Phase Space Methods for Computing Creeping Rays

MOHAMMAD MOTAMED

Licentiate Thesis
Stockholm, Sweden 2006
Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i numerisk analys måndagen den 30 oktober 2006 klockan 10.15 i D41, Huvudbyggnaden, Lindstedsvägen 3, Kungl Tekniska högskolan, Stockholm.

© Mohammad Motamed, 2006

Tryck: Universitetsservice US AB
This thesis concerns the numerical simulation of creeping rays and their contribution to high frequency scattering problems.

Creeping rays are a type of diffracted rays which are generated at the shadow line 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, antenna coupling problems, radar cross section (RCS) computations and control of scattering properties of metallic structures coated with dielectric materials.

First, assuming the scatterer surface can be represented by a single parameterization, we propose a new Eulerian formulation for the ray propagation problem by deriving a set of escape partial differential equations in a three-dimensional phase space. The equations are solved on a fixed computational grid using a version of fast marching algorithm. The solution to the equations contain information about all possible creeping rays. This information includes the phase and amplitude of the ray field, which are extracted by a fast post-processing. The advantage of this formulation over the standard Eulerian formulation is that we can compute multivalued solutions corresponding to crossing rays. Moreover, we are able to control the accuracy everywhere on the scatterer surface and suppress the problems with the traditional Lagrangian formulation. To compute all possible creeping rays corresponding to all shadow lines, the algorithm is of computational order $O(N^3 \log N)$, with $N^3$ being the total number of grid points in the computational phase space domain. This is expensive for computing the wave field for only one shadow line, but if the solutions are sought for many shadow lines (for many illumination angles), the phase space method is more efficient than the standard methods such as ray tracing and methods based on the eikonal equation.

Next, we present a modification of the single-patch phase space method to a multiple-patch scheme in order to handle realistic problems containing scatterers with complicated geometries. In such problems, the surface is split into multiple patches where each patch has a well-defined parameterization. The escape equations are solved in each patch, individually. The creeping rays on the scatterer are then computed by connecting all individual solutions through a fast post-processing.

We consider an application to mono-static radar cross section problems where creeping rays from all illumination angles must be computed. The numerical results of the fast phase space method are presented.
Preface

This thesis consists of an introduction and two papers. The author of this thesis contributed to the ideas presented, performed the numerical computations and wrote parts of the manuscript in both papers.


Acknowledgments

First and foremost, I would like to thank my supervisor Olof Runborg for his continuous encouragement and support. His deep understanding of mathematics and ability to simplify difficult problems have been a great help for me. Thank you for giving me the opportunity to work with Heinz Otto Kreiss on hyperbolic problems beside on your excellent idea of computing creeping rays by the fast phase space method.

Heinz, thank you for your strong enthusiasm for mathematics which has been an invaluable source of inspiration for me. I have learned so much from you that goes far beyond science.

I would like to thank my former and present colleagues and friends here at CSC and at IPAM, UCLA and Einstein Institute in Germany. I have had a pleasant time working and being with you. A special thank to Harriett Johansson at CSC for her kindly helps.

Thank to my mother, for all she has given me over the years, and thank you Neda for being always there. I would like to dedicate this thesis to you on your 25th birthday.

Financial support from the Swedish Research Council (VR), the Swedish Foundation for Strategic Research (SSF), NORFA (for the winter school, Norway), NORDITA (for visit at Niels Bohr Institute, Denmark) and IPAM (for a three-month visit at UCLA) are gratefully acknowledged.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High-frequency Scattering Problems</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Time-harmonic Helmholtz Equation</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Asymptotic Approximations</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Creeping Rays</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Importance of Creeping Rays</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Governing Equations</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Existing Numerical Methods for Computing Creeping Rays</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Phase Space Method</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Escape Equations</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Numerical Method for Solving Escape Equations</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Post-Processing</td>
<td>13</td>
</tr>
<tr>
<td>3.4</td>
<td>Multiple Patch Phase Space Method</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Summary of Papers</td>
<td>15</td>
</tr>
<tr>
<td>4.1</td>
<td>Paper 1: A Fast Phase Space Method for Computing Creeping Rays</td>
<td>15</td>
</tr>
<tr>
<td>4.2</td>
<td>Paper 2: A Multiple-Patch Phase Space Method with Application to Creep-ing Rays Computation</td>
<td>15</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>17</td>
</tr>
</tbody>
</table>
Chapter 1

High-frequency Scattering Problems

High-frequency wave scattering problems arise in many situations in nature. For instance, sunlight is scattered by rain drops to form a rainbow. In human vision, our brain constructs a three dimensional map of the world around us from the scattered light which reaches our retinas. Dolphins and bats are able to distinguish an object by listening to scattered sound waves. Scattering problems have a vast number of applications, including radar and sonar technology, seismic tomography, medical imaging and non-destructive testing.

Wave scattering theory, in general, is the study of scattering of an incident wave with some object or inhomogeneity. There are two types of scattering problems:

- **Direct problems** determine the scattered field, given the incident field and the scattering object.

- **Inverse problems** determine properties of the object based on measurements of scattered field for sufficiently many incident fields.

The underlying equation, in the time-dependent formulation of the problem, might be the scalar wave equation, elastic equation or Maxwell’s equations. However, scattering problems are usually stated in a time-independent formulation by taking Fourier transform in time.

Here, we consider the linear scalar wave equation

\[ u_{tt}(t, x) - \frac{1}{n(x)^2} \Delta u(t, x) = 0, \quad (t, x) \in \mathbb{R}^+ \times \mathbb{R}^3, \]  

(1.1)

where \( n(x) \) is the index of refraction. We then formulate it as a time-independent equation in a frequency domain and discuss different numerical approximations.

1.1 Time-harmonic Helmholtz Equation

We are concerned with time-harmonic wave scattering problems in which wave fields are of the form

\[ u(t, x) = v(x) e^{-i\omega t}, \]  

(1.2)

where \( \omega \) is the angular frequency. The wave equation can then be reduced to the Helmholtz equation,

\[ \Delta v(x) + n(x)^2 \omega^2 v(x) = 0, \quad x \in \mathbb{R}^3. \]  

(1.3)

In direct scattering problems with a bounded scatterer \( D \subset \mathbb{R}^3 \), the total field is split into an incident and a scattered field. The scattered field can be formulated as a boundary
value problem in the region outside the scatterer, consisting of the Helmholtz equation, a Dirichlet, Neumann or Robin boundary condition on the boundary of the scatterer $\partial D$ and the Sommerfeld radiation condition at infinity.

When the wave frequency is high and the wavelength is short compared to the size of the computational domain, we encounter a multi-scale problem with a highly oscillatory solution. It is, therefore, a difficult computational problem, and computations are a major challenge. To illustrate this, let the size of the computational domain be $1$ in each dimension, and the shortest wavelength be $\epsilon$. The number of operations to achieve a certain accuracy using direct numerical methods for the Helmholtz equation in $d$ dimensions is then $O((N_\epsilon \epsilon^{-d})^r)$, where $N_\epsilon$ is the number of unknowns per wavelength\(^1\), and $r$ is is the exponent for the number of flops per unknown in the numerical method\(^2\). Even with the best possible numerical methods (if $r = 1$ and $N_\epsilon$ is bounded), the computational complexity will be $O(\epsilon^{-d})$. Therefore, at sufficiently high frequencies ($\epsilon \ll 1$), direct numerical methods based on the full Helmholtz equation are not feasible. This gives a strong motivation for deriving and solving effective high frequency equations with a narrow range of scales, instead of directly solving the full multi-scale problem.

Numerical methods for high frequency scattering problems can in general be classified into three categories:

- **Direct methods** based on integral equations: Given a boundary condition, the problem can be formulated as an integral equation on the boundary. Therefore only the boundary needs to be discretized instead of the whole domain, and the effective dimension is $d - 1$. The accuracy of the solution is then determined by the number of grid points or elements per wavelength, and the computational cost for a fixed accuracy increases with increasing frequency. Standard methods for solving the boundary integral equations include the method of moment \cite{19} and finite element methods \cite{60, 42}. Using a fast iterative solver such as the fast multilevel multipole technique \cite{11, 59, 35}, the complexity of these methods will be $O(\omega^{d-1})$. There are, however, efforts to find robust algorithms of complexity $O(1)$ \cite{17}.

- **Asymptotic methods**: These methods are based on constructing asymptotic expansions of the solution which are valid when $\omega \to \infty$. The accuracy increases with increasing frequency for a fixed computational cost. Most asymptotic techniques rely on geometrical optics equations with frequency independent unknowns. Among other asymptotic methods are wave optical methods (physical optics and physical theory of diffraction) and Gaussian beam methods.

- **Hybrid methods**: They combine direct and asymptotic techniques \cite{32, 18}. Direct methods are applied on the regions where the geometric variations are of the same scale as the wavelength, and asymptotic methods are applied elsewhere. In some cases a linear combination of both methods are used.

In the next section, variants of geometrical optics approximations will be discussed.

### 1.2 Asymptotic Approximations

In order to solve the Helmholtz equation (1.3) for large values of $\omega$, we seek solutions of the form

$$v(x) = a(x, \omega)e^{i\omega \phi(x)}, \quad x \in \mathbb{R}^3.$$

\(^1\) For example, for a standard second order finite difference method $N_\epsilon \approx O(\epsilon^{-1/2})$.

\(^2\) For Gaussian elimination of dense matrices, for instance, $r = 3$.\n
1.2. ASYMPTOTIC APPROXIMATIONS

The phase function \( \phi(x) \) is independent of \( \omega \), and the amplitude function \( a(x, \omega) \) is assumed to be expanded in inverse powers of \( \omega \),

\[
a(x, \omega) \approx \sum_{k=0}^{\infty} a_k(x)(i\omega)^{-k} = \sum_{k=0}^{n} a_k(x)(i\omega)^{-k} + \mathcal{O}(\omega^{-n}).
\]

(1.5)

It means that the series is an asymptotic expansion of \( a \) as \( \omega \to \infty \). Geometrical optics (GO) only considers the leading term of the series \( (k=0) \), which is called the geometrical optics term. Putting (1.4) with the leading term of (1.5) into (1.3) and canceling the phase factor \( e^{i\omega \phi} \), we get

\[
(\nabla \phi)^2 = n(x)^2,
\]

(1.6)

and

\[
2\nabla \phi \cdot \nabla a_0 + a_0 \Delta \phi = 0.
\]

(1.7)

Equation (1.6) is the eikonal equation, which is a first order non-linear partial differential equation (PDE) for \( \phi(x) \). Equation (1.7) is the transport equation, which is a linear PDE with variable coefficients for \( a_0 \), once \( \phi \) is known.

GO can also be formulated in terms of ordinary differential equations (ODEs). We first note that the eikonal equation is a nonlinear Hamilton-Jacobi equation with Hamiltonian

\[
H(x, p) = |p|/n(x) \equiv 1,
\]

where \( p = \nabla \phi \) is the slowness vector. We let \( (x(t), p(t)) \) be a bi-characteristic related to this Hamiltonian. Since \( H \) is constant along them, \( H(x(t), p(t)) = H(x_0, p_0) \), we get the so called ray equations,

\[
\frac{dx}{dt} = \nabla_p H = \frac{1}{n^2} p,
\]

(1.8a)

\[
\frac{dp}{dt} = -\nabla_x H = \frac{\nabla n}{n}.
\]

(1.8b)

There are also ODEs for the amplitude, [13].

There is yet another formulation for GO based on a kinetic viewpoint. Considering rays as trajectories of particles (photons) and introducing the phase space \((t, x, p)\), we note that the evolution of these particles in the phase space is given by the ray equations (1.8). We let \( f(t, x, p) \) be a particle density function. It will then satisfy the Liouville equation,

\[
f_t + \nabla_p H \cdot \nabla_x f - \nabla_x H \cdot \nabla_p f = 0,
\]

(1.9)

where \( \nabla_p H \) and \( \nabla_x H \) are given by (1.8).

There are different numerical techniques based on the three different mathematical models of GO:

1. Numerical methods based on the ray equations (1.8) include ray tracing [9, 22, 30]. In this method the ODEs (1.8) together with the ODEs for the amplitude are solved with standard ODE solvers such as 2nd or 4th order Runge-Kutta methods, giving the phase and amplitude along the rays. The solution at a desired point is then interpolated from the solutions along the rays. This can be rather difficult in the regions where ray tracing produces diverging or crossing rays. Moreover, ray tracing is only of interest for problems involving a few number of source points. For problems with many source points, ray tracing may be computationally expensive.

2. Numerical methods based on the eikonal equation (1.6) are Hamilton-Jacobi methods. They solve the eikonal and transport equations on a uniform Eulerian grid to control the error everywhere. Different types of numerical techniques have been proposed to compute the unique viscosity solution of the eikonal equation, including upwind...
methods of ENO or WENO type [56, 55, 37], fast marching method [54, 45, 46, 41],
group marching method [25] and sweeping method [44, 26, 53]. However, since the
eikonal equation is a nonlinear equation for which the superposition principle does
not hold, these methods fail to capture multivalued solutions corresponding to crossing
rays. Among the methods proposed for computing multivalued solutions are a
domain decomposition based method by detecting kinks [15], big ray tracing [3, 1]
and slowness matching method [49, 50]. The multivalued solutions, in these meth-
ods, are constructed by putting together the solutions of several eikonal equations.
Nevertheless, finding a robust technique to compute multivalued solutions is still a
computational challenge.

3. Numerical methods based on the \textit{kinetic} equation (1.9) are so called phase space
methods. The Liouville equation, like ray equations, benefits from the linear superposition
principle. Moreover, its solution can be computed on a fixed Eulerian grid. There is,
however, a drawback with directly solving the Liouville equation. Because of intro-
ducing the phase space and increasing the number of independent variables, a direct
simulation will computationally be very expensive. There are two different approaches
to overcome this drawback; wave front methods and moment-based methods. In the
former, special wave front solutions are computed, and the later is based on trans-
forming the Liouville equation to a system of conservation law equations for moments
of $f$ in the reduced space $(t, x)$. See, for instance, [31, 7, 12, 43]. The classical wave
front methods include Lagrangian front tracking, wave front construction [57], the
segment projection method [14, 52] and level set method [36]. Related methods are
the fast phase space method [16] and the phase flow method [62].

See [13, 4] for a survey of geometrical optics approximations.

There are two deficiencies in the GO solution described above. First, it does not include
diffraction effects. Secondly, it breaks down at caustics, where $a_0$ is unbounded. In addition
to the incident and reflected rays of GO, new classes of rays, namely \textit{diffracted rays}, should
be introduced to construct the full asymptotic expansion of the solution.

Geometrical theory of diffraction (GTD), developed by J. Keller [23], adds diffracted
rays to GO by adding extra correction terms to the asymptotic solution (1.4-1.5) as,

$$ v(x) = a(x, \omega) e^{i\omega \phi_0(x)} + b(x, \omega) e^{i\omega \phi_d(x)}, \quad b(x, \omega) \approx \sum_{k=0}^{\infty} b_k(x)(i\omega)^{-k - \frac{1}{2}}, $$ (1.10)

where $\phi_d(x)$ and $b_k(x)$ are the phase and amplitudes associated with diffracted rays.

There are various kinds of diffracted rays. One type of diffracted rays is generated
when there is a discontinuity in the scatterer surface $\partial D$, such as edges, tips or changes in
material properties. In Figure 1.1, the incident ray hitting the tip of a wedge generates a
reflected ray, another ray that continues past the tip, and infinitely many diffracted rays
in all directions. Another type of diffracted rays is generated at tangency points of smooth
scatterers, where the incident rays are tangent to the surface. Such grazing rays are called
creeping rays, and if the medium is homogeneous ($n(x) = \text{constant}$), they propagate along
geodesics on the surface. In Figure 1.1, the incident ray hitting the north pole of a perfectly
conducting circular cylinder generates a creeping ray propagating on the cylinder boundary
and shedding diffracted rays along its way. Note that another creeping ray will be generated
by the incident ray hitting the south pole.

In the next chapter, the importance of creeping rays, the governing equations and nu-
dernical methods for computing such rays will be discussed.
Figure 1.1: Diffraction by discontinuous and smooth scatterers. Top figure shows diffraction of an incident field $u_{inc}$ by a wedge. The incident ray hitting the tip of the wedge generates a reflected ray, another ray that continues past the tip, and infinitely many diffracted rays $u_d$ in all directions. Bottom figure shows a creeping ray $u_c$ induced by the incident field $u_{inc}$ at the north pole of a perfectly conducting cylinder, where the incident direction is orthogonal to the surface normal. As the creeping ray propagates on the boundary, it continuously emits surface-diffracted rays $u_d$ with exponentially decreasing initial amplitude.
Chapter 2

Creeping Rays

Creeping rays are a type of diffracted rays which are generated at the shadow line\(^1\) 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. See Figure 2.1.

Figure 2.1: Creeping rays on smooth surfaces. Left figure shows a creeping ray on a convex surface generated at the tangency point. For a perfectly conducting scatterer, it decays exponentially by tangentially shedding diffracted rays as it propagates on the surface. Right figure shows a creeping ray on a concave surface propagating without attenuation.

2.1 Importance of Creeping Rays

The study of creeping rays is important in many high frequency problems, such as design of sophisticated and conformal antennas [24], antenna coupling problems [28], radar cross section (RCS) computations [5, 24, 47, 33] and control of scattering properties of metallic structures coated with dielectric materials [39, 2, 29, 38].

In antenna design, the creeping ray mechanism frequently provides the dominant coupling. It is therefore a need to address the creeping ray mechanism in antenna coupling analysis [28, 21].

In RCS computations, the contribution of direct reflected rays usually dominates the one arising from creeping rays, due to the exponential attenuation of creeping rays. However, there are cases where creeping rays can give important contributions, including

\(^1\)Shadow line or horizon is the locus of the points at which the incident rays are tangent to the scatterer surface.
1. Low observable objects for which the creeping ray contributions are dominant [5, 24]: Modern military aircrafts, for instance, demand low observables over a full range of angles necessitating the inclusion of creeping ray effects in any useful analysis technique. Such analysis can then be used to design radar absorbing material coatings and place them on the critical regions on the aircraft in order to suppress the creeping ray effects.

2. Concave scatterers on which creeping rays do not attenuate: In the absence of attenuation, creeping ray effects can be as important as direct reflection effects.

3. Convex scatterers coated with lossless dielectric materials [39, 47]: On such surfaces, depending on the thickness of coating, both leaky (highly attenuated) and trapped (low attenuated) creeping modes can be generated. It is then important to consider the trapped creeping modes in the analysis.

4. Caustic creeping rays [33]: Even in the presence of attenuation, creeping rays can give an important contribution to RCS, depending on their geometrical spreading. A caustic creeping ray, which has a zero geometrical spreading at its detachment point, has infinite amplitude. It can particularly be important in near-field RCS investigations. Caustic creeping rays can also happen at infinity and therefore be important for far-field RCS computations. For example for a sphere, there are infinitely many backscattered creeping rays forming a caustic at infinity.

5. Creeping ray-reflection interaction: For a non-convex object, an incoming creeping ray, generated on the convex part of the surface, can excite whispering gallery modes close to the concave part. The whispering gallery wave field, concentrated near to the surface, does not exhibit exponential decay along geodesics and can then be of special interest to the aerospace industry [10]. Creeping ray-reflection interaction can also be important for multiple-scattering problems, where a creeping ray, generated on one scatterer, sheds diffracted rays to another scatterer, which in turn can be reflected back to the receiver. It can be more crucial if the incoming creeping ray is a caustic creeping ray. See Figure 2.2.

We will now briefly review the governing equations for computing creeping rays.

### 2.2 Governing Equations

In order to compute creeping rays, one needs to compute the geodesics and the associated surface wave field.

We consider a hyper-surface with a regular explicit parameterization $\bar{X} = \bar{X}(u)$, where $\bar{X} = (x, y, z) \in \mathbb{R}^3$, and the parameters $u = (u, v)$ belong to a bounded set $\Omega \subset \mathbb{R}^2$. A wave field, associated to a creeping ray, is generated on the surface

\[ v_s(u) = a(u)e^{i\omega\phi(u)}, \tag{2.1} \]

where $\phi(u)$ and $a(u)$ are surface phase and amplitude. The creeping rays are now related to (2.1) in the same way as the standard GO rays are related to (1.4).

A geodesic is uniquely characterized by its location and direction on the surface and is given by a system of three first-order ODEs [33],

\[ \dot{\gamma} = g(\gamma), \quad \gamma = (u, \theta). \tag{2.2} \]
2.2. GOVERNING EQUATIONS

Figure 2.2: Creeping ray-reflection interaction. Left figure shows the interaction for a non-convex object. An incoming creeping ray, generated on the convex part of the surface excites whispering gallery modes close to the concave part. Right figure shows the interaction for multiple scatterers. A creeping ray, generated on one scatterer, sheds diffracted rays to another scatterer, which are in turn reflected back to the receiver. It can be more crucial if the incoming creeping ray is a caustic creeping ray.

Here, dot denotes differentiation with respect to arc length along the geodesic in the physical space, \( \tau \), and \( g(\gamma) \) is a function depending on the surface parametric equation and its derivatives.

Moreover, the phase \( \phi \), which is the length of the ray, satisfies the ODE

\[
d\frac{\phi(u(\tau))}{d\tau} = 1, \quad \phi(0) = \phi_0(u_0). \tag{2.3}
\]

and the amplitude \( a \) is computed by,

\[
a(\tau) = a_0 Q(s, \tau) \frac{1}{\rho_g} \exp \left( -\omega^{1/3} \beta(\tau) \right), \tag{2.4}
\]

where \( a_0 \) is the amplitude at the attachment point on the shadow, \( s \) is the arc length parameterization along the shadow line, \( Q(s, \tau) \) is the geometrical spreading at distance \( \tau \) from the attachment point, and \( \beta(\tau) \) is a function representing the attenuation given by,

\[
\beta(\tau) = \int_0^\tau \tilde{\alpha}(\gamma(r)) dr, \quad \tilde{\alpha} = \frac{\alpha_0}{\rho_0} \exp \left( \frac{\pi}{6} \left( \frac{\rho_g}{2} \right)^{1/3} \right), \quad q_0 \approx 2.33811. \tag{2.5}
\]

Here \( \rho_0(\gamma) \) is the radius of curvature of the surface along the ray trajectory.

We note that these ODEs are analogous to the ray equations in standard GO, and we refer to them as surface ray equations. There are also surface eikonal and surface transport equations analogous to the eikonal and transport equations of GO. The surface eikonal equation is, for example, given by,

\[
|\tilde{\nabla} \phi| = n, \quad \tilde{\nabla} \phi := J G^{-1} \nabla \phi, \quad G = J^T J, \quad J = [\tilde{X}_u \tilde{X}_v] \in \mathbb{R}^{3 \times 2}. \tag{2.6}
\]
As in standard GO, depending on the formulation, there are different numerical techniques for computing creeping rays.

2.3 Existing Numerical Methods for Computing Creeping Rays

There are two different numerical techniques for computing creeping rays:

1. Lagrangian techniques based on surface ray equations. The simplest and most common method to solve these ODEs is standard ray tracing [21, 40]. It gives the surface phase and amplitude solutions along creeping rays. Interpolation must then be applied to obtain the solution everywhere. But, in regions where rays cross or diverge this can be rather difficult. However, the interpolation can be simplified by using wave front methods [58, 18] in which, instead of individual rays, an interface representing a wave front is evolved. Nevertheless, for some problems, such as RCS where creeping rays from all illumination angles must be computed, Lagrangian methods can be computationally expensive.

2. Eulerian techniques based on surface eikonal and surface transport equations. These PDEs are discretized on fixed computational grids, and there is no problem with interpolation [27]. However, these equations only give the correct solution when it is a single wave. Therefore, in the case of crossing waves, more elaborate schemes must be devised to capture multivalued solutions.

The third family of numerical methods for standard GO, discussed in Section 1.2, have so far not been used for the creeping ray case. Inspired by numerical methods based on the kinetic formulation of GO, we formulate the creeping ray equations in a phase space and present a new technique for computing creeping rays which does not suffer from the drawbacks of the existing methods.

\[^{2}\text{Recently, Ying and Candes have modified the phase flow method for geodesics computation [61].}\]
Chapter 3

Phase Space Method

We present a new scheme for computing creeping rays. The method is an adaptation of a similar scheme for standard geometrical optics by Fomel and Sethian [16]. We note that this technique is not only useful for creeping ray computations, but also for many other applications such as inverse problems in seismology and shadow line computations. The same construction can be made for these applications as soon as there are a set of ODEs describing the problem.

3.1 Escape Equations

We introduce the phase space $\mathbb{P} = \mathbb{R}^2 \times S$, where $S$ is the periodic sphere. We consider the triplet $\gamma = (u, v, \theta)$ as a point in this space. The geodesics on the scatterer are then confined to a subdomain $\Omega_p = \Omega \times S \subset \mathbb{P}$ in phase space.

Now consider the geodesic starting at $\gamma = (u, v, \theta) \in \Omega_p$ and ending at the boundary of $\Omega_p$. We call the end point escape point, where the geodesic crosses the boundary and wants to escape from the domain. See Figure 3.1.

We define the following functions for this geodesic:

- $F : \mathbb{P} \to \mathbb{P}$, $F(\gamma) = (U, V, \Theta)$ is the escape location and direction.
• $\Phi : \mathbb{P} \rightarrow \mathbb{R}$ is the length of geodesic.

• $B : \mathbb{P} \rightarrow \mathbb{R}$ is the difference between the $\beta$-values at the escape and starting points.

In order to derive equations for these functions, we notice that $F$ is constant for all $\gamma(\tau)$ along the geodesic, and therefore we have

$$\frac{d}{d\tau} F(\gamma(\tau)) = 0 \Rightarrow g_1 F_u + g_2 F_v + g_3 F_\theta = 0, \quad \gamma \in \Omega_p, \quad (3.1)$$

with the boundary condition at inflow points\(^1\), i.e., the points on $\partial \Omega_p$ at which geodesics are out-going,

$$F(\gamma) = \gamma, \quad \gamma \in \partial \Omega_p^{\text{inflow}}.$$  

Here the coefficients $g = (g_1, g_2, g_3)^\top$ in (3.1) are known and given by (2.2).

Moreover, by (2.3) and (2.5), we can write the equations for $\Phi$ and $B$,

$$\frac{d}{d\tau} \Phi(\gamma(\tau)) = 1 \Rightarrow g_1 \Phi_u + g_2 \Phi_v + g_3 \Phi_\theta = 1, \quad \gamma \in \Omega_p, \quad (3.2)$$

with the boundary condition at inflow points

$$\Phi(\gamma) = 0, \quad \gamma \in \partial \Omega_p^{\text{inflow}},$$

and

$$\frac{d}{d\tau} B(\gamma(\tau)) = \tilde{\alpha}(\gamma) \Rightarrow g_1 B_u + g_2 B_v + g_3 B_\theta = 1, \quad \gamma \in \Omega_p, \quad (3.3)$$

with the boundary condition at inflow points

$$B(\gamma) = 0, \quad \gamma \in \partial \Omega_p^{\text{inflow}}.$$  

The escape equations (3.1)-(3.3) are linear hyperbolic PDEs in the three dimensional phase space. Note that to compute the amplitude, in addition to the attenuation function $B$, we also need to compute geometrical spreading, which has been discussed in detail in [33, 34].

### 3.2 Numerical Method for Solving Escape Equations

To solve the escape equations (3.1)-(3.3), we use a version of fast marching algorithm given by Fomel and Sethian [16]. One important characteristic of the solutions to the escape PDEs is discontinuity due to discontinuous boundary conditions. Therefore it is important to use suitable interpolation techniques to avoid the unphysical Gibbs oscillations. For a second order accurate fast marching algorithm, we then use a version of two dimensional essentially non-oscillatory (ENO) interpolation based on Newton divided differences and the Newton formulation of the interpolation polynomial, see [48]. For a first order accurate algorithm, we can simply use linear interpolations. The algorithm is a one-pass algorithm and is of complexity $O(N^3 \log N)$, with $N$ being the number of grid points in each space of $\Omega_p$. See [34] for a complete discussion on the algorithm.

Note that there are other methods for solving the escape equations. One is to discretize the PDEs in the phase space using a finite difference or finite volume approxiamtion and arrive at a system of linear equations $Af = b$, where $A$ is a $N^3 \times N^3$ matrix with

\(^1\)Note that inflowing characteristics correspond to out-going geodesics.
3.3. POST-PROCESSING

a sparse structure and $b$ represents the boundary conditions. This system can then be solved iteratively, and one can speed up the computations using suitable preconditioners \cite{20, 6}. 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.

3.3 Post-Processing

The escape PDEs solutions contain information about all possible creeping rays in all directions. To extract properties like phase and amplitude for a ray family, post-processing of the solution is needed.

First, we note that $F(u_1, v_1, \theta_1) = F(u_2, v_2, \theta_2)$ implies that $(u_1, v_1, \theta_1)$ and $(u_2, v_2, \theta_2)$ lie on the same geodesic. See Figure 3.2.

![Figure 3.2: If $\gamma_1 = (u_1, v_1, \theta_1)$ and $\gamma_2 = (u_2, v_2, \theta_2)$ lie on the same geodesic, then $F(\gamma_1) = F(\gamma_2)$ and vice versa.](image)

Suppose we want to compute the surface phase at a point on the illuminated scatterer. We assume that the shadow line $\gamma_0(s)$ is known. For each point $u \in \Omega$ covered by the surface wave, there is at least one creeping ray, starting at the shadow line, which passes through it. We can thus find $s = s^*(u)$ and phase angle $\theta = \theta^*(u)$, as the solution to

$$F(\gamma_0(s)) = F(u, \theta). \quad (3.4)$$

The phase at $u$ is then given by

$$\phi(u) = \phi_0(u_0(s^*)) + \Phi(\gamma_0(s^*)) - \Phi(\gamma^*), \quad \gamma^* = (u, \theta^*) \in \mathbb{L}(\gamma_0),$$

where $\mathbb{L}(\gamma_0)$ is a sub-manifold of phase space $\mathbb{P}$ on which the creeping rays generated at $\gamma_0(s)$ lie. Note that there may be multiple solutions $(s^*, \theta^*)$ to (3.4), giving multiple phases. Using the same technique, the amplitude can also be computed.
CHAPTER 3. PHASE SPACE METHOD

To solve (3.4), we reduce \( F = (U, V, \Theta) \in \partial \Omega_p \), to a point \((S, \Theta)\) in \( \mathbb{R}^2 \). The left and right hand sides of (3.4) are then curves in \( \mathbb{R}^2 \) parameterized by \( s \) and \( \theta \). Therefore, solving the algebraic equation (3.4) amounts to finding crossing points of these curves. This can be done with a complexity of \( O(N) \), assuming the shadow line is discretized in \( N \) grid points. See e.g. [51]. The post-processing for all \( N^2 \) points on the surface can be done at a cost of \( O(N^3) \). The total complexity, including solving the escape PDEs, will therefore be \( O(N^3 \log N) \). This is expensive for computing the field for only one shadow line. For example by using wave front tracking or solvers based on the surface eikonal equation, the complexity is \( O(N^2) \). However, if the solutions are sought for many shadow lines, the phase space method is more efficient.

3.4 Multiple Patch Phase Space Method

Most scatterer surfaces with complex geometries, such as an aircraft, cannot be represented by a single non-singular explicit parameterization. One way to handle such realistic problems is to split 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 order to modify the single-patch phase space method to these cases, we consider a multiple-patch surface represented by different parameterizations. The PDEs are solved in every patch, individually. The creeping rays on the scatterer are then computed by connecting all individual solutions through a fast post-processing.

Multiple-patch (or multi-block) finite difference schemes are a sub-class of domain decomposition (DD) methods for solving PDEs by iteratively solving sub-problems on smaller sub-domains [8]. However, the scheme presented here is not based on iterations. See [34] for a detailed discussion on the scheme together with numerical examples.
Chapter 4

Summary of Papers


In Paper 1 we consider creeping ray contributions to high frequency scattering problems. We assume that the scatterer surface can be represented by a single parameterization and present a new Eulerian formulation for the problem. Following the discussions in Section 3, we derive a set of escape partial differential equations in a three-dimensional phase space. The equations are then solved on a fixed computational grid using a version of first-order accurate fast marching algorithm. The solution to the escape equations contain information about all possible creeping rays. This information includes the phase and amplitude of the ray field, which are extracted by a fast post-processing.

We consider an application to mono-static radar cross section problems where creeping rays from all illumination angles must be computed and present the numerical results of the fast phase space method.

4.2 Paper 2: A Multiple-Patch Phase Space Method with Application to Creeping Rays Computation.

In Paper 2 we consider surfaces that cannot be represented by one non-singular parameterization. We modify the single-patch phase space method to this case in order to handle realistic problems containing scatterers with complicated geometries. We split the surface into multiple patches where each patch has a well-defined parameterization. The escape equations are solved independently in each patch using a second-order accurate fast marching method. The creeping rays on the scatterer are then computed by connecting all individual solutions through a fast post-processing.

We present an application of the method and sample numerical results from a prototype implementation of this scheme.
Bibliography


Paper I
A fast phase space method for computing creeping rays

Mohammad Motamed *, Olof Runborg

Department of Numerical Analysis and Computer Science, Royal Institute of Technology (KTH), Lindstadsvagen 3, 10044 Stockholm, Sweden

Received 17 October 2005; received in revised form 20 March 2006; accepted 21 March 2006

Abstract

Creeping rays can give an important contribution to the solution of medium to high frequency scattering problems. They are generated at the shadow lines of the illuminated scatterer by grazing incident rays and propagate along geodesics on the scatterer surface, continuously shedding diffracted rays in their tangential direction.

In this paper, we show how the ray propagation problem can be formulated as a partial differential equation (PDE) in a three-dimensional phase space. To solve the PDE we use a fast marching method. The PDE solution contains information about all possible creeping rays. This information includes the phase and amplitude of the field, which are extracted by a fast post-processing. Computationally, the cost of solving the PDE is less than tracing all rays individually by solving a system of ordinary differential equations.

We consider an application to mono-static radar cross section problems where creeping rays from all illumination angles must be computed. The numerical results of the fast phase space method and a comparison with the results of ray tracing are presented.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Creeping rays; High frequency wave propagation; Scattering problems; Numerical methods; Geometrical theory of diffraction; Eikonal equation

1. Introduction

The general problem that we are interested in is the scattering of a time-harmonic incident field by a bounded scatterer $D$. If the total field is split into an incident and a scattered field, this can be formulated as a boundary value problem for the scattered field in the region outside $D$, consisting of the Helmholtz equation,

$$\Delta W + n(x)^2 \omega^2 W = 0, \quad x \in \mathbb{R}^3 \setminus D,$$

augmented with Dirichlet, Neumann or Robin boundary conditions on the boundary of the scatterer $\partial D$, and the Sommerfeld radiation condition at infinity. Here $n(x)$ is the index of refraction, and $\omega$ is the angular frequency.

* Corresponding author.
E-mail addresses: mohamad@nada.kth.se (M. Motamed), olofr@nada.kth.se (O. Runborg).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.03.024
In direct numerical simulations of (1) the accuracy of the solution is determined by the number of grid points or elements per wave length. The computational cost to maintain constant accuracy grows algebraically with the frequency, and for sufficiently high frequencies, a direct numerical simulation is no longer feasible. Numerical methods based on approximations of (1) are needed.

Fortunately, there exist good such approximations precisely for the difficult case of high frequency solutions. In free space, a typical high frequency solution can be approximated by a simple wave,

\[ W(x) \approx a(x)e^{i\omega(x)}, \quad x \in \mathbb{R}^3, \]  

(2)

where the amplitude \( a(x) \) and the phase function \( \phi(x) \) depend only mildly on the parameter \( \omega \) and vary on a much coarser scale than \( W(x) \) itself. Geometrical optics (GO) considers the case when \( \omega \to \infty \). The frequency then disappears from the model and the equations can be solved at a computational cost independent of \( \omega \). GO can be formulated as the partial differential equations for \( \phi \) and \( a \). The phase function \( \phi \) satisfies the eikonal equation,

\[ |\nabla \phi| = n(x), \]  

(3)

and the leading order amplitude term \( a \) satisfies the transport equation,

\[ 2\nabla \phi \cdot \nabla a + \Delta \phi a = 0. \]  

(4)

GO can also be formulated in terms of ordinary differential equations (ODE). It corresponds to solving the eikonal equation (3) through the method of characteristics, i.e. solving the system of ODEs,

\[ \frac{dx}{dt} = \nabla_p H(x, p), \quad \frac{dp}{dt} = -\nabla_x H(x, p), \quad H(x, p) = \frac{|p|}{n(x)}, \]  

(5)

where \( t \) is time. As long as \( \phi \) is smooth, the relationship between the models is given by \( \phi(x(t)) = \phi(x(0)) + t \). There are also ODEs giving the amplitude \( a(x(t)) \) along the characteristics.

The main drawbacks of the infinite frequency approximation of geometrical optics are that diffraction effects at boundaries are lost, and that the approximation breaks down at caustics, where the predicted amplitude \( a \) is unbounded. Geometrical theory of diffraction (GTD), pioneered by J. Keller in the 1950s [14], adds diffraction effects to the GO approximations. One type of diffracted rays are creeping rays, which are generated at the shadow line of the scatterer, i.e. where the incident ray strikes the surface of the scatterer at grazing angle. At this point the incident ray divides into two parts: one part continues straight on, and a second part propagates along geodesics on the surface, continuously shedding diffracted rays in its tangential direction. See Fig. 1. In analogy with (2), a wave field is generated on the surface

\[ W_s(u) = a(u)e^{i\phi(u)}, \]  

(6)

where \( \phi(u) \) and \( a(u) \) are now the surface phase and amplitude and \( u \in \mathbb{R}^2 \) is a parameterization of the surface. The creeping rays satisfy a system of ODEs similar to (5). They are related to (6) in the same way as the standard GO rays are related to (2).

Creeping rays can give an important contribution to the solution at medium to high frequencies, for instance in radar cross section (RCS) computations for low observable objects [3] and in antenna coupling problems [16]. We want to compute the creeping rays and the associated wave field in (6).

Various methods have been devised to compute the geometrical optics solution. They can be divided into Lagrangian and Eulerian methods.

Lagrangian methods are based on the ODE formulation (5). The simplest Lagrangian method is standard ray tracing where the ODEs in (5) together with ODEs for the amplitude are solved directly with numerical methods for ODEs. This approach is very common in standard free space GO, [4,19], but is also done for the creeping ray case, [12,22]. Ray tracing gives the phase and amplitude solution along a ray, and interpolation must be applied to obtain those quantities everywhere. This can be rather difficult, in particular in regions where rays cross. Another problem with ray tracing is that it may produce diverging rays that fail to cover the domain. Even for smooth \( n(x) \) there may be shadow zones where the field is hard to resolve. The interpolation can be simplified by instead using so-called wave front methods [30,11]. They are related to ray tracing, but instead of individual rays, an interface representing a wave front is evolved according to the ray equations.
More recently, Eulerian methods based on PDEs have been proposed to avoid some of the drawbacks of ray tracing. These methods discretize the PDEs on fixed computational grids to control accuracy everywhere and there is no need for interpolation. The simplest Eulerian methods solves the eikonal and transport equations \((3,4)\). This technique has been used in standard GO, \([29,28,7]\) and also in the surface case, \([15]\). However, the eikonal and transport equations only give the correct solution when it is a single wave of the form \((2)\). When there are crossing waves, more elaborate schemes must be devised. In the free space GO case a number of methods have been developed in the last ten years using different approaches. Many of them are based on a third formulation of geometrical optics as a kinetic equation set in phase space. They include “big” ray tracing \([1]\), patching together multiple eikonal solutions \([2]\), moment methods \([24,25,9]\), segment projection method \([6]\), level set methods \([21,23]\), slowness matching \([26]\), the phase flow method \([31]\) and fast phase space methods \([8]\). A survey of this research effort is given in \([5]\).

These more advanced methods have so far not been used for the creeping ray case. In this paper we propose an adaptation of the fast phase space method of Fomel and Sethian \([8]\) to this case. This method is computationally expensive if only a few solutions are computed. It becomes attractive when the solution is sought for many different sources but with the same index of refraction. In the creeping ray case this happens for instance when the solution for all illumination angles of a fixed scatterer is of interest. We consider one such example: computing the mono-static RCS.

Fig. 1. Diffraction by a smooth cylinder. Top figure shows the solution schematically. The incident field \(u_{\text{inc}}\) induces a creeping ray \(u_c\) at the north (and south) pole of the cylinder, where the incident direction is orthogonal to the surface normal. As the creeping ray propagates along the surface, it continuously emits surface-diffracted rays \(u_d\) with exponentially decreasing initial amplitude. Bottom figure shows real part of a solution to the Helmholtz equation. The surface diffracted waves can be seen behind the cylinder.
Following [8] we formulate the ray propagation problem as a time-independent partial differential equation (PDE) in a three-dimensional phase space. We use a fast marching method to solve the PDE. The PDE solution contains information for all incidence angles. The phase and amplitude of the field are extracted by a fast post-processing. Computationally the cost of solving the PDE is less than tracing all rays individually. If the surface is discretized by \( N^2 \) points the complexity is \( \mathcal{O}(N^3 \log N) \), while ray tracing would cost \( \mathcal{O}(N^4) \) if a comparable number of incidence angles \( (N^2) \) and rays per angle \( (N) \) are considered.

In Section 2, we formulate the governing equations. The numerical method for solving the equations are discussed in Section 3. In Section 4, we show how to extract the information for a particular ray through post-processing. An application to a mono-static RCS problem is shown as an example in Section 5.

2. Governing equations

For simplicity we consider the case when the scatterer surface has an explicit parameterization. Let \( \mathbf{x} \) be a regular hypersurface, representing a scatterer surface, with the parametric equations \( \mathbf{x} = \mathbf{x}(\mathbf{u}) \), where \( \mathbf{x} = (x, y, z) \in \mathbb{R}^3 \) is the coordinate in 3D physical space, and the parameters \( \mathbf{u} = (u, v) \) belong to a bounded set \( \Omega \subset \mathbb{R}^2 \). Let the scatterer be illuminated by incident rays in a direction represented by a normalized vector \( \mathbf{I} = [t_1, t_2, t_3] \). The shadow line is then defined as the set of points where

\[
\hat{\mathbf{n}}^\top \mathbf{I} = 0, \tag{7}
\]

where \( \hat{\mathbf{n}}(\mathbf{u}) \) is the surface normal at \( \mathbf{x}(\mathbf{u}) \),

\[
\hat{\mathbf{n}} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|}. \tag{8}
\]

Here the subscripts denote differentiation with respect to \( u \) and \( v \). We will assume that (7) defines a curve in parameter space, which we denote \( \mathbf{u}_0(s) \), and \( s \) is the arc length parameterization.

2.1. Geodesics

We start by deriving the equations for creeping rays, which are indeed geodesics on the scatterer surface. According to Keller and Lewis [13], the surface phase satisfies the surface eikonal equation,

\[
|\nabla \phi| = n,
\]

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

\[
\nabla \phi := J G^{-1} \nabla \phi, \quad G = J^\top J,
\]

with

\[
J = [\mathbf{x}_u \mathbf{x}_v] \in \mathbb{R}^{3 \times 2}.
\]

We prescribe boundary conditions for (9) on the shadow line, which acts as the source for the creeping rays. The boundary condition is that the surface phase agrees with \( \phi_{inc} \), the phase of the incoming wave,

\[
\phi(\mathbf{u}_0(s)) = \phi_0(\mathbf{u}_0) := \phi_{inc}(\mathbf{x}(\mathbf{u}_0(s))), \tag{10}
\]

To avoid ambiguities as to which direction the surface waves propagate, we add the condition

\[
\nabla \phi(\mathbf{u}_0(s)) = \nabla \phi_{inc}(\mathbf{x}(\mathbf{u}_0(s))), \tag{11}
\]

which is consistent with (9) since \( \phi_{inc} \) satisfies the free space eikonal equation (3) and with (10) since

\[
\frac{d}{ds} (\phi(\mathbf{u}_0(s)) - \phi_{inc}(\mathbf{x}(\mathbf{u}_0(s)))) = \nabla \phi^\top \mathbf{u}_0' - \nabla \phi_{inc}^\top \frac{d\mathbf{x}}{ds} = (J^\top \nabla \phi)^\top \mathbf{u}_0' - \nabla \phi_{inc}^\top \frac{d\mathbf{x}}{ds} = (\nabla \phi - \nabla \phi_{inc})^\top \frac{d\mathbf{x}}{ds}.
\]

In the case when \( n = 1 \) and the incoming wave is a plane wave in direction \( \mathbf{I} \), we have \( \phi_{inc}(\mathbf{x}) = \mathbf{I}^\top \mathbf{x} \). Then (10), (11) reduce to
\[ \phi_0(u_0(s)) := \tilde{T} \chi(u_0(s)), \quad \tilde{\nabla} \phi(u_0(s)) = \tilde{I}. \]  
(12)

We can write (9) as a Hamilton–Jacobi equation \( H(u, \nabla \phi) = 0 \), with the Hamiltonian

\[ H(u, p) = \frac{1}{2} p^T G^{-1}(u)p - \frac{n^2(u)}{2}. \]

Note that in the case \( n = \text{constant} \), the geometrical rays associated with the eikonal equation (3) becomes straight lines. Analogously, for the surface eikonal equation (9), the creeping rays for constant \( n \) are geodesics, or shortest paths between two points on the surface. Henceforth, we will assume \( n = 1 \) and a plane incoming wave.

Introducing a parameter \( \tau \), the bicharacteristics \((u(\tau), p(\tau))\) are determined by the solution of the following Hamiltonian equations

\[ \dot{u} = H_p = G^{-1} p, \quad \dot{p} = -H_u. \]  
(13a, 13b)

Here the dot denotes differentiation with respect to the parameter \( \tau \). At the shadow line, the initial direction of the geodesic should be parallel to the incident field. We demand that

\[ J^+ G^{-1} \nabla \phi(u_0(s)) = J^+ \tilde{I} = J^+ \tilde{I}. \]

This implies that

\[ p(0) = G(u_0) = J^T J \dot{u}(0) = J^T \chi(0) = J^T \tilde{I}. \]

The initial condition for the system (13) therefore reads,

\[ u(0) = u_0(s), \quad p(0) = p_0(s) := J^T(u_0(s))\tilde{I}. \]  
(14a, 14b)

We note that by (12),

\[ p(0) = J^T(u_0(s)) \tilde{\nabla} \phi(u_0(s)) = J^T J G^{-1} \nabla \phi(u_0(s)) = \nabla \phi(u(0)). \]

As for any Hamiltonian system it therefore follows that

\[ p(\tau) = \nabla \phi(u(\tau)), \]  
(15)

for all \( \tau > 0 \), as long as \( \phi \) is smooth. As a consequence, (13) and (15) give

\[ |\tilde{\chi}| = \left| \frac{d \chi}{d \tau} \right| = |J\dot{u}| = |JH_p| = |JG^{-1}p| = 1, \]  
(16)

and we can identify the parameter \( \tau \) with arc length along the creeping rays \( \chi(u(\tau)) \). In this case, the system of four first-order ODEs (13) can be written as a system of two second-order equations [13],

\[ \ddot{u} + \Gamma^1_{11} \dot{u}^2 + 2 \Gamma^1_{12} \dot{u} \dot{v} + \Gamma^1_{22} \dot{v}^2 = 0, \]  
\[ \ddot{v} + \Gamma^2_{11} \dot{u}^2 + 2 \Gamma^2_{12} \dot{u} \dot{v} + \Gamma^2_{22} \dot{v}^2 = 0. \]  
(17a, 17b)

Here \( \Gamma^k_{ij} \) are Christoffel symbols, defined by

\[ \Gamma^k_{ij} = \sum_{m=1}^2 \frac{1}{2} g^{km} [(g_{mj})_i + (g_{mi})_j - (g_{ij})_m], \]

where \( (g_{ij}) = G \) and \( (g^{ij}) = G^{-1} \), and subscripts 1 and 2 denote differentiation with respect to \( u \) and \( v \), respectively.

Now if we set \( \dot{u} = \frac{du}{d\tau} = \rho \cos \theta \) and \( \dot{v} = \frac{dv}{d\tau} = \rho \sin \theta \), then \( \ddot{v} = \dot{u} \tan \theta \), and by differentiating with respect to \( \tau \),

\[ \ddot{v} = \dot{u} \tan \theta + \dot{u} \frac{1}{\cos^2 \theta} \dot{\theta}. \]  
(18)

Moreover by (16),
\[
\rho = \rho(u, v, \theta) = \left| \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \right|^{-1} = |X_u \cos \theta + X_v \sin \theta|^{-1}.
\]

Let \( \gamma := (u, v, \theta) \). Using (18), we get
\[
\dot{\theta} = \rho(\gamma) \mathcal{Z}(\gamma),
\]
where
\[
\mathcal{Z}(\gamma) := (\Gamma_{11}^1 \cos^2 \theta + 2\Gamma_{12}^1 \cos \theta \sin \theta + \Gamma_{22}^1 \sin^2 \theta) \sin \theta - (\Gamma_{11}^2 \cos^2 \theta + 2\Gamma_{12}^2 \cos \theta \sin \theta + \Gamma_{22}^2 \sin^2 \theta) \cos \theta.
\]

Therefore the system of ODEs (17), for geodesics, reduces to
\[
\begin{pmatrix}
\dot{u} \\
\dot{v} \\
\dot{\theta}
\end{pmatrix} = \begin{pmatrix}
\rho(\gamma) \cos \theta \\
\rho(\gamma) \sin \theta \\
\rho(\gamma) \mathcal{Z}(\gamma)
\end{pmatrix} =: \mathbf{g}(\gamma). \tag{19}
\]

2.2. Phase and amplitude

Let us now derive the ODEs for the surface phase \( \phi \) and amplitude \( a \). As before, we parametrize the creeping ray with the arc length \( \tau \) in the physical space. In the surface field associated with the creeping ray (6), the phase function \( \phi(u(\tau)) \) and the amplitude \( a(u(\tau)) \) of the field vary with the distance \( \tau \) along the ray.

From (13) and (15) it follows that the phase of the geodesic satisfies the ODE,
\[
\frac{d\phi(u(\tau))}{d\tau} = \nabla \phi \cdot \mathbf{u} = \nabla \phi \cdot G^{-1} \nabla \phi = |\nabla \phi|^2 = 1, \quad \phi(0) = \phi_0(u_0). \tag{20}
\]

Hence, the phase is the length of the ray.

Now consider a narrow strip of a creeping ray, starting at the incident point \( Q_0 \) on the shadow line and propagating along a geodesic on the scatterer surface. See Fig. 2.

To determine an equation for the amplitude, we apply the optical form of energy conservation principle in a small interval from \( \tau \) to \( \tau + d\tau \), [18], and get
\[
\frac{d}{d\tau} [a(\tau)^2 d\sigma(\tau)] = -2z(\tau) [a(\tau)^2 d\sigma(\tau)], \tag{21}
\]
where \( d\sigma(\tau) \) is the width of the strip at distance \( \tau \) from \( Q_0 \), and \( z(\tau) \) is an attenuation factor. Solving (21) gives us
\[
a(\tau) = a_0 \left( \frac{d\sigma_0}{d\tau} \right)^{\frac{1}{2}} \exp \left( - \int_0^\tau z(r) dr \right), \tag{22}
\]
where \( a_0 \) and \( d\sigma_0 \) are the amplitude and strip width at \( Q_0 \), respectively. There are thus two parts in this equation which we can treat separately: the attenuation, represented by the exponential, and the geometrical spreading of the creeping ray, represented by \( \frac{d\sigma}{d\tau} \).

Fig. 2. A narrow strip of a creeping ray on the surface.
2.2.1. Attenuation

We will here show that the attenuation can be obtained by solving an ODE coupled to the geodesic system (19).

The attenuation factor $\alpha$ is given by [18,20],

$$\alpha = \frac{q_0}{\rho_\phi} \exp \left( \frac{\pi}{6} \left( \frac{\omega \rho_\phi}{2} \right)^{1/3} \right) := \omega^{1/3} \tilde{\alpha}.$$ 

Here $q_0 \approx 2.33811$ is the smallest positive zero of the Airy function, and $\rho_\phi$ is the radius of curvature of the surface with respect to arc length along the ray trajectory, given by [10],

$$\rho_\phi = \frac{1}{\tilde{T} \cdot D_\phi \tilde{N} \tilde{u}}, \quad \tilde{T} = \frac{d\tilde{X}}{d\tau}(u(\tau)) = \tilde{J} \tilde{u}.$$ 

Here, $\tilde{T}$ is the tangent vector to the surface in the geodesics direction, and $D_\phi \tilde{N} = [\tilde{N}, \tilde{N}_s]$ is the Jacobian of the normal vector $\tilde{N}$. Note that $|\tilde{T}| = 1$ by (16). Since $\tilde{T}$, $\tilde{N}$ and $\tilde{u}$ are functions of $(u,v,\theta)$, so is $\tilde{\alpha} = \tilde{\alpha}(u,v,\theta)$. We can therefore add the ODE

$$\frac{d\beta}{d\tau} = \tilde{\alpha}(u,v,\theta), \quad \beta(0) = 0,$$ 

(23)

to the geodesic system (19), and then express the attenuation as

$$\exp \left( - \int_0^\tau \alpha(r)dr \right) = \exp(-\omega^{1/3} \beta(\tau)).$$

Note that $\beta$ is independent of the frequency $\omega$.

2.2.2. Geometrical spreading

To compute the amplitude of the creeping ray from (22), we also need to compute the geometrical spreading. We consider again a narrow strip of a geodesics, as in Fig. 2, and let $d\sigma_0(s)$ and $d\sigma(s,\tau)$ be the strip width at the shadow line and at the distance $\tau$ from the shadow line, respectively.

Set $\tilde{u}(s,\tau) := u(\tau)$, where $(u(\tau),p(\tau))$ is a solution to (13) with the initial data (14) so that $\tilde{u}(s,0) = u_0(s)$. Moreover, let

$$\tilde{X}(s,\tau) := \tilde{X}(\tilde{u}(s,\tau)).$$

Then $\tilde{X}$ is the point on the geodesic at the distance $\tau$ from the shadow line, and $\tilde{X}_0(s) = \tilde{X}(s,0)$ is the starting point on the shadow line. Denote the geometrical spreading of the creeping ray at the point $\tilde{X}(s,\tau)$ in the physical space by

$$\mathcal{G}(s,\tau) := \frac{d\sigma(s,\tau)}{d\sigma_0(s)}.$$ 

Moreover, let $d\sigma_0$ and $d\sigma'$ be the strip width in the direction of the shadow line, defined by $d\sigma_0 = \tilde{X}_0 ds$ and $d\sigma' = |\tilde{X}| ds$. See Fig. 3. Then we have

$$\cos \beta_0 = \frac{d\sigma_0}{d\sigma'} = \frac{\tilde{X}_{0r}^+ \cdot \tilde{X}_{0r}}{|\tilde{X}_{0r}^+| \cdot |\tilde{X}_{0r}|}, \quad \cos \beta = \frac{d\sigma}{d\sigma'} = \frac{\tilde{X}_{s}^+ \cdot \tilde{X}_s}{|\tilde{X}_{s}^+| \cdot |\tilde{X}_s|},$$ 

(24)

(25)

where the $\tau$- and $s$-subscripts denote differentiation along the ray and the shadow line, respectively, and $\tilde{X}_{s}^+$ is orthogonal to $\tilde{X}_s$ in the tangent plane to the surface. Since $|\tilde{X}_{0r}^+| = |\tilde{X}_{s}^+| = 1$ by (16), the geometrical spreading is then computed as,

$$\mathcal{G}(s,\tau) = \frac{\tilde{X}_{s}^+ \cdot \tilde{X}_s}{\tilde{X}_{0r}^+ \cdot \tilde{X}_{0r}}.$$ 

(26)
We will show how to calculate the right hand side of (26) numerically, below.

2.3. Eulerian formulation

There are a number of drawbacks with Lagrangian methods based on solving the ODEs (19), (20) and (23). In particular, in the regions where rays diverge or cross, interpolation can be difficult. Instead, we use an Eulerian formulation and derive time-independent PDEs, which can be solved on a fixed computational grid.

Let us now introduce an unknown function $F : \mathbb{P} \rightarrow \mathbb{P}$,

$$F(\gamma) = \begin{pmatrix} U(\gamma) \\ V(\gamma) \\ \Theta(\gamma) \end{pmatrix},$$

(27)

which is the point where the geodesic starting at $u = (u, v) \in \Omega$ with direction $\theta \in \mathbb{S}$ will cross the boundary of $\Omega_p$. See Fig. 4. Since $F$ is constant along a geodesic, we have

$$0 = \frac{d}{d\tau} F(u(\tau), v(\tau), \theta(\tau)) = \frac{du}{d\tau} F_u + \frac{dv}{d\tau} F_v + \frac{d\theta}{d\tau} F_\theta.$$  

(28)

Fig. 3. Geometrical spreading of a creeping ray on the surface, starting at the shadow line and ending at the boundary.

Fig. 4. A geodesic in the parameter space. The function $F$ is defined as $F(u, v, \theta) = (U, V, \Theta)$, with the notation as in the figure.
Using (28) and (19), we can write the escape PDE for $F$ as
\[
\cos \theta F_u + \sin \theta F_v + \nu(\gamma) F_\theta = 0, \quad \gamma \in \Omega_p, \tag{29}
\]
with the boundary condition at inflow points, i.e., the points on $\partial \Omega_p$ at which geodesics are out-going,
\[
F(\gamma) = \gamma, \quad \gamma \in \partial \Omega_p^\text{inflow}.
\]
Note that inflowing characteristics correspond to out-going geodesics.

Now we define a surface phase $\Phi : \mathbb{P} \to \mathbb{R}$, such that $\Phi(\gamma)$ is the distance traveled by a geodesic starting at the point $u$ with direction $\theta$ before it hits the boundary of $\Omega_p$. Using (20), we can derive the PDE for $\Phi$ as
\[
\cos \theta \Phi_u + \sin \theta \Phi_v + \nu(\gamma) \Phi_\theta = \frac{1}{\rho(\gamma)}, \quad \gamma \in \Omega_p, \tag{30}
\]
with the boundary condition at inflow points
\[
\Phi(\gamma) = 0, \quad \gamma \in \partial \Omega_p^\text{inflow}.
\]
In the same way we define a function $B : \mathbb{P} \to \mathbb{R}$ as the $\beta$-value of a geodesic starting at the point $\gamma \in \Omega_p$ when it hits the boundary of $\Omega_p$. We then use (23) and derive the PDE for $B$ as
\[
\cos \theta B_u + \sin \theta B_v + \nu(\gamma) B_\theta = \frac{\tilde{Z}(\gamma)}{\rho(\gamma)}, \quad \gamma \in \Omega_p, \tag{31}
\]
with the boundary condition at inflow points
\[
B(\gamma) = 0, \quad \gamma \in \partial \Omega_p^\text{inflow}.
\]

For the geometrical spreading we consider a fixed shadow line $\gamma_0(s) = (u_0(s), v_0(s), 0_0(s))$ and like in Section 2.2.2 we define
\[
\tilde{u}(s, \tau) = u(s), \quad \tilde{v}(s, \tau) = v(s), \quad \tilde{\theta}(s, \tau) = \theta(s),
\]
where $(u, v, \theta)$ solves (19) with initial data $(u_0(s), v_0(s), 0_0(s))$. Setting $\tilde{\gamma} = (\tilde{u}, \tilde{v}, \tilde{\theta})$ we thus have
\[
\tilde{\gamma}(s, 0) = \gamma_0(s),
\]
with $g$ defined in (19).

For a given shadow line, the creeping rays will lie on a submanifold of phase space $\mathbb{P}$ which we define as $\mathbb{L}(\gamma_0) = \{ \tilde{\gamma}(s, \tau) : \tau \geq 0 \}$. We then introduce the function $Q : \mathbb{L}(\gamma_0) \to \mathbb{R}$ as
\[
Q(\tilde{\gamma}(s, \tau)) := 2(s, \tau).
\]
which is a Eulerian version of the geometrical spreading, restricted to $\mathbb{L}(\gamma_0)$. We will use the following simple Lemma.

**Lemma 1.** The Jacobian $D_\gamma F(\gamma) \in \mathbb{R}^{3 \times 3}$ has rank two for all $\gamma \in \Omega_p$ where it is well-defined. Its null space is spanned by $g(\gamma)$.

**Proof 1.** That $D_\gamma F(\gamma) g(\gamma) = 0$ is just a restatement of (29). Suppose $D_\gamma F(\gamma) v = 0$ and construct a curve $\gamma_0(s) \in \mathbb{P}$ satisfying $\gamma_0(0) = \gamma$ and $\gamma'_0(0) = v$. Let $\tilde{\gamma}(s, \tau)$ be defined for this curve in the same way as above. Then $\frac{d}{ds} F(\gamma_0(s)) = 0$ for $s = 0$. Moreover, since $D_\gamma F(\gamma)$ is well-defined there is a differentiable function $\tilde{\tau}(s)$ such that $F(\gamma_0(s)) = \tilde{\gamma}(s, \tilde{\tau}(s))$ in a neighborhood of $s = 0$. Together this means that
\[
0 = \frac{d}{ds} \tilde{\gamma}(s, \tilde{\tau}(s)) \bigg|_{s=0} = \tilde{\gamma}_s(0, \tilde{\tau}(0)) + \tilde{\tau}'(0) \tilde{\gamma}_s(0, \tilde{\tau}(0)). \tag{32}
\]
Since $-\tilde{\tau}'(0) \tilde{\gamma}_s(0, \tau)$ is a solution to the ODE $(\tilde{\gamma}_s)_s = D_\gamma g(\tilde{\gamma}) \tilde{\gamma}_s$, for $s = 0$, uniqueness of ODE solutions implies that (32) holds for all $\tau \geq 0$, in particular
\[
\tilde{\gamma}_s(0, 0) + \tilde{\tau}'(0) \tilde{\gamma}_s(0, 0) = 0 \iff v = -\tilde{\tau}'(0) g(\gamma).
In order to compute \( Q \) we first find a solution \( z = z(s, \tau) \) to

\[
D_s F(\gamma) z = \frac{d}{ds} F(\gamma_0(s)).
\]  

(33)

We note that \( F(\gamma(s, \tau)) = F(\gamma_0(s)) \) for all \( \tau \geq 0 \), so this \( z \) satisfies

\[
D_s F(\gamma) z = D_s F(\tilde{\gamma}) \tilde{\gamma}_s.
\]

By Lemma 1 we therefore get

\[
z(s, \tau) = \tilde{\gamma}_s + \alpha g(\tilde{\gamma}) = \tilde{\gamma}_s + \alpha \tilde{\gamma}_s,
\]

for some \( \alpha \) and since \( \tilde{X}_s = \tilde{T}(\gamma) \) by (16), we have

\[
[\tilde{T}(\gamma) \times \tilde{N}(\tilde{u}, \tilde{v})] \cdot J(\tilde{u}, \tilde{v}) \tilde{z} = \tilde{X}_s \cdot (\tilde{X}_s + \alpha \tilde{X}_s) = \tilde{X}_s \cdot \tilde{X}_s,
\]

where \( \tilde{z} \in \mathbb{R}^2 \) contains the first two components of \( z \). Consequently, since \( \tilde{T}(\gamma_0(s)) = \tilde{T} \),

\[
Q(\gamma) = \left[ \frac{\tilde{T}(\gamma) \times \tilde{N}(\tilde{u}, \tilde{v})}{[\tilde{T} \times \tilde{N}(u_0(s))] \tilde{X}_0} \right] \tilde{X}_s.
\]

(34)

On the boundary, when \( \gamma \in \partial \Omega_p \) we can simplify the computation and avoid solving for \( z \) in (33). Let \( \tilde{X} : \mathbb{R} \rightarrow \mathbb{R}^2 \) be defined by \( \tilde{X}(s) := \tilde{X}(U(\gamma_0(s)), V(\gamma_0(s))) \) with \( U, V \) defined in (27). As in the proof of Lemma 1 there is a function \( \tilde{\tau}(s) \) such that

\[
\tilde{X}(s) = \tilde{X}(s, \tilde{\tau}(s)).
\]

(35)

After differentiating (35) with respect to \( s \), we get

\[
\tilde{X}_s(s) = \tilde{X}_s \tilde{\tau}(s) + \tilde{X}_s.
\]

Therefore, for \( \gamma \) on the boundary, i.e. \( \gamma = F(\gamma_0) \),

\[
Q(\gamma) = \left[ \frac{\tilde{T}(\gamma) \times \tilde{N}(\tilde{u}, \tilde{v})}{[\tilde{T} \times \tilde{N}(u_0(s))] \tilde{X}_0} \right] \tilde{X}_s(s).
\]

(36)

Note that \( \tilde{X}_s(s) \) can easily be computed from the numerical solution to the PDE (29).

3. Numerical solution of the PDEs

All PDEs (29–31) are of the general form

\[
a f_u + b f_v + c f_\theta = d(u, v, \theta),
\]

(37)

which are time-independent hyperbolic equations.

In the phase space \( \mathbb{P} \), the direction of characteristics at the points on the boundary determines if boundary conditions are needed at that point. We assign boundary conditions at the points where a characteristic is ingoing. For example a characteristic is in-going if \( \dot{u} = \rho \cos \theta > 0 \) on the left boundary and if \( \dot{v} = \rho \sin \theta > 0 \) on the lower boundary. More precisely, suppose \( \Omega \) is the unit square and \( -\pi < \theta \leq \pi \). Then we prescribe boundary condition on \( \partial \Omega_p^{\text{inflow}} \) given by

\[
\partial \Omega_p^{\text{inflow}} = \left\{ u = 0, \ |\theta| < \frac{\pi}{2} \right\} \cup \left\{ u = 1, \ |\theta - \pi| < \frac{\pi}{2} \right\} \cup \left\{ v = 0, \ \theta > 0 \right\} \cup \left\{ v = 1, \ \theta < 0 \right\}.
\]

We always use periodic boundary conditions in the \( \theta \) direction.

To solve these equations, we use a Fast Marching algorithm, given by Fomel and Sethian [8]. We let \( f = (F, \Phi, B) \) and discretize the phase space domain \( \Omega_p = \Omega \times \mathcal{S} \) uniformly, setting \( u_0 = i \Delta u, \ v_j = j \Delta v \) and \( \theta_k = k \Delta \theta \), with the step sizes \( \Delta u = \Delta v = \frac{1}{\mathcal{N}} \) and \( \Delta \theta = \frac{\pi}{\mathcal{N}} \). Then by solving the PDEs (37), we get the approximate solution

\[
\text{ARTICLE IN PRESS}
\]
\[ f_{ijk} = (F_{ijk}, \Phi_{ijk}, B_{ijk}) \approx (F(u_i, v_j, \theta_k), \Phi(u_i, v_j, \theta_k), B(u_i, v_j, \theta_k)). \]

The complexity is \( O(N^3 \log N) \). See [8] for more details.

4. Post-processing

To extract properties like phase and amplitude for a ray family, post-processing of the solution to the escape PDEs (37) is needed. It is based on the following simple observation. By the uniqueness of solutions of ODEs,

\[ F(\gamma_1) = F(\gamma_2), \]

if and only if the points \( \gamma_1 \) and \( \gamma_2 \) lie on the same geodesic.

As an example, suppose we want to compute the surface phase at a point on the scatterer, when the scatterer is illuminated. We assume that the shadow line \( \gamma_0(s) = (u_0(s), v_0(s), \theta_0(s)) \) is known. For each point \((u, v) \in \Omega \) covered by the surface wave there is at least one creeping ray passing that point starting at the shadow line \( \gamma_0(s) \). By the argument above, we can thus find \( s = s^*(u, v) \) and phase angle \( \theta = \theta^*(u, v) \), as the solution to

\[ F(\gamma_0(s)) = F(u, v, \theta). \]  

The phase at \((u, v)\) is then given by

\[ \phi(u, v) = \phi_0(u_0(s^*)) + \Phi(\gamma_0(s^*)) - \Phi(\gamma^*), \quad \gamma^* = (u, v, \theta^*), \]

with \( \phi_0 \) as in (12). Note that \( \gamma^* \) is now in the submanifold \( \mathbb{L}(\gamma_0) \) which was defined in Section 2.3. There may be multiple solutions \((s^*, \theta^*)\) to (38), giving multiple phases.

We now introduce a function \( A : \mathbb{L}(\gamma_0) \to \mathbb{R} \) as the amplitude at the point \( \gamma \in \mathbb{L}(\gamma_0) \) on the geodesic starting at the shadow line \( \gamma_0(s) \). By (22) we can write

\[ A(\gamma^*) = A_0Q(\gamma^*)^{\frac{1}{2}} \exp \left(-\omega^\frac{1}{2}(B(\gamma_0(s^*)) - B(\gamma^*))\right), \]

where \( A_0 \) is the amplitude at the point \( \gamma_0(s^*) \), and \( Q(\gamma^*) \) is computed by (34).

The main difficulty here is to solve (38). We now show how to solve it. Since \( F = (U, V, \Theta) \) is a point on the phase space boundary \( \partial \Omega_p \), it can be reduced to a point \((S, \Theta)\) in \( \mathbb{R}^2 \). For example in a rectangular domain \( \Omega \), Fig. 5, we choose \( S \in [0, 2\pi] \) along \( \partial \Omega \) to be zero at the lower left corner, \( \pi \) at the upper right corner, and \( 2\pi \) again at the lower left corner. Now the left and right hand sides of (38) are curves in \( \mathbb{R}^2 \) parameterized by \( s \) and \( \theta \), and solving the algebraic equation (38) amounts to finding crossing points of these curves. See Fig. 5.

![Fig. 5. Left figure shows a geodesic in a rectangular domain in the parameter space and the choice of \( S \) on the boundary. Right figure shows two crossing curves. One curve is for all points on the shadow line, parameterized by \( s \). The other curve is for a single point in the parameter space with all directions, parameterized by \( \theta \).](image)
Numerically, we discretize the parameterization of the shadow line in \( N \) grid points \( \{s_m\}, m = 1, \ldots, N \). For each point \( \{u_0(s_m)\} \) on the parameter space shadow line, the ray direction \( \theta_0(s_m) \) at the shadow line is computed using the fact that the tangential vector \( \hat{T} \) to the hypersurface at the point \( \gamma_0(s_m) \) should be in the same direction as the incident angle \( \hat{I} \):

\[
\hat{T}(\gamma_0(s_m)) = \hat{I}.
\]

After obtaining the discretized phase space shadow line \( \{\gamma_0(s_m)\} \), we then interpolate the approximate solution \( f_{ijk} \) (available on a regular grid) to find the approximate solution on the shadow line:

\[
\tilde{f}_{s_m} = (\tilde{F}_{s_m}, \tilde{\Phi}_{s_m}, \tilde{B}_{s_m}) \approx (F(\gamma_0(s_m)), \Phi(\gamma_0(s_m)), B(\gamma_0(s_m))).
\]

Having the discretized solution on the shadow line and at the point \( (u, v) \in \Omega \) for all \( N \) directions \( \theta \in [0, 2\pi] \), we then need to find crossing points of two complex lines of \( N \) straight line segments. These crossing points will then be the solutions to (38). The amount of work to do this is proportional to \( N \), by using a monotonic sections algorithm; see e.g. [27]. For all \( N^2 \) points on the surface the computational cost for finding crossing points will then be \( \mathcal{O}(N^3) \). The complexity to solve the PDEs using the Fast Marching method is \( \mathcal{O}(N^3 \log N) \). Therefore the total complexity will be \( \mathcal{O}(N^3 \log N) \).

If we only need to compute the field for one shadow line, it could be done faster. For example by using wave front tracking or solvers based on the surface eikonal equation, the complexity is \( \mathcal{O}(N^2) \). But there are applications when we need the field for many shadow lines. In such cases, using the Fast Marching method can be much faster. We will show one such application in the next section.

Fig. 6. Ray propagation on the shadow zone of an ellipsoid. Top figures show the creeping rays (left) and iso-phase curves (right) in the parameter space between two shadow lines. Bottom figure shows the iso-phase curves and the shadow line (bold) in the physical space.
As an example, in Fig. 6, the iso-phase curves are shown for an ellipsoid illuminated by incident rays in direction \( \vec{I} = [0, 1, 0] \). In the shadow zone between the two shadow lines, there are either one, two or three phases. As it can be seen, multiple phases can be captured. The solution here is computed by the Fast Marching method on a 120^3 grid and using the post-processing described above.

Fig. 7. Shadow line in the physical and parameter space: (a) shadow line in \((x, y, z)\)-space; (b) shadow line in \((u, v)\)-space.

Fig. 8. Right figure shows all creeping rays starting at the shadow line (dashed) and ending at the boundary. The two bold curves are the backscattered ray. Left figure shows two curves corresponding to the rays hitting the top and bottom boundaries in the parameter space. Circles denote the values computed by the Fast Marching method and solid lines denote the values computed by a high order accurate ray tracing method. The crossing point corresponds to the backscattered ray. (a) \( F(\gamma(s_1)) \) and \( F(\gamma(s_2)) + C \); (b) creeping rays in \((u, v)\) space.
5. An application to mono-static RCS computations

Mono-static RCS is a measure of backscattered radiation in the direction of incident waves, when an object is irradiated. Normally most part of it consists of direct reflections, but for not too high frequencies there are situations where creeping rays can give important contribution [3]. The rays that propagate on the surface of the scatterer and return in the opposite direction of incident waves are called backscattered creeping rays.

In this section we apply the fast phase space method on a scattering problem and compute the contribution of the backscattered creeping rays to RCS. For simplicity we only consider the amplitude on the scatterer, ignoring the effect of diffraction coefficients and geometrical spreading outside the scatterer. We assume that the incoming amplitude is one on the shadow line and compute the backscattered amplitude on the shadow line before the ray leaves the scatterer. We compare the results with standard ray tracing.

5.1. Scattering problem

As a test case we consider a hypersurface \( X = X(u,v) \) which is a patch of an ellipsoid with the following parametric equations:

\[
\begin{align*}
x &= -r_1 \cos u , \\
y &= r_2 \sin u \cos v , \\
z &= r_3 \sin u \sin v ,
\end{align*}
\]

where \( r_1 = 2 \), \( r_2 = 1 \), and \( r_3 = 0.5 \) are the ellipsoid’s semiaxes. Notice that in order to avoid the irregularity at the points \((\pm r_1, 0, 0)\), we cut off these points from the parameter space.

First, we need to compute the shadow lines on the scatterer. For this hypersurface we can find them analytically. By (7) and (8), the shadow line corresponding to the incident direction \( \hat{I} = [t_1, t_2, t_3] \) is given by

\[
t_1 r_2 r_3 \cos u_0(s) - t_2 r_1 r_3 \sin u_0(s) \cos v_0(s) - t_3 r_1 r_2 \sin u_0(s) \sin v_0(s) = 0.
\]

The ray directions \( \theta_0(s) \) at the shadow line are then computed using (39). For example, in Fig. 7 the shadow line is shown for \( \hat{I} \parallel [0.9, 1, 0.1] \) in physical and parameter space, respectively.

![Fig. 9. Length of the backscattered creeping rays for many illumination angles.](image-url)
5.2. Finding the backscattered rays

The goal is to find the length and amplitude of the backscattered creeping rays for different incident angles. In order to find the backscattered creeping rays, we use post-processing as before. A backscattered ray starting at point $s_1$ and ending at point $s_2$ on shadow line should satisfy

$$ F(\gamma(s_1)) = F(\gamma(s_2)) + C, $$

where the constant $C$ accounts for the fact that the upper and lower boundaries in the parameter space coincide on the hypersurface. It means that the points with $S = \pi, \ldots, 3\pi/2$ should be changed to $S = \pi/2, \ldots, 0$ and at the same time their $\Theta$ values should be added by $\pi$. The reason for adding by $\pi$ is that we need to reverse the direction of the geodesic starting at $s_2$. Notice that we only consider the geodesics which hit the upper and lower boundaries, because the left and right boundaries are indeed artificial boundaries, introduced to avoid the irregularity.

As before, the right and left hand sides of (40) are curves in $\mathbb{R}^2$ parameterized by $s$, and to find the backscattered ray we need to find crossing points of these curves. Fig. 8(a) shows the intersecting curves in the ($S, \Theta$)-plane for the points on the shadow line corresponding to geodesics hitting the lower and upper boundaries in parameter space, c.f. Fig. 5. Fig. 8(b) shows the creeping rays starting at all $N$ points on the shadow line and the backscattered ray (bold line).

5.3. Length and amplitude of backscattered ray

The length and amplitude of the backscattered creeping rays are computed by a third order interpolation of the solution to the PDEs (37). For a given incident direction $\mathbf{I} = [t_1, t_2, t_3]$, the horizontal and vertical incident angles are calculated as

![Amplitude of the backscattered creeping rays for many illumination angles for $\omega = 1$.](image-url)
Fig. 11. The backscattered creeping rays for four different illumination angles and two different frequencies. Left figures show the backscattered rays in the physical space by bold solid lines. The view direction is in the illumination direction, so that the shadow line is the outermost curve around the ellipsoid. Right figures show the backscattered rays in the parameter space. Shadow lines here are shown by dashed lines. The amplitudes for $\omega = 1$ and $\omega = 20$ are denoted by $a_1$ and $a_{20}$, respectively. (a) $\psi_1 = 0$, $\psi_2 = 0$, length = 2.44, $a_1 = 0.022$; (b) $\psi_1 = 0$, $\psi_2 = 56$, length = 2.43, $a_1 = 0.044$; $a_{20} = 2.40 \times 10^{-5}$; (c) $\psi_1 = 58$, $\psi_2 = 0$, length = 2.89, $a_1 = 0.010$; $a_{20} = 6.84 \times 10^{-6}$; (d) $\psi_1 = 58$, $\psi_2 = 56$, length = 2.16, $a_1 = 0.012$; $a_{20} = 8.73 \times 10^{-6}$. 
They vary from $-60^\circ$ to $60^\circ$. Fig. 9 shows the length for different incident angles.

For computing the geometrical spreading, we again use the fact that the upper and lower boundaries of the domain $\Omega$ in the parameter space coincide on the hypersurface. Therefore, one can consider a new domain $\tilde{\Omega}$ consisting of two domains $\Omega$ on top of each other, connected by the boundary $v = 0$. The creeping ray starting at the point $\gamma(s_1)$ in the upper domain continues in the lower domain and hits the shadow line at the point $\tilde{\gamma}(s_2) = \gamma(s_2) + C$, with $C = (0, -2\pi, \pi)$. Now, let $\tilde{F}$ be the escape location and direction on $\tilde{\Omega}$ for the extended domain $\tilde{\Omega}$. We will have $\tilde{F}(\gamma(s_1)) = F(F(\gamma(s_1)) + \tilde{C})$ and $\tilde{F}(\tilde{\gamma}(s_2)) = F(\gamma(s_2) + \tilde{C})$ where $\tilde{C} = (0, 2\pi, 0)$. We can then use (34) to compute the geometrical spreading $Q(\tilde{\gamma}(s_2))$ at the point $\tilde{\gamma}(s_2)$ from the starting point $\gamma(s_1)$. The amplitude is computed by

$$A(\gamma(s_2)) = A(\gamma(s_1))(Q(\tilde{\gamma}(s_2)))^{\frac{1}{2}} \exp \left( -\omega \mu (B(\gamma(s_1)) + B(\gamma(s_2))) \right).$$

Fig. 10 shows the amplitude for different incident angles. For some incident angles, the geometrical spreading of the creeping ray becomes zero. These rays are called *caustic backscattered creeping rays*, and their amplitude is infinite at the shadow line. However, away from the scatterer their contribution is bounded because of geometrical spreading outside the scatterer. Note that in Fig. 10 the amplitudes larger than a certain value are not shown.

Fig. 11 shows the backscattered creeping rays in the physical and parameter space for four different incident directions.

### 5.4. Convergence and complexity

We use a first order Fast Marching algorithm. Fig. 12 shows the length $\Phi(u, \pi, \pi/2)$ obtained using a coarse mesh of the size $60^3$ and a fine mesh of the size $120^3$. We compare the solution with a reference solution obtained by a high order accurate Ray tracing method. It confirms the first order accuracy of the Fast Marching algorithm.

![Fig. 12](image_url)
The convergence of the length and amplitude (at $x = 1$) of the backscattered creeping ray is shown in Fig. 13 for a fixed vertical incident angle $\psi_2 = 6^\circ$ and different horizontal incident angles $\psi_1$. Although the relative error is worse for the amplitude than for the phase, the rate of convergence confirms the first-order accuracy of the method. The accuracy of amplitude can be improved either by using a higher order fast marching method or by computing the geometrical spreading $Q$ directly by using another ODE instead of numerically differentiating the functions $U$ and $V$ with respect to $u$, $v$ and $\theta$ to compute $\hat{X}_s(s)$ in (36) as done in [17,31].

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

Fig. 13. Length and amplitude (at $\omega = 1$) of the backscattered ray for different horizontal incident angles $\psi_1$ and a fixed vertical incident angle $\psi_2 = 6$. Solutions of Fast Marching algorithm converge to a reference solution obtained by Ray tracing as we use a finer grid. (a) Length; (b) amplitude
rays for each shadow line is \( O(N) \). For all \( N^2 \) shadow lines, it is \( O(N^3) \). Therefore the total complexity will be \( O(N^3 \log N) \). The total cost by using other methods, like wave front tracking and solvers based on the surface eikonal equation, will be \( O(N^4) \), if the cost for each shadow line is \( O(N^2) \). In this case, using the Fast Marching method will then be much faster.

6. Conclusion

We have presented a new phase space method for computing creeping rays in an Eulerian framework. We have formulated the ray propagation problem as a set of time-independent PDEs in a three-dimensional phase space. To solve the PDEs we have used a first-order fast marching method. Properties like phase and amplitude for a ray family as well as wavefronts can be extracted through a fast post-processing. The method is computationally attractive when the solution is sought for many different sources but with the same index of refraction, for example in RCS computations.

In this paper, the surface is assumed to be represented by a single parameterization. In future work, we plan to extend the method to be applicable to more complicated and realistic geometries which can be represented by multiple parameterizations. The information can then be extracted by combining multi-patches through a post-processing. Moreover, we will use a higher order method in order to increase the accuracy.

References

Paper II
A Multiple-Patch Phase Space Method with Application to Creeping Rays Computation

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

October 5, 2006

Abstract. Creeping rays are important in many problems in computational electromagnetics, such as scattering and antenna coupling problems. We present a modification of the single-patch phase space method, [10], to a higher order multiple-patch scheme in order to handle realistic problems containing scatterers with complicated geometries. We split the surface into multiple patches where each patch has a well-defined parameterization. The escape equations are solved in each patch, individually. The creeping rays on the scatterer are then computed by connecting all individual solutions through a fast post-processing.

As an application, we consider a mono-static radar cross section problem and compute the creeping rays contribution for scatterers represented by multiple parameterizations.

1 Introduction

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 outside \( D \) is then given by the Helmholtz equation,

\[
\Delta W + n(x)^2 \omega^2 W = 0, \quad x \in \mathbb{R}^3 \setminus \bar{D},
\]

where \( n(x) \) is the index of refraction, and \( \omega \) is the angular frequency. We can impose either a Dirichlet, Neumann or Robin boundary condition on the boundary of the scatterer \( \partial D \) 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,

\[
W(x) \approx a(x)e^{i\omega \phi(x)}, \quad x \in \mathbb{R}^3
\]

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

Geometrical theory of diffraction (GTD), [6] is a correction to the GO approximations by adding diffraction effects. One type of diffracted rays are creeping rays. They are generated at the shadow line of the scatterer and propagate along geodesics on the surface, continuously shedding diffracted rays in its tangential direction. A wave field, associated to a creeping ray, is generated on the surface

\[
W_s(u) = a(u)e^{i\omega \phi(u)},
\]

1
where $\phi(u)$ and $a(u)$ are surface phase and amplitude and $u \in \mathbb{R}^2$ 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).

Creeping rays can give an important contribution to the solution at medium to high frequencies in many problems such as RCS computations for low observable objects [1, 7] and antenna coupling problems [5, 9], where the creeping ray mechanism provides the dominant coupling.

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

Lagrangian methods are based on ODEs. The simplest Lagrangian method is standard ray tracing [5, 11] 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 [14, 4]. 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 surface eikonal and transport equations [8]. However, these equations only give the correct solution when it is a single wave. In the case of crossing waves, more elaborate schemes must be devised. See [15] for a recent work on this area.

Recently, the authors proposed a fast phase space method [10] based on escape equations which are time-independent PDEs in a three-dimensional phase space. The PDE solution, computed by a fast marching method [3], contains information for all incident angles. The phase and amplitude of the field are 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). 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. However, 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 multiple-patch surface represented by different parameterizations. We modify the single-patch phase space method to this case. The PDEs are solved in each patch, individually. The creeping rays on the scatterer are then computed by connecting all individual solutions through a fast post-processing. The inter-patch boundaries
are treated by the continuity of characteristics. We have also improved the accuracy of the method from first to second order.

Multiple-patch (or multi-block) finite difference schemes have long been used in computational science. They are a sub-class of domain decomposition (DD) methods for solving PDEs by iteratively solving sub-problems on smaller sub-domains [2]. However, in the case of creeping rays, the scheme is not based on iterations.

In Section 2, we review the treatment of a single-patch surface. The construction of the multiple-patch scheme is described in detail in Section 3. In Section 4, we present an application of the method and sample numerical results from a prototype implementation of this scheme.

2 Single-Patch Surfaces

We consider the case when the scatterer surface has a regular explicit parameterization, represented by $\bar{X} = \bar{X}(\mathbf{u})$, where $\bar{X} = (x, y, z) \in \mathbb{R}^3$, and the parameters $\mathbf{u} = (u, v)$ belong to a bounded set $\Omega \subset \mathbb{R}^2$. Let the scatterer be illuminated by incident rays in a direction denoted by a normalized vector $\hat{I} = [\hat{I}_1, \hat{I}_2, \hat{I}_3]$. We assume that the shadow line $u_0(s)$ is represented by a curve in parameter space, with $s$ being the arc length parameterization.

The objective is to compute the phase and amplitude of creeping rays, which are geodesics on the scatterer surface. First, the system of ODEs, describing geodesics and their phase and amplitude, are formulated as time-independent Eulerian PDEs in a phase space. These equations are then solved numerically on a fixed computational grid. The solution to the PDEs is post-processed to extract the information for a particular family of rays.

2.1 Governing Equations

In this section, we write (without derivation) the set of PDEs which are used in computing creeping rays. For derivations see [10].

A geodesic is uniquely characterized by its location and direction on the surface. Letting $\gamma := (u, v, \theta)$, the geodesics satisfy a system of three first-order equations,

\[
\begin{pmatrix}
\dot{u} \\
\dot{v} \\
\dot{\theta}
\end{pmatrix} =
\begin{pmatrix}
\rho(\gamma) \cos \theta \\
\rho(\gamma) \sin \theta \\
\rho(\gamma) \mathcal{V}(\gamma)
\end{pmatrix} := \mathbf{g}(\gamma).
\]

Here, dot denotes differentiation with respect to the parameter $\tau$, which is the arc length along the geodesic in the physical space, and

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

and

\[\mathcal{V}(\gamma) = (\Gamma^1_{11} \cos^2 \theta + 2\Gamma^1_{12} \cos \theta \sin \theta + \Gamma^1_{22} \sin^2 \theta) \sin \theta -
(\Gamma^2_{11} \cos^2 \theta + 2\Gamma^2_{12} \cos \theta \sin \theta + \Gamma^2_{22} \sin^2 \theta) \cos \theta,\]

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

Moreover, we know that the phase $\phi$ is the length of the ray, and the amplitude $a$ is computed by,

\[a(\tau) = a_0 Q(s, \tau)^{-\frac{1}{2}} \exp \left(-\omega^{1/3} \beta(\tau) \right),\]
where $a_0$ is the amplitude at the starting point on the shadow line, $Q(s, \tau)$ is the geometrical spreading at distance $\tau$ from the starting point, and $\beta(\tau)$ is a function representing the attenuation factor given by,

$$\beta(\tau) = \int_0^\tau \tilde{\alpha}(\tau)(r) dr, \quad \tilde{\alpha} = \frac{q_0}{\rho_g} \exp \left( i \frac{\pi}{6} \left( \frac{\rho_g}{2} \right)^{1/3} \right), \quad q_0 \approx 2.33811. \quad (6)$$

Here $\rho_g(u,v,\theta)$ is the radius of curvature of the surface along the ray trajectory.

We introduce the phase space $\mathbb{P} = \mathbb{R}^2 \times S$, where $S$ is the periodic sphere, and consider the triplet $\gamma = (u, v, \theta)$ as a point in this space. The geodesics on the scatterer are then confined to a sub-domain $\Omega_p = \Omega \times S \subset \mathbb{P}$ in phase space.

We now consider the geodesic starting at $\gamma \in \Omega_p$ and ending at the boundary of $\Omega_p$. We call this end point the escape point of the geodesic. See Figure 1. We then define an unknown function $f(\gamma) := (F, \Phi, B)$ for this geodesic, as follows:

- $F: \mathbb{P} \to \mathbb{P}$, $F(\gamma) = (U, V, \Theta)$ is the escape location and direction.
- $\Phi: \mathbb{P} \to \mathbb{R}$ is the distance traveled by the geodesic.
- $B: \mathbb{P} \to \mathbb{R}$ is the difference between the $\beta$-values at the escape and starting points.

![Figure 1: A geodesic in the parameter space, starting at $\gamma = (u, v, \theta) \in \Omega_p$ and ending at the escape point $F(\gamma) = (U, V, \Theta)$.](image)

The set of escape PDEs for $f$ reads

$$g_1(\gamma) f_u + g_2(\gamma) f_v + g_3(\gamma) f_\theta = h(\gamma), \quad h(\gamma) = (0, 0, 0, 1, \tilde{\alpha}), \quad \gamma \in \Omega_p, \quad (7)$$

with the boundary condition at inflow points, i.e., the points on $\partial \Omega_p$ at which geodesics are out-going,

$$f(\gamma) = (\gamma, 0, 0), \quad \gamma \in \partial \Omega_p^{\text{inflow}}.$$

Note that the velocity coefficients $g_1, g_2$ and $g_3$ are the components of $\mathbf{g}(\gamma)$ in (4).

### 2.2 Geometrical Spreading

To compute geometrical spreading, we set $\mathbf{u}(s, \tau) := \mathbf{u}(\tau)$, with $\mathbf{u}(s, 0) = \mathbf{u}_0(s)$. Now let $X(s, \tau) := X(\mathbf{u}(s, \tau))$ be a point on the geodesic at the distance $\tau$ 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 then given by, [10],

\[
Q(s, \tau) = \frac{\tilde{X}_\tau \cdot \tilde{X}_s}{\tilde{X}_\tau \cdot \tilde{X}_0s}. \tag{8}
\]

We now 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 \( \gamma \) solves (4) with initial data \( \gamma_0(s) \). We thus have

\[
\dot{\tilde{\gamma}} = \mathbf{g}(\tilde{\gamma}), \quad \tilde{\gamma}(s, 0) = \gamma_0(s).
\]

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. We then introduce the function \( Q: \mathbb{L}(\gamma_0) \rightarrow \mathbb{R} \) as

\[
Q(\tilde{\gamma}(s, \tau)) := Q(s, \tau),
\]

which is an Eulerian version of the geometrical spreading, restricted to \( \mathbb{L}(\gamma_0) \). It can be computed by

\[
Q(\tilde{\gamma}) = \frac{[\dot{T}(\tilde{\gamma}) \times \hat{N}(\tilde{u}, \tilde{v})] \cdot J(\tilde{u}, \tilde{v})}{[I \times \hat{N}(\mathbf{u}_0(s))] \cdot \tilde{X}_0s(s)}, \quad \dot{T} = J\mathbf{u}, \quad J = [X_u X_v] \in \mathbb{R}^{3 \times 2}, \tag{9}
\]

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

\[
D_\tau F(\tilde{\gamma})z = \frac{d}{ds}F(\gamma_0(s)). \tag{10}
\]

For \( \tilde{\gamma} \) on the boundary, i.e. \( \tilde{\gamma} = F(\gamma_0) \), the formula (9) can be simplified as,

\[
Q(\tilde{\gamma}) = \frac{[\dot{T}(\tilde{\gamma}) \times \hat{N}(\tilde{u}, \tilde{v})] \cdot \tilde{X}_s(s)}{[I \times \hat{N}(\mathbf{u}_0(s))] \cdot \tilde{X}_0s(s)}, \tag{11}
\]

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

Note that \( \tilde{X}_s(s) \) in (11) and \( D_\tau F(\tilde{\gamma}) \) and \( F_s(\gamma_0(s)) \) in (10) can be computed by numerically differentiating the solution to the first three PDEs in (7), as was done in [10]. Instead, one can also directly compute \( \tilde{X}_s \) in (8) by adding other ODEs to the geodesic system (4) as follows: First, we note that \( \tilde{X}_s = J\mathbf{u}_s \). We then differentiate (4) with respect to \( s \) and derive the following ODE system

\[
\dot{\tilde{\gamma}}_s = D_\gamma g \tilde{\gamma}_s, \quad \tilde{\gamma}_s(s, 0) = \gamma_{0s}(s). \tag{12}
\]

By solving this ODE, \( \mathbf{u}_s \) and therefore \( \tilde{X}_s \) can be computed. One can also write the escape PDE for (12) in the same way as before and post-process the phase space solution.

### 2.3 Numerical Solution of the PDEs

The equations (7) are a set of linear hyperbolic equations, and the variable velocity coefficients \( \mathbf{g} = (g_1, g_2, g_3) \) are known and determine the characteristic direction at every point \( \gamma \in \Omega_p \).

The direction of characteristics at the points on the boundary \( \partial \Omega_p \) determines if boundary conditions are needed at that point. We assign boundary conditions at the points where a
characteristic is in-going. Suppose \( \Omega \) is the unit square and \( -\pi < \theta \leq \pi \). Then we prescribe boundary condition on \( \partial \Omega_{\text{inflow}} \) given by

\[
\partial \Omega_{\text{inflow}} = \left\{ \gamma \in \partial \Omega_p \mid \hat{n}(\gamma)^\top g(\gamma) < 0 \right\} = \\
\left\{ u = 0, \ |\theta| < \frac{\pi}{2} \right\} \cup \left\{ u = 1, \ |\theta - \pi| < \frac{\pi}{2} \right\} \cup \left\{ v = 0, \ \theta > 0 \right\} \cup \left\{ v = 1, \ \theta < 0 \right\},
\]

with \( \hat{n} \) being the outward normal vector in the phase space. We always use periodic boundary conditions in the \( \theta \) direction. One important characteristic 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 plane tangentially, such that at some points on the plane the characteristic is in-going, and suddenly it becomes out-going.

We let \( f = (F, \Phi, B) \) and discretize the phase space domain \( \Omega_p = \Omega \times S \) 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 \( \Delta \theta = \frac{2\pi}{N} \). One way to solve the equations (7) is to compute the approximate solution

\[
f_{ijk} = (F_{ijk}, \Phi_{ijk}, B_{ijk}) \approx (F(u_i, v_j, \theta_k), \Phi(u_i, v_j, \theta_k), B(u_i, v_j, \theta_k)),
\]

at each grid point using a cell-based ray tracing method, which traces back along the characteristic to the initial boundary. Instead, we use a Fast Marching algorithm, given by Fomel and Sethian [3]. The basic idea of the algorithm is to march the solution outwards from the boundary and use the characteristic directions to update grid values. The grid points are divided into three classes:

- **Accepted**: the correct value of \( f_{ijk} \) has been computed.
- **Considered**: adjacent to **Accepted** for which \( f_{ijk} \) has already been computed, but may be corrected by a later computation.
- **Far**: the correct value of \( f_{ijk} \) is 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 them a large value. Put the boundary nodes \((u_i, v_j, \theta_k) \in \partial \Omega_{\text{inflow}} \) in **Accepted**, and assign them 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 \( \Phi_{ijk} \) 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 a new value for \( f_{ijk} \). In the case we can compute a new value for a **Far** node, put it in **Considered**.
4. Loop to step 1 until all points are **Accepted**.

\(^1\)We approximate the characteristic by a linear curve for a first order method and parabolic for a second order method.
Since in [3] 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 $\Omega_p$, 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 2.

Let $\tau$ be the arc length parameterization along the characteristic $\gamma(\tau)$. We start at $\gamma(\tau = 0) = (u_i, v_j, \theta_k)$, where we want to compute a possibly new value, and trace backwards along the characteristic to intersect a cell face at $\gamma(\tau = \tau^*)$, $\tau^* < 0$. Knowing $\frac{d}{d\tau}F(\gamma(\tau)) = 0$ and $\frac{d}{d\tau}\Phi(\gamma(\tau)) = 1$, we have

$$F(\gamma(\tau^*)) = F(\gamma(0)), \quad \Phi(\gamma(\tau^*)) = \Phi(\gamma(0)) + \tau^*.$$  \hspace{1cm} (13)

Moreover, since $\frac{d}{d\tau}B(\gamma(\tau)) = \tilde{\alpha}(\gamma(\tau))$, we Taylor expand $B$ near the starting point,

$$B(\gamma(\tau^*)) = B(\gamma(0)) + \tau^* \frac{d}{d\tau}B(\gamma(0)) + \frac{\tau^{12}}{2} \frac{d^2}{d\tau^2}B(\gamma(0)) + O(\tau^{33}), \quad \hspace{1cm} (15)$$

with local truncation error $O(\tau^{33}) \approx O(\Delta u^3)$. Therefore, to find $f(\gamma(0))$, we need to know $\tau^*$ and $f(\gamma(\tau^*))$. Note that the values of $\frac{d}{d\tau}B(\gamma(0))$ and $\frac{d^2}{d\tau^2}B(\gamma(0))$ in (15) are known:

$$\frac{d}{d\tau}B(\gamma(0)) = \tilde{\alpha}(\gamma(0)),$$

$$\frac{d^2}{d\tau^2}B(\gamma(0)) = \frac{d}{d\tau}\tilde{\alpha}(\gamma(0)) = g_1(\gamma(0)) \tilde{\alpha}_u(\gamma(0)) + g_2(\gamma(0)) \tilde{\alpha}_v(\gamma(0)) + g_3(\gamma(0)) \tilde{\alpha}_\theta(\gamma(0)).$$

2.3.1 First Order Method

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

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

Note that $\sigma_1$ and $\sigma_2$ are known. There are six possible planes, $u = u_{i\pm 1}$, $v = v_{j\pm 1}$, $\theta = \theta_{k\pm 1}$, 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 relations (13)-(14) and the first two terms of the Taylor expansion (15) to compute a new value for $f_{ijk} = f(\gamma(0))$. Put this node in $Considered$. Since the method is first order, a two dimensional bilinear interpolation is used. See Figure 2.

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 previous crossing point is considered as the new starting point. At the end, we need to add all $\tau^*$ values to compute $\Phi_{ijk}$ and all $\tau^* \frac{d}{d\tau}B$ values to compute $B_{ijk}$.
2.3.2 Second Order Method

We assume that characteristics are parabolic in each cell and write

\[ \gamma(s) \approx \sigma_1 + \sigma_2 s + \sigma_3 s^2, \quad \sigma_1 = \gamma(0), \quad \sigma_2 = \dot{\gamma}(0), \quad \sigma_3 = \frac{1}{2} \ddot{\gamma}(0) = \frac{1}{2} D \gamma \dot{\gamma}(0) \dot{\gamma}(0). \]

Note that \( \sigma_1, \sigma_2 \) and \( \sigma_3 \) 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 \textit{Accepted}, use these points to interpolate a value of \( f(\gamma(\tau^*)) \). Then use relations (13)-(14) and the first three terms of the Taylor expansion (15) to compute a new value for \( f_{ijk} = f(\gamma(0)) \). Put this node in \textit{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 [12].

b. If no points on the cell face are \textit{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 previous cross point is considered as the new start point. At the end, we need to add all \( \tau^* \) values to compute \( \Phi_{ijk} \) and all \( \tau^* \frac{d}{d\tau} B + \frac{\tau^{2*}}{2} \frac{d^2}{d\tau^2} B \) values to compute \( B_{ijk} \).

The algorithm is a one-pass algorithm and is of complexity \( O(N^3 \log N) \). Note that we use heap sort algorithm for extracting the smallest arrival time \( \Phi_{ijk} \) of \textit{Considered} nodes and for inserting new updated values of \textit{Considered} nodes.

2.4 Post-Processing

Solutions of the escape PDEs (7) need to be post-processed to extract properties like phase and amplitude for a ray family.

We consider an illuminated scatterer and assume that the shadow line is known and given by \( \gamma_0(s) = (u_0(s), v_0(s), \theta_0(s)) \). We first observe that \( F(\gamma_1) = F(\gamma_2) \), if and only if the points...
\[ \gamma_1 \text{ and } \gamma_2 \text{ lie on the same geodesic. For each point } (u, v) \in \Omega \text{ covered by the surface wave there is at least one creeping ray, starting at the shadow line, which passes through it. We can thus find } s = s^*(u, v) \text{ and phase angle } \theta = \theta^*(u, v), \text{ as the solution to} \]
\[ F(\gamma_0(s)) = F(u, v, \theta). \quad (16) \]

The phase at \((u, v)\) is then computed as
\[ \phi(u, v) = \phi_0(u_0(s^*)) + \Phi(\gamma_0(s^*)) - \Phi(\gamma^*), \quad \gamma^* = (u, v, \theta^*) \in \mathbb{L}(\gamma_0), \]
where \(\phi_0(u_0(s^*))\) is the phase at \(u_0(s^*)\). Note that there may be multiple solutions \((s^*, \theta^*)\) to \((16)\), giving multiple phases.

We now introduce a function \(A : \mathbb{L}(\gamma_0) \to \mathbb{R}\) as the amplitude at the point \(\gamma \in \mathbb{L}(\gamma_0)\) on the geodesic starting at \(\gamma_0(s)\). By \((5)\) we can write
\[ A(\gamma^*) = A_0 Q(\gamma^*) \frac{i}{2} \exp \left( -\omega \frac{i}{3} (B(\gamma_0(s^*)) - B(\gamma^*)) \right), \]
where \(A_0\) is the amplitude at \(\gamma_0(s^*)\), and \(Q(\gamma^*)\) is computed by \((9)\).

To solve \((16)\), we reduce \(F = (U, V, \Theta) \in \partial \Omega_p\), to a point \((S, \Theta)\) in \(\mathbb{R}^2\). The left and right hand sides of \((16)\) are then curves in \(\mathbb{R}^2\) parameterized by \(s\) and \(\theta\), and solving the algebraic equation \((16)\) amounts to finding crossing points of these curves.

We now discretize the shadow line in \(N\) grid points \(\{\gamma_0(s_m)\}, m = 1, \ldots, N\) and interpolate the approximate solution \(f_{ijk}\) (available on a regular grid) to find the approximate solution on the shadow line:
\[ \tilde{f}_{s_m} = (\tilde{F}_{s_m}, \tilde{\Phi}_{s_m}, \tilde{B}_{s_m}) \approx (F(\gamma_0(s_m)), \Phi(\gamma_0(s_m)), B(\gamma_0(s_m))). \]

Having the solution also at the point \((u, v) \in \Omega\) for all \(N\) directions \(\theta \in \mathcal{S}\), we then need to find crossing points of two complex lines of \(N\) straight line segments as the solutions to \((16)\). This can be done with a complexity of \(\mathcal{O}(N)\); see e.g. [13]. The complexity of post-processing for all \(N^2\) points on the surface 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 the field for only one shadow line. 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 shadow lines, the phase space method is more efficient. See Section 4 for such an example.

3 Multiple-Patch Surfaces

We now consider the more complicated and realistic case when the surface cannot be described by one single parameterization. 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 reference boundary.

3.1 Multiple-Patch Scheme

We assume that the scatterer surface is a two-dimensional closed\(^2\) manifold \(M\) embedded in \(\mathbb{R}^3\). Let \(T_x M\) be the tangent plane (the set of tangent vectors) to \(M\) at some point \(x \in M\),

\(^2\)Note that the assumption of a closed surface is not really necessary but simplifies the discussion. We could assume an open surface too.
and let $TM = \bigcup_{x \in M} T_x M$ be the tangent bundle of $M$. The dimension of $TM$ is twice the dimension of $M$. An element of $TM$ is a pair $(x, \nu)$ where $x \in M$ and $\nu \in T_x M$. Since on a geodesic $\hat{T}$ in (9) has unit length, it will be enough for us to consider the unit tangent bundle $T^1 M$ of $M$ which contains all unit-normed tangent vectors. Note that $T^1 M$ is a three-dimensional manifold embedded in $\mathbb{R}^6$.

We now want to define a function $F$ for the multiple patch case that corresponds to the single patch solution $F$ described in Section 2. Let $R$ be some curve in $M$, representing an escape boundary. We consider a geodesic starting at a point $\Gamma \in T^1 M$ and define $F(\Gamma) : T^1 M \to T^1 M$ as mapping the point $\Gamma$ to another point in the unit tangent bundle where the geodesic first crosses $R$ (assuming such a point exists).

Let $M$ be described by an atlas of charts $(M_j, w_j)$, with $j = 1, \ldots, P$, where the closed sets $M_j$ collectively cover $M$, and the mapping $w_j : M_j \to \Omega$ is one to one. We will therefore have $M = \bigcup_j w_j^{-1}([0,1]^2)$. Note that specifically $M_j$ are the patches and $w_j = X_j^{-1}$, where

$$X_j = X_j(u) : [0,1]^2 \to M_j \subset \mathbb{R}^3,$$

are the parametric equations of the patches.

We assume further that the patches stick together along their sides 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 $S_{jj'} = M_j \cap M_{j'}$) Denote the set of all sides by $\mathcal{S}$.

For a closed scatterer there is no obvious escape boundary, as in the single patch case. In fact, the characteristics never escape the scatterer, and given two points on $M$ there will in general be an infinite number of rays connecting them, rays that may traverse a long distance on the scatterer surface in between. We will define a global reference boundary where we capture the rays. Let

$$R = \bigcup_{(j,j') \in \mathcal{R}} S_{jj'} \subset \mathcal{S} \quad (17)$$

be this boundary, where $\mathcal{R}$ is some index set, to be determined (see below).

For each patch $j$, we define a mapping $W_j : T^1 M_j \to \Omega_p$ by $W_j(x, \nu) = \gamma$, where

$$\gamma = (w_j(x), \theta), \quad J_j(w_j(x))\dot{s}(\theta) = \nu, \quad J_j = D_x X_j, \quad \dot{s}(\theta) = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}. \quad (18)$$

Since $w_j$ is one to one, and the columns of the Jacobian $J_j$ span the tangent plane at $x$, and $\nu$ belongs to this plane, the mapping $W_j$ is also one to one. Note that $T^1 M = \bigcup_j W_j^{-1}(\Omega_p)$. Figure 3 shows a schematic representation of the two-dimensional manifold $M$ and the three-dimensional unit tangent bundle $T^1 M$ and the corresponding one to one mappings to the parameter space $\Omega$ and phase space $\Omega_p$.

### 3.2 Computing $F(\Gamma)$

To compute $F(\Gamma)$, we first note that, by construction, for each point $\Gamma = (x, \nu) \in T^1 M$, except when $\Gamma$ is on a patch boundary, there is a well-defined patch id number $j = J(\Gamma)$ as well as a well-defined mapping $W_j$ such that $\Gamma \in T^1 M_j$. We extend this function also to the patch boundaries by defining $J(\Gamma)$ to be the id of the patch into which the geodesic starting at $\Gamma$ enters.
Figure 3: A schematic representation of the two-dimensional surface $M$ embedded in $\mathbb{R}^3$ and its three-dimensional unit tangent bundle $T^1 M$ embedded in $\mathbb{R}^6$. The one to one mappings $w_j$ and $W_j$ map a chart $j$ of these manifolds to the parameter space $\Omega$ and phase space $\Omega_p$, respectively.

Now, suppose $F_j(\gamma)$ are the solutions to the escape PDE (7) in each patch in $\Omega_p$ with $j = 1, \ldots, P$. The function $F(\Gamma)$ is then given recursively by

$$\Gamma_0 = \Gamma,$$

and while $x_n \notin R$, where $\Gamma_n = (x_n, \nu_n)$,

$$\Gamma_{n+1} = W_{j}^{-1} F_j(W_j(\Gamma_n)), \quad j = J(\Gamma_n).$$

Then $F(\Gamma) = \Gamma_{n^*}$, where $n^*$ is the smallest index for which $x_{n^*} \in R$.

From the above recursive formula, it is easy to see that in order to compute the function $F$ for all points in $T^1 M$ it is enough to know the escape PDE solutions $F_j$ in all patches and the patch transfer functions $T_{jj'} := W_{j'}^{-1} W_j$ at all sides connecting two patches $j$ and $j'$. Note that these transfer functions can easily be computed by (18).

Similar to $F(\Gamma)$, we can define the functions $\Phi(\Gamma)$ and $B(\Gamma)$ in $T^1 M$ for the multiple patch case corresponding to the single patch functions $\Phi$ and $B$ described in Section 2. Assuming $\Phi_j(\gamma)$ and $B_j(\gamma)$ are the solutions to the escape PDE (7) in each patch in $\Omega_p$ with $j = 1, \ldots, P$, we can write

$$\Phi(\Gamma) = \sum_{n=0}^{n^*-1} \Phi_{j(\Gamma_n)}(W_j(\Gamma_n)), $$

$$B(\Gamma) = \sum_{n=0}^{n^*-1} B_{j(\Gamma_n)}(W_j(\Gamma_n)), $$

with $\Gamma_n$ as in (19).

Remark. 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 4 shows an ellipsoid represented by 6 non-singular patches as an example. Figure 5 shows the graph corresponding to the ellipsoid.
Figure 4: 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.

divided into six patches which are connected through twelve sides. 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 \( E \)) is therefore connected to six other nodes through two patches which are connected by side \( E \). See Figure 5. This structure can be useful in computing \( \mathbf{F} \).

### 3.3 Post Processing

In order to compute phase and amplitude of a ray family, post-processing of the solutions to the escape PDE (7) are needed.

For a given illumination direction, assume that the shadow line is known and given by \( \Gamma_0(s) \) in the unit tangent bundle \( T^1M \). For each point \( x \in M_j \) 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, We first choose the reference boundary \( R \) as the boundaries
Figure 5: 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.

of $M_j$. We find $F(W_j^{-1}(w_j(x), \theta))$ for all directions $\theta \in \mathbb{S}$. Moreover, for all points on the shadow line we find

$$F_n(\Gamma_0(s)) := F \circ F \cdots \circ F(\Gamma_0(s)),$$

where $n$ can be either one or two, depending on whether $\Gamma_0(s)$ is inside or outside the patch $j$. Note that in the case that the ray crosses $R$ more than once before it passes through the point $x$, we may need to take $n$ more than two. We then find $s = s^*$ and $\theta = \theta^*$ as the solutions to the algebraic equations

$$F(\Gamma_0(s)) = F_n(W_j^{-1}(w_j(x), \theta)),$$

(20)

analogous to (16) in the single-patch case. There will be at most four systems of equations corresponding to four sides of patch $j$. The solutions to (20) can be computed by finding intersections of four sets of possibly crossing curves.

Now we can use (10) to compute $z$ with $\gamma_0 = W_{j_0}(\Gamma_0(s^*))$ and $\tilde{\gamma} = (w_j(x), \theta^*)$ where $j_0 = J(\Gamma_0(s^*))$. The geometrical spreading $Q(\tilde{\gamma})$ at point $x$ will be therefore computed by (9), and phase and amplitude are given by

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

$$A(\tilde{\gamma}) = A_0 Q(\tilde{\gamma})^{\frac{1}{2}} \exp \left( -\frac{i}{2} \left( B(W_{j_0}^{-1}(\gamma_0)) - B(W_j^{-1}(\tilde{\gamma})) \right) \right),$$

where $\phi_0$ and $A_0$ are the phase and amplitude at the point $\gamma_0$, respectively.

### 3.4 Number and Shape of Patches and Coordinates

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

1. They should cover the scatterer surface with nonsingular coordinates.
2. Gradient of curvature should not be too high in one patch comparing to another.
3. They should not have relatively large coordinate distortions which makes finite differencing less accurate.

Remark. Using overlapping patches, one can possibly reduce the number of patches. For ellipsoid, for instance, one can use two overlapping patches instead of six non-overlapping patches. However, the objective in this work has not been optimizing the number of patches.

3.5 Choosing Reference Boundary

Another key design choice is the choice of reference boundary. Two things are important about \( R \), and \( R^\prime \):

1. Each creeping ray originating on the surface should cross \( R \) at some point, otherwise \( F(\Gamma) \) is not well defined for all points. It is not obvious how to verify this. In the single patch case having nonzero coefficients, \( g(\gamma) \neq 0 \), everywhere is enough.

2. The rays that are of interest should cross \( R \) only once. In principle also later escape points can be captured by composing \( F \) by itself (i.e., using \( F_n \)).

3.6 Limitations and Extra Problems

There are a couple of difficulties and problems:

1. One may cannot capture all creeping rays by one choice of reference boundary. Different choices of reference boundary might be needed. A good implementation of the algorithm will then be the one which considers different combinations of patch boundaries as the reference boundary. Note that this is done in post-processing.

2. There are extra problems with interpolation along the inter-patch boundaries. These problems are more crucial if a creeping ray is tangent to the inter-patch boundary. One possible way to overcome such problems is to use overlapping patches. Another possibility is to choose another atlas of charts for the surface.

4 An Application to Mono-static RCS Computations

In this section we 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 consider two scatterer surfaces; a scalene ellipsoid (an ellipsoid with different semi-axes) and a balloon. 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.

4.1 Example 1 - A Scalene Ellipsoid

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 4) and solve for $f(\gamma)$ in each patch, as described in Section 2.3.

In order to find the backscattered creeping ray by post-processing, we first choose a reference boundary consisting of six sides $R = \bigcup R_m$, $m = 1, \ldots, 6$, 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 $\gamma_0(s) = (u_0(s), \theta_0(s))$ in the phase space $\Omega_p$ 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 $F \left( W_{j(a)}^{-1}(\gamma_0(s_n)) \right)$ as discussed in Section 3.2.

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

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

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 $\pi$ on the reference 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 reference boundary $R_1, \ldots, R_6$.

To find the backscattered ray we need to find crossing points of these curves, as is done in the single patch case.

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 3.3 with $\gamma_0 = \gamma_0(s_d)$ 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,

\[
\phi = \Phi(\Gamma_{sa}) + \Phi(\Gamma_{sd}), \quad \Gamma_{sa} = W_{ja}^{-1}(\gamma(s_a)), \quad \Gamma_{sd} = W_{jd}^{-1}(\gamma(s_d)),
\]

\[
A_1 = Q(\gamma(s_a))^{\frac{1}{2}} \exp \left( -\omega \frac{1}{2} \left( B(\Gamma_{sa}) + B(\Gamma_{sd}) \right) \right),
\]

\[
A_2 = Q(\gamma(s_d))^{\frac{1}{2}} \exp \left( -\omega \frac{1}{2} \left( B(\Gamma_{sa}) + B(\Gamma_{sd}) \right) \right).
\]

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.

![Image of backscattered rays](image)

We notice that in [10], 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. 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 [10], where a first order algorithm was used, the accuracy of amplitude has been improved dramatically.

4.2 Example 2 - A balloon

We consider a balloon-shape surface consisting of a hemisphere in the positive side of $z$-axis, centered at the origin and with radius $R$, and the surface created by rotating the parabola $z^2 = 2R(R - y)$ about $z$-axis in its negative side. This is a simple smooth version of the cone-hemisphere studied in [1]. We divide this surface into six patches, as shown in Figure 10;
Figure 8: Length and amplitudes (with $\omega = 1$) of backscattered creeping rays for many illumination angles.
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.
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 10.

Figure 10: 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.

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 11 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 12 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.
5 Conclusion

We have modified the single-patch phase space method for computing creeping rays to handle problems with more complicated geometries. In such problems the scatterer surface is described by multiple parameterizations. The *escape* PDEs are independently solved in each patch using a second order fast marching algorithm. The solutions are connected to each other by a multiple-patch scheme in which the inter-patch boundaries are treated by continuity of characteristics. The creeping rays are then computed 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 and antenna coupling problems.

References


Figure 12: 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.

1953.


