Electronic Journal of Differential Equations, Vol. 2013 (2013), No. 40, pp. 1–20.
ISSN: 1072-6691. URL: http://ejde.math.txstate.edu or http://ejde.math.unt.edu
ftp ejde.math.txstate.edu

A STEP-LIKE APPROXIMATION AND A NEW NUMERICAL SCHEMA FOR THE KORTEWEG-DE VRIES EQUATION IN THE SOLITON REGION

JASON BAGGETT, ODILE BASTILLE, ALEXEI RYBKIN

Abstract. We discuss a numerical schema for solving the initial value problem for the Korteweg-de Vries equation in the soliton region which is based on a new method of evaluation of bound state data. Using a step-like approximation of the initial profile and a fragmentation principle for the scattering data, we obtain an explicit procedure for computing the bound state data. Our method demonstrates an improved accuracy on discontinuous initial data. We also discuss some generalizations of this algorithm and how it might be improved by using Haar and other wavelets.

1. Introduction

In this article, we consider the initial value problem for the Korteweg/de Vries (KdV) equation

$$u_t - 6uu_x + u_{xxx} = 0, \quad u(x,0) = V(x) \tag{1.1}$$

on the whole line. The function $V(x)$, called initial profile, is assumed to be bounded (but not necessarily continuous), compactly supported (i.e. zero outside of a finite interval), and $V(x) \leq 0$ (assumed for simplicity only). In particular, we are concerned with numerical algorithms for (1.1) for large times t.

The KdV equation is "exactly solvable" by relating it to the Schrödinger equation

$$-\phi_{xx} + V(x)\phi = \lambda\phi \tag{1.2}$$

through the so-called Inverse Scattering Transform (IST) (see, e.g. [1]). In a sense, the IST linearizes the KdV (as well as some other nonlinear evolution PDEs) and provides us with an extremely powerful tool to analyze its solutions. The main feature of the IST is that it solves (1.1) purely in terms of the so-called scattering data associated with (1.2) (see Section 4 for more detail).

The scattering data (More precisely, the right scattering data) of the Schrödinger equation consists of finitely many bound states $\{-\kappa_j^2\}_{j=1}^J$, the corresponding (left) norming constants $\{c_j\}_{j=1}^J$, and the (right) reflection coefficient R. The bound

2000 *Mathematics Subject Classification.* 35P25, 35Q53, 37K15, 37K10, 37K40, 42C40, 65N25.
Key words and phrases. KdV equation; Haar wavelets; potential fragmentation; inverse scattering transform.

Submitted September 27, 2011. Published February 4, 2013.
Supported in part by the NSF under grants DMS 0707476 and DMS 1009673.

states are precisely the eigenvalues λ of the Schrödinger equation that give square-integrable solutions ϕ. The right and left reflection coefficients R and L, respectively, and the transmission coefficient T come from the asymptotic behavior of the left and right scattering solutions to the Schrödinger equation ϕ_r and ϕ_1, respectively, where for $\lambda = k^2$

$$\phi_r(x,k) = \begin{cases} e^{-ikx} + R(k)e^{ikx} + o(1) & x \to \infty \\ T(k)e^{-ikx} + o(1) & x \to -\infty, \end{cases} \tag{1.3}$$

and

$$\phi_1(x,k) = \begin{cases} e^{ikx} + L(k)e^{-ikx} + o(1) & x \to -\infty \\ T(k)e^{ikx} + o(1) & x \to \infty. \end{cases} \tag{1.4}$$

If $\lambda = -\kappa^2$ is a bound state, then $\phi_1(x, i\kappa)$ is square-integrable. The corresponding (left) norming constant is defined by

$$c = \Big(\int_{-\infty}^{\infty} |\phi_1(x,i\kappa)T^{-1}(i\kappa)|^2 dx \Big)^{-1/2}.$$

The IST procedure for (1.1) consists of two main steps: computing the scattering data $\{-\kappa_j^2, c_j, R(k)\}$ for V and then constructing $u(x,t)$ using the scattering data. Unfortunately, neither step is explicit in general and numerical algorithms based upon the (full scale) IST have not so far shown a noticeable improvement over conventional methods of direct numerical integration of the KdV (see, e.g. [9, 14]). However direct numerical computations work best for small t and become unpractical for large times. The real power of the IST, on the other hand, is exactly in capturing the large time behavior of solutions to (1.1) (i.e. solitons) which is of particular interest in applications. That is to say, that the long-time asymptotic behavior of $u(x,t)$ is explicitly known in terms of $\{-\kappa_j^2, c_j, R(k)\}$ to any degree of accuracy in all physically important regions of the plane (x,t) (see, e.g. the recent expository paper [13] and the extensive literature cited therein). Loosely speaking, for our initial profile V the solution $u(x,t)$ evolves into a finite number of solitons and a dispersive tail. In the present paper we are concerned with the soliton part. It is well-known (see, e.g. [13]) that for our V in the soliton region; i.e., $x/t \geq C$ with some $C > 0$, for any positive l

$$u(x,t) = -2\sum_{j=1}^{J} \kappa_j^2 \operatorname{sech}^2(\kappa_j x - 4\kappa_j^3 t + \gamma_j) + O\big(t^{-l}\big), \tag{1.5}$$

where

$$\gamma_j = \ln\Big(\frac{\sqrt{2\kappa_j}}{c_j} \prod_{m=1}^{j-1} \frac{\kappa_j + \kappa_m}{\kappa_j - \kappa_m} \Big),$$

and the problem essentially boils down to effective computation of κ_j and c_j without the full machinery of the IST.

With our assumption that V has compact support, the reflection coefficients R and L are meromorphic functions in the upper half complex plane with simple poles at $\{i\kappa_j\}$, and the (left) norming constants $\{c_j\}$ can be retrieved from the residues of R at these poles. The poles of R can be numerically approximated by using root finders. However, computing residues is numerically more difficult. It is more convenient to introduce a related function B which is a rotation of the left reflection coefficient L. Then B has the same poles as R, and its corresponding residues are equal to the residues of R times a computable scaling factor. We give a new algorithm for computing the residues of B as presented below.

Our approach is based on approximating our potential V by N piecewise constant functions (using e.g. a Haar basis). The reflection and transmission coefficients L_n, R_n, and T_n for the n-th block can be explicitly derived in terms of elementary functions. The scattering coefficients L, R, and T for the whole approximation can then be recursively computed. More specifically, let

$$\Lambda = \begin{pmatrix} 1/T & -R/T \\ L/T & 1/\overline{T} \end{pmatrix}.$$

Then the reflection and transmission coefficients L, R, and T for the total potential can be derived from the principle of potential fragmentation (see, e.g. [7, 5]):

$$\Lambda = \Lambda_N \dots \Lambda_2 \Lambda_1,$$

where bars denote complex conjugation and Λ_n are the transition matrices

$$\Lambda_n = \begin{pmatrix} 1/T_n & -R_n/T_n \\ L_n/T_n & 1/\overline{T_n} \end{pmatrix}.$$

This gives us a recursive formula based on the Möbius transforms for the left and right reflection coefficients and also for the function B. Using this recursive formula for B, we can derive a recursive matrix formula for the residues of B at the poles in the upper-half plane. Consequently, we are able to evaluate the bound state data $\{\kappa_j, c_j\}$ and hence solve (1.1) numerically in the soliton region by (1.5).

Recursive methods similar in nature to the one we employ are quite common in physics and applied mathematics and have been used in a variety of physical situations related to wave propagation. For instance, back in the 50s one such method was used in the particularly influential paper [21] for the study of solids by reflection of X-rays. We also mention [15] where some important results on the run-up of solitary waves on bathymetries with piecewise linear topographies were obtained. In the mathematical literature what we call potential fragmentation is also referred to as layer stripping. We just name [23] where layer stripping was used in the context of radar imaging for both theoretical and computational purposes. However, besides [7, 5], we could not locate any literature where fragmentation would be used in connection with bound states. To deal with bound state data in this context one needs to use analyticity properties of the scattering coefficients in order to work with complex momentum k as opposed to real k considered in the previous literature. Computing residues of bulky recursively generated analytic functions does not look at first sight promising at all. A simple and effective way of handling recursion relations which produces surprisingly accurate results is the main contribution of the present paper.

We do not provide any error estimates; instead, the accuracy is verified on explicitly solvable examples. We also give a comparison of computing bound states as poles of R and B and computing norming constants with our algorithm as opposed to other common algorithms. Although our algorithm is slower than standard (unrelated to fragmentation) methods for obtaining bound state data, we demonstrate that it tends to be more accurate, especially for discontinuous initial profiles when conventional ODE solvers suffer from the Gibbs phenomenon. We also provide a comparison of the asymptotic solution to the KdV versus numerically integrated solutions. We will discuss some of them in the main body of the paper.

Lastly, we also note that there are many ways to improve our algorithm. For example, one can produce step-like approximations of our potentials $V(x)$ using Haar wavelets. The Haar wavelets are piecewise constant functions that form an orthogonal system. Wavelets are closely related to Fourier series, and they exhibit many properties that are numerically desirable. Furthermore, it is quite natural to consider piecewise linear approximations. The partial scattering coefficients L_n, R_n, and T_n are still explicit computable in terms of Airy functions. One can also use better root finders. For instance, it is not difficult to create an algorithm that captures the fact that if an extra fragment is added to the potential then bound states move in a certain way and new ones can only emerge from zero.

The referees pointed out the recent papers [8, 18]. The former is devoted to the numerical solution to the initial value problem for the focusing Nonlinear Schrödinger (NLS) equation. In this very interesting paper, the authors developed a new numerical algorithm for NLS which is also based upon the IST. Their method uses the structure of the Gelfand-Levitan-Marchenko kernel in an optimal way and produces very accurate numerical solutions to NLS with certain initial data. Being more subtle, the method requires more steps and appears to work best when the number of bound states is at most one. Extending their method to many bound states is left in [8] as an open problem. It would be also very interesting to compare their method adjusted to the KdV setting with ours. That could probably be done by applying the algorithms from [18] developed for numerical solution of the Marchenko integral equations for the 1D Schrö dinger equation. It is worth mentioning that [18] is one of very few papers devoted to numerical aspects of the Gelfand-Levitan-Marchenko inverse scattering procedure. The approach of [18] is based upon structured matrix systems.

The paper is organized as follows. In Section 2, we briefly introduce our main notation. In Section 3, we provide a brief overview of Scattering Theory for the one-dimensional time independent Schrödinger equation. In Section 4, we discuss the Inverse Scattering Transform and the asymptotic behavior of the KdV equation. In Section 5, we determine explicitly the scattering data in the case where the potential consists of a single well. In Section 6, using potential fragmentation we deduce some recursive relationships that yield the scattering data in the case where the potentials consists of multiple wells. In Section 7, we provide some numerical simulations to compare calculating the scattering data using our recursive methods as opposed to using traditional methods. Finally in Section 8, we discuss some possible improvements for our proposed algorithms; in particular, we discuss modifying the algorithm to utilize Haar wavelets.

2. Notation

We will denote the upper-half complex plane by $\mathbb{C}^+$. For a function $f:\mathbb{C}\to\mathbb{C}$, we will let $\overline{f}(z)$ denote complex conjugation and $\widetilde{f}(z)=f(-z)$ reflection. As is customary in analysis, we will let $L^2(\mathbb{R})$ be the class of functions f such that $\int_{\mathbb{R}}|f|^2<\infty$. We will let $L^1_1(\mathbb{R})$ denote the class of functions f such that $\int_{\mathbb{R}}(1+|x|)|f(x)|<\infty$. Given functions $f,g\in L^2(\mathbb{R})$, we define the L^2 inner product to be

$$\langle f,g\rangle_2=\int_{\mathbb{R}}f\overline{g}.$$

and the L^2-norm $\|\cdot\|_2$ to be the norm with respect to this inner product, i.e.

$$\|f\|_2 = \Big[\int_{\mathbb{R}} |f|^2\Big]^{1/2}.$$

For a set $A \subseteq \mathbb{R}$, χ_A will denote the characteristic function on A; i.e.

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases}$$

3. Direct scattering theory for the Schrödinger equation on the line

Consider the KdV equation

$$u_t - 6uu_x + u_{xxx} = 0.$$

A particular stable solution of the KdV equation is given by

$$u(x,t) = -2\kappa^2 \operatorname{sech}^2(\kappa x - 4\kappa^3 t + \gamma)$$

where κ and γ are real constants. Solutions of this form are called *solitons.* For more general initial profiles $u(x,0)$, one applies the inverse scattering formalism or IST (see e.g. [1]). The linearization process of the IST consists of first finding a pair of linear operators L, B, known as the *Lax pair*, satisfying

$$\begin{cases} L\phi = \lambda\phi \\ \phi_t = B\phi \end{cases}.$$

The operators L, B are constructed in terms of $u(x,t)$ and its partial derivatives such that the nonlinear evolution equation one wishes to solve then appears as a compatibility condition for

$$L_t = BL - LB$$

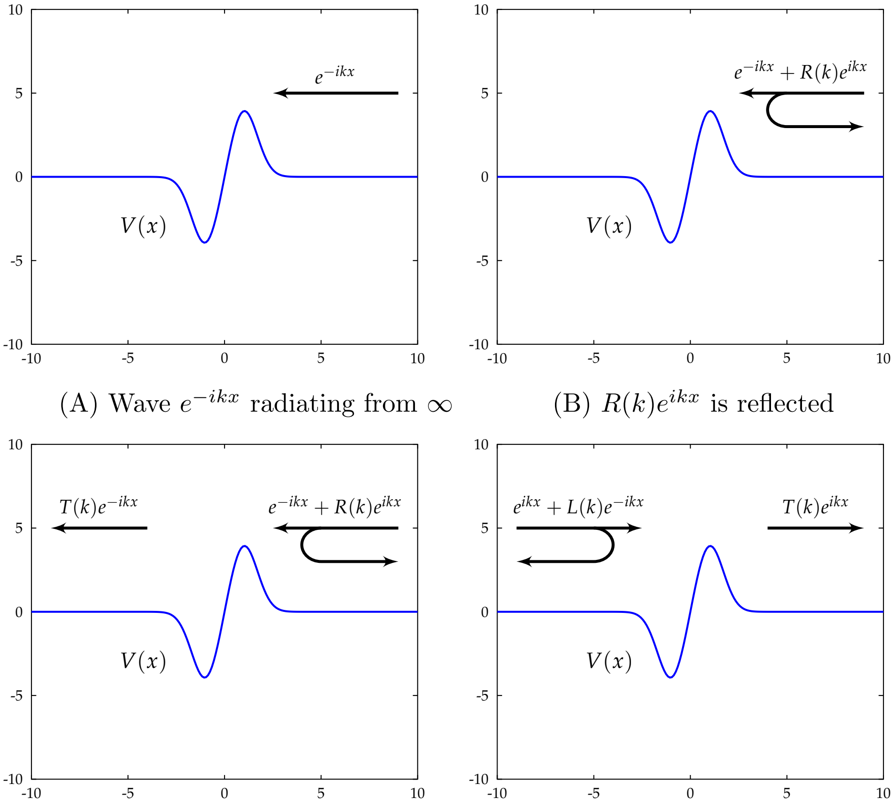
leading to $\lambda_t = 0$. The eigenvalue problem associated to L in $L^2(\mathbb{R})$ is known as the scattering problem whereas B is used to determine the time evolution of the system. In the KdV case, one easily verifies that L and B can be chosen as follows:

$$L = -\frac{\partial^2}{\partial x^2} + u(x,t)$$
$$B = -4\frac{\partial^3}{\partial x^3} + 3\Big(u(x,t)\frac{\partial}{\partial x} + \frac{\partial}{\partial x}u(x,t)\Big).$$

But the equation $-\phi_{xx} + u(x,t)\phi = \lambda\phi$ In direct scattering, one considers the initial profile $u(x,0) = V(x)$, and solve

$$-\phi_{xx} + V(x)\phi = \lambda\phi.$$

Let us define H as the Schrödinger operator associated to the initial profile V, i.e. $H = -\frac{d^2}{dx^2} + V(x)$. Suppose further that $V \in L_1^1(\mathbb{R})$. Under this condition, for each $\lambda > 0$, there is a nontrivial solution (generalized eigenfunction) to $H\phi = \lambda\phi$ that behaves asymptotically like a sinusoid (see e.g. [11]). However, these eigenfunctions ϕ are not contained in $L^2(\mathbb{R})$. We call the set of such λ the continuous spectrum of H. There may also be finitely many negative eigenvalues, called bound states [11]. The negative eigenvalues give square-integrable eigenfunctions ϕ. The set of bound states is called the discrete spectrum. The continuous spectrum gives rise to a component of the solution of the KdV which acts like a solution to the linear equation $u_t + u_{xxx} = 0$. This part of the solution is the dispersive portion of the



(A) Wave e^{-ikx} radiating from ∞

(B) $R(k)e^{ikx}$ is reflected

(C) $T(k)e^{-ikx}$ is transmitted

(D) Wave coming from $-\infty$

FIGURE 1. The scattering solutions $\phi_r(x)$ and $\phi_l(x)$ as waves radiating from $\pm\infty$

wave and becomes of negligible amplitude for large times. The discrete spectrum gives rise to the solitons. This portion of the solution of the KdV is structurally stable in that each soliton's shape and velocity is preserved over time (outside of brief-in-time elastic interactions) as it moves to the right. Thus, we focus on knowing the discrete spectrum for large times.

Suppose $\lambda = k^2 \in \mathbb{R}$. In the left and right scattering solutions to the Schrödinger equation given by respectively (1.3) and (1.4), one can view ϕ_r as a wave e^{-ikx} radiating from infinity, and $R(k)e^{ikx}$ is the portion of the wave that is reflected while $T(k)e^{-ikx}$ is the portion that is transmitted (see Figure 1). Hence the terminology of *right reflection coefficient* for $R(k)$[1]and *transmission coefficient* for $T(k)$. [1]Similarly, for ϕ_l, $T(k)$ is the same transmission coefficient and $L(k)$ is the *left reflection coefficient*.

4. The classical inverse scattering transform

Since $V(x) \in L_1^1(\mathbb{R})$, we have that there are finitely many bound states $\lambda = k^2$ where $k = i\kappa$. Let J denote the number of bound states, and let

$$\kappa_1 > \kappa_2 > \cdots > \kappa_J > 0.$$

Let c_j denote the left norming constant at $k = i\kappa_j$; i.e.,

$$c_j = \|\phi_1(x, i\kappa_j)T^{-1}(i\kappa_j)\|_2^{-1}.$$

Once we know the scattering data for the Schrödinger operator, we can use the IST to obtain the soliton solutions of the KdV equation.

$$\begin{tikzcd}
u(x,0) \arrow[r, "\text{direct scattering}"] & S(0) \arrow[d, "\text{time evolution}"] \\
u(x,t) & S(t) \arrow[l, "\text{inverse scattering}"]
\end{tikzcd}$$

In the Direct Scattering step, we map the initial potential $u(x, 0)$ into the scattering data

$$S(0) = \{\{-\kappa_j^2, c_j\}_{j=1}^J, R(k), k \in \mathbb{R}\}.$$

Next, we evolve the scattering data over time in a simple fashion:

- $\kappa_j(t) = \kappa_j$,
- $c_j(t) = c_j e^{4\kappa_j^3 t}$,
- $R(k,t) = R(k)e^{8ik^3 t}$.

Then the scattering data becomes

$$S(t) = \{\{-\kappa_j(t)^2, c_j(t)\}_{j=1}^J, R(k,t), k \in \mathbb{R}\}.$$

We can reclaim the solution to the KdV using Inverse Scattering as follows:

- Form the Gelfand-Levitan-Marchenko (GLM) kernel:
$$F(x,t) = \sum_{j=1}^J c_j^2(t)e^{-\kappa_j x} + \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx} R(k,t)dk.$$
- Solve the Gelfand-Levitan-Marchenko equation for $K(x, y, t)$, $y \geq x$:
$$K(x,y,t) + F(x+y,t) + \int_x^\infty F(s+y,t)K(x,s,t)ds = 0.$$
- The solution to the KdV equation is
$$u(x,t) = -2\frac{d}{dx}K(x,x,t).$$

Luckily, for large times t we can simplify the GLM kernel. Slightly adjusting arguments used to prove the Riemann-Lebesgue lemma, we have that

$$\int_{-\infty}^{\infty} e^{i(kx+k^3t)} R(k)dk \to 0 \text{ as } t \to \infty$$

for every x. Thus, for large times we can approximate the GLM kernel by

$$F(x,t) \approx \sum_{j=1}^J c_j^2(t)e^{-\kappa_j x}.$$

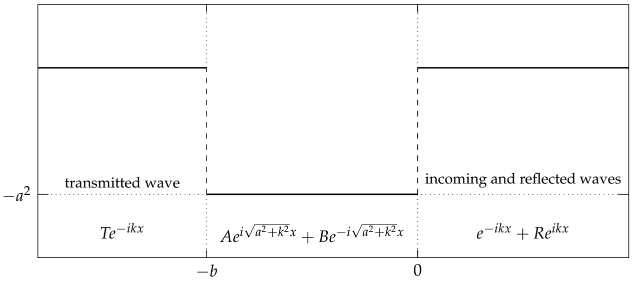

$-a^2$
transmitted wave
incoming and reflected waves
Te^{-ikx}
$Ae^{i\sqrt{a^2+k^2}x} + Be^{-i\sqrt{a^2+k^2}x}$
$e^{-ikx} + Re^{ikx}$
$-b$
0

FIGURE 2. The setup for a single block potential

Let

$$C(x,0) = \begin{pmatrix} c_{11}(x) & c_{12}(x) & \dots & c_{1,J}(x) \\ c_{21}(x) & c_{22}(x) & \dots & c_{2,J}(x) \\ \vdots & & \ddots & \\ c_{J,1}(x) & c_{J,2}(x) & \dots & c_{J,J}(x) \end{pmatrix}$$

where

$$c_{mj}(x) = \frac{c_m c_j}{\kappa_m + \kappa_j} e^{-(\kappa_m+\kappa_j)x}.$$

The matrix C evolves in time by

$$c_{mj}(x,t) = c_{mj}(x) e^{4(\kappa_m^3+\kappa_j^3)t}.$$

Then for large times, our solution to the KdV is [1]

$$u(x,t) \approx -2\frac{\partial^2}{\partial x^2} \ln[\det(I + C(x,t))]$$

From this, one obtains the asymptotic formula (1.5). Notice that the large time solution $u(x,t)$ of the KdV behaves like a finite sum of single solitons. Moreover, we no longer need to do the full IST to solve the KdV for large times. We need only find the bound states $-\kappa_j^2$ and norming constants c_j.

The coefficients R and T can be analytically continued to $\mathbb{C}^+$, and their poles are precisely $i\kappa_j$. That is, all of the poles of R and T in $\mathbb{C}^+$ lie on the imaginary axis and correspond with the bound states. Better yet, these poles are actually simple [6, 19]. Furthermore, if we assume that V is supported on $(-\infty, 0)$, then [2]

$$\operatorname{Res}_{k=i\kappa_j} R(k) = ic_j^2. \tag{4.1}$$

Consequently, in this case, the bound states and norming constants can be obtained from knowledge of $R(k)$ for $k \in \mathbb{C}^+$, and we can approximate the solution of the KdV for large times from only the knowledge of $R(k)$ for $k \in \mathbb{C}^+$.

5. The scattering quantities for a block (well) potential

Consider the case when our potential V is a single nonpositive well which is $-a^2$ on the interval $[-b, 0]$ and 0 elsewhere, i.e. $V(x) = -a^2\chi_{[-b,0]}(x)$ (see Figure 2).

In this case, we can obtain an exact solution to the Schrödinger equation. Moreover, using the continuity of the solution ϕ and its derivative ϕ_x, we can set up a system of equations and solve for R and T. Doing this, we obtain

$$R(k)=\omega^2\frac{1-\xi}{\xi-\omega^4}\,,\quad L(k)=\omega^2\frac{1-\xi}{\xi-\omega^4}e^{-iab(\omega-1/\omega)}\,,\quad T(k)=\frac{1-\omega^4}{\xi-\omega^4}e^{i\frac{ab}{\omega}} \tag{5.1}$$

where

$$\omega=\frac{k}{a}+\sqrt{\big(\frac{k}{a}\big)^2+1}\,,\quad \xi=e^{i(\omega+1/\omega)ab}.$$

These formulas for R, L, and T are actually meromorphic in $\mathbb{C}$ if we choose the branch cut along the imaginary axis between $-ia$ and ia. Using these formulas, R, L, and T can be analytically continued in $\mathbb{C}^+$. The only difficulty lies in considering the branch cut. However, we have that $\omega(-\overline{k})=-\overline{\omega(k)}$ and $\xi(-\overline{k})=\overline{\xi(k)}$. It follows that $R(-\overline{k})=\overline{R(k)}$ and $T(-\overline{k})=\overline{T(k)}$. For $k\in i\mathbb{R}$, we have that $R(k)=R(-\overline{k})=\overline{R(k)}$ and $T(k)=T(-\overline{k})=\overline{T(k)}$, so R and T are real-valued for $k\in i\mathbb{R}$. For $k=+0+i\kappa$, we have that $-\overline{k}=-0+i\kappa$. Therefore, $R(-0+i\kappa)=\overline{R(+0+i\kappa)}$ and since R is real-valued on $i\mathbb{R}$, $\overline{R(+0+i\kappa)}=R(+0+i\kappa)$. Hence, $R(-0+i\kappa)=R(+0+i\kappa)$, so R is continuous along the branch cut between $-ia$ and ia. It follows that R is meromorphic in $\mathbb{C}$. Similarly, T is meromorphic in $\mathbb{C}$ as well.

Consider the poles $i\kappa_j$ and residues ic_j^2 of R. The poles of R and T satisfy $\xi=\omega^4$. If we let $y_j=\frac{\kappa_j}{a}$, then κ_j and c_j can be explicitly computed by the following formulas:

$$\frac{ab}{\pi}\sqrt{1-\big(\frac{\kappa_j}{a}\big)^2}-\frac{2}{\pi}\arctan\frac{\kappa_j}{a\sqrt{1-\big(\frac{\kappa_j}{a}\big)^2}}=j-1 \tag{5.2}$$

and

$$c_j^2=\frac{2\kappa_j\big(1-(\frac{\kappa_j}{a})^2\big)}{2+b\kappa_j} \tag{5.3}$$

for $j=1,\dots,\lceil\frac{ab}{\pi}\rceil$.

6. The potential fragmentation and the scattering quantities for potentials composed of blocks

We define the scattering matrix to be

$$S=\begin{pmatrix}T & R\\ L & T\end{pmatrix}$$

The matrix S is unitary; i.e., $S^{-1}=S^*$ where S^* is the conjugate transpose of S [6]. This gives us a few identities, namely [3, 19]

$$L\overline{T}+T\overline{R}=0 \tag{6.1}$$

for $k\in\mathbb{R}$. If we were to shift our potential to the right by p, then the scattering matrix would change as follows [7]:

$$L(k)\quad\to\quad L(k)e^{2ikp} \tag{6.2}$$

$$T(k)\quad\to\quad T(k) \tag{6.3}$$

$$R(k)\quad\to\quad R(k)e^{-2ikp} \tag{6.4}$$

Now suppose that our potential V consists of N nonpositive blocks. Let R_n, L_n, T_n be the reflection and transmission coefficients on the n -th block: $V_n(x)=-a_n^2$ on $[-b_n,-b_{n-1}]$ where $b_0=0$. Let R_n^0,L_n^0,T_n^0 be the reflection and transmission coefficients on the n-th block shifted to the origin: $V_n(x)=-a_n^2$ on $[-(b_n-b_{n-1}),0]$.

Let $R_{1,2,\dots,n}$, $L_{1,2,\dots,n}$, $T_{1,2,\dots,n}$ be the reflection and transmission coefficients on the first n blocks. R, L, T without subscripts or superscripts will denote the reflection and transmission coefficients for the overall potential.

Let

$$\Lambda = \begin{pmatrix} 1/T & -R/T \\ L/T & 1/\overline{T} \end{pmatrix}. \tag{6.5}$$

The fragmentation principle, or layer stripping principle as it is also known [7, 5, 3, 23], says that for $k \in \mathbb{R}$

$$\Lambda = \Lambda_N \dots \Lambda_2 \Lambda_1 \tag{6.6}$$

where Λ_n are the transition matrices

$$\Lambda_n = \begin{pmatrix} 1/T_n & -R_n/T_n \\ L_n/T_n & 1/\overline{T_n}. \end{pmatrix}$$

Note that blocks with $a_n = 0$ may be simply ignored since this implies Λ_n is the identity matrix. Also note that for $f \in \{R, L, T\}$ with any potential and for all $k \in \mathbb{C}$, gives us that

$$\widetilde{f}(k) = f(-k) = \overline{f(\overline{k})}. \tag{6.7}$$

Using this fact, all $\widetilde{f}$ and $\overline{f}$ where $f \in \{R, L, T\}$ are interchangeable for $k \in \mathbb{R}$ but not for general complex k.

We can use potential fragmentation to come up with some recursive formulas. Using (6.1)-(6.7), we obtain that for $k \in \mathbb{R}$

$$R_{1,\dots,n+1} = -\frac{L_{1,\dots,n}}{\widetilde{R}_{1,\dots,n}} \frac{R^0_{n+1} e^{2ikb_n} - \widetilde{L}_{1,\dots,n}}{1 - R^0_{n+1} L_{1,\dots,n} e^{2ikb_n}}. \tag{6.8}$$

A similar expression may be obtained for the left reflection coefficient:

$$L_{1,\dots,n+1} = -\frac{R^0_{n+1}}{\widetilde{R}^0_{n+1}} \frac{L_{1,\dots,n} e^{2ikb_n} - \widetilde{R}^0_{n+1}}{1 - R^0_{n+1} L_{1,\dots,n} e^{2ikb_n}} e^{-2ikb_{n+1}}. \tag{6.9}$$

Since $|L_{1,\dots,n}| = |R_{1,\dots,n}|$ for $k \in \mathbb{R}$, we have $L_{1,\dots,n} = R_{1,\dots,n} e^{-2ik\beta_n}$ for some $\beta_n : \mathbb{R} \to \mathbb{R}$. Equations (6.8) and (6.9) then give us that

$$R_{1,\dots,n+1} = -\frac{R_{1,\dots,n}}{\widetilde{R}_{1,\dots,n}} \frac{R^0_{n+1} e^{2ik(b_n - \beta_n)} - \widetilde{R}_{1,\dots,n}}{1 - R^0_{n+1} R_{1,\dots,n} e^{2ik(b_n - \beta_n)}}, \tag{6.10}$$

where $\beta_1 = b_1$ and

$$e^{-2ik\beta_{n+1}} = \frac{R^0_{n+1}}{\widetilde{R}^0_{n+1}} \frac{\widetilde{R}_{1,\dots,n}}{R_{1,\dots,n}} \frac{R_{1,\dots,n} e^{2ik(b_n - \beta_n)} - \widetilde{R}^0_{n+1}}{R^0_{n+1} e^{2ik(b_n - \beta_n)} - \widetilde{R}_{1,\dots,n}} e^{-2ikb_{n+1}}. \tag{6.11}$$

Define $A_n = \frac{L_{1,\dots,n}}{R_{1,\dots,n}} e^{2ikb_n}$. Then $A_n = e^{2ik(b_n - \beta_n)}$ for $k \in \mathbb{R}$. Equations (6.10) and (6.11) give us that

$$R_{1,\dots,n+1} = -\frac{R_{1,\dots,n}}{\widetilde{R}_{1,\dots,n}} \frac{A_n R^0_{n+1} - \widetilde{R}_{1,\dots,n}}{1 - A_n R^0_{n+1} R_{1,\dots,n}} \tag{6.12}$$

and

$$A_{n+1} = \frac{R^0_{n+1}}{\widetilde{R}^0_{n+1}} \frac{\widetilde{R}_{1,\dots,n}}{R_{1,\dots,n}} \frac{A_n R_{1,\dots,n} - \widetilde{R}^0_{n+1}}{A_n R^0_{n+1} - \widetilde{R}_{1,\dots,n}} \tag{6.13}$$

where $A_1 = 1$. Let us next define $B_n = A_n R_{1,\dots,n} = L_{1,\dots,n} e^{2ikb_n}$. Then we get the following recursive formula:

$$B_{n+1} = -\frac{R^0_{n+1}}{\widetilde{R}^0_{n+1}} \frac{B_n - \widetilde{R}^0_{n+1}}{1 - R^0_{n+1} B_n} \tag{6.14}$$

where $B_1 = R_1$. Notice that B_{n+1} is a Möbius transform of B_n, and that the recursive formula for B_n is much simpler than the recursive formula for $R_{1,\dots,n}$. Moreover, this formula only depends on B_n, R^0_{n+1}, and $\widetilde{R}^0_{n+1}$. From (5.1),

$$R^0_n = \frac{\omega_n^2(1-\xi_n)}{\xi_n - \omega_n^4}. \tag{6.15}$$

where $h_n = b_n - b_{n-1}$ is the width for the n-th block,

$$\omega_n(k) = \frac{k}{a_n} + \sqrt{\big(\frac{k}{a_n}\big)^2 + 1},$$

and $\xi_n(k) = e^{-ia_n h_n(\omega_n(k) + 1/\omega_n(k))}$. For $k \in \mathbb{R}$, we have that $\overline{R^0_n} = \widetilde{R}^0_n$. By taking the complex conjugate of (6.15), we obtain that for $k \in \mathbb{R}$,

$$\widetilde{R}^0_n = \frac{\omega_n^2(1-\xi_n)}{\xi_n \omega_n^4 - 1}. \tag{6.16}$$

Since R^0_n is meromorphic in $\mathbb{C}$, $\widetilde{R}^0_n$ is meromorphic in C as well (in particular, both are meromorphic in $\mathbb{C}^+$ where the poles of interest lie). Since the formula in (6.16) is meromorphic in $\mathbb{C}$, it follows that (6.16) holds for all $k \in \mathbb{C}$. Continuing inductively using equations (6.12)-(6.14), it follows that $R_{1,\dots,n}$, A_n, and B_n can be continued to meromorphic functions in $\mathbb{C}$ (and in particular, $\mathbb{C}^+$) for all $1 \le n \le N$.

Since $B_n = A_n R_{1,\dots,n} = L_{1,\dots,n} e^{2ikb_n}$, we have that B_n has the same poles $k = i\kappa_j$ in $\mathbb{C}^+$ as $L_{1,..,n}$. Consequently, B_n and $R_{1,\dots,n}$ have the same poles in $\mathbb{C}^+$. Since the poles $k = i\kappa_j$ in $\mathbb{C}^+$ of $L_{1,\dots,n}$ and $R_{1,\dots,n}$ are simple, we have that $A_n = \frac{L_{1,\dots,n}}{R_{1,\dots,n}} e^{2ikb_n}$ is analytic in $\mathbb{C}^+$ and nonzero at all $k = i\kappa_j$. It follows from equation (4.1) then that

$$\operatorname{Res}_{k=i\kappa_j} B_n = A_n(i\kappa_j) \operatorname{Res}_{k=i\kappa_j} R_{1,\dots,n} = ic_j^2 A_n(i\kappa_j) \tag{6.17}$$

The value of $A_n(i\kappa_j)$ can be determined via the recursive formula (6.13). If we can determine an algorithm for determining the residues of B_N, then this would effectively give us an algorithm for calculating the (left) norming constants.

Now suppose B_n has the form

$$B_n = -\frac{R^0_n}{\widetilde{R}^0_n} \frac{p_n}{q_n}. \tag{6.18}$$

Applying (6.14), we get a linear system of recurrence relations for p_n and q_n:

$$\begin{gathered} p_{n+1} = -R^0_n p_n - \widetilde{R}^0_{n+1} \widetilde{R}^0_n q_n \\ q_{n+1} = R^0_{n+1} R^0_n p_n + \widetilde{R}^0_n q_n \end{gathered}$$

or in matrix form

$$\begin{pmatrix} p_{n+1} \\ q_{n+1} \end{pmatrix} = M_n \begin{pmatrix} p_n \\ q_n \end{pmatrix} = M_n \dots M_2 M_1 \begin{pmatrix} p_1 \\ q_1 \end{pmatrix} \tag{6.19}$$

where

$$M_m = \begin{pmatrix} -R^0_m & -\widetilde{R}^0_{m+1}\widetilde{R}^0_m \\ R^0_{m+1}R^0_m & \widetilde{R}^0_m \end{pmatrix}.$$

Let N denote the number of blocks. If $q_N(k) = 0$ but k is not a pole of B_N, then $p_N = 0$ as well. From (6.19), this means that $\det(M_n) = 0$ for some $1 \le n \le N-1$ or $\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = 0$. Since $B_1 = R_1$, from (6.18) we have that $\frac{p_1}{q_1} = -\widetilde{R}_1$. Our choice of p_1 and q_1 is arbitrary, as long as this ratio is preserved, since our resulting solution of B_{n+1} is independent of our choice for p_1 and q_1. Some choices for our initial vector may be preferable for numerical computations, but for our purposes we will choose $\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} -\widetilde{R}_1 \\ 1 \end{pmatrix}$, because it is nonzero for all k. Thus, if $q_N = 0$ but k is not a pole of B_N, then $\det(M_{N-1} \dots M_2M_1) = 0$. Equivalently, if $q_N = 0$ and $\det(M_{N-1} \dots M_2M_1) \neq 0$, then k is a pole of $R_{1,\dots,N}$.

We claim that $\det(M_{N-1} \dots M_2M_1)(k) = 0$ for some $k \in \mathbb{C}^+$ if and only if $k = ia_N$, or for some $1 \le n \le N-1$ and some $0 \le j \le \lfloor \frac{a_n h}{\pi} \rfloor$,

$$k = i\sqrt{a_n^2 - (\frac{\pi j}{h_n})^2}. \tag{6.20}$$

We have that $\det(M_{N-1} \dots M_2M_1) = 0$ if and only if $\det(M_n) = 0$ for some $1 \le n \le N-1$. Moreover,

$$\det(M_n) = R^0_n\widetilde{R}^0_n(R^0_{n+1}\widetilde{R}^0_{n+1} - 1).$$

Thus, $\det(M_n) = 0$ if and only if $R^0_n = 0$ (equivalently $\widetilde{R}^0_n = 0$) or $R^0_{n+1}\widetilde{R}^0_{n+1} = 1$. The second case occurs when

$$\omega^4_{n+1}(1 - \xi_{n+1})^2 = (\xi_{n+1} - \omega^4_{n+1})(\xi_{n+1}\omega^4_{n+1} - 1).$$

After some algebra and noting that $\xi_{n+1} = e^{-ia_{n+1}h_{n+1}(\omega_{n+1}+1/\omega_{n+1})} \neq 0$, this simplifies to $\omega^4_{n+1} = 1$. A simple calculation then gives us that $\omega^4_{n+1} = 1$ for $k \in \mathbb{C}^+$ if and only if $k = ia_{n+1}$. After a lengthy computation using (6.15), we obtain that $R^0_n(k) = 0$ for $k \in \mathbb{C}^+$ if and only if equation (6.20) holds.

Now suppose that $q_N(k) = 0$, $\det(M_{N-1} \dots M_2M_1)(k) \neq 0$ at $k = i\kappa$, and that k is not a pole of $\widetilde{R}^0_N$. Then k is a pole of B_N, and consequently a pole of $R = R_{1,\dots,N}$ as well. Consequently, k^2 is a bound state of the Schrödinger equation. Since $\det(M_{N-1} \dots M_2M_1)(k) \neq 0$, we have that $p_N(k) \neq 0$. Since k is not a pole of $\widetilde{R}^0_N$ and since R^0_N and $\widetilde{R}^0_N$ have the same zeros with the same multiplicity, $-\frac{R^0_N}{\widetilde{R}^0_N}p_n \neq 0$. However, $q_N = 0$ and the poles of B_N are simple, so

$$\operatorname{Res}_{k=i\kappa} B_N = -\frac{R^0_N}{\widetilde{R}^0_N}\frac{p_N}{q'_N}. \tag{6.21}$$

To find q'_N, we can differentiate (6) to acquire

$$\begin{pmatrix} p'_{n+1} \\ q'_{n+1} \end{pmatrix} = M_n \begin{pmatrix} p'_n \\ q'_n \end{pmatrix} + M'_n \begin{pmatrix} p_n \\ q_n \end{pmatrix}. \tag{6.22}$$

Therefore, for the poles of R where $\det(M_N \dots M_2M_1) \neq 0$ and that are not poles of $\widetilde{R}^0_N$, the residues of R can be recovered through (6) and (6.21).

7. Numerical simulations

Tables 1 and 3 give a comparison of the algorithms listed below for calculating bound states, while tables 2 and 4 give a comparison of the algorithms listed below for calculating norming constants. The exact bound states in table 1 were calculated using equation (5.2). All calculations were performed using MATLAB, all integrals were approzimated using the trapezoidal method, and all differential equations were numerically integrated using the Runge-Kutta 4-5 algorithm.

There are two commonly used numerical methods for approximating the bound states:

(1) Matrix methods - Estimate the Schrödinger operator $H = -\frac{d^2}{dx^2} + V(x)$ using a finite-dimensional matrix and find the eigenvalues of the matrix. In particular, [24] describes how this can be done using the Fourier basis. In tables 1 and 3, a 512×512 matrix is used.

(2) Shooting Method - The Shooting Method involves recursively choosing values of λ and "shooting" from both end points to a point $c \in [a, b]$. Define the miss-distance function to be the Wronskian determinant
$$D(\lambda) = \begin{vmatrix} u_L(c,\lambda) & u_R(c,\lambda) \\ u'_L(c,\lambda) & u'_R(c,\lambda) \end{vmatrix}.$$
where u_L is the interpolated function from the left endpoint and u_R is from the right endpoint. If λ is an eigenvalue that satisfies the boundary value problem, then $D(\lambda) = 0$. For more details, see for example [22].

There are of course other existing methods for approximating bound states; see for example [12, 16, 17, 10]. However, we will only focus on these two.

If one approximates the potential using finitely many blocks, then we can use the following algorithms for estimating bound states:

(3) Use the recursive formulas (6.12) and (6.13) to find the bound states as zeros of $1/R_{1,\dots,N}$.

(4) Similarly, one can use the recursive formula (6.14) to find the bound states as zeros of $1/B_N$.

(5) Using (6.19), the bound states can be found as zeros of q_N. One must also check the values of k listed in (6.20) where $\det(M_{N-1}\dots M_1) = 0$.

Theoretically, algorithms (3)-(5) are very similar to one another; all of them involve finding bound states as zeros of functions that are multiples of each other by a well-behaved nonzero multiplier. However, computationally these are different from one another. Algorithm (3) involves a pair of recursive equations that must be calculated in tandem. Algorithm (4) involves a simple single recursive equation and is the least computationally expensive of these three. Algorithm (5) involves a recursive matrix equation. A natural question is how do these three differ from each other computationally in terms of accuracy and stability.

Algorithm (1) seems to be the fastest of these algorithms, followed closely by (2). Moreover, algorithm (1) has great accuracy when the initial potential is smooth. However, for discontinuous potentials, the Gibbs phenomenon severely hinders the accuracy of the algorithm. All of algorithms (2)-(5) rely on finding roots of some function, so inheritently all of these functions have all of the problems that root finders tend to have. In particular, for general potentials, the exact number of bound states is unknown, hence one does not know how many roots to search for. In practice, one could partition the positive imaginary axis and search for roots in each interval. However, the question is still open as to how small the width of each interval needs to be. This is further complicated by the fact that in the case of a single block, the bound states are known to cluster towards zero as the width of the block goes to infinity (this can be easily derived from (5.2)). Lastly, for root finders that do not incorporate the bisection method, given a good initial approximation of a bound state, the root finder might converge to the wrong bound state. This leads to multiple approximations to the same bound state that appear to be different bound states; the clustering effect makes it difficult to spot these repeats in the case noted above. However, for the bound states that are calculated, algorithms (3)-(5) are extremely accurate. Algorithms (3)-(5) also seem to be much slower than algorithms (1) and (2), with (5) being the slowest and most computationally expensive.

In summary, the commonly used algorithms (1) and (2) for calculating bound states are much faster than the other algorithms. Moreover, algorithm (1) tends to be extremely accurate, especially when the potential is smooth. However, although algorithms (3)-(5) are much slower, they also tend to be very accurate, especially with discontinuous potentials.

Supposing the bound states have been calculated, tables 2 and 4 give a comparison of some of the various algorithms for computing (left) norming constants. First is the algorithm described in the present paper:

(i) The potential is approximated using finitely many blocks, and the norming constants are calculated as residues via equations (6.21) and (6).

Next we have the obvious algorithm using the definition of the left norming constant:

(ii) Suppose V has compact support $[A, B]$. Then $\phi(x,k) = \phi_1(x,k)/T(k)$ satisfies $\phi(x,k) = e^{ikx}$ for $x \geq B$. One can numerically integrate the Schrödinger equation from B to A. Then $c^2 = \|\phi_1\|_2^{-1}$, which can be numerically integrated.

The authors were also presented the following algorithms by Paul Sacks: letting $a = 1/T$ and $b = -R/T$, then $R = -\frac{b}{a}$ and the transition matrix Λ given in (6.5) becomes

$$\Lambda = \begin{pmatrix} a & b \\ \widetilde{b} & \widetilde{a} \end{pmatrix}.$$

Moreover, b is analytic everywhere in $\mathbb{C}^+$, and the simple poles of T in $\mathbb{C}^+$ are simple zeros of a. Consequently, (4.1) gives us that

$$c_j^2 = i\frac{b(i\kappa_j)}{a'(i\kappa_j)}.$$

The derivative a' with respect to k can be approximated using the central difference

$$a'(k) \approx \frac{a(k+\eta/2) - a(k-\eta/2)}{\eta}.$$

The question then becomes how one evaluates $a(k)$ and $b(k)$. Here are two approaches:

(iii) The potential is approximated using a finite number of blocks, and a and b are calculated using potential fragmention (6.6). The transition matrices are evaluated using equation (5.1).

Table 1. $V(x) = -4\chi_{[-4,0]}(x)$, domain chosen $[-10, 10]$, spacial step size $h = 0.01$

Algor.	κ_1	κ_2	κ_3	Rel. Error	Time (sec)
Exact	1.899448036751944	1.571342556813314	0.876610362727433	0	0.004355000
(1)	1.898826427139628	1.568514453040000	0.867505110670815	365 E-5	0.126239000
(2)	1.899418261950639	1.572105829640451	0.872097420881459	175 E-5	0.505034000
(3)	1.899448036751942	1.571342556813313	0.876610362727439	3 e-15	4.168762000
(4)	1.899448036751949	1.571342556813312	0.876610362727428	3 e-15	5.425778000
(5)	1.899448036751942	1.571342556813315	0.876610362727434	1 e-15	10.268152000

Table 2. $V(x) = -4\chi_{[-4,0]}(x)$, domain chosen $[-4, 0]$, spacial step size $h = 0.01$, energy step size $\eta = 0.001$, exact bound states used

Algor.	c_1^2	c_2^2	c_3^2	Rel. Error	Time (sec)
Exact	0.038798932148319	0.145167980693995	0.257227284424067	0	0.005992000
(i)	0.038798932148326	0.145167980694058	0.257227284424741	227 E-14	2.008827000
(ii)	0.039619680931665	0.160080616838866	0.364236083957119	363 E-3	0.032151000
(iii)	0.038798937542783	0.145168027811526	0.257226712349713	193 E-8	2.070128000
(iv)	0.051311576782601	0.109225786002665	-0.041977058580690	1.012	0.147137000

(iv) Supposing the potential has compact support $[\alpha, \beta]$, the Schrödinger equation can be numerically integrated from α to β with the initial conditions $\phi(\alpha, k) = e^{-ik\alpha}$, $\phi'^{-ik\alpha}$. Then $\phi(x, k) = \phi_r(x, k)/T(k)$, so for $x \geq \beta$

$$\phi(x, k) = a(k)e^{-ikx} - b(k)e^{ikx}.$$

Consequently, a and b can be retrieved from

$$\begin{pmatrix} a(k) \\ b(k) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} e^{ik\beta} & \frac{ie^{ik\beta}}{k} \\ -e^{-ik\beta} & \frac{ie^{-ik\beta}}{k} \end{pmatrix} \begin{pmatrix} \phi(\alpha, k) \\ \phi'(\alpha, k) \end{pmatrix}.$$

Algorithms (ii) and (iv) seem to be the fastest of these four algorithms. For smooth potentials, algorithm (iii) seems to be the most accurate with the other three algorithms being approximately the same order of accuracy. However, for discontinuous potentials, algorithm (i) seems to be the most accurate and algorithm (iv) is the least accurate. Since algorithms (ii) and (iv) involve numerically integrating the Schrödinger equation, these two algorithms should do well with smooth potentials and horribly with discontinuous ones. Since algorithms (i) and (iii) involve approximating the initial potential with step functions, one would expect that these algorithms would do better for discontinuous potentials than with smooth ones. What is surprising is that these algorithms seem to do about as well if not better than algorithms (ii) and (iv) even for smooth potentials. Moreover, the accuracy of algorithms (i) and (iii) increases when the bound states are approximated using algorithms (3)-(5). Furthermore, as we discuss in the next section, algorithms (i) and (iii) can be revised to use higher order spline interpolants of the initial potential, leading to even greater accuracy for smooth potentials with very little change in computational time.

Lastly, Figures 3 and 4 compare the asymptotic formula given in [1] with the numerically integrated solution obtained by using the split step Fourier method. In Figure 3, the initial potential is smooth, giving great accuracy for the split step Fourier method. However, in Figure 4, the potential is discontinuous, giving extra noise in the solution from the split step Fourier method. Despite this, the soliton solutions closely match the asymptotic solution.

TABLE 3. $V(x) = -2\,\mathrm{sech}^2(x)$, domain chosen $[-5,5]$, spacial step size $h = 0.01$

Algorithm	κ	*Relative Error*	*Time (sec)*
Exact	*1.000000000000000*	*0*	*0*
(1)	*1.000181385743159*	*0.000181385743159*	*0.123699000*
(2)	*1.000010661550817*	*0.000010661550817*	*0.165820000*
(3)	*0.999997769556372*	*0.000002230443628*	*8.624536000*
(4)	*0.999997769556371*	*0.000002230443629*	*8.780760000*
(5)	*0.999997769556372*	*0.000002230443628*	*14.264264000*

TABLE 4. $V(x) = -2\,\mathrm{sech}^2(x)$, domain chosen $[-5,5]$, spacial step size $h = 0.01$, energy step size $\eta = 0.001$, exact bound state used

Algorithm	c^2	*Relative Error*	*Time (sec)*
Exact	*2.000000000000000*	*0*	*0*
(i)	*2.004086813877857*	*0.002043406938928*	*3.460512000*
(ii)	*1.996830443518782*	*0.001584778240609*	*0.023956000*
(iii)	*1.999683571279579*	*0.000158214360211*	*1.631856000*
(iv)	*1.993946894122799*	*0.003026552938601*	*0.088449000*

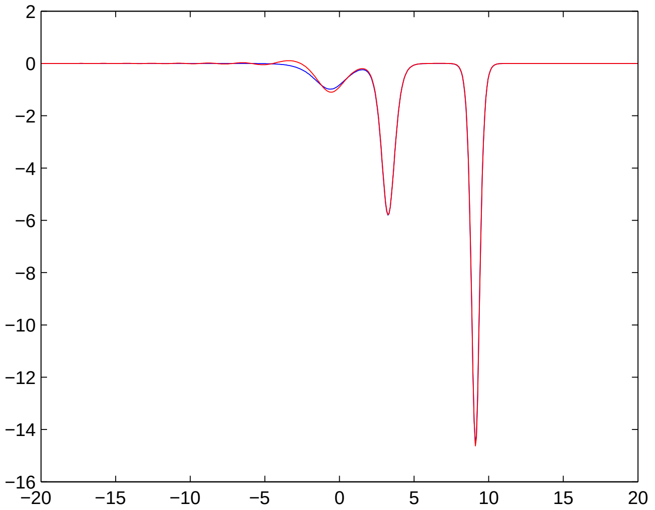


FIGURE 3. $V(x) = -10\,\mathrm{sech}^2(x)$, $t = 0.3$

8. HAAR SYSTEMS AND A KDV LARGE-TIME SOLVER

Suppose now that V is finite, nonpositive, and has compact support. Then V can be well approximated using finitely many nonpositive blocks. For such potentials V, we now summarize the algorithm for solving the KdV for large times:

- Approximate the potential $V(x)$ using N nonpositive blocks
- Bound states are found as zeros of $1/R_{1,\dots,N}$ with initial estimates, for example, derived from a spectral matrix estimate of the Schödinger operator
- The norming constants are calculated as residues of B_N at the bound states using the previously described recursive formulas

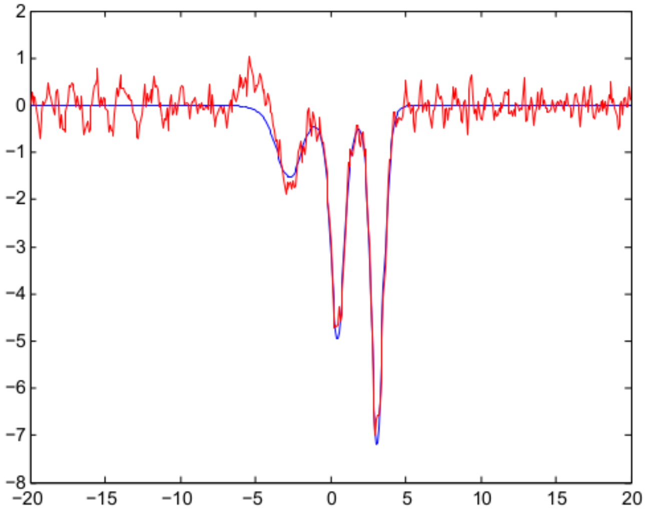


FIGURE 4. $V(x) = -4\chi_{[-4,0]}(x))$, $t = 0.3$

- The solution to the KdV is obtained from the formula (1.5).

There are a number of possible improvements to this algorithm. For example, the number of bound states is known for a single block, so the results in [4] could possibly be implemented to obtain an accurate estimate for the number of bound states for the potential. As another example, instead of piecewise-constant functions, one could instead use higher order spline interpolants of the potential. All of the recursive formulas in section 6 were derived from potential fragmentation, which holds for arbitrary potentials; the only things that would change would be the formula for R_n^0, the initial values in the recursive formulas, and the values for k in (6.20). For example, in the case of piecewise-linear spline interpolants, the formula for R_n^0 would involve the Airy functions.

Another possible route for improvement would be the use of Haar wavelets to obtain a step-like approximation. We will only consider Haar wavelets in the current paper. For a great exposition on Haar and other wavelets, see [20]. Consider the *scaling function*

$$\varphi(x) = \varphi_0(x) = \begin{cases} 1 & \text{if } 0 < x \le 1, \\ 0 & \text{otherwise,} \end{cases}$$

and the *mother wavelet*

$$w(x) = \begin{cases} 1 & \text{if } 0 < x \le 1/2, \\ -1 & \text{if } 1/2 < x \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

We form the Haar wavelets as follows: let

$$w_{j,0}(x) = w(2^j x).$$

Then $w_{j,0}$ has support $[0, 2^{-j}]$. Next, we translate $w_{j,0}$ so as to fill up the entire interval $[0,1]$ with 2^j subintervals of length 2^{-j}:

$$w_{j,k}(x) = \varphi_{2^j+k} = w_{j,0}(x-k) = w(2^j(x-k)), \quad k = 0, 1, \dots, 2^j - 1.$$

Then $w_{j,k}$ has support $[2^{-j}k, 2^{-j}(k+1)]$. The collection of *Haar wavelets*

$$\mathcal{H}_{2^n} = \{\varphi_m : 0 \le m \le 2^n - 1\}$$

forms an orthogonal system with respect to the L^2 norm of dimension 2^n; the collection $\mathcal{H}_\infty$ forms a complete orthogonal system for $L^2([0,1])$. For $\mathcal{H}_{2^n}$, let $\boldsymbol{\varphi}_r$ denote the vector in $\mathbb{R}^{2^n}$ corresponding to φ_r; i.e., the entries of $\boldsymbol{\varphi}_r$ are the function values of φ_r on the 2^n intervals.

By translating and scaling, suppose without loss of generality that V has compact support $[0,1]$. Since V is finite, we have that $V \in L^2([0,1])$, so V can be expressed in terms of the Haar basis:

$$V = \sum_{r=0}^{\infty} c_r \varphi_r$$

where

$$c_r = \frac{\langle V, \varphi_r \rangle_2}{\|\varphi_r\|_2}.$$

Let V_0 denote the piecewise-constant approximation of V on the 2^n intervals mentioned above, and let $\mathbf{V}$ denote the corresponding column vector in $\mathbb{R}^{2^n}$. Then V_0 can be represented as a linear combination of the Haar wavelets in $\mathcal{H}_{2^n}$:

$$V_0 = \sum_{r=0}^{2^n-1} c_r \varphi_r$$

where the coefficients c_r are as described above. Letting $\mathbf{c}$ denote the column vector of coefficients c_r, the *discrete wavelet transform* (DWT) is the map $H_{2^n} : \mathbf{V} \mapsto \mathbf{c}$; that is, H_{2^n} is a change of basis from the standard basis to the Haar basis. Letting W_{2^n} denote the matrix whose r-th column is $\boldsymbol{\varphi}_r$, we have that

$$\mathbf{V} = W_{2^n}\mathbf{c},$$

so

$$\mathbf{c} = W_{2^n}^{-1}\mathbf{V},$$

implying that $H_{2^n} = W_{2^n}^{-1}$. (Note: often, the columns are normalized so that W_{2^n} is an orthogonal matrix. In this case, $H_{2^n} = W_{2^n}^*$ where $*$ denotes the transpose).

The Discrete Wavelet Transform is analogous to the Fast Fourier Transform (FFT), which expresses $\mathbf{V}$ in the orthogonal basis corresponding to the Fourier basis $\{e^{i2^n x} : -2^{n-1} < r \le 2^n\}$ in $L^2([-\pi,\pi])$. However, the Fourier basis is not localized, unlike the Haar basis, so the Fourier basis has difficulty capturing data concentrated in a relatively small region. The Fourier basis tends to accurately approximate smoother functions, while exhibiting the so called *Gibbs phenomenon* at discontinuities. On the other hand, the Haar basis tends to accurately approximate discontinuous functions, while only slowly converging to smoother functions.

In the context of solving the KdV, Haar wavelets may possibly be implemented in a couple ways. One approach would be to approximate the potential using Haar wavelets since it generally gives more accurate piecewise-constant interpolants than, say the midpoint rule. Then the interpolating potential would be changed to the standard basis and used in our algorithm.

8.1. **Acknowledgements.** This work was done as part of the REU program run by the third author in the summer of 2009, and was supported by NSF grants DMS 0707476 and DMS 1009673. The government support is highly appreciated.

We are grateful to the other participants who have also contributed to this project: Lyman Gillispie and Sigourney Walker. We would particularly like to thank Paul Sacks for providing us algorithms (iii) and (iv) for calculating norming constants. We are also grateful to Constantine Khroulev and Anton Kulchitsky for useful consultations and discussions. And last but not least we would like to thank the anonymous referee for bringing our attention to [8, 18, 21], and numerous valuable comments.

References

[1] Ablowitz, Mark J.; Clarkson, Peter A.; *Solitons, nonlinear evolution equations and inverse scattering*, volume 149 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, Cambridge, UK.

[2] Aktosun, Tuncay; Klaus, Martin; Inverse theory: Problem on the line. pages 770–785. Chapter 2.2.4.

[3] Aktosun, Tuncay; Klaus, Martin; van der Mee, Cornelis; Factorization of scattering matrices due to partitioning of potentials in one-dimensional schrödinger-type equations. *J. Math. Phys.*, 37(12):5897–5915.

[4] Aktosun, Tuncay; Klaus, Martin; van der Mee, Cornelis On the number of bound states for the one-dimensional schrödinger equation. *J. Math. Phys.*, 39(9):4249–4256.

[5] Aktosun, Tuncay; Sacks, Paul E.; Potential splitting and numerical solution of the inverse scattering problem on the line. *Math. Methods Appl. Sci.*, 25(4):347–355.

[6] Aktosun, Tuncay; Bound states and inverse scattering for the Schrödinger equation in one dimension. *J. Math. Phys.*, 35(12):6231–6236.

[7] Aktosun, Tuncay. A factorization of the scattering matrix for the Schrödinger equation and for the wave equation in one dimension. *J. Math. Phys.*, 33(11):3865–3869.

[8] Aricò, Antonio; van der Mee, Cornelis; Seatzu, Sebastiano; Structured matrix solution of the nonlinear Schrödinger equation by the inverse scattering transform. *Electron. J. Diff. Eqns*, 2009(15):1–21.

[9] Belashov, Vasily Y.; Vladimirov, Sergey V.; *Solitary waves in dispersive complex media*, volume 149 of *Solid-State Sciences.* Springer, Springer-Verlag Berlin.

[10] Chanane, Bilal; Computation of the eigenvalues of Sturm-Liouville problems with parameter dependent boundary conditions using the regularized sampling method. *Math. Comp.*, 74(252):1793–1801.

[11] Deift, Percy; Trubowitz, Eugene; Inverse scattering on the line. *Comm. Pure Appl. Math.*, 32(2):121–251.

[12] Fack, Veerle; Vanden Berghe, Guido; (Extended) Numerov method for computing eigenvalues of specific Schrödinger equations. *J. Phys. A.*, 20(13):4153–4160.

[13] Grunert, Katrin; Teschl, Gerald; Long-time asymptotics for the Korteweg-de Vries equation via nonlinear steepest descent. *Math. Phys. Anal. Geom.*, 12(3):287–324.

[14] Holden, Helge; Karlsen, Kenneth H.; Risebro, Nils H.; Tao, Terence; Operator splitting for the kdv equation. *Math. Comp.*, 80(274):821–846.

[15] Kânoğlu, Utku; Synolakis, Costas E.; Long wave runup on piecewise linear topographies. *J. Fluid Mech.*, 374:1–28, 1998.

[16] Katatbeh, Qutaibeh D.; Spectral bisection algorithm for solving schrödinger equation using upper and lower solutions. *Electron. J. Diff. Eqns*, 2007(129):11 pp.

[17] Korsch, Hans J.; Laurent, Helmut; Milne's differential equation and numerical solutions of the schrödinger equation i. bound-state energies for single- and double-minimum potentials. *J. Phys. B: At. Mol. Phys.*, 14(22):4213–4230.

[18] van der Mee, Cornelis; Seatzu, Sebastiano;Theis, Daniela Structured matrix algorithms for inverse scattering on the line. *Calcolo*, 44(2):59–87, 2007.

[19] Munteanu, Ligia; Donescu, Stefania; *Introduction to soliton theory: applications to mechanics*, volume 143 of *Fundamental Theories of Physics*. Kluwer Academic Publishers, Dordrecht, Netherlands.

[20] Olver, Peter J.; Chapter 13: Fourier analysis. Lecture notes and book preprints, available at http://www.math.umn.edu/ olver/am_/fa.pdf

[21] Parratt, Lyman G.; Surface studies of solids by total reflection of X-rays. *Phys. Rev.*, 95(2):359–369, 1954.

[22] Pryce, John D.; *Numerical solution of Sturm-Liouville problems*. Oxford Science Publications. Oxford University Press, Oxford, UK.

[23] Sylvester, John; Winebrenner, Dale; Gylys-Colwell, Fred; Layer stripping for the Helmholtz equation. *J. Appl. Math.*, 56(3):736–754.

[24] Trefethen, Lloyd N. *Spectral Methods in MATLAB*. SIAM, Philadelphia, PA.

Jason Baggett
Department of Mathematics and Statistics, University of Alaska Fairbanks, PO Box 756660, Fairbanks, AK 99775, USA
E-mail address: jabaggett@alaska.edu

Odile Bastille
Department of Mathematics and Statistics, University of Alaska Fairbanks, PO Box 756660, Fairbanks, AK 99775, USA
E-mail address: orbastille@alaska.edu

Alexei Rybkin
Department of Mathematics and Statistics, University of Alaska Fairbanks, PO Box 756660, Fairbanks, AK 99775, USA
E-mail address: arybkin@alaska.edu