AN ABSTRACT OF THE THESIS OF

Brian Sherson for the degree of Doctor of Philosophy in Mathematics presented on December 10, 2015.

Title: Some Results in Single-Scattering Tomography

Abstract approved: _____

David V. Finch

Single-scattering tomography describes a model of photon transfer through a object in which photons are assumed to scatter at most once. The Broken Ray transform arises from this model, and was first investigated by Lucia Florescu, Vadim A. Markel, and John C. Schotland, [2], in 2010, followed by an inversion, [1], in the case of fixed initial and terminal directions. Later, in 2013, Katsevich and Krylov, [6], investigated settings where terminal rays were permitted to vary, either heading towards or away from a focal point, providing inversion formulas in two- and three-detector settings.

In this thesis, we will explore these transforms, give them distributional meaning, and analyze how these transforms propagate singularities. This requires analysis of the relationships between a function and its antiderivative obtained from integration over a ray.

This thesis also introduces the Polar Broken Ray transform, in which the source position is fixed, initial direction varies, and the scattering angle is held constant. We will discover that the Polar Broken Ray transform is injective on spaces of functions supported in an annulus that is bounded away from the origin, and also derive distributional meaning to the Polar Broken Ray transform, and derive a relationship between a wavefront set of a function and that of its Polar Broken Ray transform.

Provided in the appendix are implementations of numerical inversions of the Broken Ray transform with fixed initial and terminal directions, as well as the Polar Broken Ray transform.

Some Results in Single-Scattering Tomography

by

Brian Sherson

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 10, 2015
Commencement June 2016

Doctor of Philosophy thesis of Brian Sherson presented on December 10, 2015

APPROVED:

Major Professor, representing Mathematics

Chair of the Department of Mathematics

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Brian Sherson, Author

ACKNOWLEDGEMENTS

Academic

I am indebted to my adviser, Professor David V. Finch. Even before you took me on as your Ph.D. student, you were the one to notify me that I was offered admission into the graduate program. When you accepted me as your Ph.D. student, you told me that you believed I had something to contribute. Through both dead ends and breakthroughs in my research, you continued to offer encouragement.

I would like to thank Professors Ralph E. Showalter, Adel Faridani, Nathan L. Gibson, and Guenther Schneider, for their willingness to serve on my committee. I would also like to thank Professor Mary Flahive. You believed in me from the start. Early on, when I felt uncertain about my future as a graduate student, you put me on a path where I could succeed.

I would also like to thank the Mathematics Department of Oregon State University for the support given to me, both in the form of academic support, and for financial support in the form of teaching assistanceships during my time here.

The Mathematics Department at Central Washington University is also due my gratitude for providing me with both encouragement and financial support during my application process to graduate school. I would also like to thank Professor James Harper, who was a great mentor during my years as an undergraduate student, and greatly encouraged my personal mathematical research.

My journey in mathematics, however, began in elementary school. I would like to extend my thanks to Ms. Janis Laybourn. You recognized my mathematical abilities before I did, and saw to it that I was properly placed at a level that I would find both challenging

and rewarding. That decision put me on this path towards a Ph.D. While your decision can only be regarded as peripheral involvement in my pursuit of my degree, it was still nevertheless an important moment in my life. I am not sure how differently my focus on life would be, had you not put me on that path.

In my role as a teaching assistant, I regularly seeked advice regarding the teaching of mathematics from Scott L. Peterson. Thank you for your mentorship.

I would also like to thank National Science Foundation for providing partial support of my research under grant DMS 1009194.

Personal

I wish to thank my parents, Jerrold Lynn and Michele, for providing for me from childhood and through my adolescence. To this day, you have continued to offer the support I have needed to get through my graduate studies. I may not be able to pay back all my debt to you, but I can try to get a good start on doing so.

When I started graduate school, my hope was that I would have my grandparents Jerrold Kent, Geraldine, and Dorothy would see me finish my degree. Unfortunately, they have since passed, but I nevertheless would like to thank them for the support and inspiration I had received from them.

I would also like to thank Jay, Kayli, Erin, and Jordan for being a part of my life. You all have inspired me to pursue success, and I wish you all success as well.

My thanks are also given to Forrest, Ashley, Thomas, Hussain, Patricia, and Sarah, for offering friendship during my years at Oregon State University.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

SOME RESULTS IN SINGLE-SCATTERING TOMOGRAPHY

1 INTRODUCTION

1.1 Single-scattering tomography and the Broken Ray transform

In single-scattering tomography, we describe photon transfer through a object with a model in which photons are assumed to scatter at most once. Given an open set $U \subseteq \mathbb{R}^n$, we consider emitting a ballistic ray of photons with initial intensity I_0 *into* U at some point $s_0 \in \partial U$, in a direction $\vec{\mathbf{v}}_0 \in \mathcal{S}^{n-1}$ (so that $s_0 + t\vec{\mathbf{v}}_0 \in U$ for sufficiently small $t > 0$). Photons originating from this ballistic ray are then scattered inside U and exit along a terminal ray. We then measure the light intensity on ∂U. Moreover, we assume that we can selectively measure the intensity of photons exiting U in a single direction.

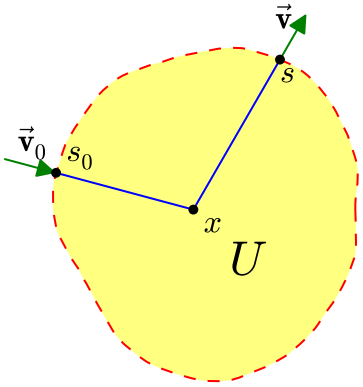

FIGURE 1.1: $\mathcal{B}\mu_{\mathrm{t}}(x; \vec{\mathbf{v}}_0, \vec{\mathbf{v}})$ is the line integral of μ_{t} over the solid blue line.

The Broken Ray transform was first investigated by Lucia Florescu, Vadim A. Markel, and John C. Schotland, [2], in 2009. In the setting of $U = (0, L) \times \mathbb{R} \subseteq \mathbb{R}^2$, if μ_{a} and μ_{s} are absorption and scattering coefficients, respectively, and μ_{t} is the sum of these two, then the measured intensity of photons leaving U at a point $s \in \partial U$ in the direction $\vec{\mathbf{v}}$ is given by

$$I(s, \vec{\mathbf{v}}) = I_0 C(s_0, \vec{\mathbf{v}}_0, s, \vec{\mathbf{v}}) \mu_{\mathrm{s}}(x) A(\vec{\mathbf{v}}_0, \vec{\mathbf{v}}) \exp\left(-\mathcal{B}\mu_{\mathrm{t}}(x; \vec{\mathbf{v}}_0, \vec{\mathbf{v}})\right), \tag{1.1.1}$$

where $x \in U$ is the location of scattering, and hence is the intersection of the ballistic and terminal rays, A is the phase function used in the Radiative Transport Equation, C is a function depending only on the choice of ballistic and terminal rays, and

$$\begin{aligned} \mathcal{B}\mu_{\mathrm{t}}\left(x;\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2\right) &= \int_0^{t_1} \mu_{\mathrm{t}}\left(s_1+t\vec{\mathbf{v}}_1\right)\,dt + \int_0^{t_2} \mu_{\mathrm{t}}\left(s_2-t\vec{\mathbf{v}}_2\right)\,dt \\ &= \int_0^{t_1} \mu_{\mathrm{t}}\left(x-t\vec{\mathbf{v}}_1\right)\,dt + \int_0^{t_2} \mu_{\mathrm{t}}\left(x+t\vec{\mathbf{v}}_2\right)\,dt, \end{aligned} \tag{1.1.2}$$

is the Broken Ray transform of μ_{t}, where $x = s_1 + t_1\vec{\mathbf{v}}_1 = s_2 - t_2\vec{\mathbf{v}}_2 \in G$, $s_1, s_2 \in \partial G$, and the line segments $\ell_1 = \{s_1 + t\vec{\mathbf{v}}_1 \,|\, 0 < t \leq t_1\}$ and $\ell_2 = \{s_2 - t\vec{\mathbf{v}}_1 \,|\, 0 < t \leq t_2\}$ are both contained in G. As presented above, $\mathcal{B}\mu_{\mathrm{t}}$ is defined on a manifold of dimension $3n-2$. We will focus on restrictions of $\mathcal{B}\mu_{\mathrm{t}}$ to submanifolds of dimension n.

In 2010, Florescu, Markel, and Schotland, [1], further investigated the Broken Ray transform, holding as constants $\vec{\mathbf{v}}_1 = (1,0)$ and $\vec{\mathbf{v}}_2 = (\cos\theta, \sin\theta)$ for some fixed angle $\theta \in \left(0, \frac{\pi}{2}\right)$, using the locations of ballistic and terminal rays as independent variables. In this setting, $\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}\mu_{\mathrm{t}} \stackrel{\mathrm{def}}{=} \mathcal{B}\mu_{\mathrm{t}}\left(\cdot;\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2\right)$ is found to be invertible. If μ_{s} is known, then recovery of μ_{t} is possible from one set of data. On the other hand, if μ_{s} is not known, recovery of μ_{t} and μ_{s} is possible from two sets of data, both with fixed initial and terminal directions.

In 2013, Rim Gouia-Zarrad and Gaik Ambartsoumian, [3], revisit the Broken Ray transform, though referred to as the V-line transform, in which they parametrize with $\vec{\mathbf{v}}_1 = (\cos\beta, -\sin\beta)$ and $\vec{\mathbf{v}}_2 = (\cos\beta, \sin\beta)$. In addition, Gouia-Zarrad and Ambartsoumian introduce a 3D Conical Radon transform in which the total attenuation coefficient, f, is integrated over cones with a fixed opening angle, defining

$$g\left(x_{\mathrm{v}}, y_{\mathrm{v}}, z_{\mathrm{v}}\right) = \int_{C(x_{\mathrm{v}},y_{\mathrm{v}},z_{\mathrm{v}})} f\left(x,y,z\right)\,ds,$$

where

$$C\left(x_{\mathrm{v}}, y_{\mathrm{v}}, z_{\mathrm{v}}\right) = \left\{ \left(x_{\mathrm{v}}+x, y_{\mathrm{v}}+y, z_{\mathrm{v}} + \sqrt{x^2+y^2}\tan\beta \right) \,\middle|\, x,y \in \mathbb{R} \right\}.$$

The paper by Katsevich and Krylov, [6], goes beyond the case of fixed initial and terminal directions, and includes settings which employ convex and concave detectors, in the shape of circular arcs, in addition to flat detectors. With concave detectors, only terminal rays in a direction away from a focal point are measured, whereas convex detectors detect terminal rays in a direction towards a focal point. Furthermore, instead of reconstructing from one set of data, Katsevich and Krylov provides reconstructions from two-detector and three-detector settings in terms of differences between measurements, in effect, cancelling out integration over the ballistic rays.

Hence, Katesvich and Krylov define

$$g_{ij}(x) = \int_0^\infty f\left(x + t\vec{\beta}_i(x)\right) dt - \int_0^\infty f\left(x + t\vec{\beta}_j(x)\right) dt,\ i \neq j, \tag{1.1.3}$$

for $f \in \mathcal{C}_0^\infty(U)$ for some open set $U \subseteq \mathbb{R}^2$, $i, j = 1, 2$ or $1, 2, 3$, where each of the $\vec{\beta}_i \in \mathcal{C}^1\left(U; \mathcal{S}^1\right)$ is one of the following:

$$\vec{\beta}_i(x) = \begin{cases} \vec{\mathbf{v}}_i, & \text{if detector } i \text{ is flat, detecting rays in direction } \vec{\mathbf{v}}_i, \\ \frac{x - x_i}{\|x - x_i\|}, & \text{if detector } i \text{ is concave, with focal point at } x_i, \\ -\frac{x - x_i}{\|x - x_i\|}, & \text{if detector } i \text{ is convex, with focal point at } x_i. \end{cases}$$

In the convex case, we must consider a focus x_i placed outside the convex hull of U, unless we replace the upper limit on the corresponding integral with $\|x - x_i\|$. Katsevich and Krylov go on to characterize a solution to (1.1.3) in terms of a first-order differential equation.

We can observe that (1.1.1) also allows us to derive g_{ij} directly from the measured data. Taking multiple measurements simultaneously, we have

$$I_j\left(s_j, \vec{\beta}_j\right) = C\left(s_0, \vec{\mathbf{v}}_0, s_j, \vec{\beta}_j\right) I_0 \mu_{\mathrm{s}}(x)\, A\left(\vec{\mathbf{v}}_0, \vec{\beta}_j\right) \exp\left(-\mathcal{B}\mu_{\mathrm{t}}\left(x; \vec{\mathbf{v}}_0, \vec{\beta}_j\right)\right)$$

for $j = 1, 2, 3$, where $s_j = x + t_j \vec{\beta}_j$ for some $t_j \geq 0$, and the dependence of $\vec{\beta}_j$ on x is implied, but omitted from notation. Division of I_i by I_j for $i \neq j$ yields

$$\begin{aligned}\frac{I_j\left(s_j, \vec{\beta}_j\right)}{I_i\left(s_i, \vec{\beta}_i\right)} &= \frac{C\left(s_0, \vec{\mathbf{v}}_0, s_j, \vec{\beta}_j\right) A\left(\vec{\mathbf{v}}_0, \vec{\beta}_j\right)}{C\left(s_0, \vec{\mathbf{v}}_0, s_i, \vec{\beta}_i\right) A\left(\vec{\mathbf{v}}_0, \vec{\beta}_i\right)} \\ &\qquad \cdot \exp\left(-\mathcal{B}\mu_{\mathrm{t}}\left(x; \vec{\mathbf{v}}_0, \vec{\beta}_j\right) + \mathcal{B}\mu_{\mathrm{t}}\left(x; \vec{\mathbf{v}}_0, \vec{\beta}_i\right)\right) \\ &= \frac{C\left(s_0, \vec{\mathbf{v}}_0, s_j, \vec{\beta}_j\right) A\left(\vec{\mathbf{v}}_0, \vec{\beta}_j\right)}{C\left(s_0, \vec{\mathbf{v}}_0, s_i, \vec{\beta}_i\right) A\left(\vec{\mathbf{v}}_0, \vec{\beta}_i\right)} \exp\left(g_{ij}\left(x\right)\right),\end{aligned}$$

and so

$$g_{ij}\left(x\right) = \ln \frac{I_j\left(s_j, \vec{\beta}_j\right)}{I_i\left(s_i, \vec{\beta}_i\right)} - \ln \frac{C\left(s_0, \vec{\mathbf{v}}_0, s_j, \vec{\beta}_j\right) A\left(\vec{\mathbf{v}}_0, \vec{\beta}_j\right)}{C\left(s_0, \vec{\mathbf{v}}_0, s_i, \vec{\beta}_i\right) A\left(\vec{\mathbf{v}}_0, \vec{\beta}_i\right)}.$$

In the three-detector case, Katesvich and Krylov also find an inversion given by a local operator, and characterize the range.

1.2 The Polar Broken Ray transform

Another problem of interest comes from allowing the initial direction of a light beam to vary instead of the initial position, while keeping the initial position fixed. This problem is modelled by what we will call the Polar Broken Ray transform, and places the emitter at the origin in $\mathbb{R}^2$. We parametrize a ray with $s\vec{\sigma}$ being the location of the single scattering event, where $\vec{\sigma} \in \mathcal{S}^1$ is the initial direction of the beam, and $s > 0$ the distance a beam travels before a scattering event. We thus introduce the Polar Broken Ray transform as follows:

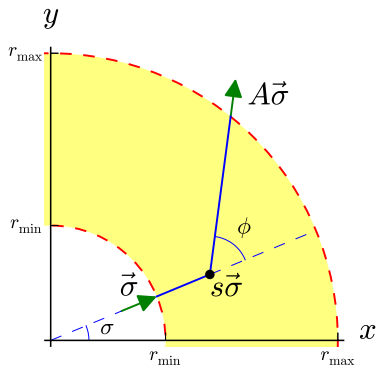

FIGURE 1.2: $\mathcal{Q}f\left(s\vec{\sigma}\right)$ is the line integral over the solid blue line.

Definition 1.2.1 (Polar Broken Ray transform)**.** Let

$$\Omega = \left\{ r\vec{\theta} \,\middle|\, r_{\min} \leq r \leq r_{\max}, \vec{\theta} \in \mathcal{S}^1 \right\},$$

$$\hat{\Omega} = \left\{ s\vec{\sigma} \,\middle|\, 0 < s \leq r_{\max}, \vec{\sigma} \in \mathcal{S}^1 \right\},$$

for some fixed $r_{\max} > r_{\min} \geq 0$. Furthermore, let A be the following rotation matrix:

$$A = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix},$$

for some fixed parameter $\phi \in \left(0, \frac{\pi}{2}\right)$. We then define the Polar Broken Ray transform in terms of polar arguments as follows:

$$\mathcal{Q}f\left(s\vec{\sigma}\right) = \int_0^s f\left(t\vec{\sigma}\right)\, dt + \int_0^{\sqrt{r_{\max}^2 - s^2\sin^2\phi} - s\cos\phi} f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt, \quad s\vec{\sigma} \in \hat{\Omega}$$

for functions f supported Ω. For convenience, if we define f to be zero outside Ω, then we can express this transform as

$$\mathcal{Q}f\left(s\vec{\sigma}\right) = \int_0^s f\left(t\vec{\sigma}\right)\, dt + \int_0^{\infty} f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt. \tag{1.2.1}$$

A quick observation is that $\mathcal{Q}$ commutes with rotations about the origin.

1.3 Propagation of singularities

The theory of wavefront sets is due to Lars Hörmander, [5], and extends the notion of singular support of a distribution – points at which a distribution is not given by a $\mathcal{C}^\infty$ function in *any* open neighborhood. The wavefront set of a distribution u is denoted by $WF\left(u\right)$, and describes not only the locations of singularities of u, but also the directions of those singularities. For instance, if u is given by a piecewise-$\mathcal{C}^\infty$ function, then $WF\left(u\right)$

will contain a pair $\left(x, \vec{\xi}\right)$ if, at a point x, $\vec{\xi}$ is a normal vector to a surface across which u (or one of its derivatives) is discontinuous.

It is through the use of wavefront sets that we describe propagation of singularities results for an operator on a space of distributions. Moreover, propagation of singularities results can account for some of the artifacts seen in reconstructed images.

1.4 Summary of results

We will define the Broken Ray transform and its variations in a distributional sense. The motivation for doing so is due the inversion formulas requiring some order of differentiability of the function we wish to recover in order to be valid in a classical sense. However, in practice, we may wish to recover a function that is not continuous. Describing these transforms and their inversion formulas in a distributional sense would overcome this technical deficit.

We will then develop propagation of singularities results for each of these variations of the Broken Ray transform. This follows Hörmander's lead of developing the theory of wavefront sets in the context of distributions.

In Chapter 3, we will give a distributional meaning to the directional antiderivative operator

$$\mathcal{J}_{\vec{\mathbf{v}}} u\left(x\right) = \int_0^\infty u\left(x - t\vec{\mathbf{v}}\right), \quad u \in \mathcal{C}_0^0\left(\mathbb{R}^n\right), x \in \mathbb{R}^n, \tag{1.4.1}$$

where $\vec{\mathbf{v}} \neq 0$ is a fixed parameter. This requires that we introduce the distribution

$$J_{\vec{\mathbf{v}}}\left(\phi\right) = \int_0^\infty \phi\left(t\vec{\mathbf{v}}\right)\, dt, \quad \phi \in \mathcal{C}_0^\infty\left(\mathbb{R}^n\right), \tag{1.4.2}$$

which is a fundamental solution to the distributional derivative operator $\mathcal{D}_{\vec{\mathbf{v}}}$. We then define $\mathcal{J}_{\vec{\mathbf{v}}}u = J_{\vec{\mathbf{v}}} \star u$, for $u \in \mathscr{D}'(\mathbb{R}^n)$ with appropriate assumptions on the support of u, to be determined later, that allows for the convolution to be defined.

Our primary goal is to establish that for distributions u for which $\mathcal{J}_{\vec{\mathbf{v}}}u$ is defined,

$$WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \subseteq WF(u) \cup \left\{ (x + t\vec{\mathbf{v}}, \vec{\eta}) \,\middle|\, (x, \vec{\eta}) \in WF(u), \vec{\eta} \in \vec{\mathbf{v}}^{\perp}, t \geq 0 \right\}. \tag{1.4.3}$$

We will then generalize this result to an operator $\mathcal{J}_{\gamma,\upsilon}$ defined by

$$\mathcal{J}_{\gamma,\upsilon}u(x) = \int_0^{\infty} u(x - \gamma(t))\, \upsilon(t)\, dt,$$

for some $\gamma \in \mathcal{C}^{\infty}([0,\infty); \mathbb{R}^n)$ and $\upsilon \in \mathcal{C}^{\infty}([0,\infty); \mathbb{R}^+)$ so that $\|\gamma(t)\| \to \infty$ as $t \to \infty$. We extend $\mathcal{J}_{\gamma,\upsilon}$ in a distributional sense by defining it as convolution with

$$J_{\gamma,\upsilon}(\phi) = \int_0^{\infty} \phi(\gamma(t))\, \upsilon(t)\, dt, \quad \phi \in \mathcal{C}_0^{\infty}(\mathbb{R}^n), \tag{1.4.4}$$

when such a convolution is possible, and then show that

$$WF(\mathcal{J}_{\gamma,\upsilon}u) \subseteq WF(u) \cup \left\{ \left(x, \vec{\xi}\right) \,\middle|\, \exists t \geq 0 : \left(x - \gamma(t), \vec{\xi}\right) \in WF(u), \vec{\xi} \in \gamma'(t)^{\perp} \right\}.$$

In Chapter 4, we will verify an inversion formula for the Broken Ray transform with fixed initial and terminal directions. We then provide two alternate inversion formulas, one of which is symmetric in nature, and is used as a basis for a numerical inversion.

We also investigate the various cases of Katsevich and Krylov's two-detector setting, discovering that if the detectors either both convex or both concave, then the mapping $f \mapsto g_{ij}$ is not injective, and hence, recovery of f is not possible in the absence of an a priori support restriction or known boundary data.

We also give a distributional meaning to $\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}$ and g_{ij}, as well as use results from Chapter 3 to establish the relationships between distributions and their Broken Ray transforms.

Furthermore, we discover support conditions necessary to give distributional meaning to the inversion formulas in Katsevich and Krylov's setting.

In Chapter 5, we will find the Polar Broken Ray transform to be injective for continuous functions with compact support in $\mathbb{R}^2\backslash 0$, with an inversion given as an infinite series. We will also define the transform in a distributional sense and derive a relationship between the wavefront set of a distribution and that of its Polar Broken Ray transform. However, our wavefront set result will be one-sided, since the solution is given as an infinite series.

1.5 Summary of notations

We will make use of the idea of the vector difference of two sets, analogous to the vector sum of sets. As such, we will adopt the notation

$$S - T = \{u - v \,|\, u \in S, v \in T\}, \quad S, T \subseteq \mathbb{R}^n,$$

and use the notation $S\backslash T$ to refer to the usual notion of set difference. Furthermore, Cartesian products will take precedence over set difference, unions, and intersections.

We also use the notation for lines, rays, and line segments and in $\mathbb{R}^n$:

$$x + I\vec{\mathbf{v}} = \{x + t\vec{\mathbf{v}} \,|\, t \in I\},$$

for $x \in \mathbb{R}^n$, $\vec{\mathbf{v}} \in \mathcal{S}^{n-1}$, and $I \subseteq \mathbb{R}$, where I is an interval, whether open, closed, or half-open. We may also take I to be half-bounded to denote a ray, or all of $\mathbb{R}$ for an entire line.

In a similar manner, we may refer to translates of subsets $U \subseteq \mathbb{R}^n$ by a vector $\vec{\mathbf{v}}$ as follows:

$$U + \vec{\mathbf{v}} = \{x + \vec{\mathbf{v}} \,|\, x \in U\}.$$

We denote $\mathcal{C}^0(X;Y)$ as the set of continuous functions defined on some open subset X of a Banach space with values in Y.

In addition to the usual Leibniz notation for derivatives, we will make other use of derivative notations. Given Banach spaces V and W, an open subset $X \subseteq V$, $\phi : V \to W$, and $x \in X$, we take $\mathcal{D}\phi(x)$ as the Fréchet derivative of ϕ at x, and is defined as the continuous linear operator from V to W satisfying

$$\lim_{\vec{\mathbf{v}} \to \vec{\mathbf{0}}} \frac{\|\phi(x+\vec{\mathbf{v}}) - \phi(x) - \mathcal{D}\phi(x)\,\vec{\mathbf{v}}\|}{\|\vec{\mathbf{v}}\|} = 0,$$

provided the limit exists, in which case we say ϕ is differentiable at x. We define $\mathcal{C}^1(X;Y)$ as the set of functions ϕ that are differentiable everywhere, with values in $Y \subseteq W$, and moreover, $\mathcal{D}\phi \in \mathcal{C}^0(X;\mathcal{L}(V,W))$. We denote the higher order Fréchet derivatives as $\mathcal{D}^k\phi(x)$, and are k-linear operators. Via recursion, we denote $\mathcal{C}^k(X;Y)$ to be the set of those $\phi \in \mathcal{C}^{k-1}(X;Y)$ for which $\mathcal{D}^{k-1}\phi$ is itself $\mathcal{C}^1$.

If $D \subseteq \mathbb{R}^n$ is closed, then $\mathcal{C}^k(D;Y)$ is the set of those functions defined on D that can be extended to a function in $\mathcal{C}^k(X;Y)$ for some open neighborhood X of D.

We then denote the set of smooth functions with

$$\mathcal{C}^\infty(X;Y) = \bigcap_{k=0}^{\infty} \mathcal{C}^k(X;Y).$$

We then denote $\mathcal{C}_0^k(X;Y)$ as the subset of $\mathcal{C}^k(X;Y)$ of functions with compact support, for $0 \leq k \leq \infty$. When Y is omitted from the notation, it shall be understood that $Y = \mathbb{C}$. In contrast to $\mathcal{C}^k(D;Y)$, however, $\mathcal{C}_0^k(D;Y)$ shall denote those functions in $\mathcal{C}^k(D;Y)$ whose support is compact *and bounded away* from ∂D.

We will consider multiple notions of differentiation in a direction $\vec{\mathbf{0}} \neq \vec{\mathbf{v}} \in V$. For $\phi \in \mathcal{C}^1(X;Y)$, we will define $\mathcal{D}_{\vec{\mathbf{v}}}\phi(x) = \mathcal{D}\phi(x)\,\vec{\mathbf{v}}$, which defines $\mathcal{D}_{\vec{\mathbf{v}}}$ as an operator from $\mathcal{C}^1(X;Y)$ to $\mathcal{C}^0(X;Y)$. In the setting that V is a subspace of $\mathbb{R}^n$, we will also speak

weak derivatives, in which we say $\mathcal{D}_{\vec{\mathbf{v}}} f \in \mathcal{L}^1_{\text{loc}}(X)$ is a weak derivative of $f \in \mathcal{L}^1_{\text{loc}}(X)$ whenever

$$\int_X \mathcal{D}_{\vec{\mathbf{v}}} f(x)\, \phi(x)\, dx = -\int_X f(x)\, \mathcal{D}_{\vec{\mathbf{v}}} \phi(x)\, dx, \quad \phi \in \mathcal{C}_0^\infty(X).$$

Here, integration is done with respect to surface measure if V is a proper subspace.

We denote $\mathscr{D}'(X)$ as the space of distributions on X. When $u \in \mathscr{D}'(X)$, $\mathcal{D}_{\vec{\mathbf{v}}} u$ is the distributional derivative of u in the direction $\vec{\mathbf{v}}$, and is defined by $\mathcal{D}_{\vec{\mathbf{v}}} u(\phi) = -u(\mathcal{D}_{\vec{\mathbf{v}}} \phi)$, for $\phi \in \mathcal{C}_0^\infty$.

With functions defined on or taking values in product spaces (or perhaps both), we will generalize the notion of a Jacobian matrix by writing

$$\mathcal{D}\phi = \begin{bmatrix} \partial_{V_1}\phi_1 & \partial_{V_2}\phi_1 & \dots & \partial_{V_M}\phi_1 \\ \partial_{V_1}\phi_2 & \partial_{V_2}\phi_2 & \dots & \partial_{V_M}\phi_2 \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{V_1}\phi_N & \partial_{V_2}\phi_N & \dots & \partial_{V_M}\phi_N \end{bmatrix}$$

to denote the derivative of a function $\phi = (\phi_1, \phi_2, \dots, \phi_N) \in \mathcal{C}^1(U; W)$ for U an open subset of $V = V_1 \times V_2 \times \cdots \times V_M$, and $W = W_1 \times W_2 \times \cdots \times W_N$. The entries, $\partial_{V_j}\phi_k$, denote partial Fréchet derivatives, and at each point, are continuous linear operators from V_j to W_k satisfying

$$\lim_{V_j \ni \vec{\mathbf{v}} \to \vec{\mathbf{0}}} \frac{\left\| \phi_k(x + \vec{\mathbf{v}}) - \phi_k(x) - \partial_{V_j}\phi_k(x)\, \vec{\mathbf{v}} \right\|}{\|\vec{\mathbf{v}}\|} = 0.$$

We will also use multiindex notation that is used by Hörmander, [5]. A multiindex $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ is an n-tuple of nonnegative integers, used in two settings. For $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, we define x^α by

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}.$$

Given the standard ordered basis $\{\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2, \ldots, \vec{\mathbf{e}}_n\}$ for $\mathbb{R}^n$, we define

$$\partial^\alpha f = \left(\mathcal{D}_{\vec{\mathbf{e}}_1}\right)^{\alpha_1} \left(\mathcal{D}_{\vec{\mathbf{e}}_2}\right)^{\alpha_2} \cdots \left(\mathcal{D}_{\vec{\mathbf{e}}_n}\right)^{\alpha_n} f, \quad |\alpha| = \sum_{k=1}^{n} \alpha_k,$$

when such derivatives exist. This notation may also be used in the context of weak derivatives of functions f defined on an open subset of $\mathbb{R}^n$, or in a distributional sense. Unless context requires otherwise, $\partial^\alpha f$ will be assumed to be a strong derivative.

While distributions are formally functions of test functions, we will make use of both notations $u(\phi)$ and $u(\Phi(x))$, relying on context to differentiate between meanings. For $u \in \mathscr{D}'(X)$ and $\phi \in \mathcal{C}_0^\infty(X)$, $u(\phi)$ will refer to evaluation of u at the test function ϕ, whereas if $u \in \mathscr{D}'(Y)$, for some open $Y \subseteq \mathbb{R}^m$ and $\Phi \in \mathcal{C}^\infty(X; Y)$, then $u(\Phi(x))$ will refer to the pullback $\Phi^\star u \in \mathscr{D}'(X)$ (given by the composition $u \circ \Phi$, if $u \in \mathcal{C}^\infty(Y)$), whenever such a pullback is possible. If Φ is a surjective local diffeomorphism, we may use $\Phi^{-\star} w$ to refer to a distribution u for which $w = \Phi^\star u$, if such a distribution exists.

If $u \in \mathscr{D}'(X)$ and ϕ is a function defined on a product space $X \times Y$ such that $\phi(\cdot, y)$ is a test function for each $y \in Y$, then $u_x(\phi(x, y))$ is interpreted as u acting on a test function $\phi(\cdot, y)$, where y is held as a fixed parameter, and hence, $u_x(\phi(x, y))$ itself is a function of y.

Much as we may use the notation $u(\Phi(x))$ to denote the pullback of a distribution u by a smooth function Φ, we will also use the more familiar notations

$$\int_0^\infty u(x - t\vec{\mathbf{v}})\, dt, \quad \int_0^\infty u(x - \gamma(t))\, \upsilon(t)\, dt$$

to refer convolution of u with distributions $J_{\vec{\mathbf{v}}}$, (1.4.2), and $J_{\gamma,\upsilon}$, (1.4.4), respectively.

2 PRELIMINARIES

2.1 Volterra integral equation of the second kind

We begin with the treatment of the Volterra Integral Equation as given in Tricomi, [8], and derive some variations. Given $h > 0$, $U = \{(x, y) \,|\, 0 \leq y \leq x \leq h\}$, $K \in \mathcal{L}^2(U)$, $\lambda \neq 0$, and $f \in \mathcal{L}^2([0, h])$, the problem of the Volterra integral equation of the second kind is to find $\phi \in \mathcal{L}^2([0, h])$ such that

$$\phi(x) - \lambda \int_0^x K(x, y)\, \phi(y)\, dy = f(x), \quad 0 \leq x \leq h. \tag{2.1.1}$$

Indeed, a solution exists, and is given by

$$\phi(x) = f(x) - \lambda \int_0^x H(x, y; \lambda)\, f(y)\, dy,$$

where $H(x, y; \lambda)$ is called the **resolvant kernel**, and is given by

$$-H(x, y; \lambda) = \sum_{\nu=0}^{\infty} \lambda^{\nu} K_{\nu+1}(x, y),$$

where K_{ν} is defined recursively by $K_1 = K$, and

$$K_{\nu+1}(x, y) = \int_y^x K_{\nu}(x, u)\, K(u, y)\, du.$$

If we multiply by an arbitrary nonzero constant c and make the substitutions $x = h - s$, $y = h - t$, (2.1.1) becomes

$$c \cdot \phi(h - s) - \lambda c \cdot \int_s^h K(h - s, h - t)\, \phi(y - t)\, dt = c \cdot f(h - s), \quad 0 \leq s \leq h,$$

and if we let $\psi(s) = \phi(x)$, $M_{\nu}(s, t) = c^{\nu} K_{\nu}(x, y)$, $G(s, t) = H(x, y)$, and $g(s) = c \cdot f(x)$, then we obtain a variation on the Volterra integral equation:

$$c \cdot \psi(s) - \lambda \int_s^h M(s, t)\, \psi(t)\, dt = g(s), \quad 0 \leq s \leq h, \tag{2.1.2}$$

and the solution ψ is given by

$$\begin{aligned}\psi(s) &= \phi(h-s) \\ &= f(h-s) - \lambda \int_0^{h-s} H(h-s, y; \lambda) f(y)\, dy \\ &= f(h-s) - \lambda \int_s^h H(h-s, h-t; \lambda) f(h-t)\, dt \\ &= \frac{1}{c} g(s) - \frac{\lambda}{c} \int_s^h G(s,t;\lambda) g(t)\, dt, \quad 0 \le s \le h, \end{aligned} \tag{2.1.3}$$

with

$$\begin{aligned}-G(s,t;\lambda) &= \sum_{\nu=0}^{\infty} \lambda^{\nu} K_{\nu+1}(h-s, h-t) \\ &= \sum_{\nu=0}^{\infty} \frac{\lambda^{\nu}}{c^{\nu+1}} M_{\nu+1}(s,t). \end{aligned}$$

Furthermore, observe that

$$\begin{aligned}M_{\nu+1}(s,t) &= c^{\nu+1} K_{\nu+1}(h-s, h-t) \\ &= c^{\nu+1} \int_{h-t}^{h-s} K_{\nu}(h-s, u) K(u, h-t)\, du \\ &= c^{\nu} c \int_s^t K_{\nu}(h-s, h-v) K(h-v, h-t)\, dv \\ &= \int_s^t M_{\nu}(s,v) M(v,t)\, dv. \end{aligned}$$

This leads to the following result:

Proposition 2.1.1. *If $g \in \mathcal{L}^2(\mathbb{R}^+)$ is supported in $[0,h]$, $M \in \mathcal{L}^2(\mathbb{R}^+)$ is supported in $[0,1]$, and c and λ are nonzero constants, then there is a solution $\psi \in \mathcal{L}^2_{\mathrm{loc}}(\mathbb{R}^+)$, also supported in $[0,h]$, to the equation*

$$c \cdot \psi - \lambda \cdot M \star \psi = g, \tag{2.1.4}$$

which is given by

$$\psi = \frac{1}{c}\left(g + \sum_{\nu=1}^{\infty} \left(\frac{\lambda}{c}\right)^{\nu} M^{\star \nu} \star g \right), \tag{2.1.5}$$

where $M \star \psi$ here is understood as the convolution with respect to dilation, i.e.,

$$M \star \psi(s) = \int_0^\infty M\left(\frac{s}{t}\right) \psi(t) \, \frac{dt}{t},$$

and $M^{\star\nu}$ denotes iterated convolution.

Proof. Let $h > 0$. Since M is supported in $[0,1]$, we can consider the following Volterra Equation:

$$c \cdot \psi(s) - \lambda \int_s^h M\left(\frac{s}{t}\right) \psi(t) \, \frac{dt}{t} = g(s), \quad 0 \le s \le h, \tag{2.1.6}$$

which we know will have a solution provided that

$$\int_0^\infty \int_0^\infty \left| \frac{1}{t} M\left(\frac{s}{t}\right) \right|^2 ds\, dt < \infty.$$

Indeed:

$$\begin{aligned} \int_0^h \int_0^h \left| \frac{1}{t} M\left(\frac{s}{t}\right) \right|^2 ds\, dt &= \int_0^h \int_0^t \left| \frac{1}{t} M\left(\frac{s}{t}\right) \right|^2 ds\, dt \\ &= \int_0^h \int_0^1 |M(\sigma)|^2 \, d\sigma\, dt \\ &= h \, \|M\|^2_{\mathcal{L}^2(\mathbb{R}^+)}. \end{aligned}$$

Thus, a solution is given by

$$\psi(s) = \frac{1}{c} g(s) - \frac{\lambda}{c} \int_s^h G(s,t;\lambda)\, g(t) \, dt, \quad 0 \le s \le h,$$

where

$$-G(s,t;\lambda) = \sum_{\nu=0}^\infty \frac{\lambda^\nu}{c^{\nu+1}} M_{\nu+1}(s,t),$$

and M_ν is defined recursively by

$$M_1(s,t) = \frac{1}{t} M\left(\frac{s}{t}\right), \quad M_{\nu+1}(s,t) = \int_s^t M_\nu(s,v) \cdot \frac{1}{t} M\left(\frac{v}{t}\right) dv.$$

Defining $\psi(s) = 0$ for $s > h$, and recalling that M is supported in $[0,1]$, (2.1.6) is in fact the same equation as (2.1.4). Furthermore,

$$M_\nu(s,t) = \frac{1}{t} M^{\star\nu}\left(\frac{s}{t}\right).$$

Hence,

$$\begin{aligned}\psi(s) &= \frac{1}{c}g(s) + \frac{\lambda}{c}\int_s^h \sum_{\nu=0}^{\infty} \frac{\lambda^\nu}{c^{\nu+1}} \cdot \frac{1}{t} M^{\star(\nu+1)}\left(\frac{s}{t}\right) g(t)\, dt \\ &= \frac{1}{c}g(s) + \frac{\lambda}{c}\sum_{\nu=0}^{\infty} \frac{\lambda^\nu}{c^{\nu+1}} \cdot M^{\star(\nu+1)} \star g(s) \\ &= \frac{1}{c}\left(g(s) + \sum_{\nu=1}^{\infty}\left(\frac{\lambda}{c}\right)^\nu \cdot M^{\star\nu} \star g(s)\right), \quad 0 \le s \le h. \end{aligned}$$ □

Another consideration we may wish to look at is the possibility that M itself might not be in $\mathcal{L}^2(U)$, but perhaps its restriction to

$$U_\varepsilon = \{(s,t) \,|\, \varepsilon \le s \le t \le h\}$$

is in $\mathcal{L}^2(U_\varepsilon)$ for every $\varepsilon \in (0,h)$. Then we can view (2.1.2) as being equivalent to solving

$$c \cdot \psi(s) - \lambda \int_s^h M(s,t)\,\psi(t)\, dt = g(s), \quad \varepsilon \le s \le h, \tag{2.1.7}$$

which, by translation, is equivalent to solving

$$c \cdot \psi(s+\varepsilon) - \lambda \int_s^{h-\varepsilon} M(s+\varepsilon, t+\varepsilon)\,\psi(t+\varepsilon)\, dt = g(s+\varepsilon), \quad 0 \le s \le h-\varepsilon, \tag{2.1.8}$$

for $\psi(s+\varepsilon)$, for every $\varepsilon > 0$.

2.2 Distributions

In this section, we shall take $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$, both open.

The theory of distributions is due to Laurent Schwartz [7]. The presentation we give here follows that of Lars Hörmander [5]. The idea of distributions is to generalize the notion of functions to permit analysis that is not possible in the classical setting. For instance, differential operators are not defined on $\mathcal{L}^p(X)$, yet $\mathcal{C}^\infty(X)$, the space of functions for

which differential operators may be applied arbitrarily many times, is too small of a class of functions. Even smaller is $\mathcal{C}_0^\infty(X)$, the space of functions in $\mathcal{C}^\infty(X)$ with compact support, and yet, $\mathcal{C}_0^\infty(X)$ offers enough flexibility to be used as a space of test functions in that given any $f \in \mathcal{L}^1_{\text{loc}}(X)$, then $\mathcal{T}_f$, defined by

$$\mathcal{T}_f(\phi) = \int_X f(x)\,\phi(x)\,dx, \quad \phi \in \mathcal{C}_0^\infty(X), \tag{2.2.1}$$

defines a linear form on $\mathcal{C}_0^\infty(X)$. Furthermore, the mapping $f \mapsto \mathcal{T}_f$ is one-to-one, modulo functions that differ on a set of measure zero. Therefore, the algebraic dual $\mathcal{C}_0^\infty(X)^\star$ contains an embedding of $\mathcal{L}^1_{\text{loc}}(X)$, and therefore also contains an embedding of $\mathcal{L}^p(X)$, for $1 \leq p \leq \infty$. Furthermore, if f is itself differentiable in at least the weak sense, then integration by parts tells us that

$$\mathcal{T}_{\mathcal{D}_{\vec{\mathbf{v}}} f}(\phi) = -\mathcal{T}_f(\mathcal{D}_{\vec{\mathbf{v}}}\phi),$$

which inspires defining $\mathcal{D}_{\vec{\mathbf{v}}}$ as an operator on $\mathcal{C}_0^\infty(X)^\star$ by

$$\mathcal{D}_{\vec{\mathbf{v}}} u(\phi) = -u(\mathcal{D}_{\vec{\mathbf{v}}}\phi), \quad u \in \mathcal{C}_0^\infty(X)^\star, \phi \in \mathcal{C}_0^\infty(X). \tag{2.2.2}$$

Unfortunately, $\mathcal{C}_0^\infty(X)^\star$ offers no topology, and in fact, is too large for analysis. We will want to restrict to a subspace of $\mathcal{C}_0^\infty(X)^\star$ that has a topology in which $\mathcal{C}_0^\infty(X)$ is embedded, and is dense, yet is not too small. Such a topology would not come from a norm. Hörmander proposes the following:

Definition 2.2.1 (Hörmander 2.1.1)**.** A distribution u in X is a linear form on $\mathcal{C}_0^\infty(X)$ such that for every compact set $K \subseteq X$, there exist constants C and k such that

$$|u(\phi)| \leq C \sum_{|\alpha| \leq k} \sup_{x \in K} |\partial^\alpha \phi(x)|, \quad \phi \in \mathcal{C}_0^\infty(K). \tag{2.2.3}$$

The set of all distributions in X is denoted by $\mathscr{D}'(X)$. If the same integer k can be used in (2.2.3) for every K, we say that u is of order $\leq k$, and denote the set of such distributions by $\mathscr{D}'^k(X)$. Their union $\mathscr{D}'_F(X) = \bigcup_{k=0}^\infty \mathscr{D}'^k(X)$ is the space of distributions of finite order.

For $1 \leq p \leq \infty$ and $k \geq 0$, we define the Sobolev space $\mathcal{W}^{k,p}(X)$ to be the space of functions $f \in \mathcal{L}^p(X)$ possessing all k-th order weak derivatives in $\mathcal{L}^p$. In particular, for each mutliindex α with $|\alpha| \leq k$, there is a function $\partial^\alpha f \in \mathcal{L}^p(X)$ for which

$$\int_X f(x)\,\partial^\alpha \phi(x)\,dx = (-1)^{|\alpha|}\int_X \partial^\alpha f(x)\,\phi(x)\,dx, \quad \phi \in \mathcal{C}_0^\infty(X).$$

One choice of norm on $\mathcal{W}^{k,p}(X)$ is then given by

$$\|f\|_{\mathcal{W}^{k,p}} = \sum_{|\alpha|\leq k} \|\partial^\alpha f\|_{\mathcal{L}^p},$$

and with this norm, $\mathcal{W}^{k,p}(X)$ is a Banach space that is continuously embedded in $\mathcal{L}^p(X)$.

In particular, $u \in \mathcal{C}_0^\infty(X)^\star$ is a distribution if for every compact subset K of X, $u|_{\mathcal{C}_0^\infty(K)}$ is a continuous linear form on $\mathcal{C}_0^\infty(K)$ under some $\mathcal{W}^{k,\infty}$ norm.

We use the weak-* topology on $\mathscr{D}'(X)$. That is, we say $u_k \to u$ in $\mathscr{D}'(X)$ if $u_k(\phi) \to u(\phi)$ for all $\phi \in \mathcal{C}_0^\infty(X)$.

If $u \in \mathscr{D}'(\mathbb{R}^n)$ additionally satisfies the estimate

$$|u(\phi)| \leq C_\beta \sum_{|\alpha|\leq k} \sup_{x\in\mathbb{R}^n} \left|x^\beta \partial^\alpha \phi(x)\right|, \quad \phi \in \mathcal{C}_0^\infty(\mathbb{R}^n), \tag{2.2.4}$$

for all multi-indices β, then u extends continuously to $\mathscr{S}(\mathbb{R}^n)$, the space of Schwartz functions, defined as

$$\mathscr{S}(\mathbb{R}^n) = \left\{\phi \in \mathcal{C}^\infty(\mathbb{R}^n) \,\middle|\, \sup_{x\in\mathbb{R}^n} \left|x^\beta \partial^\alpha \phi(x)\right| < \infty,\ \forall \alpha, \beta\right\}.$$

Continuous linear functionals on $\mathscr{S}(\mathbb{R}^n)$ are called **tempered distributions**, and we denote the space of such linear functionals as $\mathscr{S}'(\mathbb{R}^n)$.

The utility of estimate (2.2.3) becomes clear in the proof of the following theorem from Hörmander:

Theorem 2.2.2 (Hörmander 2.1.3)**.** *If $\phi \in \mathcal{C}^\infty(X \times Y)$, where Y is an open set in $\mathbb{R}^m$, and if there is a compact set $K \subseteq X$ such that $\phi(x,y) = 0$ when $x \notin K$, then*

$$y \mapsto u(\phi(x,y))$$

is a $\mathcal{C}^\infty$ function of y if $u \in \mathscr{D}'(X)$ and

$$\partial_y^\alpha u(\phi(x,y)) = u\left(\partial_y^\alpha \phi(x,y)\right).$$

The proof of this result can be seen by treating ϕ as smooth function of y, taking values in $\mathcal{C}_0^\infty(X)$. More specifically, if we let k be the order of u on K, we can identify ϕ with an infinitely differentiable map, in the Fréchet sense, from Y into $\mathcal{W}^{k,\infty}(X)$, an argument which requires verifying that for each $y_0 \in Y$, $\partial^\alpha \phi(x,y) \to \partial^\alpha \phi(x,y_0)$ *uniformly* in x as $y \to y_0$ for each multiindex α. The result then follows via chain rule, using the continuity requirement of u.

The notation used in the above theorem is due to Hörmander, however, we will hereafter use the notation $u_x(\phi(x,y))$ in place of $y \mapsto u(\phi(x,y))$ to denote that u is acting on ϕ as a function of x only, holding y as a parameter. Do notice that can slightly weaken the support requirement on ϕ specified in Theorem 2.2.2 by only requiring that each $y_0 \in Y$ has an open neighborhood to which the restriction of ϕ still satisfies the above support requirement.

The usefulness of this result then becomes clear when we see that if ϕ is itself in $\mathcal{C}_0^\infty(X \times Y)$, then $u_x(\phi(x,y)) \in \mathcal{C}_0^\infty(Y)$. This gives meaning to the formula $w_y(u_x(x,y))$ when $w \in \mathscr{D}'(Y)$. In fact, in view of Theorem 5.1.1 of Hörmander, this very formula defines a distribution on $X \times Y$, which we call the **tensor product** of u and w, denoted by $u \otimes w$; it is furthermore equal to $u_x(w_y(\phi(x,y)))$. The equality of these two formulas is easily seen as an application of Fubini-Tonelli when both u and w are identified with $\mathcal{L}^1_{\text{loc}}$ functions.

As it turns out, beyond the linear structure, we can carry over some ideas from classical analysis over to the theory of distributions.

2.2.1 Support of a distribution

Motivated by the fact that $\int_X f(x)\,\phi(x)\,dx = 0$ for every $\phi \in \mathcal{C}_0^\infty(X \backslash \operatorname{supp}(f))$, and $\operatorname{supp}(f)$ is the smallest closed set satisfying such property, we can easily extend the definition of support to distributions. Hörmander first defines the restriction of a distribution u to an open set $Y \subseteq X$ as being the restriction (in the usual sense) to $\mathcal{C}_0^\infty(Y)$, before the definition of support as follows:

Definition 2.2.3 (Hörmander 2.2.2)**.** If $u \in \mathscr{D}'(X)$, then the support of u, denoted $\operatorname{supp}(u)$, is the set of points in X having no open neighborhood to which the restriction of u is 0.

In other words, the complement of $\operatorname{supp}(u)$ is the largest open subset $U \subseteq X$ for which $u(\phi) = 0$ for all $\phi \in \mathcal{C}_0^\infty(U)$. We define $\mathscr{E}'(X)$ as being the set of compactly-supported distributions on X. This notation is due to Schwartz defining $\mathscr{E}(X)$ as the space of $\mathcal{C}^\infty$ functions with the topology defined by the family of seminorms

$$\phi \mapsto \sum_{|\alpha| \leq k} \sup_{x \in K} \left|\partial^\alpha \phi(x)\right|, \quad k \geq 0, K \subset\subset X.$$

Theorem 2.3.1 from Hörmander then states that the space of compactly-supported distributions is identical to the dual space of $\mathscr{E}(X)$. While in general, $\mathscr{D}'(X)$ cannot be embedded in $\mathscr{D}'(\mathbb{R}^n)$, we can embed $\mathscr{E}'(X)$ inside $\mathscr{D}'(\mathbb{R}^n)$ in a natural way, and as such, any operators on $\mathscr{D}'(\mathbb{R}^n)$ can act via this embedding on $\mathscr{E}'(X)$.

The singular support of u, denoted as $\operatorname{sing\,supp}(u)$, is defined as the smallest closed set D for which u is given by a $\mathcal{C}^\infty$ function on $X \backslash D$. That is, there exists $f \in \mathcal{C}^\infty(X \backslash D)$ such

that

$$u(\phi) = \int_{X \backslash D} f(x)\, \phi(x)\, dx, \quad \phi \in \mathcal{C}_0^\infty(X \backslash D).$$

While distributions are defined as linear forms on $\mathcal{C}_0^\infty$, distributions with support restrictions offer flexibility to be defined as a linear form on a larger space than $\mathcal{C}_0^\infty$. For instance, in the case that a distribution u is compactly-supported, it is nearly trivial to show that $u(\phi)$ can be defined for *any* $\phi \in \mathcal{C}^\infty(X)$, by taking $u(\phi) = u(\psi\phi)$ for some $\psi \in \mathcal{C}_0^\infty(X)$ that is equal to 1 on an open neighborhood of $\operatorname{supp}(u)$. Such an extension will be unique, as an alternate choice for ψ can be shown to yield the same extension.

Even if u is not compactly supported, it will be similarly possible to define $u(\phi)$ from only the condition that $\operatorname{supp}(\phi) \cap \operatorname{supp}(u)$ is compact by defining $u(\phi)$ as above, but instead taking ψ equal to 1 on an open neighborhood of $\operatorname{supp}(\phi) \cap \operatorname{supp}(u)$. Consequently, this allows us to give meaning to $w_y(u_x(\phi(x,y)))$ for $\phi \in \mathcal{C}^\infty(X \times Y)$, when $\operatorname{supp}_y(u_x(\phi(x,y))) \cap \operatorname{supp}(w)$ is compact, as is $\operatorname{supp}_x(\phi(x,y)) \cap \operatorname{supp}(u)$ for all $y \in Y$.

2.2.2 Multiplication

From formula (3.1.2) of Hörmander, one can multiply a distribution $u \in \mathscr{D}'(X)$ by a smooth function $\psi \in \mathcal{C}^\infty(X)$ by defining $(\psi u)(\phi) = u(\psi\phi)$. It is easy to extend to the case that ψ is only $\mathcal{C}^\infty$ on an open neighborhood of $\operatorname{supp} u$ by replacing ψ with a function in $\mathcal{C}^\infty(X)$ that is equal to ψ on $\operatorname{supp}(u)$, which then allows us to define the quotient $\frac{u}{\psi}$ when $\psi \in \mathcal{C}^\infty(X)$ is nonzero on $\operatorname{supp}(u)$.

In general, multiplication of two distributions is not defined. However, two distrubtions may have a well-defined product under some conditions. For instance, if two distributions have disjoint singular supports, then their product can be defined. Later, wavefront set analysis will help us establish a weaker condition when the product of two distributions

can indeed be defined, thereby extending multiplication of distributions past the case of disjoint singular supports.

2.2.3 Composition with smooth functions

If Φ is a diffeomorphism from X to Y, then for $f \in \mathcal{L}^1_{\text{loc}}(Y)$, one has

$$\int_X f(\Phi(x))\,\phi(x)\,dx = \int_Y f(y)\,\phi\left(\Phi^{-1}(y)\right)\left|\det \mathcal{D}\Phi^{-1}(y)\right|\,dy, \quad \phi \in \mathcal{C}_0^\infty(X),$$

and so for $u \in \mathscr{D}'(Y)$, we define $\Phi^\star u \in \mathscr{D}'(X)$ by

$$\Phi^\star u(\phi) = u\left(\left|\det \mathcal{D}\Phi^{-1}\right| \cdot \Phi^{-\star}\phi\right), \quad \phi \in \mathcal{C}_0^\infty(X).$$

If we only require Φ be a local diffeomorphism, then we can similarly define $\Phi^\star u$ with localization methods. Additionally, if Φ is a surjection onto Y, then $\Phi^\star$ is a one-to-one map from $\mathscr{D}'(Y)$ into $\mathscr{D}'(X)$. Thus, even if Φ is not globally invertible, we may define $\Phi^{-\star}w$ for $w \in \mathscr{D}'(Y)$ as being the distribution $u \in \mathscr{D}'(X)$ satisfying $w = \Phi^\star u$, if such a distribution exists. If a distribution w is in the range of $\Phi^\star$, then one can verify that for any pair of open sets U and V of X for which the restrictions $\Phi|_U$ and $\Phi|_V$ are injective, and $\Phi(U) = \Phi(V)$, we have

$$w(\phi) = w\left(\left|\det \mathcal{D}\Psi^{-1}\right| \cdot \Psi^{-\star}\phi\right), \quad \phi \in \mathcal{C}_0^\infty(U),$$

where $\Psi = \Phi|_V^{-1} \circ \Phi|_U : U \to V$ is the induced transition map. As it turns out, this condition is also sufficient for w to be in the range of $\Phi^\star$, and $\Phi^{-\star}w$ can also be constructed using localization methods.

The usefulness of reversing a pullback comes when working in an alternate coordinate system. For example, if a distribution $w \in \mathscr{D}'(X)$, where

$$X = \{(r,\theta) \mid r > 0, \theta \in \mathbb{R}\},$$

is 2π periodic in θ – that is, $w_{(r,\theta)}(\psi(r,\theta+2\pi)) = w(\psi)$ for $\psi \in \mathcal{C}_0^\infty(X)$, then w can be seen as a distribution in polar coordinates, and be pushed back to a distribution $u = \Phi^{-\star} w \in \mathscr{D}'\left(\mathbb{R}^2 \backslash 0\right)$ in Cartesian coordinates, where $\Phi(r,\theta) = (r\cos\theta, r\sin\theta)$.

Hörmander takes composition further with Theorem 6.1.2 by defining $\Phi^\star u$ when $\mathcal{D}\Phi(x)$ is merely surjective for every $x \in X$. This, of course, requires $n \geq m$. Such a construction requires we find a smooth function $\Psi : X \to \mathbb{R}^{n-m}$ so that $\Phi \oplus \Psi$ is a local diffeomorphism, and then we define

$$\Phi^\star u = (\Phi \otimes \Psi)^\star (u \otimes 1),$$

where 1 is identified as the constant map defined on $\mathbb{R}^{n-m}$.

In the case that $m > n$, $\Phi^\star u$ may be defined in some cases, but requires analysis of wavefront sets.

2.2.4 Convolution

When $f \in \mathcal{L}^1_{\text{loc}}(\mathbb{R}^n)$ and $\phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$, the convolution $f \star \phi$ is defined by

$$f \star \phi(x) = \int_{\mathbb{R}^n} f(y)\,\phi(x-y)\,dy,$$

and so in keeping with this formula, Hörmander defines the convolution $u \star \phi$ between $u \in \mathscr{D}'(\mathbb{R}^n)$ and $\phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$ in section 4.1 with

$$u \star \phi(x) = u_y(\phi(x-y)).$$

Theorem 4.1.1 from Hörmander establishes that this function is $\mathcal{C}^\infty(\mathbb{R}^n)$, with $\mathcal{D}_{\vec{\mathbf{v}}}(u \star \phi) = \mathcal{D}_{\vec{\mathbf{v}}} u \star \phi = u \star \mathcal{D}_{\vec{\mathbf{v}}}\phi$, which follows immediately from Theorem 2.2.2. Furthermore, we also have $\operatorname{supp}(u \star \phi) \subseteq \operatorname{supp}(u) + \operatorname{supp}(\phi)$. Theorem 4.1.2 also establishes that convolution with a smooth function also satisfies an associative property, in the sense that if we also have $\psi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$, then $(u \star \phi) \star \psi = u \star (\phi \star \psi)$.

In Theorem 4.2.1, Hörmander then establishes that for $v, w \in \mathscr{E}'(\mathbb{R}^n)$, if either v or w has compact support, a unique distribution u exists satisfying

$$u \star \phi = v \star (w \star \phi), \quad \phi \in \mathcal{C}_0^\infty(\mathbb{R}^n), \tag{2.2.5}$$

which is then defined as the convolution of v and w, denoted by $u = v \star w$. In particular, by resolving $u \star \{\phi(-x)\}$, we recover an explicit formula for the convolution of two distributions:

$$u(\phi) = v_x(w_y(\phi(x+y))). \tag{2.2.6}$$

This formula is found in Chapter 5 of Schwartz. Theorem 4.1.5 of Hörmander then establishes by construction through convolutions that $\mathcal{C}_0^\infty(X)$ is dense in $\mathscr{D}'(X)$ with the weak-* topology.

The Dirac distribution, δ, is defined by $\delta(\phi) = \phi(0)$. It is a quick computation to verify that $\delta \star w = w \star \delta = w$ for all $w \in \mathscr{D}'(\mathbb{R}^n)$.

While Hörmander requires either v or w to be compactly-supported to define $v \star w$, Schwartz weakens the support restriction in Chapter 6, Section 5, to only require that $(K - \operatorname{supp}(w)) \cap \operatorname{supp}(v)$ be compact for every compact set K. This weaker support condition ensures that for any $\phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$,

$$\operatorname{supp}_x(w_y(\phi(x+y))) \subseteq \operatorname{supp}(\phi) - \operatorname{supp}(w),$$

has a compact intersection with $\operatorname{supp}(v)$.

It should be observed that it is *not enough* for $(\{x_0\} - \operatorname{supp}(v)) \cap \operatorname{supp}(w)$ to be compact for every $x_0 \in \mathbb{R}^n$, as the following example illustrates:

Example 2.2.4. Consider the distribution w on $\mathbb{R}^2$, defined by

$$w(\phi) = \int_{\mathbb{R}} \phi(x, e^x)\, dx, \quad \phi \in \mathcal{C}_0^\infty\left(\mathbb{R}^2\right).$$

If we were to convolve w with $J_{\vec{\mathbf{e}}_1}$, where $\vec{\mathbf{e}}_1 = (1,0)$ in the xy plane, we would consider that the support of $(\{x_0\} - \operatorname{supp}(w)) \cap \operatorname{supp}(J_{\vec{\mathbf{e}}_1})$ is at most a single point, but $(K - \operatorname{supp}(w)) \cap \operatorname{supp}(J_{\vec{\mathbf{e}}_1})$ fails to be compact if K contains some open set intersecting the x-axis. The convolution, according to (2.2.6), would then be

$$\begin{aligned} w \star J_{\vec{\mathbf{e}}_1}(\phi) &= w_{(x,y)}\left(\int_0^\infty \phi(x+t,y)\, dt\right) \\ &= \int_{\mathbb{R}}\int_0^\infty \phi(x+t,e^x)\, dt\, dx \\ &= \int_{\mathbb{R}}\int_x^\infty \phi(t,e^x)\, dt\, dx \\ &= \int_0^\infty \int_{\ln y}^\infty \frac{1}{y}\cdot\phi(x,y)\, dx\, dy, \end{aligned}$$

which fails to converge if ϕ does not identically vanish on the x-axis.

In view of composing a distribution with smooth functions, if we composed w with a translation, e.g., $\Phi^\star w = w(x - x_0)$, it is easy to verify that

$$\Phi^\star(v \star w) = \Phi^\star v \star w = v \star \Phi^\star w.$$

2.2.5 The Fourier transform

The Fourier transform of a compactly-supported distribution u is defined as

$$\hat{u}(\vec{\xi}) = u_x\left(e^{-ix\cdot\vec{\xi}}\right). \tag{2.2.7}$$

The right-hand side in fact is defined as an entire function of $\vec{\xi} \in \mathbb{C}^n$, and is known as the Fourier-Laplace transform of u (Hörmander Theorem 7.1.14). If u is compactly-supported with order k, then

$$\begin{aligned} \left|\hat{u}\left(\vec{\xi}\right)\right| &\le C_1 \left\|e^{ix\cdot\vec{\xi}}\right\|_{\mathcal{W}^{k,\infty}(\mathbb{R}^n)} \\ &\le C_2\left(1 + \|\vec{\xi}\|\right)^k, \end{aligned} \tag{2.2.8}$$

for some constants C_1 and C_2. This is quite a weakening of the Riemann-Lebesgue lemma regarding the decay of Fourier transforms of $\mathcal{L}^1$ functions at infinity. As can be expected, the Fourier transform of compactly-supported distributions satisfies many of the properties of the Fourier transform operator on $\mathcal{L}^2(\mathbb{R}^n)$, including the convolution theorem, as seen by

$$\begin{aligned}
\widehat{u \star w}\left(\vec{\xi}\right) &= (u \star w)_x \left(e^{-ix\cdot\vec{\xi}}\right) \\
&= u_x \left(w_y \left(e^{-i(x+y)\cdot\vec{\xi}}\right)\right) \\
&= u_x \left(e^{-ix\cdot\vec{\xi}} w_y \left(e^{-iy\cdot\vec{\xi}}\right)\right) \\
&= u_x \left(e^{-ix\cdot\vec{\xi}}\right) w_y \left(e^{-iy\cdot\vec{\xi}}\right) \\
&= \hat{u}\left(\vec{\xi}\right) \hat{w}\left(\vec{\xi}\right),
\end{aligned}$$

for $u, w \in \mathscr{E}'(\mathbb{R}^n)$.

Inversion must be done in a distributional sense, however. Since the Fourier transform maps $\mathscr{S}(\mathbb{R}^n)$ to itself, we can define the Fourier transform on tempered distributions in a distributional sense by

$$\hat{u}(\phi) = u(\hat{\phi}), \quad u \in \mathscr{S}'(\mathbb{R}^n), \phi \in \mathscr{S}(\mathbb{R}^n),$$

which is compatible with (2.2.7) when $u \in \mathscr{E}'(\mathbb{R}^n)$. The inverse Fourier transform is thusly defined in a similar manner, and can be applied to $\hat{u}$.

A feature of the Fourier transform which will prove useful later is the following symmetry result:

Theorem 2.2.5. *If $w \in \mathscr{E}'(\mathbb{R}^n)$ has odd symmetry across a hyperplane $\{x \cdot \vec{\mathbf{v}} = t_0\}$ for some fixed $\vec{\mathbf{v}} \in \mathcal{S}^{n-1}$ and $t_0 \in \mathbb{R}^n$, i.e.,*

$$w_x\left(\phi\left(\bar{x} + (2t_0 - t)\,\vec{\mathbf{v}}\right)\right) = -w(\phi), \quad \phi \in \mathcal{C}_0^\infty(\mathbb{R}^n),$$

where we write $x = \bar{x} + t\vec{\mathbf{v}}$ for $\bar{x} \in \vec{\mathbf{v}}^\perp$ and $t \in \mathbb{R}$, then $\hat{w}$ vanishes on $\vec{\mathbf{v}}^\perp$.

We will omit the proof of this result.

2.3 Wavefront Sets

The idea of the wavefront set is to not only describe the locations of singularities of $u \in \mathscr{D}'(X)$, but also the directions of these singularities. If $x_0 \notin \operatorname{sing\,supp}(u)$, then one can choose an open neighborhood U of x_0 for which u is $\mathcal{C}^\infty$ on U, and then choose $\phi \in \mathcal{C}_0^\infty(U)$ with $\phi(x_0) \neq 0$. It would then follow that $\phi u \in \mathcal{C}_0^\infty(\mathbb{R}^n)$, and as such, $\widehat{\phi u}$ rapidly decays on $\mathbb{R}^n$ – for each $N \geq 0$, there is a $C_N \geq 0$ such that:

$$\left|\widehat{\phi u}\left(\vec{\xi}\right)\right| \leq C_N \left(1 + \|\vec{\xi}\|^2\right)^{-N/2}, \quad \vec{\xi} \in \mathbb{R}^n. \tag{2.3.1}$$

If $x \in \operatorname{sing\,supp}(u)$, then the above estimate fails for all choices of $\phi \in \mathcal{C}_0^\infty(X)$ where $\phi(x_0) \neq 0$. The wavefront set describes on which conic subset the estimate fails for at least one value of N.

Definition 2.3.1 (Wavefront set)**.** Let $u \in \mathscr{D}'(X)$. For $x_0 \in X$ and nonzero $\vec{\xi}_0 \in \mathbb{R}^n$, we say $\left(x_0, \vec{\xi}_0\right) \notin WF(u)$ if there exists a $\phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$ with $\phi(x_0) \neq 0$ and open conic (closed under positive scaling) neighborhood Γ of $\vec{\xi}_0$ such that the estimate (2.3.1) is valid for $\vec{\xi} \in \Gamma$, for all N.

This defines $WF(u)$ as a subset of $X \times \mathbb{R}^n \backslash 0$. Alternately, $\left(x_0, \vec{\xi}_0\right) \in WF(u)$ if for every $\phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$ with $\phi(x_0) \neq 0$ and open conic neighborhood Γ of $\vec{\xi}_0$, the estimate (2.3.1) fails on Γ for some N.

An alternate characterization of the wavefront set is that $\left(x_0, \vec{\xi}_0\right) \in WF(u)$ if and only if for every open neighborhood U of x_0 and open conic neighborhood Γ of $\vec{\xi}_0$, there exists $\phi \in \mathcal{C}_0^\infty(U)$ such that (2.3.1) fails on Γ for some N.

Some initial results following from this definition are that

$$WF(\psi u) \subseteq \operatorname{supp}\psi \times \mathbb{R}^n \cap WF(u),$$

$$WF(u \pm w) \subseteq WF(u) \cup WF(w),$$

whenever $\psi \in \mathcal{C}^\infty(\mathbb{R}^n)$, $u, w \in \mathscr{D}'(\mathbb{R}^n)$. The estimate (2.2.8) is essential in proving the latter of these two results.

Since the definition of conic is closure with respect to positive scaling, it follows from this definition that wavefront set of a distribution is invariant under positive scaling of the second component. However, if a distribution is real (in the sense of mapping real-valued test functions into $\mathbb{R}$), then we have invariance with respect to negative scaling as well.

Of greater interest is how partial differential operators and their inverses (when they exist) alter wavefront sets. Indeed, when P is a linear partial differential operator with coefficients in $\mathcal{C}^\infty$, then $WF(Pu) \subseteq WF(u)$, Hörmander formula (8.1.11), a corollary from the following result:

Theorem 2.3.2. *For $u \in \mathscr{D}'(X)$, $WF(\mathcal{D}_{\vec{\mathbf{v}}} u) \subseteq WF(u)$.*

Proof. Suppose $\left(x_0, \vec{\xi}_0\right) \notin WF(u)$. Then there is an open neighborhood U of x_0 such that $\widehat{\psi u}$ is rapidly decaying in some open conic neighborhood of $\vec{\xi}_0$ for all $\psi \in \mathcal{C}_0^\infty(U)$, In particular, choose ψ to be equal to one on some smaller open neighborhood V of x_0. Now choose $\phi \in \mathcal{C}_0^\infty(V)$ with $\phi(x_0) \neq 0$. Then

$$\phi \mathcal{D}_{\vec{\mathbf{v}}}(\psi u) = \phi\psi \mathcal{D}_{\vec{\mathbf{v}}} u,$$

and so we then have

$$\mathcal{F}(\phi\psi \mathcal{D}_{\vec{\mathbf{v}}} u) = \hat{\phi} \star \left\{ i\vec{\mathbf{v}} \cdot \vec{\xi}\, \widehat{\psi u} \right\}.$$

Since $\widehat{\psi u}$ rapidly decays on open conic neighborhood of $\vec{\xi}_0$, so must $i\vec{\mathbf{v}} \cdot \vec{\xi}\, \widehat{\psi u}$, and likewise, so must the convolution $\hat{\phi} \star \left\{ i\vec{\mathbf{v}} \cdot \vec{\xi}\, \widehat{\psi u} \right\}$. □

While the above result indicates that linear partial differential operators do not add singularities, the following result describes their capacity to remove them:

Theorem 2.3.3 (Microlocal property [5]). *If P is a differential operator of order m with $\mathcal{C}^\infty$ coefficients on a manifold X, then*

$$WF(u) \subseteq \operatorname{Char} P \cup WF(Pu), \quad u \in \mathscr{D}'(X),$$

where the characteristic set $\operatorname{Char} P$ *is defined by*

$$\operatorname{Char} P = \left\{ \left(x, \vec{\xi}\right) \in T^\star(X) \,\middle|\, P_m\left(x, \vec{\xi}\right) = 0 \right\},$$

and P_m is the principal symbol of P.

If P is merely a directional derivative, i.e., $P = \mathcal{D}_{\vec{\mathbf{v}}}$, then $P_m\left(x, \vec{\xi}\right) = i\vec{\xi} \cdot \vec{\mathbf{v}}$, and so

$$\operatorname{Char} P = \mathbb{R}^n \times \vec{\mathbf{v}}^{\perp}.$$

Hence, $\mathcal{D}_{\vec{\mathbf{v}}}$ may remove elements from the wavefront set of a distribution u whose second components are orthogonal to the direction of differentiation.

As for antidifferentiation, we will want to make use of the following theorem:

Theorem 2.3.4 (Hörmander 8.1.5). *Let V be a linear subspace of $\mathbb{R}^n$ and $u = u_0\, dS$, where $u_0 \in \mathcal{C}^\infty(V)$ and dS is Euclidean surface measure (on V). That is,*

$$u(\phi) = \int_V u_0(x)\, \phi(x)\, dS(x).$$

Then

$$WF(u) \subseteq \operatorname{supp}(u) \times \left(V^{\perp} \backslash 0\right).$$

Proof. If $\chi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$, then

$$\mathcal{F}_{\mathbb{R}^n}(\chi u)\left(\vec{\xi} + \vec{\eta}\right) = \int_V \chi(x)\, u_0(x)\, e^{-ix\cdot\left(\vec{\xi}+\vec{\eta}\right)}\, dS(x)$$

$$= \int_V \chi(x)\, u_0(x)\, e^{-ix\cdot\vec{\xi}}\, dS(x)$$
$$= \mathcal{F}_V(\chi u_0)\left(\vec{\xi}\right),$$

for $\vec{\xi} \in V$, $\vec{\eta} \in V^{\perp}$. Since $\chi u_0 \in \mathcal{C}_0^{\infty}(V)$, $\mathcal{F}_V(\chi u_0)$ decays rapidly in $\vec{\xi}$. Thus, if Γ is an open cone in $\mathbb{R}^n$, $\chi|_V \neq 0$, and $\int_V \chi u_0\, dS \neq 0$, then $\mathcal{F}_{\mathbb{R}^n}(\chi u)$ will rapidly decay on Γ if and only if $\Gamma \cap V = \emptyset$. □

The consequence of this result is the following:

Theorem 2.3.5. *The wavefront set of $J_{\vec{\mathbf{v}}}$ is given by*

$$WF(J_{\vec{\mathbf{v}}}) = \{0\} \times (\mathbb{R}^n \backslash 0) \cup \mathbb{R}^+\vec{\mathbf{v}} \times \left(\vec{\mathbf{v}}^{\perp} \backslash 0\right). \tag{2.3.2}$$

Proof. We note that $J_{\vec{\mathbf{v}}}$ is equal to Euclidean line measure on $V = \mathbb{R}\vec{\mathbf{v}}$ on the half-space $H = \{x \,|\, x \cdot \vec{\mathbf{v}} > 0\}$. Hence,

$$WF(J_{\vec{\mathbf{v}}}) \cap H \times \mathbb{R}^n = \mathbb{R}^+\vec{\mathbf{v}} \times \left(\vec{\mathbf{v}}^{\perp} \backslash 0\right).$$

Meanwhile, since $\delta_0 = \mathcal{D}_{\vec{\mathbf{v}}} J_{\vec{\mathbf{v}}}$, $\{0\} \times (\mathbb{R}^n \backslash 0) \subseteq WF(J_{\vec{\mathbf{v}}})$, and since the only point of $\operatorname{supp}(J_{\vec{\mathbf{v}}})$ outside of H is at the origin, all of the elements of $WF(J_{\vec{\mathbf{v}}})$ thus accounted for, and so

$$WF(J_{\vec{\mathbf{v}}}) = \{0\} \times (\mathbb{R}^n \backslash 0) \cup \mathbb{R}^+\vec{\mathbf{v}} \times \left(\vec{\mathbf{v}}^{\perp} \backslash 0\right). \qquad \square$$

Since $J_{\vec{\mathbf{v}}}$ is a fundamental solution to $\mathcal{D}_{\vec{\mathbf{v}}}$, our interest in $J_{\vec{\mathbf{v}}}$ is in use as a convolution kernel, and so formula (8.2.16) from Hörmander will be necessary:

$$WF(k \star u) \subseteq \left\{ \left(x + y, \vec{\xi}\right) \,\middle|\, \left(x, \vec{\xi}\right) \in WF(k), \left(y, \vec{\xi}\right) \in WF(u) \right\}. \tag{2.3.3}$$

Applying this result to $J_{\vec{\mathbf{v}}}$, we obtain the following:

Theorem 2.3.6. *For $u \in \mathscr{D}'(\mathbb{R}^n)$ for which $\operatorname{supp}(u) \cap (K - \mathbb{R}_{\geq 0}\vec{\mathbf{v}})$ is compact for every compact set K,*

$$WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \subseteq WF(u) \cup \left\{ \left(x + t\vec{\mathbf{v}}, \vec{\xi} \right) \,\middle|\, \left(x, \vec{\xi} \right) \in WF(u), \vec{\xi} \in \vec{\mathbf{v}}^{\perp}, t > 0 \right\}.$$

We will later give a demonstration of this result independent from Hörmander (8.2.16) in Chapter 3.

It will become necessary to perform analysis in alternate coordinates in some circumstances. Recall that we define composition $\Phi^{\star}u$ when $u \in \mathscr{D}'(Y)$, $\Phi \in \mathcal{C}^{\infty}(X;Y)$, and $\mathcal{D}\Phi(x)$ is surjective for all $x \in X$. Using wavefront sets, it is possible to extend composition further. Suppose $\Gamma \subseteq X \times \mathbb{R}^n \backslash 0$ is closed, and conic in the second component. We then define the space

$$\mathscr{D}'_{\Gamma}(X) = \left\{ u \in \mathscr{D}'(X) \,\middle|\, WF(u) \subseteq \Gamma \right\},$$

with which we then assign a topology through application of the following result:

Proposition 2.3.7 (Hörmander 8.2.1)**.** *A distribution $u \in \mathscr{D}'(X)$ is in $\mathscr{D}'_{\Gamma}(X)$ if and only if for every $\phi \in \mathcal{C}_0^{\infty}(X)$ and closed cone $V \subseteq \mathbb{R}^n$ with*

$$\Gamma \cap \operatorname{supp}(\phi) \times V = \emptyset \tag{2.3.4}$$

we have

$$\sup_{\vec{\xi} \in V} \left\| \vec{\xi} \right\|^N \left| \widehat{\phi u}\left(\vec{\xi} \right) \right| < \infty, \quad N = 1, 2, \ldots.$$

We then define a topology on $\mathscr{D}'_{\Gamma}(X)$ in which we say

Definition 2.3.8 (Hörmander 8.2.2)**.** For a sequence $u_j \in \mathscr{D}'_{\Gamma}(X)$, we shall say that $u_j \to u$ in $\mathscr{D}'_{\Gamma}(X)$ if and only if

$$u_j \to u \text{ in } \mathscr{D}'(X) \text{ (weakly)}$$

$$\lim_{j\to\infty}\sup_{\vec{\xi}\in V}\left\|\vec{\xi}\right\|^N\left|\widehat{\phi u}\left(\vec{\xi}\right)-\widehat{\phi u_j}\left(\vec{\xi}\right)\right|=0,$$

for $N=1,2,\ldots$ if $\phi\in\mathcal{C}_0^\infty(X)$ and V is a closed cone in $\mathbb{R}^n$ such that (2.3.4) is valid.

Hormander Theorem 8.2.3 then implies that $\mathcal{C}_0^\infty(X)$ is dense in $\mathscr{D}'_\Gamma(X)$ with respect to this topology. We may now define composition of a distribution with a smooth function in a more general setting.

Theorem 2.3.9 (Hörmander 8.2.4)**.** *Let X and Y be open subsets of $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively and let $f:X\to Y$ be a $\mathcal{C}^\infty$ map. Denote the set of normals of the map by*

$$N_f=\left\{(f(x),\vec{\eta})\in Y\times\mathbb{R}^n\;\middle|\;\mathcal{D}f(x)^T\vec{\eta}=\vec{\mathbf{0}}\right\}. \tag{2.3.5}$$

Then the pullback $f^\star u$ can be defined in one and only one way for all $u\in\mathscr{D}'(Y)$ with

$$N_f\;\cap\;WF(u)=\emptyset \tag{2.3.6}$$

so that $f^\star u=u\circ f$ when $u\in\mathcal{C}^\infty$ and for any closed conic subset Γ of $Y\times\mathbb{R}^n\backslash 0$ with $\Gamma\;\cap\;N_f=\emptyset$ we have a continuous map $f^\star:\mathscr{D}'_\Gamma(Y)\to\mathscr{D}'_{f^\star\Gamma}(X)$,

$$f^\star\Gamma=\left\{\left(x,\mathcal{D}f(x)^T\vec{\eta}\right)\;\middle|\;(f(x),\vec{\eta})\in\Gamma\right\}. \tag{2.3.7}$$

In particular we have for every $u\in\mathscr{D}'(Y)$ satisfying (2.3.6)

$$WF(f^\star u)\subseteq f^\star WF(u).$$

Note that if f is a diffeomorphism, then the above wavefront set inclusion can be replaced with equality.

3 DISTRIBUTIONAL ANTIDERIVATIVES AND CONVOLUTIONS WITH DISTRIBUTIONS GIVEN BY LINE INTEGRALS

3.1 The distributional antiderivative

3.1.1 Definition

We may begin by extending the idea of directional antiderivatives to distributions, contingent on a support restriction found in subsection 2.2.4. In particular, given $\vec{\mathbf{v}} \in \mathcal{S}^{n-1}$, we want to focus on the particular directional antiderivatives of the form

$$\mathcal{J}_{\vec{\mathbf{v}}} f(x) = \int_0^\infty f(x - t\vec{\mathbf{v}})\, dt, \quad x \in \mathbb{R}^n.$$

Definition 3.1.1 (Distributional antiderivative)**.** Let $u \in \mathscr{D}'(\mathbb{R}^n)$, and assume

$$\operatorname{supp}(u) \cap (K - \mathbb{R}_{\geq 0}\vec{\mathbf{v}}) \subset\subset \mathbb{R}^n, \quad \forall K \subset\subset \mathbb{R}^n. \tag{3.1.1}$$

Then we define $\mathcal{J}_{\vec{\mathbf{v}}} u = J_{\vec{\mathbf{v}}} \star u$. We may also denote $\mathcal{J}_{\vec{\mathbf{v}}} u$ with the more familiar notation

$$\int_0^\infty u(x - t\vec{\mathbf{v}})\, dt. \tag{3.1.2}$$

Hence

$$\begin{aligned} \mathcal{J}_{\vec{\mathbf{v}}} u(\phi) &= u_x \left(\int_0^\infty \phi(x + t\vec{\mathbf{v}})\, dt \right) \\ &= u(\mathcal{J}_{-\vec{\mathbf{v}}}\phi). \end{aligned}$$

We note that $\operatorname{supp}(\mathcal{J}_{-\vec{\mathbf{v}}}\phi) \subseteq \operatorname{supp}(\phi) - \mathbb{R}_{\geq 0}\vec{\mathbf{v}}$, and so $u(\mathcal{J}_{-\vec{\mathbf{v}}}\phi)$ is defined in view of the support hypothesis on w. Furthermore, we will make (3.1.1) an implicit assumption of arbitrarily chosen distributions from $\mathscr{D}'(\mathbb{R}^n)$ for the remainder of this section.

3.1.2 Microlocal analysis of the distributional antiderivative operator

We now prove a series of results necessary to establish a relationship between $WF(u)$ and $WF(\mathcal{J}_{\vec{\mathbf{v}}}u)$.

Proposition 3.1.2. *Let $U \subseteq \mathbb{R}^n$ be open and assume U is invariant under translation in the direction $-\vec{\mathbf{v}}$. That is, $U - t\vec{\mathbf{v}} \subseteq U$ for $t \geq 0$. If $u, w \in \mathscr{D}'(\mathbb{R}^n)$ are equal on U, then $\mathcal{J}_{\vec{\mathbf{v}}}u = \mathcal{J}_{\vec{\mathbf{v}}}w$ are equal on U.*

Proof. Let $\phi \in \mathcal{C}_0^\infty(U)$. Then by the assumed translation invariance of U, $\operatorname{supp}(\mathcal{J}_{-\vec{\mathbf{v}}}\phi) \subseteq U$. Thus, $u(\mathcal{J}_{-\vec{\mathbf{v}}}\phi) = w(\mathcal{J}_{-\vec{\mathbf{v}}}\phi)$, giving us our desired result. □

Convention 3.1.3. *For $x \in \mathbb{R}^n$, we write $x = \bar{x} + t\vec{\mathbf{v}}$ for $\bar{x} \in \vec{\mathbf{v}}^\perp$ and $t \in \mathbb{R}$.*

Proposition 3.1.4. *Let $t_0 \in \mathbb{R}$, assume $w \in \mathscr{D}'(\mathbb{R}^n)$ has odd symmetry across the hyperplane $\{t = t_0\}$. Then $\mathcal{J}_{\vec{\mathbf{v}}}w$ has even symmetry across are hyperplane $\{t = t_0\}$.*

Proof. First, we note that due to symmetry, a stronger support restriction than (3.1.1) holds in that we may replace $t \geq 0$ with $t \in \mathbb{R}$. Furthermore, given the odd symmetry, if ϕ is any $\mathcal{C}^\infty(\mathbb{R}^n)$ function whose support projected onto $\vec{\mathbf{v}}^\perp$ is compact, and $\mathcal{D}_{\vec{\mathbf{v}}}\phi = 0$, then $w(\phi) = 0$. Then for arbitrary $\phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$, we observe

$$\begin{aligned}
\mathcal{J}_{\vec{\mathbf{v}}}w_x\left(\phi\left(\bar{x} + (2t_0 - t)\,\vec{\mathbf{v}}\right)\right) &= w_x\left(\mathcal{J}_{-\vec{\mathbf{v}}}\left\{\phi\left(\bar{x} + (2t_0 - t)\,\vec{\mathbf{v}}\right)\right\}\right) \\
&= w_x\left(\int_0^\infty \phi\left(\bar{x} + (2t_0 - t - s)\,\vec{\mathbf{v}}\right)\,ds\right) \\
&= -w_x\left(\int_0^\infty \phi\left(\bar{x} + (t - s)\,\vec{\mathbf{v}}\right)\,ds\right) \\
&= -w_x\left(\mathcal{X}_{\vec{\mathbf{v}}}\phi(\bar{x}) - \int_0^\infty \phi\left(\bar{x} + (t + s)\,\vec{\mathbf{v}}\right)\,ds\right) \\
&= -w_x\left(-\int_0^\infty \phi\left(\bar{x} + (t + s)\,\vec{\mathbf{v}}\right)\,ds\right) \\
&= -w\left(-\mathcal{J}_{-\vec{\mathbf{v}}}\phi\right)
\end{aligned}$$

$$= \mathcal{J}_{\vec{\mathbf{v}}} w\left(\phi\right).$$

Here, $\mathcal{X}_{\vec{\mathbf{v}}}\phi \in \mathcal{C}_0^\infty\left(\vec{\mathbf{v}}^\perp\right)$ denotes the X-ray transform of ϕ, defined by

$$\mathcal{X}_{\vec{\mathbf{v}}}\phi\left(\bar{x}\right) = \int_{\mathbb{R}} \phi\left(\bar{x} + s\vec{\mathbf{v}}\right)\, ds, \quad \bar{x} \in \vec{\mathbf{v}}^\perp. \qquad \square$$

We are now in a position to establish a simple relationship between $WF\left(u\right)$ and $WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right)$. While Theorem 2.3.3 implies that $\mathcal{J}_{\vec{\mathbf{v}}}$ will extend the wavefront set of a distribution by ***at most*** $\mathbb{R}^n \times \left(\vec{\mathbf{v}}^\perp \backslash 0\right)$, the following result states that $\mathcal{J}_{\vec{\mathbf{v}}}u$ ***will not*** contain an element $(x_0, \vec{\eta}_0) \in \mathbb{R}^n \times \left(\vec{\mathbf{v}}^\perp \backslash 0\right)$ in its wavefront set if $WF\left(w\right)$ omits $\mathbb{R}^n \times \{\vec{\eta}_0\}$ altogether.

Theorem 3.1.5. *Let $u \in \mathscr{D}'\left(\mathbb{R}^n\right)$, and $\vec{\eta}_0 \in \vec{\mathbf{v}}^\perp$. If*

$$WF\left(u\right) \cap \mathbb{R}^n \times \{\vec{\eta}_0\} = \emptyset,$$

then

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \cap \mathbb{R}^n \times \{\vec{\eta}_0\} = \emptyset.$$

Proof. Let $x_0 = \bar{x}_0 + t_0\vec{\mathbf{v}} \in \mathbb{R}^n$, where $\bar{x}_0 \in \vec{\mathbf{v}}^\perp$, and $t_0 \in \mathbb{R}$. Now let U be an open neighborhood of x_0 that is invariant under translation in the direction $-\vec{\mathbf{v}}$, and $\psi \in \mathcal{C}^\infty\left(\mathbb{R}^n\right)$ be chosen so that ψ is equal to 1 on U. Furthermore, we may both choose U and ψ so that $\operatorname{supp}\left(u\right) \cap \operatorname{supp}\left(\psi\right)$ is compact, and that $\operatorname{supp}\left(\psi\right)$ lies in some open half space $\{t < t_1\}$ for some $t_1 > t_0$. We then construct a distibution w that is equal to u on U and has odd symmetry across $\{t = t_1\}$ by

$$w\left(\phi\right) = \left(\psi u\right)_x\left(\phi\left(x\right) - \phi\left(\bar{x} + \left(2t_1 - t\right)\vec{\mathbf{v}}\right)\right).$$

Then $\mathcal{J}_{\vec{\mathbf{v}}}\left(\psi w\right) = \mathcal{J}_{\vec{\mathbf{v}}}w = \mathcal{J}_{\vec{\mathbf{v}}}u$ on X, and furthermore, both w and $\mathcal{J}_{\vec{\mathbf{v}}}w$ are compactly supported, and

$$WF\left(w\right) \cap \mathbb{R}^n \times \{\vec{\eta}_0\} = \emptyset.$$

Because of the odd symmetry of w across the plane $\{t = t_1\}$, $\hat{w}$ vanishes on $\vec{\mathbf{v}}^{\perp}$, and in fact, we have

$$i\tau\widehat{\mathcal{J}_{\vec{\mathbf{v}}}w}\left(\vec{\eta}+\tau\vec{\mathbf{v}}\right) = \hat{w}\left(\vec{\eta}+\tau\vec{\mathbf{v}}\right), \quad \vec{\eta}\in\vec{\mathbf{v}}^{\perp}, \tau\in\mathbb{R}.$$

Thus,

$$\widehat{\mathcal{J}_{\vec{\mathbf{v}}}w}\left(\vec{\eta}+\tau\vec{\mathbf{v}}\right) = \begin{cases} \frac{\hat{w}(\vec{\eta}+\tau\vec{\mathbf{v}})}{i\tau}, & \text{if } \tau\neq 0, \\ -i\mathcal{D}_{\vec{\mathbf{v}}}\hat{w}\left(\vec{\eta}\right), & \text{if } \tau = 0, \end{cases}$$

for $\vec{\eta}\in\vec{\mathbf{v}}^{\perp}$ and $\tau\in\mathbb{R}$. The case $\tau = 0$ comes from an application of l'Hôpital's rule. However, $-i\mathcal{D}_{\vec{\mathbf{v}}}\hat{w} = -\widehat{tw}$, and:

$$WF\left(-tw\right)\subseteq WF\left(w\right),$$

so $-i\mathcal{D}_{\vec{\mathbf{v}}}\hat{w}\left(\vec{\xi}\right)$ decays rapidly in an open conic neighborhood Γ of $\vec{\eta}_0$. We may assume without loss of generality that Γ is chosen to be convex in τ, i.e., if $\vec{\eta}+\tau_k\vec{\mathbf{v}}\in\Gamma$ for $k = 1, 2$, and $\tau_1 < \tau_2$, then $\vec{\eta}+\tau\vec{\mathbf{v}}\in\Gamma$ for $\tau_1\leq\tau\leq\tau_2$. In particular, we can set

$$\Gamma = \left\{\sigma\vec{\eta}_0+\vec{\xi}+\tau\vec{\mathbf{v}} : \sigma,\tau > 0, \vec{\xi}\in\{\vec{\eta}_0,\vec{\mathbf{v}}\}^{\perp}, \max\left\{\left\|\vec{\xi}\right\|, |\tau|\right\} < \varepsilon\sigma\right\},$$

for some $\varepsilon > 0$ sufficiently small. Then for each $N\geq 0$, we can choose C_N so that

$$\left|\mathcal{D}_{\vec{\mathbf{v}}}\hat{w}\left(\vec{\eta}+\tau\vec{\mathbf{v}}\right)\right| \leq C_N\left(1+\left\|\vec{\eta}\right\|^2+\tau^2\right)^{-N/2}, \quad \vec{\eta}\in\vec{\mathbf{v}}^{\perp}, \tau\in\mathbb{R}, \vec{\eta}+\tau\vec{\mathbf{v}}\in\Gamma.$$

We then deduce a bound on $\hat{w}$, performing integration by parts along the way. For $\tau > 0$, consider:

$$\begin{aligned} \left|\hat{w}\left(\vec{\eta}+\tau\vec{\mathbf{v}}\right)\right| &\leq C_N\int_0^{\tau}\left(1+\left\|\vec{\eta}\right\|^2+t^2\right)^{-N/2}\,dt \\ &= C_N\int_0^{\tau}\left(1+\left\|\vec{\eta}\right\|^2+t^2\right)^{-N/2}\cdot\frac{\partial}{\partial t}\left\{t\right\}\,dt \\ &= C_N\left(\tau\left(1+\left\|\vec{\eta}\right\|^2+\tau^2\right)^{-N/2}-\int_0^{\tau}\frac{\partial}{\partial t}\left\{\left(1+\left\|\vec{\eta}\right\|^2+t^2\right)^{-N/2}\right\}\cdot t\,dt\right) \\ &= C_N\left(\tau\left(1+\left\|\vec{\eta}\right\|^2+\tau^2\right)^{-N/2}+N\int_0^{\tau}t^2\left(1+\left\|\vec{\eta}\right\|^2+t^2\right)^{-N/2-1}\,dt\right) \end{aligned}$$

$$\begin{aligned}
&\leq C_N \left(\tau \left(1 + \|\vec{\eta}\|^2 + \tau^2 \right)^{-N/2} + N\tau \int_0^\tau t \left(1 + \|\vec{\eta}\|^2 + t^2 \right)^{-N/2-1} \, dt \right) \\
&= C_N \left(2\tau \left(1 + \|\vec{\eta}\|^2 + \tau^2 \right)^{-N/2} - \tau \left(1 + \|\vec{\eta}\|^2 \right)^{-N/2} \right) \\
&\leq 2C_N \tau \left(1 + \|\vec{\eta}\|^2 + \tau^2 \right)^{-N/2}.
\end{aligned}$$

A similar estimate will be obtained in the case $\tau < 0$, and will give us the following bound on $\hat{w}$:

$$|\hat{w}(\vec{\eta} + \tau\vec{\mathbf{v}})| \leq 2C_N |\tau| \left(1 + \|\vec{\eta}\|^2 + \tau^2 \right)^{-N/2}, \quad \vec{\eta} \in \vec{\mathbf{v}}^\perp, \tau \in \mathbb{R}, \vec{\eta} + \tau\vec{\mathbf{v}} \in \Gamma.$$

It then follows immediately that

$$\left| \widehat{\mathcal{J}_{\vec{\mathbf{v}}} w}(\vec{\eta} + \tau\vec{\mathbf{v}}) \right| \leq 2C_N \left(1 + \|\vec{\eta}\|^2 + \tau^2 \right)^{-N/2}, \quad \vec{\eta} \in \vec{\mathbf{v}}^\perp, \tau \in \mathbb{R}, \vec{\eta} + \tau\vec{\mathbf{v}} \in \Gamma.$$

As this is true for arbitrary $N \geq 0$, this proves that $(x_0, \vec{\eta}_0) \notin WF(\mathcal{J}_{\vec{\mathbf{v}}} w)$, therefore, $(x_0, \vec{\eta}_0) \notin WF(\mathcal{J}_{\vec{\mathbf{v}}} u)$. Since x_0 was chosen arbitrarily, we have

$$WF(\mathcal{J}_{\vec{\mathbf{v}}} u) \cap (\mathbb{R}^n \times \{\vec{\eta}_0\}) = \emptyset. \qquad \square$$

The above result implies that the only way $\mathcal{J}_{\vec{\mathbf{v}}} w$ can include an element $(x_0, \vec{\eta}_0) \in \mathbb{R}^n \times \vec{\mathbf{v}}^\perp$ in its wavefront set is if $WF(w)$ itself already contains some element of $\mathbb{R}^n \times \{\vec{\eta}_0\}$. The following result further refines the previous result by describing a necessary condition on x_0 in order for $\mathcal{J}_{\vec{\mathbf{v}}} w$ to contain $(x_0, \vec{\eta}_0)$ in its wavefront set. Intuition tells us that $(x_0 - t\vec{\mathbf{v}}, \vec{\eta}_0)$ must already belong to the wavefront set of w for some $t \geq 0$.

Proposition 3.1.6. *Let $U \subseteq \mathbb{R}^n$ be open and assume U is invariant under translation in the direction $-\vec{\mathbf{v}}$, and $\vec{\eta}_0 \in \vec{\mathbf{v}}^\perp$. If*

$$WF(u) \cap U \times \{\vec{\eta}_0\} = \emptyset, \tag{3.1.3}$$

then

$$WF(\mathcal{J}_{\vec{\mathbf{v}}} u) \cap U \times \{\vec{\eta}_0\} = \emptyset.$$

Proof. Let $U^\star$ be an open subset of U whose closure is entirely contained in U, and is also closed under translation in the direction $-\vec{\mathbf{v}}$, and choose a $\psi \in \mathcal{C}^\infty(U)$ that is equal to one on $U^\star$. Then $\psi u = u$ on $U^\star$, so $\mathcal{J}_{\vec{\mathbf{v}}}(\psi u) = \mathcal{J}_{\vec{\mathbf{v}}} u$, and in fact, we also have $\psi \mathcal{J}_{\vec{\mathbf{v}}} u = \mathcal{J}_{\vec{\mathbf{v}}} u$ on $U^\star$. Furthermore, (3.1.3) implies

$$WF(\psi u) \cap \mathbb{R}^n \times \{\vec{\eta}_0\} = \emptyset,$$

and so

$$WF(\mathcal{J}_{\vec{\mathbf{v}}}(\psi u)) \cap \mathbb{R}^n \times \{\vec{\eta}_0\} = \emptyset.$$

We then have

$$\begin{aligned} WF(\mathcal{J}_{\vec{\mathbf{v}}} u) \cap U^\star \times \{\vec{\eta}_0\} &= WF(\mathcal{J}_{\vec{\mathbf{v}}}(\psi u)) \cap U^\star \times \{\vec{\eta}_0\} \\ &\subseteq WF(\mathcal{J}_{\vec{\mathbf{v}}}(\psi u)) \cap \mathbb{R}^n \times \{\vec{\eta}_0\} \\ &= \emptyset. \end{aligned}$$

Since $U^\star$ was arbitrary, and each $x \in U$ has such an open neighborhood $U^\star$, we can deduce that

$$WF(\mathcal{J}_{\vec{\mathbf{v}}} u) \cap U \times \{\vec{\eta}_0\} = \emptyset. \qquad \square$$

We now prove the primary result of this section.

Theorem 3.1.7 (Propagation of singularities of the distributional antiderivative)**.** *Let $u \in \mathscr{D}'(\mathbb{R}^n)$. Then*

$$WF(\mathcal{J}_{\vec{\mathbf{v}}} u) \subseteq WF(u) \cup \left\{ (x + t\vec{\mathbf{v}}, \vec{\eta}) \,\middle|\, (x, \vec{\eta}) \in WF(u), \vec{\eta} \in \vec{\mathbf{v}}^\perp, t \geq 0 \right\}. \tag{3.1.4}$$

Proof. For each $\vec{\eta} \in \vec{\mathbf{v}}^\perp$, define

$$V_{\vec{\eta}} = \bigcup_{(x,\vec{\eta}) \in WF(u)} (x + \mathbb{R}_{\geq 0}\vec{\mathbf{v}}), \quad U_{\vec{\eta}} = V_{\vec{\eta}}^C.$$

Since $\mathcal{D}_{\vec{\mathbf{v}}}\mathcal{J}_{\vec{\mathbf{v}}}u = u$, we can start with

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \subseteq WF\left(u\right) \cup \mathbb{R}^n \times \left(\vec{\mathbf{v}}^{\perp} \backslash 0\right). \tag{3.1.5}$$

For each fixed $\vec{\eta}_0 \in \vec{\mathbf{v}}^{\perp}$, $V_{\vec{\eta}_0}$ is invariant under translations in the direction $\vec{\mathbf{v}}$, and furthermore is closed. To show that $V_{\vec{\eta}_0}$ is closed, choose a sequence $\{x_k + t_k\vec{\mathbf{v}}\}_{k\in\mathbb{N}}$ in $V_{\vec{\eta}_0}$ that converges to some $L \in \mathbb{R}^n$. We will show that $L \in V_{\vec{\eta}_0}$. Setting

$$K = \overline{\{x_k + t_k\vec{\mathbf{v}} \,|\, k \geq 0\}},$$

we obtain a compact set for which $x_k \in \operatorname{supp}\left(u\right) \cap \left(K - \mathbb{R}_{\geq 0}\vec{\mathbf{v}}\right)$. The assumption made in (3.1.1) requires this intersection to be compact, and as such, $\{x_k\}_{k\in\mathbb{N}}$ must have a convergent subsequence, with limit $x \in \operatorname{supp}\left(u\right) \cap \left(K - \mathbb{R}_{\geq 0}\vec{\mathbf{v}}\right)$. The corresponding subsequence of $\{t_k\}_{k\in\mathbb{N}}$ would then converge to a limit $t \geq 0$ so that $t\vec{\mathbf{v}} = L - x$. Furthermore, $(x_k, \vec{\eta}_0) \in WF\left(u\right)$, and as $WF\left(u\right)$ is closed, we must also have $(x, \vec{\eta}_0) \in WF\left(u\right)$, and so $L = x + t\vec{\mathbf{v}} \in V_{\vec{\eta}_0}$.

Thus, $U_{\vec{\eta}_0}$ is open and invariant under translations in the direction $-\vec{\mathbf{v}}$. Furthermore,

$$WF\left(u\right) \cap U_{\vec{\eta}_0} \times \{\vec{\eta}_0\} = \emptyset,$$

and so by Theorem 3.1.6,

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \cap U_{\vec{\eta}_0} \times \{\vec{\eta}_0\} = \emptyset. \tag{3.1.6}$$

Therefore, if $(x_0, \vec{\eta}_0) \in WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \backslash WF\left(u\right)$, then $\vec{\eta}_0 \in \vec{\mathbf{v}}^{\perp}$, and (3.1.6) requires $x_0 \notin U_{\vec{\eta}_0}$, and so $(x_0, \vec{\eta}_0) \in V_{\vec{\eta}_0} \times \{\vec{\eta}_0\}$. Thus,

$$\begin{aligned} WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \backslash WF\left(u\right) &\subseteq \bigcup_{\vec{\eta}\in\vec{\mathbf{v}}^{\perp}} V_{\vec{\eta}} \times \{\vec{\eta}\} \\ &= \bigcup_{\vec{\eta}\in\vec{\mathbf{v}}^{\perp}} \bigcup_{(x,\vec{\eta})\in WF(u)} (x + \mathbb{R}_{\geq 0}\vec{\mathbf{v}}) \times \{\vec{\eta}\} \\ &= \left\{(x + t\vec{\mathbf{v}}, \vec{\eta}) \,\middle|\, (x, \vec{\eta}) \in WF\left(u\right), \vec{\eta} \in \vec{\mathbf{v}}^{\perp}, t \geq 0\right\}. \end{aligned}$$

□

Hence, singularities are propagated along rays in the direction of $\vec{\mathbf{v}}$, starting at existing singularities for which the direction component $\vec{\xi}_0$ is orthogonal to $\vec{\mathbf{v}}$. As an example, if we consider a distribution u whose wavefront set consists of a smooth curve, along with some of its normal vectors, singularities are propagated only from places where $\vec{\mathbf{v}}$ is tangent to the curve.

Figure 3.1 illustrates the propagation of singularities result for $\mathcal{J}_{\vec{\mathbf{v}}}$. Showing $WF(u)$ in blue, we see the maximum possible extent of $WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \setminus WF(u)$ is given in green. However, $WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \setminus WF(u)$ need not be the maximum possible extent of propagation. In fact, propagation may come to a halt along the ray, and intuition would tell us that such termination would happen only at another point at which u has a singularity corresponding to the direction of integration. In other words, given a point x_0 and $\vec{\eta}_0 \in \vec{\mathbf{v}}^{\perp}$, $WF(u)$ must contain the endpoints of each connected component of the intersection of $WF(\mathcal{J}_{\vec{\mathbf{v}}}u)$ with an embedding of the real line $(x_0 + \mathbb{R}\vec{\mathbf{v}}) \times \{\vec{\eta}_0\}$.

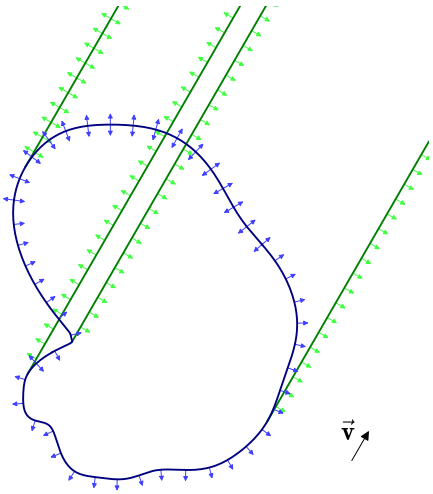

FIGURE 3.1: Visualisation of $WF(u)$, and possible extent of $WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \setminus WF(u)$.

Such a result would supplement our main result of this section. To show this result, we begin with the following proposition:

Proposition 3.1.8. *Let $u \in \mathscr{D}'(\mathbb{R}^n)$, $\vec{\eta}_0 \in \vec{\mathbf{v}}^{\perp}$, and U_0 be a bounded open set, $t_1 > 0$, and*

$$U_t = U_0 + t\vec{\mathbf{v}}, \quad t \in \mathbb{R},$$

$$U = \bigcup_{0 \le t \le t_1} U_t.$$

If

$$WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \cap U_0 \times \{\vec{\eta}_0\} = \emptyset, \tag{3.1.7}$$

and

$$WF\left(u\right)\ \cap\ U\times\{\vec{\eta}_0\}=\emptyset, \tag{3.1.8}$$

then

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right)\ \cap\ U\times\{\vec{\eta}_0\}=\emptyset. \tag{3.1.9}$$

Proof. We may assume without loss of generality that U_0 is convex, otherwise, we can apply the following argument to every convex open subset of U_0. Let $U_0^\star$ be an open set whose closure is contained in U_0, then define $U^\star$ in much the same way as U. Now let $\psi\in\mathcal{C}_0^\infty\left(U\right)$ be equal to 1 on $U^\star$. Consider the distributional derivative

$$\mathcal{D}_{\vec{\mathbf{v}}}\left(\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u\right)=\mathcal{D}_{\vec{\mathbf{v}}}\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u+\psi\cdot u.$$

Since $\mathcal{J}_{\vec{\mathbf{v}}}\psi$ vanishes on $U^\star$, we have $\mathcal{D}_{\vec{\mathbf{v}}}\left(\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u\right)=u$ on $U^\star$, and since

$$WF\left(u\right)\ \cap\ U^\star\times\{\vec{\eta}_0\}\subseteq WF\left(u\right)\ \cap\ U\times\{\vec{\eta}_0\}=\emptyset,$$

we have

$$WF\left(\mathcal{D}_{\vec{\mathbf{v}}}\left(\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u\right)\right)\ \cap\ U^\star\times\{\vec{\eta}_0\}=\emptyset.$$

Furthermore, since

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right)\ \cap\ U_0\times\{\vec{\eta}_0\}=\emptyset,$$

it must also follow that

$$WF\left(\mathcal{D}_{\vec{\mathbf{v}}}\left(\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u\right)\right)\ \cap\ U_0\times\{\vec{\eta}_0\}=\emptyset,$$

and so

$$WF\left(\mathcal{D}_{\vec{\mathbf{v}}}\left(\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u\right)\right)\ \cap\ \left(U_0\ \cup\ U^\star\right)\times\{\vec{\eta}_0\}=\emptyset.$$

We can replace $U_0\ \cup\ U^\star$ with $\bigcup_{t\leq 0}U_t\ \cup\ U^\star$, as that introduces no points that are inside the support of ψ (hence the requirement that U_0 be convex), and is also invariant under translation in the direction $-\vec{\mathbf{v}}$, and so we again apply (3.1.5) to obtain:

$$WF\left(\psi\cdot\mathcal{J}_{\vec{\mathbf{v}}}u\right)\ \cap\ \left(U_0\ \cup\ U^\star\right)\times\{\vec{\eta}_0\}=\emptyset.$$

In particular,

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \cap U^{\star} \times \{\vec{\eta}_0\} = \emptyset,$$

and since $U_0^{\star}$ was arbitrary, we can replace $U^{\star}$ with U:

$$WF\left(\mathcal{J}_{\vec{\mathbf{v}}}w\right) \cap U \times \{\vec{\eta}_0\} = \emptyset. \qquad \square$$

We then go on to prove the following:

Theorem 3.1.9. *Let $(x_0, \vec{\eta}_0) \in WF(u)$ with $\vec{\eta}_0 \in \vec{\mathbf{v}}^{\perp}$, and assume $(x_0 + I\vec{\mathbf{v}}) \times \{\vec{\eta}_0\}$ is a connected component of $WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right) \cap (x_0 + \mathbb{R}\vec{\mathbf{v}}) \times \{\vec{\eta}_0\}$, where either $I = [t_{\min}, t_{\max}]$ or $[t_{\min}, \infty)$. Then*

$$(x_0 + t_{\min}\vec{\mathbf{v}}, \vec{\eta}_0) \in WF(u),$$

and in the bounded case,

$$(x_0 + t_{\max}\vec{\mathbf{v}}, \vec{\eta}_0) \in WF(u).$$

Proof. We will focus efforts on proving the first inclusion. Choose sequence t_k, increasing to $t_{\min}$, so that $(x_0 + t_k\vec{\mathbf{v}}, \vec{\eta}_0) \notin WF\left(\mathcal{J}_{\vec{\mathbf{v}}}u\right)$. For each k, we can place $x_0 + t_k\vec{\mathbf{v}}$ inside an open neighborhood U_0, with diameter less than $\frac{1}{k}$, so that (3.1.7) holds. However, (3.1.9) fails when we take

$$U = \bigcup_{0 \le t \le t_{\min} - t_k} U_t,$$

and so $WF(u) \cap U \times \{\vec{\eta}_0\}$ is non-empty. In particular, we can find $(x_k, \vec{\eta}_0) \in WF(u) \cap U_{s_k} \times \{\vec{\eta}_0\}$ for some s_k between 0 and $t_{\min} - t_k$. We then notice that

$$\begin{aligned} \|x_0 + t_{\min}\vec{\mathbf{v}} - x_k\| &\le t_{\min} - t_k + s_k + \|(x_0 + (t_k + s_k)\,\vec{\mathbf{v}}) - x_k\| \\ &\le t_{\min} - t_k + s_k + \frac{1}{k}, \end{aligned}$$

which converges to 0 as $k \to \infty$. Hence, $x_k \to x_0 + t_{\min}\vec{\mathbf{v}}$, and as $WF(u)$ is a closed set, we must conclude that $(x_0 + t_{\min}\vec{\mathbf{v}}, \vec{\eta}_0) \in WF(u)$.

To obtain the inclusion of $(x_0 + t_{\max}\vec{\mathbf{v}}, \vec{\eta}_0)$, we may repeat the above argument in terms of $\mathcal{J}_{-\vec{\mathbf{v}}}$, using a cutoff function as needed. □

An implication of the above result is that if $(x_0, \vec{\eta}_0) \in WF(\mathcal{J}_{\vec{\mathbf{v}}}u) \setminus WF(u)$, which itself requires that $\vec{\eta}_0 \in \vec{\mathbf{v}}^{\perp}$, and $WF(u) \cap (x_0 + I\vec{\mathbf{v}}) \times \{\vec{\eta}_0\}$ is empty for some open interval I containing 0 in $\mathbb{R}$, then $(x_0 + I\vec{\mathbf{v}}) \times \{\vec{\eta}_0\} \subseteq WF(\mathcal{J}_{\vec{\mathbf{v}}}u)$. This result can be seen as a case of Theorem 8.3.3' from Hörmander.

3.2 Convolution of distributions with distributions given by (weighted) curve integrals

3.2.1 Definition

Now that we have given meaning to the integral (3.1.2), we now wish to give meaning to the following integral:

$$\int_0^{\infty} u(x - \gamma(t))\, v(t)\, dt, \tag{3.2.1}$$

given $\gamma \in \mathcal{C}^{\infty}([0,\infty); \mathbb{R}^n)$ and positive-valued $v \in \mathcal{C}^{\infty}([0,\infty), \mathbb{R})$, with $\gamma(0) = 0$, $\gamma'(t) \neq 0$, and $\|\gamma(t)\| \to \infty$ as $t \to \infty$. It is necessary to investigate such integrals in order to give a distributional meaning to the Polar Broken Ray transform. In particular, it will help to understand the meaning such integrals in order to give meaning to, and investigate the wavefront set of

$$\mathcal{Q}_2 f(s\vec{\sigma}) = \int_0^{\infty} f(s\vec{\sigma} + tA\vec{\sigma})\, dt.$$

We will then extend Theorem 3.1.7 to a more general result for (3.2.1).

Definition 3.2.1. The integral (3.2.1) is defined as the convolution of u with the distribution $J_{\gamma,v}$ defined by

$$J_{\gamma,v}(\phi) = \int_0^{\infty} \phi(\gamma(t))\, v(t)\, dt,$$

and denoted as $\mathcal{J}_{\gamma,\upsilon}u$, subject to the support restriction

$$\operatorname{supp}(u) \cap (K - \{\gamma(t) \mid t \geq 0\}) \subset\subset \mathbb{R}^n, \quad \forall K \subset\subset \mathbb{R}^n. \tag{3.2.2}$$

3.2.2 Microlocal analysis of $\mathcal{J}_{\gamma,\upsilon}$

We will employ the use of pullbacks so that we may use the wavefront set results from the previous section. Extend γ and υ to $\mathcal{C}^\infty$ maps defined on $\mathbb{R}$, and supported in $(-\varepsilon, \infty)$ for some $\varepsilon > 0$, then define the maps $\chi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and $\psi_0 : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}$ by

$$\chi(x, y) = x - \gamma(y), \quad \psi_0(x) = (x, 0).$$

we will want to interpret $\mathcal{J}_{\gamma,\upsilon}u$ as the pullback

$$\mathcal{J}_{\gamma,\upsilon}u = \psi_0^\star \mathcal{J}_{(\vec{\mathbf{0}},-1)}(\upsilon \cdot \chi^\star u), \tag{3.2.3}$$

where the multiplication by υ is with υ identified with a function in $\mathcal{C}^\infty(\mathbb{R}^n \times \mathbb{R})$ whose dependence lies only in y. We will prove (3.2.2) later.

First, we will interpret (3.2.3) with more familiar notation. Notice that we may use $u(x - \gamma(y))$ to denote the pullback $\chi^\star u$ for $u \in \mathscr{D}'(\mathbb{R}^n)$. Multiplication of a distribution $w \in \mathscr{D}'(\mathbb{R}^n \times \mathbb{R})$ by υ is then denoted by $w(x, y)\upsilon(y)$. The operator $\mathcal{J}_{(0,-1)}$ is integration in the y-coordinate, and as such,

$$\mathcal{J}_{(\vec{\mathbf{0}},-1)}w = \int_0^\infty w(x, y + t)\, dt.$$

Finally, since ψ_0 simply embeds $\mathbb{R}^n$ inside $\mathbb{R}^n \times \mathbb{R}$ by setting $y = 0$, we may use the notation $w(x, y)|_{y=0}$ to denote the pullback $\psi_0^\star w$.

Putting this notation together, we write

$$\psi_0^\star \mathcal{J}_{(\vec{\mathbf{0}},-1)}(\upsilon \cdot \chi^\star u) = \int_0^\infty u(x - \gamma(y + t))\upsilon(y + t)\, dt \Big|_{y=0}.$$

Theorem 3.2.2. *If $u \in \mathscr{D}'(\mathbb{R}^n)$ is a distribution satisfying the support restriction* (3.2.2)*, then*

$$WF\left\{\int_0^\infty u(x-\gamma(t))\,\upsilon(t)\,dt\right\}\backslash WF(u) \subseteq \left\{\left(x,\vec{\xi}\right)\,\middle|\,\exists t\geq 0:\vec{\xi}\perp\gamma'(t)\ \ \&\ \ \left(x-\gamma(t),\vec{\xi}\right)\in WF(u)\right\}.$$

Proof. We may, without loss of generality, set $\upsilon = 1$, as doing so will not alter the wavefront sets involved in this proof. With

$$\chi(x,y) = x-\gamma(y),$$

we observe that

$$\mathcal{D}\chi(x,y) = \begin{bmatrix}\mathcal{I}_n & -\gamma'(y)\end{bmatrix},\quad \mathcal{D}\chi(x,y)^T = \begin{bmatrix}\mathcal{I}_n \\ -\gamma'(y)^T\end{bmatrix},$$

where $\mathcal{I}_n$ is the identity map on $\mathbb{R}^n$, and so $\ker\mathcal{D}\chi(x,y)^T$ is trivial, indicating that the pullback $u(x-\vec{\gamma}(y))$ is indeed well-defined, and

$$WF\{u(x-\gamma(y))\}\subseteq\chi^\star WF(u),$$

where

$$\chi^\star WF(u) = \left\{\left((x,y),\left(\vec{\xi},\gamma'(y)\cdot\vec{\xi}\right)\right)\,\middle|\,\left(x-\gamma(y),\vec{\xi}\right)\in WF(u)\right\}. \tag{3.2.4}$$

We now wish to show that

$$WF\{u(x-\gamma(y))\} = \chi^\star WF(u).$$

Indeed, let $y_0 > -\varepsilon$, and define

$$\psi_{y_0}(x) = \begin{bmatrix}x+\gamma(y_0)\\ y_0\end{bmatrix}.$$

Then

$$\mathcal{D}\psi_{y_0}(x) = \begin{bmatrix} \mathcal{I}_n \\ 0 \end{bmatrix}. \quad \mathcal{D}\psi_{y_0}(x)^T = \begin{bmatrix} \mathcal{I}_n & 0 \end{bmatrix},$$

and so

$$\ker \mathcal{D}\psi_{y_0}(x)^T = \{0\} \times \mathbb{R}.$$

Therefore, the set of normals for ψ_{y_0} satisfies

$$N_{\psi_{y_0}} \subseteq (\mathbb{R}^n \times (-\varepsilon, \infty)) \times (\{0\} \times (\mathbb{R} \backslash 0)),$$

and so $N_{\psi_{y_0}} \cap WF\{u(x-\gamma(y))\} = \emptyset$. Thus, the pullback $\psi_{y_0}^\star\{u(x-\gamma(y))\}$ is well-defined, and is in fact equal to u. Then:

$$WF(u) = WF\left(\psi_{y_0}^\star\{u(x-\gamma(y))\}\right) \subseteq \psi_{y_0}^\star WF\{u(x-\gamma(y))\}, \tag{3.2.5}$$

where

$$\begin{aligned} &\psi_{y_0}^\star WF\{u(x-\gamma(y))\} \\ &\qquad = \left\{ \left(x, \vec{\xi}\right) \,\middle|\, \left((x+\gamma(y_0), y_0), \left(\vec{\xi}, \eta\right)\right) \in WF\{u(x-\gamma(y))\} \right\}. \end{aligned} \tag{3.2.6}$$

Thus, if we chose $\left((x, y_0), \left(\vec{\xi}, \gamma'(y_0)\cdot\vec{\xi}\right)\right) \in \chi^\star WF(u)$, this choice was based on choosing $\left(x-\gamma(y_0), \vec{\xi}\right) \in WF(u)$. We then have by (3.2.5) that

$$\left(x-\gamma(y_0), \vec{\xi}\right) \in \psi_{y_0}^\star WF\{u(x-\gamma(y))\}.$$

That is,

$$\left((x, y_0), \left(\vec{\xi}, \eta\right)\right) = \left((x-\gamma(y_0)+\gamma(y_0), y_0), \left(\vec{\xi}, \eta\right)\right) \in WF\{u(x-\gamma(y))\},$$

for some η. Of course, one only needs to choose $\eta = \gamma'(y_0)\cdot\vec{\xi}$, and this will yield the desired set inclusion.

Next, we can use (3.1.7) to describe $WF\left\{\int_0^\infty u(x-\gamma(y+t))\,dt\right\}$ as follows:

$$WF\left\{\int_0^\infty u(x-\gamma(y+t))\,dt\right\} \backslash WF\{u(x-\gamma(y))\}$$

$$
\begin{aligned}
&\subseteq \left\{ \left((x, y-t), \left(\vec{\xi}, \eta \right) \right) \,\middle|\, \left(\vec{\xi}, \eta \right) \perp \left(\vec{\mathbf{0}}, -1 \right), \right. \\
&\qquad\qquad \left. \left((x, y), \left(\vec{\xi}, \eta \right) \right) \in WF\left\{ u\left(x - \gamma\left(y \right) \right) \right\}, t \geq 0 \right\} \\
&= \left\{ \left((x, y-t), \left(\vec{\xi}, 0 \right) \right) \,\middle|\, \left((x, y), \left(\vec{\xi}, 0 \right) \right) \in \chi^{\star} WF\left(u \right), t \geq 0 \right\} \\
&= \left\{ \left((x, y-t), \left(\vec{\xi}, 0 \right) \right) \,\middle|\, \left(x - \gamma\left(y \right), \vec{\xi} \right) \in WF\left(u \right), \vec{\xi} \perp \gamma'\left(y \right), t \geq 0 \right\}.
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&WF\left\{ \int_0^{\infty} u\left(x - \gamma\left(y + t \right) \right) \, dt \Big|_{y=0} \right\} \\
&\quad \subseteq \chi_0^{\star} WF\left\{ u\left(x - \gamma\left(y \right) \right) \right\} \, \cup \, \chi_0^{\star} \left\{ \left((x, y-t), \left(\vec{\xi}, 0 \right) \right) : \right. \\
&\qquad \left. \xi \in \gamma\left(y \right)^{\perp}, \left(x - \gamma\left(y \right), \vec{\xi} \right) \in WF\left(u \right), t \geq 0 \right\} \\
&\quad = WF\left(u \right) \\
&\qquad \cup \, \left\{ \left(x, \vec{\xi} \right) \,\middle|\, \exists t \geq 0 : \left(x - \gamma\left(t \right), \vec{\xi} \right) \in WF\left(u \right), \vec{\xi} \in \vec{\gamma}\left(t \right)^{\perp} \right\}.
\end{aligned}
$$

Reintroducing the weight function υ, we write the above result as

$$
\begin{aligned}
&WF\left\{ \int_0^{\infty} u\left(x - \gamma\left(y + t \right) \right) \upsilon\left(t \right) \, dt \Big|_{y=0} \right\} \\
&\quad \subseteq WF\left(u \right) \, \cup \, \left\{ \left(x, \vec{\xi} \right) \,\middle|\, \exists t \geq 0 : \left(x - \gamma\left(t \right), \vec{\xi} \right) \in WF\left(u \right), \vec{\xi} \in \vec{\gamma}\left(t \right)^{\perp} \right\}. \quad (3.2.7)
\end{aligned}
$$

Finally, to show that $\mathcal{J}_{\gamma,\upsilon} u$ is in fact given by (3.2.3), let $u_k \in \mathcal{C}_0^{\infty}\left(\mathbb{R} \right)$ converge in distribution to u. Then for $\phi \in \mathcal{C}_0^{\infty}\left(\mathbb{R}^n \right)$, we have

$$
\begin{aligned}
\mathcal{J}_{\gamma,\upsilon} u\left(\phi \right) &= u_x \left(\int_0^{\infty} \phi\left(x + \gamma\left(t \right) \right) \upsilon\left(t \right) \, dt \right) \\
&= \lim_{k \to \infty} \int_{\mathbb{R}^n} u_k\left(x \right) \int_0^{\infty} \phi\left(x + \gamma\left(t \right) \right) \upsilon\left(t \right) \, dt \, dx \\
&= \lim_{k \to \infty} \int_0^{\infty} \int_{\mathbb{R}^n} u_k\left(x \right) \phi\left(x + \gamma\left(t \right) \right) \upsilon\left(t \right) \, dx \, dt \\
&= \lim_{k \to \infty} \int_0^{\infty} \int_{\mathbb{R}^n} u_k\left(x - \gamma\left(t \right) \right) \phi\left(x \right) \upsilon\left(t \right) \, dx \, dt
\end{aligned}
$$

$$
\begin{aligned}
&= \lim_{k\to\infty} \int_{\mathbb{R}^n} \int_0^\infty u_k\left(x-\gamma\left(t\right)\right) \upsilon\left(t\right)\, dt\, \phi\left(x\right)\, dx \\
&= \lim_{k\to\infty} \int_{\mathbb{R}^n} \int_0^\infty u_k\left(x-\gamma\left(y+t\right)\right) \upsilon\left(y+t\right)\, dt\Big|_{y=0}\, \phi\left(x\right)\, dx \\
&= \psi_0^\star \mathcal{J}_{(0,-1)}\left(\upsilon\cdot\chi^\star u\right)\left(\phi\right)
\end{aligned}
$$

The last of these equalities follows from the proof of Theorem 8.2.4 from Hörmander which defines the pullback of distributions in this manner. □

The propagation result (3.2.7) may be reparametrized as

$$
\begin{aligned}
&WF\left\{\int_0^\infty u\left(x-\gamma\left(t\right)\right)\upsilon\left(t\right)\, dt\right\} \\
&\qquad \subseteq WF\left(u\right) \cup \left\{\left(x+\gamma\left(t\right),\vec{\xi}\right) \,\middle|\, t\geq 0, \left(x,\vec{\xi}\right)\in WF\left(u\right), \vec{\xi}\in\vec{\gamma}\left(t\right)^\perp\right\}.
\end{aligned}
$$

Figure 3.2 illustrates the propagation of singularities result for $\mathcal{J}_{\gamma,\upsilon}$. Here, u is depicted as having only three singularities. Again, $WF\left(u\right)$ is depicted in blue, while the maximum possible extent of $WF\left(\mathcal{J}_{\gamma,\upsilon}u\right)\backslash WF\left(u\right)$ is shown in green. The curve γ is shown in maroon, with its translates shown as red dashed curves.

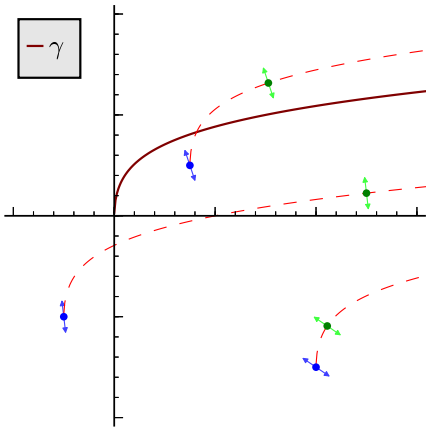

FIGURE 3.2: Visualisation of $WF\left(u\right)$, and possible extent of $WF\left(\mathcal{J}_{\gamma,\upsilon}u\right)\backslash WF\left(u\right)$.

It should be noted that the integral (3.2.1) reduces to the distributional antiderivative developed in the previous section when γ parametrizes a ray:

$$\gamma\left(t\right) = t\vec{\mathbf{v}}, \quad t\geq 0, \vec{\mathbf{v}}\in\mathcal{S}^{n-1}.$$

Furthermore, the result obtained in Theorem 3.2.2 in this case is:

$$WF\left\{\int_0^\infty u\left(x-\gamma\left(t\right)\right)\upsilon\left(t\right)\, dt\right\}\backslash WF\left(u\right)$$

$$\subseteq \left\{ \left(x+\gamma\left(t\right) ,\vec{\xi}\right) \left| \, t\geq 0,\left(x,\vec{\xi}\right) \in WF\left(u\right) ,\vec{\xi}\in \gamma ^{\prime }\left(t\right) ^{\perp }\right. \right\}$$
$$= \left\{ \left(x+t\vec{\mathbf{v}},\vec{\xi}\right) \left| \, t\geq 0,\left(x,\vec{\xi}\right) \in WF\left(u\right) ,\vec{\xi}\in \vec{\mathbf{v}}^{\perp }\right. \right\} ,$$

and so the above theorem agrees with (3.1.4) for this choice of γ.

4 THE BROKEN RAY TRANSFORM

4.1 Introduction

In this chapter, we will explore the various flavors of the Broken Ray transform previously investigated by Florescu, Markel, and Schotland, [1], Gouia-Zarrad and Ambartsoumian, [3], and Katsevich and Krylov, [6]. We start with verifying an inversion formula in the case of fixed inital and terminal directions, and then perform microlocal analysis of the Broken Ray transform, describing the scattering of singularities by both the transform itself, and its inversion formulas (a so-called "two-sided" microlocal analysis). We then investigate the setting of having two sets of simultaneous measurements with differing terminal directions, and also perform microlocal analysis in this setting.

We will also find explicit inversion formulas for Katsevich and Krylov's curved detector settings, as well as determining restrictions on support necessary to guarantee injectivity of these transforms, as well as restrictions required to give a distributional meaning to the inversion formulas. We then follow up with a two-sided microlocal analysis of the curved detector setting.

4.2 Broken Ray transform with fixed direction parameters

If $\vec{\mathbf{v}}_1$ and $\vec{\mathbf{v}}_2$ are distinct and held as fixed parameters, then for $f \in \mathcal{C}_0^2(\mathbb{R}^n)$, one can easily apply the Fundamental Theorem of Calculus to verify that an inversion formula for (1.1.2) is given by:

$$f(x) = \int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1} \mathcal{D}_{\vec{\mathbf{v}}_2} \mathcal{B} f\left(x - s\left(\vec{\mathbf{v}}_2 - \vec{\mathbf{v}}_1\right)\right) ds. \tag{4.2.1}$$

Indeed:

$$\begin{aligned}
&\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)\right)\,ds \\
&\quad=\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\left\{\int_0^\infty \left(f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)-t\vec{\mathbf{v}}_1\right)\right.\right. \\
&\qquad\left.\left.+f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)+t\vec{\mathbf{v}}_2\right)\right)\right\}\,dt\,ds \\
&\quad=\int_0^\infty\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)-t\vec{\mathbf{v}}_1\right)\,dt\,ds \\
&\qquad+\int_0^\infty\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)+t\vec{\mathbf{v}}_2\right)\,dt\,ds \\
&\quad=\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_2}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)\right)\,ds-\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)\right)\,ds \\
&\quad=\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)\right)\,ds \\
&\quad=f\left(x\right).
\end{aligned}$$

Recall that the Fundamental Theorem of Calculus gives us the identity

$$\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}}g\left(x\pm t\vec{\mathbf{v}}\right)\,dt=\mp g\left(x\right),\quad \vec{\mathbf{0}}\neq\vec{\mathbf{v}}\in\mathbb{R}^n, g\in\mathcal{C}_0^1\left(\mathbb{R}^n\right).$$

which we have used three times in the above computation.

This is not the only possible inversion formula, as we can verify the following alternate inversion in a similar manner:

$$f\left(x\right)=-\int_0^\infty \mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}f\left(x+s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)\right)\,ds. \tag{4.2.2}$$

Notice this alternate inversion formula differs from the first in that the integration occurs in the reverse direction from the first. We can then combine these two inversion formulas to arrive at a more symmetric inversion formula:

$$f\left(x\right)=\frac{1}{2}\int_{\mathbb{R}}\operatorname{sgn}\left(s\right)\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}f\left(x-s\left(\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1\right)\right)\,ds. \tag{4.2.3}$$

For a numerical implementation of (4.2.3), see Appendix A.

We can also reveal Gouia-Zarrad and Ambartsoumian's inversion formula, [3], by replacing $\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}$ in (4.2.2) with $\frac{1}{4}\left(\mathcal{D}^2_{\vec{\mathbf{v}}_1+\vec{\mathbf{v}}_2} - \mathcal{D}^2_{\vec{\mathbf{v}}_1-\vec{\mathbf{v}}_2}\right)$ and performing the appropriate simplifcation:

$$f(x) = \frac{1}{4}\mathcal{D}_{\vec{\mathbf{v}}_1-\vec{\mathbf{v}}_2}\mathcal{B}f(x) - \frac{1}{4}\int_0^\infty \mathcal{D}^2_{\vec{\mathbf{v}}_1+\vec{\mathbf{v}}_2}\mathcal{B}f\left(x + s\left(\vec{\mathbf{v}}_2 - \vec{\mathbf{v}}_1\right)\right)\, ds. \tag{4.2.4}$$

Gouia-Zarrad and Ambartsoumian take $\vec{\mathbf{v}}_1 = (\sin\beta, -\cos\beta)$ and $\vec{\mathbf{v}}_2 = (\sin\beta, \cos\beta)$, with f supported in $[0, x_{\max}]\times[0, y_{\max}]$, which of course gives $\mathcal{D}_{\vec{\mathbf{v}}_1-\vec{\mathbf{v}}_2} = -2\cos\beta\frac{\partial}{\partial y}$ and $\mathcal{D}^2_{\vec{\mathbf{v}}_1+\vec{\mathbf{v}}_2} = 4\sin^2\beta\frac{\partial^2}{\partial x^2}$.

4.2.1 Microlocal analysis of the Broken Ray transform with fixed initial and terminal directions

We can easily extend the Broken Ray transform in a distributional sense. In view of (1.1.2), we define $\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}$, with two distinct direction parameters $\vec{\mathbf{v}}_1$ and $\vec{\mathbf{v}}_2$ in $\mathcal{S}^{n-1}$, in a distributional sense as

$$\begin{aligned}\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}u &= \int_0^\infty u\left(x - t\vec{\mathbf{v}}_1\right)\, dt + \int_0^\infty u\left(x + t\vec{\mathbf{v}}_2\right)\, dt \\ &= \mathcal{J}_{\vec{\mathbf{v}}_1}u + \mathcal{J}_{-\vec{\mathbf{v}}_2}u, \quad u \in \mathscr{E}'\left(\mathbb{R}^n\right).\end{aligned} \tag{4.2.5}$$

The inversions given in (4.2.1) and (4.2.2) also hold in a distributional sense, in that

$$u = \mathcal{J}_{\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}u = -\mathcal{J}_{\vec{\mathbf{v}}_1-\vec{\mathbf{v}}_2}\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}u, \tag{4.2.6}$$

for $u \in \mathscr{E}'\left(\mathbb{R}^n\right)$.

In view of (4.2.5) and (4.2.6), we can use Theorem 3.1.7 to describe the relationship between $WF(u)$ and $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}u\right)$, and note that Theorem 3.1.9 applies to all directions of integration in both the Distributional Broken Ray transform and its inversion.

Proposition 4.2.1 (Propagation of singularities of the Distributional Broken Ray transform)**.** *Let $u \in \mathscr{E}'\left(\mathbb{R}^n\right)$. Then*

$$WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}u\right) \subseteq WF(u)$$

$$\cup \left\{ \left(x + t\vec{\mathbf{v}}_1, \vec{\xi} \right) \,\middle|\, \left(x, \vec{\xi} \right) \in WF\left(u \right), \vec{\xi} \in \vec{\mathbf{v}}_1^{\perp}, t > 0 \right\}$$
$$\cup \left\{ \left(x - t\vec{\mathbf{v}}_2, \vec{\xi} \right) \,\middle|\, \left(x, \vec{\xi} \right) \in WF\left(u \right), \vec{\xi} \in \vec{\mathbf{v}}_2^{\perp}, t > 0 \right\},$$

and

$$WF\left(u \right) \subseteq WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right)$$
$$\cup \left\{ \left(x + t\left(\vec{\mathbf{v}}_2 - \vec{\mathbf{v}}_1 \right), \vec{\xi} \right) \,\middle|\, \left(x, \vec{\xi} \right) \in WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right), \vec{\xi} \in \left(\vec{\mathbf{v}}_2 - \vec{\mathbf{v}}_1 \right)^{\perp}, t > 0 \right\}, \tag{4.2.7}$$

$$WF\left(u \right) \subseteq WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right)$$
$$\cup \left\{ \left(x + t\left(\vec{\mathbf{v}}_1 - \vec{\mathbf{v}}_2 \right), \vec{\xi} \right) \,\middle|\, \left(x, \vec{\xi} \right) \in WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right), \vec{\xi} \in \left(\vec{\mathbf{v}}_2 - \vec{\mathbf{v}}_1 \right)^{\perp}, t > 0 \right\}. \tag{4.2.8}$$

Moreover, in view of Theorem 3.1.9, if $\left(x_0, \vec{\xi}_0 \right) \in WF\left(u \right) \backslash WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right)$*, then* $\vec{\xi}_0 \in \left(\vec{\mathbf{v}}_2 - \vec{\mathbf{v}}_1 \right)^{\perp}$*, and* x_0 *must lie on some line segment* $x_0 + I\vec{\mathbf{v}}$ *for which* $\left(x_0 + I\vec{\mathbf{v}} \right) \times \left\{ \vec{\xi}_0 \right\} \subseteq WF\left(u \right)$*, and whose endpoints lie in* $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right)$*.*

The following example shows that $WF\left(u \right) \backslash WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right)$ can in fact be nonempty:

Example 4.2.2. A distribution $u \in \mathscr{E}'\left(\mathbb{R}^n \right)$ exists such that

$$WF\left(u \right) \nsubseteq WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right).$$

Proof. Let $\vec{\mathbf{v}}_1 = \left(\cos\theta, -\sin\theta \right)$ and $\vec{\mathbf{v}}_2 = \left(\cos\theta, \sin\theta \right)$. We now take $u \in \mathcal{L}^1\left(\mathbb{R}^2 \right)$ with

$$u\left(x, y \right) = \begin{cases} -1, & \text{if } -1 < x < 0,\ |y| < 1 + 2\tan\theta, \\ 1, & \text{if } 0 < x < 1,\ |y| < 1 + 2\tan\theta, \\ 0, & \text{otherwise,} \end{cases}$$

so that $WF\left(u \right)$ contains $\ell \times \{ \vec{\mathbf{e}}_1 \}$, where ℓ is the portion of the y-axis where $|y| \leq 1 + 2\tan\theta$. However, $\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u\left(x, y \right) = 0$ whenever $|y| < 1$, and so $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2} u \right)$ omits that part of the y-axis. □

A numerical inversion of a similar function in shown in Appendix A, Figure A.4.

Figure 4.1 depicts the wavefront of a distribution u in blue, illustrating the maximum extent to which $\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}$ will extend (in green) and remove (in red) singularities from u.

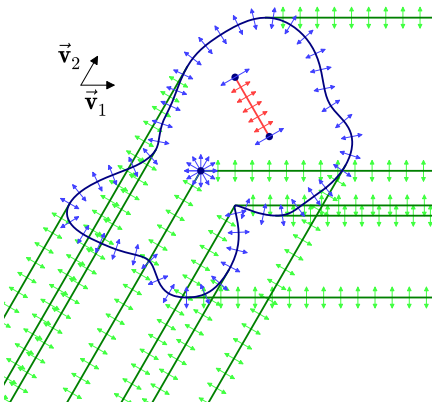

FIGURE 4.1: Visualisation of $WF\left(u\right)$, and $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}u\right)$.

4.2.2 Inversion and microlocal analysis with two sets of data with common incident beams

If we have two sets of data, $\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1}f$ and $\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2}f$, and we have

$$\vec{\mathbf{v}}_0 = a_1\vec{\mathbf{v}}_1 + a_2\vec{\mathbf{v}}_2, \tag{4.2.9}$$

for some coefficients a_1 and a_2 subject to the condition $a_1 + a_2 \neq 1$, and unit vectors $\vec{\mathbf{v}}_0$, $\vec{\mathbf{v}}_1$, and $\vec{\mathbf{v}}_2$, then the inversion from both sets of data turns out to be far simpler than the inversion from one set of data. Consider:

$$\begin{aligned} a_1\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1}f + a_2\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2}f &= a_1\mathcal{D}_{\vec{\mathbf{v}}_1}\left(\mathcal{J}_{\vec{\mathbf{v}}_0} + \mathcal{J}_{-\vec{\mathbf{v}}_1}\right)f + a_2\mathcal{D}_{\vec{\mathbf{v}}_2}\left(\mathcal{J}_{\vec{\mathbf{v}}_0} + \mathcal{J}_{-\vec{\mathbf{v}}_2}\right)f \\ &= a_1\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{J}_{\vec{\mathbf{v}}_0}f - a_1f + a_2\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{J}_{\vec{\mathbf{v}}_0}f - a_2f \\ &= \mathcal{D}_{a_1\vec{\mathbf{v}}_1+a_2\vec{\mathbf{v}}_2}\mathcal{J}_{\vec{\mathbf{v}}_0}f - a_1f - a_2f \\ &= f - a_1f - a_2f, \end{aligned}$$

which then yields the following inversion:

$$f = \frac{a_1\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1}f + a_2\mathcal{D}_{\vec{\mathbf{v}}_2}\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2}f}{1 - a_1 - a_2}. \tag{4.2.10}$$

The inversion formula, of course, requires that $a_1 + a_2 \neq 1$. If it were the case that equality did hold, then

$$\left\|a_1\vec{\mathbf{v}}_1 + a_2\vec{\mathbf{v}}_2\right\|^2 = a_1^2 + 2a_1a_2\vec{\mathbf{v}}_1 \cdot \vec{\mathbf{v}}_2 + a_2^2$$

$$
\begin{aligned}
&= a_1^2 + 2a_1a_2\vec{\mathbf{v}}_1 \cdot \vec{\mathbf{v}}_2 + a_2^2 \\
&= (a_1 + a_2)^2 + 2a_1a_2\vec{\mathbf{v}}_1 \cdot \vec{\mathbf{v}}_2 - 2a_1a_2 \\
&= 1 + 2a_1a_2 \left(\vec{\mathbf{v}}_1 \cdot \vec{\mathbf{v}}_2 - 1\right),
\end{aligned}
$$

which then would imply that either a_1 or a_2 would be zero, which would mean that $\vec{\mathbf{v}}_0$ is either $\vec{\mathbf{v}}_1$ or $\vec{\mathbf{v}}_2$, or that $\vec{\mathbf{v}}_1 \cdot \vec{\mathbf{v}}_2 = 1$, which would mean $\vec{\mathbf{v}}_1 = \vec{\mathbf{v}}_2$, all of which are uninteresting degenerate cases.

It follows from this inversion formula that

$$
WF\left(f\right) \subseteq WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1} f\right) \cup WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2} f\right),
$$

and so whatever $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1} f\right)$ omits from $WF\left(f\right)$, $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2} f\right)$ will not, and vice versa. Also, since $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1} f\right)$ and $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2} f\right)$ add at most $\mathbb{R} \times \left(\vec{\mathbf{v}}_0^{\perp} \cup \vec{\mathbf{v}}_1^{\perp}\right)$ and $\mathbb{R} \times \left(\vec{\mathbf{v}}_0^{\perp} \cup \vec{\mathbf{v}}_2^{\perp}\right)$, respectively, we can be certain that

$$
\begin{aligned}
&\left(WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1} f\right) \cup W_1\right) \cap \left(WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2} f\right) \cup W_2\right) \backslash V_0 \\
&\quad \subseteq WF\left(f\right) \\
&\quad \subseteq \left(WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1} f\right) \backslash V_1\right) \cup \left(WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2} f\right) \backslash V_2\right),
\end{aligned}
$$

where

$$
\begin{aligned}
V_k &= \mathbb{R}^n \times \left(\vec{\mathbf{v}}_k^{\perp} \backslash 0\right), k = 0, 1, 2, \\
W_k &= \mathbb{R}^n \times \left(\left(\vec{\mathbf{v}}_k - \vec{\mathbf{v}}_0\right)^{\perp} \backslash 0\right), k = 1, 2.
\end{aligned}
$$

The only degree of uncertainty lies in what portion of V_0 that $WF\left(f\right)$ will contain in relation to $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_1} f\right)$ and $WF\left(\mathcal{B}_{\vec{\mathbf{v}}_0,\vec{\mathbf{v}}_2} f\right)$.

4.3 Broken Ray transform with curved detectors

In Katsevich and Krylov's two-detector setting, [6], inversion is given by the following differential equation, in terms of pairwise differences between Broken Ray transforms:

$$(\ell_j - \ell_i)(1 - c_{ij}) f + \left(\mathcal{D}_{\vec{\beta}_i} - \mathcal{D}_{\vec{\beta}_j}\right) f = \mathcal{D}_{\vec{\beta}_j} \mathcal{D}_{\vec{\beta}_i} g_{ij} - (\ell_i c_{ij} + \ell_j) \mathcal{D}_{\vec{\beta}_i} g_{ij}, \tag{4.3.1}$$

where $c_{ij} = \vec{\beta}_i \cdot \vec{\beta}_j$, and ℓ_i satisfies

$$\mathcal{D}_{\vec{\beta}_i^{\perp}} \vec{\beta}_i = -\ell_i \vec{\beta}_i^{\perp},$$

and is in fact given by

$$\ell_i(x) = \begin{cases} 0, & \text{if detector } i \text{ is flat,} \\ -\|x - x_i\|^{-1}, & \text{if detector } i \text{ is concave, with focal point at } x_i, \\ \|x - x_i\|^{-1}, & \text{if detector } i \text{ is convex, with focal point at } x_i. \end{cases}$$

In the three-detector setting, an inversion is given directly as

$$f = -\frac{s_{32} \mathcal{D}_{\vec{\beta}_1} g_{12} + s_{21} \mathcal{D}_{\vec{\beta}_3} g_{32}}{s_{32} + s_{21} + s_{13}},$$

where $s_{jk} = \vec{\beta}_j \cdot \vec{\beta}_k^{\perp}$. If g_i is chosen for $i = 1, 2, 3$ so that $g_{jk} = g_j - g_k$, e.g., when g_i are three different measurements with a common initial beam for each scattering location x, then we can arrive at a more symmetric inversion:

$$f = -\frac{s_{32} \mathcal{D}_{\vec{\beta}_1} g_1 + s_{13} \mathcal{D}_{\vec{\beta}_2} g_2 + s_{21} \mathcal{D}_{\vec{\beta}_3} g_3}{s_{32} + s_{21} + s_{13}}. \tag{4.3.2}$$

Note that this inversion formula can also be used to verify (4.2.10) by using $\mathcal{B}_{\vec{\mathbf{v}}_0, \vec{\mathbf{v}}_0} f$ as the third set of data, even though $\mathcal{B}_{\vec{\mathbf{v}}_0, \vec{\mathbf{v}}_0}$ is not invertible. Notice, of course, that $\mathcal{D}_{\vec{\mathbf{v}}_0} \mathcal{B}_{\vec{\mathbf{v}}_0, \vec{\mathbf{v}}_0} f = 0$.

4.3.1 Solutions to (4.3.1)

In this subsection, we shall use x as a coordinate variable instead of as a point in an open set. Furthermore, we shall let

$$h_{ij} = \mathcal{D}_{\vec{\beta}_j}\mathcal{D}_{\vec{\beta}_i} g_{ij} - (\ell_i c_{ij} + \ell_j)\, \mathcal{D}_{\vec{\beta}_i} g_{ij}.$$

By reversing the roles of i and j in (4.3.1), and performing the appropriate simplifications, we will find that

$$h_{ij} = \mathcal{D}_{\vec{\beta}_j}\mathcal{D}_{\vec{\beta}_i} g_{ij} - (\ell_j c_{ij} + \ell_i)\, \mathcal{D}_{\vec{\beta}_j} g_{ij}.$$

In the case that both detectors are flat in Katsevich and Krylov, then the ℓ_i vanish, and the inversion is identical to that of $\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}$ given in (4.2.1), with $\vec{\mathbf{v}}_i = \vec{\beta}_i$. We will solve (4.3.1) for f in the various cases where at least one detector is curved using the method of characteristics.

Concave and flat detectors

Suppose detector 1 is concave, detector 2 is flat, and assume without loss of generaltiy that the focal point of detector 1 is at the origin, and detector 2 detects rays in the direction $\vec{\mathbf{e}}_1$. This gives us $\vec{\beta}_1 = \frac{z}{\|z\|}$ and $\vec{\beta}_2 = \vec{\mathbf{e}}_1$. Generalizations from this case is obtained via translations, and rotations. Consider the parabolic coordinate system, given by the change of variables

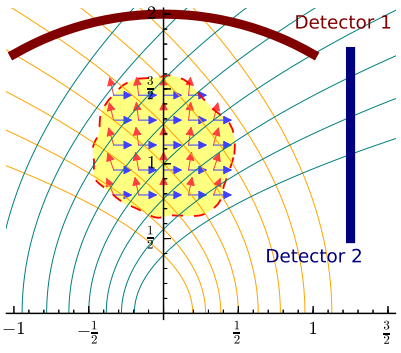

FIGURE 4.2: Concave and flat detectors with $\sigma\tau$ grid.

$$z = \begin{bmatrix} x \\ y \end{bmatrix} = \Phi(\sigma,\tau) \overset{\text{def}}{=} \begin{bmatrix} \frac{1}{2}\left(\sigma^2 - \tau^2\right) \\ \sigma\tau \end{bmatrix},$$

$$\sigma \in \mathbb{R}, \tau > 0.$$

This coordinate system is a variation on that given in Hazewinkel's Encyclopedia of Mathematics, [4]. Note that in terms of complex variables, this is the same as $z = w^2$ for $\Re w > 0$. The coordinate grid consists of parabolas opening either to the right or left, with focus at the origin. Notice that

$$\begin{aligned}
\vec{\beta}_2 - \vec{\beta}_1 &= \vec{\mathbf{e}}_1 - \frac{z}{\|z\|} \\
&= \vec{\mathbf{e}}_1 - \frac{2z}{\sqrt{\left(\sigma^2-\tau^2\right)^2 + 4\sigma^2\tau^2}} \\
&= \frac{\left(\sigma^2+\tau^2\right)\vec{\mathbf{e}}_1 - 2z}{\sigma^2+\tau^2} \\
&= \frac{1}{\sigma^2+\tau^2} \cdot \begin{bmatrix} \left(\sigma^2+\tau^2\right) - \left(\sigma^2-\tau^2\right) \\ -2\sigma\tau \end{bmatrix} \\
&= \frac{1}{\sigma^2+\tau^2} \cdot \begin{bmatrix} 2\tau^2 \\ -2\sigma\tau \end{bmatrix} \\
&= -\frac{2\tau}{\sigma^2+\tau^2} \cdot \frac{\partial \Phi}{\partial \tau},
\end{aligned}$$

and so $\mathcal{D}_{\vec{\beta}_2} - \mathcal{D}_{\vec{\beta}_1} = -\frac{2\tau}{\sigma^2+\tau^2} \cdot \frac{\partial}{\partial \tau}$. Thus, in this coordinate system, (4.3.1) with simplification, and multiplication by the proper integrating factor becomes

$$2\tau \cdot f + \left(\sigma^2+\tau^2\right) \cdot \frac{\partial}{\partial \tau} f = \frac{\left(\sigma^2+\tau^2\right)^2}{2\tau} \cdot h_{21}, \tag{4.3.3}$$

which has solution

$$f(x,y) = \frac{\sigma^2+\tau_0^2}{\sigma^2+\tau^2} \cdot f|_{\tau=\tau_0} + \frac{1}{\sigma^2+\tau^2} \int_{\tau_0}^{\tau} \frac{\left(\sigma^2+t^2\right)^2}{2t} \Phi^\star h_{21}(\sigma,t)\, dt, \tag{4.3.4}$$

where $f|_{\tau=\tau_0}$ is understood as $\Phi^\star f(\sigma,\tau_0)$. It should be noted that if f is at least $\mathcal{C}^1$ in a neighborhood of of $\tau = 0$, then h_{21} will vanish at $\tau = 0$, which would then imply convergence of the above integral if we choose $\tau_0 = 0$.

If f is supported away from positive x-axis, then

$$f(x,y) = \frac{1}{\sigma^2+\tau^2} \int_{0}^{\tau} \frac{\left(\sigma^2+t^2\right)^2}{2t} \Phi^\star h_{21}(\sigma,t)\, dt. \tag{4.3.5}$$

Viewing this inversion formula in a distributional sense, however, will require that the support of h_{21} be bounded away from $\tau = 0$, so that the division in the above integrand by $2t$, and also by $\sigma^2 + \tau^2$ after the integration, are both valid. Notice that in this setting, while we may require that the support of f is bounded away from $\tau = 0$, g_{ij} will not generally inherit this requirement. We note, however, that $\mathcal{D}_{\vec{\beta}_1} g_{21} = -f - \mathcal{D}_{\vec{\beta}_1} \mathcal{J}_{-\vec{\beta}_2} f$, and since $\mathcal{J}_{-\vec{\beta}_2}$ spreads support only in the horizontal direction in the xy-plane, the desired support condition on h_{21} is satisfied.

We then interpret integration from 0 to τ in a distributional sense by considering $\Phi^\star h_{21}$ as a distribution on the entire $\sigma\tau$-plane whose support is contained in the open half-space $\{\tau > 0\}$. After multiplication of the integrand by $\frac{\left(\sigma^2+t^2\right)^2}{2t}$, we apply $\mathcal{J}_{(0,1)}$, resulting in a distribution also supported in the open half space $\{\tau > 0\}$.

Convex and flat detectors

If instead, detector 1 was convex, we can still use the parabolic coordinates, but instead imposing the domain restriction $\sigma > 0$, $\tau \in \mathbb{R}$ on the formula given for Φ. We then have $\vec{\beta}_1 = -\frac{z}{\|z\|}$, and find that

$$\begin{aligned}
\vec{\beta}_2 - \vec{\beta}_1 &= \vec{\mathbf{e}}_1 + \frac{z}{\|z\|} \\
&= \vec{\mathbf{e}}_1 + \frac{2z}{\sqrt{\left(\sigma^2 - \tau^2\right)^2 + 4\sigma^2\tau^2}} \\
&= \frac{\left(\sigma^2 + \tau^2\right)\vec{\mathbf{e}}_1 + 2z}{\sigma^2 + \tau^2} \\
&= \frac{1}{\tau^2 + \sigma^2} \cdot \begin{bmatrix} \left(\sigma^2 - \tau^2\right) + \left(\tau^2 + \sigma^2\right) \\ 2\sigma\tau \end{bmatrix} \\
&= \frac{1}{\tau^2 + \sigma^2} \cdot \begin{bmatrix} 2\sigma^2 \\ 2\sigma\tau \end{bmatrix}
\end{aligned}$$

$$= \frac{2\sigma}{\tau^2+\sigma^2} \cdot \frac{\partial \Phi}{\partial \sigma}.$$

Thus, $\mathcal{D}_{\vec{\beta}_2} - \mathcal{D}_{\vec{\beta}_1} = \frac{2\sigma}{\sigma^2+\tau^2} \cdot \frac{\partial}{\partial \sigma}$, so we then write (4.3.1) as

$$2\sigma \cdot f + \left(\sigma^2+\tau^2\right) \cdot \frac{\partial}{\partial \sigma} f = -\frac{\left(\sigma^2+\tau^2\right)^2}{2\tau} \cdot h_{21}, \tag{4.3.6}$$

which has solution

$$f(x,y) = \frac{\sigma_0^2+\tau^2}{\sigma^2+\tau^2} \cdot f|_{\sigma=\sigma_0} - \frac{1}{\sigma^2+\tau^2} \int_{\sigma_0}^{\sigma} \frac{\left(s^2+\tau^2\right)^2}{2s} \Phi^\star h_{21}(s,\tau)\, ds, \tag{4.3.7}$$

and if f is supported away from the negative x-axis, then

$$f(x,y) = -\frac{1}{\sigma^2+\tau^2} \int_{0}^{\sigma} \frac{\left(s^2+\tau^2\right)^2}{2s} \Phi^\star h_{21}(s,\tau)\, ds. \tag{4.3.8}$$

In the same manner determined in the concave/flat setting, we impose a support restriction on f, this time requiring the support be bounded away from $\sigma = 0$.

Two concave detectors

Now suppose both detectors are concave, and assume without loss of generality that the focal points are at $e_1 = (1,0)$ and $-e_1 = (-1,0)$, giving us $\vec{\beta}_1 = \frac{z-e_1}{\|z-e_1\|}$ and $\vec{\beta}_2 = \frac{z+e_1}{\|z+e_1\|}$. Generalizing to an arbitrary pair of focal points entails using translations, dialations, and rotations.

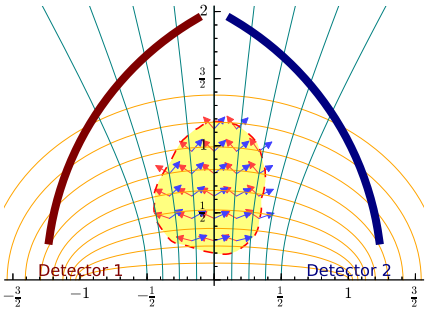

FIGURE 4.3: Two concave detectors with $\mu\nu$ grid.

We consider use of the elliptical coordinate system, [4], given by the change of variables

$$z = \begin{bmatrix} x \\ y \end{bmatrix} = \Psi(\mu,\nu) \stackrel{\text{def}}{=} \begin{bmatrix} \cosh\mu \cos\nu \\ \sinh\mu \sin\nu \end{bmatrix}, \quad \mu \geq 0, 0 \leq \nu \leq 2\pi. \tag{4.3.9}$$

In terms of complex variables, this is equivalent to $z = \cosh w$ for $0 \leq \Im w \leq 2\pi$. The coordinate grid consists of ellipses and hyperbolas whose focal points are e_1 and $-e_1$. After use of trigonometric and hyperbolic trigonometric identities, we find that $\|z \pm e_1\| = \cosh\mu \pm \cos\nu$. Thus, in the case that both detectors are concave, we compute

$$\begin{aligned}
\vec{\beta}_2 - \vec{\beta}_1 &= \frac{z+e_1}{\|z+e_1\|} - \frac{z-e_1}{\|z-e_1\|} \\
&= \frac{z+e_1}{\cosh\mu + \cos\nu} - \frac{z-e_1}{\cosh\mu - \cos\nu} \\
&= \frac{(\cosh\mu - \cos\nu)(z+e_1) - (\cosh\mu + \cos\nu)(z-e_1)}{\cosh^2\mu - \cos^2\nu} \\
&= 2\frac{e_1\cosh\mu - z\cos\nu}{\cosh^2\mu - \cos^2\nu} \\
&= \frac{2}{\cosh^2\mu - \cos^2\nu}\begin{bmatrix}\cosh\mu - \cosh\mu\cos^2\nu \\ -\sinh\mu\sin\nu\cos\nu\end{bmatrix} \\
&= \frac{2}{\cosh^2\mu - \cos^2\nu}\begin{bmatrix}\cosh\mu\sin^2\nu \\ -\sinh\mu\sin\nu\cos\nu\end{bmatrix} \\
&= \frac{-2\sin\nu}{\cosh^2\mu - \cos^2\nu}\begin{bmatrix}-\cosh\mu\sin\nu \\ \sinh\mu\cos\nu\end{bmatrix} \\
&= \frac{-2\sin\nu}{\cosh^2\mu - \cos^2\nu}\cdot\frac{\partial\Psi}{\partial\nu}.
\end{aligned}$$

Hence, $\mathcal{D}_{\vec{\beta}_2} - \mathcal{D}_{\vec{\beta}_1} = \frac{-2\sin\nu}{\cosh^2\mu - \cos^2\nu}\cdot\frac{\partial}{\partial\nu}$, and the characteristic curves associated with the differential equation (4.3.1) are in fact the aforementioned ellipses. We can therefore write (4.3.1) in this case as

$$2\sin\nu\cos\nu\cdot f + \left(\cosh^2\mu - \cos^2\nu\right)\cdot\frac{\partial}{\partial\nu}f = \frac{\left(\cosh^2\mu - \cos^2\nu\right)^2}{2\sin\nu}\cdot h_{21},$$

which has solution

$$\begin{aligned}
f(x,y) = {} & \frac{\cosh^2\mu - \cos^2\nu_0}{\cosh^2\mu - \cos^2\nu} f|_{\nu=\nu_0} \\
& + \frac{1}{\cosh^2\mu - \cos^2\nu}\int_{\nu_0}^{\nu}\frac{1}{\sin t}\frac{\left(\cosh^2\mu - \cos^2 t\right)^2}{2}\Psi^\star h_{21}(\mu,t)\,dt.
\end{aligned}$$

Given that the characteristic curves are ellipses, this suggests that the mapping $f \mapsto g_{ij}$ is not injective. Non-injectivity is verfied by choosing

$$f(x,y) = \frac{\cosh^2 \mu - 1}{\cosh^2 \mu - \cos^2 \nu} f_0(\mu),$$

where f_0 is a choice of boundary conditions on $\eta = 0$. consider the following reparametrization of (1.1.3):

$$g_{21}(z_0) = \int_{p_0}^{\infty} f\left(-e_1 + p\vec{\beta}_2\right) dp - \int_{q_0}^{\infty} f\left(e_1 + q\vec{\beta}_1\right) dq, \tag{4.3.10}$$

where p_0, q_0, $\vec{\beta}_1$, and $\vec{\beta}_2$ are held to the constraint

$$z_0 = \Psi(\mu_0, \eta_0) = -e_1 + p_0\vec{\beta}_2 = e_1 + q_0\vec{\beta}_1.$$

With regards to the ray along which we integrate f in the first integral,

$$-e_1 + p\vec{\beta}_2 = \Psi(\mu, \nu)$$

defines both μ and ν implicitly as functions of p, and implicit differentiation with respect to p yields

$$\vec{\beta}_2 = \mathcal{D}\Psi(\mu,\nu) \cdot \begin{bmatrix} \frac{\partial \mu}{\partial p} & \frac{\partial \nu}{\partial p} \end{bmatrix},$$

and so

$$\frac{\partial \mu}{\partial p} = \pi_1 \mathcal{D}\Psi(\mu,\nu)^{-1} \vec{\beta}_2 = \frac{\sinh \mu}{\cosh^2 \mu - \cos^2 \nu}.$$

Hence, a substitution allows us to express the first integral in (4.3.10) as follows:

$$\begin{aligned}
\int_{p_0}^{\infty} f\left(-e_1 + p\vec{\beta}_2\right) dp &= \int_{p_0}^{\infty} f(\Psi(\mu,\nu))\, dp \\
&= \int_{p_0}^{\infty} \frac{\cosh^2 \mu - 1}{\cosh^2 \mu - \cos^2 \nu} f_0(\mu)\, dp \\
&= \int_{p_0}^{\infty} \sinh \mu f_0(\mu) \frac{\partial \mu}{\partial p}\, dp \\
&= \int_{\mu_0}^{\infty} \sinh \mu f_0(\mu)\, d\mu.
\end{aligned}$$

A similar substitution in the second integral in (4.3.10) will yield the same integral in μ, and hence the two integrals will cancel. Hence, knowledge of boundary data independent of knowledge of g_{ij} is necessary. Knowledge that f is supported away from $\nu = 0$ or $\nu = \pi$ will be sufficient. Notice that $\nu = 0$ and $\nu = \pi$ are parts of the x-axis, the former being where $x > 1$, and the latter corresponding to $x < -1$.

Thus, for f supported away from $\nu = 0$, we write

$$f(x,y) = \frac{1}{\cosh^2\mu - \cos^2\nu}\int_0^\nu \frac{1}{\sin t}\frac{\left(\cosh^2\mu - \cos^2 t\right)^2}{2}\Psi^\star h_{21}(\mu,t)\,dt,$$

whereas for f supported away from $\nu = \pi$, we write

$$f(x,y) = -\frac{1}{\cosh^2\mu - \cos^2\nu}\int_\nu^\pi \frac{1}{\sin t}\frac{\left(\cosh^2\mu - \cos^2 t\right)^2}{2}\Psi^\star h_{21}(\mu,t)\,dt.$$

Two convex detectors

Not surprisingly, the case of two convex detectors is not exceedingly different from two convex detectors. With the focal points still kept at e_1 and e_2, we find that $\mathcal{D}_{\vec{\beta}_2} - \mathcal{D}_{\vec{\beta}_1}$ is merely the negation of that given in the concave case. Therefore, the characteristics remain the same as those in the concave/concave case. Thus, the result is that the inversion instead becomes

$$\begin{aligned} f(x,y) = {} & \frac{\cosh^2\mu - \cos^2\nu_0}{\cosh^2\mu - \cos^2\nu} f|_{\nu=\nu_0} \\ & - \frac{1}{\cosh^2\mu - \cos^2\nu}\int_{\nu_0}^\nu \frac{1}{\sin t}\frac{\left(\cosh^2\mu - \cos^2 t\right)^2}{2}\Psi^\star h_{21}(\mu,t)\,dt. \end{aligned}$$

This case still suffers from the same non-injectivity as the concave case in the absense of known boundary data. A similar computation will verify that g_{ij} vanishes if $f(x,y) = \frac{\cosh^2\mu - \cos^2\nu_0}{\cosh^2\mu - \cos^2\nu} f_0(\mu)$.

In the case of the detectors being either both concave or both convex, requiring the support of f to be compact and bounded away from the x-axis will guarantee that h_{ij} be supported away from the x-axis as well, which makes multiplication in the distributional sense by $\frac{1}{\sin\nu}$ and $\frac{1}{\cosh^2\mu-\cos^2\nu}$ possible.

Concave and convex detectors, different focal points

While the unmixed concave/concave and convex/convex cases presented above are strikingly similar to each other, the mixed case presents a different geometry, with the only commonality with the unmix case being that we may continue to use the elliptic coordinate system. In the mixed case, we will let detector 1 be concave and detector 2 convex, with focal points placed at e_1 and $-e_1$ as before. We obtain through a similar computation that

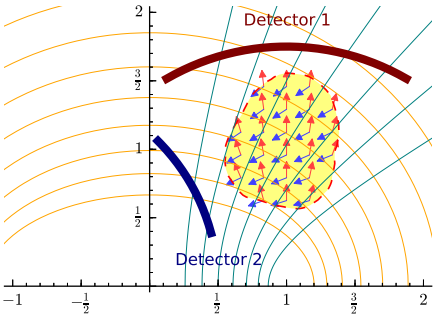

FIGURE 4.4: Concave and convex detectors with $\mu\nu$ grid.

$$\begin{aligned}
\vec{\beta}_2 - \vec{\beta}_1 &= \frac{z+e_1}{\|z+e_1\|} + \frac{z-e_1}{\|z-e_1\|} \\
&= \frac{z+e_1}{\cosh\mu+\cos\nu} + \frac{z-e_1}{\cosh\mu-\cos\nu} \\
&= \frac{(\cosh\mu-\cos\nu)(z+e_1)+(\cosh\mu+\cos\nu)(z-e_1)}{\cosh^2\mu-\cos^2\nu} \\
&= 2\frac{z\cosh\mu - e_1\cos\nu}{\cosh^2\mu-\cos^2\nu} \\
&= \frac{2}{\cosh^2\mu-\cos^2\nu}\begin{bmatrix}\cosh^2\mu\cos\nu-\cos\nu\\ \sinh\mu\cosh\mu\sin\nu\end{bmatrix} \\
&= \frac{2}{\cosh^2\mu-\cos^2\nu}\begin{bmatrix}\sinh^2\mu\cos\nu\\ \sinh\mu\cosh\mu\sin\nu\end{bmatrix}
\end{aligned}$$

$$= \frac{2\sinh\mu}{\cosh^2\mu - \cos^2\nu} \begin{bmatrix} \sinh\mu\cos\nu \\ \cosh\mu\sin\nu \end{bmatrix}$$
$$= \frac{2\sinh\mu}{\cosh^2\mu - \cos^2\nu} \frac{\partial\Psi}{\partial\mu}.$$

Therefore, the characteristic curves are then the hyperbolas, and as a result, no boundary data is necessary if f is known to have compact support.

In this coordinate system, (4.3.1) becomes

$$\frac{2\cosh\mu}{\cosh^2\mu - \cos^2\nu} \cdot \frac{2\sinh^2\mu}{\cosh^2\mu - \cos^2\nu} \cdot f + \frac{2\sinh\mu}{\cosh^2\mu - \cos^2\nu} \cdot \frac{\partial f}{\partial\mu} = h_{21},$$

and multiplication by $\frac{\left(\cosh^2\mu - \cos^2\nu\right)^2}{2\sinh\mu}$, we obtain

$$2\sinh\mu\cosh\mu \cdot f + \left(\cosh^2\mu - \cos^2\nu\right) \cdot \frac{\partial f}{\partial\mu} = -\frac{\left(\cosh^2\mu - \cos^2\nu\right)^2}{2\sinh\mu} h_{21}$$

and so a solution is

$$f = \frac{\cosh^2\mu_0 - \cos^2\nu}{\cosh^2\mu - \cos^2\nu} f|_{\mu=\mu_0}$$
$$- \frac{1}{\cosh^2\mu - \cos^2\nu} \int_{\mu_0}^{\mu} \frac{\left(\cosh^2 t - \cos^2\nu\right)^2}{2\sin t} \Psi^\star h_{21}(t,\nu)\, dt.$$

Concave and convex detectors, common focal point

Finally, if the focal points are at the origin, then g_{21} is obtained by integrating f in the radial direction as follows:

$$g_{21}\left(r\vec{\theta}\right) = \int_r^\infty f\left(t\vec{\theta}\right) dt - \int_0^r f\left(t\vec{\theta}\right) dt, \quad r > 0, \vec{\theta} \in \mathcal{S}^1.$$

Hence, we can short circuit (4.3.1) and immediately see that

$$f = -\frac{1}{2}\frac{\partial}{\partial r} g_{21}.$$

4.3.2 Microlocal analysis of the Broken Ray transform with curved detectors

Define the following pair of transforms:

$$\breve{\mathcal{J}} f\left(r\vec{\theta}\right) = \int_0^r f\left(t\vec{\theta}\right)\, dt, \quad \hat{\mathcal{J}} f\left(r\vec{\theta}\right) = \int_r^\infty f\left(t\vec{\theta}\right)\, dt,$$
$$r > 0, \vec{\theta} \in \mathcal{S}^1, f \in \mathcal{C}_0^0\left(\mathbb{R}^2 \backslash 0\right).$$

We observe that $\breve{\mathcal{J}}$ models a convex detector, while $\hat{\mathcal{J}}$ models a concave detector, with the origin serving as the focus for both. We can use an arbitrary point x_0 in $\mathbb{R}^2$ as a focus by conjugating these transforms with a translation, defining

$$\breve{\mathcal{J}}_{x_0} f\left(x_0 + r\vec{\theta}\right) = \int_0^r f\left(x_0 + t\vec{\theta}\right)\, dt,$$
$$\hat{\mathcal{J}}_{x_0} f\left(x_0 + r\vec{\theta}\right) = \int_r^\infty f\left(x_0 + t\vec{\theta}\right)\, dt,$$

for $r > 0$, $\vec{\theta} \in \mathcal{S}^1$, and $f \in \mathcal{C}_0^0\left(\mathbb{R}^2 \backslash \{x_0\}\right)$. We can then characterize each g_{ij} as the difference between any pair of transforms of the form $\mathcal{J}_{\vec{\mathbf{v}}}$, $\breve{\mathcal{J}}_{x_0}$, and $\hat{\mathcal{J}}_{x_0}$.

Since $\breve{\mathcal{J}}_{x_0}$ and $\hat{\mathcal{J}}_{x_0}$ differ from $\breve{\mathcal{J}}$ and $\hat{\mathcal{J}}$ by only conjugation with a translation, it will suffice to examine $\breve{\mathcal{J}}$ and $\hat{\mathcal{J}}$. If we pull back f, $\breve{\mathcal{J}} f$, and $\hat{\mathcal{J}} f$ by the map $\Psi(\rho, \theta) = (e^\rho \cos\theta, e^\rho \sin\theta)$, and performing the proper substitutions, we obtain

$$\Psi^\star \breve{\mathcal{J}} f(\rho, \theta) = \int_{-\infty}^\rho e^\tau \Psi^\star f(\tau, \theta)\, d\tau,$$
$$\Psi^\star \hat{\mathcal{J}} f(\rho, \theta) = \int_\rho^\infty e^\tau \Psi^\star f(\tau, \theta)\, d\tau.$$

This gives us a method of defining both $\breve{\mathcal{J}}$ and $\hat{\mathcal{J}}$ in a distributional sense when $f \in \mathscr{E}'\left(\mathbb{R}^n \backslash 0\right)$:

$$\breve{\mathcal{J}} = \Psi^{-\star} \mathcal{J}_{(1,0)}\left(e^\tau \Psi^\star f\right), \quad \hat{\mathcal{J}} = \Psi^{-\star} \mathcal{J}_{(-1,0)}\left(e^\tau \Psi^\star f\right).$$

Hence, we can identify the following wavefront set relationships:

$$WF\left(\Psi^{\star}\breve{\mathcal{J}}f\right) \subseteq WF\left(\Psi^{\star}f\right) \cup \left\{\left((\rho+t,\theta),(0,\eta)\right) \mid \left((\rho,\theta),(0,\eta)\right) \in WF\left(\Psi^{\star}f\right), t>0\right\},$$

$$WF\left(\Psi^{\star}\widehat{\mathcal{J}}f\right) \subseteq WF\left(\Psi^{\star}f\right) \cup \left\{\left((\rho-t,\theta),(0,\eta)\right) \mid \left((\rho,\theta),(0,\eta)\right) \in WF\left(\Psi^{\star}f\right), t>0\right\}.$$

Undoing the pullback, we arrive at:

$$WF\left(\breve{\mathcal{J}}f\right) \subseteq WF\left(f\right) \cup \left\{\left(tx,\vec{\xi}\right) \,\middle|\, \left(x,\vec{\xi}\right) \in WF\left(f\right), t>1, \vec{\xi} \in x^{\perp}\right\},$$

$$WF\left(\widehat{\mathcal{J}}f\right) \subseteq WF\left(f\right) \cup \left\{\left(tx,\vec{\xi}\right) \,\middle|\, \left(x,\vec{\xi}\right) \in WF\left(f\right), 0<t<1, \vec{\xi} \in x^{\perp}\right\}.$$

Then by pulling back by translation, we obtain the following:

$$\begin{aligned} WF\left(\breve{\mathcal{J}}_{x_0}f\right) &\subseteq WF\left(f\right) \\ &\quad \cup \left\{\left(t\left(x-x_0\right)+x_0,\vec{\xi}\right) \,\middle|\, \left(x,\vec{\xi}\right) \in WF\left(f\right), t>1, \vec{\xi} \in \left(x-x_0\right)^{\perp}\right\}, \\ WF\left(\widehat{\mathcal{J}}_{x_0}f\right) &\subseteq WF\left(f\right) \\ &\quad \cup \left\{\left(t\left(x-x_0\right)+x_0,\vec{\xi}\right) \,\middle|\, \left(x,\vec{\xi}\right) \in WF\left(f\right), 0<t<1, \vec{\xi} \in \left(x-x_0\right)^{\perp}\right\}. \end{aligned}$$

Hence, with f and g_{ij} as given in (1.1.3), regardless of the curvature of the detectors, we have

$$\begin{aligned} WF\left(g_{ij}\right) &\subseteq WF\left(f\right) \\ &\quad \cup \left\{\left(x-t\vec{\beta}_1\left(x\right),\vec{\xi}\right) \,\middle|\, \left(x,\vec{\xi}\right) \in WF\left(f\right), \vec{\xi} \in \vec{\beta}_1^{\perp}\left(x\right), t \geq 0\right\} \\ &\quad \cup \left\{\left(x-t\vec{\beta}_2\left(x\right),\vec{\xi}\right) \,\middle|\, \left(x,\vec{\xi}\right) \in WF\left(f\right), \vec{\xi} \in \vec{\beta}_2^{\perp}\left(x\right), t \geq 0\right\}. \end{aligned}$$

In the case that a detector is convex, then one must include the restriction $t<\|x\|$ in the corresponding part of the formula above.

As much as inversion of $\mathcal{B}_{\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2}$ propagates singularities along rays in the direction $\vec{\mathbf{v}}_2-\vec{\mathbf{v}}_1$, inversion of (4.3.1) will propagate singularities along the parabolas, hyperbolas, and ellipses over which the inversion formulas found in Section 4.3.1. Since the inversion formula for the mixed convex-concave case with common origin is given only by differentiation, inversion in that case propagates no singularities. We will omit the details of these results.

5 THE POLAR BROKEN RAY TRANSFORM

Of particular interest is investigating conditions for which f can be recovered from $\mathcal{Q}f$, as defined in (1.2.1). An immediate observation that uniqueness of f is known if the support in θ of f is sufficiently small, since this then the limited angle X-ray transform problem. Specifically, if the support of f contained in the sector $0 \leq \theta \leq \theta_{\max}$ with $\theta_{\max} < \phi$, then for $0 < \theta < \phi$,

$$\mathcal{P}_{\vec{\theta}} f\,(p) = \mathcal{Q}f\left(p \csc\phi \cdot A^{-1}\vec{\theta}\right), \quad \vec{\theta} = (\cos\theta, \sin\theta)\,.$$

However, reconstructions of f in the limited angle problem are unstable.

5.1 Mapping Properties

Our next few observations will be of mapping properties of $\mathcal{Q}$.

Proposition 5.1.1. *Let $p \geq 1$, and $\hat{w} : [0, r_{\max}] \to \mathbb{R}$ be a nonnegative weight function, and define*

$$w\,(r) = \frac{1}{r}\int_r^{r_{\max}} s^{2-1/p}\hat{w}\,(s)\;ds + \int_0^r \frac{s\left(\sqrt{r_{\max}^2 - s^2\sin^2\phi} - s\cos\phi\right)^{1-1/p}\hat{w}\,(s)}{\sqrt{r^2 - s^2\sin^2\phi}}\,ds, \quad r_{\min} \leq r \leq r_{\max}.$$

Then $\mathcal{Q}$ continuously maps $\mathcal{L}^p\,(\Omega, w)$ to $\mathcal{L}^p\left(\hat{\Omega}, \hat{w}\right)$.

Proof. We let $f \in \mathcal{L}^p\,(\Omega, w)$, set $t_{\max} = \sqrt{r_{\max}^2 - s^2\sin^2\phi} - s\cos\phi$ and observe:

$$\|\mathcal{Q}f\|^p_{\mathcal{L}^p(\hat{\Omega},\hat{w})} \int_{\mathcal{S}^1}\int_0^{r_{\max}} s\,|\mathcal{Q}f\,(s\vec{\sigma})|^p\,\hat{w}\,(s)\;ds\,d\vec{\sigma}$$

$$
\begin{aligned}
&= \int_{\mathcal{S}^1} \int_0^{r_{\max}} s \left| \int_0^s f\left(t\vec{\sigma}\right)\, dt + \int_0^{t_{\max}} f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \right|^p \hat{w}\left(s\right)\, ds\, d\vec{\sigma} \\
&\leq 2^{p-1} \left(\int_{\mathcal{S}^1} \int_0^{r_{\max}} s \left| \int_0^s f\left(t\vec{\sigma}\right)\, dt \right|^p \hat{w}\left(s\right)\, ds\, d\vec{\sigma} \right. \\
&\qquad \left. + \int_{\mathcal{S}^1} \int_0^{r_{\max}} s \left| \int_0^{t_{\max}} f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \right|^p \hat{w}\left(s\right)\, ds\, d\vec{\sigma} \right) \\
&\leq 2^{p-1} \left(\int_{\mathcal{S}^1} \int_0^{r_{\max}} s \cdot s^{1-1/p} \int_0^s \left| f\left(t\vec{\sigma}\right) \right|^p\, dt \hat{w}\left(s\right)\, ds\, d\vec{\sigma} \right. \\
&\qquad \left. + \int_{\mathcal{S}^1} \int_0^{r_{\max}} s t_{\max}^{1-1/p} \int_0^{\infty} \left| f\left(s\vec{\sigma} + tA\vec{\sigma}\right) \right|^p\, dt \hat{w}\left(s\right)\, ds\, d\vec{\sigma} \right) \\
&\leq 2^{p-1} \left(\int_{\mathcal{S}^1} \int_0^{r_{\max}} \int_0^s s^{2-1/p} \left| f\left(t\vec{\sigma}\right) \right|^p \hat{w}\left(s\right)\, dt\, ds\, d\vec{\sigma} \right. \\
&\qquad \left. + \int_{\mathcal{S}^1} \int_0^{r_{\max}} \int_0^{t_{\max}} s t_{\max}^{1-1/p} \left| f\left(s\vec{\sigma} + tA\vec{\sigma}\right) \right|^p \hat{w}\left(s\right)\, dt\, ds\, d\vec{\sigma} \right) \\
&= 2^{p-1} \left(\int_0^{r_{\max}} s\hat{w}\left(s\right) \int_{\mathcal{S}^1} \int_0^s \left| f\left(t\vec{\sigma}\right) \right|^p\, dt\, d\vec{\sigma}\, ds \right. \\
&\qquad \left. + \int_0^{r_{\max}} s t_{\max}^{1-1/p} \hat{w}\left(s\right) \int_{\mathcal{S}^1} \int_0^{t_{\max}} \left| f\left(s\vec{\sigma} + tA\vec{\sigma}\right) \right|^p\, dt\, d\vec{\sigma}\, ds \right).
\end{aligned}
$$

At this point, we shall relabel t and $\vec{\sigma}$ as r and $\vec{\theta}$ in the first integral, and make the multivariable substitution $r\vec{\theta} = s\vec{\sigma} + tA\vec{\sigma}$ in the second integral, holding s as a parameter:

$$
\begin{aligned}
\left\| \mathcal{Q}f \right\|_{\mathcal{L}^p\left(\hat{\Omega},\hat{w}\right)}^p &\leq 2^{p-1} \left(\int_0^{r_{\max}} s^{2-1/p} \hat{w}\left(s\right) \int_{\mathcal{S}^1} \int_0^s \left| f\left(r\vec{\theta}\right) \right|^p\, dr\, d\vec{\theta}\, ds \right. \\
&\qquad \left. + \int_0^{r_{\max}} s t_{\max}^{1-1/p} \hat{w}\left(s\right) \int_{\mathcal{S}^1} \int_s^{r_{\max}} \left| f\left(r\vec{\theta}\right) \right|^p \cdot \frac{r}{\sqrt{r^2 - s^2 \sin^2 \phi}}\, dr\, d\vec{\theta}\, ds \right) \\
&= 2^{p-1} \left(\int_{\mathcal{S}^1} \int_0^{r_{\max}} \left| f\left(r\vec{\theta}\right) \right|^p \int_r^{r_{\max}} s^{2-1/p} \hat{w}\left(s\right)\, ds\, dr\, d\vec{\theta} \right. \\
&\qquad \left. + \int_{\mathcal{S}^1} \int_0^{r_{\max}} \left| f\left(r\vec{\theta}\right) \right|^p \cdot \int_0^r \frac{r t_{\max}^{1-1/p}}{\sqrt{r^2 - s^2 \sin^2 \phi}} s\hat{w}\left(s\right)\, ds\, dr\, d\vec{\theta} \right) \\
&= 2^{p-1} \left(\int_{\mathcal{S}^1} \int_0^{r_{\max}} r \left| f\left(r\vec{\theta}\right) \right|^p \cdot \frac{1}{r} \int_r^{r_{\max}} s^{2-1/p} \hat{w}\left(s\right)\, ds\, dr\, d\vec{\theta} \right. \\
&\qquad \left. + \int_0^{r_{\max}} \int_{\mathcal{S}^1} r \left| f\left(r\vec{\theta}\right) \right|^p \cdot \int_0^r \frac{s t_{\max}^{1-1/p} \hat{w}\left(s\right)}{\sqrt{r^2 - s^2 \sin^2 \phi}}\, ds\, dr\, d\vec{\theta} \right) \\
&= 2^{p-1} \int_{\mathcal{S}^1} \int_0^{r_{\max}} r \left| f\left(r\vec{\theta}\right) \right|^p \left(\frac{1}{r} \int_r^{r_{\max}} s^{2-1/p} \hat{w}\left(s\right)\, ds \right.
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \int_0^r \frac{s t_{\max}^{1-1/p} \hat{w}(s)}{\sqrt{r^2 - s^2 \sin^2 \phi}}\, ds \Bigg)\, dr\, d\vec{\theta} \\
&= 2^{p-1} \int_{\mathcal{S}^1} \int_0^{r_{\max}} r \left| f\left(r\vec{\theta}\right) \right|^p w(r)\, dr\, d\vec{\theta} \\
&= 2^{p-1} \left\| f \right\|_{\mathcal{L}^p(\Omega, w)}^p .
\end{aligned}
$$

Thus, $\left\| \mathcal{Q} f \right\|_{\mathcal{L}^p\left(\hat{\Omega}, \hat{w}\right)} \leq 2^{1-1/p} \left\| f \right\|_{\mathcal{L}^p(\Omega, w)}$. If $r_{\min} > 0$, we may set $\hat{w}(s) = 1$, and find that w is bounded on $[r_{\min}, r_{\max}]$, and so we can drop the weight functions to describe $\mathcal{Q}$ as mapping $\mathcal{L}^p(\Omega)$ continuously into $\mathcal{L}^p\left(\hat{\Omega}\right)$.

In view of the substitution in the second integral, we will take:

$$r\vec{\theta} = s\vec{\sigma} + tA\vec{\sigma}$$

in the context of the second integral for the rest of this chapter, which defines r as a function of s and t, and θ as a function of s, σ, and t. □

Next, we will see how $\mathcal{Q}$ acts on Sobolev Spaces. First, we should see how $\mathcal{Q}$ interacts with partial derivative operators. For convenience, we shall look at partial differentiation with respect to polar variables. Since $\mathcal{Q}$ commutes with rotations, it is easy to see that $\mathcal{Q}$ also commutes with differentiation with respect to the angular variable, i.e., $\frac{\partial \mathcal{Q} f}{\partial \sigma} = \mathcal{Q}\left(\frac{\partial f}{\partial \theta}\right)$, whenever $\frac{\partial f}{\partial \theta}$ also exists in $\mathcal{L}^p(\Omega)$.

Proposition 5.1.2. *Let $f \in \mathcal{L}^p(\Omega, w)$, and suppose $\frac{\partial f}{\partial \theta}$ exists. Then*

$$\frac{\partial \mathcal{Q} f}{\partial \sigma} = \mathcal{Q}\left(\frac{\partial f}{\partial \theta}\right).$$

Proof. This is a straightforward computation. Recall that $\frac{\partial}{\partial \sigma}\vec{\sigma} = \vec{\sigma}^{\perp}$ and for any nonzero vector $\vec{\mathbf{v}}$, $\vec{\mathbf{v}}^{\perp} \cdot \nabla f(\vec{\mathbf{v}}) = \frac{\partial f}{\partial \theta}(\vec{\mathbf{v}})$, where $\vec{\mathbf{v}}^{\perp}$ denotes a counter-clockwise rotation of $\vec{\mathbf{v}}$ by an angle of $\frac{\pi}{2}$:

$$\frac{\partial \mathcal{Q} f}{\partial \sigma} = \int_0^s t\vec{\sigma}^{\perp} \cdot \nabla f(t\vec{\sigma})\, dt + \int_0^{\infty} \left(s\vec{\sigma}^{\perp} + tA\vec{\sigma}^{\perp}\right) \cdot \nabla f(s\vec{\sigma} + tA\vec{\sigma})\, dt$$

$$= \int_0^s \frac{\partial f}{\partial \theta}\left(t\vec{\sigma}\right)\, dt + \int_0^\infty \frac{\partial f}{\partial \theta}\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt$$
$$= \mathcal{Q}\left(\frac{\partial f}{\partial \theta}\right). \qquad \square$$

Proposition 5.1.3. *Let $f \in \mathcal{L}^p(\Omega, w)$, and suppose f vanishes at the boundary $r = r_{\max}$ and $\frac{\partial f}{\partial r}$ exists in $\mathcal{L}^p(\Omega, w)$. Then:*

$$s\frac{\partial \mathcal{Q}f}{\partial s} = \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{rf\right\}\right).$$

Proof. Recall that $\mathcal{D}_{\vec{\mathbf{v}}} f(\vec{\mathbf{v}}) = \|\vec{\mathbf{v}}\| \frac{\partial f}{\partial r}(\vec{\mathbf{v}})$ for nonzero vectors $\vec{\mathbf{v}}$. Then

$$\begin{aligned}
s\frac{\partial \mathcal{Q}f}{\partial s} &= sf\left(s\vec{\sigma}\right) + \int_0^\infty s\mathcal{D}_{\vec{\sigma}} f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \\
&= \int_0^s \frac{\partial}{\partial r}\left\{rf\left(r\vec{\theta}\right)\right\}\Big|_{r\vec{\theta}=t\vec{\sigma}}\, dt + \int_0^\infty \mathcal{D}_{s\vec{\sigma}+tA\vec{\sigma}} f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \\
&\qquad - \int_0^\infty t\mathcal{D}_{A\vec{\sigma}} \cdot \nabla f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \\
&= \int_0^s \frac{\partial}{\partial r}\left\{rf\left(r\vec{\theta}\right)\right\}\Big|_{r\vec{\theta}=t\vec{\sigma}}\, dt + \int_0^\infty \left\|s\vec{\sigma} + tA\vec{\sigma}\right\| \frac{\partial f}{\partial r}\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \\
&\qquad - \int_0^\infty t\frac{\partial}{\partial t}\left\{f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\right\}\, dt \\
&= \int_0^s \frac{\partial}{\partial r}\left\{rf\left(r\vec{\theta}\right)\right\}\Big|_{r\vec{\theta}=t\vec{\sigma}}\, dt + \int_0^\infty \left\|s\vec{\sigma} + tA\vec{\sigma}\right\| \frac{\partial f}{\partial r}\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \\
&\qquad - \left[tf\left(s\vec{\sigma} + tA\vec{\sigma}\right)\right]_{t=0}^\infty + \int_0^\infty f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \\
&= \int_0^s \frac{\partial}{\partial r}\left\{rf\left(r\vec{\theta}\right)\right\}\Big|_{r\vec{\theta}=t\vec{\sigma}}\, dt \\
&\qquad + \int_0^\infty \left(\left\|s\vec{\sigma} + tA\vec{\sigma}\right\| \frac{\partial f}{\partial r}\left(s\vec{\sigma} + tA\vec{\sigma}\right) + f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\right)\, dt \\
&= \int_0^s \frac{\partial}{\partial r}\left\{rf\left(r\vec{\theta}\right)\right\}\Big|_{r\vec{\theta}=t\vec{\sigma}}\, dt + \int_0^\infty \frac{\partial}{\partial r}\left\{rf\left(r\vec{\theta}\right)\right\}\Big|_{r\vec{\theta}=s\vec{\sigma}+tA\vec{\sigma}}\, dt \\
&= \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{rf\right\}\right). \qquad \square
\end{aligned}$$

We next consider higher order derivatives. It is clear that $\frac{\partial^n \mathcal{Q}f}{\partial \sigma^n} = \mathcal{Q}\left(\frac{\partial^n f}{\partial \theta^n}\right)$ whenever $\frac{\partial^n f}{\partial \theta^n}$ exists in $\mathcal{L}^p(\Omega, \hat{w})$. The real work is in resolving $\frac{\partial^n \mathcal{Q}f}{\partial s^n}$.

Proposition 5.1.4. *Let $f \in \mathcal{L}^p(\Omega, w)$ and suppose $\frac{\partial^n f}{\partial r^n}$ exists, and also that f vanishes at the boundary $r = R$, along with each of $\frac{\partial^k f}{\partial r^k}$, for $1 \leq k < n$. Then:*

$$s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n} = \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right).$$

Proof. We will use induction.

(*Basis.*) We observe the base case is:

$$s\frac{\partial \mathcal{Q} f}{\partial s} = \mathcal{Q}\left(\frac{\partial}{\partial r}\{rf\}\right),$$

which we have already proven in (5.1.3).

(*Induction.*) Now let $n \geq 1$ and suppose:

$$s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n} = \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right).$$

We will now show:

$$s^{n+1} \frac{\partial^{n+1} \mathcal{Q} f}{\partial s^{n+1}} = \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^{n+1} \frac{\partial^n f}{\partial r^n}\right\}\right).$$

Indeed, we find that

$$\begin{aligned}
s^{n+1} \frac{\partial^{n+1} \mathcal{Q} f}{\partial s^{n+1}} &= s^{n+1} \frac{\partial}{\partial s}\left\{\frac{1}{s^n} \cdot s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n}\right\} \\
&= s^{n+1}\left(\frac{-n}{s^{n+1}} \cdot s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n} + \frac{1}{s^n} \cdot \frac{\partial}{\partial s}\left\{s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n}\right\}\right) \\
&= -n s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n} + s\frac{\partial}{\partial s}\left\{s^n \frac{\partial^n \mathcal{Q} f}{\partial s^n}\right\} \\
&= -n\mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right) + s\frac{\partial}{\partial s}\left\{\mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right)\right\} \\
&= -n\mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right) + \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right\}\right) \\
&= -n\mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^n \frac{\partial^{n-1} f}{\partial r^{n-1}}\right\}\right) + \mathcal{Q}\left(\frac{\partial}{\partial r}\left\{nr^n \frac{\partial^{n-1} f}{\partial r^{n-1}} + r^{n+1}\frac{\partial^n f}{\partial r^n}\right\}\right).
\end{aligned}$$

Then by exploiting linearity of $\mathcal{Q}$ and $\frac{\partial}{\partial r}$, this reduces down to:

$$\frac{\partial^{n+1} \mathcal{Q} f}{\partial s^{n+1}} = \frac{1}{s^{n+1}}\mathcal{Q}\left(\frac{\partial}{\partial r}\left\{r^{n+1}\frac{\partial^n f}{\partial r^n}\right\}\right),$$

which is our desired result. □

We can now deduce Sobolev estimates on $\mathcal{Q}$. We will define $\mathcal{W}^{n,p}(\Omega, w)$ and $\mathcal{W}^{n,p}\left(\hat{\Omega}, \hat{w}\right)$ by:

$$\mathcal{W}^{n,p}(\Omega, w) = \left\{ f \in \mathcal{L}^p(\Omega, w) \;\middle|\; \frac{\partial^k f}{\partial r^j \partial \theta^{k-j}} \in \mathcal{L}^p\left(\Omega, r^{pj} w\right), 0 \leq j \leq k \leq n \right\},$$

$$\mathcal{W}^{n,p}\left(\hat{\Omega}, \hat{w}\right) = \left\{ g \in \mathcal{L}^p\left(\hat{\Omega}, \hat{w}\right) \;\middle|\; \frac{\partial^k g}{\partial s^j \partial \sigma^{k-j}} \in \mathcal{L}^p\left(\hat{\Omega}, s^{pj}\hat{w}\right), 0 \leq j \leq k \leq n \right\},$$

with norms defined by:

$$\|f\|^p_{\mathcal{W}^{n,p}(\Omega;w)} = \sum_{k=0}^{n}\sum_{j=0}^{k} \left\| \frac{\partial^k f}{\partial r^j \partial \theta^{k-j}} \right\|^p_{\mathcal{L}^p(\Omega, r^{pj}w)},$$

$$\|g\|^p_{\mathcal{W}^{n,p}\left(\hat{\Omega};\hat{w}\right)} = \sum_{k=0}^{n}\sum_{j=0}^{k} \left\| \frac{\partial^k g}{\partial s^j \partial \sigma^{k-j}} \right\|^p_{\mathcal{L}^p\left(\hat{\Omega}, s^{pj}\hat{w}\right)}.$$

Then define:

$$\mathcal{W}^{n,p}_0(\Omega, w) = \left\{ f \in \mathcal{W}^{n,p}(\Omega, w) \;\middle|\; \frac{\partial^j f}{\partial r^j} = 0 \text{ on } \partial\Omega, 0 \leq j < n \right\}.$$

First, we consider an estimate on seminorms.

Proposition 5.1.5. *For $f \in \mathcal{W}^{k,p}_0(\Omega; w)$:*

$$\left\| \frac{\partial^k \mathcal{Q} f}{\partial s^k} \right\|^p_{\mathcal{L}^p\left(\hat{\Omega}, s^{pk}\hat{w}(s)\right)} \leq 2^{2p-2}\left(k^p \left\| \frac{\partial^{k-1} f}{\partial r^{k-1}} \right\|^p_{\mathcal{L}^p\left(\Omega, r^{p(k-1)}w\right)} + \left\| \frac{\partial^k f}{\partial r^k} \right\|^p_{\mathcal{L}^p\left(\Omega, r^{kp}w\right)} \right).$$

Proof. Observe:

$$\begin{aligned}
\left\| \frac{\partial^k \mathcal{Q} f}{\partial s^k} \right\|^p_{\mathcal{L}^p\left(\hat{\Omega}, s^{pk}\hat{w}\right)} &= \left\| s^k \frac{\partial^k \mathcal{Q} f}{\partial s^k} \right\|^p_{\mathcal{L}^p\left(\hat{\Omega}, \hat{w}\right)} \\
&= \left\| \mathcal{Q}\left(k r^{k-1} \frac{\partial^{k-1} f}{\partial r^{k-1}} + r^k \frac{\partial^k f}{\partial r^k} \right) \right\|^p_{\mathcal{L}^p\left(\hat{\Omega}, \hat{w}\right)} \\
&\leq 2^{p-1} \left\| k r^{k-1} \frac{\partial^{k-1} f}{\partial r^{k-1}} + r^k \frac{\partial^k f}{\partial r^k} \right\|^p_{\mathcal{L}^p(\Omega, w)} \\
&\leq 2^{2p-2} \left(\left\| k r^{k-1} \frac{\partial^{k-1} f}{\partial r^{k-1}} \right\|^p_{\mathcal{L}^p(\Omega, w)} + \left\| r^k \frac{\partial^k f}{\partial r^k} \right\|^p_{\mathcal{L}^p(\Omega, w)} \right) \\
&= 2^{2p-2} \left(k^p \left\| \frac{\partial^{k-1} f}{\partial r^{k-1}} \right\|^p_{\mathcal{L}^p\left(\Omega, r^{p(k-1)}w\right)} + \left\| \frac{\partial^k f}{\partial r^k} \right\|^p_{\mathcal{L}^p\left(\Omega, r^{kp}w\right)} \right).
\end{aligned}$$

□

Theorem 5.1.6. *For $n \geq 1$ and $f \in \mathcal{W}_0^{n,p}(\Omega; w)$,*

$$\|\mathcal{Q}f\|^p_{\mathcal{W}^{n,p}(\hat{\Omega};\hat{w})} \leq 2^{2p-2}\left[(n-1)^p + 1\right]\|f\|^p_{\mathcal{W}^{n,p}(\Omega,w)}.$$

Proof. Observe:

$$\begin{aligned}
\|\mathcal{Q}f\|^p_{\mathcal{W}^{n,p}(\hat{\Omega};\hat{w})} &= \sum_{j=0}^{n}\sum_{k=0}^{j}\left\|\frac{\partial^j \mathcal{Q}f}{\partial s^k \partial\sigma^{j-k}}\right\|^p_{\mathcal{L}^p\left(\hat{\Omega},s^{pj}\hat{w}\right)} \\
&\leq 2^{2p-2}\left(\sum_{j=1}^{n}\sum_{k=1}^{j}k^p\left\|\frac{\partial^{j-1}f}{\partial r^{k-1}\partial\sigma^{j-k}}\right\|^p_{\mathcal{L}^p\left(\Omega,r^{p(k-1)}w\right)}\right. \\
&\qquad \left. + \sum_{j=0}^{n}\sum_{k=0}^{j}\left\|\frac{\partial^j f}{\partial r^k\partial\sigma^{j-k}}\right\|^p_{\mathcal{L}^p\left(\Omega,r^{pk}w\right)}\right) \\
&= 2^{2p-2}\left(\sum_{j=0}^{n-1}\sum_{k=0}^{j-1}(k+1)^p\left\|\frac{\partial^j f}{\partial r^k\partial\sigma^{j-k}}\right\|^p_{\mathcal{L}^p\left(\Omega,r^{pk}w\right)}\right. \\
&\qquad \left. + \sum_{j=0}^{n}\sum_{k=0}^{j}\left\|\frac{\partial^j f}{\partial r^k\partial\sigma^{j-k}}\right\|^p_{\mathcal{L}^p\left(\Omega,r^{pk}w\right)}\right) \\
&\leq 2^{2p-2}\left[(n-1)^p+1\right]\|f\|^p_{\mathcal{W}^{n,p}(\Omega,w)}.
\end{aligned}$$

□

5.2 Inversion via Fourier Series from one set of data

Because of the rotationally-invariant behavior of $\mathcal{Q}$, we are motivated to look at how the Fourier transform with respect to the angular variable interacts with $\mathcal{Q}$. In this setting, the partial Fourier transforms of f and $\mathcal{Q}f$, with respect to $\vec{\theta}$ and $\vec{\sigma}$ respectively, are as follows:

$$\tilde{f}(r,n) = \frac{1}{\sqrt{2\pi}}\int_{\mathcal{S}^1} f\left(r\vec{\theta}\right)e^{-in\theta}\,d\vec{\theta}, \quad \widetilde{\mathcal{Q}f}(s,n) = \frac{1}{\sqrt{2\pi}}\int_{\mathcal{S}^1}\mathcal{Q}f(s\vec{\sigma})\,e^{-in\sigma}\,d\vec{\sigma},$$

where we identify a real number θ with the vector $\vec{\theta} = \langle\cos\theta, \sin\theta\rangle$, and similarly for σ and $\vec{\sigma}$. These integrals are well-defined for every r and s for which the restrictions $f|_{r\mathcal{S}^1}$

and $\mathcal{Q}f|_{s\mathcal{S}^1}$ are integrable. The inversions are given as

$$f\left(r\vec{\theta}\right) = \frac{1}{\sqrt{2\pi}} \sum_{n\in\mathbb{Z}} \tilde{f}\left(r,n\right) e^{in\theta}, \quad \mathcal{Q}f\left(s\vec{\sigma}\right) = \frac{1}{\sqrt{2\pi}} \sum_{n\in\mathbb{Z}} \widetilde{\mathcal{Q}f}\left(s,n\right) e^{in\sigma}, \tag{5.2.1}$$

for every r and s where $\tilde{f}\left(r,\cdot\right), \widetilde{\mathcal{Q}f}\left(s,\cdot\right) \in \ell^1$.

These mappings then extend continuously to $\mathcal{L}^2$ isometries, from Ω and $\hat{\Omega}$ to $[r_{\min}, r_{\max}]\times\mathbb{Z}$ and $[0, r_{\max}] \times \mathbb{Z}$ respectively, where

$$\left\|\tilde{f}\right\|_{\mathcal{L}^2}^2 = \sum_{n\in\mathbb{N}} \int_{r_{\min}}^{r_{\max}} r\left|\tilde{f}\left(r,n\right)\right|^2 w\left(r\right)\, dr, \quad \left\|\widetilde{\mathcal{Q}f}\right\|_{\mathcal{L}^2}^2 = \sum_{n\in\mathbb{N}} \int_0^{r_{\max}} s\left|\widetilde{\mathcal{Q}f}\left(s,n\right)\right|^2 \hat{w}\left(s\right)\, ds.$$

We then compute the partial Fourier transform $\widetilde{\mathcal{Q}f}$ as follows:

$$\begin{aligned}
\widetilde{\mathcal{Q}f}\left(s,n\right) &= \frac{1}{\sqrt{2\pi}} \int_{\mathcal{S}^1} \left(\int_0^s f\left(t\vec{\sigma}\right)\, dt + \int_0^\infty f\left(s\vec{\sigma} + tA\vec{\sigma}\right)\, dt \right) e^{-in\sigma}\, d\vec{\sigma} \\
&= \frac{1}{\sqrt{2\pi}} \left(\int_0^s \int_{\mathcal{S}^1} f\left(t\vec{\sigma}\right) e^{-in\sigma}\, d\vec{\sigma}\, dt + \int_0^\infty \int_{\mathcal{S}^1} f\left(s\vec{\sigma} + tA\vec{\sigma}\right) e^{-in\sigma}\, d\vec{\sigma}\, dt \right).
\end{aligned}$$

Much as we did in the proof of (5.1.1), we relabel $t\vec{\sigma} = r\vec{\theta}$ in the first integral, and perform the multivariable substitution $s\vec{\sigma} + tA\vec{\sigma} = r\vec{\theta}$ in the second integral to obtain:

$$\begin{aligned}
\widetilde{\mathcal{Q}f}\left(s,n\right) &= \frac{1}{\sqrt{2\pi}} \left(\int_0^s \int_{\mathcal{S}^1} f\left(r\vec{\theta}\right) e^{-in\theta}\, d\vec{\theta}\, dr \right. \\
&\qquad \left. + \int_s^\infty \int_{\mathcal{S}^1} f\left(r\vec{\theta}\right) e^{-in\left(\theta - \phi + \arcsin\left(\frac{s}{r}\sin\phi\right)\right)} \cdot \frac{1}{\sqrt{1 - \left(\frac{s}{r}\right)^2 \sin^2\phi}}\, d\vec{\theta}\, dr \right) \\
&= \int_0^s \tilde{f}\left(r,n\right)\, dr + \int_s^\infty \tilde{f}\left(r,n\right) e^{in\left(\phi - \arcsin\left(\frac{s}{r}\sin\phi\right)\right)} \cdot \frac{1}{\sqrt{1 - \left(\frac{s}{r}\right)^2 \sin^2\phi}}\, dr \\
&= \int_0^\infty \tilde{f}\left(r,n\right) K\left(\frac{s}{r}, n\right)\, dr \\
&= \widetilde{rf} \star K\left(s,n\right),
\end{aligned} \tag{5.2.2}$$

where $\star$ is convolution only in the first variable, and

$$K\left(q,n\right) = \begin{cases} 1 & \text{if } q > 1, \\ e^{in\left(\phi - \arcsin\left(q\sin\phi\right)\right)} \cdot \frac{1}{\sqrt{1-q^2\sin^2\phi}} & \text{if } q \leq 1. \end{cases}$$

While the inversion formula given in (4.2.1) required f to be $\mathcal{C}^2$ in order to be valid in a classical sense, we merely require f be continuous here. From this assumption, we differentiate result (5.2.2) with respect to s, to obtain

$$\begin{aligned}\frac{\partial\widetilde{\mathcal{Q}f}}{\partial s}(s,n) &= \tilde{f}(s,n) + \int_s^\infty \tilde{f}(r,n)\cdot\frac{1}{r}\frac{\partial K}{\partial q}\left(\frac{s}{r},n\right)dr - \tilde{f}(s,n)\,K(1,n)\\ &= (1-\sec\phi)\,\tilde{f}(s,n) + \tilde{f}\star\frac{\partial K}{\partial q}(s,n),\end{aligned} \tag{5.2.3}$$

which is a Volterra integral equation of the second kind in convolution form, (2.1.1), for which a solution $\tilde{f}$ is given by

$$\tilde{f} = \frac{1}{1-\sec\phi}\left(\frac{\partial\widetilde{\mathcal{Q}f}}{\partial s} + \sum_{\nu=1}^{\infty}\left(\frac{-1}{1-\sec\phi}\right)^{\nu}\left(\frac{\partial K}{\partial q}\right)^{\star\nu}\star\frac{\partial\widetilde{\mathcal{Q}f}}{\partial s}\right). \tag{5.2.4}$$

Alternately, if we multiply both sides of (5.2.3) by s, we obtain

$$\begin{aligned}s\frac{\partial\widetilde{\mathcal{Q}f}}{\partial s}(s,n) &= (1-\sec\phi)\,s\tilde{f}(s,n) + \int_s^\infty \tilde{f}(r,n)\cdot\frac{s}{r}\frac{\partial K}{\partial q}\left(\frac{s}{r},n\right)dr\\ &= (1-\sec\phi)\,\widetilde{rf}(s,n) + \widetilde{rf}\star\left(q\frac{\partial K}{\partial q}\right)(s,n),\end{aligned}$$

and obtain yet another Volterra Integral Equation of the Second Kind, also in convolution form, in which the solution $\widetilde{rf}$ is given by

$$\widetilde{rf} = \frac{1}{1-\sec\phi}\left(s\frac{\partial\widetilde{\mathcal{Q}f}}{\partial s} + \sum_{\nu=1}^{\infty}\left(\frac{-1}{1-\sec\phi}\right)^{\nu}\left(q\frac{\partial K}{\partial q}\right)^{\star\nu}\star\left(s\frac{\partial\widetilde{\mathcal{Q}f}}{\partial s}\right)\right). \tag{5.2.5}$$

One does notice, however, that reconstructing f (by way of reconstructing $\tilde{f}$) for $r \geq r_{\min}$ only requires knowledge of $\mathcal{Q}f$ for $s \geq r_{\min}$, and thus the problem of reconstructing f when $\mathcal{Q}f$ is also known for $s < r_{\min}$ is in a sense **overdetermined**.

5.3 Numerical Inversion of the Polar Broken Ray transform

We will assume $\mathcal{Q}f$ will be sampled on a polar grid, and wish to reconstruct f on the same grid. If we let $H > 0$ and $W > 0$, $\Delta r = \frac{r_{\max} - r_{\min}}{W-1}$, and $\Delta\theta = \frac{2\pi}{H}$, let

$$s_k = r_k = r_{\min} + k\Delta r,$$

$$\sigma_k = \theta_k = j\Delta\theta,$$

$$\mathbf{f}[j,k] = f\left(r_k\vec{\theta}_j\right), \quad 0 \le j < H,\, 0 \le k < W,$$

and

$$\mathbf{Qf}[j,k] = \mathcal{Q}f\left(s_k\vec{\sigma}_j\right), \quad 0 \le j < H,\, 0 \le k < W.$$

Here, $\mathbf{f}$ and $\mathbf{Qf}$ represent sampled values of f and $\mathcal{Q}f$.

Formula (5.2.4) indicates we would reconstruct f by reconstructing $\tilde{f}(\cdot, n)$ from $\widetilde{Qf}(\cdot, n)$ for each n, a task which is easily parallelizable. However, since the inversion formula is an infinite sum, it is impractical to devise a numerical implementation of the inversion formula directly, and instead, we will model the forward transform (5.2.3) instead. For $n \in \mathbb{Z}$, define

$$\mathcal{A}_n g(s) = (1 - \sec\phi)\, g(s) + \int_s^{r_{\max}} g(r)\, B_n(r,s)\, dr, \quad B_n(r,s) = \frac{1}{r}\frac{\partial K}{\partial q}\left(\frac{s}{r}, n\right).$$

If $\widetilde{\mathcal{Q}f}(s,n)$ is zero for $|n| > \frac{H}{2}$ then so will $\tilde{f}(r,n)$, and then f and $\mathcal{Q}f$ are uniquely determined on each circle $r = r_k$ and $s = s_k$, for $0 \le k < W$. Furthermore, values of $\tilde{f}(r_k, n)$ and $\widetilde{\mathcal{Q}f}(s_k, n)$ are given exactly by the partial Discrete Fourier transforms $\tilde{\mathbf{f}}$ and $\widetilde{\mathbf{Qf}}$ of $\mathbf{f}$ and $\mathbf{Qf}$ with respect to j.

We then proceed with an a priori assumption that f is interpolated linearly in r between sampled values, giving us

$$f\left(r\vec{\theta}_j\right) = \sum_{k=0}^{W-1} \mathbf{f}\left[j,k\right] \kappa\left(r - r_k\right),$$

where

$$\kappa\left(r\right) = \begin{cases} 1 - \frac{|r|}{\Delta r} & \text{if } |r| \leq \Delta r, \\ 0 & \text{if } |r| > \Delta r. \end{cases}$$

Then $\tilde{f}$ is interpolated in the same way:

$$\tilde{f}\left(r,j\right) = \sum_{k=0}^{W-1} \tilde{\mathbf{f}}\left[j,k\right] \kappa\left(r - r_k\right),$$

and

$$\begin{aligned} \mathcal{A}_n \tilde{f}\left(s_m,n\right) &= \left(1-\sec\phi\right)\tilde{f}\left(s_m,n\right) + \int_s^{r_{\max}} \tilde{f}\left(r,n\right) B_n\left(r,s_m\right)\, dr \\ &= \left(1-\sec\phi\right)\tilde{\mathbf{f}}\left[n,m\right] + \int_s^{r_{\max}} \sum_{k=0}^{W-1} \tilde{\mathbf{f}}\left[n,k\right] \kappa\left(r-r_k\right) B_n\left(r,s_m\right)\, dr \\ &= \left(1-\sec\phi\right)\tilde{\mathbf{f}}\left[n,m\right] + \sum_{k=0}^{W-1} \tilde{\mathbf{f}}\left[n,k\right] \int_s^{r_{\max}} \kappa\left(r-r_k\right) B_n\left(r,s_m\right)\, dr \\ &= \left(1-\sec\phi\right)\tilde{\mathbf{f}}\left[n,m\right] + \sum_{k=0}^{W-1} \tilde{\mathbf{f}}\left[n,k\right] \cdot \kappa * B_n\left(r_k,s_m\right). \end{aligned}$$

Thus, for each fixed n, $\mathbf{g}\left[n,m\right] = \mathcal{V}_n \tilde{f}\left(s_m,n\right)$ is the result of a matrix multiplication $\mathbf{g}\left[n\right] = \mathbf{A}_n \cdot \tilde{\mathbf{f}}\left[n\right]$, where

$$\mathbf{A}_n\left[m,k\right] = \kappa * B_n\left(r_k,s_m\right) + \left(1-\sec\phi\right)\delta_{m,k}, \tag{5.3.1}$$

where $\delta_{m,k}$ is the Kronecker delta. Notice that $\mathbf{A}_n$ is an upper triangular matrix, making inversion feasible. Thus, the reconstruction scheme is to approximate $\mathbf{g}$ by applying an appropriate numerical differentiation scheme to $\widetilde{\mathbf{Qf}}$, solving the upper triangular system for $\tilde{\mathbf{f}}\left[n\right]$, from which then recover $\mathbf{f}$ via inverse FFT.

For further details and Python scripts implementing this inversion, see Appendix B.

5.4 Microlocal analysis of the distributional Polar Broken Ray transform

Recall:

$$\mathcal{Q}f\left(s\vec{\sigma}\right)=\int_0^s f\left(t\vec{\sigma}\right)\,dt+\int_0^\infty f\left(s\vec{\sigma}+tA\vec{\sigma}\right)\,dt.$$

Also recall that in the case of the second integral, we take

$$r\vec{\theta}=s\vec{\sigma}+tA\vec{\sigma}, \tag{5.4.1}$$

and so we rewrite our formula for $\mathcal{Q}f$ as

$$\mathcal{Q}f\left(s\vec{\sigma}\right)=\int_0^s f\left(t\vec{\sigma}\right)\,dt+\int_s^\infty f\left(r\vec{\theta}\right)\cdot\frac{\partial t}{\partial r}\,dr,$$

where in the second integral, we view t and θ as implicitly being functions of r and s in view of (5.4.1).

Pulling both f and $\mathcal{Q}f$ by the map

$$\omega\left(\ell,\sigma\right)=\left(e^\ell\cos\sigma,e^\ell\sin\sigma\right),$$

and making the subtitution $t=e^{\ell-\tau}$ in the first integral, and $r=e^{\ell+\tau}$ in the second integral, we can find that

$$\theta=\sigma+\phi-\arcsin\left(e^{-\tau}\sin\phi\right).$$

This gives us the following:

$$\begin{aligned}\omega^\star\mathcal{Q}f\left(\ell,\sigma\right)=&\int_0^\infty e^{\ell-\tau}\omega^\star f\left(\ell-\tau,\sigma\right)\,d\tau\\&+\int_0^\infty e^{\ell+\tau}\omega^\star f\left(\ell+\tau,\sigma+\phi-\arcsin\left(e^{-\tau}\sin\phi\right)\right)\cdot\left.\frac{\partial t}{\partial r}\right|_{\frac{r}{s}=e^\tau}d\tau.\end{aligned}$$

Definition 5.4.1 (The Distributional Polar Broken Ray transform)**.** Let $f\in\mathcal{E}'\left(\mathbb{R}^2\backslash 0\right)$. Then $\mathcal{Q}f$ is defined as the distribution satisfying

$$\omega^\star\mathcal{Q}f\left(\ell,\sigma\right)=\int_0^\infty e^{\ell-\tau}\omega^\star f\left(\ell-\tau,\sigma\right)\,d\tau$$

$$+\int_0^\infty e^{\ell+\tau}\omega^\star f\left(\ell+\tau,\sigma+\phi-\arcsin\left(e^{-\tau}\sin\phi\right)\right)\cdot\left.\frac{\partial t}{\partial r}\right|_{\frac{r}{s}=e^\tau}d\tau.$$

The first of these integrals is interpreted as the integral (3.1.2), with $\vec{\mathbf{v}}=(1,0)$, and the second integral is interpreted as the integral (3.2.1) with:

$$\gamma(\tau)=\left(-\tau,\arcsin\left(e^{-\tau}\sin\phi\right)-\phi\right),\quad \upsilon(\tau)=\left.\frac{\partial t}{\partial r}\right|_{\frac{r}{s}=e^\tau}=\frac{1}{\sqrt{1-e^{-2\tau}\sin^2\phi}}.$$

In both of these integrals, we take $u=e^\ell\omega^\star f$. In setting $s=e^\ell$, and making the substitution $r=e^{\ell+\tau}$ in the second integral, we observe that τ is implicitly a function of s and t.

Since $\omega^\star\mathcal{Q}f$, as defined above, is obtained from convolving $\omega^\star f$ with some distribution, $\omega^\star\mathcal{Q}f$ inherits the necessary and sufficient condition for being in the range of $\omega^\star$ – namely that $\omega^\star\mathcal{Q}f$ is 2π-periodic in σ. Thus, the above is a valid definition of $\mathcal{Q}f$.

Theorem 5.4.2 (Propagation of singularities of the Distributional Polar Broken Ray transform)**.** *For $f\in\mathcal{E}'\left(\mathbb{R}^2\right)$ with support bounded away from the origin:*

$$\begin{aligned}WF(\mathcal{Q}f)\subseteq\ & WF(f)\\ &\cup\left\{\left(s\vec{\sigma},\vec{\xi}\right)\,\middle|\,\exists t\le s:\left(t\vec{\sigma},\vec{\xi}\right)\in WF(f),\vec{\xi}\perp\vec{\sigma}\right\}\\ &\cup\left\{\left(s\vec{\sigma},\rho_{\sigma-\theta}\vec{\xi}\right)\,\middle|\,\exists t\ge 0:\left(r\vec{\theta},\vec{\xi}\right)\in WF(f),\vec{\xi}\perp A\vec{\sigma}\right\},\end{aligned}\tag{5.4.2}$$

where $r\vec{\theta}=s\vec{\sigma}+tA\vec{\sigma}$ in (5.4.2)*, and $\rho_{\sigma-\theta}$ is counterclockwise rotation by an angle of $\sigma-\theta$.*

Proof. From Theorem 3.2.2, we observe that

$$\begin{aligned}&WF\left\{\int_0^\infty e^{\ell-\tau}\omega^\star f(\ell-\tau,\sigma)\,d\tau\right\}\backslash WF(\omega^\star f)\\ &\qquad\subseteq\left\{((\ell,\sigma),\vec{\eta})\,|\,\exists\tau\ge 0:((\ell-\tau,\sigma),\vec{\eta})\in WF(\omega^\star f),\vec{\eta}\perp(-1,0)\right\},\end{aligned}$$

and

$$WF\left\{\int_0^\infty e^{\ell+\tau}\omega^\star f((\ell,\sigma)-\vec{\gamma}(\tau))\cdot\frac{1}{\sqrt{1-e^{-2\tau}\sin^2\phi}}\,d\tau\right\}\backslash WF(\omega^\star f)$$

$$\subseteq \left\{((\ell,\sigma),\vec{\eta}) \,\middle|\, \exists \tau \geq 0 : \vec{\eta} \perp \vec{\gamma}'(\tau) \;\;\&\;\; ((\ell,\sigma) - \vec{\gamma}(\tau),\vec{\eta}) \in WF(\omega^\star f)\right\}.$$

Hence:

$$\begin{aligned} WF(\omega^\star \mathcal{Q} f) &\subseteq WF(\omega^\star f) \\ &\cup \{((\ell,\sigma),\vec{\eta}) \,|\, \exists \tau \geq 0 : ((\ell-\tau,\sigma),\vec{\eta}) \in WF(\omega^\star f), \vec{\eta} \perp (-1,0)\} \qquad (5.4.3) \\ &\cup \left\{((\ell,\sigma),\vec{\eta}) \,\middle|\, \exists \tau \geq 0 : ((\ell,\sigma) - \vec{\gamma}(\tau),\vec{\eta}) \in WF(\omega^\star f), \vec{\eta} \perp \vec{\gamma}'(\tau)\right\}. \qquad (5.4.4) \end{aligned}$$

In applying $\omega^{-\star}$ to (5.4.3), we recall that

$$\mathcal{D}\omega(\ell,\sigma) = s\rho_\sigma,$$

when we set $s = e^\ell$. We also set $t = e^{\ell-\tau}$, and compute

$$\begin{aligned} &\omega^{-\star}\{((\ell,\sigma),\vec{\eta}) \,|\, \exists \tau \geq 0 : ((\ell-\tau,\sigma),\vec{\eta}) \in WF(\omega^\star f), \vec{\eta} \perp (-1,0)\} \\ &\quad = \left\{\left(\omega(\ell,\sigma), (s\rho_\sigma)^{-T}\vec{\eta}\right) \,\middle|\, \right. \\ &\qquad \left. \exists \tau \geq 0 : \left(\omega(\ell-\tau,\sigma), (s\rho_\sigma)^{-T}\vec{\eta}\right) \in WF(f), \vec{\eta} \perp (-1,0)\right\} \\ &\quad = \{(s\vec{\sigma},\rho_\sigma\vec{\eta}) \,|\, \exists t \leq s : (t\vec{\sigma},\rho_\sigma\vec{\eta}) \in WF(f), \vec{\eta} \perp (-1,0)\} \\ &\quad = \left\{\left(s\vec{\sigma},\vec{\xi}\right) \,\middle|\, \exists t \leq s : \left(t\vec{\sigma},\vec{\xi}\right) \in WF(f), \rho_{-\sigma}\vec{\xi} \perp (-1,0)\right\} \\ &\quad = \left\{\left(s\vec{\sigma},\vec{\xi}\right) \,\middle|\, \exists t \leq s : \left(t\vec{\sigma},\vec{\xi}\right) \in WF(f), \vec{\xi} \perp \rho_\sigma(-1,0)\right\} \\ &\quad = \left\{\left(s\vec{\sigma},\vec{\xi}\right) \,\middle|\, \exists t \leq s : \left(t\vec{\sigma},\vec{\xi}\right) \in WF(f), \vec{\xi} \perp \vec{\sigma}\right\}. \end{aligned}$$

Applying $\omega^{-\star}$ to (5.4.4), we recall that

$$\omega((\ell,\sigma) - \vec{\gamma}(\tau)) = r\vec{\theta},$$

and so

$$\mathcal{D}\omega((\ell,\sigma) - \vec{\gamma}(\tau)) = r\rho_\theta.$$

We also notice that t and τ are implicitly functions of each other, and so

$$\begin{aligned}
&\omega^{-\star}\left\{\left((\ell,\sigma),\vec{\eta}\right)\,\middle|\,\exists\tau\geq 0:\left((\ell,\sigma)-\vec{\gamma}\left(\tau\right),\vec{\eta}\right)\in WF\left(\omega^{\star}f\right),\vec{\eta}\perp\vec{\gamma}'\left(\tau\right)\right\}\\
&\quad=\left\{\left(\omega\left(\ell,\sigma\right),\left(s\rho_{\sigma}\right)^{-T}\vec{\eta}\right)\,\middle|\,\exists\tau\geq 0:\left(r\vec{\theta},\left(r\rho_{\theta}\right)^{-T}\vec{\eta}\right)\in WF\left(f\right),\vec{\eta}\perp\vec{\gamma}'\left(\tau\right)\right\}\\
&\quad=\left\{\left(s\vec{\sigma},\rho_{\sigma}\vec{\eta}\right)\,\middle|\,\exists\tau\geq 0:\left(r\vec{\theta},\rho_{\theta}\vec{\eta}\right)\in WF\left(f\right),\vec{\eta}\perp\vec{\gamma}'\left(\tau\right)\right\}\\
&\quad=\left\{\left(s\vec{\sigma},\rho_{\sigma-\theta}\vec{\xi}\right)\,\middle|\,\exists\tau\geq 0:\left(r\vec{\theta},\vec{\xi}\right)\in WF\left(f\right),\rho_{-\theta}\vec{\xi}\perp\vec{\gamma}'\left(\tau\right)\right\}\\
&\quad=\left\{\left(s\vec{\sigma},\rho_{\sigma-\theta}\vec{\xi}\right)\,\middle|\,\exists\tau\geq 0:\left(r\vec{\theta},\vec{\xi}\right)\in WF\left(f\right),\vec{\xi}\perp\rho_{\theta}\vec{\gamma}'\left(\tau\right)\right\}.
\end{aligned}$$

Fixing $s\vec{\sigma}$, and differentiating both sides of

$$r\vec{\theta}=\omega\left(\left(\ell,\sigma\right)-\vec{\gamma}\left(\tau\right)\right),$$

with respect to t, we obtain

$$A\vec{\sigma}=-\mathcal{D}\omega\left(\left(\ell,\sigma\right)-\vec{\gamma}\left(\tau\right)\right)\vec{\gamma}'\left(\tau\right)\cdot\frac{\partial\tau}{\partial t}=-r\rho_{\theta}\vec{\gamma}'\left(\tau\right)\cdot\frac{\partial\tau}{\partial t}.$$

This indicates that we can replace $\rho_{\theta}\vec{\gamma}'\left(\tau\right)$ with $A\vec{\sigma}$. We can also replace $\tau\geq 0$ with $t\geq 0$ to obtain

$$\begin{aligned}
&\omega^{-\star}\left\{\left((\ell,\sigma),\vec{\eta}\right)\,\middle|\,\exists\tau\geq 0:\vec{\eta}\perp\vec{\gamma}'\left(\tau\right)\ \ \&\ \ \left((\ell+\tau,\theta),\vec{\eta}\right)\in WF\left(\omega^{\star}f\right)\right\}\\
&\qquad=\left\{\left(s\vec{\sigma},\rho_{\sigma-\theta}\vec{\xi}\right)\,\middle|\,\exists t\geq 0:\left(r\vec{\theta},\vec{\xi}\right)\in WF\left(f\right),\vec{\xi}\perp A\vec{\sigma}\right\}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
WF\left(\mathcal{Q}f\right)&\subseteq\omega^{-\star}WF\left(\omega^{\star}\mathcal{Q}f\right)\\
&\subseteq WF\left(f\right)\\
&\quad\cup\left\{\left(s\vec{\sigma},\vec{\xi}\right)\,\middle|\,\exists t\leq s:\left(t\vec{\sigma},\vec{\xi}\right)\in WF\left(f\right),\vec{\xi}\perp\vec{\sigma}\right\}\\
&\quad\cup\left\{\left(s\vec{\sigma},\rho_{\sigma-\theta}\vec{\xi}\right)\,\middle|\,\exists t\geq 0:\left(r\vec{\theta},\vec{\xi}\right)\in WF\left(f\right),\vec{\xi}\perp A\vec{\sigma}\right\}.
\end{aligned}$$

□

Unfortunately, since our inversion formula for the Polar Broken Ray transform is expressed in terms of an infinite series, and also in terms of partial Fourier transforms, we are left

with a one-sided microlocal analysis result as opposed to the two-sided results found in Chapter 4 for the Florescu, et. al., and Katsevich and Krylov Broken Ray transforms.

Shown below are figures illustrating the propagation of singularities result obtained for the Polar Broken Ray transform.

Using a scattering angle of $\phi = \frac{\pi}{3}$, figure 5.1 shows the maximum extent of $WF(\mathcal{Q}f)$, where f is the characteristic function of a disc. Here, $WF(f)$ is depicted by the circle, along with its normal vectors. The green curve and its normal vectors are the maximum extent of $WF(\mathcal{Q}f) \setminus WF(f)$. The green curves that meet at the origin are due to the second integral in $\mathcal{Q}f$, and correspond to points $s\vec{\sigma}$ for which the terminal ray, parametrized by $s\vec{\sigma} + tA\vec{\sigma}$, $t \geq 0$, is tangent to $\operatorname{sing\,supp}(f)$. Two broken rays of integration are shown to illustrate this.

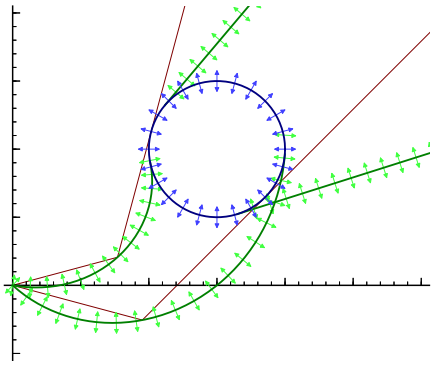

FIGURE 5.1: Wavefront sets of f and $\mathcal{Q}f$.

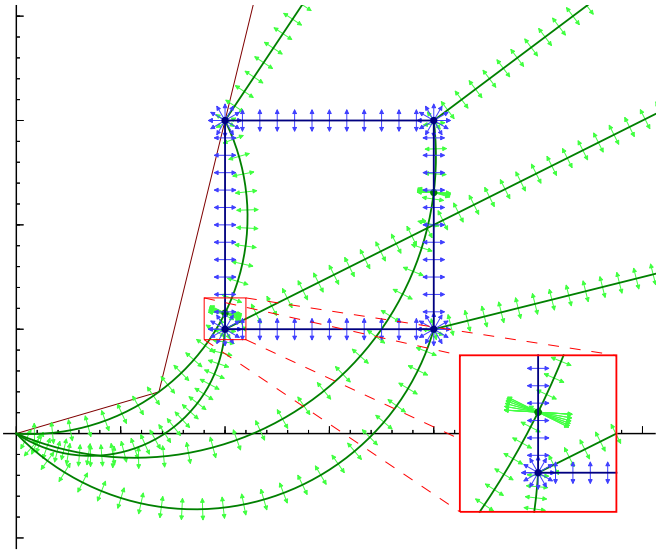

FIGURE 5.2: Wavefront sets of the characteristic function of a square and its Polar Broken Ray transform.

Figure 5.2 shows the propagation of singularities of the characteristic function of a square. Each of the vertices propagate singularities both outward from the origin radially, and inward along circular arcs connecting each vertex to the origin. The circular arcs correspond to broken rays of integration for which the terminal ray passes through a vertex of the square; such a broken ray of integration is shown in the figure.

Zooming in near the lower left corner of the square reveals a single point at which one of the circular arcs intersects the left edge. This intersection corresponds with a broken ray of integration for which the terminal ray coincides with much of the left side of the square. As a result, singularities are propagated into an interval of directions at this point. This behavior is also seen at a point on the right edge.

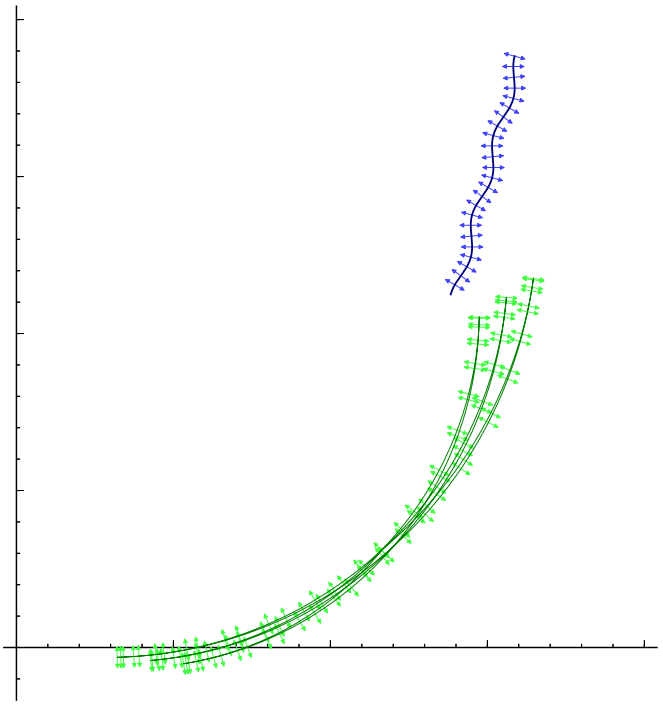

FIGURE 5.3: Propagation of singularities along an oscillatory curve by the Polar Broken Ray transform.

The propagation of singularities by the Polar Broken Ray transform become more interesting when we view how the Polar Broken Ray transform acts on distributions having a

portion of its singular support fall along an oscillatory curve, as shown in Figure 5.3. The resulting propagation in this case is given along a path that intersects with itself several times, and furthermore, is bounded away from both the origin and the original curve. Notice the amplification of the oscillations.

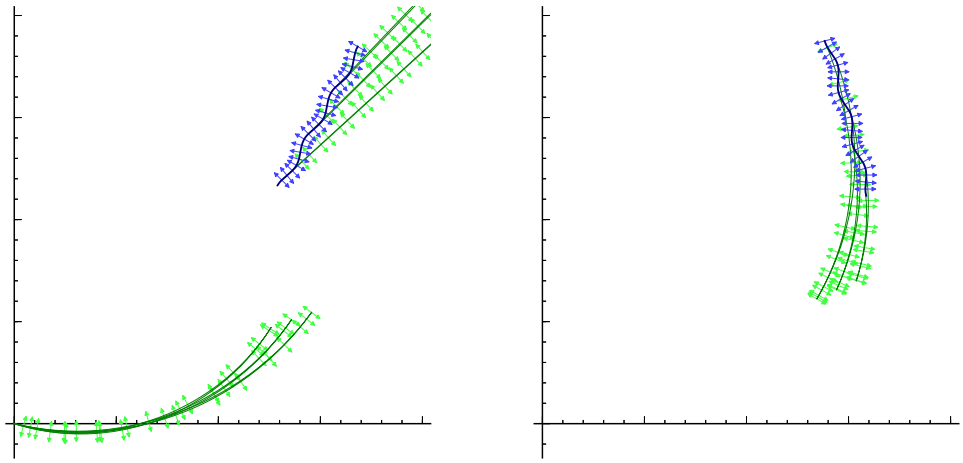

FIGURE 5.4: The effects of rotating the oscillatory curve on the propagation of singularities.

Of further interest to seeing how the Polar Broken Ray transform propagates singularities coming from an oscillatory curve is what would happen if the curve was rotated. The effects are seen in Figure 5.4. On the left, we see singularities being propagated all the way to the origin, as well as propagation in the radial direction being introduced. On the right, we see propagation into multiple paths that terminate at the curve.

6 DISCUSSION

We have investigated the variations of the Broken Ray transform introduced by Florescu, et. al., and by Katsevich and Krylov, and furthermore introduced the Polar Broken Ray transform. The results for the Florescu, et. al. and the Katsevich and Krylov Broken Ray transforms are remarkably similar in that they and their inversions differ only in multiplication by smooth functions and a change of coordinates. As such, the microlocal analysis of these two transforms are essentially identical.

The Polar Broken Ray transform presented a peculiar geometry. To analyze this transform in the distributional setting, we represented the transform as a composition of whose factors we could analyze. This factorization required the introduction of the operator $\mathcal{J}_{\gamma,v}$, which itself required further factorization in terms of pullbacks and integration to perform microlocal analysis. In performing microlocal analysis on each of these factors, we obtained a propagation of singularities result that presented a behavior not seen with the Florescu, et. al. and Katsevich and Krylov Broken Ray transforms.

Upon discovery of the propagation of singularities result for the Polar Broken Ray transform, (5.4.2), and viewing illustrations of this result, it was realized that an operator A giving an inversion for the Polar Broken Ray must necessarily have the capacity to remove the singularities of Qf that resulted from propagation. The case illustrated in Figure 5.2 indicates it is possible that an entire interval of directions at a single point be found in $WF(\mathcal{Q}f)\backslash WF(f)$. Nonvanishing partial differential operators with smooth coefficients lack the capacity to cancel out an entire interval of directions at a single point in a wavefront set, and as such, if a closed-form inversion for the Polar Broken Ray transform is to be found, it will not be a composition of such a partial differential operator with an integration operator, as was the case with the Florescu, et. al., and by Katsevich and Krylov

Broken Ray transforms.

In Chapter 5, the injectivity result of the Polar Broken Ray transform was established for continuous functions f with compact support in $\mathbb{R}^2\backslash 0$. As such, an injectivity result for distributions in general has yet to be established. A future area of investigation is to check for injectivity of the Polar Broken Ray transform for distributions $u \in \mathscr{E}'\left(\mathbb{R}^2\backslash 0\right)$. In the short term, however, one may pursue a result for distributions u whose Fourier coefficients with respect to the angular variable are in $\mathcal{L}^2\left(\mathbb{R}^+\right)$, as such a result would prove injectivity on $\mathcal{L}^2\left(\mathbb{R}^2\backslash 0\right)$.

Future work also includes continuing the search for a closed-form inversion formula for the Polar Broken Ray transform. Such an inversion formula may also establish injectivity result for the Polar Broken Ray transform defined on $\mathscr{E}'\left(\mathbb{R}^2\backslash 0\right)$. Additionally, alternate sampling used in numerical inversion may also yield better results than those shown in Appendix B. In particular, since the Polar Broken Ray transform is a two-dimensional convolution when expressed in $\ell\sigma$-coordinates, we may wish to try logarithmic sampling in the radial variable.

Since the propagation of singularities results presented in this thesis make no distinction between a discontinuity of a function and a discontinuity on any of its derivatives, one may also pursue $\mathcal{H}^s$ propagation of singularities results, in which we replace rapid decay of the Fourier transform in the definition of wavefront set with decay on the order of $\|\vec{\xi}\|^{-s}$.

BIBLIOGRAPHY

1. Lucia Florescu, Vadim A. Markel, and John C. Schotland, *Inversion formulas for the broken-ray Radon transform*, Inverse Problems 27, 025002 (2011) (2010).

2. Lucia Florescu, Vadim A Markel, and John C Schotland, *Single-scattering optical tomography: Simultaneous reconstruction of scattering and absorption*, Physical Review E **81** (2010), no. 1, 016602.

3. Rim Gouia-Zarrad and Gaik Ambartsoumian, *Exact inversion of the conical Radon transform with a fixed opening angle*, Inverse Problems **30** (2014), no. 4, 045007.

4. Michiel Hazewinkel, *Encyclopaedia of Mathematics (set)*, Springer, 1994.

5. L. Hörmander, *The Analysis of Linear Partial Differential Operators I. Distribution Theory and Fourier Analysis*, second ed., Springer Verlag, Berlin, 1990.

6. A Katsevich and R Krylov, *Broken ray transform: inversion and a range condition*, Inverse Problems **29** (2013), no. 7, 075008.

7. L. Schwartz, *Théorie des distributions*, Hermann, Paris, 1951.

8. F.G. Tricomi, *Integral Equations*, Dover Publications, New York, 1985.

APPENDICES

A Numerical inversion of the Florescu, et. al. Broken Ray transform

A numerical implementation of formula (4.2.3) to recover f is performed by way of reconstructing the approximation

$$\phi_\alpha \star f = \mathcal{B}^{-1}\phi_\alpha \star \mathcal{B}f = \left(\kappa \star \mathcal{B}^{-1}\phi_\alpha\right) * \mathbf{Bf},$$

where we view $\mathcal{B}f$ as being obtained through bilinear interpolation of an array $\mathbf{Bf}$ sampling on a mesh grid with the step sizes in x and y being given as Δx and Δy, respectively. Here, ϕ_α is an approximation to the Dirac delta distribution on $\mathbb{R}^2$ with weak derivatives to at least the second order, andκ is a bilinear interpolation kernel given by

$$\kappa(x,y) = \begin{cases} \left(1 - \frac{|x|}{\Delta x}\right)\left(1 - \frac{|y|}{\Delta y}\right) & |x| \le \Delta x, |y| \le \Delta y, \\ 0 & \text{otherwise,} \end{cases}$$

and $*$ denotes discrete convolution between the data $\mathbf{Bf}$ and an array sampling values of $\kappa \star \mathcal{B}^{-1}\phi_\alpha$.

In our reconstructions here, we choose $\phi_\alpha(x,y) = \frac{1}{\alpha^2}\phi\left(\frac{x}{\alpha}, \frac{y}{\alpha}\right)$, where

$$\phi(x,y) = \begin{cases} \left(\frac{15}{16}\right)^2 \left(1 - x^2\right)^2 \left(1 - y^2\right)^2 & \text{if } |x| \le 1 \ \ \& \ \ |y| \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

Reconstruction is carried out in two stages. In stage one, we use SAGE 6.9 to compute $\Phi = \Phi_1$ symbollically, where Φ_α satisfies the differential equation

$$\frac{\partial^4}{\partial x^2 \partial y^2}\Phi_\alpha = \mathcal{B}^{-1}\phi_\alpha.$$

It then follows that

$$\kappa \star \mathcal{B}^{-1}\phi_\alpha = \Delta x \cdot \Delta y \cdot \mathbf{D}_y^2 \mathbf{D}_x^2 \Phi_\alpha, \tag{A.1}$$

where second-order difference operator $\mathbf{D}^2$ with respect to a variable t is defined by

$$\mathbf{D}^2 g(t) = \frac{g(t+\Delta t) + g(t-\Delta t) - 2g(t)}{\Delta t^2},$$

and is applied in both in x and y.

The direction of integration in the inversion formula is given in the direction (a, b), where $a = \cos\theta - 1$, and $b = \sin\theta$. We will chose Φ_α to be obtained from integrating over right triangles with a hypotenuse along the line $bx - ay = 0$. Notice that

$$\begin{aligned}
\Phi_\alpha(x,y) &= \int_{\frac{b}{a}x}^{y}\int_{\frac{a}{b}v}^{x}(x-u)(y-v)\,\mathcal{B}^{-1}\phi_\alpha(u,v)\;du\,dv \\
&= \frac{1}{2}\int_{\frac{b}{a}x}^{y}\int_{\frac{a}{b}v}^{x}(x-u)(y-v)\int_{\mathbb{R}}\operatorname{sgn}(s)\,\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\phi_\alpha(u-sa,v-sb)\;ds\,du\,dv \\
&= \frac{1}{2}\int_{\mathbb{R}}\int_{\frac{b}{a}x-sb}^{y-sb}\int_{\frac{a}{b}v}^{x-sa}\operatorname{sgn}(s)(x-u-sa)(y-v-sb)\,\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\phi_\alpha(u,v)\;du\,dv\,ds \\
&= \frac{\alpha}{2}\int_{\mathbb{R}}\int_{\frac{b}{a}\frac{x}{\alpha}-sb}^{\frac{y}{\alpha}-sb}\int_{\frac{a}{b}v}^{\frac{x}{\alpha}-sa}\operatorname{sgn}(s)\left(\frac{x}{\alpha}-u-sa\right)\left(\frac{y}{\alpha}-v-sb\right) \\
&\qquad\cdot\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\phi(u,v)\;du\,dv\,ds \\
&= \alpha\Phi\left(\frac{x}{\alpha},\frac{y}{\alpha}\right).
\end{aligned}$$

Thus, we can compute Φ_α from Φ, using

$$\Phi(x,y) = \int_{\mathbb{R}}\int_{\frac{b}{a}x-sb}^{y-sb}\int_{\frac{a}{b}v}^{x-sa}\operatorname{sgn}(s)\,\Psi(x,y;u,v,s)\;du\,dv\,ds$$

$$\Psi(x,y;u,v,s) = \frac{1}{2}(x-u-sa)(y-v-sb)\,\mathcal{D}_{\vec{\mathbf{v}}_1}\mathcal{D}_{\vec{\mathbf{v}}_2}\phi(u,v).$$

We find that Φ is a piecewise-defined function on the plane, and we will then convert the constituent formulas for Φ into RPN code, and exported for use inside a Python environment independent of SAGE.

Stage two is where the actual inversion occurs, importing the RPN code obtained in stage one to generate the convolution kernel according to (A.1), and performing extension and cropping of arrays as needed.

A.1 Kernel-generating functions (Stage 1)

The code presented in this section is written for a SAGE environment. As of this writing, updates to the SAGE worksheet file containing the following code will be found at http://github.com/shersonb/brokenray/. Two modules, `rpncalc` and `sageextras`, are required by the worksheet. The `rpncalc` module, and successive versions, can be downloaded from http://github.com/shersonb/python-rpncalc/, whereas `sageextras` is packaged with `brokenray`. Both `rpncalc` and `sageextras` should be placed in `$SAGE_ROOT/local/lib/python/site-packages/`.

Additionally, the `array2im.py` file provides functionality to convert two-dimensional NumPy arrays into graphic files, and can be imported as a module, and be invoked as a command line utility. As of this writing, successive versions of `array2im.py` can be found at http://github.com/shersonb/python-array2im/,

Here, we define our choice of ϕ, and set up initial assumptions for use in the symbolic computation of Φ.

```
1 # Initialization of required modules and variables.
2 import rpncalc
3 import bz2
4 from sageextras import *
5 from itertools import product
6
7 # Φ will be given as a piecewise-defined function, where the pieces are
8 # obtained by dividing xy-plane across the lines x = −1, x = 1, x = −a/b,
9 # x = a/b, y = −1, y = 1, p = a − b, p = −2a, p = −a − b, p = a + b, p = 2a,
```

```
# p = b − a, and p = 0, where p = bx − ay.

var("x, y, p, q, s, u, v, a, b")
phi(x, y) = (15/16)^2*(1 - x^2)^2*(1 - y^2)^2

# Case 0 < θ < arctan(3/4)  (b > −3a):
case1 = [a < 0, b > 0, -a < b, b > -3*a]

# Case arctan(3/4) < θ < π/2  (b < −3a):
case2 = [a < 0, b > 0, -a < b, b < -3*a]

# Boundary cases - Not for use with 'Phi_symbolic'. Here, "equal" will
# mean "close enough" to get around floating point error.

# Case θ = π/2
bdcase1 = [abs(a+1) < 1e-14, abs(b-1) < 1e-14]
bdsubs1 = dict(a=-1, b=1)

# Case θ = arctan(3/4)
bdcase2 = [abs(a+0.2) < 1e-14, abs(b-0.6) < 1e-14]
bdsubs2 = dict(a=-1/5, b=3/5)
```

Here, for a scattering angle $0 < \theta \leq \frac{\pi}{2}$, it is always the case that $a < 0$ and $b > 0$. However, the computation of Φ depends on knowledge of the ordering of $\{-a, b, -3a\}$. The boundary cases $\theta = \frac{\pi}{2}$ ($a = -1$, $b = 1$) and $\theta = 2\arctan\left(\frac{1}{3}\right)$ ($a = -\frac{1}{5}$, $b = \frac{3}{5}$), are done with fewer cases, and may be obtained by making the proper substitutions into the formulas obtained from the second case.

`uvs_integral` function

This function evaluates a single instance of the integral

$$\int_{s_{\min}}^{s_{\max}} \int_{v_{\min}}^{v_{\max}} \int_{u_{\min}}^{u_{\max}} \operatorname{sgn}(s)\, \Psi(x, y; u, v, s)\; du\, dv\, ds,$$

where the limits on the two inner integrals are determined from $s_{\min}$ and $s_{\max}$.

```
def uvs_integral(integrand, smin, smax):
    # Evaluates ∫_{s_min}^{s_max} ∫_{b/a x - sb}^{y - sb} ∫_{a/b v}^{x - sa} sgn(s) Ψ(x, y; u, v, s)  du  dv  ds  for
    # the given s_min and s_max.

    u_limits = {x - s*a, -1, 1}
    v_limits_lower = {b/a*x - s*b, -1, 1}
    v_limits_upper = {y - s*b, -1, 1}

    u_limits = sorted(u_limits, cmp=cmp_symbolic)
    v_limits_lower = sorted(v_limits_lower, cmp=cmp_symbolic)
    v_limits_upper = sorted(v_limits_upper, cmp=cmp_symbolic)

    # Since φ is supported in the square [-1, 1]^2,
    # we must choose the u and v limits accordingly.

    umin = a/b*v

    umax = min([x-s*a, 1], key=u_limits.index)
    umax = max([-1, umax], key=u_limits.index)
```

```
20
21      vmin = min([b/a*x - s*b, 1], key=v_limits_lower.index)
22      vmin = max([-1, vmin], key=v_limits_lower.index)
23
24      vmax = min([y - s*b, 1], key=v_limits_upper.index)
25      vmax = max([-1, vmax], key=v_limits_upper.index)
26
27      uvs_limits = (u, umin, umax), (v, vmin, vmax), (s, smin, smax)
28
29      s_integral = multivar_integral(integrand, *uvs_limits)
30
31      return sgn(s).simplify()*s_integral
```

`Phi_symbolic_part` function

This function evaluates

$$\Phi(x,y) = \sum_k \int_{s_{\min}^{(k)}}^{s_{\max}^{(k)}} \int_{v_{\min}^{(k)}}^{v_{\max}^{(k)}} \int_{u_{\min}^{(k)}}^{u_{\max}^{(k)}} \operatorname{sgn}(s)\, \Psi(x,y;u,v,s)\, du\, dv\, ds,$$

given the case $x_{\min} \leq x \leq x_{\max}$, $y_{\min} \leq y \leq y_{\max}$, and $p_{\min} \leq p \leq p_{\max}$, The limits on x, y, and p, along with additional assumptions on x, y, and $p = bx - ay$ put in place as needed, allows us to determine the order of 's_limits', after which, we integrate over each s interval $\left[s_{\min}^{(k)}, s_{\max}^{(k)}\right]$ separately.

```
1 def Phi_symbolic_part(phi, xmin, xmax, ymin, ymax, pmin, pmax):
2     # Evaluates Φ(x,y) symbolically for x_min < x < x_max,
3     # y_min < y < y_max, and p_min < p < p_max.
```

```
4
5       psi = phi.diff(x, x)*(a + 1) + phi.diff(x, y)*b
6       Psi = (x - u - s*a)*(y - v - s*b)*psi(x=u, y=v)/2
7
8       s_limits = {0, (y - 1)/b, (y + 1)/b, (x - 1)/a, (x + 1)/a, \
9                   1/b + x/a, -1/b + x/a}
10
11      s_limits = sorted(s_limits, cmp=cmp_symbolic)
12
13      s_intervals = intervals(s_limits, unbounded=True)
14
15      integrals = []
16      for (smin, smax) in s_intervals:
17          s_assumptions = []
18
19          if smin is not -infinity:
20              for sk in s_limits[:s_limits.index(smin)+1]:
21                  s_assumptions.append(sk<s)
22
23          if smax is not infinity:
24              for sk in s_limits[s_limits.index(smax):]:
25                  s_assumptions.append(s<sk)
26
27          s_assumptions = assume2(*s_assumptions)
28
29          try:
30              uvs_integ = uvs_integral(Psi, smin, smax)
```

```
31          finally:
32              if s_assumptions:
33                  forget(*s_assumptions)
34
35          integrals.append(uvs_integ)
36
37      return sum(integrals)
```

`Phi_symbolic` function

This function, given `cases`, iterates over all possible cases of x, y, and p which produce different orderings of `s_limits` in `Phi_symbolic_part`.

```
1  def Phi_symbolic(phi, case):
2      # We generate Φ by specifying a function φ defined by formula on the
3      # square [−1, 1]^2. The returned value will be in the format accepted by
4      # the function 'rpncalc.piecewise_to_rpn':
5      # [(case1, formula1), (case2, formula2), ... ]
6
7      # We first initialize all limits on x, y, p, s, u, and v.
8      x_limits = {-1, 1, -a/b, a/b}
9      y_limits = {-1, 1}
10     p_limits = {a - b, -2*a, -a - b, a + b, 2*a, b - a, 0*x}
11
12     formulas = [ ]
13     case = assume2(*case)
```

```
14
15      # Sort the x, y, and p limits in ascending order.
16      x_limits = sorted(x_limits, cmp=cmp_symbolic)
17      y_limits = sorted(y_limits, cmp=cmp_symbolic)
18      p_limits = sorted(p_limits, cmp=cmp_symbolic)
19
20      x_intervals = intervals(x_limits, unbounded=True)
21      y_intervals = intervals(y_limits, unbounded=True)
22      p_intervals = intervals(p_limits, unbounded=True)
23
24      # We iterate over the x, y, p intervals.
25
26      for (xmin, xmax), (ymin, ymax), (pmin, pmax) in \
27              product(x_intervals, y_intervals, p_intervals):
28
29          # Determine if this is an empty case
30
31          max_ay = a*ymin if ymin is not -infinity else infinity
32          min_ay = a*ymax if ymax is not infinity else -infinity
33
34          max_bx = b*xmax if xmax is not infinity else infinity
35          min_bx = b*xmin if xmin is not -infinity else -infinity
36
37          if bool(pmax <= min_bx - max_ay) or bool(pmin >= max_bx - min_ay):
38              continue
39
40          conditions = [ ]
```

```
41
42          if xmin is not -infinity:
43              conditions.append(xmin <= x)
44          if xmax is not infinity:
45              conditions.append(x <= xmax)
46          if ymin is not -infinity:
47              conditions.append(ymin <= y)
48          if ymax is not infinity:
49              conditions.append(y <= ymax)
50          if pmin is not -infinity:
51              conditions.append(pmin <= p)
52          if pmax is not infinity:
53              conditions.append(p <= pmax)
54
55          k1 = x_intervals.index((xmin, xmax))
56          k2 = y_intervals.index((ymin, ymax))
57          k3 = p_intervals.index((pmin, pmax))
58
59          xyp_assumptions = [ ]
60          if xmin is not -infinity:
61              for xk in x_limits[:k1]:
62                  xyp_assumptions.append(xk < x)
63          if ymin is not -infinity:
64              for yk in y_limits[:k2]:
65                  xyp_assumptions.append(yk < y)
66          if pmin is not -infinity:
67              for pk in p_limits[:k3]:
```

```
68                  xyp_assumptions.append(pk < b*x - a*y)
69          if xmax is not infinity:
70              for xk in x_limits[k1:]:
71                  xyp_assumptions.append(x < xk)
72          if ymax is not infinity:
73              for yk in y_limits[k2:]:
74                  xyp_assumptions.append(y < yk)
75          if pmax is not infinity:
76              for pk in p_limits[k3:]:
77                  xyp_assumptions.append(b*x - a*y < pk)
78
79          xyp_assumptions = assume2(*xyp_assumptions)
80
81          try:
82              formula = Phi_symbolic_part(phi, xmin, xmax, \
83                      ymin, ymax, pmin, pmax)
84          finally:
85              if xyp_assumptions:
86                  forget(*xyp_assumptions)
87
88          if pmin is not -infinity and pmax is not infinity \
89                  and (xmin is -infinity or xmax is infinity) \
90                  and (ymin is -infinity or ymax is infinity):
91              formula = formula(x=(b*p+a*q)/(a^2+b^2), \
92                      y=(-a*p+b*q)/(a^2+b^2))
93
```

```
94          formulas.append((conditions, formula))
95
96      forget(*case)
97      return formulas
```

`reduce_formulas` **function**

This function is used to simplify the formulas obtained from the `Phi_symbolic` function. This is particularly useful in computing the aforementioned boundary cases.

```
1  def reduce_formulas(formulas, case, **subs):
2      # Reduces formulas to a simpler form given substitutions.
3      # Useful in computing boundary cases.
4
5      formula_dict = {tuple(case): formula for (case, formula) in formulas}
6
7      x_limits = {-1, 1, -a/b, a/b}
8      y_limits = {-1, 1}
9      p_limits = {a - b, -2*a, -a - b, a + b, 2*a, b - a, 0*x}
10
11     formulas = [ ]
12     case = assume2(*case)
13
14     # Sort the x, y, and p limits in ascending order.
15     x_limits = sorted(x_limits, cmp=cmp_symbolic)
16     y_limits = sorted(y_limits, cmp=cmp_symbolic)
```

```
    p_limits = sorted(p_limits, cmp=cmp_symbolic)

    x_intervals = intervals(x_limits, unbounded=True)
    y_intervals = intervals(y_limits, unbounded=True)
    p_intervals = intervals(p_limits, unbounded=True)

    # We iterate over the x, y, p intervals.

    for (xmin, xmax), (ymin, ymax), (pmin, pmax) in \
            product(x_intervals, y_intervals, p_intervals):

        # Determine if this is an empty case.

        max_ay = a*ymin if ymin is not -infinity else infinity
        min_ay = a*ymax if ymax is not infinity else -infinity

        max_bx = b*xmax if xmax is not infinity else infinity
        min_bx = b*xmin if xmin is not -infinity else -infinity

        if bool(pmax <= min_bx - max_ay) or bool(pmin >= max_bx - min_ay):
            continue

        # And also check if this is a degenerate case.
        xmin_sub = xmin(**subs) if type(xmin) \
            is sage.symbolic.expression.Expression \
            else xmin
```

```
43          xmax_sub = xmax(**subs) if type(xmax) \
44              is sage.symbolic.expression.Expression \
45              else xmax
46          ymin_sub = ymin(**subs) if type(ymin) \
47              is sage.symbolic.expression.Expression \
48              else ymin
49          ymax_sub = ymax(**subs) if type(ymax) \
50              is sage.symbolic.expression.Expression \
51              else ymax
52          pmin_sub = pmin(**subs) if type(pmin) \
53              is sage.symbolic.expression.Expression \
54              else pmin
55          pmax_sub = pmax(**subs) if type(pmax) \
56              is sage.symbolic.expression.Expression \
57              else pmax
58
59          if bool(xmin_sub == xmax_sub) or \
60                  bool(ymin_sub == ymax_sub) or \
61                  bool(pmin_sub == pmax_sub):
62              continue
63
64          conditions = [ ]
65          conditions_sub = [ ]
66
67          if xmin is not -infinity:
68              conditions.append(xmin <= x)
```

```
69              conditions_sub.append(xmin_sub <= x)
70          if xmax is not infinity:
71              conditions.append(x <= xmax)
72              conditions_sub.append(x <= xmax_sub)
73          if ymin is not -infinity:
74              conditions.append(ymin <= y)
75              conditions_sub.append(ymin_sub <= y)
76          if ymax is not infinity:
77              conditions.append(y <= ymax)
78              conditions_sub.append(y <= ymax_sub)
79          if pmin is not -infinity:
80              conditions.append(pmin <= p)
81              conditions_sub.append(pmin_sub <= p)
82          if pmax is not infinity:
83              conditions.append(p <= pmax)
84              conditions_sub.append(p <= pmax_sub)
85
86          formula = formula_dict[tuple(conditions)](**subs)
87          formulas.append((conditions_sub, formula))
88
89      forget(*case)
90      return formulas
```

Generation and export of RPN code

The following generate formulas for **all** the cases of θ, and writes to files for use by the `brokenray` module.

```
kernel = rpncalc.PW_Function()

for case in (case1, case2):
    formulas = Phi_symbolic(phi, case)
    formulas = [(subcase, hornerrational(formula, x, y, p, a, b))
        for (subcase, formula) in formulas]

    kernel_part = rpncalc.piecewise_to_rpn(formulas)
    case_rpn = rpncalc.conditions_to_rpn(*case)
    kernel.append((case_rpn, kernel_part))

    # Compute the boundary cases
    if case is case2:
        for bdcase, subs in ((bdcase1, subs1), (bdcase2, subs2)):
            reduced = reduce_formulas(formulas, case, **subs)
            reduced = [(subcase,
                hornerrational(formula, x, y, p, a, b))
                for (subcase, formula) in reduced]

            case_rpn = rpncalc.conditions_to_rpn(*bdcase)
            kernel_part = rpncalc.piecewise_to_rpn(reduced)
            kernel.insert(0, (case_rpn, kernel_part))

f = bz2.BZ2File("fmsbrt-inv-kernel.rpn.bz2", "w")
print >>f, kernel.encode().encode("utf8")
f.close()
```

```
27
28 placeholder = rpncalc.PW_Function()
29
30 # To separate each case into its own file.
31 for k, (case_rpn, kernel_part) in enumerate(kernel):
32     f = bz2.BZ2File("fmsbrt-inv-kernel-%d.rpn.bz2" % (k+1), "w")
33     print >>f, kernel_part.encode().encode("utf8")
34     f.close()
35
36     placeholder.append((case_rpn, rpncalc.RPNProgram()))
37
38 f = bz2.BZ2File("fbrt-inv-kernel-placeholder.rpn.bz2", "w")
39 print >>f, placeholder.encode().encode("utf8")
40 f.close()
```

A.2 Reconstruction

`FMSBrokenRayInversion` class

This is the Python class used to generate a convolution kernel and perform inversion. This class depends on the existence of all the files generated in the previous section. The code is contained in `brokenray/florescu.py`, part of the `brokenray` module. The module can either be imported directly into Python, or be used as command line utility to perform the inversions. This module requires the `numpy`, `scipy`, and `rpncalc` modules, as well as the `fmsbrt-inv-kernel.*.rpn.bz2` files generated in stage 1. This module does not require SAGE.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

from numpy import array, zeros, linspace, meshgrid, ndarray
from numpy import float64, float128, complex128, complex256
from numpy import exp, sin, cos, tan, arcsin, arctan
from numpy import floor, ceil
from numpy.fft import fft, ifft
from numpy import pi
from numpy import concatenate as concat
from scipy.signal import fftconvolve as conv

import os
import bz2
import rpncalc


class FMSBrokenRayInversion(object):

    # This class defines a series of methods used in the reconstruction
    # of a function μ_t from its Florescu, et. al. Broken Ray transform
    # given as an H×W array, sampling values on
    # ROI = [x_min, x_max] × [y_min, y_max].
    # It will be assumed that 0 < θ ≤ π/2. Scattering angles greater than π/2
    # are not supported.

```

```
27      # Initialize 'kernel_generator'. Currently a placeholder that does
28      # not actually hold the RPN code needed. This class will load the
29      # proper RPN code from an external file as it is needed.
30
31      dirname, fname = os.path.split(__file__)
32      src = "fbrt-inv-kernel-placeholder.rpn.bz2"
33
34      f = bz2.BZ2File(os.path.join(dirname, src), "r")
35      rpn = f.read().decode("utf8")
36      f.close()
37
38      kernel_generator = rpncalc.decode(rpn)
39      del dirname, fname, src, f, rpn
40
41      def __init__(self, H, W, xmin, xmax, ymin, ymax, theta):
42
43          # Creates an instance of FMSBrokenRayInversion,
44          # with the following parameters:
45
46          # H, W -- The shape of the data.
47          # xmin, xmax -- The x bounds on the data.
48          # ymin, ymax -- The y bounds on the data.
49          # theta -- The fixed scattering angle.
50
51          self.H = H
52          self.W = W
53          self.xmin = xmin
```

```
54          self.xmax = xmax
55          self.ymin = ymin
56          self.ymax = ymax
57          self.theta = theta
58          if not (0 < self.theta <= pi / 2):
59              raise ValueError, \
60                  "Values of theta outside of (0, pi/2) not supported."
61
62      @property
63      def dx(self):
64          return float128(self.xmax - self.xmin) / (self.W - 1)
65
66      @property
67      def dy(self):
68          return float128(self.ymax - self.ymin) / (self.H - 1)
69
70      def _extend(self, data, alpha):
71          # Extend data to a larger array needed to perform
72          # inversion. With an a priori assumption that a function
73          # μ_t to be reconstructed is supported in ROI, then
74          # knowledge of Bμ_t on ROI is sufficient, given that Bμ_t
75          # is then uniquely determined by its values on ∂ROI.
76
77          # Since an approximation to μ_t is obtained via convolution
78          # of Bμ_t with B^{-1}φ_α, it is necessary to extend the data
79          # to a rectangle R so that supp(B^{-1}φ_α(z − ·)Bμ_t) ⊆ R
```

```
80          # for each z ∈ ROI.
81
82          # To reduce the amount of self.* in the code below...
83          dx, dy = self.dx, self.dy
84          xmin, xmax = self.xmin, self.xmax
85          ymin, ymax = self.ymin, self.ymax
86          theta = self.theta
87          H, W = self.H, self.W
88
89          # A very quick validation test, that 'data' is an
90          # H  ×  W  array.
91          if data.shape != (H, W):
92              raise ValueError, \
93                  "data.shape must be (%d, %d)." % (H, W)
94
95          # At a minimium, we take R = [x_1, x_2] × [y_1, y_max], where:
96          #       -- x_1 = x_min − ((α − h) tan θ/2 − α) cos θ
97          #       -- x_2 = x_max + (α + h) tan θ/2 + α
98          #       -- y_1 = y_min − (α tan θ/2 − α − w) sin (θ)
99          #       -- w = x_max − x_min
100         #       -- h = y_max − y_min
101         # It is not necessary to extend above the existing ROI.
102
103         ext_left = ((ymax - ymin - alpha)
104                     * tan(theta / 2) - alpha) * cos(theta)
105         ext_right = (alpha + ymax - ymin)*tan(theta / 2)
```

```
106         ext_down = (alpha * tan(theta / 2) + alpha
107                     + xmax - xmin) * sin(theta)
108
109         # If this module is running inside an instance of SAGE,
110         # coerce these variables into float128, because numpy
111         # does not play well with SAGE data types.
112
113         xmin_ext = float128(xmin - ext_left)
114         xmax_ext = float128(xmax + ext_right)
115         ymin_ext = float128(ymin - ext_down)
116
117         W_ext = W + int(ceil(ext_left / dx)
118                         + ceil(ext_right / dx))
119
120         H_ext = H + int(ceil(ext_down / dy))
121
122         # Extending to the right is easy.
123         # Repeat the boundary data at the right.
124         # But we will do that later.
125
126         # Extending to the left and downard takes more effort.
127         # We must take boundary data and translate it as it is
128         # extended. We use a Fourier multiplier in both
129         # extensions to perform the job.
130
131         # Extract left boundary data, pad it in the y direction
```

```
132         # by ext_left · tan θ, then repeat it in the x direction by
133         # ext_left.
134
135         pad_up = int(ceil(
136             (-(alpha - ymax + ymin) * tan(theta / 2) - alpha) *
137             sin(theta) / dy))
138         pad_left = int(ceil(ext_left / dx))
139
140         # Left boundary data, padded upward so that we may later
141         # throw away the garbage data that will appear when we
142         # perform the translation needed to repeat the data
143         # in the direction of −v⃗_2...
144         left = concat((data[::, 0], zeros(pad_up)))
145
146         # ... then repeated to have width 'pad_left'.
147         left = array((left,) * pad_left).transpose()
148
149         # We now prepare the Fourier multiplier.
150         # Initialize the xη-grid.
151
152         X = float128(linspace(xmin_ext, xmin - dx,
153                               pad_left))
154
155         H_up = H + pad_up
156         eta_max = float128(2 * pi * (1 - 1.0 / H_up) / dy)
157
```

```
158         Eta = float128(linspace(0, eta_max, H_up))
159         Eta[H_up / 2:] -= float128(2 * pi / dy)
160
161         X, Eta = meshgrid(X, Eta)
162
163         # 'translation' - array giving amount of translation
164         # to be performed in the vertical direction.
165         translation = float128(tan(theta)) * (X - xmin)
166
167         # The Fourier multiplier used to perform the translation.
168         multiplier = exp(-1j * Eta * translation)
169
170         # Note the 'axis=0' keyword argument.
171         left_fft = fft(complex128(left), axis=0)
172         left = ifft(left_fft * multiplier, axis=0)
173
174         # Concatenate 'left' to 'data', after discarding
175         # garbage data.
176         data_ext = concat((left[:-pad_up], data), axis=1)
177
178         # We now do the same in extending the data downward.
179         pad_right = int(ceil(ext_down * cos(theta) / sin(theta) / dx))
180         pad_down = int(ceil(ext_down / dy))
181
182         # Start with the bottom row of the data...
183         bottom = concat((data_ext[0], zeros(pad_right)))
```

```
184         # ...repeated to have height 'pad_down'.
185         bottom = array((bottom,) * pad_down)
186
187         # Initialize the ξy-grid to prepare the
188         # Fourier multiplier.
189         Y = float128(linspace(ymin_ext, ymin - dy,
190                               pad_down))
191
192         W += pad_right + pad_left
193         xi_max = float128(2 * pi * (1 - 1.0 / W) / dx)
194
195         Xi = float128(linspace(0, xi_max, W))
196         Xi[W / 2:] -= float128(2 * pi / dx)
197
198         Xi, Y = meshgrid(Xi, Y)
199
200         translation = float128(cos(theta) / sin(theta)) * (Y - ymin)
201         multiplier = exp(-1j * Xi * translation)
202
203         bottom_fft = fft(complex128(bottom), axis=1)
204         bottom = ifft(bottom_fft * multiplier, axis=1)
205
206         data_ext = concat((bottom[:, :-pad_right], data_ext),
207                           axis=0)
208
209         # Now we extend to the right and return the result.
```

```
210         pad_right = int(ceil(ext_right / dx))
211         data_ext = concat((data_ext,) +
212                           (data_ext[:, -1:],) * pad_right, axis=1)
213
214         return data_ext
215
216     def make_kernel(self, alpha):
217         """Prepares a convolution kernel for use to perform numerical
218         inversion of the Florescu Broken Ray transform."""
219
220         # We now make a convolution kernel $\mathcal{B}^{-1}\phi_\alpha$.
221         # We must determine a rectangle $S$ needed so that
222         # $\operatorname{supp}\left(\mathcal{B}\mu_t\left(z-\cdot\right)\mathcal{B}^{-1}\phi_\alpha\right)\subseteq S$ for each $z\in\mathrm{ROI}$.
223         # This time, we take $S=[x_1,x_2]\times[y_1,y_2]$, where:
224         #       -- $x_1=\left(-\left(\alpha+h\right)\tan\frac{\theta}{2}-\alpha+w\right)\cos\theta-w$
225         #       -- $x_2=\alpha+\left(\alpha+h\right)\tan\frac{\theta}{2}$
226         #       -- $y_1=-h$
227         #       -- $y_2=\left(\left(\alpha-h\right)\tan\frac{\theta}{2}+\alpha+w\right)\sin\theta+h$
228         #       -- $w=x_{\max}-x_{\min}$
229         #       -- $h=y_{\max}-y_{\min}$
230
231         dx, dy = self.dx, self.dy
232         xmin, xmax = self.xmin, self.xmax
233         ymin, ymax = self.ymin, self.ymax
234         theta = self.theta
235
```

```
236         # Determine the bounds on the convolution kernel array.
237         xmin_ker = (-(alpha + ymax - ymin)
238                     * tan(theta / 2)
239                     - alpha + xmax - xmin) \
240             * cos(theta) \
241             - xmax + xmin
242         xmax_ker = alpha + (alpha + ymax - ymin) \
243             * tan(theta / 2)
244         ymin_ker = ymin - ymax
245         ymax_ker = ((alpha - ymax + ymin)
246                     * tan(theta / 2)
247                     + alpha + xmax - xmin) \
248             * sin(theta) \
249             + ymax - ymin
250
251         # Adjust these bounds to be integer multiples of Δx
252         # and Δy as needed. Also pad around the edge
253         # by a single pixel.
254
255         xmin_ker = floor(xmin_ker / dx) * dx
256         xmax_ker = ceil(xmax_ker / dx) * dx
257         ymin_ker = floor(ymin_ker / dy) * dy
258         ymax_ker = ceil(ymax_ker / dy) * dx
259
260         W = int(ceil(xmax_ker / dx) - floor(xmin_ker / dx)) + 1
261         H = int(ceil(ymax_ker / dy) - floor(ymin_ker / dy)) + 1
262
```

```
263         # Prepare xy grid, with an extra pixel border
264         # around the edge.
265
266         X = float128(linspace(xmin_ker - dx, xmax_ker + dx, W + 2))
267         Y = float128(linspace(ymin_ker - dy, ymax_ker + dy, H + 2))
268         X, Y = meshgrid(X, Y)
269
270         A = cos(float128(theta)) - 1
271         B = sin(float128(theta))
272
273         # Prepare pq grid.
274
275         P = B * X - A * Y
276         Q = A * X + B * Y
277
278         alpha = float128(alpha)
279
280         case_num = self.kernel_generator.findcase(a=A, b=B)
281         case, formula = self.kernel_generator[case_num]
282
283         if len(formula) == 0:
284             dirname, fname = os.path.split(__file__)
285             src = "fbrt-inv-kernel-%d.rpn.bz2" % (case_num + 1)
286
287             f = bz2.BZ2File(os.path.join(dirname, src), "r")
288             rpn = f.read().decode("utf8")
289             f.close()
```

```
290
291             self.kernel_generator[case_num] = (case,
292                                                rpncalc.decode(rpn))
293
294         Phi = alpha * self.kernel_generator(a=A, b=B,
295                                             x=X / alpha, y=Y / alpha,
296                                             p=P / alpha, q=Q / alpha)
297
298         Ker = (Phi[2:] + Phi[:-2] - 2 * Phi[1:-1]) / dx
299         Ker = (Ker[:, 2:] + Ker[:, :-2] - 2 * Ker[:, 1:-1]) / dy
300
301         return Ker
302
303     def reconstruct(self, data, kernel, alpha):
304         # This is where the reconstruction finally happens.
305
306         # Apparently, scipy's fftconvolve does not support
307         # float128. But that is okay. The purpose of using
308         # 128-bit precision was for computation of 'Ker'
309         # in the above method.
310         data_ext = self._extend(data, alpha)
311         result = conv(kernel, data_ext)
312
313         # Need to determine how 'data_ext' is to be cropped.
314
315         dx, dy = self.dx, self.dy
316         xmin, xmax = self.xmin, self.xmax
```

```
317         ymin, ymax = self.ymin, self.ymax
318         theta = self.theta
319
320         # Determine the bounds on the convolution kernel array.
321         xmin_ker = (-(alpha + ymax - ymin)
322                     * tan(theta / 2)
323                     - alpha + xmax - xmin) \
324             * cos(theta) \
325             - xmax + xmin
326         xmax_ker = alpha + (alpha + ymax - ymin) \
327             * tan(theta / 2)
328         ymin_ker = ymin - ymax
329         ymax_ker = ((alpha - ymax + ymin)
330                     * tan(theta / 2)
331                     + alpha + xmax - xmin) \
332             * sin(theta) \
333             + ymax - ymin
334
335         # How much the data was extended
336         ext_left = ((ymax - ymin - alpha)
337                     * tan(theta / 2) - alpha) * cos(theta)
338         ext_right = (alpha + ymax - self.ymin)\
339             * tan(theta / 2)
340         ext_down = (alpha * tan(theta / 2) + alpha
341                     + xmax - xmin) * sin(theta)
342
```

```
343         crop_bottom = int(ceil(ext_down / dy) - floor(ymin_ker / dy))
344         crop_top = int(ceil(ymax_ker / dy))
345         crop_left = int(ceil(ext_left / dx) - floor(xmin_ker / dx))
346         crop_right = int(ceil(xmax_ker / dx) + ceil(ext_right / dx))
347 
348         return result[crop_bottom:-crop_top, crop_left:-crop_right]
```

To run a reconstruction from the Python prompt, given an array **Qf**, along with $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, and θ, one executes the following, choosing a reconstruction parameter α:

```
1 >>> H, W = Qf.shape
2 >>> brinv = FMSBrokenRayInversion(H, W, xmin, xmax, ymin, ymax, theta)
3 >>> alpha = 3*sqrt(brinv.dx*brinv.dy)
4 >>> kernel = brinv.generate_kernel(alpha)
5 >>> reconstructed = brinv.reconstruct(Qf, kernel, alpha)
```

The file `array2im.py` provides the ability to convert a 2-dimensional NumPy array into an image file through the Python Imaging Library (the PIL module). This file can also be used as both a Python module and as a command line utility. At the time of this writing, `array2im.py` generates images with red corresponding to positive values, and blue to negative values, with brightness corresponding to absolute value. The imaginary parts of complex-valued arrays are ignored.

A.3 Results

Reconstructions of images are from simulated data. Of note is the visible error in the reconstruction of a square whose sides are not parallel to the coordinate axes, but instead having a pair of its sides parallel to the terminal beam. Each of these reconstructions were done on a 1025×1025 grid, with regions of interest $[-6, 6] \times [-6, 6]$.

FIGURE A.1: Recovery of the characteristic function of a square from its Broken Ray transform, with scattering angle $\theta = \frac{\pi}{3}$. Two of the edges are parallel to the initial beam direction.

A reconstruction of a square in Figure A.1, whose sides are parallel to the coordinate axes, is shown to have almost no error, save for some lines corresponding to the direction of integration, passing through the vertices of the square. The error towards the lower left is due to the manner in which the data is repeated.

The square in Figure A.2, on the other hand, is tilted so that two of its sides correspond to the terminal direction is reconstructed with visible noise. Error was highly abundant when taking $\alpha = 0.05$. The reconstruction shown is given with $\alpha = 0.15$, which reduced the error, but at the cost of blurring the image.

This can be remedied by reconstructing from data sampled on an alternate grid that lines up with both the initial and terminal directions. This would be equivalent, up to scalar

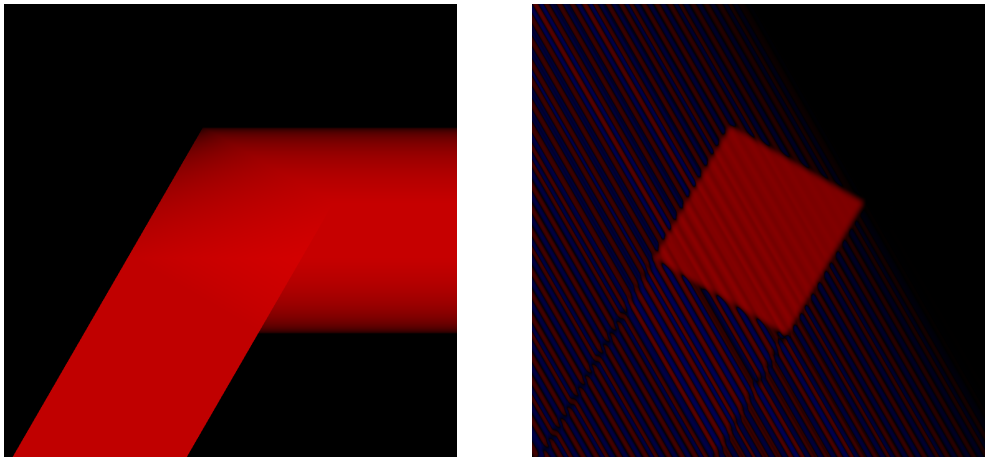

FIGURE A.2: Recovery of the characteristic function of a tilted square from its Broken Ray transform, with scattering angle $\theta = \frac{\pi}{3}$. Two of the edges are parallel to the terminal beam direction.

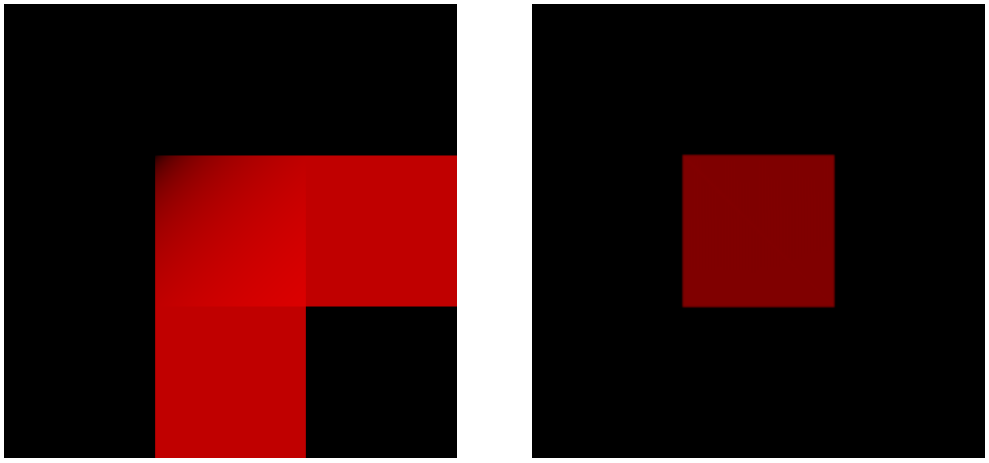

FIGURE A.3: Recovery of the characteristic function of a square from its Broken Ray transform, with scattering angle $\theta = \frac{\pi}{2}$.

multiple, to reconstructing with a scattering angle of $\frac{\pi}{2}$. In this setting, $a = -1$ and $b = 1$, and so we would be able to simplify the symbolic expression for $\kappa \star \mathcal{B}^{-1}\phi$, and therefore cut down on the computation time. Indeed, the reconstruction in this setting appears to be a perfect reconstruction with almost no perceptible error, as shown in Figure A.3. Also notice that the repetition of the boundary data at the bottom becomes trivial.

Figure A.4 depicts a reconstruction of a function f where $WF(f) \setminus WF(\mathcal{B}f)$ is nonempty,

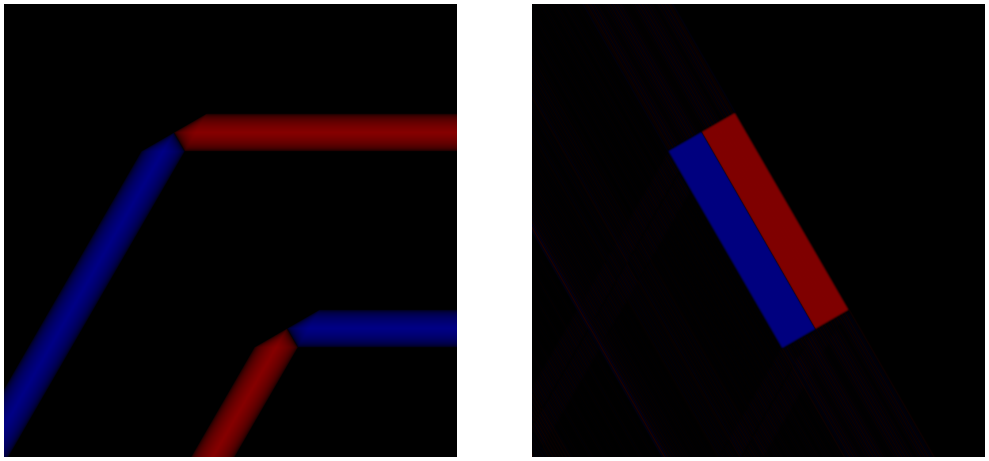

FIGURE A.4: Recovery of a piecewise-constant function (taking values 1 and -1 on its support) having singularities vanish in its Broken Ray transform.

and in particular, consists of three line segments. This is similar to what was shown in Example 4.2.2. Here, the scattering angle of $\frac{\pi}{3}$.

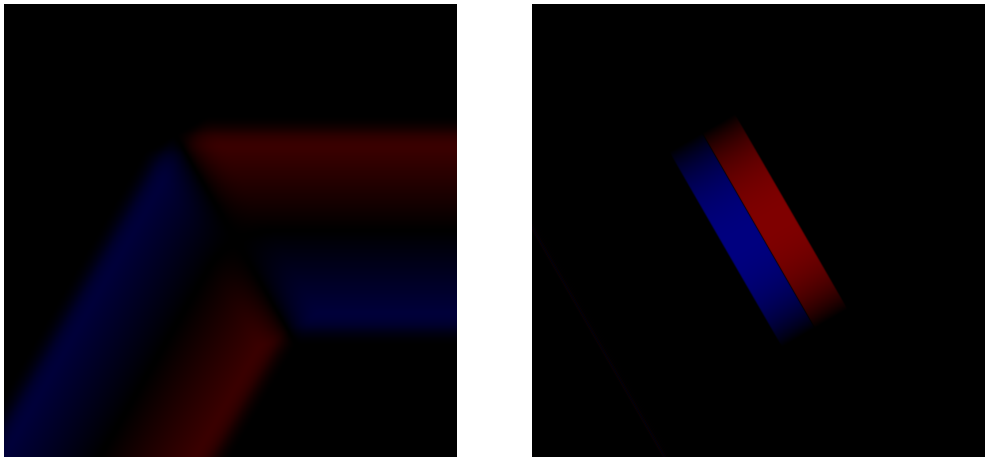

FIGURE A.5: Recovery of a piecewise-quadratic function.

In Figure A.5, we reconstruct the function f, defined by

$$f(x,y) = \begin{cases} \operatorname{sgn}(u)\left(v^2 - 9\right), & \text{if } |u| < 1,\ |v| < 3, \\ 0, & \text{otherwise,} \end{cases}$$

where the uv-plane is a counter-clockwise rotation of the xy-plane by $\frac{\pi}{6}$. As with the function shown in Figure A.4, notice that $\mathcal{B}f$ here vanishes along the v-axis, and this time,

the size of the jump discontinuity in f along the v-axis varies between $v = -3$ and $v = 3$.

B Numerical inversion of the Polar Broken Ray transform

The code in this section is found in `brokenray/polar.py` of the `brokenray` package. We first import the necessary modules and RPN functions that were originally generated in SAGE. Inversion requires modules `iqueue`, which is derived from Python's `queue` module, as well as the `rpncalc` and `parallel` modules. At the time of this writing, these modules, and successive versions, can be downloaded from their respective repositories at http://github.com/shersonb/.

Also, the file `polar-to-cartesian.py` is a standalone Python script used to filter and resample data given on a polar grid onto a Cartesian grid, and is included with the `brokenray` module.

As with the reconstructions of the Florescu, et. al. Broken Ray transform, the `array2im` module can also be used to convert the NumPy arrays resulting from the following inversion code into graphic files, and is found at http://github.com/shersonb/python-array2im/.

We first import some required tools:

```
1 from numpy import array, zeros, linspace, meshgrid, ndarray, diag
2 from numpy import uint8, float64, int8, int0, float128, complex128
3 from numpy import exp, sqrt, cos, tan, arctan
4 from numpy import minimum, maximum
5 from numpy import ceil, floor
```

```
6 from numpy import matrix as npmatrix
7 from numpy.fft import fft, ifft
8 from numpy import pi
9 from scipy.linalg import solve_triangular as solve
10 from scipy.signal import fftconvolve as conv
11 from scipy.ndimage import geometric_transform as transform
12
13 # We will make use of *reentrant* locks.
14 from threading import RLock as Lock
15 from threading import Condition, Thread
16
17 # This module is a modification on python's queue module,
18 # which allows one to interrupt a queue.
19 import iqueue
20
21 # This is a module written to execute code in parallel.
22 # While python is limited by the Global Interpreter Lock,
23 # numerical operations on NumPy arrays are generally not
24 # limited by the GIL.
25 import parallel
26
27 # This module allows the conversion of SAGE symbolic expressions
28 # to RPN code through the symbolic_to_rpn. RPNProgram is a subclass
29 # of list that comes equipped with a __call__ method that implements
30 # execution of the RPN code.
31 import rpncalc
```

Much of the effort to numerically invert the Polar Broken Ray transform goes into computation and inversion of the matrix given in (5.3.1). Computation of $\kappa * B_n$ is done via approximation with power series, giving us

$$\kappa * B_n\left(r_j, s_k\right) = \begin{cases} 2\sum_{m\in 2\mathbb{N}} \frac{1}{(m+2)!} \cdot \frac{\partial^m B_n}{\partial r^m}\left(r_j, s_k\right) \cdot \Delta r^{m+1} & j < k, \\ \sum_{m\in \mathbb{N}} \frac{1}{(m+2)!} \cdot \frac{\partial^m B_n}{\partial r^m}\left(r_j, s_k\right) \cdot \Delta r^{m+1} & j = k. \end{cases}$$

B.1 Utility functions

We introduce variables as follows:

$$q = \frac{s}{r}, \quad u = \arcsin\left(q \sin\phi\right),$$
$$w = e^{i(\phi - u)}, \quad v = r \cdot \frac{\partial u}{\partial r} = -q \cdot \frac{\partial u}{\partial q},$$

so that

$$\begin{aligned} K_n &= K\left(\cdot, n\right) \\ &= w^n \cdot \frac{\partial u}{\partial q} \csc\phi \\ &= -\frac{r}{s} w^n v \csc\phi. \end{aligned}$$

Notice that

$$\frac{\partial w}{\partial r} = -\frac{iwv}{r}, \quad \frac{\partial v}{\partial r} = -\frac{v\left(1+v^2\right)}{r},$$

and so

$$\begin{aligned} B_n\left(r, s\right) &= \frac{1}{r}\frac{\partial K_n}{\partial q} \\ &= -\frac{r}{s^2} w^n v^2 \left(v + in\right) \csc\phi. \end{aligned}$$

Furthermore, recursively defining

$$c_m\left(x, n^2\right) = n^2 d_{m-1}\left(x, n^2\right) - \left(1 + 3x + m\right) c_{m-1}\left(x, n^2\right) - 2x\left(1+x\right)\frac{\partial c_{m-1}}{\partial x}\left(x, n^2\right),$$

$$d_m\left(x,n^2\right)=-xc_{m-1}\left(x,n^2\right)-\left(2x+m\right)d_{m-1}\left(x,n^2\right)-2x\left(1+x\right)\frac{\partial d_{m-1}}{\partial x}\left(x,n^2\right),$$

for $m\geq 1$, subject to initial values

$$c_0\left(x,n^2\right)=d_0\left(x,n^2\right)=-1,$$

we find that

$$\frac{\partial^m B_n}{\partial r^m}=\frac{w^n v^2 r^{1-m}\csc\phi}{s^2}\left(vc_m\left(v^2,n^2\right)+nd_m\left(v^2,n^2\right)i\right),\quad m\geq 0,$$

and hence,

$$\kappa * B_n\left(r_j,s_k\right)=\frac{w^n v^2 r\csc\phi\Delta r}{s^2}\cdot\begin{cases}2\sum_{m\in 2\mathbb{N}}\frac{1}{(m+2)!}\cdot\left(vc_m+nd_m i\right)\left(\frac{\Delta r}{r}\right)^m & j<k,\\ \sum_{m\in\mathbb{N}}\frac{1}{(m+2)!}\cdot\left(vc_m+nd_m i\right)\left(\frac{\Delta r}{r}\right)^m & j=k.\end{cases}$$

In the interests of time, we will forego use of SAGE's symbolic computational capabilities to compute the polynomials c_m and d_m, and instead generate the coefficients using NumPy matrix multiplication.

Assume $C^{(m)}$ and $D^{(m)}$ are matrices representing the coefficients of c_m and d_m:

$$c_m\left(x,n^2\right)=\sum_{j,k\geq 0}C^{(m)}_{j,k}x^j n^{2k},$$

$$d_m\left(x,n^2\right)=\sum_{j,k\geq 0}D^{(m)}_{j,k}x^j n^{2k}.$$

Then by the recursions given above, $C^{(m)}$ and $D^{(m)}$ are $(m+1)\times\left(\left\lceil\frac{m}{2}\right\rceil+1\right)$ and $(m+1)\times\left(\left\lfloor\frac{m}{2}\right\rfloor+1\right)$, respectively, and are given by

$$\begin{aligned}C^{(m)}&=E_m D^{(m-1)}X^T_{\frac{m+1}{2}}\\&\quad-\left(\left(1+m\right)E_m+3X_m+2\left(E_m X_{m-1}+X_m X_{m-1}\right)\Delta_m\right)C^{(m-1)}E^T_{\frac{m+1}{2}}\\&\in\mathbb{R}^{(m+1)\times\frac{m+3}{2}},\end{aligned}$$

$$D^{(m)} = X_m C^{(m-1)} - \left(mE_m + 2X_m + 2\left(E_m X_{m-1} + X_m X_{m-1}\right)\Delta_m\right) D^{(m-1)}$$
$$\in \mathbb{R}^{(m+1)\times\frac{m+1}{2}},$$

for m odd, and

$$C^{(m)} = E_m D^{(m-1)} X_{\frac{m}{2}}^T$$
$$- \left((1+m)E_m + 3X_m + 2\left(E_m X_{m-1} + X_m X_{m-1}\right)\Delta_m\right) C^{(m-1)}$$
$$\in \mathbb{R}^{(m+1)\times\left(\frac{m}{2}+1\right)},$$

$$D^{(m)} = X_m C^{(m-1)} - \left(mE_m + 2X_m + 2\left(E_m X_{m-1} + X_m X_{m-1}\right)\Delta_m\right) D^{(m-1)} E_{\frac{m}{2}}^T$$
$$\in \mathbb{R}^{(m+1)\times\left(\frac{m}{2}+1\right)},$$

for m even, where

$$\Delta_m = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & m-1 \end{bmatrix} \in \mathbb{R}^{(m-1)\times m},$$

$$X_m = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times m}, \quad E_m = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times m}.$$

Computation of $C^{(m)}$ and $D^{(m)}$ are done through the `getcoeffs` method of an instance of `_CD_RPN`, while the RPN code is generated by invoking `__getitem__`.

```
# Some utility functions used by PolarBrokenRayInversion
def _E(m):
    return int0(npmatrix(diag((1,) * int(m + 1), k=0)[:, :-1]))


def _X(m):
    return int0(npmatrix(diag((1,) * int(m), k=-1)[:, :-1]))

def _Del(m):
    return int0(npmatrix(diag(xrange(1, int(m)), k=1)[:-1]))

class _CD_RPN:

    def __init__(self):
        self.coeffs = [(npmatrix((-1,)), npmatrix((-1,)))]
        self.rpn = [(rpncalc.RPNProgram([-1]), rpncalc.RPNProgram([-1]))]

        # In case this class is utilized by multiple threads.
        self.lock = Lock()

    def getcoeffs(self, m):
        # Returns coefficients for c_m and d_m.
        # If they already exist in cache, just return what is there.
        with self.lock:
            if len(self.coeffs) <= m:
                # Need to generate coefficients for c_m and d_m.
```

```
27                # Fetch the coefficients for c_{m-1} and d_{m-1}.
28                C, D = self.getcoeffs(m - 1)
29
30                if m % 2:  # m is odd
31                    C_new = _E(m)*D*_X((m + 1)/2).transpose() \
32                        - ((1 + m)*_E(m) + 3*_X(m)
33                           + 2*(_E(m) + _X(m))*_X(m - 1)*_Del(m))*C \
34                        *_E((m + 1)/2).transpose()
35                    D_new = _X(m)*C - (m*_E(m) + 2*_X(m)
36                               + 2 * (_E(m) + _X(m))*_X(m - 1)*_Del(m))*D
37
38                else:  # m is even
39                    C_new = _E(m)*D*_X(m/2).transpose() \
40                        - ((1 + m)*_E(m) + 3*_X(m)
41                           + 2*(_E(m) + _X(m))*_X(m - 1)*_Del(m))*C
42
43                    D_new = _X(m)*C - (m*_E(m) + 2*_X(m)
44                               + 2*(_E(m) + _X(m))*_X(m - 1)*_Del(m))*D \
45                        *_E(m / 2).transpose()
46
47                self.coeffs.append((C_new, D_new))
48
49            return self.coeffs[m]
50
51    def __getitem__(self, m):
52        n2 = rpncalc.wild("n2")
```

```
53          v2 = rpncalc.wild("v2")
54          mul = rpncalc.rpn_funcs["*"]
55          add = rpncalc.rpn_funcs["+"]
56          # Returns RPN code for c_j and d_j. Generate on the fly if needed.
57          with self.lock:
58              while len(self.rpn) <= m:
59                  cm_rpn = []
60                  dm_rpn = []
61
62                  C, D = self.getcoeffs(len(self.rpn))
63
64                  # Generate RPN code for c_j and d_j.
65                  for row in array(C[::-1]):
66                      npoly_rpn = []
67                      for coeff in row[::-1]:
68                          if coeff:
69                              if len(npoly_rpn):
70                                  npoly_rpn.extend([n2, mul])
71                                  npoly_rpn.extend([coeff, add])
72                              else:
73                                  npoly_rpn.append(coeff)
74                          elif len(npoly_rpn):
75                              npoly_rpn.extend([n2, mul])
76                      if len(cm_rpn):
77                          cm_rpn.extend([v2, mul])
78                          cm_rpn.extend(npoly_rpn)
```

```
79                      cm_rpn.append(add)
80                  else:
81                      cm_rpn.extend(npoly_rpn)
82
83              for row in array(D[::-1]):
84                  npoly_rpn = []
85                  for coeff in row[::-1]:
86                      if coeff:
87                          if len(npoly_rpn):
88                              npoly_rpn.extend([n2, mul])
89                              npoly_rpn.extend([coeff, add])
90                          else:
91                              npoly_rpn.append(coeff)
92                      elif len(npoly_rpn):
93                          npoly_rpn.extend([n2, mul])
94                  if len(dm_rpn):
95                      dm_rpn.extend([v2, mul])
96                      dm_rpn.extend(npoly_rpn)
97                      dm_rpn.append(add)
98                  else:
99                      dm_rpn.extend(npoly_rpn)
100             self.rpn.append(
101                 (rpncalc.RPNProgram(cm_rpn),
102                  rpncalc.RPNProgram(dm_rpn)))
103         return self.rpn[m]
```

The following is a numerical implementation of $\frac{\partial}{\partial s}$, based on approximating

$$\frac{\partial f}{\partial s} = \psi'_\alpha \star f,$$

where $\psi_\alpha \to \delta$ in distribution as $\alpha \to 0$. We only require ψ_α to have a first-order weak derivative.

```
class Sderiv:

    def __init__(self, alpha):
        self.alpha = alpha

    def __call__(self, A, ds):
        H, W = A.shape
        psi = rpncalc.decode(u"\xab x 3 ^ 4 / +/- 3 x * 4 / + \xbb")
        N = ceil(self.alpha / ds)
        X = linspace(-N * ds - ds, N * ds + ds, 2 * N + 3)

        Psi = psi(x=X / self.alpha)
        Psi[X > self.alpha] = psi(x=1)
        Psi[X < -self.alpha] = psi(x=-1)

        stencil = (Psi[:-2] + Psi[2:] - 2 * Psi[1:-1]) / ds

        diff = conv([stencil], A)

        return N, N, diff[:, 2 * N:-2 * N]
```

B.2 Reconstruction

`PolarBrokenRayInversion` class

This task is parallelizable, and so we will write our code as such, using the `parallel` module.

```
class PolarBrokenRayInversion(parallel.BaseTaskClass):
    _cd = _CD_RPN()
    _u = rpncalc.decode(u"\xab q phi sin * arcsin \xbb")
    _v = rpncalc.decode(u"\xab q phi sin * +/- \
             1 q 2 ^ phi sin 2 ^ * - sqrt / \xbb")
    _w = rpncalc.decode(u"\xab i phi u - * exp \xbb")
    _tm = rpncalc.decode(u"\xab i dm * n * cm v * + dlnr m ^ * \
             m 2 + ! / \xbb")
    _cf = rpncalc.decode(u"\xab dr r * v 2 ^ * phi csc * s 2 ^ / \xbb")
    _invlock = Lock()

    def __init__(self, Qf, Phi, smin, smax, alpha, nmax=200):
        # Parameters:
        # Qf -- Qf, sampled on an rθ grid.
        # Phi (ϕ) -- Scattering angle
        # rmin -- r_min, defaults to 1.
        # rmax -- r_max, defaults to 6.
        # D -- Numerical implemenation of ∂/∂r.
        # nmax -- n_max, reconstructs f̃(r,n)
        # for |n| ≤ n_max. Defaults to 200.
```

```
21
22          # This reconstruction will assume that Qf is real and exploit
23          # conjugate symmetry in the Fourier series.
24
25          # Initialize variables.
26          self.Qf = Qf
27          self.Phi = Phi
28          self.smin = smin
29          self.smax = smax
30
31          H, W = Qf.shape
32
33          self.thetamin = thetamin = -pi
34          self.thetamax = thetamax = pi*(1-2.0/H)
35          self.nmax = nmax
36
37          self.F = None
38          self.F_cartesian = None
39
40          self.lock = Lock()
41          self.status = Condition(self.lock)
42          self.jobsdone = 0
43          self.jobcount = nmax + 1
44          self.running = False
45          self.projectioncount = 0
46          self.projecting = False
47
```

```
48          self.dr = dr = ds = (smax - smin) / float(W - 1)
49          self.dtheta = dtheta = (thetamax - thetamin) / float(H)
50
51          # Compute $\widetilde{\mathcal{Q}f}$.
52          self.FQf = FQf = fft(Qf, axis=0)
53
54          # Perform differentiation of $\widetilde{\mathcal{Q}f}$.
55          D = Sderiv(alpha)
56          try:
57              clip_left, clip_right, self.DFQf = D(FQf, ds)
58          except:
59              clip_left, clip_right, self.DFQf = D(float64(FQf), ds)
60
61          # Initialize array that will store $\tilde{f}$.
62          self.Ff = zeros(self.DFQf.shape, dtype=complex128)
63
64          # Initialize $rs$ grid.
65          self.rmin = self.smin + clip_left * ds
66          self.rmax = self.smax - clip_right * ds
67          R = linspace(self.rmin, self.rmax, W - clip_left - clip_right)
68          self.R, self.S = meshgrid(R, R)
69
70          # Compute $q$, $u$, $v$, $w$, and $v^2 r * \csc(\phi) * \Delta r / s^2$.
71          self.Q = self.S / self.R
72
73          args = dict(q=self.Q, r=self.R, s=self.S, phi=self.Phi, dr=dr)
```

```
74          args["u"] = self.U = self._u(**args)
75          args["v"] = self.V = self._v(**args)
76          self.W = self._w(**args)
77          self.Factor = self._cf(**args)
78
79      def A(self, n, eps=0.0000001, p=16):
80          # Compute matrix A_n.
81
82          H, W = self.DFQf.shape
83
84          # Initialize the An matrix (as an array for now).
85          An = zeros(self.R.shape, dtype=complex128)
86
87          # First compute a partial sum for the upper triangular part.
88          # Start with m = 0
89
90          mask = self.S < self.R
91
92          Sum = zeros(self.R.shape, dtype=complex128)
93
94          for m in xrange(0, p + 1, 2):
95              cm_rpn, dm_rpn = self._cd[m]
96              Term = self._tm(v=self.V[mask], v2=self.V[mask] ** 2,
97                              dlnr=self.dr / self.R[mask],
98                              n=n, n2=n ** 2, m=m, cm=cm_rpn, dm=dm_rpn)
99
100             Sum[mask] += Term
```

```
101             mask[mask] *= abs(Term) >= eps
102             if not mask.any():
103                 break
104
105         mask = self.S < self.R
106
107         An[mask] = 2 * self.W[mask] ** n * self.Factor[mask] * Sum[mask]
108
109         # Now to do the diagonal.
110         # Since r = s here, we have q = 1, u = phi, v = -tan(phi),
111         # and w = 1.
112
113         mask = self.S == self.R
114
115         Sum = zeros(self.R.shape, dtype=complex128)
116
117         for m in xrange(0, p + 1):
118             cm_rpn, dm_rpn = self._cd[m]
119
120             Term = self._tm(v=-tan(self.Phi), v2=tan(self.Phi) ** 2,
121                             dlnr=self.dr / self.R[mask],
122                             n=n, n2=n ** 2, m=m, cm=cm_rpn, dm=dm_rpn)
123
124             Sum[mask] += Term
125             mask[mask] *= abs(Term) >= eps
126             if not mask.any():
127                 break
```

```
128
129         mask = self.S == self.R
130         An[mask] = self.Factor[mask] * Sum[mask] + \
131             array([1 - 1 / cos(self.Phi)] * W)
132
133         return npmatrix(An)
134
135     def f(self, n):
136         # This is the function that is run in parallel.
137         An = self.A(n, eps=10 ** -9, p=24)
138
139         DFQf = self.DFQf[n]
140
141         #AnInv = inv(An).transpose()
142         #Ff = array(DFQf*AnInv)[0]
143         Ff = solve(An, DFQf)
144         return Ff
145
146     def populatequeue(self, queue):
147         for n in xrange(self.nmax + 1):
148             queue.put(n)
149
150     def postproc(self, (n, Ff)):
151         with self.status:
152             self.Ff[n] = Ff
153             if n > 0:
154                 self.Ff[-n] = Ff.conjugate()
```

```
155
156             self.jobsdone += 1
157             self.status.notifyAll()
158
159     def reconstruct(self):
160         with self.lock:
161             self.F = ifft(self.Ff, axis=0)
162             return self.F
```

Given $\mathbf{Qf}$, ϕ, $s_{\min}$, $s_{\max}$, a parameter α used in a numeric implementation of $\frac{\partial}{\partial s}$, and $n_{\max}$, inversion to reconstruct $\tilde{\mathbf{f}}(\cdot, n)$ for $|n| \leq n_{\max}$ is started with:

```
1 >>> H, W = Qf.shape
2 >>> ds = (smax - smin)/(W - 1)
3 >>> alpha = 4*ds
4 >>> qinv = PolarBrokenRayInversion(Qf, phi, smin, smax, alpha, nmax)
5 >>> jm = parallel.JobManager()
6 >>> jm.jobqueue.put(qinv)
```

At the SAGE prompt, one may choose to render the reconstruction on a polar grid as it is running:

```
1 >>> reconstructed = qinv.reconstruct()
```

Since reconstruction done is on a polar grid, one may use `polar-to-cartesian.py` to resample the reconstruction to a Cartesian grid.

It should be observed that reconstruction of $\tilde{f}(r, n)$ becomes unstable for $|n|$ large as r approaches $r_{\min}$. Filtering in the form of smoothing in the angular direction will help mitigate this instability. This filtering is implemented in the `polar-to-cartesian.py` script.

B.3 Results

We present reconstructions of images are from simulated data. While the reconstructions are done on polar grids, the image shown here are reprojected onto Cartesian grids.

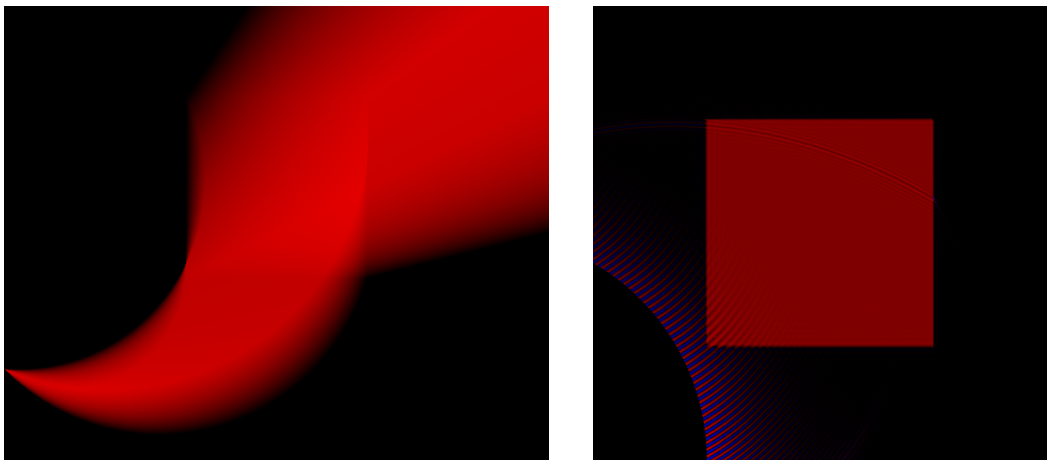

FIGURE B.1: Recovery of the characteristic function of a square from its Polar Broken Ray transform, with scattering angle $\phi = \frac{\pi}{3}$.

Figure B.1 shows the reconstruction of the characteristic function of the square $[2, 4] \times [1, 3]$, with the scattering angle $\phi = \frac{\pi}{3}$. Notice the error increases as r approaches $r_{\min}$.

Figure B.2 shows the reconstruction of the characteristic function of the disc of radius 1, centered at $(3, 2)$, with the scattering angle $\phi = \frac{\pi}{3}$. In both reconstructions, the circle $r = r_{\min}$ shown in the reconstruction as an abrupt termination of error.

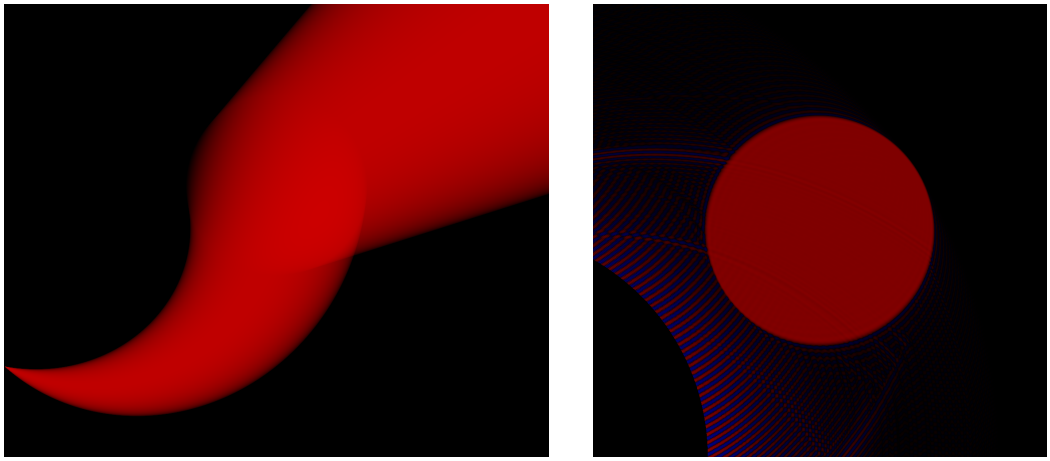

FIGURE B.2: Recovery of the characteristic function of a disc from its Polar Broken Ray transform, with scattering angle $\phi = \frac{\pi}{3}$.

The next three reconstructions, Figures B.3, B.4, and B.5, show reconstructions of a checker board residing in the square $[2, 4] \times [1, 3]$, but with differing scattering angles, $\phi = \frac{\pi}{6}$, $\frac{\pi}{4}$, and $\frac{\pi}{3}$. Interestingly, smaller values of ϕ appear to yield better reconstructions.

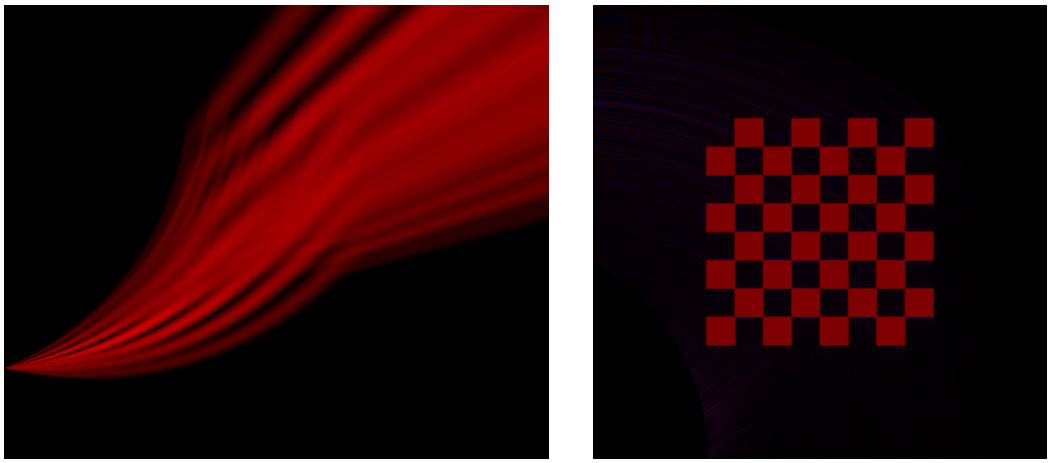

FIGURE B.3: Recovery of the checker board from its Polar Broken Ray transform, with scattering angle $\theta = \frac{\pi}{6}$.

Whereas in the case of the Florescu, et. al. Broken Ray transform, inversions can be made equivalent to inversion from a scattering angle of $\theta = \frac{\pi}{2}$, our injectivity result for the Polar Broken Ray transform did not include the case $\phi = \frac{\pi}{2}$, let alone discover that inversions from arbitary scattering angles can be made equivalent to the case $\phi = \frac{\pi}{2}$. Indeed, an

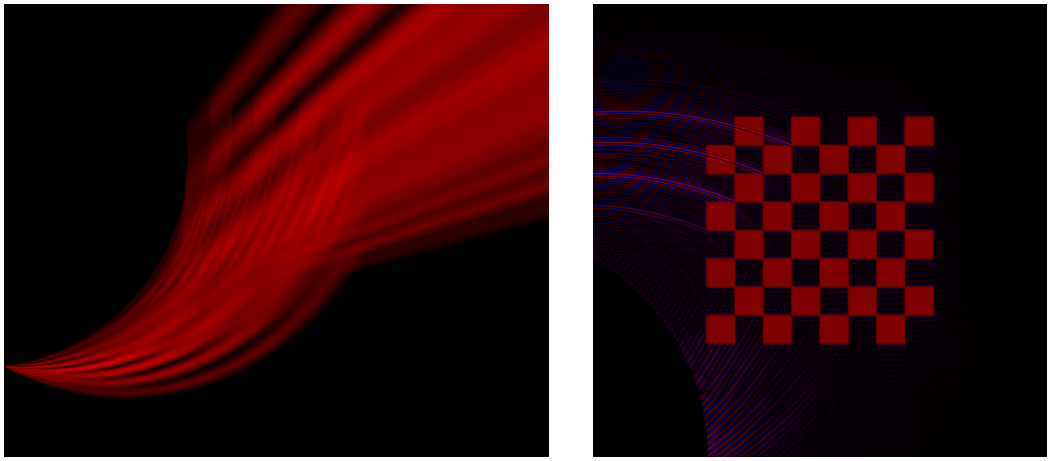

FIGURE B.4: Recovery of the checker board from its Polar Broken Ray transform, with scattering angle $\phi = \frac{\pi}{4}$.

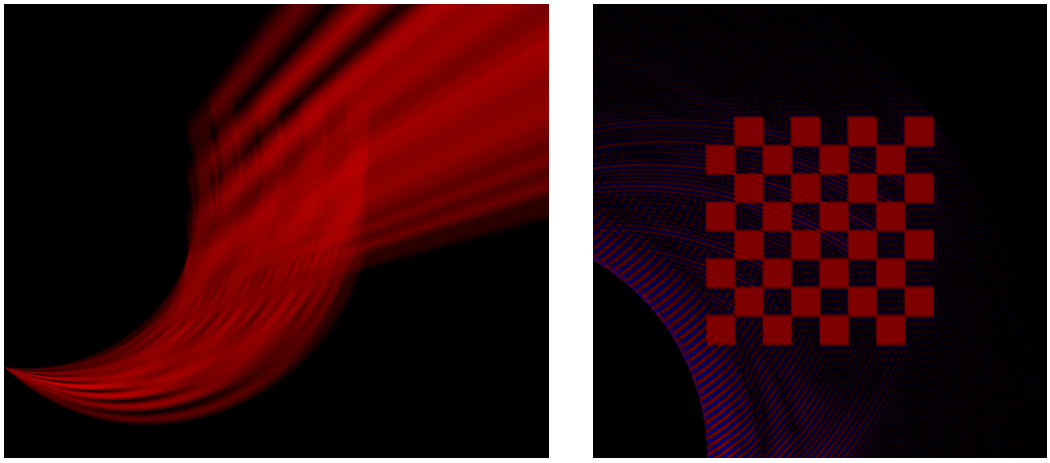

FIGURE B.5: Recovery of the checker board from its Polar Broken Ray transform, with scattering angle $\phi = \frac{\pi}{3}$.

avenue of future work with the Polar Broken Ray transform may include discovering a dependence of stability of reconstruction on the the scattering angle that is not seen with the Florescu, et. al. Broken Ray transform.