A Multiple-Patch Phase Space Method for Computing Trajectories on Manifolds with Applications to Wave Propagation Problems

Mohammad Motamed, Olof Runborg
Department of Numerical Analysis and Computer Science,
Royal Institute of Technology (KTH),
10044 Stockholm, Sweden
E-mail: mohamad@nada.kth.se, olofr@nada.kth.se

April 26, 2007

Abstract. We present a multiple-patch phase space method for computing trajectories on two-dimensional manifolds possibly embedded in a higher-dimensional space. The dynamics of trajectories are given by systems of ordinary differential equations (ODEs). We split the manifold into multiple patches where each patch has a well-defined regular parameterization. The ODEs are formulated as *escape* equations, which are hyperbolic partial differential equations (PDEs) in a three-dimensional phase space. The escape equations are solved in each patch, individually. The solutions of individual patches are then connected using suitable inter-patch boundary conditions. Properties for particular families of trajectories are obtained through a fast post-processing.

We apply the method to two different problems: the creeping ray contribution to mono-static radar cross section computations and the multivalued travel-time of seismic waves in multi-layered media. We present numerical examples to illustrate the accuracy and efficiency of the method.

Keywords. ODEs on a manifold; Phase space method; Escape equations; High frequency wave propagation; Geodesics; Creeping rays; Seismic waves; Travel-time

1 Introduction

We want to compute trajectories on two-dimensional compact manifolds possibly embedded in a higher-dimensional space. The dynamics of the trajectories we consider are given by systems of ODEs in a phase space. In many problems, we need to compute a large number of trajectories. In other words, the dynamical systems of ODEs need to be integrated for many different initial conditions. Examples include geodesics computation in computational geometry [11], robotics [2] and the theory of general relativity.

Our motivation for this comes from high frequency wave propagation problems. We consider the problem of scattering of a time-harmonic incident field by a bounded scatterer D. We split the total field into an incident and a scattered field. The scattered field in the region outside D is given by the Helmholtz equation,

$$\Delta W + n(\mathbf{x})^2\omega^2 W = 0, \qquad \mathbf{x} \in \mathbb{R}^3 \setminus \bar{D}, \tag{1}$$

where $n(\mathbf{x})$ is the index of refraction, and ω is the angular frequency. We can impose either a Dirichlet, Neumann or Robin boundary condition on the boundary of the scatterer ∂D and the Sommerfeld radiation condition at infinity.

The computational cost of direct numerical simulations of (1) grows algebraically with the frequency. Therefore, at high frequencies, numerical methods based on approximations of (1) are needed.

Geometrical optics (GO), for example, considers simple waves,

$$W(\mathbf{x}) \approx a(\mathbf{x})e^{i\omega\phi(\mathbf{x})}, \qquad \mathbf{x} \in \mathbb{R}^3, \tag{2}$$

when $\omega \to \infty$. The amplitude $a(\mathbf{x})$ and the phase function $\phi(\mathbf{x})$ depend only mildly on ω, and the computational cost will then be independent of ω. GO can be formulated either as PDEs for ϕ and a, known as *eikonal* and *transport* equation, respectively, or as a system of ordinary differential equations (ODEs).

Geometrical theory of diffraction (GTD), [18] is a correction to the GO approximations by adding diffraction effects. One type of diffracted rays are *creeping rays* which are generated at the *shadow line* of the scatterer and propagate along geodesics on the surface, continuously shedding diffracted rays in their tangential direction. A wave field, associated to a creeping ray, is generated on the surface

$$W_s(\mathbf{u}) = a(\mathbf{u})e^{i\omega\phi(\mathbf{u})}, \tag{3}$$

where $\phi(\mathbf{u})$ and $a(\mathbf{u})$ are surface phase and amplitude and $\mathbf{u} \in \mathbb{R}^2$ is a parameterization of the surface. The creeping rays are related to (3) in the same way as the standard GO rays are related to (2). Similar to GO rays, creeping rays can also be formulated either as PDEs or as a system of ODEs.

There are two different approaches to compute the standard GO and creeping rays and the associated wave fields in (2) and (3); Lagrangian and Eulerian methods.

Lagrangian methods are based on ODEs. The simplest Lagrangian method is standard ray tracing [6, 24, 13, 29] which gives the phase and amplitude solution along a ray. Interpolation must then be applied to obtain the solution everywhere. But, in regions where rays cross or diverge this can be rather difficult. The interpolation can be simplified by using wave front methods [38, 10]. In these methods, instead of individual rays, an interface representing a wave front is evolved.

Eulerian methods, on the other hand, are based on PDEs. The PDEs are discretized on fixed computational grids to control accuracy everywhere, and there is no problem with interpolation. The simplest Eulerian methods solves the eikonal and transport equations [37, 36, 8, 20]. However, these equations only give the correct solution when it is a single wave. In the case of crossing waves, more elaborate schemes have been devised based on a third formulation of geometrical optics as a kinetic equation set in phase space. A survey of this research effort, in the free space GO case, is given in [7, 31, 25]. In the surface ray case, see [26, 39] for some recent works.

Fomel and Sethian [9] presented a fast phase space method for computing solutions of static Hamilton-Jacobi equations in phase space. Their method is based on *escape* equations which are time-independent PDEs in a three-dimensional phase space. The PDE solutions, computed by a fast marching method, give the information for all trajectories from all possible starting configurations.

Recently, the authors extended the fast phase space method [26] to efficiently computing all possible creeping rays on a hypersurface. The escape solutions contains information for all incident angles. The phase and amplitude of the field are then extracted by a fast post-processing. This method is computationally attractive when the solution is sought for many different sources but with the same index of refraction, for example for computing the mono-static radar cross section (RCS). The computational cost of solving the PDEs is less than tracing all rays individually. If the surface is discretized by N^2 points the complexity is $\mathcal{O}(N^3 \log N)$, which is close to optimal. In the mono-static RCS case, direct ray tracing would cost $\mathcal{O}(N^4)$ if a comparable number of incidence angles (N^2) and rays per angle (N) are considered.

However, it is only applicable for the scatterer surfaces with simple geometries. It assumes that the surface is represented by a single parameterization, and therefore surfaces with coordinate singularities cannot be treated, and the singularity has to be excised. Most scatterer surfaces with complicated geometries, for example, cannot be represented by a single non-singular explicit parameterization. This problem can be resolved by splitting the scatterer surface into several simpler surfaces with explicit parameterizations. These multiple patches collectively cover the scatterer surface in a non-singular manner. Moreover, one can get other benefits by this way:

1. Smaller gradients in the solution by refining the patches with higher varying velocity coefficients.
2. Possibility to parallelize, since the patches can be handled independently.
3. Less internal memory needed.
4. Using the possible symmetry of the scatterer (for example for an ellipsoid).

In this paper, we consider a two-dimensional compact manifold M embedded in $\mathbb{R}^d$ and compute trajectories on the manifold. We first consider the case when the manifold is represented by a single regular parameterization and modify the fast phase space method [9, 26] to a more general class of problems. Second, we consider the case when the manifold is represented by an atlas of charts and modify the single-patch phase space method to this case. In both cases, dynamics of trajectories are given by systems of first-order ODEs.

Multiple-patch (or multi-block) finite difference schemes have long been used in computational science. They are a sub-class of domain decomposition methods for solving PDEs by iteratively solving sub-problems on smaller sub-domains [5]. However, the scheme presented here is not based on iterations. Another domain decomposition method related to the multiple-patch algorithm is the slowness matching Eulerian method [34], where local single-valued solutions of the eikonal equations are patched together by slowness matching to obtain a global, multi-valued traveltime field.

In Section 2, we give the governing equations describing the dynamics of trajectories on two classes of compact manifolds: the manifolds which can be represented by a single regular parameterization and the manifolds which are described by an atlas of charts. The construction of the single and multiple-patch schemes are described in Section 3 and 4, respectively. In Section 5 and 6, we present applications in computing creeping rays and seismic waves, together with sample numerical results from a prototype implementation of the scheme.

2 Governing Equations

Consider a two-dimensional compact manifold M embedded in $\mathbb{R}^d$. We want to compute trajectories on the manifold. Since we are interested in applications to wave propagation problems, it is natural to consider the trajectories as rays, and we will use this terminology henceforth.

We consider two cases: when the manifold is represented by a single regular parameterization, and when the manifold is represented by an atlas of charts. In both cases, dynamics of rays are given by systems of three first-order ODEs describing the rate of change of the rays' location and direction along the ray trajectories.

2.1 Single-Patch Manifolds

First, assume that the manifold can be represented by a regular parameterization $\mathbf{x} = \bar{X}(\mathbf{u})$, where $\mathbf{x} \in M$, and the parameters $\mathbf{u} = (u, v)$ belong to a set $\Omega \subset \mathbb{R}^2$. Note that if M is a hypersurface or a plane embedded in $\mathbb{R}^3$, then $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$, and if M is a plane in $\mathbb{R}^2$, then $\mathbf{x} = (x, y) \in \mathbb{R}^2$.

We introduce the phase space $\mathbb{P} = \mathbb{R}^2 \times \mathbb{S}$, where $\mathbb{S} = [0, 2\,\pi]$, and consider the triplet $\gamma = (u, v, \theta)$ as a point in this space. Let the rays be given by a system of three ODEs

$$\dot{\gamma} = \mathbf{g}(\gamma), \tag{4}$$

where the dot denotes differentiation with respect to the parameter τ being the arc length along the rays, and $\mathbf{g} = (g_1, g_2, g_3)$ is a given three-vector function which is periodic in $\theta \in \mathbb{S}$. The ray trajectories on M are then confined to a subdomain $\Omega_p = \Omega \times \mathbb{S} \subset \mathbb{P}$ in phase space. Note that the parameter values $\mathbf{u} = (u, v)$ represent the rays' location $\bar{X}(\mathbf{u})$ on M, and the angle θ represents the direction of the rays.

Remark 1. *A generic Hamiltonian system with Hamiltonian $H(\mathbf{u}, \mathbf{p})$ in four-dimensional space $\Omega \times \mathbb{R}^2$, with $\mathbf{p} \in \mathbb{R}^2$, can typically be reduced to the form (4). Here, θ can, for instance, be an angle representing the direction of vector $\mathbf{p}$. For example, if $H = |\mathbf{p}|^2 + V(\mathbf{u}) \equiv C$, one can reduce it by setting*

$$\mathbf{p} = (C - V(\mathbf{u}))^{1/2} \, (\cos\theta \;\; \sin\theta)^\top,$$

where C is determined by initial data. See also Section 5 and Section 6 for more examples.

Moreover, let any information transporting along the rays, represented by a (possibly vector-valued) function $\beta(\tau)$, be given by a more general system of the form

$$\dot{\beta} = \alpha(\gamma, \beta), \tag{5}$$

where $\alpha(\gamma, \beta)$ is some given function. For example, when β is the length of the ray, we have $\alpha \equiv 1$.

2.2 Multiple-Patch Manifolds

There are two main classes of problems for which representing the manifold by a single parameterization is not applicable: the manifolds which cannot be described by a single regular parameterization due to singularities, e.g. an airplane surface, and the manifolds with different (discontinuous) material properties, e.g. earth consisting of materials with different seismic velocities. The former is of topological and geometrical nature related to the underlying manifold, and the latter is more special application oriented.

We therefore, secondly, consider the more general case when M is described by an atlas of charts (M_j, w_j), with $j = 1, \ldots, P$, where the sets M_j collectively cover M, and the mapping $w_j : M_j \to \Omega$ is bijective. In particular, we assume that Ω is the unit square and M_j are patches with parametric equations

$$\mathbf{x} = \bar{X}_j(\mathbf{u}) : [0, 1]^2 \to M_j \subset \mathbb{R}^d,$$

and the mappings are $w_j = \bar{X}_j^{-1}$. Then $M = \bigcup_j w_j^{-1}([0, 1]^2)$. Note that although the sets are closed, we still consider M as an atlas. We assume further that the patches stick together

along their sides (patch boundaries) and denote the side between two connected patches M_j and $M_{j'}$ by $S_{jj'}$. Note that it is possible to have $j = j'$, for instance when M is a torus. When $j \neq j'$, we have $S_{jj'} = M_j \cap M_{j'}$. It is also possible that a patch does not share a side with another patch, for example, if the manifold has boundary (e.g. a finite cylinder). We denote such a side by S_{0j} which belongs only to M_j. Denote the set of all sides by $\mathcal{S}$.

For each patch with the id number j, let the rays be given by a system of three equations set in Ω_p,

$$\dot{\gamma} = \mathbf{g}_j(\gamma), \tag{6}$$

where $\mathbf{g}_j = (g_1^j, g_2^j, g_3^j)$ is a given three-vector function. Note that $\mathbf{g}_j$ may be different for different j. As before, the systems (6) are natural structures for Hamiltonian systems on four-dimensional spaces $\Omega \times \mathbb{R}^2$ with Hamiltonian $H_j(\mathbf{u}, \mathbf{p})$ whose order are reduced by one.

Correspondingly, let any information transporting along the rays, represented by a (possibly vector-valued) function $\beta(\tau)$, be given by a system of the form

$$\dot{\beta} = \alpha_j(\gamma, \beta), \tag{7}$$

where $\alpha_j(\gamma, \beta)$ is a given function.

A main difference between the numerical methods for the single patch representation of the manifold and the multiple-patch case is that in the latter we need to connect the solutions of adjacent patches and impose suitable conditions at the inter-patch boundaries. In order to treat this problem, we need to introduce a global space, which is bijective with the space $\mathbb{Z}_P \times \Omega_p$, and in which the boundary conditions are defined and can easily be handled. Here $\mathbb{Z}_P = \{1, 2, \ldots, P\}$. We first note that by our assumptions above, there is a bijective mapping between $(j, \mathbf{u}) \in \mathbb{Z}_P \times \Omega$ and $\mathbf{x} \in M$, except when $\mathbf{x}$ is at patch boundaries ($\mathbf{x} \in S_{jj'}$). Now, let $T_{\mathbf{x}}M$ be the tangent plane (the set of tangent vectors) to M at point $\mathbf{x} \in M$ and $TM = \bigcup_{\mathbf{x}\in M} T_{\mathbf{x}}M$ be the tangent bundle of M. The dimension of TM is twice the dimension of M. An element of TM is a pair $\Gamma := (\mathbf{x}, \mathbf{q})$ where $\mathbf{x} \in M$ and $\mathbf{q} \in T_{\mathbf{x}}M$. We consider the unit tangent bundle UTM of M which contains all unit-normed tangent vectors ($\|\mathbf{q}\| = 1$). Note that UTM is a three-dimensional manifold embedded in $\mathbb{R}^{2d}$.

We now want to prove that the unit tangent bundle UTM is in fact the global manifold which is bijective with the space $\mathbb{Z}_P \times \Omega_p$. But, before the proof, we notice that, by construction, for each point $\Gamma = (\mathbf{x}, \mathbf{q}) \in UTM$, there is a well-defined patch id number $j = \mathcal{J}(\Gamma)$, except when $\mathbf{x}$ is on patch boundaries. We extend this function also to the patch boundaries as follows:

- if $\mathbf{x} \in S_{jj'}$ and $\mathbf{q} \nparallel S_{jj'}$, then $\mathcal{J}(\Gamma) = \lim_{\epsilon\to 0} \arg\min_j \ dist(\mathbf{x} + \epsilon\mathbf{q}, M_j)$, which means that $\mathcal{J}(\Gamma)$ is the id of the patch into which the ray starting at Γ enters.
- if $\mathbf{x} \in S_{jj'}$ and $\mathbf{q} \parallel S_{jj'}$, then $\mathcal{J}(\Gamma) = \max(j, j')$.

Where by $\mathbf{q} \parallel S_{jj'}$, we mean that $\mathbf{q}$ is parallel to the patch boundary in an interval around $\mathbf{x} \in S_{jj'}$. Therefore in this case, Γ belongs to both UTM_j and $UTM_{j'}$, and we can choose either of j' and j' as the value of the function $\mathcal{J}(\Gamma)$. In order to have a well-defined function, we choose the larger one. Moreover, if $\mathbf{x}$ is at a corner sharing several patches $j, j', j'', \ldots,$ and $\mathbf{q}$ is parallel to $S_{jj'}$, we again choose $\mathcal{J}(\Gamma) = \max(j, j')$.

We now prove the following Lemma.

Lemma 1. *Suppose the Jacobian $J_j = D_{\mathbf{u}}\bar{X}_j \in \mathbb{R}^{d\times 2}$ has full rank for all $(j,\mathbf{u}) \in \mathbb{Z}_P \times \Omega$. For each j there is then a bijective mapping $W_j : UTM_j \to \Omega_p$ given by $W_j(\Gamma) = \gamma$, where*

$$\Gamma = (\mathbf{x}, \mathbf{q}), \quad \mathbf{q} = \frac{J_j(w_j(\mathbf{x}))\hat{s}(\theta)}{|J_j(w_j(\mathbf{x}))\hat{s}(\theta)|}, \quad \hat{s}(\theta) = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}, \quad \gamma = (w_j(\mathbf{x}), \theta). \tag{8}$$

Moreover, there is a bijective mapping between $(j,\gamma) \in \mathbb{Z}_P \times \Omega_p$ and $\Gamma = (\mathbf{x},\mathbf{q}) \in UTM$.

Proof. First assume that $\mathbf{x} \in M_j$ and $\mathbf{q} \in UT_{\mathbf{x}}M_j$. Since the mapping $w_j = \bar{X}_j^{-1}$ is bijective, there is $\mathbf{u}$ such that $\bar{X}_j(\mathbf{u}) = \mathbf{x}$, given by $\mathbf{u} = w_j(\mathbf{x})$. Moreover, since the Jacobian $J_j(\mathbf{u})$ has full rank, its columns span the tangent plane at $\mathbf{x}$, and since $\mathbf{q}$ belongs to this plane, there exists a solution θ to

$$\frac{J_j(\mathbf{u})\hat{s}(\theta)}{|J_j(\mathbf{u})\hat{s}(\theta)|} = \mathbf{q}, \qquad \hat{s}(\theta) = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}.$$

The second statement follows since $\mathcal{J}(\Gamma)$ is well-defined for all $\Gamma \in UTM$. This proves the lemma. □

Note that the atlas of charts (UTM_j, W_j) describe the space $UTM = \bigcup_j W_j^{-1}(\Omega_p)$. Figure 1 shows a schematic representation of the two-dimensional manifold M, the three-dimensional space UTM and the corresponding bijective mappings to the parameter space Ω and phase space Ω_p.

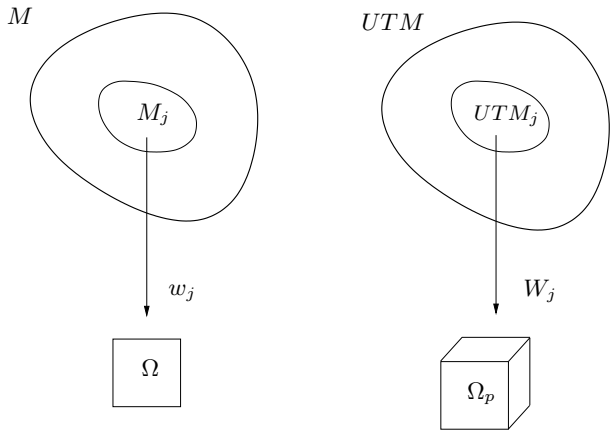


Figure 1: A schematic representation of the two-dimensional manifold M embedded in $\mathbb{R}^d$ and the three-dimensional space UTM embedded in $\mathbb{R}^{2d}$. The bijective mappings w_j and W_j map a chart j of these manifolds to the two-dimensional parameter space Ω and the three-dimensional phase space Ω_p, respectively.

2.2.1 Boundary Conditions

We may have different boundary conditions at the patch boundaries. In some problems, the rays are continuous at the patch boundaries. Such problems include geodesics and creeping rays computations on a hypersurafce with constant index of refraction. In these problems, the boundary conditions are determined easily by the continuity of rays. In some problems, the rays may not be continuous at the patch boundaries. For example, seismic propagation in a multi-layered media with different seismic velocities is such a problem, in which the boundary conditions are determined by Snell's law of refraction or the law of reflection.

As was mentioned before, the inter-patch boundary conditions are given in physical space in terms of $\Gamma \in UTM$, rather than in terms of $\gamma \in \Omega_p$. Let $\Gamma = (\mathbf{x}, \mathbf{q})$, where $\mathbf{x} \in S_{jj'}$ and $j' = \mathcal{J}(\Gamma) \neq j$, which means that the ray arrives at the side $S_{jj'}$ from patch M_j. The inter-patch boundary condition at $S_{jj'}$ is given by,

$$\tilde{\Gamma} = \mathcal{L}_{jj'}(\Gamma),$$

where $\mathcal{L}_{jj'}$ is some known function, and $\tilde{\Gamma} = (\tilde{\mathbf{x}}, \tilde{\mathbf{q}}) \in UTM_{\mathcal{J}(\tilde{\Gamma})}$. For example, depending on the ray arriving at the side $S_{jj'}$ from patch M_j, we may have the following boundary conditions:

- if the ray is continuous, then $\mathcal{L}_{jj'}$ is the identity function

$$\tilde{\mathbf{x}} = \mathbf{x}, \quad \tilde{\mathbf{q}} = \mathbf{q}.$$

- if the ray is refracted, then

$$\tilde{\mathbf{x}} = \mathbf{x}, \quad \tilde{\mathbf{q}} = \tilde{\mathcal{S}}(\mathbf{x}, \mathbf{q}).$$

- if the ray is reflected, then

$$\tilde{\mathbf{x}} = \mathbf{x}, \quad \tilde{\mathbf{q}} = \tilde{\mathcal{R}}(\mathbf{x}, \mathbf{q}).$$

Here, the functions $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{R}}$ are determined by Snell's law of refraction and the law of reflection, respectively. See Section 6.2 for more details.

In the next two sections, we present a patch-based phase space method for computing ray trajectories on manifolds. First, we consider the case when the manifold is represented by a single parameterization and construct a single-patch phase space method based on writing the systems (4-5) in a Eulerian framework. Next, we consider a wider class of manifolds which are represented by multiple parameterizations and introduce a multiple-patch phase space method based on solving the Eulerian version of systems (6-7) in each patch and connecting the solutions of individual patches using suitable inter-patch boundary conditions. In both methods, properties for particular ray families are obtained through a fast post-processing.

3 Single-Patch Phase Space Scheme

We consider the case when the two-dimensional manifold M embedded in $\mathbb{R}^d$ is represented by a single regular parameterization. The objective is to compute the ray trajectories together with the information transported along them on M. First, the system of ODEs (4) and (5), describing rays and other information, are formulated as time-independent Eulerian PDEs in phase space. These equations are then solved numerically on a fixed computational grid. The solution to the PDEs is post-processed to extract information for a particular family of rays.

3.1 Mathematical Formulation

We consider a ray $\bar{\gamma}(\tau)$ satisfying (4), starting at $\bar{\gamma}(0) = \gamma = (u, v, \theta) \in \Omega_p$ and ending at the boundary $\partial\Omega_p = \partial\Omega \times \mathbb{S}$. We call this end point $(U, V, \Theta) \in \partial\Omega_p$ the *escape point* of the ray. See Figure 2. We then define three types of unknown *escape functions* for this ray, as follows:

- $F : \mathbb{P} \to \mathbb{P}$, $F(\gamma) = (U, V, \Theta)$ is the escape point.
- $\Phi : \mathbb{P} \to \mathbb{R}$ is the length of the ray. We also refer to this as the travel-time of the ray.
- $B : \mathbb{P} \to \mathbb{R}$ is a function representing a relation between the β-values at the escape and starting points, where β satisfies (5).

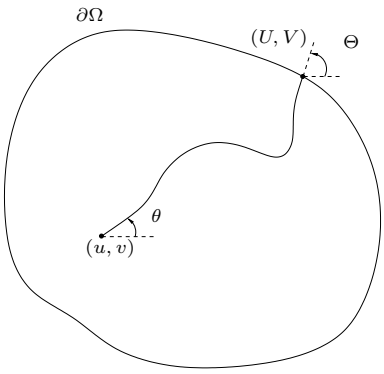


Figure 2: A ray trajectory in the parameter space, starting at $\gamma = (u, v, \theta) \in \Omega_p$ and ending at the escape point $F(\gamma) = (U, V, \Theta) \in \partial\Omega_p$.

Each escape function $f(\gamma)$ of the above types satisfies an ODE,

$$\frac{d}{d\tau} f(\gamma(\tau)) = h\big(\gamma(\tau), f(\gamma(\tau))\big), \tag{9}$$

where the forcing term h is 0, 1 and $\alpha(\gamma, f)$ for $f = F$, $f = \Phi$ and $f = B$, respectively.

Using the chain rule, the *escape* PDE for each escape function $f(\gamma)$ reads

$$g_1(\gamma)\, f_u + g_2(\gamma)\, f_v + g_3(\gamma)\, f_\theta = h(\gamma, f), \qquad \gamma \in \Omega_p, \tag{10}$$

with the boundary condition at inflow points of $\partial\Omega_p$,

$$f(\gamma) = b, \qquad \gamma \in \partial\Omega_p^{\text{inflow}}, \qquad \partial\Omega_p^{\text{inflow}} = \left\{\gamma \in \partial\Omega_p \,|\, \hat{n}(\gamma)^\top \mathbf{g}(\gamma) < 0\right\},$$

with $\hat{n}$ being the outward normal vector in the phase space.

Note that for the first two types of escape functions $f = F$ and $f = \Phi$, the boundary value b is γ and 0, respectively. For the third type $f = B$, if for instance B is the difference or ratio between β-values at the escape and starting points, the boundary value are $b = 0$ or $b = 1$, respectively.

The escape equation (10) is a linear hyperbolic equation, and the variable velocity coefficients $\mathbf{g} = (g_1, g_2, g_3)$ are known and determine the characteristic direction at every point $\gamma \in \Omega_p$.

One important property of the solutions to the escape PDEs is that they are in general discontinuous due to discontinuous boundary conditions. This happens, for example, when a characteristic touches a boundary tangentially, such that at some points on the plane the characteristic is in-going, and suddenly it becomes out-going.

3.2 Numerical Solution of the escape PDEs

We now want to solve (10) numerically. We discretize the phase space domain $\Omega_p = \Omega \times \mathbb{S}$ uniformly, setting $u_i = i\Delta u$, $v_j = j\Delta v$ and $\theta_k = k\Delta\theta$, with the step sizes $\Delta u = \Delta v = \frac{1}{N}$ and $\Delta\theta = \frac{2\pi}{N}$, assuming Ω is the unit square. Moreover, let f_{ijk} approximate an escape function $f(u_i, v_j, \theta_k)$.

In addition to the boundary condition at inflow points, since the function f is periodic in θ, we use periodic boundary conditions,

$$f(u, v, 0) = f(u, v, 2\,\pi),$$

as numerical boundary conditions.

There are different methods for solving the escape equation (10). One way is to discretize the PDEs in the phase space using a finite difference, finite volume or finite element approximation and arrive at a system of linear equations $A\bar{f} = \bar{b}$, where A is a $N^3 \times N^3$ matrix with a sparse structure and $\bar{b} \in \mathbb{R}^{N^3}$ represents the boundary conditions. This system can then be solved iteratively, and one can speed up the computations using suitable preconditioners [12, 4]. However, in the case that characteristics change direction many times in the phase space domain, it is difficult to find good preconditioners.

Another way to solve the escape equations is to write them as

$$f_t + g_1\, f_u + g_2\, f_v + g_3\, f_\theta = h,$$

and solve these time-dependent equations until the steady state $f_t = 0$. This method can be seen as an iterative method. Finding a fast algorithm which is not much restricted by the CFL condition is analogous to finding a good preconditioner in the iterative method.

Yet, another way to solve the equation (10) is to compute the approximate solution f_{ijk} using a ray tracing method, which traces back along the characteristic to the initial boundary from each grid point (i, j, k). The main drawback with this method is that it will be expensive, because one needs to trace back all N^3 points in the domain all the way to the boundary.

Instead, we use a Fast Marching algorithm, given by Fomel and Sethian [9]. A similar method in two-dimensional space was also proposed in [16]. The basic idea of the algorithm is to march the solution outwards from the boundary and use the characteristic directions to update grid values. Note that in the algorithm, we always also compute Φ_{ijk} besides f_{ijk}.

First, the grid points are divided into three classes:

- *Accepted*: the correct values of f_{ijk} and Φ_{ijk} have been computed.
- *Considered*: adjacent to *Accepted* for which f_{ijk} and Φ_{ijk} have already been computed, but may be corrected by a later computation.
- *Far*: the correct values of f_{ijk} and Φ_{ijk} are not known.

The major steps of the algorithm are then as follows:

0. Start with all nodes $(u_i, v_j, \theta_k) \in \Omega_p$ in *Far*, and assign Φ_{ijk} at these nodes a large value. This large value needs to be greater than the length (travel-time) of every possible ray in the computational domain. Put the boundary nodes $(u_i, v_j, \theta_k) \in \partial\Omega_p^{\text{inflow}}$ in *Accepted*, and assign f_{ijk} and Φ_{ijk} at these nodes the correct boundary values. Put all nodes

adjacent to *Accepted*, for which the characteristic[1] at that node points back to the boundary, in *Considered*. Each *Considered* node is then given a value by using a local cell characteristic method.

1. Take the *Considered* node with the smallest arrival time Φ_{ijk} as *Accepted*.
2. Find the octant toward which the characteristic going through that node points.
3. For each neighboring grid point in the octant which is not *Accepted* use the local cell characteristic method to (possibly) compute new values for f_{ijk} and Φ_{ijk}. In the case we can compute new values for a *Far* node, put it in *Considered*.
4. Loop to step 1 until all points are *Accepted*.

Since in [9] the local cell characteristic method, used in steps 0 and 3 of the algorithm, is not discussed, we will here describe a version of first and second order local cell-based ray tracing methods using a local linear and parabolic ray tracing and the Taylor expansion of the trajectory near the starting point.

Consider a grid cell in Ω_p, and assume we want to compute the value of f_{ijk} at a corner of this cell, knowing the correct values of f at some neighboring grid points. The output of the local ray tracing would be either a new value for f_{ijk} or no new value, depending on whether the neighboring points, to which the characteristic points back, are *Accepted* or not. See Figure 3.

Let τ be the arc length parameterization along the characteristic $\gamma(\tau)$. We start at $\gamma(0) = (u_i, v_j, \theta_k)$, where we want to compute a possibly new value, and trace backwards along the characteristic to intersect a cell face at $\gamma(\tau^*)$, $\tau^* < 0$. We Taylor expand f near the starting point,

$$f(\gamma(\tau^*)) = f(\gamma(0)) + \tau^* \frac{d}{d\tau} f(\gamma(0)) + \frac{\tau^{*2}}{2} \frac{d^2}{d\tau^2} f(\gamma(0)) + \mathcal{O}(\tau^{*3}), \tag{11}$$

with local truncation error $\mathcal{O}(\tau^{*3}) \approx \mathcal{O}(\Delta u^3)$. Note that $\frac{d}{d\tau} f(\gamma(0))$ and $\frac{d^2}{d\tau^2} f(\gamma(0))$ in (11) are given by:

$$\begin{aligned}
\frac{d}{d\tau} f(\gamma(0)) &= h(\gamma(0), f(\gamma(0))), \\
\frac{d^2}{d\tau^2} f(\gamma(0)) &= \frac{d}{d\tau} h\big(\gamma(0), f(\gamma(0))\big) \\
&= \mathbf{g}(\gamma(0)) \cdot \nabla_\gamma h\big(\gamma(0), f(\gamma(0))\big) + h_f\big(\gamma(0), f(\gamma(0))\big)\, h\big(\gamma(0), f(\gamma(0))\big).
\end{aligned}$$

Therefore, to find $f(\gamma(0))$, with accuracy of $\mathcal{O}(\tau^{*3})$, we need to know τ^* and $f(\gamma(\tau^*))$. Note that for $f = F$ and $f = \Phi$, since $\frac{d}{d\tau} F(\gamma(\tau)) = 0$ and $\frac{d}{d\tau} \Phi(\gamma(\tau)) = 1$, the expansion (11) reduces to

$$F(\gamma(\tau^*)) = F(\gamma(0)), \tag{12}$$

$$\Phi(\gamma(\tau^*)) = \Phi(\gamma(0)) + \tau^*. \tag{13}$$

[1] We approximate the characteristic by a piecewise linear curve for a first order method and piecewise parabolic for a second order method.

3.2.1 First Order Method

We assume that characteristics are linear in each cell. Therefore, we can write

$$\gamma(\tau) \approx \sigma_1 + \sigma_2\,\tau, \quad \sigma_1 = \gamma(0), \quad \sigma_2 = \dot{\gamma}(0) = \mathbf{g}(\gamma(0)).$$

Note that σ_1 and σ_2 are known. There are six possible planes, $u = u_{i\pm1}$, $v = v_{j\pm1}$ and $\theta = \theta_{k\pm1}$, which this line can intersect. We, therefore, get six crossing points $\tau_1, \ldots, \tau_6$, which are solutions of six linear equations. It is then clear that $\tau^* = \max_{\tau_j<0} \tau_j$. Knowing the crossing face and the crossing point $\gamma(\tau^*)$, we continue as follows:

a. If all four points of the cell face are *Accepted*, use these points to interpolate a value of $f(\gamma(\tau^*))$. Then use the first two terms of the Taylor expansion (11) to compute a new value for $f_{ijk} \approx f(\gamma(0))$. Note that we need to solve a (possibly) nonlinear algebraic equation, when h depends on f,

$$f(\gamma(0)) = f(\gamma(\tau^*)) - \tau^*\, h(\gamma(0), f(\gamma(0))).$$

Put this node in *Considered*. Since the method is first order, a two dimensional bilinear interpolation is used. See Figure 3.

b. If no points on the cell face are *Accepted*, do not update the value.

c. Else, continue tracing along the characteristic until either (a) or (b) occurs. Note that each time the characteristic enters a new cell, the new starting point needs to be updated.

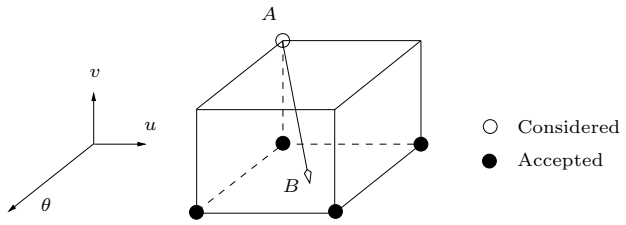


Figure 3: A grid cell in Ω_p. Point A is updated by tracing the characteristic back to point B and interpolating from the accepted values. Here, points A and B correspond to $\gamma(0)$ and $\gamma(\tau^*)$, respectively.

3.2.2 Second Order Method

We assume that characteristics are parabolic in each cell and write

$$\gamma(\tau) \approx \sigma_1 + \sigma_2\,\tau + \sigma_3\,\tau^2, \quad \sigma_1 = \gamma(0), \quad \sigma_2 = \dot{\gamma}(0), \quad \sigma_3 = \frac{1}{2}\ddot{\gamma}(0) = \frac{1}{2} D_\gamma \dot{\gamma}(0)\,\dot{\gamma}(0).$$

Note that σ_1, σ_2 and σ_3 are known. In this case, there are nine possible cell faces which can intersect this parabola; $u = u_i$, $v = v_j$, $\theta = \theta_k$ and the six faces in the linear case. By intersecting the parabola with the faces, we get nine crossing points $\tau_1, \ldots, \tau_9$, which are solutions of simple quadratic equations. We then get $\tau^* = \max_{\tau_j<0} \tau_j$ and continue in the following way:

a. Pick the crossing face and eight faces around it in the same plane, sharing sixteen grid points in total. If all sixteen points are *Accepted*, use these points to interpolate a value of $f(\gamma(\tau^*))$. Then use the first three terms of the Taylor expansion (11) to compute a new value for $f_{ijk} \approx f(\gamma(0))$. Note that, again, we need to solve a (possibly) nonlinear algebraic equation, when h depends on f,

$$f(\gamma(0)) = f(\gamma(\tau^*)) - \tau^* h(\gamma(0), f(\gamma(0))) - \frac{\tau^{*2}}{2} \frac{d}{d\tau} h\big(\gamma(0), f(\gamma(0))\big).$$

Put this node in *Considered*. Because the solution can be discontinuous, we use a version of two dimensional essentially non-oscillatory (ENO) interpolation based on Newton divided differences and the Newton formulation of the interpolation polynomial. Among four points in each dimension, we pick up either the left three or the right three points which have a smaller divided difference and use a second order polynomial. See [33].

b. If no points on the cell face are *Accepted*, do not update the value.

c. Else, continue tracing along the characteristic until either (a) or (b) occurs. Note that each time the characteristic enters a new cell, the new starting point needs to be updated.

The algorithm is a one-pass algorithm and is of complexity $\mathcal{O}(N^3 \log N)$. Note that we use heap sort algorithm for extracting the smallest arrival time Φ_{ijk} of *Considered* nodes and for inserting new updated values of *Considered* nodes. There is however no proof of convergence for the method.

3.3 Post-Processing

Solutions of the escape PDEs (10) give the escape point, length and other information for rays with all possible starting points in the phase space. These solutions need to be post-processed to extract properties for a ray family.

As an example, suppose we want to compute the length of the ray between two points $\mathbf{u}_1$ and $\mathbf{u}_2$ in the parameter space Ω. We first observe that $F(\gamma_1) = F(\gamma_2)$, if and only if the points γ_1 and γ_2 lie on the same ray. We can thus find θ_1 and θ_2, as the solution to

$$F(\mathbf{u}_1, \theta_1) = F(\mathbf{u}_2, \theta_2). \tag{14}$$

The length is then given by $|\Phi(\mathbf{u}_1, \theta_1) - \Phi(\mathbf{u}_2, \theta_2)|$. Note that there may be multiple solutions to (14), giving multiple lengths. If $\mathbf{u}_2 \in \partial\Omega$, the expression simplifies to solving

$$\big(U(\mathbf{u}_1, \theta_1), V(\mathbf{u}_1, \theta_1)\big) = \mathbf{u}_2, \tag{15}$$

for θ_1 to get the length $\Phi(\mathbf{u}_1, \theta_1)$.

To solve (14), we note that since $F = (U, V, \Theta) \in \partial\Omega_p$ is a point on the phase space boundary, it can be reduced to a point (S, Θ) in $\mathbb{R}^2$, where S represents the escape location on the boundary $\partial\Omega$. For example if $\Omega = [0, 1]^2$, we can choose $S \in [0, 2\,\pi]$ along $\partial\Omega$ such that $S = 0$, $S = \pi$ and $S = 2\pi$ for $(U, V) = (0, 0)$, $(U, V) = (1, 1)$ and $(U, V) = (0, 0)$, respectively. The left and right hand sides of (14) are then curves in $\mathbb{R}^2$ parameterized by θ_1 and θ_2, and solving the algebraic equation (14) amounts to finding crossing points of these curves.

Having the discrete solutions at the points $\mathbf{u}_1$ and $\mathbf{u}_2$ for all N directions, we then need to find crossing points of two complex lines of N straight line segments as the solutions to (14). This can be done with a complexity of $\mathcal{O}(N)$; see e.g. [35]. We note that in the case that a second order method for solving the escape equations is used, the linear intersection algorithm will not affect second order accuracy of the method. In fact, the intersection algorithm is performed only to find the intersection's neighboring points. We use a higher order interpolation to compute the initial angles θ_1 and θ_2 and the escape functions corresponding to these angles. The complexity of finding the ray length between one fixed source point and all other N^2 points in Ω is then $\mathcal{O}(N^3)$, and the total complexity, including solving the escape PDEs, will therefore be $\mathcal{O}(N^3 \log N)$. This is expensive for computing this so called travel-time field for only one source point. For example by using wave front tracking or solvers based on the surface eikonal equation, the complexity is $\mathcal{O}(N^2)$. However, if the solutions are sought for many source points, the phase space method can be more efficient. See Section 5 for such an example.

4 Multiple-Patch Phase Space Scheme

We now consider the more complicated and realistic case when the manifold M cannot be represented by one regular parameterization. We let M be described by an atlas of charts or multiple patches and want to compute the ray trajectories together with the information transported along them on the manifold. First, the system of ODEs (6) and (7) in each chart (patch) are formulated as time-independent Eulerian PDEs and solved numerically on a fixed computational grid in phase space. The solutions to the PDEs in each chart are then connected using suitable inter-patch boundary conditions. Information for a particular family of rays are then extracted through a fast post-processing.

We describe the multiple-patch scheme and the key design choices in such a scheme, including the number and shape of patches, the treatment of inter-patch boundaries and the choice of escape boundary.

4.1 Multiple-Patch Construction

We first want to define a function $\mathbf{F}$ for the multiple patch case that corresponds to the single patch solution F described in Section 3. Let $\mathcal{R}$ be some curve in M, representing an escape boundary. We consider a ray starting at a point $\Gamma \in UTM$ and define $\mathbf{F}(\Gamma) : UTM \to UTM$ as mapping the point Γ to another point in the space UTM where the projection of the ray onto M first crosses $\mathcal{R}$ (assuming such a point exists).

If the compact manifold has a boundary (e.g. a finite cylinder), we let this be the escape boundary, similar to the case of a single-patch manifold. Hence, $\mathcal{R} = \bigcup_j S_{0j}$. However, for a compact boundaryless manifold (e.g. a sphere or a torus), there is no obvious escape boundary, as in the single patch case. In this case we will let

$$\mathcal{R} = \bigcup_{(j,j')\subset R} S_{jj'} \subset \mathcal{S} \tag{16}$$

be the escape boundary, where R is some index set, to be determined (see below).

To compute $\mathbf{F}(\Gamma)$, we first recall that, by construction, for each point $\Gamma = (\mathbf{x}, \mathbf{q}) \in UTM$, there is a well-defined patch id number $j = \mathcal{J}(\Gamma)$ and a well-defined mapping $W_j : UTM_j \to$

Ω_p.

Now, suppose $F_j(\gamma)$ are the solutions to the escape PDE (10), with $f = F$, in Ω_p corresponding to each patch with $j = 1, \ldots, P$. The function $\mathbf{F}(\Gamma)$ is then given recursively by

$$\tilde{\Gamma}_0 = \Gamma, \tag{17}$$

and while $\tilde{\mathbf{x}}_n \notin \mathcal{R}$, where $\tilde{\Gamma}_n = (\tilde{\mathbf{x}}_n, \tilde{\mathbf{q}}_n)$,

$$j = \mathcal{J}(\tilde{\Gamma}_n), \quad \Gamma_{n+1} = W_j^{-1} F_j(W_j(\tilde{\Gamma}_n)), \quad j' = \mathcal{J}(\Gamma_{n+1}), \quad \tilde{\Gamma}_{n+1} = \mathcal{L}_{jj'}(\Gamma_{n+1}), \tag{18}$$

where $\mathcal{L}_{jj'}$ is the operator representing the inter-patch boundary conditions between patches M_j and $M_{j'}$. Then $\mathbf{F}(\Gamma) = \Gamma_{n^*}$, where n^* is the smallest index for which $\mathbf{x}_{n^*} \in \mathcal{R}$.

Remark 2. *If the rays are continuous at the patch boundaries, $\mathcal{L}_{jj'}$ will be the identity function ($\tilde{\Gamma}_{n+1} = \Gamma_{n+1}$). From the above recursive formula, it is easy to see that, in order to compute the function $\mathbf{F}$ for all points in UTM it is enough to know the escape PDE solutions F_j in all patches and the patch transfer functions $\mathcal{T}_{jj'} = W_{j'} W_j^{-1}$ at all sides connecting two patches M_j and $M_{j'}$. Note that these transfer functions can be easily calculated from the mappings W_j. As an example, in Section 5, we will discuss the computation of creeping rays which are continuous at patch boundaries.*

If the rays are not continuous at the patch boundaries, each time they pass a boundary, the coordinates of Γ_{n+1} may change ($\tilde{\Gamma}_{n+1} \neq \Gamma_{n+1}$). It happens when, for example, the rays change their direction as they enter another patch with different properties. The patch transfer functions are then changed to $\mathcal{T}_{jj'} = W_{j'} \mathcal{L}_{jj'} W_j^{-1}$. Here, transfer functions are again easily calculated from the mappings W_j and the inter-patch boundary conditions. We will consider such examples in Section 6, where the rays change direction according to Snell's law of refraction and the law of reflection.

Similar to $\mathbf{F}(\Gamma)$, we can define the functions $\mathbf{\Phi}(\Gamma)$ and $\mathbf{B}(\Gamma)$ in UTM for the multiple patch case corresponding to the single patch functions Φ and B described in Section 3. Assuming $\Phi_j(\gamma)$ and $B_j(\gamma)$ are the solutions to the escape PDE (10), with $f = \Phi$ and $f = B$, respectively, in Ω_p corresponding to each patch with $j = 1, \ldots, P$, we can write

$$\mathbf{\Phi}(\Gamma) = \sum_{n=0}^{n^*-1} \Phi_j(W_j(\tilde{\Gamma}_n)),$$

with j and $\tilde{\Gamma}_n$ as in (17)-(18), and

$$\mathbf{B}(\Gamma) = \sum_{n=0}^{n^*-1} B_j(W_j(\tilde{\Gamma}_n)),$$

if B is, for example, the difference between β-values at Γ_{n^*} and Γ_0, and

$$\mathbf{B}(\Gamma) = \prod_{n=0}^{n^*-1} B_j(W_j(\tilde{\Gamma}_n)),$$

if B represents, for example, the ratio between β-values.

4.2 Post-Processing

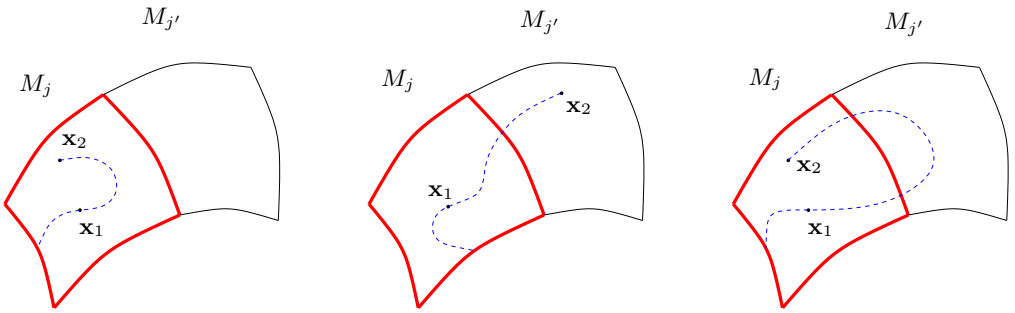
Suppose we want to compute the length of a ray connecting two points $\mathbf{x}_1 \in M_{j_1}$ and $\mathbf{x}_2 \in M_{j_2}$. In order to find this ray, if the manifold has a boundary, we let this be the escape boundary, and the post-processing is similar to the single patch case with F replaced by $\mathbf{F}$. In the case of a boundaryless manifold, we choose the boundaries of M_{j_1} as the escape boundary $\mathcal{R}$. We then find $\mathbf{F}\left(W_{j_1}^{-1}(w_{j_1}(\mathbf{x}_1), \theta_1)\right)$ for all directions $\theta_1 \in \mathbb{S}$.

We now modify the function $\mathbf{F}(\Gamma)$ by $\mathbf{F}_n(\Gamma)$, where n is the number of times which the ray starting at Γ hits the escape boundary. It is therefore obvious that $\mathbf{F}_1(\Gamma) = \mathbf{F}(\Gamma)$. In the case where the rays at the patch boundaries are continuous, we have,

$$\mathbf{F}_n(\Gamma) = \underbrace{\mathbf{F} \circ \mathbf{F} \cdots \circ \mathbf{F}}_{n \text{ times}}(\Gamma). \tag{19}$$

In general, the boundary function $\mathcal{L}_{jj'}$ must be applied in composition too. Analogously, we can define functions $\mathbf{\Phi}_n$ and $\mathbf{B}_n$.

For all directions $\theta_2 \in \mathbb{S}$ we then find $\mathbf{F}_n\left(W_{j_2}^{-1}(w_{j_2}(\mathbf{x}_2), \theta_2)\right)$. Since we do not know how many times the ray, which starts at $\mathbf{x}_2$ and passes through $\mathbf{x}_1$, hits $\mathcal{R}$, we need to find $\mathbf{F}_n$ for several values $n = 1, 2, \ldots$. See Figure 4 for three different cases where n is 1, 2 and 3.



(a) $n = 1$, $(j_1 = j_2 = j)$ (b) $n = 2$, $(j_1 = j,\ j_2 = j')$ (c) $n = 3$, $(j_1 = j_2 = j)$

Figure 4: Two neighboring patches M_j and $M_{j'}$. The ray (dashed curve) starting at $\mathbf{x}_2$ and passing through $\mathbf{x}_1$, hits the escape boundary $\mathcal{R}$ (thick curves) n times. Here, three different cases are shown where n is 1, 2 and 3.

We then find θ_1, θ_2 and n as the solutions to the algebraic equations

$$\mathbf{F}\left(W_{j_1}^{-1}(w_{j_1}(\mathbf{x}_1), \theta_1)\right) = \mathbf{F}_n\left(W_{j_2}^{-1}(w_{j_2}(\mathbf{x}_2), \theta_2)\right), \tag{20}$$

analogous to (14) in the single-patch case. There will be at most four systems of equations corresponding to four sides of patch M_{j_1}, for each value of n. The solutions to (20) can be computed by finding intersections of four sets of possibly crossing curves.

The length is then given by

$$\left|\mathbf{\Phi}\left(W_{j_1}^{-1}(\gamma_1)\right) - \mathbf{\Phi}_n\left(W_{j_2}^{-1}(\gamma_2)\right)\right|,$$

with $\gamma_1 = (w_{j_1}(\mathbf{x}_1), \theta_1)$ and $\gamma_2 = (w_{j_2}(\mathbf{x}_2), \theta_2)$.

4.3 Number and Shape of Patches and Parameterizations

One of the key design choices in such a multiple-patch scheme is the choice of patches and parameterizations. The important things are:

1. Patches should cover the physical domain with nonsingular parameterizations.
2. Parameterizations should have small coordinate distortions to make finite differencing accurate.
3. The right hand side $h(\gamma, f)$ in the escape PDEs should be well resolved by the patch discretization.

Remark 3. *Using overlapping patches, one can possibly reduce the number of patches. However, the objective in this work has not been optimizing the number of patches.*

4.4 Choosing Escape Boundary

Another key design choice is the choice of escape boundary. Two things are important about $\mathcal{R}$, and R:

1. The projection of each ray, which is of interest, onto M should cross $\mathcal{R}$ at some point. Otherwise $\mathbf{F}(\Gamma)$ is not well defined for all points. It is not obvious how to verify this rigorously. Having nonzero coefficients, $\mathbf{g}(\gamma) \neq 0$, everywhere is a necessary condition, but it is still possible to have rays that never reaches a given boundary, see e.g. [23].
2. If the compact manifold has a boundary, we can choose this as the escape boundary, similar to the single-patch manifold.

4.5 Limitations and Extra Problems

There are a couple of difficulties and problems:

1. In some cases, one cannot capture all rays of interest by only one choice of escape boundary. Different choices of escape boundary might be needed. A good implementation of the algorithm will then be the one which considers different combinations of patch boundaries as the escape boundary. Note that this is done in post-processing and does not require recomputation of the f_j solutions.
2. When a ray hits an inter-patch boundary, in order to find the escape solution at this point, we need to interpolate the discrete solutions computed on a fixed grid. The interpolation can be difficult if a ray is tangent to the inter-patch boundary. One possible way to overcome such a problem is to use overlapping patches. Another possibility is to choose another atlas of charts for the manifold.

5 Application to Creeping Ray Computations

Creeping rays are a type of diffracted rays which are generated at the *shadow line*[2] of the scatterer and propagate along geodesic paths on the scatterer surface. On a perfectly conducting convex body, they attenuate along their propagation path by tangentially shedding diffracted rays and losing energy. On a concave scatterer, they propagate on the surface and importantly, in the absence of dissipation, experience no attenuation.

The study of creeping rays is important in many high frequency problems, such as design of sophisticated and conformal antennas [19], antenna coupling problems [21], radar cross

[2]Shadow line or *horizon* is the locus of the points at which the incident rays are tangent to the scatterer surface.

section (RCS) computations [3, 19, 32, 26] and control of scattering properties of metallic structures coated with dielectric materials [28, 1, 22, 27].

In this section, we consider the application of the multiple-patch phase space method to computing creeping rays. Here, the computational domain is a scatterer surface which is a two-dimensional hypersurface embedded in $\mathbb{R}^3$. We split the surface into multiple patches represented by different parameterizations. The escape PDEs describing creeping rays are solved in each patch, individually. The creeping rays on the scatterer are then computed by connecting all individual solutions. The inter-patch boundaries are treated by the continuity of characteristics.

We first consider the case when the scatterer surface has a regular explicit parameterization and write the governing equations for computing creeping rays. We then discuss the multiple-patch scheme and give two numerical examples where the contribution of creeping rays to mono-static RCS is computed.

5.1 Governing Equations

We consider a scatterer surface with a regular explicit parameterization, represented by $\mathbf{x} = \bar{X}(\mathbf{u})$, where $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$, and the parameters $\mathbf{u} = (u, v)$ belong to a set $\Omega \subset \mathbb{R}^2$. Let the scatterer be illuminated by incident rays in a direction denoted by a normalized vector $\hat{I} = [\imath_1, \imath_2, \imath_3]$. We assume that the shadow line $\mathbf{u}_0(s)$ is represented by a curve in parameter space, with s being the arc length parameterization. The objective is to compute the geodesic paths on the scatterer surface together with the phase and amplitude of the wave field of creeping rays generated on the scatterer.

According to Keller and Lewis [17], the surface phase satisfies the *surface eikonal equation*,

$$|\tilde{\nabla}\phi| = n, \tag{21}$$

where $n(\mathbf{u})$ is the index of refraction at the surface, and $\tilde{\nabla}$ is the surface gradient, defined as

$$\tilde{\nabla}\phi := JG^{-1}\nabla\phi, \qquad G = J^\top J, \qquad J = [\bar{X}_u\, \bar{X}_v] \in \mathbb{R}^{3\times 2}.$$

We can write (21) as a Hamilton-Jacobi equation $H(\mathbf{u}, \nabla\phi) = 0$, with the Hamiltonian

$$H(\mathbf{u}, \mathbf{p}) \equiv \frac{1}{2}\,\mathbf{p}^\top\, G^{-1}(\mathbf{u})\,\mathbf{p} - \frac{n^2(\mathbf{u})}{2}. \tag{22}$$

Note that in the case n = constant, the rays associated with the surface eikonal equation (21) are geodesics, or shortest paths between two points on the surface. Henceforth, we will assume $n \equiv 1$.

We write (without derivation) the set of equations which are used in computing creeping rays and are obtained by reducing the order of the Hamiltonian system corresponding to (22) by one. For derivations see [26].

A geodesic on the surface is uniquely characterized by its location, (u, v), and direction, θ. Letting $\gamma := (u, v, \theta)$, the geodesics satisfy the system of ODEs (4) with

$$\mathbf{g}(\gamma) = \begin{pmatrix} \rho(\gamma)\,\cos\theta \\ \rho(\gamma)\,\sin\theta \\ \rho(\gamma)\,\mathcal{V}(\gamma) \end{pmatrix}. \tag{23}$$

The parameter τ is the arc length along the geodesic in the physical space, and

$$\rho = \rho(u, v, \theta) = \left|\bar{X}_u \cos\theta + \bar{X}_v \sin\theta\right|^{-1},$$

$$\begin{aligned}\mathcal{V}(\gamma) = &(\Gamma^1_{11}\cos^2\theta + 2\Gamma^1_{12}\cos\theta\sin\theta + \Gamma^1_{22}\sin^2\theta)\sin\theta - \\ &(\Gamma^2_{11}\cos^2\theta + 2\Gamma^2_{12}\cos\theta\sin\theta + \Gamma^2_{22}\sin^2\theta)\cos\theta,\end{aligned}$$

where $\Gamma^k_{ij}(\mathbf{u})$ are Christoffel symbols.

Moreover, we know that the phase ϕ is the length of the ray, given by (5) with $\beta = \phi$ and $\alpha \equiv 1$, and the amplitude a is computed by,

$$a(\tau) = a_0 \mathcal{Q}(s,\tau)^{\frac{-1}{2}} \exp\left(-\omega^{1/3}\beta(\tau)\right), \tag{24}$$

where a_0 is the amplitude at the starting point on the shadow line, $\mathcal{Q}(s,\tau)$ is the geometrical spreading at distance τ from the starting point, and $\beta(\tau)$ is a function representing the attenuation factor given by (5) with

$$\alpha(\gamma) = \frac{q_0}{\rho_g(\gamma)} \exp\left(i\frac{\pi}{6}\left(\frac{\rho_g(\gamma)}{2}\right)^{1/3}\right), \quad q_0 \approx 2.33811. \tag{25}$$

Here $\rho_g(\gamma)$ is the radius of curvature of the surface along the ray trajectory. We then let the escape function B be the difference between the β-values at the escape and starting points. All escape functions F, Φ and B satisfy equation (10), with the right hand side h being 0, 1 and α given by (25), respectively.

In order to compute the amplitude, in addition to β, we need also to compute geometrical spreading. We set $\tilde{\mathbf{u}}(s,\tau) := \mathbf{u}(\tau)$, with $\tilde{\mathbf{u}}(s,0) = \mathbf{u}_0(s)$ and let $\tilde{X}(s,\tau) := \bar{X}\big(\tilde{\mathbf{u}}(s,\tau)\big)$ be a point on the geodesic at the distance τ from the starting point $\tilde{X}_0(s) = \tilde{X}(s,0)$ on the shadow line. The geometrical spreading of the creeping ray at $\tilde{X}(s,\tau)$ in the physical space is given by, [26],

$$\mathcal{Q}(s,\tau) = \frac{\tilde{X}^{\perp}_{\tau}\cdot\tilde{X}_s}{\tilde{X}^{\perp}_{0\tau}\cdot\tilde{X}_{0s}}. \tag{26}$$

We consider a fixed shadow line $\gamma_0(s) = (u_0(s), v_0(s), \theta_0(s))$ and define $\tilde{\gamma}(s,\tau) := \gamma(\tau)$, where γ solves (4) with initial data $\gamma_0(s)$. Let $\mathbb{L}(\gamma_0) = \{\tilde{\gamma}(s,\tau) \ : \ \tau \geq 0\}$ be a sub-manifold of phase space $\mathbb{P}$ on which the creeping rays generated at $\gamma_0(s)$ lie. The Eulerian version of the geometrical spreading $Q : \mathbb{L}(\gamma_0) \to \mathbb{R}$, restricted to $\mathbb{L}(\gamma_0)$ and defined as $Q(\tilde{\gamma}(s,\tau)) := \mathcal{Q}(s,\tau)$ is then given by

$$Q(\tilde{\gamma}) = \frac{[\hat{T}(\tilde{\gamma})\times\hat{N}(\tilde{u},\tilde{v})]^{\top} J(\tilde{u},\tilde{v}) z}{[\hat{I}\times\hat{N}(\mathbf{u_0}(s))]^{\top}\tilde{X}_{0s}(s)}, \quad \hat{T} = J\dot{\mathbf{u}}, \tag{27}$$

where $z = z(s,\tau)$ is a solution to

$$D_\gamma F(\tilde{\gamma}) z = \frac{d}{ds} F(\gamma_0(s)). \tag{28}$$

For $\tilde{\gamma}$ on the boundary, i.e. $\tilde{\gamma} = F(\gamma_0)$, the formula (27) can be simplified as,

$$Q(\tilde{\gamma}) = \frac{[\hat{T}(\tilde{\gamma})\times\hat{N}(\tilde{u},\tilde{v})]^{\top}\hat{X}_s(s)}{[\hat{I}\times\hat{N}(\mathbf{u_0}(s))]^{\top}\tilde{X}_{0s}}, \tag{29}$$

where $\hat{X} : \mathbb{R} \to \mathbb{R}^3$ is defined by $\hat{X}(s) := X(U(\gamma_0(s)), V(\gamma_0(s)))$.

Note that $\hat{X}_s(s)$ in (29) and $D_\gamma F(\tilde{\gamma})$ and $F_s(\gamma_0(s))$ in (28) can be computed by numerically differentiating the solution to the PDEs in (10) with $f = F$, as was done in [26]. Instead, one can also directly compute $\tilde{X}_s$ in (26) by adding other ODEs to the geodesic system (4) as follows: First, we note that $\tilde{X}_s = J\tilde{\mathbf{u}}_s$. We then differentiate (4) with respect to s and derive the following ODE system

$$\dot{\tilde{\gamma}}_s = D_\gamma \mathbf{g}\, \tilde{\gamma}_s, \qquad \tilde{\gamma}_s(s,0) = \tilde{\gamma}_{0s}(s). \tag{30}$$

By solving this ODE, $\tilde{\mathbf{u}}_s$ and therefore $\tilde{X}_s$ can be computed. One can also write the escape PDE for (30) in the same way as before and post-process the phase space solution.

5.2 Multiple-Patch Scheme

We now split the scatterer surface M into several simpler surfaces with explicit regular parameterizations. As before, let M be given by an atlas of charts (M_j, w_j), where the patches $M_j \subset \mathbb{R}^3$ have the parametric equations $\mathbf{x} = \bar{X}_j(\mathbf{u}) : [0,1]^2 \to M_j$ and collectively cover M. Moreover, the mappings $w_j = \bar{X}_j^{-1} : M_j \to [0,1]^2$ are bijective.

Since on a geodesic $\hat{T}$ in (27) has unit length, we can consider the unit tangent bundle UTM of M as the global space. Note that UTM is a three-dimensional manifold embedded in $\mathbb{R}^6$. By Lemma 1, there is therefore a bijective mapping $W_j : UTM_j \to \Omega_p$ for each j, defined by $W_j(\Gamma) = \gamma$, with γ and Γ as in (8).

Knowing the bijective mappings w_j and W_j, and the solution to the escape PDEs in each patch, F_j, Φ_j and B_j, we can compute the multiple-patch escape functions $\mathbf{F}$, $\mathbf{\Phi}$ and $\mathbf{B}$ as described in Section 4.1.

5.3 Post-Processing

In order to compute phase and amplitude of a ray family, post-processing of the solutions to the escape PDEs (10) is needed.

For a given illumination direction, assume that the shadow line is known and given by $\Gamma_0(s)$ in the unit tangent bundle UTM. For each point $\mathbf{x} \in M_j$ covered by the surface wave, there is at least one creeping ray which starts at the shadow line and passes through that point. In order to find this ray, assuming the scatterer surface is boundaryless, we first choose the escape boundary $\mathcal{R}$ as the boundaries of M_j. Note that in the case of a surface with boundary, we choose its boundary as the escape boundary, and the post-processing will be similar to the single-patch case discussed in [26]. We then find $\mathbf{F}\left(W_j^{-1}(w_j(\mathbf{x}), \theta)\right)$ for all directions $\theta \in \mathbb{S}$. Moreover, for all points on the shadow line we find $\mathbf{F}_n\left(\Gamma_0(s)\right)$, defined by (19), with $n = 1, 2, \ldots$. We then find $s = s^*$, $\theta = \theta^*$ and $n = n^*$ as the solutions to the algebraic equations

$$\mathbf{F}\left(W_j^{-1}(w_j(\mathbf{x}), \theta)\right) = \mathbf{F}_n\left(\Gamma_0(s)\right), \tag{31}$$

analogous to (14) in the single-patch case. There will be at most four systems of equations corresponding to four sides of patch M_j, for each value of n. The solutions to (31) can be computed by finding intersections of four sets of possibly crossing curves.

Now we can use (28) to compute z with $\gamma_0 = W_{j_0}(\Gamma_0(s^*))$ and $\tilde{\gamma} = (w_j(\mathbf{x}), \theta^*)$ where $j_0 = \mathcal{J}(\Gamma_0(s^*))$. Note that $F(\tilde{\gamma})$ and $F(\gamma_0)$ in the left and right hand sides of (28) are

replaced by $W_j\big(\mathbf{F}(W_j^{-1}(\tilde{\gamma}))\big)$ and $W_j\big(\mathbf{F}_{n^*}(\Gamma_0(s^*))\big)$, respectively. The geometrical spreading $Q(\tilde{\gamma})$ at point $\mathbf{x}$ will be therefore computed by (27), and phase and amplitude are given by

$$\phi(w_j(\mathbf{x})) = \phi_0 + \mathbf{\Phi}\big(W_{j_0}^{-1}(\gamma_0)\big) - \mathbf{\Phi}\big(W_j^{-1}(\tilde{\gamma})\big),$$

$$A(\tilde{\gamma}) = A_0\, Q(\tilde{\gamma})^{\frac{-1}{2}} \exp\bigg(- \omega^{\frac{1}{3}}\Big(\mathbf{B}\big(W_{j_0}^{-1}(\gamma_0)\big) - \mathbf{B}\big(W_j^{-1}(\tilde{\gamma})\big)\Big)\bigg),$$

where ϕ_0 and A_0 are the phase and amplitude at the point γ_0, respectively.

5.4 Example 1 - A Scalene Ellipsoid

We consider the scatterer surfaces to be a scalene ellipsoid (an ellipsoid with different semi-axes) and apply the multiple-patch phase space method to compute the contribution of *backscattered* creeping rays to mono-static RCS, i.e., the rays that propagate on the surface of the scatterer and return in the opposite direction of incident waves. We assume that the incoming amplitudes are one at attachment points on the shadow line and compute the backscattered amplitude at detachment points on the shadow line. We also compute the length of the backscattered rays.

We consider an ellipsoid given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1,$$

with $a = 2$, $b = 1$ and $c = 0.5$. Since there is no single non-singular parameterization for the ellipsoid, we split it into six patches with non-singular parameterizations (see Figure 5) and solve for $f(\gamma)$ in each patch, as described in Section 3.2.

In order to find the backscattered creeping ray by post-processing, we first choose the escape boundary consisting of six sides, as highlighted in Figure 6. We then continue as follows,

0. Given a pair of incident angles $(\Psi_1, \Psi_2) \in [0, 90]^2$, find the incident direction $\hat{I} = [\sin\Psi_1 \cos\Psi_2,\ \cos\Psi_1 \cos\Psi_2,\ \sin\Psi_2]$.

1. Find the shadow line $\gamma_0(s) = (\mathbf{u}_0(s), \theta_0(s))$ in the phase space Ω_p using the relations $\hat{N}^\top \hat{I} = 0$ and $\hat{T}(\gamma_0(s)) = \hat{I}$ in patch $j(s)$. Let the parameterization of the shadow line be discretized in N grid points $\{s_n\}$ with $n = 1, \ldots, N$.

2. For each point on the shadow line find $\mathbf{F}\left(W_{j(s_n)}^{-1}(\gamma_0(s_n))\right)$ as discussed in Section 4.1.

3. A backscattered ray starting at attachment point s_a and ending at detachment point s_d on the shadow line should satisfy

$$\mathbf{F}\left(W_{j_a}^{-1}(\gamma_0(s_a))\right) = \mathbf{F}\left(W_{j_d}^{-1}(\gamma_0(s_d))\right) + C,$$

where $j_a = j(s_a)$ and $j_d = j(s_d)$, and C is a constant accounting for the fact that the directions of creeping rays starting at s_a and s_d differ by a π on the escape boundary. The right and left hand sides of this equation can be represented as six sets of curves in $\mathbb{R}^2$ parameterized by s, corresponding to six sides of the escape boundary. To find the backscattered ray we need to find crossing points of these curves, as is done in the single patch case.

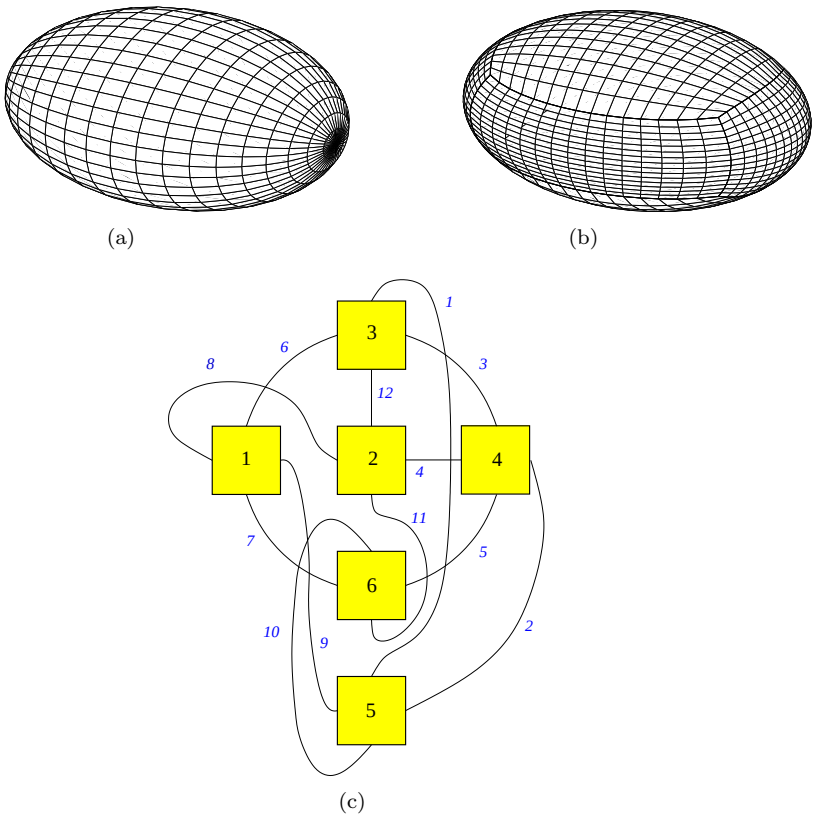


Figure 5: Upper left figure shows an ellipsoid with a single patch parameterization which is singular at two poles. Upper right figure shows the ellipsoid divided into 6 patches. Note that the singularities have been removed using non-singular multiple parameterizations. Lower figure shows the structure of patches and patch boundaries in parameter space. Patches $j = 1, \ldots, 6$ correspond to left, front, up, right, back and down patches, respectively. These 6 patches share 12 sides in total, shown with italic numbers.

4. For each crossing point, there is a pair of backscattered rays (two backscattered rays lying on top of each other); one starting at point s_a and ending at point s_d, the other starting at point s_d and ending at point s_a. Although these two rays have the same lengths, they do not have the same geometrical spreading and therefore not the same amplitude. Compute two geometrical spreadings as described in Section 5.3 with $\gamma_0 = \gamma(s_d)$ and $\tilde{\gamma} = \gamma(s_a)$ for the first backscattered ray and $\gamma_0 = \gamma(s_a)$ and $\tilde{\gamma} = \gamma(s_d)$ for the second one.

5. The length and amplitudes are then computed as,

$$\phi = \mathbf{\Phi}(\Gamma_{s_a}) + \mathbf{\Phi}(\Gamma_{s_d}), \quad \Gamma_{s_a} = W_{j_a}^{-1}(\gamma(s_a)), \quad \Gamma_{s_d} = W_{j_d}^{-1}(\gamma(s_d)),$$

$$A_1 = Q(\gamma(s_a))^{\frac{-1}{2}} \exp\left(-\omega^{\frac{1}{3}}\left(\mathbf{B}(\Gamma_{s_a}) + \mathbf{B}(\Gamma_{s_d})\right)\right),$$

$$A_2 = Q(\gamma(s_d))^{\frac{-1}{2}} \exp\left(-\omega^{\frac{1}{3}}\left(\mathbf{B}(\Gamma_{s_a}) + \mathbf{B}(\Gamma_{s_d})\right)\right).$$

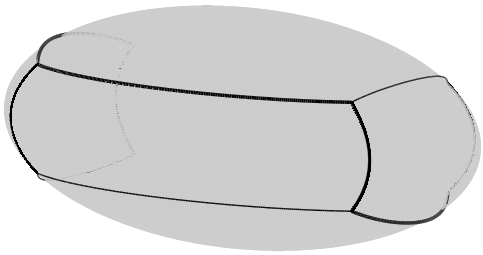


Figure 6: The ellipsoid with its patch boundaries. Thick lines show the escape boundary.

Figure 7 shows the backscattered rays for two different incident angles. There are three pairs of backscattered rays which can be detected by the algorithm. Every two rays of each pair lie on top of each other.

Figure 7: Left figure shows the backscattered creeping rays (thick curves) for $\Psi_1 = 30$ and $\Psi_2 = 0$. Right figure shows the backscattered creeping rays for $\Psi_1 = 30$ and $\Psi_2 = 10$. Thin curves represent the shadow lines.

We notice that in [26], because of using a single patch and excising the singularity at two poles, only the shortest backscattered ray could be captured. Figure 8 shows the length and amplitudes of the shortest backscattered ray for different incident angles, with $\omega = 1$. The peaks in the amplitude correspond to *caustic backscattered creeping rays* which have infinite amplitudes. Such rays are particularly important in near-field RCS computations. However, in far-field RCS, due to the the geometrical spreading outside the scatterer, their contribution may not be as important.

Figure 9 shows the convergence of length and amplitudes of the backscattered creeping ray for a fixed vertical angle $\Psi_2 = 70$ and different horizontal incident angles $\Psi_1 \in [-90, 90]$. We use a second order Fast Marching algorithm on a coarse grid of the size 50^3 and a fine grid of the size 100^3. We compare them with a reference solution obtained by a high order ray tracing method. The rate of convergence confirms the second order accuracy of the algorithm. We note that comparing to the results in [26], where a first order algorithm was used, the accuracy of amplitude has been improved dramatically. It shows that using a first order accurate method for computing the phase and amplitude results in a worse relative

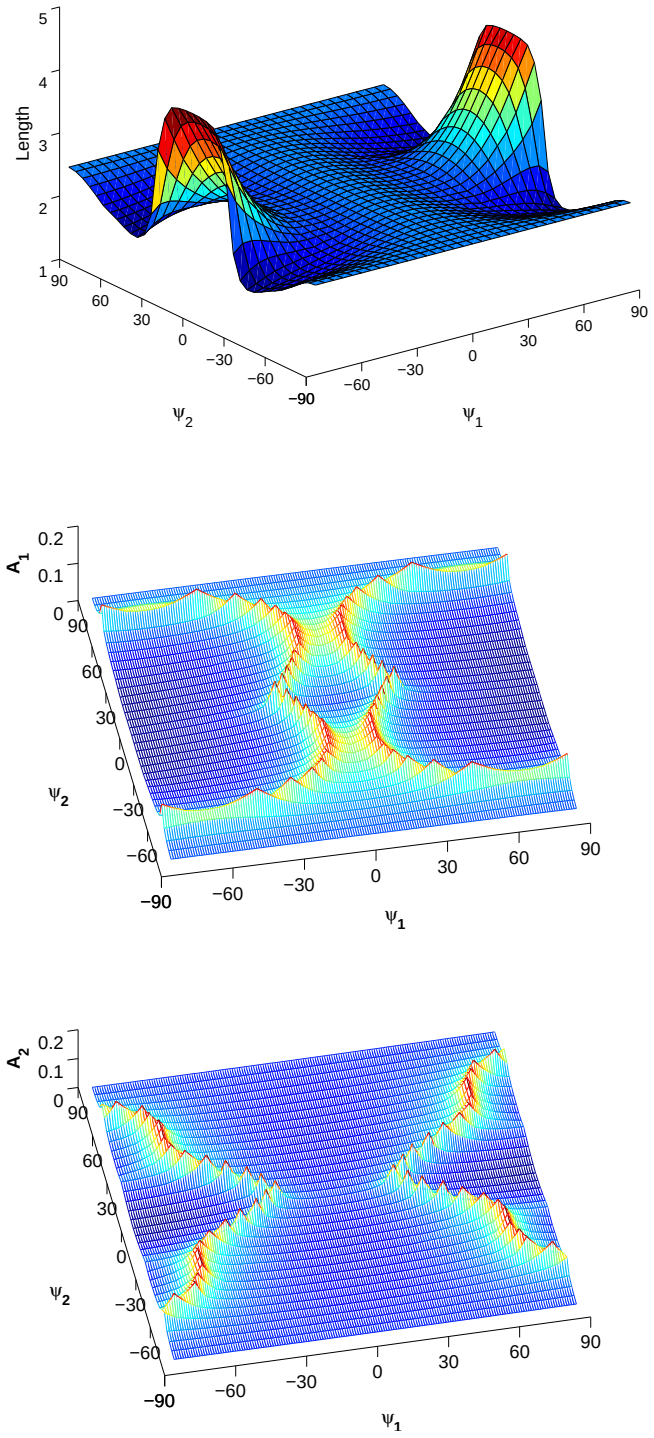


Figure 8: Length and amplitudes (with $\omega = 1$) of backscattered creeping rays for many illumination angles.

error for the amplitude than for the phase. Therefore, higher order algorithms are required to obtain low relative errors for the amplitude, as observed also in [30].

The complexity of using the fast phase space method proposed here consists of two parts. First, the cost of solving the PDEs by the Fast Marching method is $\mathcal{O}(N^3 \log N)$. Second, the cost of finding the backscattered rays for each shadow line is $\mathcal{O}(N)$. For all N^2 shadow lines, it is $\mathcal{O}(N^3)$. Therefore the total complexity will be $\mathcal{O}(N^3 \log N)$. The total cost by using other methods, like wave front tracking and solvers based on the surface eikonal equation, will be $\mathcal{O}(N^4)$, if the cost for each shadow line is $\mathcal{O}(N^2)$. In this case, using the phase space method will then be much faster.

Remark 4. *A graph structure can be useful for a general computer implementation. The topology of the surface can be described by a graph, in which each patch is a node and the edges go between connected patches. Figure 10a shows the graph corresponding to the ellipsoid divided into six patches which are connected through twelve sides (see Figure 5). The graph therefore has six nodes and twelve edges.*

We can also introduce another topology graph, in which the nodes are the sides of the patches and the edges correspond to the patches themselves. Each node (side) is therefore connected to six other nodes through two patches which are connected by that side. See Figure 10b. This structure can be useful for imposing inter-patch boundary conditions and computing $\mathbf{F}$.

5.5 Example 2 - A balloon

We consider a balloon-shape surface consisting of a hemisphere in the positive side of the z-axis, centered at the origin and with radius r, and the surface created by rotating the parabola $z^2 = 2r(r-y)$ about the z-axis in its negative side. This is a simple smooth version of the cone-hemisphere studied in [3] as a model for low-observable objects where creeping rays are important for RCS. We divide this surface into six patches, as shown in Figure 11; The hemisphere is split into five patches $j = 1, \ldots, 5$, and the parabolic part is represented by one patch $j = 6$. We excise the singularity at the vertex of the balloon by cutting it off. The lower boundary of patch $j = 6$ will therefore be an excision boundary and is not considered as a patch boundary. We also partition the upper boundary of patch $j = 6$ into four boundaries connecting to lower boundaries of patches $j = 1, \ldots, 4$. Note that the left and right boundaries of patch $j = 6$ are in fact the same. Therefore, there are in total thirteen sides connecting six patches. See Figure 11.

Since the surface is symmetric about the z-axis, we consider a fixed horizontal incident angle $\Psi_1 = 90$, and due to symmetry about the yz-plane, we consider the vertical angles $\Psi_2 \in [-90, 90]$. Figure 12 shows the backscattered rays for two different incident angles $\Psi_2 = 40$ and $\Psi_2 = -40$. For positive vertical incident angles, there are four pairs of backscattered rays which can be detected by the algorithm. Two of them are symmetric and have the same length and amplitudes. For negative vertical incident angles, only one backscattered ray can be captured. We notice that in the case $\Psi_2 = 90$, there will be infinitely many backscattered rays which results in high observability of the object in this incident direction. On the other hand, for $\Psi_2 = -90$, there will be no backscattered ray because we have excised the vertex. In fact even if we did not excise it, all creeping rays would go to the vertex and diffract in different directions.

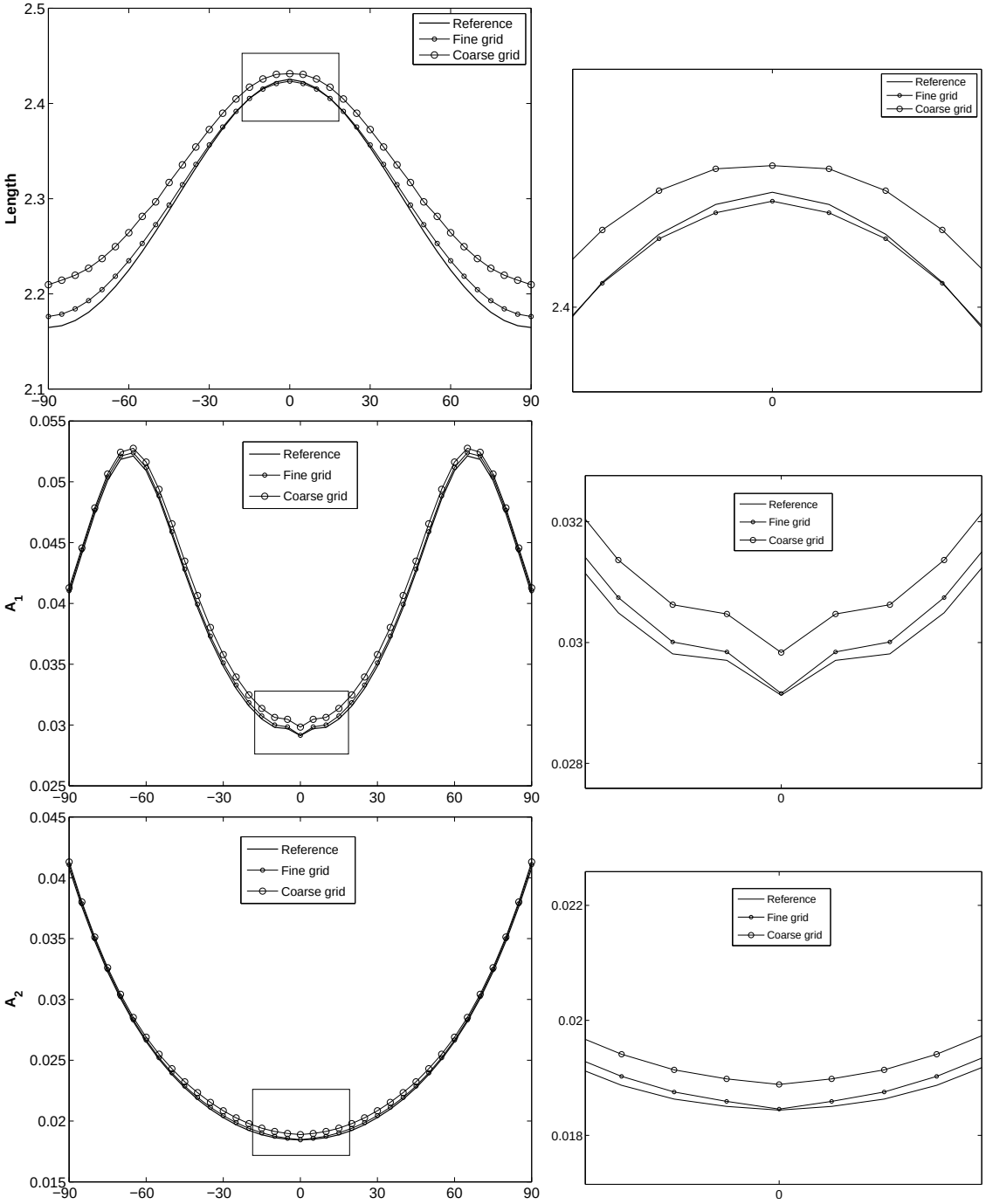


Figure 9: Length and amplitude (with $\omega = 1$) of the backscattered creeping rays for different horizontal incident angles and a fixed vertical angle $\Psi_2 = 70$. By refining the grid, solutions of the second order phase space algorithm converge to a reference solution obtained by a high order ray tracing method with a correct rate. Right figures show zoomed views of left figures.

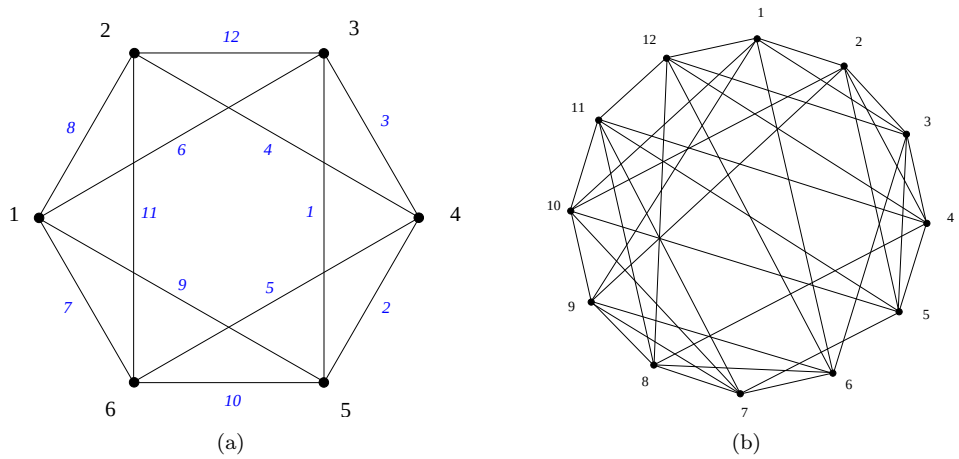


Figure 10: Representation of an ellipsoid divided into 6 patches by two different graph structures. Left figure shows the graph with 6 nodes and 12 edges. Here, the nodes 1 to 6 denotes the left, front, up, right, back and down patches, respectively. Right figure shows the graph with 12 nodes and 72 edges.

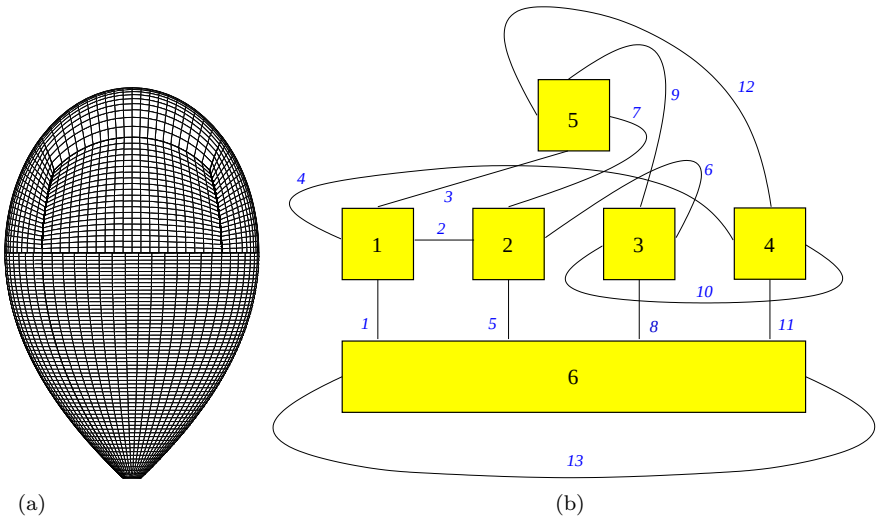


Figure 11: Left figure shows the balloon divided into 6 patches. Right figure shows the structure of patches and patch boundaries in parameter space. Patches $j = 1, \ldots, 6$ correspond to front, right, back, left, up and down patches, respectively. These 6 patches share 13 sides in total, shown with italic numbers.

Figure 13 shows the length and amplitude of backscattered rays in a polar coordinate system for all incident directions $\Psi \in [0, 360]$. The angles $\Psi \in [0, 90]$ in the polar system correspond to $\Psi_2 \in [0, -90]$, and the angles $\Psi \in [270, 360]$ correspond to $\Psi_2 \in [90, 0]$. The values for $\Psi \in [90, 270]$ are then calculated using the symmetry of the surface about the yz-plane.

Figure 12: Backscattered creeping rays (thick curves) for $\Psi_2 = 40$ (left figure) and $\Psi_2 = -40$ (right figure). Thin curves represent the shadow lines.

6 Application to Seismic Wave Computations

The inhomogeneity of earth causes deflection and reflection of seismic waves. The numerical study of seismic wave propagations, therefore, helps us to learn about the inhomogeneous structure of earth, which is important in direct and inverse problems of seismology and seismic exploration of oil.

In this section, we apply the multiple-patch phase space method to compute the travel-time of seismic rays. We consider a two-dimensional multi-layered medium whose different layers have different wave speeds. We split the medium into multiple patches corresponding to different layers. The escape PDEs describing seismic waves are solved in each patch, individually. The travel-time of the waves in the medium are then computed by connecting all individual solutions. The inter-patch boundaries are treated by Snell's law and the law of reflection.

We first consider the case when the medium has a regular explicit parameterization and derive the governing equations. We then discuss the multiple-patch scheme and give a numerical example for computing the travel-times.

6.1 Governing Equations

Consider a two-dimensional medium M represented by parametric equations $\mathbf{x} = \bar{X}(\mathbf{u})$, where $\mathbf{x} = (x, y) \in M \subset \mathbb{R}^2$ and $\mathbf{u} = (u, v) \in \Omega \subset \mathbb{R}^2$.

The phase ϕ of the wave satisfies the eikonal equation,

$$|\nabla \phi| = n(\mathbf{x}), \tag{32}$$

which is a Hamilton-Jacobi equation. The Hamiltonian for the eikonal equation can be written in the form

$$H(\mathbf{x}, \mathbf{p}) = c(\mathbf{x})|\mathbf{p}| \equiv 1, \tag{33}$$

where $c(\mathbf{x}) = 1/n(\mathbf{x})$ is the wave speed and $\mathbf{p} = \nabla \phi$. Introducing the arc length parameter

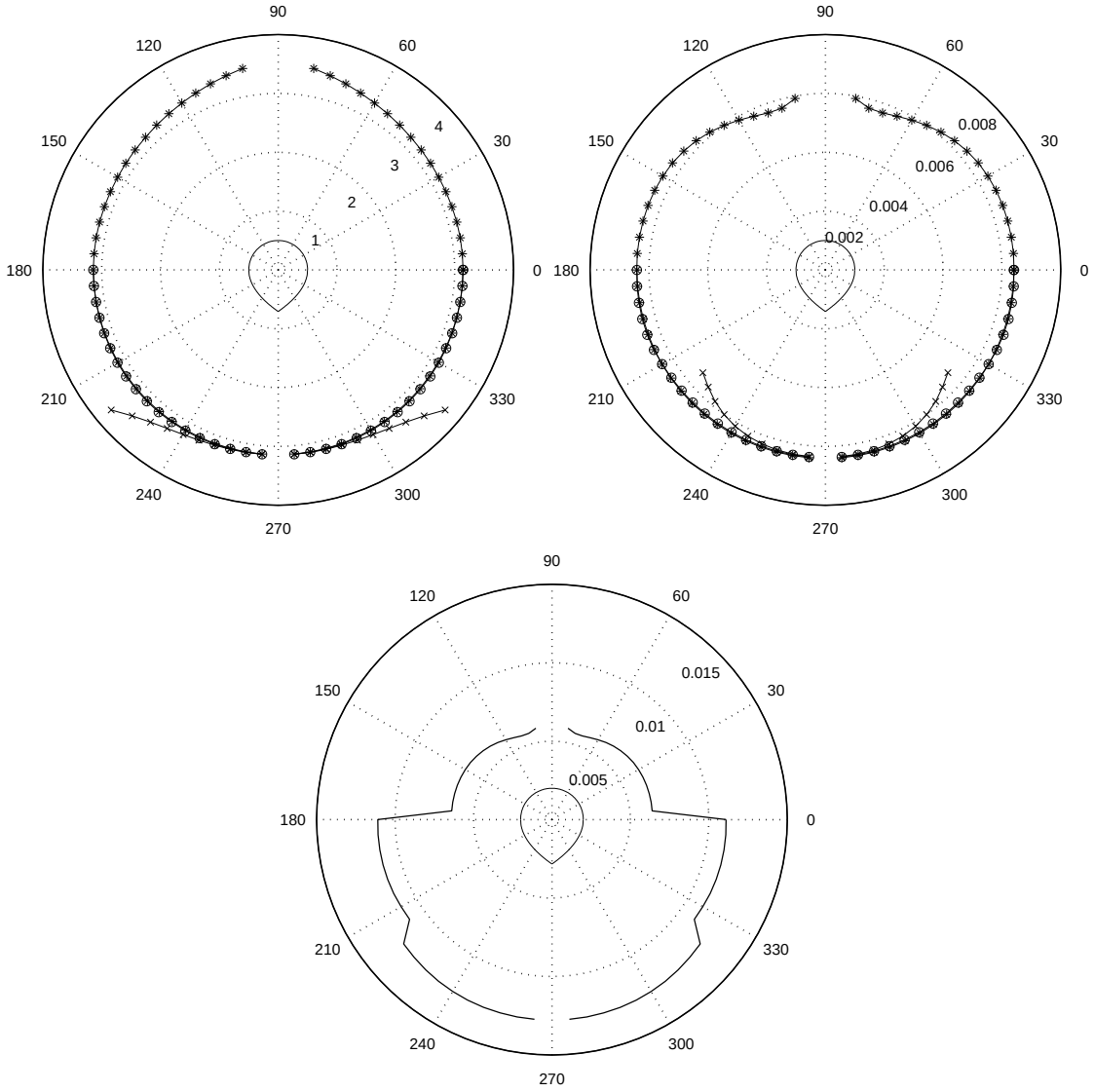


Figure 13: Length and amplitude (with $\omega = 1$) of the backscattered creeping rays for all illumination directions $\Psi \in [0, 360]$. Upper left and right figures show the length and amplitude of the backscattered rays, respectively. There are four pairs of rays among which two (illustrated by $\circ$) are symmetric. Note that at $\Psi = 90$ ($\Psi_2 = -90$), there will be no backscattered ray because all creeping rays go to the vertex and diffract in different directions. At $\Psi = 270$ ($\Psi_2 = 90$), however, there are infinitely many backscattered rays resulting in high observability of the object in this incident direction, and therefore the values are not shown. Because of the excision, the longest backscattered ray (illustrated by $\times$) can be captured only for $\Psi \in [220, 320]$ ($\Psi_2 \geq 40$). Bottom figure shows the total amplitude, $A_{tot} = \sqrt{A_1^2 + A_2^2 + A_3^2 + A_4^2}$, of all four backscattered creeping rays.

τ, a ray trajectory $(\mathbf{u}(\tau), \mathbf{p}(\tau))$ in $\Omega \times \mathbb{R}^2$ is given by the Hamiltonian system

$$\dot{\mathbf{x}} = c(\mathbf{x}) \frac{\mathbf{p}}{|\mathbf{p}|} = c^2(\mathbf{x})\, \mathbf{p}, \tag{34a}$$

$$\dot{\mathbf{p}} = -|\mathbf{p}|\, \nabla_{\mathbf{x}} c(\mathbf{x}) = -\frac{\nabla_{\mathbf{x}} c(\mathbf{x})}{c(\mathbf{x})}, \tag{34b}$$

where the dot denotes differentiation with respect τ.

Since $\dot{\mathbf{x}} = J(\mathbf{u})\,\dot{\mathbf{u}}$ with the Jacobian $J = [\bar{X}_u\, \bar{X}_v] \in \mathbb{R}^{2\times 2}$, we have

$$\dot{\mathbf{u}} = J^{-1}(\mathbf{u})\,\dot{\mathbf{x}} = c^2(\bar{X}(\mathbf{u}))\, J^{-1}(\mathbf{u})\,\mathbf{p}. \tag{35}$$

Moreover, inspired by $|\mathbf{p}| = \frac{1}{c(\mathbf{x})}$, we set $\mathbf{p} = (p_1, p_2)^\top = \frac{1}{c(\mathbf{x})}\,(\cos\theta, \sin\theta)^\top$. Differentiating $\mathbf{p}$ with respect to τ, we get

$$\dot{\mathbf{p}} = \begin{pmatrix} \nabla_{\mathbf{x}} \frac{1}{c(\mathbf{x})} \cdot \dot{\mathbf{x}}\, \cos\theta - \frac{1}{c(\mathbf{x})} \sin\theta\, \dot{\theta} \\ \nabla_{\mathbf{x}} \frac{1}{c(\mathbf{x})} \cdot \dot{\mathbf{x}}\, \sin\theta + \frac{1}{c(\mathbf{x})} \cos\theta\, \dot{\theta} \end{pmatrix}. \tag{36}$$

By (34) and (36), we get $\dot{\theta} = c_x(\mathbf{x}) \sin\theta - c_y(\mathbf{x}) \cos\theta$. Therefore, setting $\gamma := (u, v, \theta) \in \Omega_p$, the function $\mathbf{g}(\gamma)$ in (4) will be

$$\mathbf{g}(\gamma) = \begin{pmatrix} c(\bar{X}(\mathbf{u}))\,(g^{11}\,\cos\theta + g^{12}\,\sin\theta) \\ c(\bar{X}(\mathbf{u}))\,(g^{21}\,\cos\theta + g^{22}\,\sin\theta) \\ c_x(\bar{X}(\mathbf{u}))\sin\theta - c_y(\bar{X}(\mathbf{u}))\cos\theta \end{pmatrix}, \tag{37}$$

where $(g^{ij}) = J^{-1}(\mathbf{u})$. Note that since $\dot{\mathbf{x}} \parallel \mathbf{p}$ by (34), the angle θ represents the direction of the ray trajectory at $\mathbf{x}$ in the physical space. Moreover, with our choice of Hamiltonian,

$$\dot{\phi}(\mathbf{x}(\tau)) = \nabla\phi(\mathbf{x}(\tau)) \cdot \dot{\mathbf{x}}(\tau) = \mathbf{p} \cdot \mathbf{p}\frac{c(\mathbf{x}(\tau))}{|\mathbf{p}|} = |\mathbf{p}|\, c(\mathbf{x}(\tau)) = 1,$$

implying that ϕ corresponds to travel-time.

6.2 Multiple-Patch Scheme

We assume that the physical domain, representing a medium, is a two-dimensional compact manifold $M \subset \mathbb{R}^2$ with boundary. Since the wave speed distribution in a multi-layered inhomogeneous medium is not continuous, it is natural to split the medium to different patches with continuous wave speed distributions. We now let M be described by an atlas of charts (M_j, w_j) as before. The three-dimensional unit tangent bundle UTM is embedded in $\mathbb{R}^4$. In this case, there is an easier way to represent UTM by simplifying the mapping $W_j : UTM_j \to \Omega_p$ to be $W_j(\Gamma) = \gamma$, where

$$\Gamma = (\mathbf{x}, \mathbf{q}), \quad \mathbf{q} = \hat{s}(\theta) = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}, \quad \gamma = (w_j(\mathbf{x}), \theta). \tag{38}$$

In the same way as before, we can define and compute multiple-patch escape functions $\mathbf{F}(\Gamma)$ and $\mathbf{\Phi}(\Gamma)$. However, here the rays are not continuous at the patch-boundaries due to the change of the wave speed at these points. When a ray passes the boundary between two layers (two neighboring patches) with different wave speeds, part of the ray is reflected (by the law of reflection), and part of it is refracted or transmitted into the second layer (by Snell's law of refraction). At each interface, therefore, the ray field splits into two new ray families, one reflected and one transmitted.

Figure 14a shows the reflection and refraction of a ray at the interface between two media of different wave speeds, with $c_L > c_R$. The law of reflection gives the relation between the angles of incidence (θ_{inc}) and of reflection (θ_{ref}) as

$$\theta_{inc} = \theta_{ref}. \tag{39}$$

The relation between the angles of incidence and of refraction (θ_{tr}) for a ray crossing a boundary between different media is given by Snell's law

$$\frac{\sin\theta_{inc}}{\sin\theta_{tr}} = \frac{c_L}{c_R}. \tag{40}$$

When a ray moves from a dense to a less dense medium ($c_L < c_R$), Snell's law cannot be used to calculate the refracted angle if $\sin\theta_{tr} = \sin\theta_{inc}\ (c_L/c_R) > 1$. At this point, the ray is reflected in the incident medium, known as *internal reflection*. There is therefore a critical angle (θ_{cr}) for which the ray travels directly along the surface between the two refractive media. The critical angle is found by Snell's law, putting in a transmitted angle of 90 degrees. This gives:

$$\theta_{cr} = \arcsin\frac{c_L}{c_R}. \tag{41}$$

For any angle of incidence larger than the critical angle ($\theta_{inc} > \theta_{cr}$), the ray is totally reflected off the interface, obeying the law of reflection. This phenomena is called *total internal reflection*. See Figure 14b.

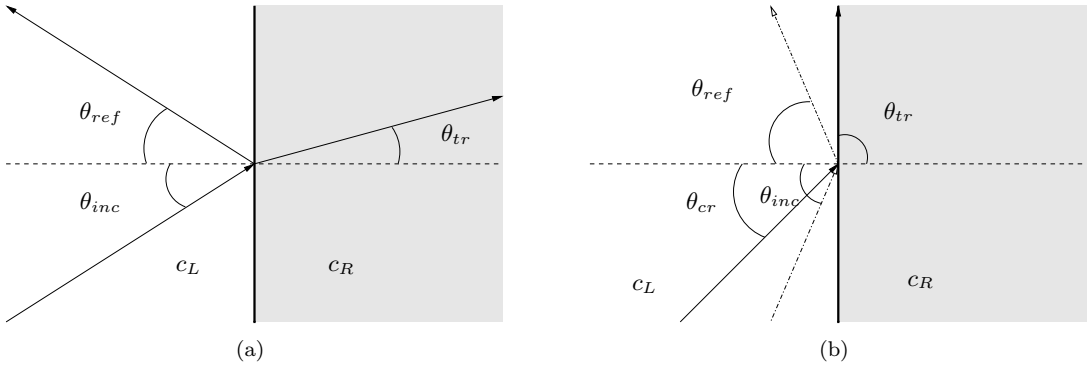


Figure 14: Reflection and refraction of a ray at the interface between two media of different wave speeds. Left figure shows the reflection and refraction when $c_L > c_R$. Right figure shows the internal reflection when $\theta_{inc} \geq \theta_{cr}$.

From (39)-(41), we can easily find the inter-patch boundary functions $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{R}}$ discussed in Section 2.2.1. Post-processing in this case is similar to that of the single-patch case, because the escape boundary that we chose coincides with the external boundary of the medium.

Note that the inter-patch boundary conditions above can be seen as a way to preserve the Hamiltonian (33) for a ray across the patch boundary. In cases where the discontinuity in $c(\mathbf{x})$ is not aligned with the patch boundary, the solution of the escape equations is not unique. Uniqueness can however be recovered by enforcing the extra condition that solutions should be continuous along constant Hamiltonian paths also inside the patches. This is the idea of so called Hamiltonian-preserving methods developed in [14, 15]. These methods capture the effect of a discontinuous $c(\mathbf{x})$ on uniform grids not aligned with the discontinuity.

6.3 Example 3 - A multi-layered medium

We consider a multi-layered medium $M = [0, 6]^2$ consisting of three layers with different wave speeds (see Figure 15):

- Top layer: $c_1(x, y) = 1 + 0.05\,(x-3)^2 + 0.25\,y,$
- Middle layer: $c_2(x, y) = \frac{3}{1+e^{-((x-3)^2+(y-3)^2)}},$
- Bottom layer: $c_3(x, y) = 0.5 + 0.2\,x + 0.5\,y.$

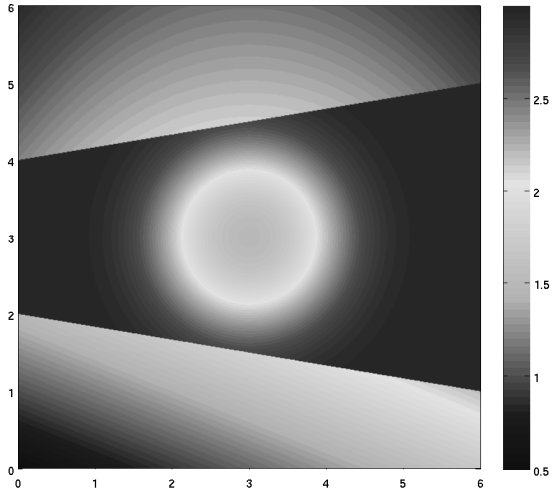


Figure 15: The medium consisting of three layers and grey scale plot of the wave speed field.

We want to compute multivalued travel-time of seismic rays in the medium from a given source point $\mathbf{x}_0$ on the boundary. We split the medium into three patches corresponding to the three layers, as shown in Figure 15. The escape equations for the escape point F and the travel-time Φ are derived and solved in each patch.

In order to find the travel-time with a given source point by post-processing, we first choose the four outermost boundaries of the entire physical domain as the escape boundary. We then continue as follows:

0. The source point $\mathbf{x}_0$ on the boundary is first reduced to a point $S_0 \in \mathbb{R}$.
1. For each point $\mathbf{x} \in M$, find $\mathbf{F}(\Gamma) = (\mathbf{U}, \mathbf{V}, \boldsymbol{\Theta})$ with $\Gamma = (\mathbf{x}, \mathbf{q}(\theta))$ for all $\theta \in \mathbb{S}$. Now $(\mathbf{U}, \mathbf{V})$ can again be reduced to points $S \in \mathbb{R}$, parameterized by θ.
2. Find $\theta = \theta^*$ such that $S_0 = S(\theta)$.
3. Travel-time at $\mathbf{x} \in M$ will then be $\boldsymbol{\Phi}(\Gamma^*)$ with $\Gamma^* = (\mathbf{x}, \mathbf{q}(\theta^*))$.

Figure 16 shows the distribution of transmitted seismic rays and equiarrival curves, i.e., the locus of all points in physical domain which have the same travel-time, from two different source points, $\mathbf{x}_0 = (3, 6)$ and $\mathbf{x}_0 = (3.5, 6)$.

Note that we can track both reflected and transmitted ray families, but not at the same time. In order to get all rays, one needs to follow all ray families. Figure 17 shows the equiarrival curves of reflected rays from the top and bottom interfaces inside the top and the middle layers, respectively, for a source point at $\mathbf{x}_0 = (3, 6)$. If we repeat this procedure, we can also capture multiple reflected rays from the two interfaces that get trapped inside the middle layer and reverberate to infinity. Here, we do not consider reflections from the

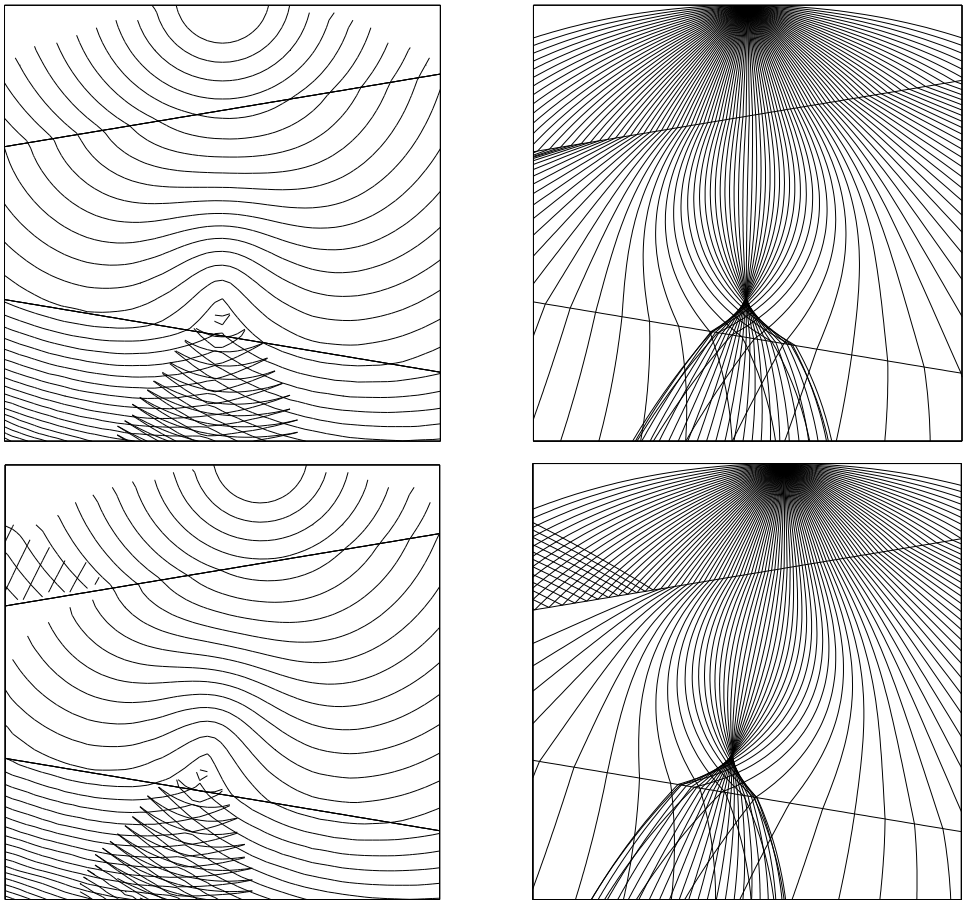


Figure 16: The equiarrival curves and the distribution of seismic rays for two different source points.

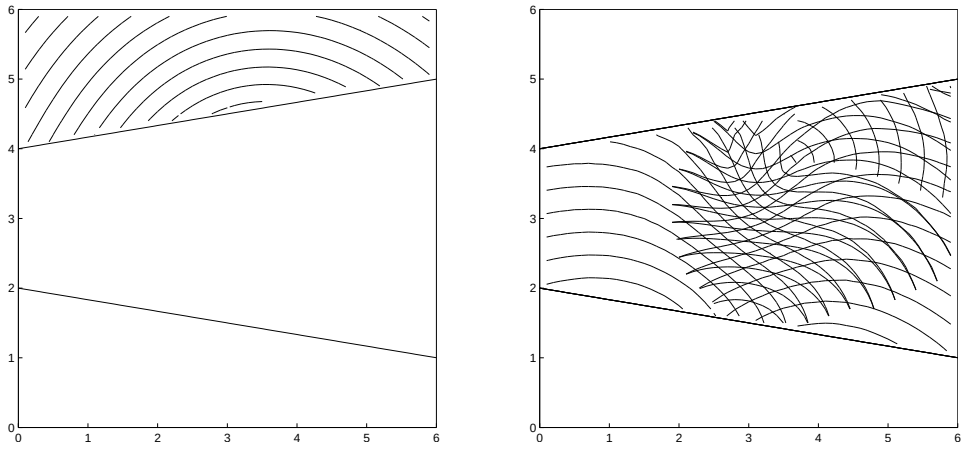


Figure 17: The equiarrival curves of reflected seismic rays from the top (left figure) and bottom (right figure) interfaces for a source point on the center of the top of the domain.

domain boundaries, as we have a truncated domain much smaller than the physical space in which the waves propagate.

7 Conclusion

We have modified the single-patch phase space method for computing creeping rays to a multiple-patch method for computing trajectories on two-dimensional manifolds possibly embedded in a higher-dimensional space. The dynamics of trajectories are given by systems of first-order ODEs in a phase space. We split the manifold into multiple patches where each patch has a well-defined regular parameterization. The *escape* equations, which are hyperbolic PDEs in a three-dimensional phase space, are derived and solved in each patch, individually, using a second-order version of the fast marching method. The solutions of individual patches are then connected using suitable inter-patch boundary conditions. Properties for particular families of trajectories are obtained through a fast post-processing. For some applications, the complexity of the method is attractive. Such applications include mono-static and bi-static RCS computations, antenna coupling problems, and travel-time computations of seismic waves when the solution is sought for many different sources.

References

[1] N. C. Albertsen. Creeping wave modes for a dielectric coated cylinder. *IEEE T. Antenn. Propag.*, 37(12):1642–1644, 1989.

[2] C. Belta and V. Kumar. Optimal motion generation for groups of robots: A geometric approach. *J. Mech. Des.*, 126:36–70, 2004.

[3] D. P. Bouche, J.-J. Bouquet, H. Manenc, and R. Mittra. Asymptotic computation of the RCS of low observable axisymmetric objects at high frequency. *IEEE T. Antenn. Propag.*, 40(10):1165–1174, 1992.

[4] H. Brandén and S. Holmgren. Convergence acceleration for hyperbolic systems using semicirculant approximations. *J. Sci. Comput.*, 14(4):357–393, 1999.

[5] T. F. Chan and T. P. Mathew. Domain decomposition algorithms. *Acta Numerica*, 3:61–143, 1994.

[6] V. Červený, I. A. Molotkov, and I. Psencik. *Ray Methods in Seismology.* Univ. Karlova Press, 1977.

[7] B. Engquist and O. Runborg. Computational high frequency wave propagation. *Acta Numerica*, 12:181–266, 2003.

[8] E. Fatemi, B. Engquist, and S. J. Osher. Numerical solution of the high frequency asymptotic expansion for the scalar wave equation. *J. Comput. Phys.*, 120(1):145–155, 1995.

[9] S. Fomel and J. A. Sethian. Fast phase space computation of multiple arrivals. *Proc. Natl. Acad. Sci. USA*, 99(11):7329–7334 (electronic), 2002.

[10] S. Hagdahl. *Hybrid Methods for Computational Electromagnetics in Frequency Domain.* PhD thesis, NADA, KTH, Stockholm, 2005.

[11] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, 2001. ACM Press, New York, NY, USA.

[12] S. Holmgren and K. Otto. Semicirculant preconditioners for first-order partial differential equations. *SIAM J. Sci. Comput.*, 15:385–407, 1994.

[13] P. E. Hussar, V. Oliker, H. L. Riggins, E.M. Smith-Rowlan, W.R. Klocko, and L. Prussner. An implementation of the UTD on facetized CAD platform models. *IEEE Antennas Propag.*, 42(2):100–106, 2000.

[14] S. Jin and X. Wen. Hamiltonian-preserving schemes for the Liouville equation with discontinuous potentials. *Comm. Math. Sci.*, 3:285–315, 2005.

[15] S. Jin and X. Wen. Hamiltonian-preserving schemes for the Liouville equation of geometrical optics with discontinuous local wave speeds. *J. Comput. Phys.*, 214(2):672–697, 2006.

[16] K. H. Karlsen, K. A. Lie, and N. H. Risebro. A fast marching method for reservoir simulation. *Computational Geosciences*, 4(2):185–206, 2000.

[17] J. Keller and R. M. Lewis. Asymptotic methods for partial differential equations: the reduced wave equation and Maxwell's equations. *Surveys Appl. Math.*, 1:1–82, 1995.

[18] J. B. Keller. The geometric theory of diffraction. In *Symposium on Microwave Optics*, Eaton Electronics Research Laboratory, McGill University, Montreal, Canada, June 1953.

[19] L. C. Kempel, J. L. Volakis, and S. Bindiganavale. Radiation and scattering from printed antennas on cylindrically conformal platforms. Technical report, Michigan Univ. Final Report, January 1994.

[20] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. USA*, 95(15):8431–8435 (electronic), 1998.

[21] E. M. Koper, W. D. Wood, and S. W. Schneider. Aircraft antenna coupling minimization using genetic algorithms and approximations. *IEEE T. Aero. Elec. Sys.*, 40(2):742–751, 2004.

[22] A. P. Krasnojen. Features of creeping waves propagation on the dielectric-coated circular cylinder. *IEE Proc. Microw. Antennas Propag.*, 145(2):179–183, 1998.

[23] K. M. Kuperberg and C. S. Reed. A dynamical system on $\mathbb{R}^3$ with uniformly bounded trajectories and no compact trajectories. In *Proceedings of the American Mathematical Society*, volume 106, pages 1095–1097, 1989.

[24] H. Ling, R. Chou, and S. W. Lee. Shooting and bouncing rays: Calculating the RCS of an arbitrarily shaped cavity. *IEEE T. Antenn. Propag.*, 37:194–205, 1989.

[25] H. Liu, S. Osher, and R. Tsai. Multi-valued solution and level set methods in computational high frequency wave propagation. *Commun. Comput. Phys.*, 1:765–804, 2006.

[26] M. Motamed and O. Runborg. A fast phase space method for computing creeping rays. *J. Comput. Phys.*, 219(1):276–295, 2006.

[27] R. Paknys and D. R. Jackson. The relation between creeping waves, leaky waves, and surface waves. *IEEE T. Antenn. Propag.*, 53(3):898–907, 2005.

[28] R. Paknys and N. Wang. Creeping wave propagation constants and modal impedence for a dielectric coated cylinder. *IEEE T. Antenn. Propag.*, 34(5):674–680, 1986.

[29] J. Perez, J. A. Saiz, O. M. Conde, R. P. Torre, and M. F. Catedra. Analysis of antennas on board arbitrary structures modeled by NURBS surfaces. *IEEE T. Antenn. Propag.*, 45(6):1045–1053, 1997.

[30] J. Qian and W. W. Symes. An adaptive finite-difference method for traveltimes and amplitudes. *Geophysics*, 67(1):167–176, January-February 2002.

[31] O. Runborg. Mathematical models and numerical methods for high frequency waves. *Commun. Comput. Phys.*, 2:827–880, 2007.

[32] J. Shim and H. T. Kim. Dominance of creeping wave modes of backscattered field from a conducting sphere with dielectric coating. *Progress In Electromagnetic Research*, 21:293–306, 1999.

[33] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. ICASE Report 97-65, Brown University, 1997. Prepared for NASA Langley Research Center.

[34] W. W. Symes and J. Qian. A slowness matching Eulerian method for multivalued solutions of eikonal equations. *J. Sci. Comput.*, 19(1-3):501–526, 2003.

[35] G. Taylor. Another look at the line intersection problem. *Int. J. Geogr. Inf. Syst.*, 3(20):192–3, 1989.

[36] J. van Trier and W. W. Symes. Upwind finite-difference calculation of traveltimes. *Geophysics*, 56(6):812–821, June 1991.

[37] J. Vidale. Finite-difference calculation of traveltimes. *B. Seismol. Soc. Am.*, 78(6):2062–2076, December 1988.

[38] V. Vinje, E. Iversen, and H. Gjøystdal. Traveltime and amplitude estimation using wavefront construction. *Geophysics*, 58(8):1157–1166, 1993.

[39] L. Ying and E. J. Candes. Fast geodesics computation with the phase flow method. *J. Comput. Phys.*, 220(1):6–18, 2006.