Fast Phase Space Computation of Multiple Arrivals

S. Fomel and J.A. Sethian *

Dept. of Mathematics
University of California, Berkeley 94720

December 4, 2001

Abstract

We present a fast, general computational technique for computing the phase-space solution of static Hamilton-Jacobi equations. Starting with the Liouville formulation of the characteristic equations, we derive "Escape Equations" which are static, time-independent Eulerian PDEs. They represent all arrivals to the given boundary from all possible starting configurations. The solution is numerically constructed through a 'one-pass' formulation, building on ideas from semi-Lagrangian methods, Fast Marching Methods, and Ordered Upwind Methods. To compute all possible trajectories corresponding to all possible boundary conditions, the technique is of computational order $O(N \log N)$, where N is the total number of points in the computational phase-space domain; any particular set of boundary conditions is then extracted through rapid post-processing. Suggestions are made for speeding up the algorithm in the case when the particular distribution of sources is provided in advance. As an application, we apply the technique to the problem of computing first, multiple, and most energetic arrivals to the Eikonal equation.

1 Introduction

We present a fast, general computational technique for computing phase space solutions of static Hamilton-Jacobi equations. We derive a set of "Escape equations" which are static, time-independent Eulerian partial differential equations which represent all arrivals to the given boundary from all possible starting configurations. Following the strategy proposed in [15] we solve these Escape Equations by systematically constructing space marching the solution in increasing order, using a 'one-pass' formulation. This means that the solution at each point in the computational mesh is computed only k times, where k does not depend on the number of points in the mesh. The algorithm combines ideas of semi-Lagrangian methods, Fast Marching Methods and Ordered Upwind Methods [28, 31]. The method is unconditionally stable, with no CFL time-step restriction, and can be made higher order accurate. We demonstrate the applicability of this technique by computing multiple arrivals to the Eikonal equation in a variety of settings.

The methods presented here are efficient. The Escape Equations are posed time-independent Eulerian equations in phase space, whose solution gives the exit time and location for all possible trajectories, starting from all interior points, initialized in all directions. The computational speed depends on whether one wants to in fact obtain results for all possible boundary conditions, or in fact only for a particular subset of possibilities.

To illustrate, consider a two-dimensional problem consisting of a region and its boundary; we discretize the region with a square mesh with N points on each side. Thus, the physical space corresponding to the interior consists of N^2 points, with N points on the boundary (we ignore constants).

*This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract Number DE-AC03-76SF00098, and the Office of Naval Research under under grant FDN00014-96-1-0381.

In the most general form of boundary conditions, such as those which occur in applications such as tomography and seismic migration, one needs to solve multiple boundary problems with $H(x, \nabla u) = 0$ and the point-source boundary condition $u(x) = 0$ for $x = s$ with a set of sources s distributed on the surface of the observational domain. In this case, the solutions span three-dimensional space, composed of x and s. Because of our use of a fast ordering scheme, we can find all possible exit times and locations for all possible trajectories in $O(N^3 \log N)$. One can use the output of such computation either for extracting multiple arrivals for a particular set of sources or directly, as in the method of angle-gather migration [36, 5]. In constrast with our approach, we note that to obtain the exit time and position from each of the N^2 interior grid points using a Lagrangian approach would require integration from each of N^3 starting values; a typical integration would require N steps, giving an operation count of N^4. Analogously, for a problem in three-dimensional physical space with N points on each edge of a computational cube in physical space, we find all possible exit trajectories for all possible boundary conditions in $O(N^5 \log N)$.

In the case where particular boundary conditions are known in advance, computational speedup is possible; this is discussed after the algorithm is introduced.

2 Formulation of Problem

Consider the static Hamilton-Jacobi equation $H(x, \nabla u) = 0$. A nonlinear Hamiltonian H may not have a unique solution, even with smooth boundary data and smooth H. A particular, *viscosity-type* solution can be selected [9, 10], corresponding to the earliest arrival from the given boundary. Fast algorithms for computing these viscosity-satisfying first arrival solutions have been developed in recent years, in particular, the Fast Marching Method, developed by Sethian [28] for computing the solution to the Eikonal equation [28] and "Ordered Upwind Methods (OUM)", introduced by Sethian and Vladimirksy [31, 32] for computing the solution of convex static Hamilton-Jacobi equations which arise in anisotropic front propagation and optimal control. These first arrivals are of considerable importance in a large collection of problems such as computing seismic travel times [29]; see [30, 35] for reviews.

However, later arrivals may carry additional valuable information, and it is often desirable to compute all possible solutions. For example, in geophysical simulations, first arrivals may not correspond to the most energetic arrivals, and this can cause problems in seismic imaging [1, 17].

There are two approaches to computing these multiple arrivals.

- The Lagrangian (ray tracing) approach [6, 26] and its variations [27, 37]. Here, the phase space characteristic equations are integrated, often from a source point, resulting in a Lagrangian structure which fans out over the domain. This is a valuable and common approach, however it can face difficulties either in low ray density zones where there are very few rays or near caustics where rays cross; in addition, the use of irregular computational grid is often inconvenient.

- A different approach is to work with an Eulerian description of the problem, in either the physical domain or phase space, and attempt to extract multiple arrivals. In recent years, this has led to many fascinating and clever Eulerian PDE-based approaches to computing multiple arrivals, including slowness matching algorithms [34], dynamic surface extension algorithms [33] and its modification [25], segment projection methods [13], and "big-ray tracing" [2]; see also [3]. We note that the regularity of the phase space has been utilized previously in theoretical studies on the asymptotic wave propagation [24, 12].

As an example, consider a one-dimensional closed curve bounding a region in the plane, and suppose one has a collection of sources located along the entire boundary; the goal is to consider a front propagating inwards from this boundary. The Lagrangian approach is to work in phase space and discretize this boundary into a set of marker points, whose motion is determined by solving the characteristic equations. The curve then evolves in three-dimensional phase space; and the projection of the curve back into two dimensional physical space produces the multiple arrivals. An Eulerian formulation of this same approach was pursued by Engquist, Runborg and Tornberg [13] by using the Vlasov equation to describe the motion of this curve; in their Segment Projection Method, the curve moving in three-dimensional phase space is viewed from several different coordinate systems so that it always remains locally a graph.

The approach presented in this paper computes the solution in phase space, but does so in a reduced, time-independent Eulerian partial differential equations framework. In contrast to other approaches, we can efficiently compute the exit time and location for all possible trajectories, starting from all interior points, initialized in all directions; extraction of particular boundary conditions comes as post-processing. Alternatively, if one is given a particular set of boundary conditions corresponding to a placement of a sources in advance, this can be incorporated into ordering process and the resulting solution may be computed more rapidly.

3 Formulation of Equations

3.1 Lagrangian Formulation of Phase-Space Solution

We begin with the static Hamilton-Jacobi equation:

$$H(x, \nabla u) = 0. \tag{1}$$

and write the well-known characteristic equations in phase space (x, p), where p corresponds to ∇u (see, for example, [14]). Let σ is a parameter varying along the trajectory. Differentiating Eqn. 1 with respect to σ, we obtain

$$\nabla_x H \cdot \frac{dx}{d\sigma} + \nabla_p H \cdot \frac{dp}{d\sigma} = 0, \tag{2}$$

where the Hamiltonian H is assumed to be twice continuously differentiable. This equation is satisfied when the characteristics obey

$$\frac{dx}{d\sigma} = \nabla_p H; \qquad \frac{dp}{d\sigma} = -\nabla_x H \tag{3}$$

Differentiating the function $u(x(\sigma))$, we obtain an additional equation for transporting the function u along the characteristics:

$$\frac{du}{d\sigma} = \nabla u \cdot \frac{dx}{d\sigma} = p \cdot \nabla_p H \tag{4}$$

The system given by Eqns. 3,4 can be initialized at $\sigma = 0$: $x(0) = x_0$, $p(0) = p_0$, $u(0) = 0$.

In the Lagrangian approach, one computes the solution of the point-source Hamilton problem by starting at a source point x_0, taking through different initial values of p_0 and filling the x space with trajectories that follow Eqns. 3. The strength of this approach is in the fact that the solution is uniquely defined in the phase space by following individual trajectories. However, when the trajectories collide, the solution in the physical space becomes multi-valued and interpolating it onto a regular x grid presents a difficult computational problem [22].

3.2 Liouville Formulation of Phase-Space Solution

We now convert the phase space approach into a set of Liouville equations; these have been used extensively in different applications by Chorin, Hald, and Kuperfman [7, 8]. Eqns. 3,4 form a system of coupled ordinary differential equations, starting with a particular set of initial conditions. The Liouville equation is a partial differential equation for the same solution with the differentiation performed with respect to the initial conditions; it describes the local change in the solution in response to changes in the initial conditions.

To simplify notation, let us denote the phase-space vector (x, p), by y, the right-hand side of system given in Eqn. 3 by vector function $R(y)$, and the right-hand side of Eqn. 4 by the function $r(y)$. In this notation, the Hamilton-Jacobi system takes the form

$$\frac{\partial y(y_0, \sigma)}{\partial \sigma} = R(y); \qquad \frac{\partial u(y_0, \sigma)}{\partial \sigma} = r(y), \tag{5}$$

and is initialized at $\sigma = 0$ as $y = y_0$ and $u = 0$.[1]

[1] The full y space has $2n$ variables. However, by using the Hamilton equation $H(y) = 0$ as an additional constraint, we can often reduce the Hamilton-Jacobi system (5) to $2n - 1$ equations of the same form. Therefore, we can assume that y has $2n - 1$ independent components.

In the Appendix, we show that the solution of system (5) as a function of both the trajectory parameter σ and the initial condition y_0 satisfies the Liouville partial differential equations

$$\frac{\partial y(y_0, \sigma)}{\partial \sigma} = \nabla_0 y \, R(y_0) \ , \tag{6}$$

and the transported function u satisfies the analogous equation

$$\frac{\partial u(y_0, \sigma)}{\partial \sigma} = \nabla_0 u \, R(y_0) + r(y_0) \ , \tag{7}$$

where ∇_0 denotes the gradient with respect to y_0. These are the Liouville equations.

3.3 Formulation of the Stationary "Escape" Equations

Assume that there exists a closed boundary $\partial\mathcal{D}$ in the y space that is crossed by every characteristic trajectory that originates in $y_0 \in \mathcal{D}$. This defines for every y_0 the function $\sigma = \widehat{\sigma}(y_0)$ of the first crossing of the corresponding characteristic with $\partial\mathcal{D}$.

Let us now introduce a differentiable function $\Gamma(y)$ that identifies the boundary, that is, $\Gamma(y) = 0$. In particular, we then have that $\Gamma\left(y(y_0, \widehat{\sigma}(y_0)\right) = 0$. We can differentiate with respect to the initial condition y_0 to obtain

$$\Gamma'(y) \left[\nabla_0 y + \frac{\partial y}{\partial \sigma} \nabla_0 \widehat{\sigma}\right] = 0 \ . \tag{8}$$

Multiplying equation (8) by $R(y_0)$ and using the Liouville equation (6), we obtain

$$\Gamma'(y) \frac{\partial y}{\partial \sigma} \left[1 + \nabla_0 \widehat{\sigma} \cdot R(y_0)\right] = 0 \ . \tag{9}$$

Eqn. 9 will be satisfied if the function $\widehat{\sigma}(y_0)$ satisfies the differential equation

$$1 + \nabla_0 \widehat{\sigma} \cdot R(y_0) = 0 \ . \tag{10}$$

We can also define the escape location $\widehat{y}(y_0) = y\left(y_0, \widehat{\sigma}(y_0)\right)$. Differentiating with respect to y_0 yields

$$\nabla_0 \widehat{y} = \nabla_0 y + \frac{\partial y}{\partial \sigma} \nabla_0 \widehat{\sigma} \ . \tag{11}$$

Multiplying both sides of equation (11) by $R(y_0)$ and applying equations (6) and (10) leads to the homogeneous differential equation

$$\nabla_0 \widehat{y} \, R(y_0) = \nabla_0 y \, R(y_0) + \frac{\partial y}{\partial \sigma} \nabla_0 \widehat{\sigma} \cdot \ R(y_0) = 0 \ . \tag{12}$$

Using a similar argument and equations (7) and (10) the escape value $\widehat{u}(y_0) = u\left(y_0, \widehat{\sigma}(y_0)\right)$ can be shown to satisfy the equation

$$\nabla_0 \widehat{u} \cdot R(y_0) + r(y_0) = 0 \ . \tag{13}$$

Eqns. 12-13 have the following boundary conditions: $\widehat{y}(y_0) = y_0$ and $\widehat{u}(y_0) = 0$ if y_0 lies on $\partial\mathcal{D}$, and the corresponding characteristic flows out of $\mathcal{D}$.

These are the static "escape" equations (12-13) that we will numerically approximate to solve the multiple arrival problem. We note that the functions $\widehat{u}(y_0)$, $\widehat{y}(y_0)$, and $\widehat{\sigma}(y_0)$ provide values for u, y, and σ respectively obtained at the boundary for a trajectory starting at the point in phase space y_0. The equations that describe these functions are linear and possess unique solutions for given boundary conditions.

Escape Equations

$$\begin{aligned} 1 + \nabla_0 \widehat{\sigma} \cdot R(y_0) &= 0 \\ \nabla_0 \widehat{y} \, R(y_0) &= 0 \\ \nabla_0 \widehat{u} \cdot R(y_0) + r(y_0) &= 0 \end{aligned} \tag{14}$$

To summarize, rather than compute in physical space x, we have transformed the problem to phase-space $y = (x, p)$. However, there are three distinct differences between our approach and the typical characteristic ray-tracing approach to computing in phase space:

- First, we have transformed the problem into a linear, time-independent partial differential equations with the differentiation with respect to the initial conditions. In the next section, we present an unconditionally stable, fast technique for computing the solution to these equations.
- Second, and equally importantly, the solution to these equations can be constructed without regards to any particular boundary conditions and/or placement of sources. Unlike a ray-tracing approach, which starts with a given source (or sources) on the boundary, and must recompute the entire solution each time these sources are changed, in this formulation this may be done in postprocessing once the essential solution has been obtained. As an example, once the $\{\widehat{u}(y_0), \widehat{y}(y_0), \widehat{\sigma}(y_0)\}$ field has been computed, we are then free to extract first arrivals, later arrivals, arrivals at a particular source on the boundary, etc.
- Third, if one has a particular known set of source boundary conditions in advance, this may be incorporated into the formulation to limit the parts of phase space which can contain the solution set, thus reducing the compute time.

3.4 Example: The Eikonal equation

As an example, which will serve as our computational test, the Hamiltonian for the Eikonal equation can be written in the form

$$H(x,p) = \frac{1}{2} p \cdot p - \frac{1}{2} n^2(x) = 0 \,, \tag{15}$$

where $n(x)$ corresponds to the wave slowness (refraction index). In accordance with equations 3, the functions $R(y)$ and $r(y)$ are specified in this case to be

$$R(y) = \begin{pmatrix} \nabla_p H \\ -\nabla_x H \end{pmatrix} = \begin{pmatrix} p \\ n(x) \nabla n \end{pmatrix} \tag{16}$$

$$r(y) = p \cdot \nabla_p H = p \cdot p = n^2(x) \,. \tag{17}$$

Correspondingly, the escape equations transform to

$$\nabla_{x_0} \widehat{y}\, p_0 + n(x_0)\, \nabla_{p_0} \widehat{y}\, \nabla_{x_0} n = 0 \tag{18}$$

$$\nabla_{x_0} \widehat{u} \cdot p_0 + n(x_0)\, \nabla_{p_0} \widehat{u} \cdot \nabla_{x_0} n + n^2(x_0) = 0. \tag{19}$$

3.4.1 Constant slowness

In the case when the refraction index n does not depend on the position, the escape equations take a simpler form

$$\nabla_{x_0} \widehat{y}\, p_0 = 0; \qquad \nabla_{x_0} \widehat{u} \cdot p_0 + n^2 = 0. \tag{20}$$

For a more specific example, let us consider the case of the region $\mathcal{D}$ defined by $z > 0$ and $p_z < 0$, where z is the first component of x_0, and p_z is the first component of p_0. The escape surface $z = 0$ corresponds to the Earth surface in Geophysics, while the condition $p_z < 0$ selects the up-going waves. Eqn. 20 is then supplied with the boundary conditions

$$\widehat{y}\,|_{z=0} = y_0; \qquad \widehat{u}\,|_{z=0} = 0. \tag{21}$$

The analytical solution of the problem 20-21 then takes the form $\widehat{z} = 0$, $\widehat{x}_i = x_i - \frac{p_i}{p_z} z$, $\widehat{p} = p_0$, and $\widehat{u} = -\frac{n^2 z}{p_z}$, where x_i is the ith component of x_0 with $i > 1$, and p_i is the corresponding component of p_0.

3.4.2 One-dimensional slowness

Another specific example of the eikonal equation is the case of a one-dimensional slowness function. If the boundary conditions take the form of Eqn.21, while the slowness $n(x)$ only depends on the first component of x, then Eqn. 19 simplifies to

$$\frac{\partial \widehat{u}}{\partial z} p_z + n'(z)\, n(z)\, \frac{\partial \widehat{u}}{\partial p_z} + n^2(z) = 0 \ . \tag{22}$$

Letting $p_h = \sqrt{n^2(z) - p_z^2}$, Eqn. 22 becomes

$$-\frac{\partial \widehat{u}}{\partial z} \sqrt{n^2(z) - p_h^2} + n^2(z) = 0 \tag{23}$$

whose analytical solution is

$$\widehat{u} = \int_0^z \frac{n^2(\xi)\, d\xi}{\sqrt{n^2(\xi) - p_h^2}} \ . \tag{24}$$

Using similar transformations, the escape equations (18) resolve to $\widehat{z} = 0$, $\widehat{x}_i = x_i + \int_0^z \frac{p_i\, d\xi}{\sqrt{n^2(\xi) - p_h^2}}$, $\widehat{p}_z = -\sqrt{n^2(0) - p_h^2}$, and $\widehat{p}_i = p_i$.

There formulas are well-known in the theory of Lagrangian ray tracing in one-dimensional media [6]. We obtained them here as the solution of the escape PDEs for the Eulerian formulation of the same problem.

3.4.3 Amplitude computation

A common use of the eikonal equation is for describing the wave propagation in the high-frequency asymptotics [6]. Here, the eikonal equation is often supplemented with the amplitude transport equation for computing not only the wavefront position but also the corresponding wave amplitude.

In the high-frequency asymptotics, the leading order amplitude contribution comes from the geometrical spreading: a measure of the ray tube focusing effect. We can extract the geometrical spreading information directly from the phase-space solution of the escape PDEs. Indeed, according to the known formula [18, 19], the squared amplitude is inversely proportional to

$$A^{-2} \sim \left| \det \left[\nabla_{p_i} \widehat{x}_i \right] p_z\, \widehat{p}_z \right| \ , \tag{25}$$

where p_z and $\widehat{p}_z$ are the components of p_0 and $\widehat{p}$, normal to the observational boundary, and p_i and $\widehat{x}_i$ correspond to the remaining components.

In the case of multiple arrivals, we can use the simple access to the amplitude information, provided by formula (25) to select the most energetic arrival from all the branches of the solution.

4 Numerical Algorithm

The numerical algorithm proposed in [15] is to solve Eqns. 18,19 for a numerical computations of traveltimes on a fixed x, z grid.[2] Although both $\widehat{u}(y_0)$ and $\widehat{y}(y_0)$ are strictly single-valued, we later can extract from them the possibly multi-valued traveltimes from every grid point x, z to a point on the boundary.

4.1 The Geometry of the Solution

We illustrate the geometry behind the algorithm through an instructive example. Consider a square, and suppose we wish to find the time $\widehat{u}(x, z, \theta)$ at which a ray leaving the initial point (x, z) inside the square, initially moving in direction θ, hits the boundary. We assume that the slowness field $n(x, z)$ is given. First, note that the set

[2] For ease of explanation in this section, we switch to the notation in which a physical domain is given by x, z, and we let θ be the angle between the vertical and the vector p.

$\widehat{u}(x,z,\theta) = T$, drawn in x, z, θ space, gives the set of all initial positions and directions which reach the boundary of the square at time T. By the uniqueness of characteristics, the set of all points paramterized by T and given by

$$\widehat{U}(T) = \{x, z, \theta \ \mid \ \widehat{u}(x,z,\theta) = T\} \tag{26}$$

sweep out the solution space. In Figure 1, we show the solution surfaces $\widehat{u}(x,z,\theta)$ for the collapsing square.

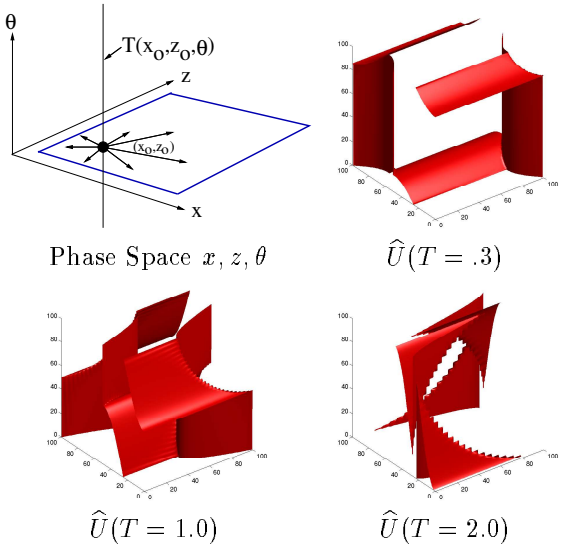


Figure 1: Geometry of Solution

We begin by noting that Eqn. 19 is a linear equation for the arrival field $\widehat{u}$ and thus we know the characteristic direction at every point of the phase space x, p. Thus, we can develop a one-pass algorithm designed in the spirit of of Fast Marching Methods [28] and Ordered Upwind Methods [31]; we march the solution surface outwards from the boundary, using the characteristic direction to update grid values. This imposes a natural ordering on the grid points: to update a grid value, the algorithm traces the corresponding characteristic until its [first] intersection with a cell, for which all of the grid values have been already computed. We note that, due to the linearity of Eqns 14, the characteristic directions are known in advance, and the grid-point ordering is considerably easier to achieve than in the case of the non-linear optimal control PDEs discussed in [31].

We note that there is considerable previous work on using characteristics for transport equations; see [23] for an early example and [20] for a recent parallel implementation.

4.2 Algorithm

Consider a discretization of phase space; in two space dimensions, this can be written as $\widehat{u}_{ijk}$, where the indices i, j, k run over x, z, and p respectively. Following the terminology of Fast Marching Methods [28] and Ordered Upwind Methods [31, 32], the nodes are divided into three classes: *Far* (no information about the correct value of $\widehat{u}_{ijk}$ is known), *Accepted* (the correct value of $\widehat{u}_{ijk}$ has been computed), and *Considered* (adjacent to *Accepted*), for which $\widehat{u}_{ijk}$ has already been computed, but may be corrected by a later computation.

We note that standing at any grid point, one can compute the value $\widehat{u}_{ijk}$ using a ray tracing method, which traces back along the characteristic to the initial boundary If we were to use this algorithm to compute $\widehat{u}_{ijk}$ at each grid point, we would have a cell-based Lagrangian ray-trace method. Instead, we use this technique to build the hypersurfaces incrementally in an outwards fashion.

Start with all nodes in *Far*. Put the boundary nodes in *Accepted*, and put all nodes adjacent to *Accepted* in *Considered*. Each *Considered* node is given a value by using a local cell characteristic method. If the characteristic at that node does not point back to the boundary, we assign it a large value. Then the algorithm proceeds as follows (see Figure 2).

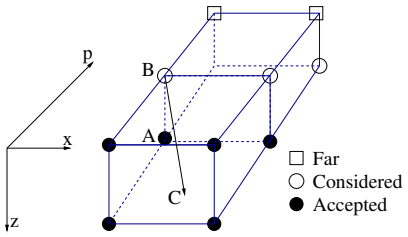


Figure 2: Point A has just been *Accepted*: *Considered* point B is updated by tracing characteristic back to point C and interpolating from *Accepted* values.

Algorithm

1. Take the *Considered* node with the smallest value for $\widehat{\sigma}_{ijk}$ and make it accepted.
2. Find the octant that the characteristic direction going through that grid point points toward
3. For each neighboring grid point in that octant which is not *Accepted*, use the discrete cell characteristic update to compute a (possibly new) value for $\widehat{u}_{ijk}$ as follows (and convert any such point which is *Far* to *Considered*): Trace backwards along the characteristic to intersect a cell face:
   (a) If all four values of that cell face are *Accepted*, use the interpolated value of $\widehat{u}_{ijk}$ at the intersection point, plus the update of $\widehat{u}_{ijk}$ along the drawn characteristic to produce the tentative value at ijk.
   (b) If no points on that cell face are *Accepted*, do not update the value
   (c) If more than one but less than four values of that cell face are *Accepted*, continue tracing along the characteristic until either (a) or (b) situation is encountered.
   (d) Compute $\widehat{y}_{ijk}$ in parallel with $\widehat{u}_{ijk}$.
4. Loop to (1) until all points are *Accepted*.

For higher accuracy, more than four points are used if available as *Accepted*. Any ODE solver (Runge-Kutta, multistep, etc.) can be employed for local characteristic tracing [6]. For efficiency, we use the method of local parabolic ray tracing, based on Taylor expansion of the trajectory near the starting point. The parabolic rays are continued until they hit the cell face, which requires a solution of a simple quadratic equation. Analogous cell-based ray tracing methods were developed in different context in [4, 21].

4.3 Boundary Conditions, Post-Processing, and Timings

4.3.1 Calculating all Trajectories

Suppose we wish to find the exit time and position for all possible trajectories in the interior. For sake of exposition, we consider the constant slowness Eikonal equation for a two-dimensional square region with N points on each side, yielding N^2 points in the interior. Discretizing over N possible phase angles, this yields a computational phase space domain of N^3 points. On the boundary of the square, the initial values of both $\widehat{\sigma}$ and $\widehat{u}$ for all trajectories that point out of the domain are set to ∞. Thus, we ordered upwind march through all of phase space, resulting in an algorithm of complexity $O(N^3)$. We store the dependency tree as the solution is computed.

We can then perform efficient post-processing to extract the results for any particular collection of sources; given any source, the dependence tree from that source determines the subset of phase space which brackets the trajectories that reach that source, and the traceback has a lower order operation count. A coarse-fine grid approach adds to the efficiency.

4.3.2 Calculating Particular Trajectories

Suppose we know in advance that we wish to find the exit time and position for trajectories which reach a particular subset of the boundary. As an example, suppose we view the boundary as the initial position of a curve propagating inwards. As boundary conditions, we can then set the initial values of both $\widehat{\sigma}$ and $\widehat{u}$ to ∞ for all trajectories which do not start orthogonal to this boundary (since the angles are discretized, we bracket the orthogonal direction). This may limit the part of phase space that contains the solution.

4.4 Timings

Finally, the timings are as follows. Consider a constant slowness field; in this case the characteristic backtrace stops in one cell. Using a Pentinum III, on a 50^3 mesh in three-dimensional phase space, the entire calculation takes 2.7 seconds; on 100^3, 20.5 seconds, and on a 100^3 mesh, 150 seconds. This verifies the $O(N \log N)$ nature of the algorithm, when the backtrace operation is counted as one unit operation.

Further details about the algorithm may be found in [16].

5 Results

5.1 Computation of Escape Solution

We begin in Figure 3 with a calculation of the full phase-space arrival field in a rectangular region. The slowness field corresponds to a Gaussian distribution around the center with peak slower than the surroundings (see Figure 5). On the upper left, we show the arrival position $\widehat{x}$ along the top wall ($z = 0$) of arrivals starting at a fixed slice $z = 0.9$ through the phase-space x, z, p cube. The vertical axis is the starting value p_o (actually, $\sin\theta$), and the horizontal axis is the starting value x_o; the color scale (shown in the bar on the right) is the arrival position $\widehat{x}$. On the upper right, we show the value of the arrival time $\widehat{u}$.

On the lower left, again limited to a fixed z slice, we show the x (horizontal axis) and p positions (vertical axis) of all trajectories that exit at the point $\widehat{x} = .5, z = 0$. Finally, in the figure on the lower right, again for a fixed z slice, we show the travel times $\widehat{u}$ required to reach the source exit point $\widehat{x} = .5, z = 0$, plotted against the *physical* space x (horizontal axis). Here, one sees the multiple arrival structure as expected, bending around the central slowness field and crossing over itself. To verify the accuracy of our methods, in Figure 3 we show the traveltime computed from Runge-Kutta Lagrangian ray tracing with a fine step, which should emulate the exact solution.

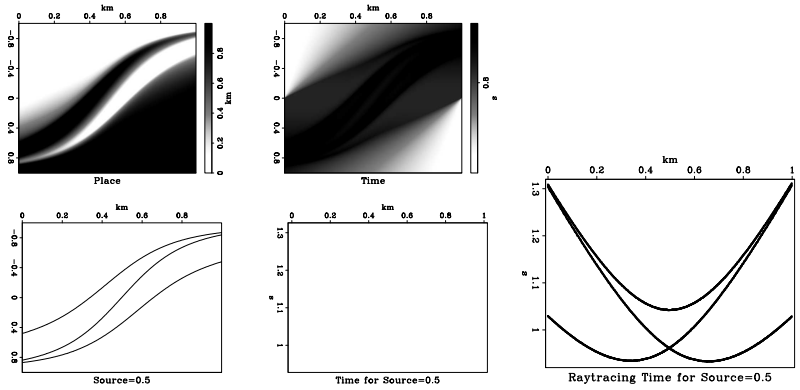


Figure 3: Computation of Escape Solutions

5.2 Extracting Multiple Arrivals

As a different and perhaps more geometrically familiar example, in Figure 4 we show the equiarrivals curves, which are the set of all points in physical space whose trajectories reach the boundary normal to the source distribution at the same time. In this case, the source distribution is the boundary of the entire square, thus we produce the non-viscosity multiply-sheeted solution of a square propagating inward with unit speed. Again, we stress that the calculation need not be repeated to obtain equiarrivals from a different set of sources.

In this case, the multiple arrivals are extracted as follows: the front emanating from the boundary passes through the point x_o, z_o at time $\widehat{u} = T_{crit}$ if T_{crit} is a critical point of $T(x_o, z_o, p)$, where differentiation is taken with respect to p. Thus, we locate the critical points of $\widehat{u}$ with respect to the variable p to determine the arrival front in the domain space.

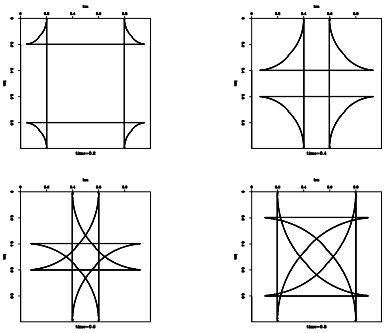
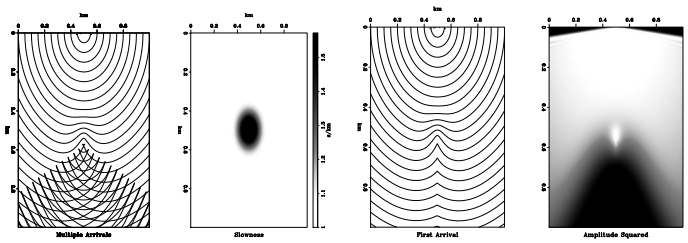


Figure 4: Multiple Arrivals from Square

5.3 Extracting Most Energetic Arrivals

As the last example, we show how to compute the most energetic arrival of the Eikonal equation from among all the potential arrivals using the amplitude computation discussed earlier. First, in Figure 5, the left pair shows all the arrivals starting from a source at the center of the top wall, together with the slowness field on the right (darker is slower). The right pair shows the first arrival and on the amplitude of the displayed arrival (the lighter the tone, the more amplitude).



All arrivals First arrival

Figure 5: All vs. First arrivals

In contrast, in Figure 6 we show the most energetic arrival and the corresponding amplitude field.
More details about the numerical algorithms and further calculations may be found in [16].

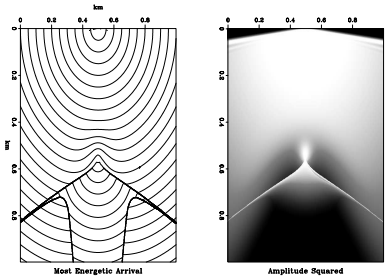


Figure 6: Most Energetic Arrival

6 Appendix: Liouville PDE

We wish to derive Eqns. 6, 7 [3]. First, we will consider the function

$$F(y_0, \sigma) = \frac{\partial y}{\partial \sigma} - \nabla_0 y \, R(y_0) = R(y) - \nabla_0 y \, R(y_0) \tag{27}$$

and prove that it is identically zero.

We note that at $\sigma = 0$, $y = y_0$ (the initial condition), $\nabla_0 y = 1$, and therefore $F(y_0, 0) = 0$. Next, differentiating F with respect to σ, we obtain

$$\begin{aligned} \frac{\partial F}{\partial \sigma} &= R'(y) \, \frac{\partial y}{\partial \sigma} - \nabla_0 \frac{\partial y}{\partial \sigma} \, R(y_0) \\ &= R'(y) \, R(y) - \nabla_0 R(y) \, R(y_0) \\ &= R'(y) \, [R(y) - \nabla_0 y \, R(y_0)] \\ &= R'(y) F(y_0, \sigma) \end{aligned} \tag{28}$$

As a function that satisfies a linear differential equation (28) and has the zero initial condition, $F(y_0, \sigma)$ must be zero for all σ, thus $F(y_0, \sigma) = 0$. According to the definition (27), this is equivalent to (6).

Equation (7) can be proved in an analogous way. Consider the function

$$G(y_0, \sigma) = \frac{\partial u}{\partial \sigma} - \nabla_0 u \cdot R(y_0) - r(y_0) = r(y) - r(y_0) - \nabla_0 u \cdot R(y_0) \; . \tag{29}$$

At $\sigma = 0$, $G(y_0, 0)$ is zero due to the zero initial conditions on u.

Additionally, at any σ,

$$\begin{aligned} \frac{\partial G}{\partial \sigma} &= r'(y) \cdot \frac{\partial y}{\partial \sigma} - \nabla_0 \frac{\partial u}{\partial \sigma} \cdot R(y_0) \\ &= r'(y) \cdot R(y) - \nabla_0 r(y) \cdot R(y_0) \\ &= r'(y) \cdot [R(y) - \nabla_0 y \, R(y_0)] \\ &= r'(y) \cdot F(y_0, \sigma) = 0 \; , \end{aligned}$$

Hence, $G(y_0, \sigma)$ is identically equal to zero, which is equivalent to equation (7).

Finally, we point out that solutions of the Liouville equations are transported along the same characteristics as solutions of the Hamilton Eqn. 1, only with the trajectory parameter running backward. This reflects the reciprocity of the Hamiltonian system: initial conditions are restored by backward ray tracing.

Acknowledgements: We thank A. Chorin, O. Hald, M. Popovici, J. Rausch, W. Symes and A. Vladimirsky for valuable discussions concerning Liouville equations and multiple arrivals.

[3] O. Hald developed this derivation of the Liouville equation, and we follow it closely.

References

[1] Audebert, F., Nichols, D., Rekdal, T., Biondi, B., Lumley, D. E. and Urdaneta, H., 1997, Imaging complex geologic structure with single-arrival Kirchhoff prestack depth migration: Geophysics, 62, 1533-1543.

[2] Benamou, J.D., *Big ray tracing : multivalued travel time field computation using viscosity solutions of the eikonal equation*, J. Comp. Physics, 128, 463-474, 1996

[3] Proceedings of the WORKSHOP on the COMPUTATION OF MULTI-VALUED TRAVELTIMES, Ed. Benamou, J-D., INRIA, Sept. 16-18 1996.

[4] Bernasconi, G. and Drufuca, G., *3-D traveltimes and amplitudes by gridded rays*, Geophysics, 66, 277-282, 2001.

[5] Brandsberg-Dahl, S., de Hoop, M. V. and Ursin, B., *Focusing in dip and AVA compensation on scattering-angle/azimuth common image gathers*, submitted to Geophysics, 2001.

[6] Cerveny, V. Seismic Ray Theory, Cambridge University Press, 2001.

[7] Chorin, A.,J., Hald, O., Kuperfman, R., *Optimal prediction and the Mori-Zwanzig representation of irreversible processes*, Proc. Nat. Acade., Sciences, USA, 97, pp. 2968-2973, 2000.

[8] Chorin, A.,J., Hald, O., Kuperfman, R., *Non-Markovian optimal prediction*, Monte-Carlo Meth. & Appl.,7, pp. 99-109, 2001.

[9] Crandall, M.G. & Lions, P-L., *Viscosity Solutions of Hamilton-Jacobi Equations*, Tran. AMS, 277, pp. 1-43, 1983.

[10] Crandall, M.G., Evans, L.C. & Lions, P-L., *Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations*, Tran. AMS, 282, pp. 487-502, 1984.

[11] Dijkstra, E.W., *A Note on Two Problems in Connection with Graphs*, Numerische Mathematik, 1, pp. 269-271, 1959.

[12] Duistermaat, J.,J., (1974), Oscillatory integrals, Lagrange immersions and unfolding of singularities, Commum. Pure Appl.Maths., Vol.27, 207-281.

[13] Engquist, B., Runborg, O., and Tornberg, A-K., *The Segment Projection Method for Geometrical Optics*, work in progress, 2001.

[14] Evans, L.C., Partial Differential Equations, American Mathematical Society, 1998.

[15] Fomel, S., and Sethian, J.A., *Multiple Arrivals using Liouville Equations*, LBL Technical Report #48682, April, 2001.

[16] Fomel, S., and Sethian, J.A., *Escape Equations for Computing Multi-Valued Solutions to Static Hamilton-Jacobi Equations*, to be submitted for publication, Sept. 2001.

[17] Geoltrain, S. and Brac, J., 1993, Can we image complex structures with first-arrival traveltime?: Geophysics, 58, 564-575.

[18] Goldin, S. V., *Seismic traveltime inversion*, Soc. Expl. Geophys., Tulsa., 1986.

[19] Gritsenko, S. A., 1984, *Time field derivatives*: Soviet Geology and Geophysics, 25, no. 4, 103-109, 1984.

[20] Plimpton, S., Hendrickson, B., Burns, S., and McLendon, W. *Parallel Algorithms for Radiation Transport on Unstructured Grids*, Proceedings from Supercomputing '00, 2001.

[21] Klimes, L., *Grid travel-time tracing: Second-order method for the first arrivals in smooth media*, PAGEOPH, 148, 539-563, 1996.

[22] Lambare, G., Lucio, P.S. and Hanyga A., *Two-dimensional multivalued traveltime and amplitude maps by uniform sampling of a ray field*, Geophys. J. Int., 125, 584-598, 1996..

[23] Lasaint, P., and Raviart, P.A., *Finite Element Methods and the Neutron Transport Equation*, in Mathematical Aspects of Finite Element Partial Differential Equations, Ed. De Boor, C., Academic Press, London, 1974.

[24] Maslov, V.P., and Fedoriuk, M.V., (1981), Semi-classical approximations in quantum mechanics, Reidel, Dordrecht.

[25] Ruuth, S., Merriman, B., and Osher, S., *A Fixed Grid Method for Capturing the Motion of Self-Intersecting Wavefronts and Related PDEs*, J. Comp. Phys., 163, 1, pp. 1-21, 2000.

[26] Sanz-Serna, J.M. and Calvo, M.P. Numerical Hamiltonian Problems, CRC Press, 1994.

[27] Sava, P. and Fomel, S., *3-D traveltime computation using Huygens wavefront tracing*, Geophysics, 66, 883-889, 2001.

[28] Sethian, J.A., *A Fast Marching Level Set Method for Monotonically Advancing Fronts*, Proc. Nat. Acad. Sci., 93, 4, pp. 1591-1595, February 1996.

[29] Sethian, J.A. & Popovici, M., *Three Dimensional Traveltimes Computation Using the Fast Marching Method*, Geophysics, 64, 2, 1999.

[30] Sethian, J.A., Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences, Cambridge University Press, 1999.

[31] Sethian, J.A., and Vladimirsky, A., *Ordered Upwind Methods for Static Hamilton-Jacobi Equations*, Proceedings of the National Academy of Sciences, 98, pp. 11069-11074, 2001.

[32] Sethian, J.A., and Vladimirsky, A., *Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms*, Center for Pure and Applied Mathematics Technical Report PAM-792, University of California, Berkeley; submitted for publication, SIAM Journal Numerical Analysis, July 2001.

[33] Steinhoff, J., Fan, M., Wang, L., *A New Eulerian Method for the Computation of Propagating Short Acoustic and Electromagnetic Pulses*, J. Comp. Phys., 157, 2, pp. 683-706, 2000.

[34] Symes, W.W., *A slowness matching finite difference algorithm for traveltimes beyond transmission caustics*, Proceedings, 68th Annual International Meeting of the Society of Exploration Geophysicists, New Orleans, Expanded astract, (1998).

[35] Sethian, J.A., *Fast Marching Methods*, SIAM Review, Vol. 41, No. 2, pp. 199-235, 1999.

[36] Xu, S., Chauris, H., Lambare, G. and Noble, M. S., 1998, Common angle image gather: A new strategy for imaging complex media, 68th Ann. Internat. Mtg: Soc. of Expl. Geophys., 1538-1541.

[37] Vinje, V., Iversen, E. and Gjoystdal, H., *Traveltime and amplitude estimation using wavefront construction*, Geophysics, 58, 1157-1166, 1993.