

A FAST EIKONAL EQUATION SOLVER USING THE SCHRÖDINGER WAVE EQUATION

KARTHIK S. GURUMOORTHY[†] AND ANAND RANGARAJAN[†]

Abstract. We use a Schrödinger wave equation formalism to solve the eikonal equation. We show that a solution to the eikonal equation is obtained in the limit as $\hbar \rightarrow 0$ of the solution to the corresponding linear Schrödinger equation. The Schrödinger equation corresponding to the eikonal turns out to be a generalized, screened Poisson equation. Despite being linear, it does not have a closed-form solution for arbitrary forcing functions. We use a standard perturbation analysis approach to derive a new algorithm which is guaranteed to converge provided the forcing function is bounded and positive. The perturbation technique requires a sequence of discrete convolutions which can be performed in $O(N \log N)$ using the Fast Fourier Transform (FFT) where N is the number of grid points. A major advantage of our approach over most other methods is that we do not require a spatial discretization of gradient operators and this contributes to the increased accuracy of our technique. The solution to the eikonal solution is recovered from the exponent of the wave function. Since the wave function is computed for a small but non-zero \hbar , the obtained solution is an approximation. We provide evidence for the usefulness of our technique by comparing the results of our approach with those obtained from Hamilton-Jacobi solvers such as the fast sweeping algorithm and the Dijkstra single source shortest path algorithm as well as with exact solutions. The latter are available for the Euclidean distance function problem—a special case of the eikonal when the forcing function is set to one.

Key words. eikonal equation, Schrödinger wave equation, perturbation theory, geometric series, Fast Fourier Transform (FFT), forcing function, discrete convolution, screened Poisson equation, Green's function

AMS subject classifications. 65Z05, 35L05

1. Introduction. The *eikonal* (from the Greek word *εικον* or “image”) equation is traditionally encountered in the wave and geometric optics literature where the principal concern is the propagation of light rays in an inhomogeneous medium [7]. Its twin roots are in wave propagation theory and in geometric optics. In wave propagation theory, it is obtained when the wave is approximated using the Wentzel–Kramers–Brillouin (WKB) approximation [23]. In geometric optics, it can be derived from Huygen’s principle [2]. In the present day, the eikonal equation has outgrown its humble optics origins and now finds application in far flung areas such as electromagnetics [23], robot motion path planning [6] and image analysis [21].

The eikonal equation is a nonlinear, first order, hyperbolic partial differential equation [26] of the form

$$(1.1) \quad \|\nabla S(X)\| = f(X), \quad X \in \Omega$$

subject to the boundary condition $S|_{\partial\Omega} = U(X)$, where Ω is an open subset of \mathbb{R}^D . The forcing function $f(X)$ is a positive valued function and ∇ denotes the gradient operator. In the special case where f equals one, the solution to the eikonal equation is the Euclidean distance function [21]. Detailed discussions on the existence and uniqueness of the solution can be found in [11].

While the eikonal equation is venerable and classical, it is only in the last twenty years that we have seen the advent of pioneering numerical methods aimed at solving this problem. To name a few are the well known fast marching [22, 24] and fast

[†]Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611-6120, USA, E-mail: {ksg,anand}@cise.ufl.edu, Phone: +1 352 392 1507, Fax: +1 352 392 1220.

sweeping [28] methods. Algorithms based on discrete structures such as the well known Dijkstra single source shortest path algorithm [10] can also be adapted to solve this problem. When we seek solutions on a discretized spatial grid with N points, the complexity of the fast marching method is $O(N \log N)$ while that of the fast sweeping method appears (empirically) to be $O(N)$ and therefore both of these efficient algorithms have seen widespread use since their inception. Typically, eikonal solvers grow the solution from a set of K seed points at which the solution is known. The complexity of the Dijkstra single source shortest path algorithm is $O(KN \log N)$ which is less efficient than the fast marching and fast sweeping methods. However, the Dijkstra algorithm is not based on solving a differential equation with a spatial discretization of the differential operator as is the case with the fast marching and fast sweeping methods and therefore has the potential to be more accurate.

The eikonal equation can also be derived from a variational principle; Fermat’s principle of least time which states that “Nature always acts by the shortest paths” [3]. From this variational principle, the theoretical physics developmental sequence proceeds as follows: The first order Hamilton’s equations of motion are derived using a Legendre transformation of the variational problem wherein new momentum variables are introduced. Subsequently, a canonical transformation converts the time varying momenta into constants of the motion. The Hamilton-Jacobi equation emerges from the canonical transformation [15]. In the Hamilton-Jacobi formalism specialized to the eikonal problem, we seek a surface $S(X, t)$ such that its increments are proportional to the speed of the light rays. This is closely related to Huygen’s principle and thus marks the *rapprochement* between geometric and wave optics [2]. It is this nexus that drives numerical analysis methods [22, 28] (focused on solving the eikonal equation) to base their solutions around the Hamilton-Jacobi formalism.

So far, our development has followed that of classical physics. Since the advent of quantum theory—specifically the Schrödinger wave equation—in the 1920s, the close relationship between the Schrödinger and Hamilton-Jacobi equations has been intensely studied [5]. Eschewing the role of probabilities, we are mainly interested in the quantum to classical transition as $\hbar \rightarrow 0$ where the nonlinear Hamilton-Jacobi equation emerges from the phase of the Schrödinger wave equation. This relationship has found very few applications in the numerical analysis literature despite being well known. In this paper, we leverage the important distinction between the Schrödinger and Hamilton-Jacobi equations, namely, that the former is *linear* whereas the latter is not. We take advantage of the linearity of the Schrödinger equation while exploiting its relationship to Hamilton-Jacobi and derive computationally efficient solutions to the eikonal equation.

A time-independent Schrödinger wave equation at the energy state E has the form $\psi(X, t) = \phi(X) \exp(\frac{iEt}{\hbar})$, ($\hbar \equiv \frac{h}{2\pi}$) [16], where $\phi(X)$ —the stationary state wave function—is the eigenstate of the Hamiltonian operator H corresponding to the eigenvalue E . When the Hamilton-Jacobi scalar field S^* appears as the exponent of the stationary state wave function, specifically $\phi(X) = \exp(\frac{-S^*(X)}{\hbar})$, and if $\phi(X)$ satisfies the Schrödinger equation, namely $H\phi = E\phi$, we show that as $\hbar \rightarrow 0$, S^* satisfies the Hamilton-Jacobi equation. Note that in the above, a nonlinear Hamilton-Jacobi equation is obtained in the limit as $\hbar \rightarrow 0$ of a linear Schrödinger equation which is novel from a numerical analysis perspective. Consequently, instead of solving the Hamilton-Jacobi equation, one can solve its Schrödinger counterpart (taking advantage of its linearity), and compute an approximate S^* for a suitably small value of \hbar . This computational procedure is approximately equivalent to solving the original

Hamilton-Jacobi equation.

Since the efficient solution of a linear wave equation is the cornerstone of our approach, we now briefly describe the actual computational algorithm used. We derive the static Schrödinger equation for the eikonal problem. The result is a generalized, *screened Poisson* equation [13] whose solution is known at K seed points. This linear equation does not have a closed-form solution and therefore we resort to a perturbation method [12] of solution—which is very closely related to the Born expansion [20]. The perturbation method comprises a sequence of multiplications with a space-varying forcing function followed by convolutions with a (modified) Green’s function (for the screened Poisson operator) which we solve using an efficient $[O(N \log N)]$ fast Fourier transform (FFT)-based technique [9, 4]. Perturbation analysis involves a geometric series approximation for which we show convergence for all bounded forcing functions.

The paper is organized as follows. In Section 2, we provide the Hamilton-Jacobi formulation for the eikonal equation as adopted by the fast sweeping and fast marching methods. In Section 3, we derive the Schrödinger wave equation from first principles for the eikonal equation by following Feynman’s path integral approach [14]. In Section 4, we provide an efficient FFT-based numerical technique for solving the Schrödinger equation. In Section 5, we provide results and comparisons (to fast sweeping and Dijkstra) where we differentiate between experiments focused on the Euclidean distance function problem (where the forcing function is a constant) and experiments devoted to the general eikonal equation. We conclude in Section 6.

2. Hamilton-Jacobi formulation for the eikonal equation.

2.1. Fermat’s principle of least time. It is well known that the Hamilton-Jacobi equation formalism for the eikonal equation can be obtained by considering a variational problem based on Fermat’s principle of least time [2] which in $2D$ is

$$(2.1) \quad I[q] = \int_{t_0}^{t_1} f(q_1, q_2, t) \sqrt{1 + \dot{q}_1^2 + \dot{q}_2^2} dt.$$

The variational problem defined above has its roots in geometric optics [7]. Consider a medium with refractive index $f(t, q_1, q_2)$ and let a space curve γ be described by $(q_1(t), q_2(t), t)$, $t \in [t_0, t_1]$. Fermat’s least time principle states that the necessary condition for γ to be a light ray is that I be an extremum. The quantity

$$(2.2) \quad L(q_1, q_2, \dot{q}_1, \dot{q}_2, t) = f(q_1, q_2, t) \sqrt{1 + \dot{q}_1^2 + \dot{q}_2^2}$$

is called the *optical Lagrangian*. The generalized momenta are defined as [15]

$$(2.3) \quad p_i \equiv \frac{\partial L}{\partial \dot{q}_i} = \frac{f \dot{q}_i}{\sqrt{1 + \dot{q}_1^2 + \dot{q}_2^2}}.$$

We now solve for \dot{q}_i as a function of p_i by first noting that

$$(2.4) \quad p_1^2 + p_2^2 - f^2 = \frac{-f^2}{1 + \dot{q}_1^2 + \dot{q}_2^2}$$

from which it follows that

$$(2.5) \quad \dot{q}_i = \frac{p_i}{\sqrt{f^2 - p_1^2 - p_2^2}}$$

A Legendre transformation [2] is used to obtain the Hamiltonian which is given by

$$(2.6) \quad H(q_1, q_2, p_1, p_2, t) = \sum_{i=1}^2 p_i \dot{q}_i - L(q_1, q_2, \dot{q}_1, \dot{q}_2, t)$$

where \dot{q}_i is as in (2.5). Hence

$$(2.7) \quad \begin{aligned} H(q_1, q_2, p_1, p_2, t) &= \frac{p_1^2 + p_2^2}{\sqrt{f^2 - p_1^2 - p_2^2}} - \frac{f^2}{\sqrt{f^2 - p_1^2 - p_2^2}} \\ &= -\sqrt{f^2 - p_1^2 - p_2^2} \end{aligned}$$

The Hamilton-Jacobi equation is obtained via a canonical transformation [15] of the Hamiltonian. In the $2D$ case, it is

$$(2.8) \quad \frac{\partial S}{\partial t} + H(q_1, q_2, \frac{\partial S}{\partial q_1}, \frac{\partial S}{\partial q_2}, t) = 0$$

where we have replaced the generalized momentum variables p_i with

$$(2.9) \quad \frac{\partial S}{\partial q_i} = p_i = \frac{\partial L}{\partial \dot{q}_i}.$$

Hence, for the given variational problem, the Hamilton-Jacobi equation is

$$(2.10) \quad \frac{\partial S}{\partial t} - \sqrt{f^2 - \left(\frac{\partial S}{\partial q_1}\right)^2 - \left(\frac{\partial S}{\partial q_2}\right)^2} = 0$$

i.e.,

$$(2.11) \quad \left(\frac{\partial S}{\partial t}\right)^2 + \left(\frac{\partial S}{\partial q_1}\right)^2 + \left(\frac{\partial S}{\partial q_2}\right)^2 = f^2$$

which can be succinctly written as $\|\nabla S\|^2 = f^2$. This equation is the familiar eikonal equation (1.1).

2.2. An equivalent variational principle. We now take an idiosyncratic approach to derive the eikonal equation by considering a different variational problem which is still very similar to Fermat's least time principle. The advantage of this variational formulation is that the corresponding Schrödinger wave equation can be easily obtained.

Consider the following variational problem namely,

$$(2.12) \quad I[q] = \int_{t_0}^{t_1} \frac{1}{2}(\dot{q}_1^2 + \dot{q}_2^2) f^2(q_1, q_2) dt$$

where the forcing term f is assumed to be independent of time and the Lagrangian L is defined as

$$(2.13) \quad L(q_1, q_2, \dot{q}_1, \dot{q}_2, t) \equiv \frac{1}{2}(\dot{q}_1^2 + \dot{q}_2^2) f^2(q_1, q_2).$$

Again, defining

$$(2.14) \quad p_i \equiv \frac{\partial L}{\partial \dot{q}_i} = f^2(q_1, q_2) \dot{q}_i$$

and applying the Legendre transformation [2], we can define the Hamiltonian of the system in $2D$ as

$$(2.15) \quad H(q_1, q_2, p_1, p_2, t) = \frac{1}{2} \frac{(p_1^2 + p_2^2)}{f^2(q_1, q_2)}.$$

From a canonical transformation of the Hamiltonian [15], we obtain the following Hamilton-Jacobi equation

$$(2.16) \quad \frac{\partial S}{\partial t} + \frac{1}{2} \frac{\left(\frac{\partial S}{\partial q_1}\right)^2 + \left(\frac{\partial S}{\partial q_2}\right)^2}{f^2(q_1, q_2)} = 0$$

Since the Hamiltonian in (2.15) is a constant *independent* of time, equation (2.16) can be simplified to the static Hamilton-Jacobi equation. By separation of variables, we get

$$(2.17) \quad S(q_1, q_2, t) = S^*(q_1, q_2) - Et$$

where E is the total energy of the system and $S^*(q_1, q_2)$ is called the Hamilton's characteristic function [2]. Observing that $\frac{\partial S}{\partial q_i} = \frac{\partial S^*}{\partial q_i}$, equation (2.16) can be rewritten as

$$(2.18) \quad \frac{1}{2} \left[\left(\frac{\partial S^*}{\partial q_1}\right)^2 + \left(\frac{\partial S^*}{\partial q_2}\right)^2 \right] = Ef^2.$$

Choosing the energy E to be $\frac{1}{2}$, we obtain

$$(2.19) \quad \|\nabla S^*\|^2 = f^2$$

which is the original eikonal equation (1.1). S^* is the required Hamilton-Jacobi scalar field which is efficiently obtained by the fast sweeping [28] and fast marching methods [22].

3. Schrödinger wave equation formalism for the eikonal equation. In this section, for the sake of completeness, we derive a Schrödinger equation for our idiosyncratic variational problem (2.12) from first principles and then recover the scalar field S^* [as in (2.17)] from the wave function.

3.1. A path integral derivation of the Schrödinger equation. We follow the Feynman path-integral approach [14, 25] to deriving the differential equation for the wave function ψ . Please note that the Feynman path integral is not considered mathematically rigorous in the general setting. However, it provides a constructive approach to deriving the Schrödinger equation and hence we include it here. The key idea is to consider the transition amplitude $K(x, t_2, \xi, t_1)$ where KK^* corresponds to the conditional transitional probability density of a particle going from $\xi(t_1)$ to $x(t_2)$. For any specific path $x(t) = \{x_1(t), x_2(t)\}$ in $2D$, the amplitude is assumed to be proportional to

$$(3.1) \quad \exp\left(\frac{i}{\hbar} \int_{t_1}^{t_2} L(x, \dot{x}, t) dt\right)$$

where the Lagrangian L is given by (2.13). If the particle can move from ξ to x over a set of paths, the transition amplitude is defined as the “sum” of the amplitudes associated with each path, so

$$(3.2) \quad K(x, t_2; \xi, t_1) = \int \exp\left(\frac{i}{\hbar} \int_{t_1}^{t_2} L(x, \dot{x}, t) dt\right) \mathcal{D}x.$$

Now suppose that a particle is moving from a starting position $x + \xi = (x_1 + \xi_1, x_2 + \xi_2)$ at time t and ends at x at time $t + \tau$ while traveling for a very short time interval τ . Using the definition of the Lagrangian from (2.13), the transition amplitude for this event is

$$(3.3) \quad \begin{aligned} K(x, t + \tau; x + \xi, t) &= \int \exp\left(\frac{i}{\hbar} \int_t^{t+\tau} \frac{1}{2}(x_1^2 + x_2^2) f^2(x_1, x_2) dt'\right) \mathcal{D}x \\ &\approx \int \exp\left\{\frac{i\tau}{2\hbar} \left[\left(\frac{\xi_1}{\tau}\right)^2 + \left(\frac{\xi_2}{\tau}\right)^2\right] f^2(x_1, x_2)\right\} \mathcal{D}x \\ &= M \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\}. \end{aligned}$$

Here $M \equiv \int \mathcal{D}x$. In order to derive the wave equation for ψ , we first recall that the wave function ψ has an interpretation that $\psi^*\psi = |\psi(x, t)|^2$ denotes the probability of finding a particle at x and at time t . Since K behaves more like a conditional transitional probability from $x + \xi$ to x , the wave function should satisfy

$$(3.4) \quad \psi(x, t + \tau) = \int K(x, t + \tau; x + \xi, t) \psi(x + \xi, t) d\xi.$$

where K is given by (3.3).

Expanding to the first order in τ and second order in ξ we get

$$(3.5) \quad \begin{aligned} \psi + \tau \frac{\partial \psi}{\partial \tau} &= \int M \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} \\ &\quad \left[\psi + \xi_1 \frac{\partial \psi}{\partial x_1} + \xi_2 \frac{\partial \psi}{\partial x_2} + \frac{\xi_1^2}{2} \frac{\partial^2 \psi}{\partial x_1^2} + \frac{\xi_2^2}{2} \frac{\partial^2 \psi}{\partial x_2^2} + \xi_1 \xi_2 \frac{\partial^2 \psi}{\partial x_1 \partial x_2}\right] d\xi \\ &= I_1 \psi + I_2 \frac{\partial \psi}{\partial x_1} + I_3 \frac{\partial \psi}{\partial x_2} + I_4 \frac{1}{2} \frac{\partial^2 \psi}{\partial x_1^2} + I_5 \frac{1}{2} \frac{\partial^2 \psi}{\partial x_2^2} + I_6 \frac{\partial^2 \psi}{\partial x_1 \partial x_2} \end{aligned}$$

where the integrals $I_1, I_2, I_3, I_4, I_5, I_6$ are defined as

$$(3.6) \quad \begin{aligned} I_1 &\stackrel{\text{def}}{=} M \int \int \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} d\xi_1 d\xi_2, \\ I_2 &\stackrel{\text{def}}{=} M \int \int \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} \xi_1 d\xi_1 d\xi_2, \\ I_3 &\stackrel{\text{def}}{=} M \int \int \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} \xi_2 d\xi_1 d\xi_2, \\ I_4 &\stackrel{\text{def}}{=} M \int \int \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} \xi_1^2 d\xi_1 d\xi_2, \\ I_5 &\stackrel{\text{def}}{=} M \int \int \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} \xi_2^2 d\xi_1 d\xi_2, \\ I_6 &\stackrel{\text{def}}{=} M \int \int \exp\left\{\frac{i}{2\hbar\tau} (\xi_1^2 + \xi_2^2) f^2(x_1, x_2)\right\} \xi_1 \xi_2 d\xi_1 d\xi_2. \end{aligned}$$

Observing that integral I_2 can be rewritten as a product of two integrals, i.e.,

$$(3.7) \quad I_2 = M \int \exp \left\{ \frac{i}{2\hbar\tau} \xi_1^2 f^2(x_1, x_2) \right\} \xi_1 d\xi_1 \int \exp \left\{ \frac{i}{2\hbar\tau} \xi_2^2 f^2(x_1, x_2) \right\} d\xi_2$$

and that the first integral is an *odd* function of ξ_1 , it follows that $I_2 = 0$. A similar argument shows that $I_3 = I_6 = 0$.

Using the relation

$$(3.8) \quad \int_{-\infty}^{\infty} \exp(i\alpha s^2) ds = \sqrt{\frac{i\pi}{\alpha}}$$

and noticing that I_1 can be written as a separate product of two integrals in ξ_1 and in ξ_2 , it follows that

$$(3.9) \quad I_1 = M \frac{i2\hbar\tau\pi}{f^2(x_1, x_2)}.$$

In order for the equation for ψ to hold, I_1 should approach 1 as $\tau \rightarrow 0$. Hence,

$$(3.10) \quad M = \frac{f^2(x_1, x_2)}{i2\hbar\tau\pi}.$$

Let I denote the integral (3.8). Then

$$(3.11) \quad \frac{1}{i} \frac{\partial I}{\partial \alpha} = \frac{i}{2} \frac{\sqrt{i\pi}}{\alpha^{\frac{3}{2}}} = \int \exp(i\alpha s^2) s^2 ds.$$

Using the relations in equations (3.8) and (3.11) with $\alpha = \frac{f^2}{2\hbar\tau}$ and writing I_4 and I_5 as product of two separate integrals in ξ_1 and ξ_2 and substituting the value of M from (3.10), we obtain

$$(3.12) \quad I_4 = I_5 = \frac{i\hbar\tau}{f^2(x_1, x_2)}.$$

Substituting back the value of these integrals in (3.5), we get

$$(3.13) \quad \psi + \tau \frac{\partial \psi}{\partial \tau} = \psi + \frac{i}{2} \frac{\hbar\tau}{f^2} \left(\frac{\partial^2 \psi}{\partial x_1^2} + \frac{\partial^2 \psi}{\partial x_2^2} \right)$$

from which we obtain the Schrödinger wave equation [16]

$$(3.14) \quad i\hbar \frac{\partial \psi}{\partial t} = H\psi$$

where the Hamiltonian operator H is given by

$$(3.15) \quad H = -\frac{\hbar^2}{2f^2} \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \right).$$

3.2. Obtaining the eikonal from Schrödinger . Since the Hamiltonian H doesn't explicitly depend on time, using separation of variables $\psi(x, t) = \phi(x)g(t)$, we get

$$(3.16) \quad \frac{\hbar \dot{g}}{i g} = \frac{\hbar^2 \nabla^2 \phi}{2f^2 \phi} = E$$

where E is the energy state of the system and ∇^2 is the Laplacian operator. By choosing the energy to be $\frac{1}{2}$ as before, we get

$$(3.17) \quad g(t) = \exp\left(\frac{it}{2\hbar}\right)$$

and hence

$$(3.18) \quad \psi(x, t) = \phi(x) \exp\left(\frac{it}{2\hbar}\right)$$

where $\phi(x)$ satisfies the equation

$$(3.19) \quad \hbar^2 \nabla^2 \phi = f^2 \phi.$$

which is a generalized, screened Poisson equation [13] (since $f > 0$ is an *arbitrary* speed function). We now show that when $\phi = \exp(\frac{-S^*}{\hbar})$ and satisfies (3.19), S^* asymptotically satisfies the eikonal equation (1.1) as $\hbar \rightarrow 0$. We show this for the 2D case but the generalization to higher dimensions is straightforward.

When $\phi(x_1, x_2) = \exp(\frac{-S^*(x_1, x_2)}{\hbar})$, the first partials of ϕ are

$$(3.20) \quad \frac{\partial \phi}{\partial x_1} = -\frac{1}{\hbar} \exp\left(\frac{-S^*}{\hbar}\right) \frac{\partial S^*}{\partial x_1}, \quad \frac{\partial \phi}{\partial x_2} = -\frac{1}{\hbar} \exp\left(\frac{-S^*}{\hbar}\right) \frac{\partial S^*}{\partial x_2}.$$

The second partials required for the Laplacian are

$$(3.21) \quad \begin{aligned} \frac{\partial^2 \phi}{\partial x_1^2} &= \frac{1}{\hbar^2} \exp\left(\frac{-S^*}{\hbar}\right) \left(\frac{\partial S^*}{\partial x_1}\right)^2 - \frac{1}{\hbar} \exp\left(\frac{-S^*}{\hbar}\right) \frac{\partial^2 S^*}{\partial x_1^2}, \\ \frac{\partial^2 \phi}{\partial x_2^2} &= \frac{1}{\hbar^2} \exp\left(\frac{-S^*}{\hbar}\right) \left(\frac{\partial S^*}{\partial x_2}\right)^2 - \frac{1}{\hbar} \exp\left(\frac{-S^*}{\hbar}\right) \frac{\partial^2 S^*}{\partial x_2^2}. \end{aligned}$$

From this, equation (3.19) can be rewritten as

$$(3.22) \quad \left(\frac{\partial S^*}{\partial x_1}\right)^2 + \left(\frac{\partial S^*}{\partial x_2}\right)^2 - \hbar \left(\frac{\partial^2 S^*}{\partial x_1^2} + \frac{\partial^2 S^*}{\partial x_2^2}\right) = f^2$$

which in simplified form is

$$(3.23) \quad \|\nabla S^*\|^2 - \hbar \nabla^2 S^* = f^2.$$

The additional $\hbar \nabla^2 S^*$ term [relative to (1.1)] is referred to as the viscosity term [11, 22] which emerges naturally from the Schrödinger equation derivation—an intriguing result. Since $|\nabla^2 S^*|$ is bounded, as $\hbar \rightarrow 0$, (3.23) tends to

$$(3.24) \quad \|\nabla S^*\|^2 = f^2$$

which is the original eikonal equation (1.1). This relationship motivates us to solve the linear Schrödinger equation (3.19) instead of the non-linear eikonal equation and then compute the scalar field S^* via

$$(3.25) \quad S^*(X) = -\hbar \log \phi(X).$$

4. Efficient computation of the solution to the Schrödinger equation.

In this section, we provide numerical techniques for efficiently solving the Schrödinger equation.

4.1. Perturbation theory. In this paper, we are mainly concerned with solving the eikonal equation for the case where we know the solution at a given set of K seed points. To this end, we begin by considering the forced version of (3.19) which is

$$(4.1) \quad -\hbar^2 \nabla^2 \phi + f^2 \phi = \sum_{k=1}^K \delta(X - Y_k).$$

The points $Y_k, k \in \{1, \dots, K\}$ can be considered to be the set of locations which encode initial knowledge about the scalar field S^* , say for example $S^*(Y_k) = 0, \forall Y_k, k \in \{1, \dots, K\}$. The case where the initial conditions depend on the seed index is a straightforward generalization: if $S^*(Y_k) = d_k$, then $\phi(Y_k) = \exp\{-\frac{d_k}{\hbar}\}$ and the delta functions on the right of (4.1) are suitably weighted.

We would like to bring out the fact that for the Euclidean distance function problem where the forcing term f is identically equal to 1, closed form solutions for ϕ (in $1D, 2D, 3D$) exist [18]. Closed form solutions can be obtained even for constant forcing functions¹, but not for arbitrary functions f which is what we face in the case of the eikonal. Consequently, we propose to solve the linear system (4.1) using *perturbation* theory [12].

We are interested in obtaining the solution for S^* at the given N grid locations. Since it is meaningful to assume that $S^*(X)$ goes to infinity for points at infinity, from relation (3.25), it follows that we can use Dirichlet boundary conditions $\phi(X) = 0$ at the boundary of an unbounded domain. Assuming that f is close to a constant *non-zero forcing function* \tilde{f} , equation (4.1) can be rewritten as

$$(4.2) \quad (-\hbar^2 \nabla^2 + \tilde{f}^2) \left[1 + (-\hbar^2 \nabla^2 + \tilde{f}^2)^{-1} \circ (f^2 - \tilde{f}^2) \right] \phi = \sum_{k=1}^K \delta(X - Y_k).$$

Now, defining the operator L as

$$(4.3) \quad L = (-\hbar^2 \nabla^2 + \tilde{f}^2)^{-1} \circ (f^2 - \tilde{f}^2)$$

and ϕ_0 as

$$(4.4) \quad \phi_0 = (1 + L)\phi$$

we see that ϕ_0 satisfies

$$(4.5) \quad (-\hbar^2 \nabla^2 + \tilde{f}^2)\phi_0 = \sum_{k=1}^K \delta(X - Y_k)$$

and

$$(4.6) \quad \phi = (1 + L)^{-1}\phi_0.$$

Let $c_0 = \|L\|_{op}$ —the operator norm—defined by

$$(4.7) \quad c_0 = \sup\{\|Lg\|, \forall \text{functions } g \in \text{Hilbert space } \mathcal{H} \text{ with } \|g\| \leq 1\}$$

¹In which case, equation (4.1) becomes the screened Poisson equation.

where \mathcal{H} is the space of square integrable functions on \mathbb{R}^D , i.e, $g \in \mathcal{H}$ iff

$$(4.8) \quad \int g^2 d\mu < \infty,$$

which then provides a natural function norm $\|g\|$ for the function $g \in \mathcal{H}$ given by

$$(4.9) \quad \|g\|^2 = \int g^2 d\mu.$$

If $c_0 < 1$, we can approximate $(1+L)^{-1}$ using the first few $T+1$ terms of the geometric series to get

$$(4.10) \quad (1+L)^{-1} \approx 1 - L + L^2 - L^3 + \dots + (-1)^T L^T$$

where the operator norm of the difference can be bounded by

$$(4.11) \quad \|(1+L)^{-1} - \sum_{i=0}^T (-1)^i L^i\|_{op} \leq \sum_{i=T+1}^{\infty} \|L^i\|_{op} \leq \sum_{i=T+1}^{\infty} c_0^i = \frac{c_0^{T+1}}{1-c_0}$$

which converges to 0 exponentially in T . We would like to point out that the above geometric series approximation is very similar to a Born expansion used in scattering theory [20]. We now derive an upper bound for c_0 .

Let $L = A_1 \circ A_2$ where $A_1 \equiv (-\hbar^2 \nabla^2 + \tilde{f}^2)^{-1}$ and $A_2 \equiv f^2 - \tilde{f}^2$. We now provide an upper bound for $\|A_1\|_{op}$. Let \mathcal{B} denote a closed unit ball in the Hilbert space \mathcal{H} , i.e

$$(4.12) \quad \mathcal{B} = \{g \in \mathcal{H} : \|g\| \leq 1\}.$$

Since we assume Dirichlet boundary conditions at the boundary, there exists a unique $z \in \mathcal{H}$ [11] such that

$$(4.13) \quad (-\hbar^2 \nabla^2 + \tilde{f}^2)z = g.$$

Since $g \in \mathcal{B}$, we see that

$$(4.14) \quad \begin{aligned} \|(-\hbar^2 \nabla^2 + \tilde{f}^2)z\|^2 &= \|-\hbar^2 \nabla^2 z\|^2 + \|\tilde{f}^2 z\|^2 + 2\hbar^2 \tilde{f}^2 \langle -\nabla^2 z, z \rangle \\ &= \|g\|^2 \leq 1. \end{aligned}$$

Using integration by parts and the boundary conditions, it can be shown that

$$(4.15) \quad \langle -\nabla^2 z, z \rangle = \int |\nabla z|^2 d\mu \geq 0.$$

Using the above relation in (4.14), we then observe that

$$(4.16) \quad \|z\| = \|A_1(g)\| \leq \frac{1}{\tilde{f}^2}, \forall g \in \mathcal{B}.$$

As we now see that $A_1(\mathcal{B})$ is bounded and less than or equal to $\frac{1}{\tilde{f}^2}$, we immediately have

$$(4.17) \quad \|A_1\|_{op} \leq \frac{1}{\tilde{f}^2}.$$

Now, let $M = \sup\{|f^2 - \tilde{f}^2|\}$. Then, for any function $g \in \mathcal{B}$

$$(4.18) \quad \|(f^2 - \tilde{f}^2)g\|^2 = \int (f^2 - \tilde{f}^2)^2 g^2 d\mu \leq M^2 \|g\|^2 \leq M^2$$

and hence from (4.7),

$$(4.19) \quad \|A_2\|_{op} \leq M = \sup\{|f^2 - \tilde{f}^2|\}.$$

Since $\|L\|_{op} \leq \|A_1\|_{op} \|A_2\|_{op}$, from equations (4.17) and (4.19), we observe that

$$(4.20) \quad c_0 = \|L\|_{op} \leq \frac{\sup\{|f^2 - \tilde{f}^2|\}}{\tilde{f}^2}.$$

It is worth commenting that the bound for c_0 is *independent* of \hbar . So, if we guarantee that $\frac{\sup\{|f^2 - \tilde{f}^2|\}}{\tilde{f}^2} < 1$, the geometric series approximation for $(1 + L)^{-1}$ (4.10) converges for *all* values of \hbar .

4.2. Deriving a bound for convergence of the perturbation series. Interestingly, for any positive, bounded forcing function f , i.e. $f(X) > 0^2$, by defining $\tilde{f} = \sup\{f(X)\}$, we observe that $|f^2 - \tilde{f}^2| < \tilde{f}^2$. From equation (4.20), we immediately see that $c_0 < 1$. This proves the existence of \tilde{f} for which the geometric series approximation (4.10) is *always* guaranteed to converge for any positive forcing function f . The choice of \tilde{f} can then be made prudently by defining it to be the value that *minimizes*

$$(4.21) \quad F(\tilde{f}) = \frac{\sup\{|f^2 - \tilde{f}^2|\}}{\tilde{f}^2}.$$

This in turn minimizes the operator norm c_0 , thereby providing a better geometric series approximation for the inverse (4.10).

Let $f_{min} = \min\{f(X)\}$ and let $f_{max} = \max\{f(X)\}$. We now show that $F(\tilde{f})$ attains its minimum at

$$(4.22) \quad \tilde{f} = \nu = \sqrt{\frac{f_{min}^2 + f_{max}^2}{2}}.$$

Case(i): If $\tilde{f} < \nu$, then $\sup\{|f^2 - \tilde{f}^2|\} = f_{max}^2 - \tilde{f}^2$. Clearly,

$$(4.23) \quad \frac{f_{max}^2 - \tilde{f}^2}{\tilde{f}^2} > \frac{f_{max}^2 - \nu^2}{\nu^2}.$$

Case(ii): If $\tilde{f} > \nu$, then $\sup\{|f^2 - \tilde{f}^2|\} = \tilde{f}^2 - f_{min}^2$. It follows that

$$(4.24) \quad \frac{\tilde{f}^2 - f_{min}^2}{\tilde{f}^2} = 1 - \frac{f_{min}^2}{\tilde{f}^2} > 1 - \frac{f_{min}^2}{\nu^2}.$$

We therefore see that $\tilde{f} = \nu = \sqrt{\frac{f_{min}^2 + f_{max}^2}{2}}$ is the optimal value.

²If $f(X) = 0$, then the velocity $v(X) = \frac{1}{f(X)}$ becomes ∞ at X . Hence it is reasonable to assume $f(X) > 0$.

Now, using the above approximation for $(1 + L)^{-1}$ (4.10) and the definition of L from (4.3), we obtain the solution for ϕ as

$$(4.25) \quad \phi = \phi_0 - \phi_1 + \phi_2 - \phi_3 + \dots + (-1)^T \phi_T$$

where ϕ_i satisfies the recurrence relation

$$(4.26) \quad (-\hbar^2 \nabla^2 + \tilde{f}^2) \phi_i = (f^2 - \tilde{f}^2) \phi_{i-1}, \quad \forall i \in \{1, 2, \dots, T\}.$$

4.3. Green's functions in 1D, 2D and 3D. Each ϕ_i can be computed efficiently by observing that the equations (4.5) and (4.26) are inhomogeneous, screened Poisson equations, very similar to the inhomogeneous Helmholtz equation except for a sign change [13]. Following a Green's function approach [1], we can write the expressions for the form of ϕ_i . The Green's function G satisfies the relation

$$(4.27) \quad (-\hbar^2 \nabla^2 + \tilde{f}^2) G = -\delta.$$

The form of the solution for G depends on the number of spatial dimensions. We provide solutions for G in 1D, 2D and 3D.

1D: In 1D, the solution for G [1] over an unbounded domain with vanishing boundary conditions at ∞ is

$$(4.28) \quad G(X, Y) = \frac{1}{2\hbar\tilde{f}} \exp\left(\frac{-\tilde{f}|X - Y|}{\hbar}\right).$$

2D: In 2D, the solution for G [1] over an unbounded domain with vanishing boundary conditions at ∞ is

$$(4.29) \quad \begin{aligned} G(X, Y) &= \frac{1}{2\pi\hbar^2} K_0\left(\frac{\tilde{f}\|X - Y\|}{\hbar}\right) \\ &\approx \frac{\exp\left(\frac{-\tilde{f}\|X - Y\|}{\hbar}\right)}{2\hbar\sqrt{2\pi\hbar\tilde{f}\|X - Y\|}}, \frac{\tilde{f}\|X - Y\|}{\hbar} \gg 0.25 \end{aligned}$$

where K_0 is the modified Bessel function of the second kind.

3D: In 3D, the solution for G [1] over an unbounded domain with vanishing boundary conditions at ∞ is

$$(4.30) \quad G(X, Y) = \frac{1}{4\pi\hbar^2} \frac{\exp\left(\frac{-\tilde{f}\|X - Y\|}{\hbar}\right)}{\|X - Y\|}.$$

The solutions for ϕ_i can then be obtained by convolution

$$(4.31) \quad \phi_0(X) = \sum_{k=1}^K G * \delta(X - Y_k) = \sum_{k=1}^K G(X - Y_k),$$

$$(4.32) \quad \phi_i(X) = G * \left[(f^2 - \tilde{f}^2) \phi_{i-1}\right], \quad \forall i \in \{1, 2, \dots, T\}.$$

Once the ϕ_i are computed, the wave function ϕ can then be determined using the approximation (4.25). The solution for the eikonal equation can be recovered using the relation (3.25).

Based on the nature of the expression for the Green’s function, it is worth emphasizing the following *very* important point. In the limiting case of $\hbar \rightarrow 0$,

$$(4.33) \quad \lim_{\hbar \rightarrow 0} \frac{\exp\left\{-\frac{\tilde{f}\|X\|}{\hbar}\right\}}{c\hbar^d\|X\|^p} = 0, \text{ for } \|X\| \neq 0$$

for c, d and p being constants greater than zero and therefore we see that if we define

$$(4.34) \quad \tilde{G}(X) = \exp\left(\frac{-\tilde{f}\|X\|}{\hbar}\right)$$

then

$$(4.35) \quad \lim_{\hbar \rightarrow 0} |G(X) - \tilde{G}(X)| = 0, \text{ for } \|X\| \neq 0$$

and furthermore, the convergence is *uniform* for $\|X\|$ away from zero. Therefore, $\tilde{G}(X)$ provides a very good approximation for the actual unbounded domain Green’s function as $\hbar \rightarrow 0$. The above observation motivates us to compute the solutions for ϕ_i by convolving with \tilde{G} instead of the actual Green’s function G . We approximate each of these convolutions with the discrete convolution between the functions computed at the given N grid locations. By the convolution theorem [4], a discrete convolution can be obtained as the inverse Fourier transform of the product of two individual transforms which for two $O(N)$ sequences can be performed in $O(N \log N)$ time [9]. Thus, the values of each ϕ_i at the N grid locations can be efficiently computed in $O(N \log N)$ making use of the values of ϕ_{i-1} determined at the earlier step. Thus, the overall time complexity to compute the approximate ϕ using the first few $T + 1$ terms is then $O(TN \log N)$. Taking the logarithm of ϕ then provides an approximate solution to the eikonal equation. The algorithm is adumbrated in Table 4.1

We would like to emphasize that the number of terms (T) used in the geometric series approximation of $(1 + L)^{-1}$ (4.10) is *independent* of N . Using more terms only improves the approximation of this truncated geometric series. From equation (4.11), it is evident that the error incurred due to this approximation converges to zero *exponentially* in T and hence even with a small value of T , we should be able to achieve good accuracy. The choice of T is investigated in the experimental section.

4.4. Solving the eikonal equation in higher dimensions. Using \tilde{G} instead of the bounded domain Green’s function G provides a straightforward generalization of our technique to higher dimensions. Regardless of the spatial dimension, the approximate solution for the eikonal equation S^* can be computed from the wave function ϕ in $O(TN \log N)$ as computing the discrete convolution using FFT [4] is always $O(N \log N)$ [9] *irrespective* of the spatial dimension. This speaks for the scalability of our technique.

4.5. Solution for the Euclidean distance function problem. It is well known in the literature that the solution for the Euclidean distance function problem can be obtained by solving the eikonal equation where the forcing term is identically equal to one, namely

$$(4.36) \quad \|\nabla S^*\| = f = 1.$$

TABLE 4.1
Algorithm for the approximate solution of the eikonal equation

1. Compute the function $\tilde{G}(X) = \exp\left(\frac{-\tilde{f}\|X\|}{\hbar}\right)$ at the grid locations.
2. Define the function $I(X)$ which takes the value 1 at the point-set locations and 0 at other grid locations.
3. Compute the FFT of \tilde{G} and I , namely $\tilde{G}_{FFT}(U)$ and $I_{FFT}(U)$ respectively.
4. Compute the function $H(U) = \tilde{G}_{FFT}(U)I_{FFT}(U)$.
5. Compute the inverse FFT of H which gives $\tilde{\phi}_0(X)$ at the grid locations.
6. Initialize $\tilde{\phi}(X)$ to $\tilde{\phi}_0(X)$.
7. For $i = 1$ to T do
 8. Define $P(X) = \left[f^2(X) - \tilde{f}^2\right] \tilde{\phi}_{i-1}(X)$.
 9. Compute the FFT of P namely $P_{FFT}(U)$.
 10. Compute the function $H(U) = \tilde{G}_{FFT}(U)P_{FFT}(U)$.
 11. Compute the inverse FFT of H defined as $\tilde{\phi}_i(X)$ at the grid locations.
 12. Update $\tilde{\phi}(X) = \tilde{\phi}(X) + (-1)^i \tilde{\phi}_i(X)$.
13. End
14. Take the logarithm of $\tilde{\phi}(X)$ and multiply it by $(-\hbar)$ to get the approximate solution for the eikonal equation at the grid locations.

The corresponding wave function ϕ then satisfies the equation

$$(4.37) \quad \hbar^2 \nabla^2 \phi = \phi.$$

Since f is a constant function ($f \equiv 1$), defining $\tilde{f} = f = 1$ and using the forced version of the equation (4.37), we obtain the solution for ϕ as

$$(4.38) \quad \phi(X) = \phi_0(X) = \sum_{k=1}^K G(X) * \delta(X - Y_k) = \sum_{k=1}^K G(X - Y_k)$$

where $G(X)$ is the desired Green's function whose expression depends upon the spatial dimension. Here $Y_k, k \in \{1, \dots, K\}$ are the given set of input locations. Again, for small values of \hbar , \tilde{G} [defined in equation (4.34)] effectively approximates G , the reasons for which are explained above. ϕ can be computed at the given N grid locations in $O(N \log N)$ by performing the discrete convolution using the FFT [9, 4] as delineated above. The solution for the Euclidean distance function can then be recovered from the wave function using the relation (3.25). The reader may refer to our previous work [18] to get a detailed explanation for computing the Euclidean distance function using the Schrödinger wave equation.

4.6. Numerical issues. In principle, we should be able to apply our technique at very small values of \hbar and obtain highly accurate results. But we noticed our technique tends to deteriorate for \hbar values very close to zero. This is due to the fact that at small values of \hbar (and also at large values of \tilde{f}), $\exp\left(\frac{-\tilde{f}\|X\|}{\hbar}\right)$ drops off very quickly and hence for grid locations which are far away from the point-set, the convolution done using FFT needs to be precise (without round-off error) for the computed distance to be meaningful³. Such high precision support may not

³Due to round of errors in computing convolution with FFT for very low values of \hbar , some values $\tilde{\phi}$ may take on negative values and hence taking the logarithm of $\tilde{\phi}$ to obtain \tilde{S} is meaningless

be available and hence our technique may produce erroneous results at these grid locations for very small values of \hbar .

We would like to emphasize that this problem is *not* inherent to our technique, but due to the lack of high precision support in the software/hardware. Setting \hbar very close to zero will result in extremely small numbers which may demand high precision. A promising start in this general direction are the quad-double and arbitrary precision packages available at Lawrence Berkeley National Labs (LBNL) [27].

Interestingly, for certain small values of \hbar , our technique did produce highly accurate results and outperformed the state of the art fast marching and fast sweeping methods. We corroborate our claim and demonstrate the usefulness of our method with the set of experiments described in the subsequent section.

4.7. Fast multipole methods. It is also worth commenting that many non-FFT based algorithms for solving the Helmholtz equation do exist in the literature, the most recent ones being the fast multipole method [17, 8] and these can usually be adapted to solve screened Poisson equations. (Fast multipole methods seems to have an edge over the classical FFT methods for solving the Helmholtz equation.) Since the FFT based solution is not as critical to our technique, the fast multipole method may allow us to use non-regular grids and may also circumvent the aforementioned problem of choosing \hbar . Furthermore, the fast multipole method is known to run in $O(N \log N)$ and sometimes even in $O(N)$ and hence is comparable in speed to the FFT (if not empirically faster). We have reserved this exploration as part of our future work.

5. Experiments.

5.1. Euclidean distance functions. In this section, we show the efficacy of our technique by computing the approximate Euclidean distance function \tilde{S}^* and comparing it to the actual Euclidean distance function S and the fast sweeping method, first on randomly generated $2D$ and $3D$ point-sets and then on a set of bounded $3D$ grid points.

Example 1: We began with a $2D$ grid consisting of points between $(-30, -30)$ and $(30, 30)$. Hence, the total number of grid points is $N = 61 \times 61 = 3721$. We randomly chose around 1000 grid locations as data points (point-set). Then 50,000 experiments were run for values of \hbar ranging from 0.1 to 0.5 in steps of 0.01. The errorbar plot in Figure 5.1 shows the mean and standard deviation of the *percentage* error at each value of \hbar . The error is less than 0.5% at $\hbar = 0.1$ demonstrating the algorithm’s ability to compute accurate Euclidean distances.

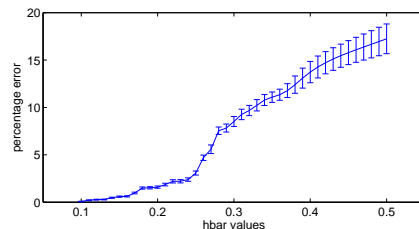


FIGURE 5.1. Percentage error versus \hbar in 50,000 $2D$ experiments.

Example 2: We pitted our technique against the fast sweeping method [28] on a 3D grid consisting of points between $(-8, -8, -8)$ and $(8, 8, 8)$ with the grid space being 0.5. The number of grid points is then $N = 33 \times 33 \times 33 = 35937$. We ran 10,000 experiments randomly choosing 300 grid locations as data points. For our technique, we set \hbar to be 0.15. We ran the fast sweeping algorithm for 12 iterations which was more than sufficient for convergence. The plots in Figure 5.2 show the average absolute difference and the average percentage error calculated as

$$(5.1) \quad \%error = \frac{\text{Average Absolute Diff}}{\text{Average Distance}} * 100$$

for each of these techniques when compared with the true Euclidean distance function. From the plots, we see that our technique provides better accuracy than the fast sweeping method.

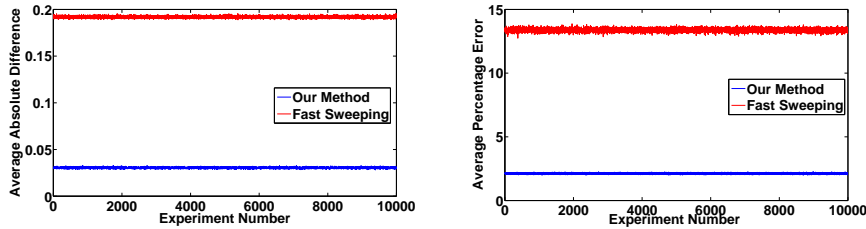


FIGURE 5.2. Average absolute difference (left) and the average percentage error (right) between the true and computed Euclidean distance function for (i) our method (in blue) (ii) fast sweeping (in red) in 10,000 3D experiments.

Example 3: In this example, we computed the distance function on a 2D grid consisting of points between $(-8, -8)$ to $(8, 8)$ with a grid spacing of 0.5, from the boundary of the rectangular region $-2 \leq x \leq 2$ and $-2 \leq y \leq 2$. The number of points on the grid $N = 33 \times 33 = 1089$ and the number of data points on the boundary of this rectangular region was 32. The \hbar value for our technique was set at 0.25. Fast sweeping was run for 5 iterations after which it converged. The true average distance of the grid points from the data points was 3.9275. The average absolute difference, the variance in absolute difference and the percentage error for our method and fast sweeping when compared with the true Euclidean distance function is adumbrated in table 5.1.

TABLE 5.1

Statistics of the Euclidean distance function computed from the boundary of the rectangular region

	Our Method	Fast Sweeping
Average absolute difference	0.1152	0.1673
Variance in absolute difference	0.0180	0.0348
Percentage error	2.9350%	4.2620%

The true Euclidean distance function contour plot and those obtained from our method and fast sweeping is delineated in Figure 5.3.

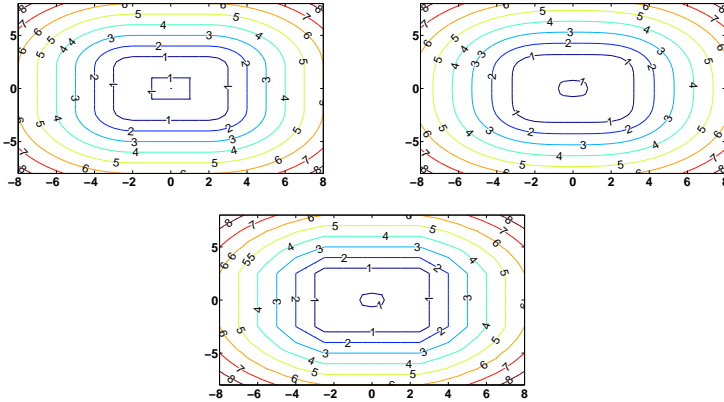


FIGURE 5.3. Contour plots (i) Top Left: True Euclidean distance function (ii) Top Right: Our approach (iii) Bottom: Fast sweeping

Example 4: We took the Stanford bunny dataset⁴ and used the coordinates of the data points on the model as our point-set locations. Since the input data locations need not be at integer locations, we scaled the space uniformly in all dimensions and rounded off the data so that the data lies at integer locations. The input data was also shifted so that it was approximately symmetrically located with respect to the x , y and z axis. We should comment that shifting the data doesn't affect the Euclidean distance function value and uniform scaling of all dimensions is also not an issue, as the distances can be rescaled once they are computed.

After these basic data manipulations, the cardinality of the point set was $K = 3019$ with the data confined to the cubic region $-16 \leq x \leq 16$, $-15 \leq y \leq 15$ and $-12 \leq z \leq 12$. Our grid consisted of the set of all integer locations within this cubic region. The number of grid locations was $N = 25575$. We computed the Euclidean distance function value at each of these grid locations using our technique for different values of \hbar and compared it to the true Euclidean distance function value.

As pointed out earlier, inaccuracies in the floating point computations of very small numbers involved in the FFT operation impedes us from choosing a very small of \hbar . But for those grid locations which are close to the point-set, the accuracy of the computed distance improves as \hbar is decreased. Hence, to circumvent this problem of choosing \hbar , we ran our technique for different values of \hbar and chose the distance function values obtained at large values of \hbar at those grid locations whose average computed distance is larger and vice versa.

When we ran our technique for the set of $\hbar \in \{0.1, 0.2, 0.3, 0.4\}$ and used $\{3, 6, 10\}$ respectively as the threshold of the average computed distance for choosing the appropriate distance function, it gave the following set of results. The maximum absolute difference between the actual and the computed Euclidean distance value over all the grid locations is 0.9066 and the average absolute difference is 0.1322. The accuracy is fairly high since the furthest grid point from the point-set is at a distance of 15.5242 and the average overall distance computed at the grid locations from the point-set is 3.5449. The average error is $\frac{0.1322}{3.5449} * 100 = 3.72\%$. This error can be lowered by using better $O(N \log N)$ numerical algorithms for discrete convolution [27].

⁴This dataset is available at http://www.cc.gatech.edu/projects/large_models/bunny.html

We plotted the isosurface obtained by connecting the grid points which are at a distance of 0.5 from the point set, determined both by the true Euclidean distance function and our technique. Figure 5.4 shows the two surfaces. Notice the similarity between the two plots. It provides anecdotal visual evidence for the usefulness of our approach.

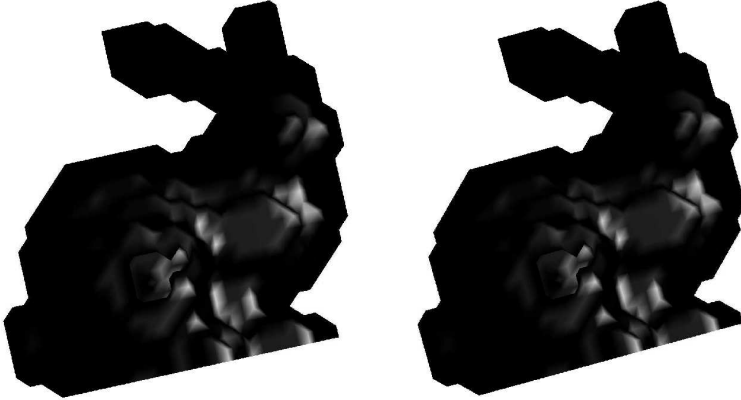


FIGURE 5.4. *Isosurfaces: (i) Left: Actual Euclidean distance function and (ii) Right: Our approach.*

5.2. The general eikonal equation. In this section, we demonstrate the usefulness of our approach by computing the approximate solution to the general eikonal equation (1.1) over a $2D$ and $3D$ grid. Since the eikonal equation cannot be solved in closed form to obtain the true solution, we considered the solution obtained by running Dijkstra’s algorithm [10] to be the ground truth. We ran Dijkstra’s algorithm on a much finer grid where each grid point had 8 neighbors in $2D$ and 26 neighbors in $3D$. The weight on the edge between a pair of grid points was defined to be average of the values of the forcing function at the respective grid points multiplied by either 1 or $\sqrt{2}$ or $\sqrt{3}$ depending on the location of the neighbor. Distances were computed at each grid location by repeatedly considering each data point to be the source node and then the minimum over all these distances was considered to be the true solution for the eikonal equation. This entire process takes $O(KN \log N)$ time. We then compared this solution obtained by running Dijkstra with our method and fast sweeping.

Example 5: In this example, we solve the eikonal equation for the following forcing function

$$(5.2) \quad f(x, y) = 1 + 0.4 \left(e^{-2[(x+0.2)^2+(y+0.2)^2]} - e^{-2[(x-0.2)^2+(y-0.2)^2]} \right)$$

from a point source located at $(x_0, y_0) = (0, 0)$ on a $2D$ grid consisting of points between $(-1.5, -1.5)$ to $(1.5, 1.5)$ with a grid spacing of 0.25. The function $f(x, y)$ is shown in figure 5.5. We defined \tilde{f} as per equation (4.22). For our technique, we set $\tilde{h} = 0.08$ and ran it for 10 iterations. The plot 5.6 shows the maximum difference $\|\tilde{\phi}_i - \tilde{\phi}_{i-1}\|_\infty$ between two consecutive iterations. Fast sweeping showed very good convergence even before 6 iterations. To obtain the ground truth, we ran Dijkstra’s algorithm on a finer grid with a grid spacing of $1/32$. The average absolute difference,

the variance in absolute difference and the percentage error for our method and fast sweeping when compared with the Dijkstra’s algorithm are tabulated in Table 5.2.

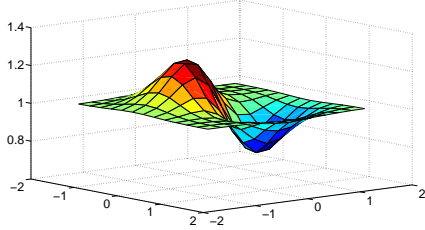


FIGURE 5.5. The forcing function used

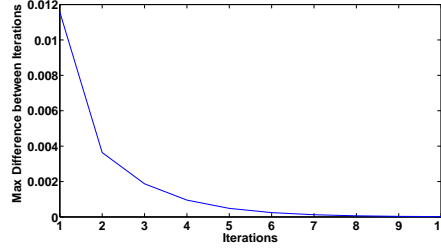


FIGURE 5.6. Maximum difference between two consecutive iterations in our method

TABLE 5.2

Statistics of the eikonal equation solution obtained from using the forcing function defined in equation (5.2)

	Our Method	Fast Sweeping
Average absolute difference	0.0453	0.0529
Variance in absolute difference	0.0029	0.0018
Percentage error	3.5490%	4.1452%

The contour plots obtained by running Dijkstra’s algorithm, our method and fast sweeping are displayed in Figure 5.7. From the result and the plots, we observe that our technique is quite competitive with fast sweeping.

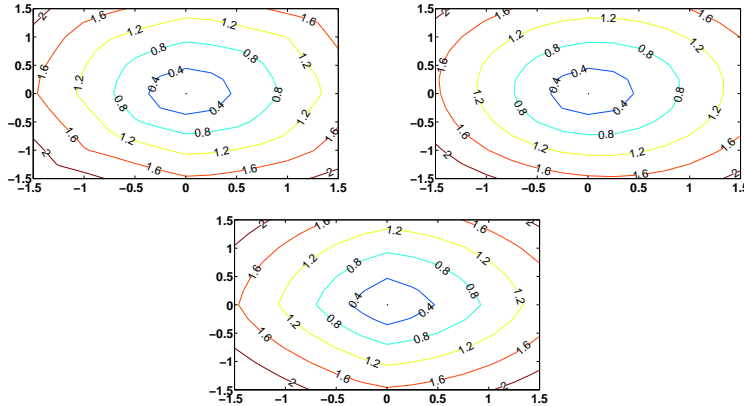


FIGURE 5.7. Contour plots (i) Top Left: Dijkstra’s algorithm (ii) Top Right: Our approach (iii) Bottom: Fast Sweeping

Example 6: In this example, we solve the eikonal equation for the following forcing function

$$(5.3) \quad f(x, y, z) = 1 + 0.3 \sin [\pi(x - 0.5)] \sin [\pi(y + 0.5)] \sin [\pi(z - 0.5)]$$

from the boundary of a cube $-1 \leq x \leq 1$, $-1 \leq y \leq 1$ and $-1 \leq z \leq 1$ on a 3D grid consisting of points between $(-3, -3, -3)$ to $(3, 3, 3)$ with a grid spacing of 0.5. We again defined \tilde{f} according to equation (4.22). This time we set $h = 0.1$ and ran our technique for 20 iterations. The plot in Figure 5.8 shows the maximum difference $\|\tilde{\phi}_i - \tilde{\phi}_{i-1}\|_\infty$ between two consecutive iterations. Fast sweeping converged after 11 iterations. To obtain the ground truth we again ran Dijkstra's algorithm on a finer grid with a grid spacing of 0.125. The average absolute difference, the variance in absolute difference and the percentage error for our method and fast sweeping when compared with Dijkstra's algorithm are tabulated in Table 5.3.

TABLE 5.3

Statistics of the eikonal equation solution obtained from using the forcing function defined in equation (5.3)

	Our Method	Fast Sweeping
Average absolute difference	0.0977	0.2195
Variance in absolute difference	0.0097	0.0304
Percentage error	5.6038%	12.5820%

We plotted the isosurface obtained by connecting the grid points which are at a distance of 0.75 from the boundary of the cube, determined by running Dijkstra algorithm, our approach and fast sweeping. Figures 5.9, 5.10, and 5.11 show the three isosurfaces. Notice the similarity between the plots obtained by running Dijkstra and our method. It again provides anecdotal visual evidence for the usefulness of our approach.

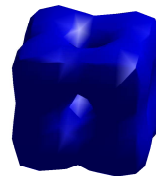
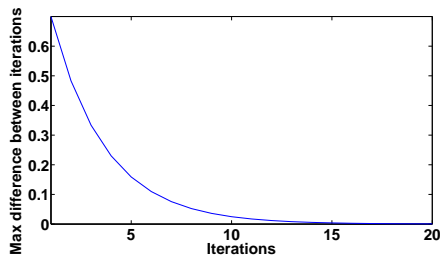


FIGURE 5.8. Maximum difference between two consecutive iterations in our method

FIGURE 5.9. Isosurface from Dijkstra's algorithm

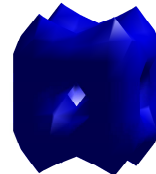
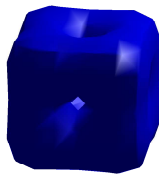


FIGURE 5.10. Isosurface from our approach

FIGURE 5.11. Isosurface from fast sweeping

6. Discussion. In this paper, we have introduced a new approach to solving the non-linear eikonal equation. We have proven that the solution of the eikonal equation can be obtained as a limiting case of the solution to a corresponding linear Schrödinger wave equation. Instead of solving the static Hamilton-Jacobi equation that is the eikonal, the Schrödinger formalism results in a generalized, screened Poisson equation which is solved for very small values of \hbar . Our Schrödinger-based approach follows pioneering Hamilton-Jacobi solvers such as the fast sweeping [28] and fast marching [22, 24] methods with the crucial difference being its linearity. We have developed a fast and efficient perturbation series method which is guaranteed to converge (provided the forcing function f is positive and bounded). Its main advantage is that it completely circumvents the need for spatial discretization of the gradient operator. Hence, we have strong reasons to believe that the accuracy of our method will be better than methods that solve a spatially discretized system of equations. The linearity of the system being solved is highly attractive as well and may lead to even more efficient solutions in the future by drawing on fast, multipole methods [17] and the like. We would like to solve the more general, static Hamilton Jacobi equation in the future, using techniques inspired from quantum mechanics as a counterpart to classical mechanics-based techniques [19]. We expect that the linearity of the Schrödinger equation will result in fast and accurate algorithms even in this more general setting.

REFERENCES

- [1] M. ABRAMOWITZ AND I.A. STEGUN, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Government Printing Office, USA, 1964.
- [2] V.I. ARNOLD, *Mathematical Methods of Classical Mechanics*, Springer, 1989.
- [3] J.-L. BASDEVANT, *Variational Principles in Physics*, Springer, 2007.
- [4] R.N. BRACEWELL, *The Fourier Transform and its Applications*, McGraw-Hill Science and Engineering, 3rd ed., 1999.
- [5] J. BUTTERFIELD, *On Hamilton-Jacobi theory as a classical root of quantum theory*, in Quovadis Quantum Mechanics, A. Elitzur, S. Dolev, and N. Kolenda, eds., Springer, 2005, ch. 13, pp. 239–274.
- [6] J.F. CANNY, *Complexity of robot motion planning*, The MIT Press, 1988.
- [7] G. CHARTIER, *Introduction to Optics*, Springer, 2005.
- [8] H. CHENG, J. HUANG, AND T.J. LEITERMAN, *An adaptive fast solver for the modified Helmholtz equation in two dimensions*, Journal of Computational Physics, 211 (2006), pp. 616–637.
- [9] J.W. COOLEY AND J.W. TUKEY, *An algorithm for the machine calculation of complex Fourier series*, Mathematics of Computation, 19 (1965), pp. 297–301.
- [10] T.H. CORMEN, C.E. LEISERSON, R.L. RIVEST, AND C. STEIN, *Introduction to Algorithms*, The MIT Press, 2nd ed., September 2001.
- [11] M.G. CRANDALL, H. ISHII, AND P.L. LIONS, *User’s guide to viscosity solutions of second order partial differential equations*, Bulletin of the American Mathematical Society, 27 (1992), pp. 1–67.
- [12] F.M. FERNANDEZ, *Introduction to Perturbation Theory in Quantum Mechanics*, CRC Press, 2000.
- [13] A.L. FETTER AND J.D. WALECKA, *Theoretical Mechanics of Particles and Continua*, Dover, 2003.
- [14] R.P. FEYNMAN AND A.R. HIBBS, *Quantum Mechanics and Path Integrals*, McGraw-Hill, 1965.
- [15] H. GOLDSTEIN, C.P. POOLE, AND J.L. SAFKO, *Classical Mechanics*, Addison-Wesley, 3rd ed., 2001.
- [16] D.J. GRIFFITHS, *Introduction to Quantum Mechanics*, Benjamin-Cummings, 2nd ed., 2004.
- [17] N. GUMEROV AND R. DURAISWAMI, *Fast multipole methods for the Helmholtz equation in three dimensions*, Elsevier series in electromagnetism, Elsevier, 2004.
- [18] K. GURUMOORTHY AND A. RANGARAJAN, *A Schrödinger equation for the fast computation of approximate Euclidean distance functions*, in Scale Space and Variational Methods in Computer Vision (SSVM), 2009.

- [19] C.-Y. KAO, S.J. OSHER, AND Y.-H. TSAI, *Fast sweeping methods for static Hamilton-Jacobi equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 2612–2632.
- [20] R.G. NEWTON, *Scattering Theory of Waves and Particles*, Springer-Verlag, New York, 2nd ed., 1982.
- [21] S.J. OSHER AND R.P. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, October 2002.
- [22] S.J. OSHER AND J.A. SETHIAN, *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, 79 (1988), pp. 12–49.
- [23] D.T. PARIS AND F.K. HURD, *Basic Electromagnetic Theory*, McGraw-Hill Education, 1969.
- [24] J.A. SETHIAN, *A fast marching level set method for monotonically advancing fronts*, in Proc. Nat. Acad. Sci, 1996, pp. 1591–1595.
- [25] C.F. STEVENS, *The Six Core Theories of Modern Physics*, The MIT Press, 1996.
- [26] G.B. WHITHAM, *Linear and Nonlinear Waves*, Pure and Applied Mathematics, Wiley-Interscience, 1999.
- [27] Y.HIDA, X.S. LI, AND D.H. BAILEY, *Algorithms for quad-double precision floating point arithmetic*, in Proceedings of the 15th Symposium on Computer Arithmetic, IEEE Computer Society Press, 2001, pp. 155–162.
- [28] H.K. ZHAO, *A fast sweeping method for eikonal equations*, Mathematics of Computation, (2005), pp. 603–627.