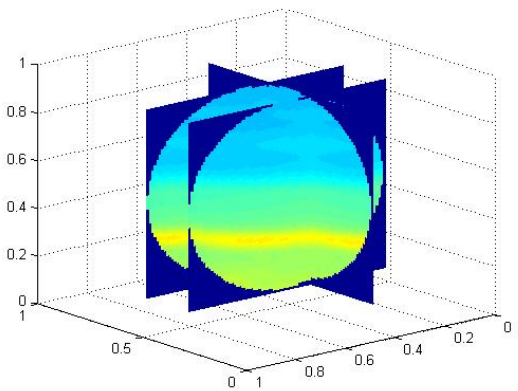


Figure: A spherical part as the test speed c_2.

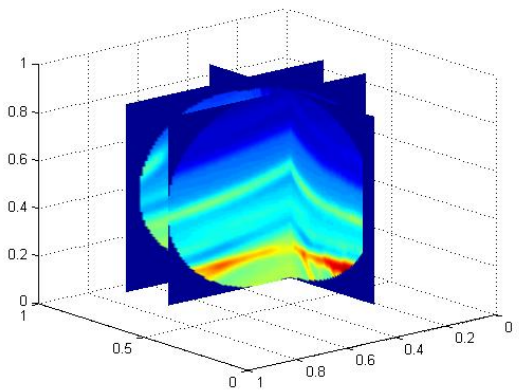


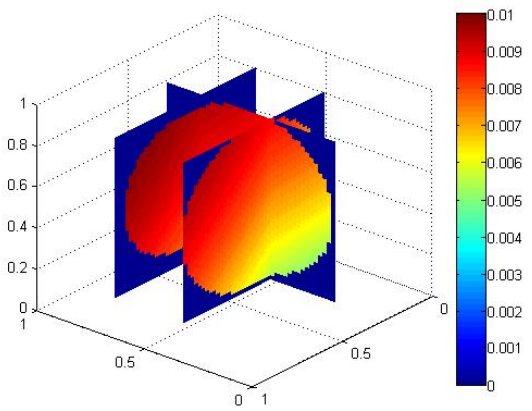
Figure: A spherical part as the test speed c_3.

Experiments with different speeds(cont.)

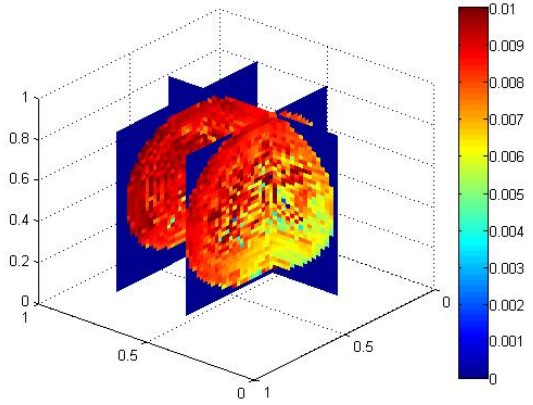
	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
relative error for test speed c_1	44.86%	22.88%	14.19%	11.62%	11.47%
relative error for test speed c_2	48.02%	25.39%	15.71%	12.42%	11.80%
relative error for test speed c_3	58.18%	35.41%	22.78%	16.25%	13.39%

Table: Relative errors for using the 3 test speeds with grid size $h = 0.02$.

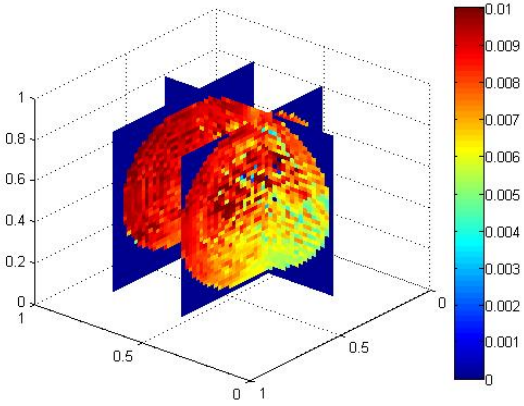
Experiments with different speeds(cont.)



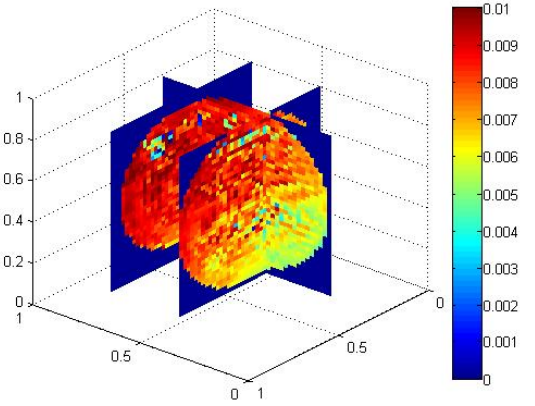
: exact solution



: approximate solution for c_1



: approximate solution for c_2



: approximate solution for c_3

Traveltime tomography

Recall the forward model:

We define $X_g(s, X^{(0)}) = (x_g(s, X^{(0)}), \xi_g(s, X^{(0)}))$ by the solution to the hamiltonian system defined by

$$\frac{dx}{ds} = \frac{\partial H_g}{\partial \xi} \quad , \quad \frac{d\xi}{ds} = -\frac{\partial H_g}{\partial x}$$

with the initial condition

$$(x^{(0)}, \xi^{(0)}) = X^{(0)}.$$

For isotopic model, we have

$$g_{ij} = \frac{1}{c^2}\delta_{ij}$$

Traveltime tomography

- Given boundary measurements for g_1, we are interested in recovering the metric g_1.
- Let g_1 and g_2 be two metrics. We link two metrics by introducing the function

$$F(s) := X_{g_2}\left(t - s, X_{g_1}(s, X^{(0)})\right),$$

  where $t = t_{g_1}$ and $t_g = t_g(X^{(0)})$ is the length of the geodesic issued from $X^{(0)}$ with the endpoint on Γ. This is the Stefanov-Uhlmann identity.
- Consequently, we have

$$\int_0^t F'(s)ds = X_{g_1}(t, X^{(0)}) - X_{g_2}(t, X^{(0)}).$$

- g_2 is an arbitrary metric.

Stefanov-Uhlmann identity

The numerical inversion is based on the theories by Stefanov and Uhlmann.

Stefanov-Uhlmann identity

$$\int_0^t F'(s)ds = \int_0^t \frac{\partial X_{g_2}}{\partial X^{(0)}}\left(t-s, X_{g_1}(s, X^{(0)})\right) \times \left(V_{g_1} - V_{g_2}\right)\left(X_{g_1}(s, X^{(0)})\right) ds$$
$$= \int_0^t J_{g_2}\left(t-s, X_{g_1}(s, X^{(0)})\right) \times \left(V_{g_1} - V_{g_2}\right)\left(X_{g_1}(s, X^{(0)})\right) ds,$$

where

$$V_{g_j} := \left(\frac{\partial H_{g_j}}{\partial \xi}, -\frac{\partial H_{g_j}}{\partial x}\right) = \left(g^{-1}\xi, -\frac{1}{2}\nabla_x (g^{-1}\xi)\cdot\xi\right).$$

Linearizing the Stefanov-Uhlmann identity(cont.)

We linearize the above identity about g_2,

$$\int_0^t F'(s)ds \approx \int_0^t J_{g_2}\left(t-s, X_{g_1}(s, X^{(0)})\right) \times \partial_{g_2} V_{g_2}(g_1 - g_2)\left(X_{g_2}(s, X^{(0)})\right)ds,$$

where $\partial_g V_g(\lambda)$ is the derivative of V_g with respect to g at λ.

Approximation of Stefanov-Uhlmann identity

$$X_{g_1}(t, X^{(0)}) - X_{g_2}(t, X^{(0)}) \approx \int_0^t J_{g_2}\left(t-s, X_{g_2}(s, X^{(0)})\right) \times \partial_{g_2} V_{g_2}(g_1-g_2)\left(X_{g_2}(s, X^{(0)})\right)ds$$

Isotropic medium

By the group property of Hamiltonian flows the Jacobian matrix is equal to

$$J_{g_2}\left(t-s, X_{g_2}(s, X^{(0)}\right) = J_{g_2}\left(t, X^{(0)}\right) \times J_{g_2}\left(s, X^{(0)}\right)^{-1}$$

Case of an isotropic medium

$$g_{ij} = \frac{1}{c^2}\delta_{ij},$$

where c is a function from $\mathbb{R}^n$ to $\mathbb{R}$. Then

$$V_{g_k} = (c_k^2\xi, -(\nabla c_k)c_k|\xi|^2).$$

Hence the derivative of V with respect to g, $\partial_g V_g(\lambda)$ is given by

$$\partial_g V_g(\lambda) = (2c\lambda\xi, -(\nabla c\cdot\lambda + \nabla\lambda\cdot c)|\xi|^2).$$

New phase space method

- Let $z_j, j = 1, 2, \ldots, p$ be the grid points.Then the set Z is defined by $Z = \{z_j, j = 1, 2, \ldots, p\}$.
- Let $X_i^{(0)} \in \mathcal{S}^-, i = 1, 2, \ldots, m$, be the initial points and directions.
- From these initial points, we can define the scattering relation $X_g(t_i, X_i^{(0)}) \in \mathcal{S}^+$. t_i is the time of ith geodesic from starting point to the end point.
- First we set an initial guess g^0. Then we construct a sequence g^n by the following way.

New phase space method(cont.)

- Define the mismatch vector

$$d_i^n = X_g(t_i, X_i^{(0)}) - X_{g^n}(t_i, X_i^{(0)}).$$

- Define an operator $\hat{I}_i$ along the ith geodesic by the integration equation

$$I_i(g - g^n) := \int_0^t J_{g^n}\left(t - s, X_{g^n}(s, X_i^{(0)})\right) \times \partial_{g^n} V_{g^n}(g - g^n)\left(X_{g^n}(s, X_i^{(0)})\right) ds.$$

- We use the above reconstruction method to recover $\lambda := g - g^n$ at each grid points by the mismatch vector.
- We define an operator I along the ith geodesic.

$$I_i(\lambda) = d_i^n.$$

New phase space method(cont.)

- For each $n \geq 0$, we use the reconstruction formula (6)

$$\lambda = \sum_{n=0}^{\infty} K^n P(A^* A - \delta\Delta)^{-1} P^* \Lambda(I\lambda),$$

where P, A is defined as the same as the previous chapter and

$$K = Id - P(A^* A)^{-1} P^* (\Lambda \circ I).$$

- We then define,

$$g^{n+1} = g^n + \lambda.$$

Then we recover the metric g by this iterative algorithm.

Numerical implementations

- We will briefly explain the details of the numerical implementations of phase space method.
  1. Discuss the detail of constructing the mismatch vector d_i^n.
  2. Explain the calculation of the line integrals, i.e. the operator I_i.
  3. Explained the detail of the reconstruction formula and how the metric is updated .

Setup of mismatch vector

- The mismatch vector is defined by $d_i^n = X_g(t_i, X_i^{(0)}) - X_{g^n}(t_i, X_i^{(0)})$. Hence, we need a set of the initial locations and directions $\{X_i^{(0)}\}$.
- For our settings, we will divide the 3D domain into different layers and thus divide the layer into many small disks. We will choose 900 uniform initial locations and directions around the boundary of the domain.
- From this set of data, we will derive a set of mismatch vectors using the guess g^n and also the observed data $X_g(t_i, X_i^{(0)})$.
- We will eliminate the geodesics which do not remain in the same layer.
- Solutions from outer layers are used as data for inner layers.

Calculation of the line integrals

- The operator I_i is defined to calculate the line integrals, which is defined by

$$I_i(g - g^n) := \int_0^{t_i} J_{g^n}\left(t_i - s, X_{g^n}(s, X_i^{(0)})\right) \times \partial_{g^n} V_{g^n}(g - g^n)\left(X_{g^n}(s, X_i^{(0)})\right) ds.$$

- Since we have the discrete phase space of each geodesic, i.e. $X_{g^n}(s_j, X_i^{(0)})$ for any $0 \leq s_j \leq t_i$, then we can approximate the operator by

$$\begin{aligned} I_i(g - g^n) \approx \sum_{s_j} & J_{g^n}\left(t_i - s_j, X_{g^n}(s_j, X_i^{(0)})\right) \\ & \times \partial_{g^n} V_{g^n}(g - g^n)\left(X_{g^n}(s_j, X_i^{(0)})\right) \left(X'_{g^n}(s_j, X_i^{(0)})\right) (s_j - s_{j-1}). \end{aligned}$$

- The operator can be approximated by a matrix and thus the adjoint operator I_i^*.

Update of the metric

- After we construct the line integral operators, we will apply the reconstruction formula to compute the update of the metric.
- Based on the reconstruction formula, it is the infinite sum of Neumann series. In computation, we need to choose some terms of the infinite sum to represent the whole term.
- Here, we choose the first five terms, since this terms represent the main part of the sum.
- After doing the update for each disks in the same layer, we will compute the final metric of this layer and move on to the next layer.
- When we compute the final metric of this layer, we take the mean of this values of overlapping regions.

Numerical results of New phase space method

- We demonstrate the performance of our method using several test examples. The domain Ω is a sphere with center $(0.5, 0.5, 0.5)$ and radius 0.4.
- To solve the system to get the geodesic curves, we applied the classical Runge-Kutta method of 4th order.
- For the calculations of the error operator K , the regularization parameter is chosen as $\delta = 0.02$.
- In the layer stripping algorithm, we divide the domain into 20 layers and each layer has 122 local regions for reconstruction.
- For each local region, the size of the matrix $A^* A$ is about 100 x100.
- The whole domain has 35,000 unknowns, which requires the inversion of a 35000 x 35000 matrix if layer by layer method is not possible.

Constant case and the linear case

- To test our algorithm, we test it on different speeds. First, we test the constant case and the linear case.
- For the constant case $g = 10$, we have the relative error 0.0004% for first layer and 0.0005% for the second layer.
- For the linear case $g = 10 + 0.1 \times (x + y + z)$, we have the relative error 0.0727% for first layer and 0.0599% for the second layer.
- We also note that the first two layers are recovered almost exactly and the errors grow after a few layers. The fact that there are larger errors in inner layers is because there are less data available for those regions.

	1^{st} layer	2^{nd} layer	3^{rd} layer	4^{th} layer	5^{th} layer
relative error (constant)	0.0004%	0.0005%	0.1643%	2.5194%	12.8080%
relative error (linear)	0.0727%	0.0599%	0.3647%	2.6736%	14.2001%

Table: Relative errors for different cases.

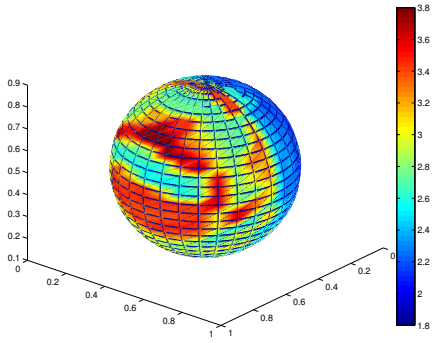
Marmousi model

- Next, we test the performance using the Marmousi model. We divide the 3D domain into 10 layers and we recover the model in the first few outermost layers.
- Then we have the relative error 8.2883% for first layer, 6.6484% for the second layer, 9.2633% for the third layer and 12.8978% for the forth layer.
- Figure 9-13, show the graphs of true and approximate solution of first,second and third layers of standard Marmousi model. We observe that the recovered solutions are in good agreement with the exact solutions.

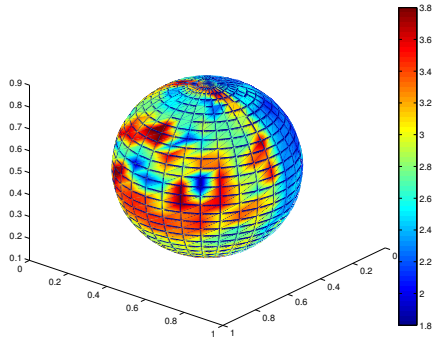
	1^{st} layer	2^{nd} layer	3^{rd} layer	4^{th} layer	5^{th} layer
relative error	8.2883%	6.6484%	9.2633%	12.8978%	13.2901%

Table: Relative errors for recovering the Marmousi model.

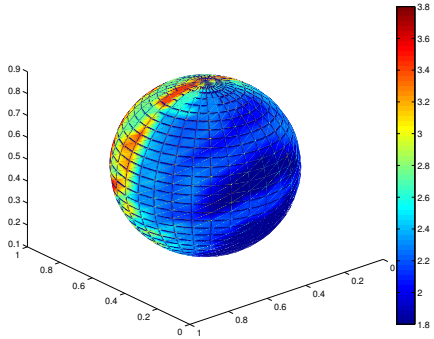
Marmousi model (cont.)



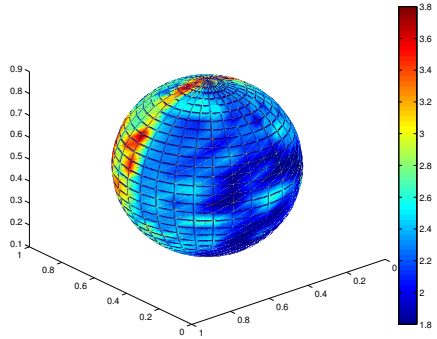
: exact solution for first layer (front)



: approx. solution for first layer (front)

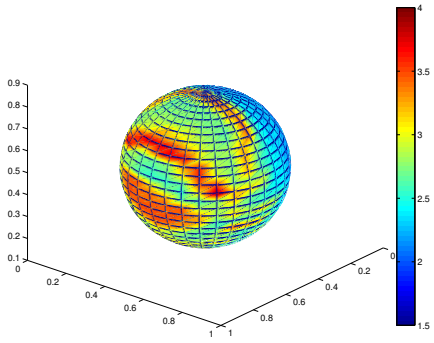


: exact solution for first layer (back)

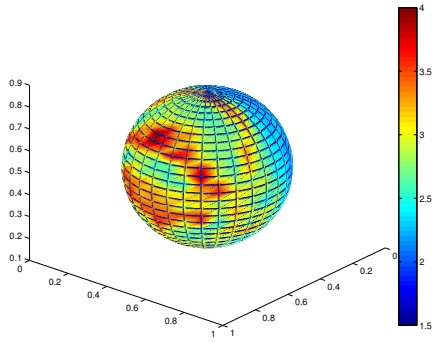


: approx. solution for first layer (back)

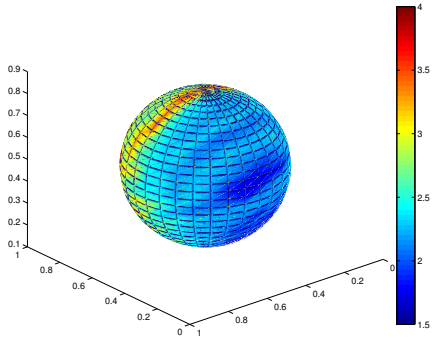
Marmousi model (cont.)



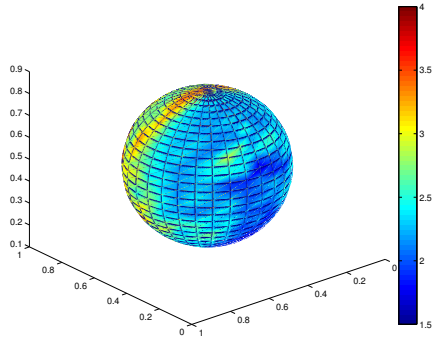
: exact solution for second layer (front)



: approx. solution for second layer (front)

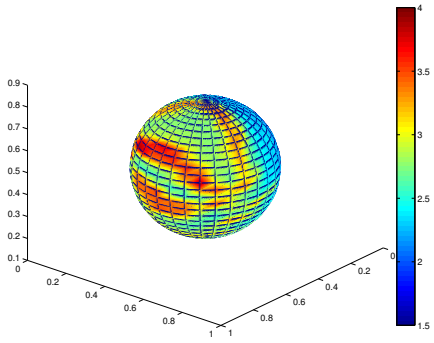


: exact solution for second layer (back)

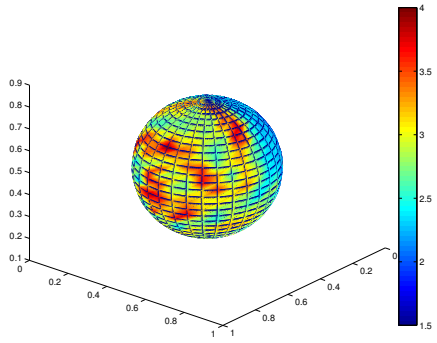


: approx. solution for second layer (back)

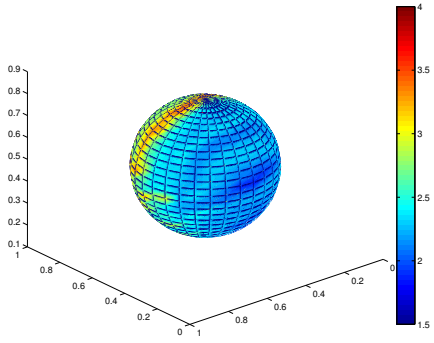
Marmousi model (cont.)



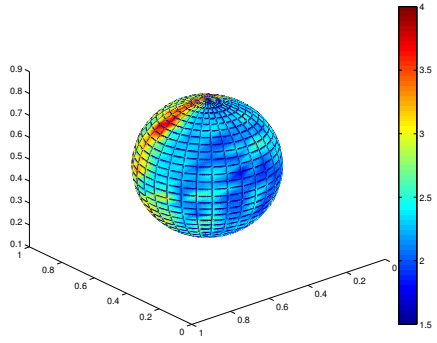
: exact solution for third layer (front)



: approx. solution for third layer (front)



: exact solution for third layer (back)



: approx. solution for third layer (back)

Conclusion

- We develop a numerical strategy for inversion of X-ray transform.
- The method is based on a convergent Neumann series and a layer-stripping techniqe.
- We develop an inverse algorithm for travel time tomography.
- The method is based on the inversion of X-ray transform and layer-stripping.
- We present some numerical results including the Marmousi model.

Thank you

References I

- Chung E, Qian J, Uhlmann G and Zhao H K 2007 A new phase space method for recovering index of refraction from travel times *Inverse Problems* 23 309-29
- Chung E, Qian J, Uhlmann G and Zhao H K 2008 A phase-space formulation for elastic-wave traveltime tomography *J. Phys.: Conf. Ser.* 124 012018
- B. Frigyik, P. Stefanov and G. Uhlmann 2008 The x-ray transform for a generic family of curves and weights *J. Geom. Anal.* 18 89-108
- K. Guo and D. Labate 2013 Optimal recovery of 3D X-ray tomographic data via sherbet decomposition *Advances in Computational Mathematics* 39(2) 227-255
- Y. Kurley, M. Lassas and G. Uhlmann 2010 Rigidity of broken geodesics flow and inverse problems *Am. J. Math.* 132 529-62
- S. Leung and J. Qian 2006 An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals *Commun. Math. Sci.* 4 249-66

References II

- S. Leung and J. Qian 2007 Transmission traveltime tomography based on paraxial Liouville equations and level set formulations. *Inverse Problems* 23(2) 799
- F. Monard 2014 Numerical implementation of two-dimensional geodesic X-ray transforms and their inversion *SIAM J. Imaging Sciences* 7 no.2 1335-1357
- L. Pestov and G. Uhlmann 2004 On characterization of the range and inversion formulas for the geodesic x-ray transform *Int. Math. Res. Not.* 80 4331-47
- L. Pestov and G. Uhlmann 2005 Two-dimensional compact simple Riemannian manifolds are boundary distance rigid *Ann. Math.* 161 1093-110
- J. Qian, P. Stefanov, G. Uhlmann, and H. Zhao 2011 An efficient neumann series-based algorithm for thermoacoustic and photoacoustic tomography with variable sound speed *SIAM J. Imaging Sciences* 4 850-83
- P. Stefanov and G. Uhlmann 1998 Rigidity for metrics with the same lengths of geodesics *Math. Res. Lett.* 5 83-96
- P. Stefanov and G. Uhlmann 2004 Stability estimates for the x-ray transform of tensor fields and boundary rigidity *Duke Math. J.* 123 445-67

References III

- P. Stefanov and G. Uhlmann 2005 Boundary rigidity and stability for generic simple metrics *J. Am. Math. Soc.* 18 975-1003
- P. Stefanov and G. Uhlmann 2005 Recent progress on the boundary rigidity problem *Electron. Res. Announc. Am. Math. Soc.* 11 64-70
- P. Stefanov and G. Uhlmann 2008 Boundary and lens rigidity, tensor tomography and analytic microlocal analysis *Algebraic Analysis of Differential Equations, Festschrift in Honor of Takahiro Kawai* edited by T. Aoki, H. Majima, Y. Katei and N. Tose 275-293
- G. Uhlmann and A. Vasy 2015 The inverse problem for the local geodesic ray transform *Invent. math.* DOI 10.1007/s00222-015-063