

Fast sweeping method for the factored eikonal equation

Sergey Fomel ^{*} Songting Luo [†] Hongkai Zhao [‡]

Abstract

We develop a fast sweeping method for factored eikonal equation. By decomposing the solution of a general eikonal equation as the product of two factors: the first factor is the solution to a simple eikonal equation (such as distance) or a previously computed solution to an approximate eikonal equation. The second factor is a necessary modification/correction. Appropriate discretization and a fast sweeping strategy is designed for the equation of the correction part. The key idea is to enforce the causality of the original eikonal equation during the Gauss-Seidel iterations. Using extensive numerical examples we demonstrate that (1) the convergence behavior of the fast sweeping method for the factored eikonal equation is the same as for the classical eikonal equation, i.e., the number of iterations for the Gauss-Seidel iterations is independent of the mesh size, (2) the numerical solution from the factored eikonal equation is more accurate than the numerical solution directly computed from the classical eikonal equation, especially for point sources.

1 Introduction

The eikonal equation

$$|\nabla T|^2 = S^2(\mathbf{x}) \tag{1}$$

describes the traveltime $T(\mathbf{x})$ of a wave propagating with slowness (refraction index) $S(\mathbf{x})$ in space $\mathbf{x} \in \mathbb{R}^n$. In the case of anisotropic wave propagation, S depends additionally on $\nabla T/|\nabla T|$. When $S(\mathbf{x})$ is equal to one, the traveltime $T(\mathbf{x})$ corresponds to the distance function.

The eikonal equation plays an important role in many practical applications: computer vision, material science, computational geometry, etc. [16]. In seismic

^{*}Jackson School of Geosciences, The University of Texas at Austin, Austin, TX, 78713, sergey.fomel@beg.utexas.edu

[†]Department of Mathematics, University of California, Irvine, CA, 92697, luos@math.uci.edu

[‡]Department of Mathematics, University of California, Irvine, CA, 92697, zhao@math.uci.edu

Research partially supported by NSF grant DMS-0513073, ONR grant N00014-02-1-0090 and DARPA grant N00014-02-1-0603.

imaging in particular, finite-difference solutions of the eikonal equation are used routinely for computing traveltimes tables for numerical modeling and migration of seismic waves [22, 20, 17, 10]. Although limited for computing only first-arrival traveltimes [7], eikonal solvers can be extended in several different ways to image multiple arrivals [3].

In this paper, we derive the factored eikonal equation by assuming that either analytical or numerical solution is available for equation (1) with similar boundary conditions but with different right-hand side. The solution $T(\mathbf{x})$ is then represented as a product of the known solution and an unknown factor, which satisfies the factored equation.

We develop a numerical algorithm based on the fast sweeping method to solve the factored eikonal equation and to evaluate the resultant gain in accuracy. The fast sweeping method (FSM) is an efficient iterative method that uses Gauss-Seidel iterations with alternating orderings to solve a wide range of Hamilton-Jacobi equations and other type of hyperbolic problems [4, 27, 19, 26, 8, 9, 25, 14, 15, 24]. With an appropriate upwind scheme that captures the causality of the underlying partial differential equation, the iteration can converge in a finite number of iterations independent of the mesh size, which was proved for special cases in [26]. The intuition is the following: Information propagates along characteristics. Using a systematic alternating ordering strategy, all directions of characteristics can be divided into a finite number of groups and each group is covered simultaneously by one of the orderings. Moreover, any characteristics can be covered by a finite number of orderings [26]. With an appropriate upwind scheme that enforces the causality of the underlying partial differential equation, a Gauss-Seidel iteration propagates correct information in each updating along characteristics whose directions agree with the orderings.

After outlining the theory and the numerical algorithm, we conduct a series of numerical experiments, where numerical solutions are compared with analytical solutions for model problems. A significant improvement in accuracy is observed in comparison with FSM applied directly to the eikonal equation. Finally, we apply our method to compute traveltimes tables for the benchmark Marmousi model.

2 Factored eikonal equation

A fundamental property of equation (1) is that scaling slowness S by a constant corresponds to scaling traveltimes T by the same constant. This property was used in seismic reflection imaging in the method of common-reflection-point scans [2, 1].

Let us consider a factored decomposition

$$S(\mathbf{x}) = S_0(\mathbf{x}) \alpha(\mathbf{x}) , \tag{2}$$

$$T(\mathbf{x}) = T_0(\mathbf{x}) \tau(\mathbf{x}) \tag{3}$$

and assume that

$$|\nabla T_0|^2 = S_0^2(\mathbf{x}) . \tag{4}$$

If both T_0 and S_0 are known (either from an analytical solution or from a previous numerical computation), one can pose the problem of a numerical evaluation of the correction $\tau(\mathbf{x})$ on a computational grid with appropriate boundary conditions. The function substitutions transform equation (1) to the *factored eikonal equation*

$$T_0^2(\mathbf{x}) |\nabla\tau|^2 + 2T_0(\mathbf{x})\tau(\mathbf{x})\nabla T_0 \cdot \nabla\tau + [\tau^2(\mathbf{x}) - \alpha^2(\mathbf{x})] S_0^2(\mathbf{x}) = 0. \quad (5)$$

When $\alpha(\mathbf{x})$ is constant, the solution of equation (5) is trivial. When $\alpha(\mathbf{x})$ is not a constant but slowly varying, the hope is that accuracy of evaluating $T(\mathbf{x})$ from solving the factored eikonal equation can be greatly improved compared to a direct numerical solution of the original eikonal equation (1). One scenario is that singularities in the original solution $T(\mathbf{x})$ are well captured by $T_0(\mathbf{x})$. So the correction $\tau(\mathbf{x})$ is a smooth function. For example, when computing the traveltimes for a point source the solution is singular at the source. Special treatment, such as using local grid refinement near the source, has to be implemented in order to achieve high order accuracy for the numerical solution to the eikonal equation [13]. However, locally the singularity of the solution to a regular eikonal equation (assuming $S(\mathbf{x})$ is smooth and strictly positive) at a point source is the same as the singularity of the distance function to that point source up to a smooth modification. Numerical tests in Section 4 show that the numerical solution based on the factored eikonal equation can be significantly more accurate than the numerical solution computed directly from the original eikonal equation. Also note that although ∇T_0 does not exist at the source point, it is well defined away from the source point. It provides good approximation of all ray directions near the point source. This is crucial for computing accurate solutions away from the point source, which cannot be approximated easily on a discrete mesh.

When taking $S_0(\mathbf{x}) = 1$, $T_0(\mathbf{x})$ is the distance function. In the case of simple domains and boundary conditions, the distance can be evaluated analytically. For example, the distance from a point source at \mathbf{x}_0 is $T_0(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_0|$, which transforms equation (5) to

$$|\mathbf{x} - \mathbf{x}_0|^2 |\nabla\tau|^2 + 2\tau(\mathbf{x})(\mathbf{x} - \mathbf{x}_0) \cdot \nabla\tau + \tau^2(\mathbf{x}) - S^2(\mathbf{x}) = 0. \quad (6)$$

A numerical solution of equation (6) was investigated previously in geophysical applications [12, 23].

Simple analytical solutions exist for several other particular cases of slowness distributions such a constant gradient of the slowness squared, a constant gradient of the velocity (inverse slowness), etc. [5].

3 Numerical Algorithm

The solution to the eikonal equation (1) is the first arrival time. So the causality, required by FSM, is very simple, the value of T at a grid point should be

determined by its neighbors whose values are smaller. The key point in discretizing equation (5) is to design an upwind scheme following the causality of the original eikonal equation.

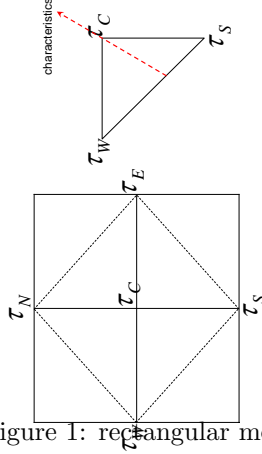


Figure 1: rectangular mesh

We present the algorithm on a rectangular mesh in 2D (the algorithm can be easily extended to higher dimensions). Figure 1 shows an interior grid point C with neighbors W, E, N, S . We discretize equation (5) on four triangles: $\triangle CEN$, $\triangle CNW$, $\triangle CWS$ and $\triangle CSE$. For example, on triangle $\triangle CWS$, we get a discretized equation:

$$T_0^2(C) \left| \left(\frac{\tau_C - \tau_W}{h}, \frac{\tau_C - \tau_S}{h} \right) \right|^2 + 2T_0(C)\tau_C \nabla T_0(C) \cdot \left(\frac{\tau_C - \tau_W}{h}, \frac{\tau_C - \tau_S}{h} \right) + \{\tau_C^2 - \alpha^2(C)\}S_0^2(C) = 0 \quad (7)$$

By solving this equation, we enforce the causality condition.

Causality condition: assume τ_C^h is an appropriate root of equation (7), then we require $\tau_C^h T_0(C) \geq \tau_W T_0(W)$, $\tau_C^h T_0(C) \geq \tau_S T_0(S)$, e.g., Figure 1.

Equation (7) may not have real roots or may have real roots that don't satisfy the causality conditions. Under such circumstance, it means that the values of τ_W and τ_S do not support a valid solution τ_C within the triangle $\triangle CWS$ using linear approximation. This may happen especially when the initial guess is arbitrary. However, notice that equation (5) is a hyperbolic equation. We can use the classical method of characteristics to pass the information of τ from W to C and from S to C along edges \overrightarrow{WC} , \overrightarrow{SC} respectively.

The characteristic equation for (5), with $(\tau_x, \tau_y) = (p, q)$, is:

$$\begin{cases} \frac{dx}{dt} = (2T_0^2 p + 2\tau T_0 T_{0x}, 2T_0^2 q + 2\tau T_0 T_{0y}) \\ \frac{d\tau}{dt} = 2T_0^2(p^2 + q^2) + 2\tau T_0(T_{0x}p + T_{0y}q) \end{cases} \quad (8)$$

By using the method of characteristics on edges $\overrightarrow{WC}, \overrightarrow{SC}$ respectively, one can calculate two values of τ : τ_{WC} and τ_{SC} . For example, on edge $\overrightarrow{WC} = (\delta x, \delta y)$, $|\overrightarrow{WC}| = L$ (in this case, $\delta x = h, \delta y = 0, L = h$), using first equation in (8), we have

$$\begin{cases} \delta x = (2T_0^2 p + 2\tau T_0 T_{0x}) \delta t \\ \delta y = (2T_0^2 q + 2\tau T_0 T_{0y}) \delta t \end{cases} \Rightarrow \begin{cases} p = \left(\frac{\delta x}{\delta t} - 2\tau T_0 T_{0x} \right) \frac{1}{2T_0^2} \\ q = \left(\frac{\delta y}{\delta t} - 2\tau T_0 T_{0y} \right) \frac{1}{2T_0^2} \end{cases}$$

Plug (p, q) into (5), we get an equation of δt , solve for δt to get $\delta t = \frac{L}{2T_0 S(C)}$. Now use second equation in (8), we have $\tau = \tau_W + [2T_0^2(p^2 + q^2) + 2\tau T_0(T_{0x}p + T_{0y}q)]\delta t$, solve for τ to get $\tau_{WC} = \frac{LS(C) + \tau_W T_0}{T_0 + (\delta x T_{0x} + \delta y T_{0y})}$.

Now, we present the local solver.

Local Solver:

1. Initialization: assign initial boundary values on boundary grid points.
2. Gauss-Seidel iteration: sweeping the domain with four alternating orderings repeatedly:

$$\begin{aligned} (1) \quad & i = 1 : I, j = 1 : J & (2) \quad & i = 1 : I, j = J : 1 \\ (3) \quad & i = I : 1, j = 1 : J & (4) \quad & i = I : 1, j = J : 1 \end{aligned}$$

- At each grid point, discretize the factored eikonal equation on 4 triangles $\triangle CEN, \triangle CNW, \triangle CWS$ and $\triangle CSE$, and solve the discretized equations on each triangle. For example, on triangle $\triangle CWS$, solve equation (7) for two possible roots, $\tau_{C,1}$ and $\tau_{C,2}$.

- If there are two real roots, $\tau_{C,1}$ and $\tau_{C,2}$, then
 - if both $\tau_{C,1}, \tau_{C,2}$ satisfy causality condition, then

$$T_{WS} = \min(\tau_{C,1}T_0(C), \tau_{C,2}T_0(C)) .$$

- else if $\tau_{C,1}$ satisfies the causality condition, then $T_{WS} = \tau_{C,1}T_0(C)$.
- else if $\tau_{C,2}$ satisfies the causality condition, then $T_{WS} = \tau_{C,2}T_0(C)$.
- else if none of the two roots satisfies the causality condition, then use method of characteristics on edges $\overrightarrow{WC}, \overrightarrow{SC}$ to get τ_{WC} and τ_{SC} . Enforce causality $\tau_{WC}T_0(C) \geq T_W, \tau_{SC}T_0(C) \geq T_S$. And choose $T_{WS} = \min(\tau_{WC}T_0(C), \tau_{SC}T_0(C))$.

- else, use method of characteristics on edges $\overrightarrow{WC}, \overrightarrow{SC}$ to get τ_{WC} and τ_{SC} . Enforce causality $\tau_{WC}T_0(C) \geq T_W, \tau_{SC}T_0(C) \geq T_S$. And choose $T_{WS} = \min(\tau_{WC}T_0(C), \tau_{SC}T_0(C))$.

- Choose the minimum from 4 triangles, $T_C = \min(T_{EN}, T_{NW}, T_{WS}, T_{SE})$.

- $\tau_C = \frac{T_C}{T_0(C)}$.

Remark: Analytically, the original eikonal equation T and the factored eikonal equation for τ , including method of characteristics are all equivalent. However, the later approach will get better approximation for ∇T , because $\nabla T = \nabla\tau T_0 + \tau\nabla T_0$ with the decomposition $T = \tau T_0$. The singularity is captured by ∇T_0 which is precisely known. And τ is smooth. Hence we avoid taking ∇T directly. By solving a differential equation for τ we achieve better approximation. Therefore, τT_0 is also more accurate. Another important point is that the causality condition for τ is equivalent to the one for T and we know the enforcement of causality condition during the fast sweeping iteration guarantees the convergence of T for the original eikonal equation [26].

4 Numerical Tests

In the section we test our numerical algorithm on a variety of examples to demonstrate both accuracy and efficiency.

Equation (1) has analytical solutions for the case of point-source and some special forms of the slowness function $S(\mathbf{x})$. In examples 1-7, we use analytical examples to test the accuracy and consistency of the proposed numerical scheme. In examples 2 and 4, we test cases with two source points. For completion analytical derivations of these examples are included in the Appendix.

Example 1: Constant gradient of slowness squared. As shown in Appendix A, when a point source is located at point \mathbf{x}_0 and the slowness distribution has the form

$$S^2(\mathbf{x}) = S_0^2 + 2\mathbf{g}_0 \cdot (\mathbf{x} - \mathbf{x}_0) \quad (9)$$

with constant S_0 and \mathbf{g}_0 , then the analytical solution is

$$T(\mathbf{x}) = \bar{S}^2 \sigma - |\mathbf{g}_0|^2 \frac{\sigma^3}{6}, \quad (10)$$

where

$$\sigma = \frac{\sqrt{2 \left(\bar{S}^2 \pm \sqrt{\bar{S}^4 - |\mathbf{g}_0|^2 |\mathbf{x} - \mathbf{x}_0|^2} \right)}}{|\mathbf{g}_0|} \quad (11)$$

and

$$\bar{S}(\mathbf{x}) = \sqrt{\frac{S^2(\mathbf{x}) + S_0^2}{2}} = \sqrt{S_0^2 + \mathbf{g}_0 \cdot (\mathbf{x} - \mathbf{x}_0)}. \quad (12)$$

In general, two different solutions exist for two different signs in equation (11). The branch of the solution that turns into the constant-slowness solution for \mathbf{g}_0 approaching $\mathbf{0}$ corresponds to the negative sign and can be written in the alternative form

$$\sigma^2 = \frac{2 \left(\bar{S}^2 - \sqrt{\bar{S}^4 - |\mathbf{g}_0|^2 |\mathbf{x} - \mathbf{x}_0|^2} \right)}{|\mathbf{g}_0|^2} = \frac{2 |\mathbf{x} - \mathbf{x}_0|^2}{\bar{S}^2 + \sqrt{\bar{S}^4 - |\mathbf{g}_0|^2 |\mathbf{x} - \mathbf{x}_0|^2}}, \quad (13)$$

which avoids division by zero.

We test one setup with parameters:

- $\mathbf{x}_0 = (0, 0)$
- computational domain: $[0, 1.5] \times [0, 0.5]$
- $S_0 = 2 \text{ s/km}$
- $\mathbf{g}_0 = \{0, -3\} \text{ s}^2/\text{km}^3$

Table 1 shows the results by the fast sweeping method applied to both factored eikonal equation and the original eikonal equation. Figure 2 shows the plots of numerical solutions on a 150×50 mesh. The result shows that the factored eikonal equation captures the right causality of the solution. So it is as efficient as the original fast sweeping method. At the same time the numerical solution is much more accurate in terms of both error magnitude and order of convergence thanks to the fact that the singularity at the point source is well captured by the distance function. For our first order upwind scheme a perfect first order of convergence is achieved. On the other hand a first order discretization of the original eikonal equation can not achieve first order of accuracy due to the singularity at the point source.

Factored eikonal equation		
Mesh	T on $[0, 0.5] \times [0, 0.5]$	# iterations
150x50	0.0010702	3
300x100	0.0005348	3
600x200	0.0002673	3
1200x400	0.0001336	3
Original eikonal equation		
Mesh	T on $[0, 0.5] \times [0, 0.5]$	# iterations
150x50	0.0214127	3
300x100	0.0129566	3
600x200	0.0076381	3
1200x400	0.0044107	3

Table 1: Constant gradient slowness squared: Constant slowness S_0

Example 2: Constant gradient of slowness squared continued. In this example, we test one case with two source points $(0, 0), (1.5, 0)$. Two choices of T_0 are tested:

- (1) $T_0 = \min(\text{dist}(x, y, 0, 0), \text{dist}(x, y, 1.5, 0))$, in which case $|\nabla T_0| = 1$ and T_0 has shocks besides two point source singularities.
- (2) $T_0 = \text{dist}(x, y, 0, 0) \times \text{dist}(x, y, 1.5, 0)$, in which case $|\nabla T_0|$ can be computed exactly and T_0 is smooth away from two point sources.

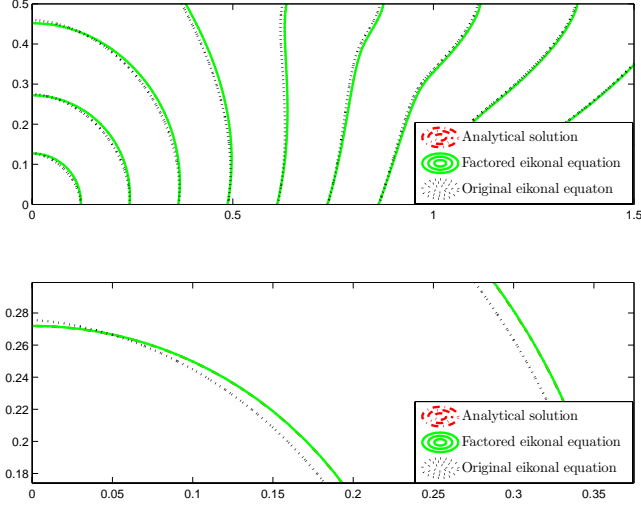


Figure 2: TOP: plots of analytical solution, original eikonal equation and factored eikonal equation. BOTTOM: Zoom in of plots.

Table 2 shows the results. First we can see again that factored eikonal equation has much better results than original eikonal equation as in Example 1, since the source singularities are well-captured by T_0 . Second we see that factored eikonal equation with $T_0 = \min(\text{dist}(x, y, 0, 0), \text{dist}(x, y, 1.5, 0))$ has better accuracy than factored eikonal equation with $T_0 = \text{dist}(x, y, 0, 0) \times \text{dist}(x, y, 1.5, 0)$. We suspect that it is because the locus of shock for T happens to be the locus of shock for $T_0 = \min(\text{dist}(x, y, 0, 0), \text{dist}(x, y, 1.5, 0))$.

Example 3: Constant gradient of velocity. When a point source is located at point \mathbf{x}_0 and the slowness distribution has the form

$$\frac{1}{S(\mathbf{x})} = \frac{1}{S_0} + \mathbf{G}_0 \cdot (\mathbf{x} - \mathbf{x}_0) \quad (14)$$

with constant S_0 and \mathbf{G}_0 (constant gradient of velocity), the analytical solution is

$$T(\mathbf{x}) = \frac{1}{|\mathbf{G}_0|} \text{arccosh} \left(1 + \frac{1}{2} S(\mathbf{x}) S_0 |\mathbf{G}_0|^2 |\mathbf{x} - \mathbf{x}_0|^2 \right), \quad (15)$$

where arccosh is the inverse hyperbolic cosine function

$$\text{arccosh}(z) = \ln \left(z + \sqrt{z^2 - 1} \right).$$

We test one setup with parameters:

- $\mathbf{x}_0 = (0, 0)$
- computational domain: $[0, 1] \times [0, 0.5]$

Factored eikonal equation: $T_0 = \min(\text{dist}(x, y, 0, 0), \text{dist}(x, y, 1.5, 0))$		
Mesh	T on $[0, 1.5] \times [0, 0.25]$	# iterations
150x50	0.0004933	5
300x100	0.0002456	5
600x200	0.0001225	5
1200x400	0.0000611	5
Factored eikonal equation: $T_0 = \text{dist}(x, y, 0, 0) \times \text{dist}(x, y, 1.5, 0)$		
Mesh	T on $[0, 1.5] \times [0, 0.25]$	# iterations
150x50	0.0050798	5
300x100	0.0025370	5
600x200	0.0012679	5
1200x400	0.0006338	5
Original eikonal equation		
Mesh	T on $[0, 1.5] \times [0, 0.25]$	# iterations
150x50	0.0205986	5
300x100	0.0124967	5
600x200	0.0073907	5
1200x400	0.0042815	5

Table 2: Constant gradient slowness squared: Constant slowness.

- $S_0 = 2 \text{ s/km}$
- $\mathbf{G}_0 = \{0, -1\} 1/\text{s}$

Table 3 shows the results by both factored eikonal equation and fast sweeping method on original eikonal equation. The result shows the same observations as in the previous example. Figure 3 shows the plots of numerical solutions on a 160×80 mesh.

Example 4: Constant gradient of velocity continued. In this example, we test one case with two source points $(0, 0), (1, 0)$. Two choices of T_0 are tested: $T_0 = \min(\text{dist}(x, y, 0, 0), \text{dist}(x, y, 1, 0))$ or $T_0 = \text{dist}(x, y, 0, 0) \times \text{dist}(x, y, 1, 0)$. Table 4 shows the results. We can see the similar results as in Example 2.

Example 5: In this example, we take the analytical solution from example 3 as T_0 in example 1. Table 5 shows the results by factored eikonal equation. Figure 4 shows the plots of numerical solutions on a 150×50 mesh.

Example 6: In this example, we take the analytical solution from Example 1 as T_0 in Example 3.

Table 6 shows the results by factored eikonal equation. Figure 5 shows the plots of numerical solutions on a 100×100 mesh.

Remark. These two examples demonstrate that the factored eikonal equation can improve the numerical results if some approximate solution is known. Again, in these two cases the singularity of the solution is similar to that of example 1 near point source.

Example 7: In this example, we test a model with a plane-wave source,

Factored eikonal equation		
Mesh	T on $[0, 0.5] \times [0, 0.5]$	# iterations
160x80	0.0007115	3
320x160	0.0003555	3
640x320	0.0001777	3
1280x640	0.0000888	3
Original eikonal equation		
Mesh	T on $[0, 0.5] \times [0, 0.5]$	# iterations
160x80	0.0140801	3
320x160	0.0084309	3
640x320	0.0049316	3
1280x640	0.0028312	3

Table 3: Constant gradient of velocity: constant slowness S_0

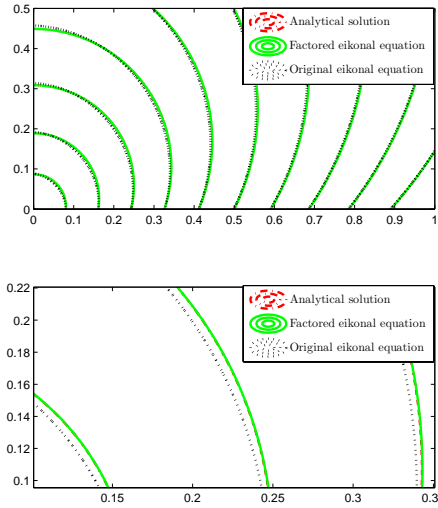


Figure 3: TOP: plots of analytical solution, original eikonal equation and factored eikonal equation. BOTTOM: Zoom in of plots.

Factored eikonal equation: $T_0 = \min(\text{dist}(x, y, 0, 0), \text{dist}(x, y, 1, 0))$		
Mesh	T	# iterations
160x80	0.0007115	5
320x160	0.0003555	5
640x320	0.0001777	5
1280x640	0.0000888	5
Factored eikonal equation: $T_0 = \text{dist}(x, y, 0, 0) \times \text{dist}(x, y, 1, 0)$		
Mesh	T	# iterations
160x80	0.0031731	5
320x160	0.0015822	5
640x320	0.0007902	5
1280x640	0.0003949	5
Original eikonal equation		
Mesh	T	# iterations
160x80	0.0140801	5
320x160	0.0084309	5
640x320	0.0049316	5
1280x640	0.0028312	5

Table 4: Constant gradient of velocity

Factored eikonal equation		
Mesh	T on $[0, 0.5] \times [0, 0.5]$	# iterations
150x50	0.0021033	3
300x100	0.0010510	3
600x200	0.0005253	3
1200x400	0.0002626	3

Table 5: Constant gradient slowness squared: analytical T_0

Factored eikonal equation		
Mesh	T on $[0, 0.5] \times [0, 0.5]$	# iterations
80x80	0.0012770	3
160x160	0.0006377	3
320x320	0.0003186	3
640x640	0.0001592	3

Table 6: Constant gradient of velocity: analytical T_0

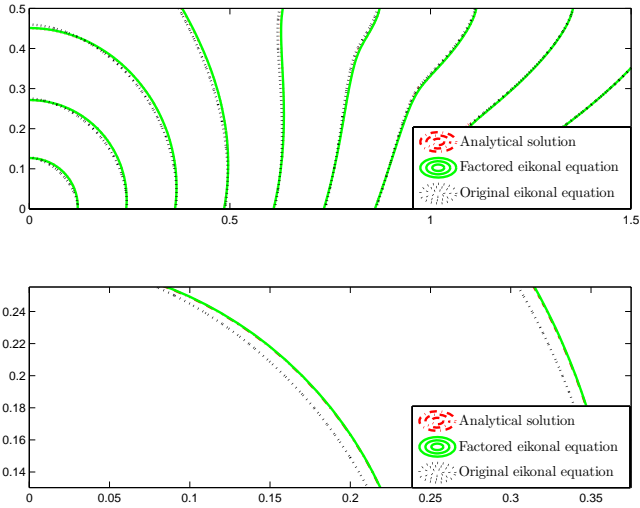


Figure 4: TOP: plots of analytical solution, original eikonal equation and factored eikonal equation. BOTTOM: Zoom in of plots.

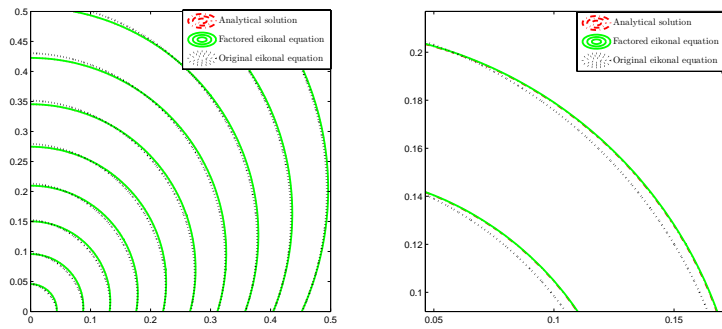


Figure 5: LEFT: plots of analytical solution, original eikonal equation and factored eikonal equation. RIGHT: Zoom in of plots.

the analytical solution is:

$$T(\mathbf{x}) = \frac{\sqrt{2}}{3} \frac{(3S_1^4 + g_1^2 z^2)z}{\sqrt{S_1^2(S_1^4 + 3g_1^2 z^2) + (S_1^4 - g_1^2 z^2)^{3/2}}} \quad (16)$$

where $\mathbf{x} = (x, z)$, $g_0 = (g_x, g_z)$, $g_1 = \sqrt{4g_x^4 + g_z^2}$, and $S_1 = \sqrt{S^2(x, z) - g_z z} = \sqrt{S_0^2 + 2g_x(x - x_0) + g_z z}$.

We test the model with parameters:

- plane-wave source on $z = 0$
- computational domain: $[0, 1.5] \times [0, 0.5]$
- $S_0 = 2\text{s/km}$
- $g_0 = (1, -3)\text{s}^2/\text{km}^3$

First we choose $T_0(x, z) = S_0 z$. Table 7 shows the results both by factored eikonal equation and fast sweeping method on original eikonal equation. Figure 6 shows the plots of numerical solutions on a 150×50 mesh. Since there is no point source singularity in the problem, we see both methods achieve first order accuracy. However, the factored eikonal equation still produces much better accuracy.

Factored eikonal equation		
Mesh	T on $[0.75, 1.5] \times [0, 0.5]$	# iterations
150x50	0.0003085	2
300x100	0.0001540	2
600x200	0.0000770	2
1200x400	0.0000385	2
Original eikonal equation		
Mesh	T on $[0.75, 1.5] \times [0, 0.5]$	# iterations
150x50	0.0039623	2
300x100	0.0019791	2
600x200	0.0009891	2
1200x400	0.0004944	2

Table 7: Plane-wave source: $T_0(x, z) = S_0 z$

Next we choose $T_0(x, z) = \int_0^z S(z) dz$ with $S(z) = \int_{x_{\min}}^{x_{\max}} S(x, z) dx$, using factored Eikonal equation.

Table 8 shows the results of factored eikonal equation. Figure 7 shows the plots of numerical solutions on a 150×50 mesh.

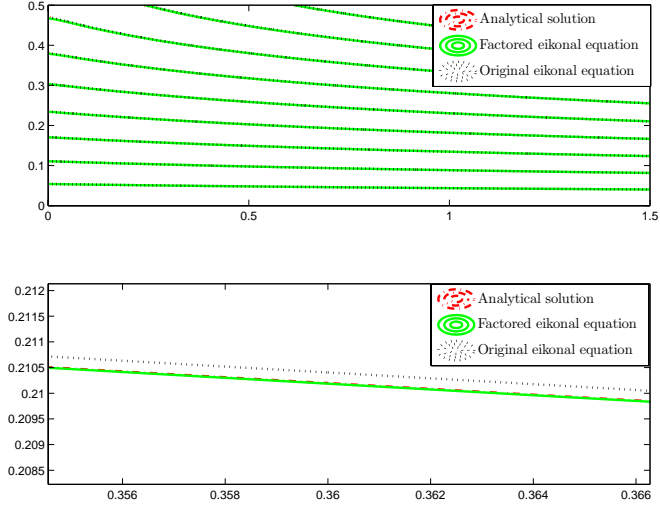


Figure 6: TOP: plots of analytical solution, original eikonal equation and factored eikonal equation. BOTTOM: Zoom in of plots.

Factored eikonal equation		
Mesh	T on $[0.75, 1.5] \times [0, 0.5]$	# iterations
150x50	0.0002538	2
300x100	0.0001267	2
600x200	0.0000633	2
1200x400	0.0000316	2
Original eikonal equation		
Mesh	T on $[0.75, 1.5] \times [0, 0.5]$	# iterations
150x50	0.0039623	2
300x100	0.0019791	2
600x200	0.0009891	2
1200x400	0.0004944	2

Table 8: plane-wave source: integral T_0

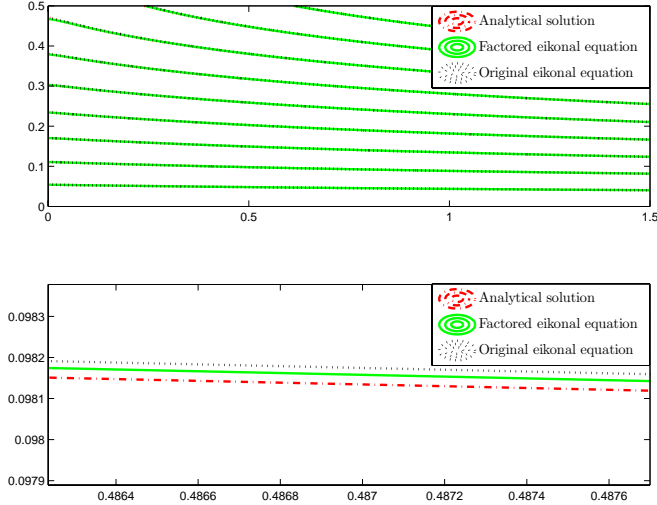


Figure 7: TOP: plots of analytical solution, original eikonal equation and factored eikonal equation. BOTTOM: Zoom in of plots.

4.1 Marmousi Model

In this example, we apply our method to the famous Marmousi model [21] using a plane-wave source. We choose $T_0(x, z) = 2z$ for factored eikonal equation.

Figure 8 shows the results by original eikonal equation and factored eikonal equation on a 2301×751 mesh (grid size 4m).

The results were computed with C code programmed by Microsoft Visual C++ 6.0 on PC(Intel(R)Core(TM)2 CPU T5500@1.66GHZ). With $tolerance = 10^{-9}$, fast sweeping method on original equation takes 28 iterations, CPU time: 62 seconds. Factored eikonal equation takes 18 iterations, CPU time: 93 seconds.

5 Conclusions

We have demonstrated that the fast sweeping method based on the factored eikonal equation improves the accuracy of the numerical solution to the original eikonal equation while maintaining the efficiency of the classical fast sweeping method. In the case of a point source problem, the singularity at the source can be removed effectively by the factored eikonal equation. Both for individual point sources and simultaneous point sources, as well as plane-wave sources, we observe a significant improvement in accuracy.

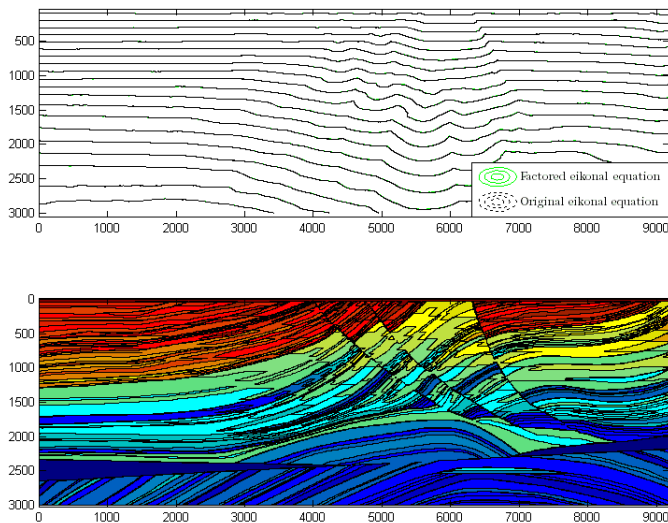


Figure 8: TOP: plots of Marmousi model by original eikonal equation and factored eikonal equation. BOTTOM: slowness field $S(x, z)$.

References

- [1] F. Audebert, J.P. Diet, P. Guillaume, I. F. Jones, and X. Zhang. CRP-scans: 3-D preSDM velocity analysis via zero-offset tomographic inversion. In *67th Ann. Internat. Mtg*, pages 1805–1808. Soc. of Expl. Geophys., 1997.
- [2] F. Audebert, J.P. Diet, and X. Zhang. CRP-scans from 3-D pre-stack depth migration: A powerful combination of CRP-gathers and velocity scans. In *66th Ann. Internat. Mtg*, pages 515–518. Soc. of Expl. Geophys., 1996.
- [3] D. Bevc. Imaging complex structures with semirecursive Kirchhoff migration. *Geophysics*, 62(02):577–588, 1997.
- [4] M. Boué and P. Dupuis. Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. *SIAM J. Numer. Anal.*, 36(3):667–695, 1999.
- [5] V. Červený. *Seismic ray theory*. Cambridge Univ. Press, 2001.
- [6] R. Courant and D. Hilbert. *Methods of mathematical physics*. John Wiley & Sons, 1989.
- [7] S. Geoltrain and J. Brac. Can we image complex structures with first-arrival traveltimes? *Geophysics*, 58(04):564–575, 1993.

- [8] C.Y. Kao, S. Osher, and J. Qian. Lax-friedrichs sweeping schemes for static hamilton-jacobi equations. *Journal of Computational Physics*, 196:367–391, 2004.
- [9] C.Y. Kao, S. Osher, and Y.H. Tsai. Fast sweeping method for static hamilton-jacobi equations. *SIAM Journal on Numerical Analysis*, 42:2612–2632, 2005.
- [10] S. Kim. 3-D eikonal solvers: First arrival traveltimes. *Geophysics*, 67(4):1225–1231, 2002.
- [11] M. Körnig. Cell ray tracing for smooth, isotropic media: a new concept based on a generalized analytical solution. *Geophys. J. Int.*, 123:391–408, 1995.
- [12] A. Pica. Fast and accurate finite-difference solutions of the 3d eikonal equation parametrized in celerity. In *67th Ann. Internat. Mtg*, pages 1774–1777. Soc. of Expl. Geophys., 1997.
- [13] J. Qian and W.W. Symes. An adaptive finite-difference method for travel-times and amplitudes. *Geophysics*, 67:167–176, 2002.
- [14] J. Qian, Y.-T. Zhang, and H.-K. Zhao. Fast sweeping methods for eikonal equations on triangulated meshes. *SIAM J. Numer. Anal.*, 45:83–107, 2007.
- [15] J. Qian, Y.-T. Zhang, and H.-K. Zhao. A fast sweeping methods for static convex hamilton-jacobi equations. *Journal of Scientific Computings*, 31(1/2):237–271, 2007.
- [16] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.
- [17] J.A. Sethian and A.M. Popovici. 3-D travelttime computation using the fast marching method. *Geophysics*, 64(2):516–523, 1999.
- [18] M. M. Slotnick. *Lessons in Seismic Computing*. Soc. of Expl. Geophys., 1959. Edited by R. A. Geyer.
- [19] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM Journal on Numerical Analysis*, 41:673–694, 2003.
- [20] J. van Trier and W.W. Symes. Upwind finite-difference calculation of traveltimes. *Geophysics*, 56(06):812–821, 1991.
- [21] R. Versteeg. The Marmousi experience: Velocity model determination on a synthetic complex data set. *The Leading Edge*, 13(09):927–936, 1994.
- [22] J. E. Vidale. Finite-difference calculation of traveltimes in three dimensions. *Geophysics*, 55(05):521–526, 1990.

- [23] L. Zhang, J.W. Rector, and G.M. Hoversten. Eikonal solver in the celerity domain. *Geophysical Journal International*, 162:1–8, 2005.
- [24] Y.-T. Zhang, H.-K. Zhao, and S. Chen. Fixed-point iterative sweeping methods for static hamilton-jacobi equations. *Methods and Applications of Analysis*, 13:299–320, 2006.
- [25] Y.-T. Zhang, H.-K. Zhao, and J. Qian. High order fast sweeping methods for static hamilton-jacobi equations. *Journal of Scientific Computing*, 29:25–56, 2006.
- [26] H.-K. Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74:603–627, 2005.
- [27] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.

A Analytical traveltimes in the linear sloth model

In this appendix, we provide analytical solutions of the eikonal equation (1) for the special case of a constant gradient of slowness squared

$$S^2(\mathbf{x}) = S_0^2 + 2 \mathbf{g}_0 \cdot (\mathbf{x} - \mathbf{x}_0) \quad (17)$$

and a point-source or plane-wave boundary conditions. More general solutions of this kind are provided in [11].

We use the classic Hamilton-Jacobi theory and the method of characteristics [6]. The characteristics of equation (1) are described by the following system of ODEs

$$\frac{d\mathbf{x}}{d\sigma} = \mathbf{p} , \quad (18)$$

$$\frac{d\mathbf{p}}{d\sigma} = S(\mathbf{x}) \nabla S , \quad (19)$$

$$\frac{dT}{d\sigma} = S^2(\mathbf{x}) . \quad (20)$$

Solving the initial-value problem $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{p}(0) = \mathbf{p}_0$ for equation (19), we obtain

$$\mathbf{p}(\sigma) = \mathbf{p}_0 + \mathbf{g}_0 \sigma . \quad (21)$$

Solving equation (18) leads to

$$\mathbf{x}(\sigma) = \mathbf{x}_0 + \mathbf{p}_0 \sigma + \mathbf{g}_0 \frac{\sigma^2}{2} . \quad (22)$$

Rays, described by equation (22) are parabolic trajectories, analogous to trajectories of mechanical particles traveling with a constant acceleration. In the

special case of a zero gradient, the trajectories are straight lines. Equation (20) transforms to

$$\frac{dT}{d\sigma} = \mathbf{p}(\sigma) \cdot \mathbf{p}(\sigma) = \mathbf{p}_0 \cdot \mathbf{p}_0 + 2 \mathbf{p}_0 \cdot \mathbf{g}_0 \sigma + \mathbf{g}_0 \cdot \mathbf{g}_0 \sigma^2 . \quad (23)$$

Its solution takes the form

$$T(\sigma) = \mathbf{p}_0 \cdot \mathbf{p}_0 \sigma + \mathbf{p}_0 \cdot \mathbf{g}_0 \sigma^2 + \mathbf{g}_0 \cdot \mathbf{g}_0 \frac{\sigma^3}{3} . \quad (24)$$

A.1 Point source

From equation (22), we derive

$$S^2(\mathbf{x}_1) - S^2(\mathbf{x}_0) = 2 \mathbf{g}_0 \cdot (\mathbf{x}_1 - \mathbf{x}_0) = 2 \mathbf{g}_0 \cdot \mathbf{p}_0 \sigma + \mathbf{g}_0 \cdot \mathbf{g}_0 \sigma^2 \quad (25)$$

or

$$\mathbf{g}_0 \cdot \mathbf{p}_0 = \frac{S^2(\mathbf{x}_1) - S^2(\mathbf{x}_0) - |\mathbf{g}_0|^2 \sigma^2}{2 \sigma} . \quad (26)$$

Additionally, equation (22) leads to

$$|\mathbf{x}_1 - \mathbf{x}_0|^2 = S^2(\mathbf{x}_0) \sigma^2 + \mathbf{g}_0 \cdot \mathbf{p}_0 \sigma^3 + |\mathbf{g}_0|^2 \frac{\sigma^4}{4} \quad (27)$$

or, substituting equation (26),

$$\begin{aligned} |\mathbf{x}_1 - \mathbf{x}_0|^2 &= S^2(\mathbf{x}_0) \sigma^2 + \frac{S^2(\mathbf{x}_1) - S^2(\mathbf{x}_0) - |\mathbf{g}_0|^2 \sigma^2}{2} \sigma^2 + |\mathbf{g}_0|^2 \frac{\sigma^4}{4} \\ &= \frac{S^2(\mathbf{x}_1) + S^2(\mathbf{x}_0)}{2} \sigma^2 - |\mathbf{g}_0|^2 \frac{\sigma^4}{4} . \end{aligned} \quad (28)$$

The last equation is a quadratic equation for σ^2 and can be solved analytically, as follows:

$$\sigma = \frac{\sqrt{2 \left(\bar{S}^2 \pm \sqrt{\bar{S}^4 - |\mathbf{g}_0|^2 |\mathbf{x}_1 - \mathbf{x}_0|^2} \right)}}{|\mathbf{g}_0|} , \quad (29)$$

where

$$\bar{S}(\mathbf{x}_0, \mathbf{x}_1) = \sqrt{\frac{S^2(\mathbf{x}_1) + S^2(\mathbf{x}_0)}{2}} = \sqrt{S^2(\mathbf{x}_0) + \mathbf{g}_0 \cdot (\mathbf{x}_1 - \mathbf{x}_0)} . \quad (30)$$

Finally, we can substitute equations (26) and (29) into (24) to find the analytical two-point traveltimes for the constant-slowness-gradient case

$$\hat{T}(\mathbf{x}_0, \mathbf{x}_1) = \bar{S}^2 \sigma - |\mathbf{g}_0|^2 \frac{\sigma^3}{6} , \quad (31)$$

where σ can be one of the two solutions in equation (29). Equation (31) coincides with (10) in the main text.

A.2 Plane-wave source

In the case of a plane-wave source $\mathbf{p}_0 = \{0, p_z\}$ at the surface $\mathbf{x}_0 = \{x_0, 0\}$, equations (25), (22), and (24) transform to

$$S^2 = S_0^2 + 2g_z p_z \sigma + (g_x^2 + g_z^2) \sigma^2, \quad (32)$$

$$z = z_0 + p_z \sigma + g_z \frac{\sigma^2}{2}, \quad (33)$$

$$T = S_0^2 \sigma + p_z g_z \sigma^2 + (g_x^2 + g_z^2) \frac{\sigma^3}{3}, \quad (34)$$

where g_x and g_z are components of \mathbf{g}_0 . Solving equations (32-33) for p_z and σ and substituting the solution in equation (34) produces equation (16) in the main text.

B Analytical traveltimes in the linear velocity model

To derive analytical traveltimes for the case of the slowness distribution of the form

$$\frac{1}{S(\mathbf{x})} = \frac{1}{S_0} + \mathbf{G}_0 \cdot (\mathbf{x} - \mathbf{x}_0), \quad (35)$$

it is convenient to change the variable in the ODE system (18-20) to put it in the form

$$\frac{d\mathbf{x}}{d\xi} = \frac{\mathbf{p}}{S^3(\mathbf{x})}, \quad (36)$$

$$\frac{d\mathbf{p}}{d\xi} = \frac{\nabla S}{S^2(\mathbf{x})} = -\nabla \left(\frac{1}{S(\mathbf{x})} \right), \quad (37)$$

$$\frac{dT}{d\xi} = \frac{1}{S(\mathbf{x})}, \quad (38)$$

where $d\xi = S^3(\mathbf{x}) d\sigma$. The solution of equation (37) for the initial condition $\mathbf{p}(0) = \mathbf{p}_0$ is

$$\mathbf{p}(\xi) = \mathbf{p}_0 - \mathbf{G}_0 \xi. \quad (39)$$

Therefore,

$$S^2(\mathbf{x}) = \mathbf{p} \cdot \mathbf{p} = \mathbf{p}_0 \cdot \mathbf{p}_0 - 2\mathbf{p}_0 \cdot \mathbf{G}_0 \xi + \mathbf{G}_0 \cdot \mathbf{G}_0 \xi^2. \quad (40)$$

Substituting (40) in equation (38), we can solve this ODE analytically to obtain

$$T(\xi) = \frac{1}{|\mathbf{g}_0|} \operatorname{arccosh} \left(1 + \frac{|\mathbf{g}_0|^2 \xi^2}{S_0^2 + S_0 S(\mathbf{x}) - \mathbf{p}_0 \cdot \mathbf{G}_0 \xi} \right). \quad (41)$$

Let $a = \mathbf{p} \cdot (\mathbf{x} - \mathbf{x}_0)$. The chain rule shows that

$$\begin{aligned} \frac{da}{d\xi} &= \frac{d\mathbf{p}}{d\xi} \cdot (\mathbf{x} - \mathbf{x}_0) + \mathbf{p} \cdot \frac{d\mathbf{x}}{d\xi} = -\mathbf{G}_0 \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{\mathbf{p} \cdot \mathbf{p}}{S^3(\mathbf{x})} \\ &= -\mathbf{G}_0 \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{S(\mathbf{x})} = \frac{1}{S_0}. \end{aligned} \quad (42)$$

Therefore $a = \xi/S_0$.

Let $r = (\mathbf{x} - \mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)$. The chain rule shows that

$$\frac{dr}{d\xi} = 2 \frac{d\mathbf{x}}{d\xi} \cdot (\mathbf{x} - \mathbf{x}_0) = 2 \frac{\mathbf{p} \cdot (\mathbf{x} - \mathbf{x}_0)}{S^3(\mathbf{x})} = \frac{2\xi}{S^3(\mathbf{x}) S_0}. \quad (43)$$

Substituting equation (40) into (43) and integrating, we obtain

$$r(\xi) = \frac{4\xi^2}{S(\mathbf{x}) S_0 \left([S(\mathbf{x}) + S_0]^2 - \mathbf{G}_0 \cdot \mathbf{G}_0 \xi^2 \right)}. \quad (44)$$

Finally, expressing ξ from equation (44) and substituting into equation (41), we obtain

$$T(\mathbf{x}) = \frac{1}{|\mathbf{G}_0|} \operatorname{arccosh} \left(1 + \frac{1}{2} S(\mathbf{x}) S_0 |\mathbf{G}_0|^2 |\mathbf{x} - \mathbf{x}_0|^2 \right), \quad (45)$$

which is equivalent to equation (15) in the main text. Equation (45) is well-known [18], its derivation is included here for completeness.