An accurate O(N ^{2}) floating point algorithm for the Crum transform of the KdV equation

Prins, Peter J.; Wahls, Sander

DOI
10.1016/j.cnsns.2021.105782

Publication date
2021

Document Version
Final published version

Published in
Communications in Nonlinear Science and Numerical Simulation

Commun Nonlinear Sci Numer Simulat 102 (2021) 105782

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns

Research paper

An accurate $\mathcal{O}(N^2)$ floating point algorithm for the Crum transform of the KdV equation

Peter J. Prins*, Sander Wahls

Delft Center for Systems and Control (DCSC), Delft University of Technology, Postbus 5, 2600 AA Delft, the Netherlands

ARTICLE INFO

Article history:
Received 5 October 2020
Revised 5 February 2021
Accepted 22 February 2021
Available online 26 February 2021

Keywords:
Darboux transform
Dressing method
Korteweg–de Vries equation (KdV)
Non-linear Fourier transform (NFT)

ABSTRACT

We present an algorithm to compute the N-fold Crum transform (also known as the dressing method) for the Korteweg–de Vries equation (KdV) accurately in floating point arithmetic. This transform can be used to generate solutions of the KdV equation, e.g. as a part of the inverse Non-linear Fourier Transform. Crum transform algorithms that sequentially add the N eigenvalues to the solution with a chain of N Darboux transforms have a computational complexity of $\mathcal{O}(N^2)$, but suffer inevitably from singular intermediate results during the computation of certain regular Crum transforms. Algorithms that add all N eigenvalues at once do not have that flaw, but have a complexity of $\mathcal{O}(N^3)$ and are often even less accurate for other reasons. Our algorithm has a complexity of $\mathcal{O}(N^2)$. It makes use of a chain of 2-fold Crum transforms and, if N is odd, one Darboux transform. Hence, our algorithm adds two eigenvalues at a time instead of one whenever possible. We prove that with the right eigenvalue ordering, this avoids artificial singularities for all regular Crum transforms. Furthermore, we demonstrate that our algorithm is considerably more accurate in floating point arithmetic than benchmark algorithms found in the literature. At the same error tolerance, N can be three to seven times as high when using our algorithm instead of the best among the benchmark algorithms.

1. Introduction

The Korteweg–de Vries equation (KdV) is a partial differential equation (PDE) that describes certain weakly non-linear wave phenomena in one space dimension. Among its numerous applications are surface waves in shallow water [1–5]; blood pressure waves in arteries [6,7]; internal waves in the ocean, Rosby waves in the atmosphere, plasma waves, acoustic waves [8,9]; and electrical waves in non-linear transmission lines [10].

From a mathematical point of view, the KdV has gained fame as the first example of a non-linear partial differential equation that is *(Lax-)integrable* [11,12]. That is, there exist a direct and an inverse Non-linear Fourier Transform (NFT) (the latter is also known as Inverse Scattering Transform [13]) that one can use to solve initial value problems for the KdV. The merit of the conventional Fourier transform is well known in linear system theory: Instead of doing hard computations on an input signal directly, one transforms it to an alternative representation called a spectrum. Certain computations, such as

* Corresponding author.
E-mail addresses: p.j.prins@tudelft.nl (P.J. Prins), s.wahls@tudelft.nl (S. Wahls).

https://doi.org/10.1016/j.cnsns.2021.105782

the evolution of the input signal, become simpler in this spectral representation. Finally one finds the result from the inverse transform of the resulting spectrum. With the KdV-NFT, initial value problems for the KdV can be solved accordingly.

Unlike the conventional Fourier transform, the NFT has long remained an analytical tool for mathematicians. Due to the non-linear nature of the transform, even simply scaling up a signal can change the non-linear spectrum dramatically. The impact of numerical inaccuracies typically increases as well, making the transform more difficult to compute. Numerical algorithms for the computation of forward and inverse NFTs that are fast and accurate enough for engineering practice started to appear only recently. See, e.g., [14–18]. A major motivation for the development of these methods is that NFTs can be used as a signal processing tool for revealing potentially hidden components such as solitons that cannot be detected using conventional linear methods [2,3,19–22].

In this paper, we are concerned with the inverse KdV-NFT. As will be explained in Section 2.2, the KdV-NFT spectra we consider consist in general of two parts: a continuous spectrum and a discrete spectrum. The discrete spectrum consists in its turn of $N \geq 0$ eigenvalues. A well-known approach for the numerical computation of inverse NFTs is to first find the inverse transform of the (suitably pre-compensated) continuous spectrum, and then add the N eigenvalues of the discrete spectrum in a second stage. See, e.g., [23, Sect. 3], [24, Chap. xvii.3.2], or [25, Sect. 4.2]. The contribution of this paper is a numerical algorithm for this second stage, which is also known as the *Crum transform*.

From an analytic perspective, the Crum transform for the KdV can be carried out in closed-form, but its numerical implementations are notoriously inaccurate. We are not aware of a publication that systematically studies these errors for the KdV case. However, Chimmalgi [26, Chap. 5] studied a Crum transform with respect to the Non-linear Schrödinger Equation (NSE). He showed that the error in the result of that Crum transform increases exponentially as a function of the number of eigenvalues N. Also for the NSE, Gelash and Agafontsev [27] resorted to high-precision arithmetic with an accuracy of 100 decimal digits in order to obtain sufficient accuracy for their needs, a strategy that comes at a substantial computational cost. In this paper, we therefore instead aim to improve the numerical accuracy of the method itself. This will enable us to handle cases that normally would have required high-precision arithmetic, using conventional 64 bit floating point arithmetic (specifically, IEEE 754 double precision), which is directly supported by conventional central processing units (CPUs). Of course, our method could also be implemented using high-precision arithmetic. In that case, the number of digits required is expected to be lower than with the existing algorithms, which again translates into a lower computational complexity.

A widely known and relatively simple and efficient way to implement any Crum transform is to add eigenvalues one by one using Darboux transforms [23]. The intermediate results generated by this approach however can have singularities in the KdV case, which then propagate into the final result and cause serious numerical problems. It is important to note that these intermediate singularities are an artificial by-product of using a chain of Darboux transforms. They can occur also when the mathematical conditions for the end result to be non-singular are fulfilled. (Further details will be provided later in Section 3.2.) A non-sequential algorithm for the KdV Crum transform that is free of artificial singularities was presented in [28]. Unfortunately, this algorithm suffers from large numerical errors for all but the most simple cases when floating point arithmetic is used. Furthermore, its complexity order is cubic instead of quadratic. In [16, Sect. IV/App. F], the authors of the present paper proposed some modifications to the algorithm of [28] that improved it just enough for the needs of that paper. However, when we recently tried to compute the inverse KdV-NFT of the discrete spectrum of surface waves measured in shallow water [22], we found that even this version was unable to reconstruct the free surface data with reasonable precision. To the best of our knowledge, these are the only (numerical) algorithms for the KdV Crum transform that have been reported in the literature so far.

In this paper we propose a completely redesigned algorithm for the KdV Crum transform. We prove (see Theorem 2) that it does not suffer from artificial singularities, in contrast to a chain of N Darboux transforms. Furthermore, it uses mathematically equivalent but numerically advantageous formulas to avoid several other major sources of numerical error. In numerical examples, *our algorithm can therefore process between three to seven times as many eigenvalues in floating point arithmetic as existing numerical algorithms (at comparable error levels). At the same time, the computational complexity of our algorithm is only quadratic.* We remark again that the proposed algorithm is not restricted to 64 bit floating point arithmetic. It could be implemented in higher precision arithmetic as well, to combine the accuracy gain of both approaches. Please be informed that the Crum transform for the KdV generates only solutions to the KdV and other partial differential equations (PDEs) that are Lax-integrable via the Schrödinger equation,[1] as well as for the Schrödinger equation itself. The same thus applies to the algorithm in this paper, although some of the underlying ideas can be carried over.

We structured the remainder of this paper as follows. In Section 2 we introduce the notation and outline the theory behind the NFT and the Crum transform as far as it is required to understand the rest of the paper. In Section 3, we discuss several sources of numerical error that will be avoided by our new algorithm. The new algorithm itself is then presented (in a fairly self-contained manner) in Section 4. The advantages of the new algorithm are illustrated with numerical examples in Section 5. The paper is concluded in Section 6. Lengthy proofs and derivations of some of the results presented in this paper are put in appendices, to improve the readability of the main text.

[1] Despite the similarity in name, the Non-linear Schrödinger Equation is *not* Lax-integrable via the Schrödinger equation.

2. Preliminaries

2.1. Notation

We use a slanted serif font for variables, upright serif for constants, sans-serif for operators and a calligraphic font for sets. Vectors and indexed vector elements are displayed in bold lower case and matrices in bold uppercase. Vertical bars, $|\cdot|$, indicate the absolute value of a scalar, the determinant of a matrix, or the cardinality of a set. We denote the nearest lower integer (floor) by $\lfloor\cdot\rfloor$, the nearest higher integer (ceil) by $\lceil\cdot\rceil$, and the nearest integer (round) by by $\lfloor\cdot\rceil$. We will use both $\exp(x)$ and e^x to denote the exponential function. The signum function is defined as $\operatorname{sign}(x) := 1$ for $x \geq 0$ and $\operatorname{sign}(x) := -1$ for $x < 0$. The binary logarithm is denoted by $\operatorname{lb}(x) := \log_2(x)$. The symbols '$\propto$' and '$\leftarrow$' are used to indicate 'is proportional to' and 'assign the right hand side to the left hand side', respectively. The Landau 'big-O' order symbol is written as $\mathcal{O}$. The sets of real, imaginary, and complex numbers are denoted by $\mathbb{R}$, $\mathbb{I}$, and $\mathbb{C}$, respectively. We furthermore use the short hand $\mathbb{N}_a^b := \{a, a+1, \ldots, b\}$. If needed, we use a superscript between round brackets to indicate to which potential a variable belongs.

2.2. Non-linear Fourier transform

The purpose we have in mind for our KdV Crum transform algorithm, is the KdV-NFT. In this section we outline briefly the KdV-NFT in order to establish the connection with the Crum transform hereafter in Section 2.3. For a comprehensive introduction to the NFT we refer to [29] and the references therein.

We consider the KdV $\frac{\partial}{\partial t}q = -3\frac{\partial}{\partial x}q^2 - \frac{\partial^3}{\partial x^3}q$, where t denotes time, x denotes location, and $q = q(x,t)$. We require that the initial condition at time t_0 is real and satisfies the vanishing boundary conditions

$$\lim_{|x|\to\infty} q(x,t_0) = 0 \qquad \text{and} \qquad \int_{-\infty}^{\infty} |\,q(x,t_0)\,|\,(1+|x|)\,\mathrm{d}x < \infty. \tag{1}$$

To find $q(x,t)$ for any t, we first compute the NFT spectrum of the initial condition [16,30]. Then we propagate this spectrum forward or backward in time by means of simple formulas [11], [31, Sect. 1.4]. Finally, the inverse NFT of the propagated spectrum gives $q(x,t)$.

For the forward KdV-NFT, one considers the initial condition $q(x,t_0)$ as the potential in the Schrödinger eigenvalue problem:[2,3]

$$\left(\tfrac{\partial^2}{\partial x^2} + q(x,t_0)\right) f(x,\zeta,t_0) = (\mathrm{j}\zeta)^2 f(x,\zeta,t_0), \qquad \text{where} \qquad \mathrm{j} := \sqrt{-1}. \tag{2}$$

We call signals $f(x,\zeta,t_0)$ that fulfil (2) *trajectories*. The *Jost solutions* $\phi(x,\pm\zeta,t_0)$ and $\psi(x,\pm\zeta,t_0)$ are the specific trajectories that satisfy the respective boundary conditions

$$\phi(x,\pm\zeta,t_0) \to \mathrm{e}^{\mp\mathrm{j}\zeta x} \text{ as } x \to -\infty; \qquad \psi(x,\pm\zeta,t_0) \to \mathrm{e}^{\pm\mathrm{j}\zeta x} \text{ as } x \to \infty. \tag{3}$$

Because the trajectories $\psi(x,\zeta,t_0)$ and $\psi(x,-\zeta,t_0)$ are linearly independent, one can find parameters $a(\zeta)$ and $b(\zeta,t_0)$ such that[4]

$$\phi(x,\zeta,t_0) \equiv a(\zeta)\,\psi(x,-\zeta,t_0) + b(\zeta,t_0)\,\psi(x,\zeta,t_0). \tag{4}$$

From these parameters one finds the KdV-NFT spectrum $(R(\zeta,t_0), \mathcal{D}(t_0))$, which consists of two parts. The continuous spectrum is defined by the so-called reflection coefficient

$$R(\zeta,t_0) := b(\zeta,t_0)/a(\zeta) \in \mathbb{C} \qquad \forall\zeta \in \mathbb{R}\setminus\{0\}. \tag{5}$$

The discrete spectrum is defined by the (possibly empty) set

$$\mathcal{D}(t_0) := \{(\gamma_m, b(\mathrm{j}\gamma_m,t_0)) \in \mathbb{R}_{>0}\times\mathbb{R}\setminus\{0\} \mid a(\mathrm{j}\gamma_m) = 0\}. \tag{6}$$

To simplify the notation, we omit the dependence on time in the remainder of this paper. The values $\mathrm{j}\gamma_m$ are known as eigenvalues and $b(\mathrm{j}\gamma_m)$ as norming constants. The eigenvalues are thus the values of ζ for which (4) reduces to

$$\phi(x,\mathrm{j}\gamma_m) \equiv b(\mathrm{j}\gamma_m)\,\psi(x,\mathrm{j}\gamma_m). \tag{7}$$

It can be shown that all eigenvalues lie on the upper half of the imaginary axis, i.e. $\gamma_m > 0$ for all $m \in \mathbb{N}_1^M = \{1,2,\ldots,M\}$, where M is the cardinality of the discrete spectrum. Furthermore, all eigenvalues are known to be simple [33, pp. 50–53]. Throughout this paper we consider the discrete spectrum as a totally ordered set with $0 < \gamma_1 < \gamma_2 < \cdots < \gamma_M$. Note that some other sources instead consider the values $(\mathrm{j}\gamma_m)^2$ as the eigenvalues. Since these values lie on the negative real axis,

[2] Eq. (2) establishes a close connection between the Non-linear Fourier Transform and quantum scattering, cf. [24].

[3] Trajectories of (2) are to be understood as solutions in a weak sence [32, pp. 1–2], so henceforward equations involving trajectories of (2) only need to hold almost everywhere.

[4] See e.g. [16, App. A] for the explicit definitions of $a(\zeta)$ and $b(\zeta,t_0)$.

the terms 'highest' and 'lowest' are prone to confusion. Therefore, we will refer to their magnitude by saying that $\mathrm{j}\gamma_M$ is the largest eigenvalue and $\mathrm{j}\gamma_1$ the smallest.

The inverse KdV-NFT recovers the potential $q(x)$ from a given spectrum $(R(\zeta), \mathcal{D})$. The three main approaches for its computation are *a)* reformulating the Schrödinger Eq. (2) as a Gel'fand–Levitan–Marchenko integral equation, *b)* solving an associated Riemann–Hilbert problem, and *c)* transferring the Schrödinger equation into the time domain and exploiting causality principles. See e.g. [24, Chap. xvii] for an exhaustive survey and [25,34,35] for some recently proposed numerical methods.

As already mentioned in Section 1, we consider the approach in which one first finds the inverse transform of the (suitably pre-compensated) continuous spectrum and subsequently adds the eigenvalues by means of a Crum transform. In this paper, we develop an accurate algorithm for the computation of the Crum transform that avoids artificial singularities and takes the effects of floating point arithmetic into account. It can either serve as the second stage of a general inverse KdV-NFT algorithm, or as a stand-alone inverse KdV-NFT algorithm for reflectionless potentials, potentials with a reflection coefficient of zero.

2.3. Crum transform

In this subsection, we briefly review the Crum transform. Detailed derivations can be found e.g. in [23, Sect. 3], [36], or [37, Chap. 1].

Recall that the *Wronskian* of N sufficiently often differentiable functions $g_1(x), \ldots, g_N(x)$ at x is given by

$$W[g_1(x); \ldots; g_N(x)] := \begin{vmatrix} g_1(x) & \frac{\mathrm{d}}{\mathrm{d}x} g_1(x) & \cdots & \left(\frac{\mathrm{d}}{\mathrm{d}x}\right)^{N-1} g_1(x) \\ \vdots & \vdots & & \vdots \\ g_N(x) & \frac{\mathrm{d}}{\mathrm{d}x} g_N(x) & \cdots & \left(\frac{\mathrm{d}}{\mathrm{d}x}\right)^{N-1} g_N(x) \end{vmatrix}. \tag{8}$$

Crum [38] used Wronskians to construct new solutions of the Schrödinger Eq. (2) by updating old ones. Since the Schrödinger equation determines the KdV-NFT spectrum, his method can be used to generate solutions of the KdV.

Theorem 1 (Crum transform)**.** *[23, Thm. 6], [36, Sect. 2.1] Let $f(x,\zeta) = f^{(0)}(x,\zeta)$ be any trajectory of (2) for a potential $q(x) = q^{(0)}(x)$ that satisfies (1). Let $f(x,\zeta_n) = \vartheta_n(x)$ for $n \in \mathbb{N}_1^N = \{1, 2, \ldots, N\}$ be N specific trajectories of (2) for $q(x) = q^{(0)}(x)$ and $\zeta = \zeta_n$. Then*

$$f^{(N)}(x,\zeta) = \mathsf{C}_{(0)}^{(N)} f^{(0)}(x,\zeta) := \frac{W[\vartheta_1(x); \vartheta_2(x); \ldots; \vartheta_N(x); f^{(0)}(x,\zeta)]}{W[\vartheta_1(x); \vartheta_2(x); \ldots; \vartheta_N(x)]} \tag{9}$$

is a trajectory of (2) for the potential

$$q^{(N)}(x) = q^{(0)}(x) + 2\frac{\mathrm{d}^2}{\mathrm{d}x^2} \ln\left(W[\vartheta_1(x); \vartheta_2(x); \ldots; \vartheta_N(x)]\right). \tag{10}$$

We call $q^{(0)}(x)$ the *background potential*, $q^{(N)}(x)$ the *target potential*, and $\vartheta_n(x)$ the *seed trajectories*. The Crum transform for $N = 1$ is also known as the *Darboux transform* and it can be shown that every Crum transform is analytically equivalent to a chain of N Darboux transforms [36, Sect. 2.1]. The Crum transform can thus be implemented in one shot, adding all N eigenvalues at once (e.g. [36, Sect. 2.1], [23, Thm. 6]); or sequentially, by repeated application of the Darboux transform (e.g., [37, Chap. 1], [23, Thm. 2], [24, Chap. xvii.3.2], [25, Sect. 4.2]). (In Section 3.1, we will propose a novel sequential approach for the numerical implementation of the Crum transform.)

To establish the connection with the KdV-NFT, we need two properties of the Crum transform. First, any Wronskian, (8), with a repeated entry (i.e., $g_n(x) \equiv g_m(x)$ for $n \neq m$) is the determinant of a singular matrix, which equals zero. Therefore, (9) implies that the Crum transform maps all of the seed trajectories to zero, i.e.

$$\mathsf{C}_{(0)}^{(N)} \vartheta_n(x) \equiv 0 \qquad \forall n \in \mathbb{N}_1^N. \tag{11}$$

Second, $\mathsf{C}_{(0)}^{(N)}$ is a *linear* differential operator by the well known fact that the determinant of a matrix is a multi-linear function of its rows. These two properties allow us to use the Crum transform for adding N eigenvalues to the discrete KdV-NFT spectrum of the background potential,[5] as follows.

Let us choose the seed trajectories to be of the form

$$\vartheta_n(x) = \phi^{(0)}(x, \mathrm{j}\kappa_n) - (-1)^N \beta_n \psi^{(0)}(x, \mathrm{j}\kappa_n), \qquad \beta_n \neq 0, \qquad \kappa_n > 0, \tag{12}$$

[5] The Crum transform can also be used to remove eigenvalues [23, Thm. 3], but in this paper we only discuss the use of the Crum transform to add eigenvalues.

where β_n and κ_n are real and finite, $\mathrm{j}\kappa_n$ is not an eigenvalue of the background potential, and $\phi^{(0)}(x,\mathrm{j}\kappa_n)$ and $\psi^{(0)}(x,\mathrm{j}\kappa_n)$ are Jost solutions of the background potential. Then the Jost solutions of the target potential turn out to be [23, Thm. 6], [36, Eq. (2.2.10)]

$$\phi^{(N)}(x,\zeta) = \Big(\prod_{n=1}^{N} \tfrac{1}{\kappa_n - \mathrm{j}\zeta}\Big) \mathsf{C}_{(0)}^{(N)}\, \phi^{(0)}(x,\zeta), \qquad \mathrm{j}\zeta \notin \{\kappa_1, \kappa_2, \dots, \kappa_n\}; \tag{13}$$

$$\psi^{(N)}(x,\zeta) = \Big(\prod_{n=1}^{N} \tfrac{-1}{\kappa_n - \mathrm{j}\zeta}\Big) \mathsf{C}_{(0)}^{(N)}\, \psi^{(0)}(x,\zeta), \qquad \mathrm{j}\zeta \notin \{\kappa_1, \kappa_2, \dots, \kappa_n\}. \tag{14}$$

Using (11), (12), (13), and (14) and the linearity of $\mathsf{C}_{(0)}^{(N)}$, we find

$$0 \equiv \mathsf{C}_{(0)}^{(N)} \vartheta_n(x) \equiv \mathsf{C}_{(0)}^{(N)} \big(\phi^{(0)}(x,\mathrm{j}\kappa_n) - (-1)^N \beta_n \psi^{(0)}(x,\mathrm{j}\kappa_n)\big) \equiv \tag{15}$$
$$\big(\mathsf{C}_{(0)}^{(N)} \phi^{(0)}(x,\mathrm{j}\kappa_n)\big) - \beta_n \big((-1)^N \mathsf{C}_{(0)}^{(N)} \psi^{(0)}(x,\mathrm{j}\kappa_n)\big) \propto \phi^{(N)}(x,\mathrm{j}\kappa_n) - \beta_n \psi^{(N)}(x,\mathrm{j}\kappa_n)$$
$$\Rightarrow \quad \phi^{(N)}(x,\mathrm{j}\kappa_n) \equiv \beta_n \psi^{(N)}(x,\mathrm{j}\kappa_n), \tag{16}$$

where '$\propto$' denotes proportionality. By comparing (16) with (7) we recognise $\mathrm{j}\kappa_n$ as eigenvalues of the target potential, with norming constants $b^{(N)}(\mathrm{j}\kappa_n) = \beta_n$. If the background potential already has eigenvalues, these are preserved, but their norming constants change sign when an odd number of eigenvalues is added.[6] Thus, when we choose the seed trajectories as in (12), the effect of the Crum transform on the discrete spectrum (6) is

$$\Big\{\big(\gamma_m^{(N)}, b^{(N)}(\mathrm{j}\gamma_m^{(N)})\big)\Big\} = \Big\{\big(\gamma_m^{(0)}, (-1)^N b^{(0)}(\mathrm{j}\gamma_m^{(0)})\big)\Big\} \cup \Big\{(\kappa_n, \beta_n)\Big| n \in \mathbb{N}_1^N\Big\}, \tag{17}$$

where $\cup$ denotes the union of sets. The effect on the continuous spectrum (5) is[7]

$$R^{(N)}(\zeta) = \Big(\prod_{n=1}^{N} \tfrac{\kappa_n - \mathrm{j}\zeta}{\kappa_n + \mathrm{j}\zeta}\Big) R^{(0)}(\zeta). \tag{18}$$

2.4. Dressing method

For numerical computations the Schrödinger Eq. (2) is typically rewritten as a system of first order ordinary differential equations. Thereto one defines an operator

$$\mathbf{V}(x,\zeta) := \begin{bmatrix} v_{11}(x,\zeta) + v_{12}(x,\zeta)\, \frac{\partial}{\partial x} \\ v_{21}(x,\zeta) + v_{22}(x,\zeta)\, \frac{\partial}{\partial x} \end{bmatrix}, \qquad \text{where} \qquad \begin{vmatrix} v_{11}(x,\zeta) & v_{12}(x,\zeta) \\ v_{22}(x,\zeta) & v_{22}(x,\zeta) \end{vmatrix} \neq 0, \tag{19}$$

such that (2) can be rewritten as

$$\frac{\partial}{\partial x} \boldsymbol{f}(x,\zeta) = \boldsymbol{A}(x,\zeta)\, \boldsymbol{f}(x,\zeta), \qquad \text{where} \qquad \boldsymbol{f}(x,\zeta) := \mathbf{V}(x,\zeta)\, f(x,\zeta) \in \mathbb{C}^{2\times 1}. \tag{20}$$

The simplest choice for the operator $\mathbf{V}(x,\zeta)$ is $\mathbf{V}_{\mathrm{C}}(x,\zeta) := \begin{bmatrix} 1 & \frac{\partial}{\partial x} \end{bmatrix}^{\top}$.

Other common choices for the operator $\mathbf{V}(x,\zeta)$ have been discussed in [16, App. C], see also Appendix B.1.

We also can rewrite the Crum transform using vector-valued trajectories. This formulation of the Crum transform is known as the *dressing method* in the literature [39]. Let $\boldsymbol{f}(x,\zeta) := \mathbf{V}(x,\zeta)\, f(x,\zeta)$ and $\boldsymbol{\vartheta}_n(x) := \mathbf{V}(x,\zeta)\, \vartheta_n(x)$ denote the vector-valued versions of the trajectories in Theorem 1. The Crum transform can then be expressed as

$$q^{(N)}(x) = q^{(0)}(x) + \Delta q_{(0)}^{(N)}(x) \qquad \text{and} \qquad \boldsymbol{f}^{(N)}(x,\zeta) = \boldsymbol{C}_{(0)}^{(N)}(x,\zeta)\, \boldsymbol{f}^{(0)}(x,\zeta), \tag{21}$$

where $\Delta q_{(0)}^{(N)}(x)$ is the *potential update* and $\boldsymbol{C}_{(0)}^{(N)}(x,\zeta)$ is the *dressing matrix*. Their exact forms depend on the choice of the operator $\mathbf{V}(x,\zeta)$, but they always satisfy

$$\begin{array}{rccccccc} \Delta q: & \mathbb{R} & \times & \mathbb{R} & \times \bigotimes_{n=1}^{N} \mathbb{C}^2 & \times & \bigotimes_{n=1}^{N} \mathbb{R}_{>0} & \to & \mathbb{R} \\ & x, & & q^{(0)}(x), & \boldsymbol{\vartheta}_1(x), \dots, \boldsymbol{\vartheta}_N(x), & & \kappa_1, \dots, \kappa_N & \mapsto & \Delta q_{(0)}^{(N)}(x); \\ \boldsymbol{C}: & \mathbb{R} & \times & (\mathbb{R} \cup \mathbb{I}) \setminus \{0\} & \times \bigotimes_{n=1}^{N} \mathbb{C}^2 & \times & \bigotimes_{n=1}^{N} \mathbb{R}_{>0} & \to & \mathbb{C}^{2\times 2} \\ & x, & & \zeta, & \boldsymbol{\vartheta}_1(x), \dots, \boldsymbol{\vartheta}_N(x), & & \kappa_1, \dots, \kappa_N & \mapsto & \boldsymbol{C}_{(0)}^{(N)}(x,\zeta). \end{array}$$

This formalistic description conveys an important practical merit of the dressing method: $\Delta q_{(0)}^{(N)}(x)$ and $\boldsymbol{C}_{(0)}^{(N)}(x,\zeta)$ are *local functions*, which means that in order to compute the Crum transform at some point x, the background potential and vector-valued seed trajectories need to be known only at the same specific point. (The Wronskian representation in Theorem 1 is not local since it requires not only the seed trajectories, but also the first $N+1$ derivatives thereof.) The elements of $\boldsymbol{\vartheta}_n(x)$

[6] To see this, let $\mathsf{C}_{(0)}^{(N)}$ operate on both sides of (7) and then substitute (13) and (14).

[7] To see this, let $\mathsf{C}_{(0)}^{(N)}$ operate on both sides of (4), use the linearity of $\mathsf{C}_{(0)}^{(N)}$ and then substitute (13) and (14). See also [23, Thm. 6].

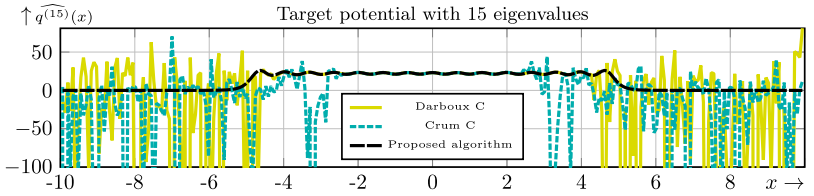

Fig. 1. Target potential with 15 eigenvalues, as sequentially computed with (21), (22), and (23) (——, Darboux C), by the dressing method for the same operator for $N = 15$ (- - - -, Crum C), and by the proposed algorithm (— —) respectively, all in 64 bit floating point arithmetic. The graph shows the numerical result of $q^{(15)}(x)$ in the domain $|x| < 10$ for the case $\nu = 15$ of the benchmark test described in Section 5.3. We remark that the graphs for the Darboux C (minimum value: -5.1×10^5) and Crum C algorithm (minimum value: -5.0×10^4) are severely truncated from below.

are independent linear combinations of $\vartheta_n(x)$ and $\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_n(x)$ by (19). In the dressing method higher order derivatives can also be replaced by certain scalar multiples of $\vartheta_n(x)$ or $\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_n(x)$ using the fact that every seed trajectory satisfies the Schrödinger Eq. (2) [23, p. 177–178]. (See also Appendix B.2.) Therefore, the computation of $\Delta q^{(N)}_{(0)}(x)$ and $\boldsymbol{C}^{(N)}_{(0)}(x,\zeta)$ can be carried out using only local, algebraic computations once the vector-valued seed trajectories $\boldsymbol{\vartheta}_n(x)$ are known. We emphasize that the dressing method is nevertheless an exact reformulation of the Crum transform.

As an example, the dressing method formulation of the Darboux transform (i.e. the $N = 1$ case of the Crum transform) with respect to the operator $\mathbf{V}_{\mathrm{C}}(x,\zeta)$ is given by

$$\Delta q^{(1)}_{(0)}(x) = -2\,q^{(0)}(x) + 2\frac{\left(\kappa_1\,\vartheta_1(x) - \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\right)\left(\kappa_1\,\vartheta_1(x) + \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\right)}{\vartheta_1^2(x)} \quad \text{and} \tag{22}$$

$$\boldsymbol{C}^{(1)}_{\mathrm{C}(0)}(x,\zeta) = \begin{bmatrix} -\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \Big/ \vartheta_1(x) & 1 \\ (\mathrm{j}\zeta)^2 - \frac{\left(\kappa_1\,\vartheta_1(x) - \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\right)\left(\kappa_1\,\vartheta_1(x) + \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\right)}{\vartheta_1^2(x)} & -\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \Big/ \vartheta_1(x) \end{bmatrix}, \tag{23}$$

where $\vartheta_1(x) = \begin{bmatrix}1 & 0\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{C}}(x)$ and $\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} = \begin{bmatrix}0 & 1\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{C}}(x)$. The derivation of (22) and (23) is shown in Appendix B.3.1.

3. Sources of numerical error and their mitigation

In practice, numerical implementations of the Crum transform suffer from surprisingly large errors, even for relatively simple cases. We show an illustrative example in Fig. 1, where 15 eigenvalues were added to a zero background potential. The first two lines in Fig. 1 were obtained with the dressing method with respect to the operator $\mathbf{V}_{\mathrm{C}}(x,\zeta)$. The difference between those two is that the solid line (——) was obtained by adding the eigenvalues one after another using (21), (22), (23), whereas the short dashed line (- - - -) was obtained by adding all 15 eigenvalues in a single shot. The long dashed line (— —) finally was obtained with the novel algorithm that will be proposed in Section 4 of this paper. All computations were carried out in 64 bit floating point arithmetic. The first two algorithms suffer from large numerical errors that can be observed in the form of rapid fluctuations. Our proposed algorithm in contrast finds the target potential with negligible numerical error.

In this section, we prepare the stage for our proposed algorithm by outlining the sources of such numerical errors and discussing mitigation strategies. The dressing method formulation of the Darboux transform in (22) and (23) serves as an example for our considerations, which also hold for many other implementations of the Crum transform. Our new algorithm is then presented in Section 4.

3.1. Sequential versus direct implementation

A Crum transform for $N \geq 2$ can be decomposed into a chain of lower order Crum transforms that is analytically equivalent. Such a decomposition is not unique, so there are typically many different ways to compute a Crum transform. The two extreme cases are

1. adding all N eigenvalues at once with a single Crum transform, and
2. adding eigenvalues sequentially with a chain of N Darboux transforms.

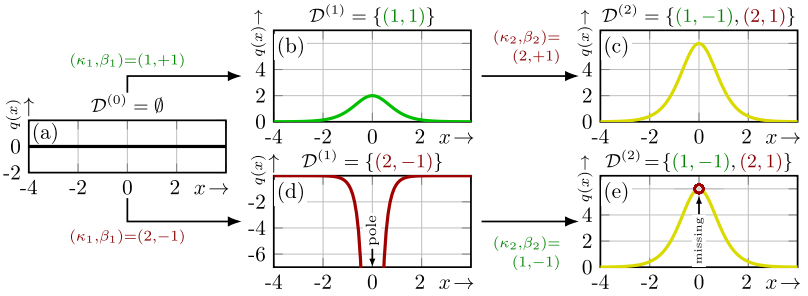

Fig. 2. Calculation of the target potential with spectrum $R(\zeta) \equiv 0$, $\mathcal{D}^{(2)} = \{(1,-1),(2,1)\}$ from a zero background potential by adding the eigenvalues in different orders with Darboux transforms. 1) Smallest first: From (a) $q^{(0)}(x) \equiv 0$, to (b) $q^{(1)}(x) = 2\,\mathrm{sech}^2(x)$, to (c) $q^{(2)}(x) = 6\,\mathrm{sech}^2(x)$. 2) Largest first: From (a) $q^{(0)}(x) \equiv 0$, to (d) $q^{(1)}(x) = -8\,\mathrm{csch}^2(2x)$, to (e) $q^{(2)}(x) = 6\,\mathrm{sech}^2(x)$ if $x \neq 0$, but undefined for $x = 0$ (○) due to the pole in (d).

One might intuitively expect that adding eigenvalues sequentially accumulates more numerical error than adding them all at once. However, Fig. 1 shows a counterexample that is typical for the dressing method in our experience. Furthermore, the computational complexity of adding N eigenvalues at once is at least in the order of $\mathcal{O}(N^3)$ floating point operation, because for each fixed x one either needs to solve a square linear system of size N, or calculate Nth order determinants. On the other hand, the computational complexity of adding N eigenvalues sequentially is only in the order of $\mathcal{O}(N^2)$ floating point operation (e.g., [24, p. 348]). Finally, it is hard to mitigate some of the error sources that are outlined in the remainder of this section when adding all eigenvalues at once (without fixing N). Since adding all eigenvalues at once is both slower and more prone to numerical error, we focus on sequential approaches in the following.

3.2. Poles in an intermediate potential

The Crum transform of a background potential that satisfies the vanishing boundary conditions (1) is in general not guaranteed to satisfy the same conditions again. In particular, the target potential may have one or more poles that cause divergence of the integral in (1). This happens if the Wronskian of the seed trajectories in (9) and (10) vanishes at one or more points x cf. [37, Sect. 1.1.2], [40, Sect. 5.3].

From a physical point of view, we are usually only interested in solutions of the KdV without poles. However, one issue with sequential implementations of the Crum transform is that they can introduce poles in intermediate potentials. These poles cause artificial singularities in the target potential that would not be present if all eigenvalues were added at once. It turns out that whether a sequential approach introduces artificial singularities or not depends on the order in which the eigenvalues are added. This is demonstrated in Fig. 2. There we construct the same target potential with two eigenvalues by adding them one by one in different orders. When the smaller eigenvalue is added first, both the intermediate and target potential satisfy (1). However, when the larger eigenvalue is added first, the intermediate potential has a pole (cf. [31, Eq. (3.4.10)], [40, Sect. 5.3]) that causes an artificial singularity in the otherwise equal target potential. Apart from an undetermined value at the artificial singularity this may lead to cusps at nearby x grid points caused by *catastrophic cancellation*: the loss of significance when two nearly equal numbers are subtracted in floating point arithmetic [41, Chap. 1.7].

To avoid artificial singularities, we need to know how to recognise the spectrum of an absolutely integrable potential, which is fortuitously simple. If a potential is absolutely integrable, then it is known from Sturm-Liouville oscillation theory that the eigenfunctions of (2) are continuous and have $M - m$ simple zeros, where M is the amount of eigenvalues and $m \in \mathbb{N}_1^M$ is the ordinal number (index) of each eigenvalue when sorted from small to large [42, Thm. 10.12.1-(4)], [43, Sect. 1]. Hence, each eigenfunction changes sign $M - m$ times for $x \in \mathbb{R}$. Because $\phi(x,\zeta) > 0$ as $x \to -\infty$ by (3) and $\phi(x,\zeta)$ changes sign $M - m$ times before approaching $+\infty$, where $\psi(x,\zeta) > 0$ by (3), (7) implies

$$\operatorname{sign}(b(\mathrm{j}\gamma_m)) = (-1)^{M-m} \qquad \forall m \in \mathbb{N}_1^M, \qquad \text{where} \qquad \gamma_1 < \gamma_2 < \ldots < \gamma_M. \tag{24}$$

If (24) is not satisfied, the potential is not absolutely integrable.

We will call a specific Crum transform *regular* if both the background and target potential are absolutely integrable. That is, if both of their spectra satisfy (24). When we use the Crum transform to add an odd number of eigenvalues, the norming constants of the background potential all change sign according to (17). Therefore a Crum transform is regular if and only if the background potential is absolutely integrable and the following rules are obeyed.

Rule 1. When adding one eigenvalue, it must be larger than all eigenvalues of the background potential [44, p. 269], [45, p. 1388]. Its norming constant must be positive in the target spectrum, cf. [46, Sect. 6.6], [45, p. 1389].

Rule 2. When adding two eigenvalues simultaneously, no eigenvalue of the background potential may lie between them. The sign of their norming constants must be such that the target spectrum satisfies (24), cf. [47].

Rule 3. When removing one or more eigenvalues, those eigenvalues must satisfy [48, Thm.]. (No additional condition on the norming constants is needed here, because these satisfy (24) automatically in this case.)

Rule 4. When adding three or more eigenvalues simultaneously, it must be equivalent to a chain of the previous three rules.

Not every regular N-fold Crum transform can be decomposed into a chain of N regular Darboux transforms. By Rule 2, a regular Crum transform can add a pair of eigenvalues that are smaller than some of the eigenvalues of the background potential without introducing artificial singularities. However, this Crum transform cannot be decomposed into two regular Darboux transforms by Rule 1. This means that an N-fold Crum transform that is numerically implemented as a chain of N Darboux transforms cannot avoid artificial singularities in the computation of certain regular Crum transforms.

The algorithm that we will propose in Section 4 decomposes every regular Crum transform into a specific chain of Crum transforms that each add either one or simultaneously two eigenvalues. This specific chain contains provably only regular Crum transform steps. That is, artificial singularities never occur.

3.3. Catastrophic cancellation for large magnitude of x

Remarkably, most implementations of the Crum transform show numerical difficulties first in the vanishing 'tails' of the target potential, whereas the potential changes mostly in the centre, cf. Fig. 1. The first cause of that effect (out of two) is the following. In the computation of the potential update $\Delta q^{(N)}_{(0)}(x)$ and the dressing matrix $\mathbf{C}^{(N)}_{\mathrm{C}(0)}(x,\zeta)$ (e.g. (22) and (23)) one encounters factors of the form $\left(\kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x)\right)$. When the seed trajectories are chosen as in (12), one can verify with (3) that $\lim_{x\to\pm\infty} \kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x) = 0$, whereas $\liminf_{|x|\to\infty} |\vartheta_n(x)| \exp(-\kappa_n|x|) > 0$ for all $n \in \mathbb{N}_1^N$, i.e. $\vartheta_n(x)$ grows exponentially as $|x| \to \infty$. Therefore, calculating the factors $\left(\kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x)\right)$ from $\boldsymbol{\vartheta}_{n\mathrm{C}}(x)$ will result in catastrophic cancellation when $|x|$ is large enough. We propose to avoid this problem by choosing a suitable operator $\mathbf{V}(x,\zeta)$. From (19) it follows that

$$\kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x) = \tag{25}$$
$$\begin{vmatrix} v_{11}(x,\zeta) & v_{12}(x,\zeta) \\ v_{21}(x,\zeta) & v_{22}(x,\zeta) \end{vmatrix}^{-1} \left[-\mathrm{j}\zeta\, v_{22}(x,\zeta) \mp v_{21}(x,\zeta) \quad \mathrm{j}\zeta\, v_{12}(x,\zeta) \pm v_{11}(x,\zeta)\right] \boldsymbol{\vartheta}_n(x)\Big|_{\zeta=\mathrm{j}\kappa_n}.$$

If the operator $\mathbf{V}(x,\zeta)$ satisfies $v_{11}(x,\zeta) \equiv -\mathrm{j}\zeta\, v_{12}(x,\zeta)$ and $v_{21}(x,\zeta) \equiv \mathrm{j}\zeta\, v_{22}(x,\zeta)$, we can calculate these factors without summation and thus avoid catastrophic cancellation.

The second cause for catastrophic cancellation in the 'tails' occurs when a Crum transform is implemented by iterating Darboux transforms. In that case we can see from (22) that every potential update consists of twice subtracting the background potential before adding a term that depends on the seed trajectories. This results in catastrophic cancellation at points where the potential update is small compared to the background potential. Because the 'tails' of the potential update are dominated by the lowest eigenvalue being added,[8] one should add larger eigenvalues before smaller ones in order to mitigate this error. However, the rules in Section 3.2 say that smaller eigenvalues must be added before larger ones when using Darboux transforms only. In our algorithm, which will be presented in the next section, we will therefore implement the Crum transform as a chain of mainly 2-fold Crum transforms, and one Darboux transform in case N is odd. That allows us to add larger eigenvalues before smaller ones.

3.4. Seed trajectories growing exponentially as x goes to infinity

The seed trajectories satisfy $\liminf_{|x|\to\infty} |\vartheta_n(x)| \exp(-\kappa_n|x|) > 0$ for all $n \in \mathbb{N}_1^N$ according to (12), as was already mentioned in Section 3.3. Therefore, they tend to grow larger than the largest representable number in IEEE 754 double precision floating point numbers (`realmax` $\approx \mathrm{e}^{710}$ [41, Sect. 2.1]) already for moderate values of $\kappa_n|x|$. This effect causes numerical problems in naive implementations of the Crum transform. To avoid these problems, the operator $\mathbf{V}(x,\zeta)$ in our new algorithm (that will be proposed in Section 4) is chosen such that the elements of $\boldsymbol{\vartheta}_n(x)$ are bounded even when $\vartheta_n(x)$ is not. Specifically, also in view of Section 3.3, we choose[9]

$$\mathbf{V}_{\mathrm{E}}(x,\zeta) := (2\mathrm{j}\zeta)^{-1}\left[\exp(\mathrm{j}\zeta x)\left(\mathrm{j}\zeta - \tfrac{\partial}{\partial x}\right) \quad \exp(-\mathrm{j}\zeta x)\left(\mathrm{j}\zeta + \tfrac{\partial}{\partial x}\right)\right]^{\top}. \tag{26}$$

In Appendix B.1 we relate $\mathbf{V}_{\mathrm{E}}(x,\zeta)$ to other operators one encounters in the literature.

3.5. Seed trajectories remaining exponentially large or small

The operator in (26) ensures that the representation of the seed trajectories is bounded, but the range of this representation may still be problematic in floating point arithmetic. For example, if the support of (the significant part of) a potential

[8] From (3), (10), and (12) one can verify that $q^{(N)}(x) - q^{(0)}(x) \propto \exp(\mp 2\kappa_1 x)$ as $x \to \pm\infty$.

[9] As an alternative solution, one could represent the seed trajectories by their logarithm, but that would necessitate cumbersome log arithmetic in every part of the Crum transform.

is far from $x = 0$, the norming constants $b(\mathrm{j}\gamma_m)$ are either very small or very large because norming constants change exponentially under a translation in x of the potential [16, App. B]. In that case we can see from (3), (12), and (26) that the magnitude of elements of $\boldsymbol{\vartheta}_{n\mathrm{E}}(x)$ can be both very small or very large, at least when $|x|$ is large. For the dressing method we typically need to calculate products of $2N$ of these elements. Even for moderate N, the magnitudes of these products can become larger than `realmax` or smaller than `realmin`.[10]

We can mitigate this problem by scaling the vector-valued seed trajectories according to $\boldsymbol{\vartheta}_n(x) \leftarrow h_n(x)\,\boldsymbol{\vartheta}_n(x)$ before computing the potential update and the dressing matrix, where $h_n(x)$ is a suitable scaling function that is non-zero for all x. It is important to note that this a priori scaling step does not require any compensating change in the computation of the potential update $\Delta q^{(N)}_{(0)}(x)$ and dressing matrix $\boldsymbol{C}^{(N)}_{(0)}(x,\zeta)$. To see this, first let η_n be a non-zero constant and replace $\vartheta_n(x)$ by $\eta_n\,\vartheta_n(x)$ in Theorem 1. By the multi-linearity of Wronskians and the well-known fact that $\ln(ab) = \ln(a) + \ln(b)$ (if $ab \neq 0$), (9) and (10) are independent of η_n. Since the dressing method is an exact reformulation of the Crum transform, the same must hold for (21), where the equivalent replacement is $\boldsymbol{\vartheta}_n(x) \leftarrow \mathbf{V}(x,\zeta)\,(\eta_n\,\vartheta_n(x)) = \eta_n\,\boldsymbol{\vartheta}_n(x)$. By the locality of the dressing method, the factor $h_n(x)$ can be treated as a constant, just like η_n. The scaling of $\boldsymbol{\vartheta}_n(x)$ by $h_n(x)$ therefore does not require any change in the computation of $\Delta q^{(N)}_{(0)}(x)$ and $\boldsymbol{C}^{(N)}_{(0)}(x,\zeta)$.

To centre the trajectory within the range of representable floating point numbers without adding unnecessary rounding errors, we will choose the scaling function

$$h_n(x) = 2^{-\lfloor \frac{1}{2}\mathrm{lb}|[1\quad 0]\,\boldsymbol{\vartheta}_n(x)| + \frac{1}{2}\mathrm{lb}|[0\quad 1]\,\boldsymbol{\vartheta}_n(x)|\rceil}, \qquad \text{where} \qquad \mathrm{lb}(a) = \log_2(a), \qquad \lfloor\cdot\rceil = \mathrm{round}(\cdot). \tag{27}$$

3.6. Explicit ratios of exponentials in the computation

The computation of the dressing matrix $\boldsymbol{C}^{(N)}_{\mathrm{E}(0)}(x,\zeta)$ and the potential update $\Delta q^{(N)}_{(0)}(x)$ involves some explicit exponentials $\exp(\pm\kappa_n x)$ that can make intermediate results exceed `realmax`. For example, the values on the diagonal of (23) could be calculated from $\boldsymbol{\vartheta}_{1\mathrm{E}}(x)$ as

$$-\frac{\mathrm{d}}{\mathrm{d}x}\frac{\vartheta_1(x)}{\vartheta_1(x)} = \frac{\kappa_1\begin{bmatrix}-\exp(\kappa_1 x) & \exp(-\kappa_1 x)\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}{\begin{bmatrix}\exp(\kappa_1 x) & \exp(-\kappa_1 x)\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}, \tag{28}$$

where we see potentially troublesome exponentials enter the computation. We propose to solve this issue by using the analytically equivalent expressions for numerical computations that are obtained after dividing the numerator and denominator by the dominant exponential for negative and positive x respectively:

$$-\frac{\mathrm{d}}{\mathrm{d}x}\frac{\vartheta_1(x)}{\vartheta_1(x)} = \begin{cases} \dfrac{\kappa_1\begin{bmatrix}-\exp(2\kappa_1 x) & 1\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}{\begin{bmatrix}\exp(2\kappa_1 x) & 1\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{E}}(x)} & x \leq 0, \\ \dfrac{\kappa_1\begin{bmatrix}-1 & \exp(-2\kappa_1 x)\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}{\begin{bmatrix}1 & \exp(-2\kappa_1 x)\end{bmatrix}\boldsymbol{\vartheta}_{1\mathrm{E}}(x)} & x > 0. \end{cases} \tag{29}$$

3.7. Division by vanishing seed trajectories

The most convenient expressions for the potential update $\Delta q^{(N)}_{(0)}(x)$ and dressing matrices $\boldsymbol{C}^{(N)}_{(0)}(x,\zeta)$ often contain terms of the form $\frac{\mathrm{d}\,\vartheta_n(x)}{\mathrm{d}x} \big/ \vartheta_n(x)$.[11] See e.g. (B.38) and (B.45). If there are no zero-crossings in the scalar-valued seed trajectories $\vartheta_n(x)$, they are safe to use since no division by zero can take place. For a regular Darboux transform ($N = 1$), this assumption can be made safely because the Wronskian of the seed trajectories in (10) does not vanish, and for $N = 1$ this Wronskian is the seed trajectory itself. However, for a regular Crum transform with $N \geq 2$, individual seed trajectories may vanish even though their Wronskian does not. When the nth seed trajectory vanishes, $\frac{\mathrm{d}\,\vartheta_n(x)}{\mathrm{d}x} \big/ \vartheta_n(x)$ becomes singular. Hence for numerical computations we must write the Crum transform such that the only trajectory-dependent factor appearing in a denominator is the Wronskian of the seed trajectories.

3.8. Clipping for reflectionless target potentials

The set of all target potentials that can be written as the Crum transform of a zero background potential are called reflectionless potentials in literature. It is known that reflectionless potentials are non-negative. However, in practice negative samples may appear due to finite precision effects. By setting negative samples to zero for reflectionless target potentials, a closer approximation of the true target is achieved. In our proposed algorithm we apply this strategy only if the background potential is zero, because in that case it is simple to determine that the target potential is reflectionless.

[10] In IEEE 754 double precision floating point numbers $\texttt{realmin} \approx \mathrm{e}^{-708}$ is the smallest representable number above zero [41, Sect. 2.1].

[11] They are convenient in symbolic manipulations because the number of scalar-valued variables on which $\Delta q^{(N)}_{(0)}(x)$ and $\boldsymbol{C}^{(N)}_{(0)}(x,\zeta)$ depend is reduced from $3N+2$ to $2N+2$.

4. Proposed algorithm

In this section, we present our new algorithm for computing Crum transforms for the Korteweg–de Vries equation. The algorithm is based on the error analysis from Section 3.

4.1. 1- and 2-fold Crum transforms

We start with numerically advantageous expressions for the addition of one or two eigenvalues. Let us represent the Schrödinger equation and the seed trajectories as in (20) using the operator $\mathbf{V}_{\mathrm{E}}$ defined in (26). In Appendix B.4, it is shown that the N-fold Crum transform with $N \in \{1, 2\}$ takes the form

$$q^{(N)}(x) = (-1)^N q^{(0)}(x) + \begin{bmatrix} 4 & 0 \end{bmatrix} \boldsymbol{M}^{(N)}_{N-2}(x, 0) \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \tag{30}$$

$$\boldsymbol{f}^{(N)}_{\mathrm{E}}(x, \zeta) = \boldsymbol{C}^{(N)}_{\mathrm{E}(0)} \boldsymbol{f}^{(0)}_{\mathrm{E}}(x, \zeta), \quad \text{where} \quad \boldsymbol{C}^{(N)}_{\mathrm{E}(0)} = \sum_{m=-1}^{N} (\mathrm{j}\zeta)^m \boldsymbol{M}^{(N)}_m(x, \zeta). \tag{31}$$

In order to evaluate (30) and (31), we require expressions for the matrices $\boldsymbol{M}^{(N)}_m(x, \zeta)$. For the 1-fold Crum transform (Darboux transform), the matrices are needed for $m \in \{-1, 0, 1\}$. Let κ_1 denote the eigenvalue to be added. Furthermore, denote the components of the corresponding vector-valued seed trajectory by $\boldsymbol{\vartheta}_{1\mathrm{E}1} := \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}}(x)$ and $\boldsymbol{\vartheta}_{1\mathrm{E}2} := \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}}(x)$. We propose to calculate the matrices as

$$\boldsymbol{M}^{(1)}_{-1}(x, \zeta) = \begin{cases} \dfrac{2\kappa_1^2 \boldsymbol{\vartheta}_{1\mathrm{E}1} \boldsymbol{\vartheta}_{1\mathrm{E}2}}{w_-^2(\vartheta_1)} \begin{bmatrix} \mathrm{e}^{2\kappa_1 x} & \mathrm{e}^{2(\mathrm{j}\zeta + \kappa_1)x} \\ -\mathrm{e}^{-2(\mathrm{j}\zeta - \kappa_1)x} & -\mathrm{e}^{2\kappa_1 x} \end{bmatrix} & x \le 0, \\ \dfrac{2\kappa_1^2 \boldsymbol{\vartheta}_{1\mathrm{E}1} \boldsymbol{\vartheta}_{1\mathrm{E}2}}{w_+^2(\vartheta_1)} \begin{bmatrix} \mathrm{e}^{-2\kappa_1 x} & \mathrm{e}^{2(\mathrm{j}\zeta - \kappa_1)x} \\ -\mathrm{e}^{-2(\mathrm{j}\zeta + \kappa_1)x} & -\mathrm{e}^{-2\kappa_1 x} \end{bmatrix} & x > 0; \end{cases} \tag{32}$$

$$\boldsymbol{M}^{(1)}_{0}(x, \zeta) = \begin{cases} \dfrac{\kappa_1 \left(\boldsymbol{\vartheta}_{1\mathrm{E}2} - \boldsymbol{\vartheta}_{1\mathrm{E}1} \mathrm{e}^{2\kappa_1 x} \right)}{w_-(\vartheta_1)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x \le 0, \\ \dfrac{\kappa_1 \left(\boldsymbol{\vartheta}_{1\mathrm{E}2} \mathrm{e}^{-2\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}1} \right)}{w_+(\vartheta_1)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x > 0; \end{cases} \tag{33}$$

$$\boldsymbol{M}^{(1)}_{1}(x, \zeta) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \text{where} \tag{34}$$

$$w_\pm(\vartheta_1) := W[\vartheta_1(x)] \mathrm{e}^{\mp \kappa_1 x} = \boldsymbol{\vartheta}_{1\mathrm{E}1} \mathrm{e}^{(1 \mp 1)\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2} \mathrm{e}^{-(1 \pm 1)\kappa_1 x}. \tag{35}$$

The derivation of these expressions can be found in Appendix B.4.1.

For the 2-fold Crum transform, we require expressions for the matrices $\boldsymbol{M}^{(2)}_m(x, \zeta)$ with $m \in \{-1, 0, 1, 2\}$ in order to evaluate (30) and (31). These matrices depend on κ_1, $\boldsymbol{\vartheta}_{1\mathrm{E}1}$, and $\boldsymbol{\vartheta}_{1\mathrm{E}2}$, as well as κ_2 (the second eigenvalue to be added), $\boldsymbol{\vartheta}_{2\mathrm{E}1} := \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\vartheta}_{2\mathrm{E}}(x)$, and $\boldsymbol{\vartheta}_{2\mathrm{E}2} := \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{\vartheta}_{2\mathrm{E}}(x)$. We propose to calculate these matrices as follows.

$$\boldsymbol{M}^{(2)}_{-1}(x, \zeta) = \begin{cases} \dfrac{2\kappa_1 \kappa_2 (\kappa_2^2 - \kappa_1^2)}{w_-^2(\vartheta_1, \vartheta_2)} \begin{bmatrix} m_-(x, 0) & m_-(x, \zeta) \\ -m_-(x, -\zeta) & -m_-(x, 0) \end{bmatrix} & x \le 0, \\ \dfrac{2\kappa_1 \kappa_2 (\kappa_2^2 - \kappa_1^2)}{w_+^2(\vartheta_1, \vartheta_2)} \begin{bmatrix} m_+(x, 0) & m_+(x, \zeta) \\ -m_+(x, -\zeta) & -m_+(x, 0) \end{bmatrix} & x > 0; \end{cases} \tag{36}$$

$$\boldsymbol{M}^{(2)}_{0}(x, \zeta) = \begin{cases} \dfrac{2(\kappa_2^2 - \kappa_1^2)}{w_-^2(\vartheta_1, \vartheta_2)} \begin{bmatrix} p_-(x) & s_-(x, \zeta) \\ s_-(x, -\zeta) & p_-(x) \end{bmatrix} - \dfrac{\kappa_2^2 + \kappa_1^2}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x \le 0, \\ \dfrac{2(\kappa_2^2 - \kappa_1^2)}{w_+^2(\vartheta_1, \vartheta_2)} \begin{bmatrix} p_+(x) & s_+(x, \zeta) \\ s_+(x, -\zeta) & p_+(x) \end{bmatrix} - \dfrac{\kappa_2^2 + \kappa_1^2}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x > 0; \end{cases} \tag{37}$$

$$\boldsymbol{M}^{(2)}_{1}(x, \zeta) = \begin{cases} \dfrac{(\kappa_2^2 - \kappa_1^2)(\boldsymbol{\vartheta}_{1\mathrm{E}1} \mathrm{e}^{2\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2})(\boldsymbol{\vartheta}_{2\mathrm{E}1} \mathrm{e}^{2\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2})}{w_-(\vartheta_1, \vartheta_2)} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} & x \le 0, \\ \dfrac{(\kappa_2^2 - \kappa_1^2)(\boldsymbol{\vartheta}_{1\mathrm{E}1} + \boldsymbol{\vartheta}_{1\mathrm{E}2} \mathrm{e}^{-2\kappa_1 x})(\boldsymbol{\vartheta}_{2\mathrm{E}1} + \boldsymbol{\vartheta}_{2\mathrm{E}2} \mathrm{e}^{-2\kappa_2 x})}{w_+(\vartheta_1, \vartheta_2)} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} & x > 0; \end{cases} \tag{38}$$

$$\boldsymbol{M}^{(2)}_{2}(x, \zeta) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \text{where} \tag{39}$$

$$w_{\pm}(\vartheta_1,\vartheta_2) := \begin{vmatrix} \vartheta_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x} + \vartheta_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x} & \kappa_1\left(\vartheta_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x} - \vartheta_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x}\right) \\ \vartheta_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x} + \vartheta_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x} & \kappa_2\left(\vartheta_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x} - \vartheta_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x}\right) \end{vmatrix}, \tag{40}$$

$$m_{\pm}(x,\zeta) := \begin{vmatrix} \vartheta_{1\mathrm{E}1}^2\mathrm{e}^{(2\mp2)\kappa_1 x} - \vartheta_{1\mathrm{E}2}^2\mathrm{e}^{-(2\pm2)\kappa_1 x} & \kappa_1\vartheta_{1\mathrm{E}1}\vartheta_{1\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_1)x} \\ \vartheta_{2\mathrm{E}1}^2\mathrm{e}^{(2\mp2)\kappa_2 x} - \vartheta_{2\mathrm{E}2}^2\mathrm{e}^{-(2\pm2)\kappa_2 x} & \kappa_2\vartheta_{2\mathrm{E}1}\vartheta_{2\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_2)x} \end{vmatrix}, \tag{41}$$

$$p_{\pm}(x) := \frac{1}{4}\left(\kappa_2^2-\kappa_1^2\right)\left(\vartheta_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x} + \vartheta_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x}\right)^2\left(\vartheta_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x} + \vartheta_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x}\right)^2, \tag{42}$$

$$s_{\pm}(x,\zeta) := \begin{vmatrix} \kappa_1^2\vartheta_{1\mathrm{E}1}\vartheta_{1\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_1)x} & \left(\vartheta_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x} + \vartheta_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x}\right)^2 \\ \kappa_2^2\vartheta_{2\mathrm{E}1}\vartheta_{2\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_2)x} & \left(\vartheta_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x} + \vartheta_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x}\right)^2 \end{vmatrix}. \tag{43}$$

The derivations of these expressions can be found in Appendix B.4.2.

Referring back to Section 3.6, we emphasize that for numerical reasons the analytically equivalent expressions for the cases $x \le 0$ and $x > 0$ should not be merged.

4.2. General N-fold Crum transform

For $N > 2$ eigenvalues, we propose to proceed as follows. First, order the eigenvalues from large to small. If N is odd, add the largest eigenvalue using the 1-fold Crum transform formula given above. Then, add the remaining eigenvalues in pairs by repeated application of the 2-fold Crum formula from above. The complete procedure is shown in Algorithm 1. The

Algorithm 1 Proposed algorithm for the N-fold Crum transform. A tilde distinguishes the variables for the current iteration from those for the overall Crum transform. The conditions in lines 3 and 4 state that $\boldsymbol{\phi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n)$ and $\boldsymbol{\psi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n)$ are the Jost solutions of (20) for the operator $\mathbf{V}_{\mathrm{E}}$, cf. (B.14).

1: **inputs:** $q^{(0)}(x)$, N, $\left\{\kappa_n,\ b^{(N)}(\mathrm{j}\kappa_n),\ \boldsymbol{\phi}_{\mathrm{E}}^{(0)}(\mathrm{x},\mathrm{j}\kappa_n),\ \boldsymbol{\psi}_{\mathrm{E}}^{(0)}(\mathrm{x},\mathrm{j}\kappa_n)\right\}\ \forall n \in \mathbb{N}_1^N$
2: **require:** $0 < \kappa_1 < \kappa_2 < \cdots < \kappa_N$ and none of $\mathrm{j}\kappa_n$ is an eigenvalue of $q^{(0)}(x)$
3: **require:** $\frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{\phi}_{\mathrm{E}}^{(0)} \equiv \frac{q^{(0)}}{2\kappa_n}\begin{bmatrix} -1 & -\exp(-2\kappa_n x) \\ \exp(2\kappa_n x) & 1 \end{bmatrix}\boldsymbol{\phi}_{\mathrm{E}}^{(0)}$, with $\lim_{x\to-\infty}\boldsymbol{\phi}_{\mathrm{E}}^{(0)} = \begin{bmatrix}1\\0\end{bmatrix}\ \forall\kappa_n$
4: **require:** $\frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{\psi}_{\mathrm{E}}^{(0)} \equiv \frac{q^{(0)}}{2\kappa_n}\begin{bmatrix} -1 & -\exp(-2\kappa_n x) \\ \exp(2\kappa_n x) & 1 \end{bmatrix}\boldsymbol{\psi}_{\mathrm{E}}^{(0)}$, with $\lim_{x\to\infty}\boldsymbol{\psi}_{\mathrm{E}}^{(0)} = \begin{bmatrix}0\\1\end{bmatrix}\ \forall\kappa_n$
5: **for all** x **do** {
6: $\quad q(x) \leftarrow q^{(0)}(x)$;
7: $\quad$ **for all** $n = 1$ **to** N **do** $\boldsymbol{\vartheta}_{n\mathrm{E}}(x) \leftarrow \boldsymbol{\phi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n) - (-1)^N b^{(N)}(\mathrm{j}\kappa_n)\,\boldsymbol{\psi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n)$;
8: $\quad$ **while** $N > 0$ **do** {
9: $\quad\quad \widetilde{N} \leftarrow 1$; $\widetilde{\kappa}_1 \leftarrow \kappa_N$; $\widetilde{\boldsymbol{\vartheta}}_{1\mathrm{E}}(x) \leftarrow 2^{-\lfloor\frac{1}{2}\mathrm{lb}\lVert[1\quad 0]\boldsymbol{\vartheta}_{N\mathrm{E}}(x)\rVert+\frac{1}{2}\mathrm{lb}\lVert[0\quad 1]\boldsymbol{\vartheta}_{N\mathrm{E}}(x)\rVert\rceil}\boldsymbol{\vartheta}_{N\mathrm{E}}(x)$;
10: $\quad\quad$ **if** N is even **then** {
11: $\quad\quad\quad \widetilde{N} \leftarrow 2$; $\widetilde{\kappa}_2 \leftarrow \kappa_{(N-1)}$;
12: $\quad\quad\quad \widetilde{\boldsymbol{\vartheta}}_{\mathbf{2E}}(\mathbf{x}) \leftarrow 2^{-\lfloor\frac{1}{2}\mathrm{lb}\lVert[1\quad 0]\boldsymbol{\vartheta}_{(N-1)\mathrm{E}}(x)\rVert+\frac{1}{2}\mathrm{lb}\lVert[0\quad 1]\boldsymbol{\vartheta}_{(N-1)\mathrm{E}}(x)\rVert\rceil}\boldsymbol{\vartheta}_{(N-1)\mathrm{E}}(x)$; }
13: $\quad\quad q(x) \leftarrow (-1)^{\widetilde{N}}q(x) + [4\quad 0]\,\mathbf{M}_{\widetilde{N}-2}^{(\widetilde{N})}(x,0)\begin{bmatrix}0\\1\end{bmatrix}$;
14: $\quad\quad$ **if** $q^{(0)}(x) \equiv 0$ **then** $q(x) \leftarrow \max(q(x),0)$;
15: $\quad\quad N \leftarrow N - \widetilde{N}$;
16: $\quad\quad$ **for** $n = 1$ **to** N **do** $\boldsymbol{\vartheta}_{n\mathrm{E}}(x) \leftarrow \sum_{m=-1}^{\widetilde{N}}(-\kappa_n)^m\,\mathbf{M}_m^{(\widetilde{N})}(x,\mathrm{j}\kappa_n)\,\boldsymbol{\vartheta}_{n\mathrm{E}}(x)$; }
17: $\quad q^{(N)}(x) \leftarrow q(x)$; }
18: **output:** $q^{(N)}(x)$

matrices $\boldsymbol{M}_m^{(\widetilde{N})}(x,\zeta)$ in the algorithm should be calculated with the equations in Section 4.1 after dropping the tildes from $\widetilde{\kappa}_1$, $\widetilde{\kappa}_2$, $\widetilde{\boldsymbol{\vartheta}}_{1\mathrm{E}}(x)$, $\widetilde{\boldsymbol{\vartheta}}_{2\mathrm{E}}(x)$ and $\widetilde{N}$. Note that the inputs $b^{(N)}(\mathrm{j}\kappa_n)$ in Algorithm 1 are the desired values of the norming constants *of the target spectrum*. The algorithm flips their sign before processing if that is required according to (13), (14) and (17).

The specific ordering and pairing of the eigenvalues used by Algorithm 1 ensures that if the requested N-fold Crum transform is regular, then every iteration is regular. As a consequence, our algorithm will never introduce artificial singularities into the target potential. This important advantage is formalized in the following theorem.

Theorem 2. *If the background potential $q^{(0)}(x)$ and the target potential $q^{(N)}(x)$ are both non-singular, then all intermediate potentials $q(x)$ of Algorithm 1 are non-singular.*

Proof. By iterative application of Lemmas 1 and 2 in Appendix A. □

The conventional method of implementing the Crum transform as a chain of N Darboux transforms does not have this property. (See Section 3.2.)

The computational complexity of the proposed algorithm is $\mathcal{O}(N^2)$. To see this, note that the mapping of the remaining seed trajectories to the next potential in line 16 of Algorithm 1 dominates the complexity for large N. This mapping is executed $\big(2\lceil N/2\rceil - 2\big) + \big(2\lceil N/2\rceil - 4\big) + \ldots + 4 + 2 + 0 = \lceil N/2\rceil^2 - \lceil N/2\rceil = \mathcal{O}(N^2)$ times.

We emphasize that the proposed algorithm is analytically equivalent to Theorem 1. In infinite precision, both methods would provide exactly the same result.

5. Numerical examples

In this section, we investigate the numerical properties of our proposed algorithm in three different examples. For each example, we generate a series of discrete target spectra with increasing numbers of eigenvalues. We compute the corresponding potentials with the proposed algorithm (▬▬) as well as with three benchmark algorithms, and assess the accuracy of the results. In this way, we can compare the impact of finite precision effects on the algorithms, which are all numerical implementations of the N-fold Crum transform.[12] The first benchmark algorithm is the sequential application of the Darboux transform in (22) and (23), where each step adds the lowest of the remaining target eigenvalues. We will refer to this $\mathcal{O}(N^2)$ algorithm as Darboux C (▬). The second benchmark algorithm is the one-shot Crum transform presented in [28], with a complexity of $\mathcal{O}(N^3)$. We will call it Neugebauer vanilla (▬·▬). The third benchmark algorithm is called Neugebauer modified (▬·▬). It is a more accurate version of Neugebauer vanilla that was used/presented in [16, Sect. IV/App. F]. The complexity of Neugebauer modified is still $\mathcal{O}(N^3)$.

5.1. Error measures

The relative 2-norm error is a standard error measure to assess the accuracy of a numerically computed signal. In our case, it takes the form

$$\mathrm{E}_0 := \Big\| \widehat{q^{(N)}} - q^{(N)} \Big\|_2 \;\Big/\; \big\| q^{(N)} \big\|_2, \tag{44}$$

where $\widehat{q^{(N)}}(x)$ is the numerically computed target potential and $q^{(N)}(x)$ the true target potential. The number of test cases for the Crum transform for which the target potential is known analytically is unfortunately quite small. We will therefore resort to error measures that can be evaluated without knowing the target potential. Specifically, we consider the error measures

$$\mathrm{E}_p := \left| \frac{\int_{-\infty}^{\infty} \big(\widehat{q^{(N)}}(x)\big)^p - \big(q^{(0)}(x)\big)^p \,\mathrm{d}x}{\int_{-\infty}^{\infty} \big(q^{(N)}(x)\big)^p - \big(q^{(0)}(x)\big)^p \,\mathrm{d}x} - 1 \right|, \qquad p \in \{1, 2\}, \tag{45}$$

where $q^{(0)}(x)$ is the background potential. When the true target potential is unknown, we can make use of the so-called conserved quantities of the Korteweg–de Vries equation to evaluate the denominator in (45) directly from the spectrum. The first two of these conserved quantities are [31, Sect. 1.6], [50]

$$\int_{-\infty}^{\infty} q(x) \,\mathrm{d}x = \tfrac{1}{\pi} \int_{-\infty}^{\infty} \ln\big(1 - |\,R(\zeta)\,|^2\big) \,\mathrm{d}\zeta + 4 \sum_{m=1}^{M} \gamma_m; \tag{46}$$

$$\int_{-\infty}^{\infty} \big(q(x)\big)^2 \,\mathrm{d}x = -\tfrac{4}{\pi} \int_{-\infty}^{\infty} \zeta^2 \ln\big(1 - |\,R(\zeta)\,|^2\big) \,\mathrm{d}\zeta + \tfrac{16}{3} \sum_{m=1}^{M} \gamma_m^3. \tag{47}$$

Since the eigenvalues of the background potential are invariant under the Crum transform, as is $|\,R(\zeta)\,|$ for ζ on the real line by (18), it follows from (45), (46) and (47) that

$$\mathrm{E}_p = \left| \frac{(2p-1) \int_{-\infty}^{\infty} \big(\widehat{q^{(N)}}(x)\big)^p - \big(q^{(0)}(x)\big)^p \,\mathrm{d}x}{4^p \sum_{n=1}^{N} \kappa_n^{2p-1}} - 1 \right|, \qquad p \in \{1, 2\}. \tag{48}$$

This formula enables us to evaluate E_1 and E_2 even if the true target potential is unknown. In Section 5.2, we will demonstrate that E_1 and E_2 are highly correlated with E_0.

[12] The analytic sensitivity of the potential to perturbations of the spectrum is discussed elsewhere [49].

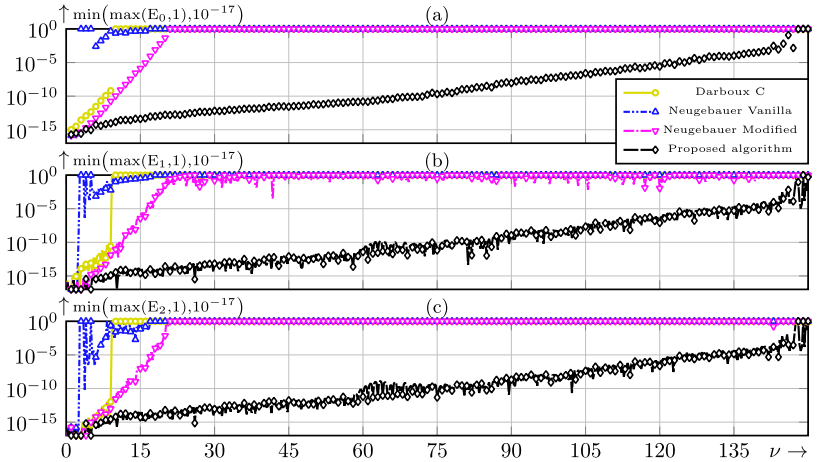

Fig. 3. Results example 'The discrete spectrum of a squared secant hyperbolic potential' (Section 5.2). The target potentials, consisting of $N = \lceil \nu \rceil$ eigenvalues, were computed with three benchmark algorithms and the proposed one. The errors E_1 and E_2 were calculated with (48) for $\nu \in \{\frac{1}{4}, \frac{2}{4}, \ldots, 150\}$ and are shown as lines (——/·····/-·-·-/——). Furthermore E_0, E_1, and E_2 were calculated with (44) and (45) for $\nu \in \mathbb{N}_1^{150}$ and are shown as markers (○/△/▽/◇).

5.2. The discrete spectrum of a squared secant hyperbolic potential

In this example, we start from a zero background potential. We add the $N = \lceil \nu \rceil$ eigenvalues $\big(\kappa_n, b^{(\lceil \nu \rceil)}(\mathrm{j}\kappa_n)\big) = \big(\nu - \lceil \nu \rceil + n, (-1)^{\lceil \nu \rceil - n}\big)$, where $n \in \mathbb{N}_1^{\lceil \nu \rceil}$ and ν is varied from $\frac{1}{4}$ to 150 in steps of $\frac{1}{4}$. For each ν and each of the 4 algorithms, we evaluate the errors E_1 and E_2 according to (48) using trapezoidal integration with x on a grid from -75 to 75 in steps of $\frac{1}{40}$. If any of the benchmark algorithms returns samples with the value Not a Number (NaN), we treat those as zero in this calculation.

It can be shown that if $\nu \in \mathbb{N}$, then the corresponding target potential is $q^{(\nu)}(x) = \nu(\nu+1)\operatorname{sech}^2(x)$ [33, Sect. 2.5]. In those cases, we additionally evaluate E_0 with (44) and evaluate E_1 and E_2 also using (45), again making use of trapezoidal integration. The results are shown in Fig. 3. We see that at any fixed error tolerance above 10^{-10}, the proposed algorithm allows us to compute a potential with roughly seven times as many eigenvalues as the best among the benchmark algorithms. Furthermore we see that the error calculation according to (48) is indeed equivalent to the error calculation according to (45), and that E_1 and E_2 are highly correlated with E_0.

5.3. The discrete spectrum of a rectangular potential

In this example, we again start from a zero background potential. We add the discrete spectrum of the rectangular potential that is given by $q_{\mathrm{rect}}(x) := \big(\frac{\pi\nu}{\ell}\big)^2$ for $|2x| < \ell = 10$ and $q_{\mathrm{rect}}(x) := 0$ otherwise. (We omit the continuous spectrum to demonstrate the proposed algorithm without needing an inverse transformation of the continuous spectrum.) By equating the reciprocal of [20, Eq. (8.11)] to zero, one finds with some work that the potential has $N = \lceil \nu \rceil$ eigenvalues $\mathrm{j}\kappa_n$. The κ_n are the solutions of

$$\frac{\sqrt{(\pi\nu)^2 - (\ell\kappa_n)^2}}{\ell\kappa_n} - \frac{\ell\kappa_n}{\sqrt{(\pi\nu)^2 - (\ell\kappa_n)^2}} = 2\cot\Big(\sqrt{(\pi\nu)^2 - (\ell\kappa_n)^2}\Big)$$
$$\text{for which } \pi\sqrt{\max\big(0,\ \nu^2 - (N-n+1)^2\big)} < \ell\kappa_n < \pi\sqrt{\nu^2 - (N-n)^2}. \tag{49}$$

Since $q_{\mathrm{rect}}(x)$ is even symmetric and non-singular, the corresponding norming constants are $b^{(\lceil \nu \rceil)}(\mathrm{j}\kappa_n) = (-1)^{\lceil \nu \rceil - n}$ by [16, Cor. 1] and (24). We vary ν from 0.5 to 50 in steps of 0.1 and evaluate the errors E_1 and E_2 according to (48)[13] for all four algorithms with trapezoidal integration on a grid from $x = -350$ to 350 in steps of 0.07. If any of the benchmark algorithms returns samples with the value NaN, we treat those as zero in this calculation. The results are shown in Fig. 4. We see that at any fixed error tolerance above 10^{-10}, the proposed algorithm allows us to compute a potential with three or more times as many eigenvalues as the best among the benchmark algorithms. We remark that the sawtooth pattern in the result of Neugebauer vanilla is caused by the problems that this algorithm in particular has with the computation of the 'tails' of the potential. The decay of these 'tails' is limited by the smallest eigenvalue. Therefore, the error peaks whenever ν is slightly larger than an integer.

[13] It is not meaningful for our purpose to compare $q^{(\lceil \nu \rceil)}(x)$ to $q_{\mathrm{rect}}(x)$ since we ignore the non-zero continuous spectrum of $q_{\mathrm{rect}}(x)$.

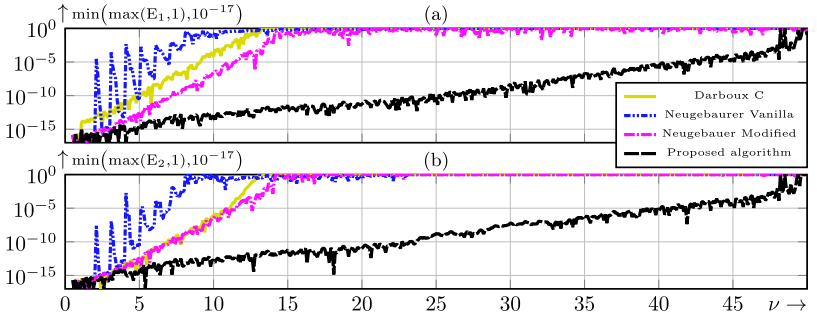

Fig. 4. Results example 'The discrete spectrum of a rectangular potential' (Section 5.3). The target potentials, consisting of $N = \lceil \nu \rceil$ eigenvalues, were computed with three benchmark algorithms (——/-·-·-/-··-··-) and the proposed one (▬▬). The errors E_1 and E_2 were calculated with (48) for $\nu \in \{0.5, 0.6, \ldots, 50\}$.

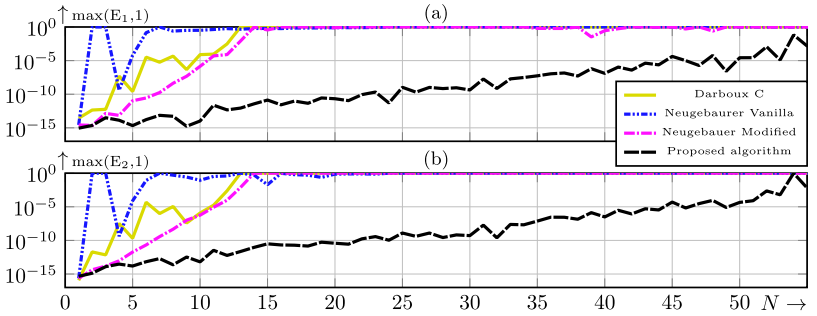

Fig. 5. Results of example 'Non-zero background potential' (Section 5.4). The target potentials, with $2 + N$ eigenvalues, were computed with three benchmark algorithms (——/-·-·-/-··-··-) and the proposed one (▬▬). The errors E_1 and E_2 were calculated with (48) for $N \in \mathbb{N}_1^{55}$.

5.4. Non-zero background potential

In this example, we choose $q^{(0)}(x) := 0.3\,\mathrm{sign}(x)$ for $-5 < x < 10$ and $q^{(0)}(x) := 0$ otherwise. This background potential has two eigenvalues, $\mathrm{j}\gamma_1 \approx \mathrm{j}0.292$ and $\mathrm{j}\gamma_2 \approx \mathrm{j}0.493$, that were computed numerically. (See, e.g., [14,51].) We want to add up to 55 eigenvalues in such a way that the target potential is non-singular for every $N \in \mathbb{N}_1^{55}$. However, the addition of any odd number of eigenvalues in the interval $(\mathrm{j}\gamma_1, \mathrm{j}\gamma_2)$ would result inevitably in a singular potential (by (17) and (24)). Instead, we chose to add $2\lfloor N/2 \rfloor$ eigenvalues in this interval. If N is odd, the remaining target eigenvalue must be larger than $\mathrm{j}\gamma_2$. We achieve this by choosing

$$\left\{ \kappa_n \middle| n \in \mathbb{N}_1^N \right\} = \left\{ \frac{K_2 + K_1}{2} + \frac{K_2 - K_1}{N} \left(p + \frac{1}{2} \right) \middle| p \in \mathbb{Z} \right\} \cap \left(K_1, K_2 \right], \tag{50}$$

where $\cap$ denotes set intersection, $\kappa_1 < \kappa_2 < \cdots < \kappa_N$, $K_1 = 0.292$, and $K_2 = 0.493 + 0.001$. We choose the norming constants as $b^{(N)}(\mathrm{j}\kappa_n) = -(-1)^{N-n} 10^{\pi_n - 4.5}$ for $n \in \mathbb{N}_1^{N-1}$ and $b^{(N)}(\mathrm{j}\kappa_N) = -(-1)^N 10^{\pi_n - 4.5}$, where π_n is the nth decimal digit of π ($\pi_1 = 3$, $\pi_2 = 1$, etc.). The purpose of using π_n is to introduce some pseudorandomness to the example that is simple to reproduce. Note that the signs of the norming constants are prescribed by the non-singularity condition (24).

The trapezoidal approximation of the integral in the error criterion (48) converged only very slowly in this example. Therefore, we instead approximated the integral in (48) with Clanshaw–Curtis quadrature [52, Sect. 2] in four intervals: $\mathcal{I}_1 = (-250, -5)$, $\mathcal{I}_2 = (-5, 0)$, $\mathcal{I}_3 = (0, 10)$, and $\mathcal{I}_4 = (10, 250)$. We chose $(2^9 + 1)$-point Chebychev grids in the closures of $\mathcal{I}_1$ and $\mathcal{I}_4$, and $(2^{13} + 1)$-point Chebychev grids in the closures of $\mathcal{I}_2$ and $\mathcal{I}_3$. At the endpoints the values were modified such that the potential was smooth in every interval.[14] If any of the benchmark algorithms returns samples with the value NaN, we treat those as zero for this calculation.

The results are shown in Fig. 5. A difference compared to the previous two examples is that the curves for the Darboux C benchmark algorithm are rather fitful. Only this benchmark algorithm suffers artificial singularities in the target potential, at different positions x for different N. However, an artificial singularity deteriorates the error curves of the Darboux C method only if it appears at or nearby a grid point, within the numerical cusp that surrounds the artificial singularity. (See Section 3.2.)

[14] Modifying the potential at a single point does not change the value of the integral.

We again observe that the proposed algorithm allows us to add at least three times as many eigenvalues as the best benchmark algorithm at any fixed error tolerance above 10^{-10}.

6. Conclusion

The Crum transform plays an important role in many inverse Non-linear Fourier Transform methods. In this paper, we have proposed a new algorithm for the numerical computation of the Crum transform for the Korteweg–de Vries equation. The numerical accuracy of the new algorithm is much better than that of previous algorithms since it carefully avoids several sources of numerical error. In particular, algorithms which add the eigenvalues one by one with Darboux transforms, suffer from artificial singularities during the calculation of certain regular Crum transforms. We have proven that the new algorithm on the other hand, computes every regular Crum transform without singularity (see Theorem 2).

The complexity of the new algorithm is quadratic in the number of eigenvalues, Therefore, it is as fast as the fastest benchmark algorithm. Nevertheless, compared to the most accurate benchmark algorithm. the new algorithm was able to process between three and seven times as many eigenvalues in numerical examples.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Peter J. Prins: Methodology, Software, Validation, Writing - original draft. **Sander Wahls:** Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 716669).

Appendix A. Proofs for the absolute integrability of intermediate potentials

In this appendix we prove two lemmas which together guarantee that the proposed algorithm never introduces artificial singularities.

Lemma 1. *If N is odd and both the background potential $q^{(0)}(x)$ and the target potential $q^{(N)}(x)$ are absolutely integrable, then the first intermediate potential, $q^{(1)}(x)$, obtained with Algorithm 1 is absolutely integrable.*

Proof. First, assume that the discrete spectrum of the background potential is non-empty. Since $q^{(N)}(x)$ and $q^{(0)}(x)$ are absolutely integrable,

$$(24) \Rightarrow \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma_m^{(N)})\right) = (-1)^{M+N-m} \quad \forall m \in \mathbb{N}_1^{M+N}, \tag{A.1}$$

$$(24) \Rightarrow \operatorname{sign}\left(b^{(0)}(\mathrm{j}\gamma_m^{(0)})\right) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^{M}, \tag{A.2}$$

$$(17), (A.2) \Rightarrow \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma_m^{(0)})\right) = (-1)^{M+N-m} \quad \forall m \in \mathbb{N}_1^{M}, \tag{A.3}$$

$$(A.1), (A.3) \Rightarrow \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma_{M+N}^{(N)})\right) \neq \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma_M^{(0)})\right), \tag{A.4}$$

$$(A.4) \Rightarrow \gamma_{M+N}^{(N)} \neq \gamma_M^{(0)} \Rightarrow \gamma_{M+N}^{(N)} = \kappa_N > \max_m \gamma_m^{(0)}. \tag{A.5}$$

Hence, $\mathrm{j}\kappa_N$ becomes the highest eigenvalue of $q^{(1)}(x)$:

$$(A.5) \Rightarrow \gamma_m^{(1)} = \begin{cases} \kappa_N & m = M+1, \\ \gamma_m^{(0)} & m \in \mathbb{N}_1^M \end{cases} \tag{A.6}$$

$$(A.6) \Rightarrow \operatorname{sign}\left(b^{(1)}(\mathrm{j}\gamma_m^{(1)})\right) = \begin{cases} \operatorname{sign}\left(b^{(1)}(\mathrm{j}\kappa_N)\right) & m = M+1, \\ \operatorname{sign}\left(b^{(1)}(\mathrm{j}\gamma_m^{(0)})\right) & m \in \mathbb{N}_1^M \end{cases} \tag{A.7}$$

$$(17),\ (A.5) \Rightarrow \qquad = \begin{cases} \operatorname{sign}\left(b^{(1)}(\mathrm{j}\gamma^{(N)}_{M+N})\right) & m = M+1, \\ -\operatorname{sign}\left(b^{(0)}(\mathrm{j}\gamma^{(0)}_{m})\right) & m \in \mathbb{N}_1^M \end{cases} \tag{A.8}$$

$$(17),\ (A.2) \Rightarrow \qquad = \begin{cases} \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma^{(N)}_{M+N})\right) & m = M+1, \\ -(-1)^{M-m} & m \in \mathbb{N}_1^M \end{cases} \tag{A.9}$$

$$(A.1) \Rightarrow \qquad = (-1)^{(M+1)-m} \quad \forall m \in \mathbb{N}_1^{M+1}. \tag{A.10}$$

From (24) and (A.10) it follows that $q^{(1)}(x)$ is absolutely integrable. If the discrete spectrum of the background potential is empty, the proof reduces to (A.6) to (A.10), where $\mathbb{N}_1^M = \mathbb{N}_1^0$ is an empty set. □

Lemma 2. *If N is even and both the background potential $q^{(0)}(x)$ and the target potential $q^{(N)}(x)$ are absolutely integrable, then the first intermediate potential, $q^{(2)}(x)$, obtained with Algorithm 1 is absolutely integrable.*

Proof. First, assume that the discrete spectrum of the background potential is non-empty. Since $q^{(N)}(x)$ and $q^{(0)}(x)$ are absolutely integrable,

$$(24),\ N \text{ even} \Rightarrow \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma^{(N)}_{m})\right) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^{M+N}, \tag{A.11}$$

$$(24) \Rightarrow \operatorname{sign}\left(b^{(0)}(\mathrm{j}\gamma^{(0)}_{m})\right) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^{M}, \tag{A.12}$$

$$(17),\ (A.12) \Rightarrow \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma^{(0)}_{m})\right) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^{M}, \tag{A.13}$$

$$(A.11),\ (A.13) \Rightarrow \begin{cases} \left|\{\kappa_n \mid \gamma^{(0)}_m < \kappa_n < \gamma^{(0)}_{m+1}\}\right| \mod 2 = 0 \forall m \in \mathbb{N}_1^{M-1}, \\ \left|\{\kappa_n \mid \kappa_n > \gamma^{(0)}_M\}\right| \mod 2 = 0. \end{cases} \tag{A.14}$$

Hence, $\mathrm{j}\kappa_N$ and $\mathrm{j}\kappa_{N-1}$ become successive eigenvalues of $q^{(2)}(x)$:

$$(A.14) \Rightarrow \gamma^{(N)}_m = \begin{cases} \gamma^{(0)}_{m-N} & m \in \mathbb{N}_{M+N-\mu+1}^{M+N}, \\ \kappa_{m+\mu-M} & m \in \mathbb{N}_{M+N-\mu-1}^{M+N-\mu}, \\ \star & m \in \mathbb{N}_1^{M+N-\mu-2}, \end{cases} \text{where } \star \text{ is not of interest and where } \mu := \left|\{\gamma^{(0)}_m \mid \gamma^{(0)}_m > \kappa_N\}\right| \in \mathbb{N}_1^M. \tag{A.15}$$

$$(A.11),\ (A.15) \Rightarrow \quad \operatorname{sign}\left(b^{(N)}(\mathrm{j}\kappa_{m+\mu-M})\right) = \operatorname{sign}\left(b^{(N)}(\mathrm{j}\gamma^{(N)}_{m})\right) = (-1)^{M-m}, \text{for } m \in \mathbb{N}_{M+N-\mu-1}^{M+N-\mu}; \tag{A.16}$$

$$(A.14) \Rightarrow \gamma^{(2)}_m = \begin{cases} \gamma^{(0)}_{m-2} & m \in \mathbb{N}_{M-\mu+3}^{M+2}, \\ \kappa_{m+\mu-M} & m \in \mathbb{N}_{M-\mu+1}^{M-\mu+2}, \\ \gamma^{(0)}_{m} & m \in \mathbb{N}_1^{M-\mu}; \end{cases} \tag{A.17}$$

$$(A.17) \Rightarrow \operatorname{sign}\left(b^{(2)}(\mathrm{j}\gamma^{(2)}_{m})\right) = \begin{cases} \operatorname{sign}\left(b^{(2)}(\gamma^{(0)}_{m-2})\right) & m \in \mathbb{N}_{M-\mu+3}^{M+2}, \\ \operatorname{sign}\left(b^{(2)}(\kappa_{m+\mu-M})\right) & m \in \mathbb{N}_{M-\mu+1}^{M-\mu+2}, \\ \operatorname{sign}\left(b^{(2)}(\gamma^{(0)}_{m})\right) & m \in \mathbb{N}_1^{M-\mu}; \end{cases} \tag{A.18}$$

$$(17),\ N \text{ even} \Rightarrow \qquad = \begin{cases} \operatorname{sign}\left(b^{(0)}(\gamma^{(0)}_{m-2})\right) & m \in \mathbb{N}_{M-\mu+3}^{M+2}, \\ \operatorname{sign}\left(b^{(N)}(\kappa_{m+\mu-M})\right) & m \in \mathbb{N}_{M-\mu+1}^{M-\mu+2}, \\ \operatorname{sign}\left(b^{(0)}(\gamma^{(0)}_{m})\right) & m \in \mathbb{N}_1^{M-\mu}; \end{cases} \tag{A.19}$$

$$(A.12),\ (A.16) \Rightarrow \qquad = (-1)^{(M+2)-m} \quad \forall m \in \mathbb{N}_1^{M+2}. \tag{A.20}$$

From (24) and (A.20) it follows that $q^{(2)}(x)$ is absolutely integrable. If the discrete spectrum of the background potential is empty, the proof reduces to (A.15) to (A.20), where $M = 0$ and therefore $\mu = 0$, so $\mathbb{N}_1^{M-\mu} = \mathbb{N}_1^0$ and $\mathbb{N}_{M-\mu+3}^{M+2} = \mathbb{N}_3^2$ are empty sets. □

Appendix B. Derivation of the proposed algorithm

Our proposed algorithm decomposes the Crum transform in a specific chain of 1- and 2-fold Crum transforms. Furthermore, it uses a specific representation of the dressing method that mitigates numerical errors. In this appendix we show the derivation of the 1- and 2-fold Crum transforms in this specific representation.

This appendix is organised as follows. In Appendix B.1 we show how to translate between different representations of the Schrödinger eigenvalue problem. In Appendix B.2 we find expressions for the Wronskians of different trajectories of the Schrödinger eigenvalue problem and their derivatives. These intermediate results are used in Appendix B.3 to derive the two building blocks of the proposed algorithm: the dressing method for $N = 1$ and $N = 2$ respectively, in their simplest representation from an analytic point of view. Finally, in Appendix B.4, we translate the results from Appendix B.3 using those from Appendix B.1 to the numerically advantageous representation in which the dressing method is used in the proposed algorithm. (See Section 4.)

B1. Relation between bases for the dressing method and the Schrödinger eigenvalue problem

In the main text we introduced two operators that can be used to write the Schrödinger Eq. (2) as a system of first order equations (20) and also for the formulation of the dressing method (21). Here we show how to translate between expressions for either operator. Both for convenience and to establish the connection to other operators appearing in literature, we include the principle operator from [16]. Hence we consider

$$\mathbf{V}_{\mathrm{C}}(x,\zeta) := \begin{bmatrix} 1 \\ \frac{\partial}{\partial x} \end{bmatrix}, \tag{B.1}$$

$$\mathbf{V}_{\mathrm{S}}(x,\zeta) := \frac{1}{2\mathrm{j}\zeta}\begin{bmatrix} \mathrm{j}\zeta - \frac{\partial}{\partial x} \\ \mathrm{j}\zeta + \frac{\partial}{\partial x} \end{bmatrix}, \tag{B.2}$$

$$\mathbf{V}_{\mathrm{E}}(x,\zeta) := \frac{1}{2\mathrm{j}\zeta}\begin{bmatrix} \mathrm{e}^{\mathrm{j}\zeta x}\left(\mathrm{j}\zeta - \frac{\partial}{\partial x}\right) \\ \mathrm{e}^{-\mathrm{j}\zeta x}\left(\mathrm{j}\zeta + \frac{\partial}{\partial x}\right) \end{bmatrix}. \tag{B.3}$$

The relation between the S basis and several other bases appearing in literature, among which the C basis, is a similarity transformation, described by transformation matrices $\boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)$, where $\mathfrak{a}$ and $\mathfrak{b}$ are different bases, see [16, App. C]. Their relation to the E basis is not a similarity transformation, because $\mathbf{V}_{\mathrm{E}}(x,\zeta)$ depends on x whereas $\mathbf{V}_{\mathrm{C}}(x,\zeta)$ and $\mathbf{V}_{\mathrm{S}}(x,\zeta)$ do not, and so $\boldsymbol{T}_{\mathrm{E}}^{\mathrm{C}}(x,\zeta)$ and $\boldsymbol{T}_{\mathrm{E}}^{\mathrm{S}}(x,\zeta)$ depend on x. Therefore, to include the E basis, we generalise [16, Eqs. (70)–(75)] to

$$\mathbf{V}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\mathbf{V}_{\mathfrak{a}}(x,\zeta), \tag{B.4}$$

$$\boldsymbol{f}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{f}_{\mathfrak{a}}(x,\zeta), \tag{B.5}$$

$$\boldsymbol{\Phi}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{\Phi}_{\mathfrak{a}}(x,\zeta), \tag{B.6}$$

$$\boldsymbol{A}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{A}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta) + \left(\tfrac{\partial}{\partial x}\boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right)\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta)\,(\text{see (B.16)}), \tag{B.7}$$

$$\boldsymbol{H}_{\mathfrak{b}}(x_1,x_2,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x_2,\zeta)\,\boldsymbol{H}_{\mathfrak{a}}(x_1,x_2,\zeta)\,\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x_1,\zeta), \tag{B.8}$$

$$\text{where } \boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta) = \left(\boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right)^{-1}, \tag{B.9}$$

and for the dressing method we add

$$\boldsymbol{C}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{C}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta). \tag{B.10}$$

From (B.1), (B.2), (B.3) one can find

$$\boldsymbol{T}_{\mathrm{S}}^{\mathrm{E}}(x,\zeta) = \begin{bmatrix} \mathrm{e}^{\mathrm{j}\zeta x} & 0 \\ 0 & \mathrm{e}^{-\mathrm{j}\zeta x} \end{bmatrix} \quad\Rightarrow\quad \boldsymbol{T}_{\mathrm{E}}^{\mathrm{S}}(x,\zeta) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}\zeta x} & 0 \\ 0 & \mathrm{e}^{\mathrm{j}\zeta x} \end{bmatrix}; \tag{B.11}$$

$$\boldsymbol{T}_{\mathrm{C}}^{\mathrm{E}}(x,\zeta) = \frac{1}{2\mathrm{j}\zeta}\begin{bmatrix} \mathrm{j}\zeta\,\mathrm{e}^{\mathrm{j}\zeta x} & -\mathrm{e}^{\mathrm{j}\zeta x} \\ \mathrm{j}\zeta\,\mathrm{e}^{-\mathrm{j}\zeta x} & \mathrm{e}^{-\mathrm{j}\zeta x} \end{bmatrix} \quad\Rightarrow\quad \boldsymbol{T}_{\mathrm{E}}^{\mathrm{C}}(x,\zeta) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}\zeta x} & \mathrm{e}^{\mathrm{j}\zeta x} \\ -\mathrm{j}\zeta\,\mathrm{e}^{-\mathrm{j}\zeta x} & \mathrm{j}\zeta\,\mathrm{e}^{\mathrm{j}\zeta x} \end{bmatrix}. \tag{B.12}$$

Using (B.7), (B.11), and

$$\boldsymbol{A}_{\mathrm{S}}(x,\zeta) = \frac{q(x)}{2\mathrm{j}\zeta}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} - \begin{bmatrix} \mathrm{j}\zeta & 0 \\ 0 & -\mathrm{j}\zeta \end{bmatrix} \quad [16\text{ Eq. (20)}] \tag{B.13}$$

$$\text{we find } \boldsymbol{A}_{\mathrm{E}}(x,\zeta) = \frac{q(x)}{2\mathrm{j}\zeta}\begin{bmatrix} 1 & \exp(2\mathrm{j}\zeta x) \\ -\exp(-2\mathrm{j}\zeta x) & -1 \end{bmatrix}, \tag{B.14}$$

which we use in Algorithm 1, lines 3 and 4. Finally, from (B.7), (B.12) and (B.14) we find

$$\boldsymbol{A}_{\mathrm{C}}(x,\zeta) = \begin{bmatrix} 0 & 1 \\ (\mathrm{j}\zeta)^2 - q(x) & 0 \end{bmatrix}, \tag{B.15}$$

which is, like it should be, the same result as what can be found directly from (2), (20) and (B.1).

The derivation of (B.7) is as follows.

$$\begin{aligned} \frac{\partial}{\partial x}\boldsymbol{f}_{\mathfrak{b}}(x,\zeta) &= \frac{\partial}{\partial x}\left(\boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{f}_{\mathfrak{a}}(x,\zeta)\right) \\ &= \boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\left(\frac{\partial}{\partial x}\boldsymbol{f}_{\mathfrak{a}}(x,\zeta)\right) + \left(\frac{\partial}{\partial x}\boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\right)\boldsymbol{f}_{\mathfrak{a}}(x,\zeta) \\ &= \left(\boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{A}_{\mathfrak{a}}(x,\zeta) + \left(\frac{\partial}{\partial x}\boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\right)\right)\boldsymbol{f}_{\mathfrak{a}}(x,\zeta) \\ &= \underbrace{\left(\boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{A}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{T}^{\mathfrak{a}}_{\mathfrak{b}}(x,\zeta) + \left(\frac{\partial}{\partial x}\boldsymbol{T}^{\mathfrak{b}}_{\mathfrak{a}}(x,\zeta)\right)\boldsymbol{T}^{\mathfrak{a}}_{\mathfrak{b}}(x,\zeta)\right)}_{=\boldsymbol{A}_{\mathfrak{b}}(x,\zeta)}\boldsymbol{f}_{\mathfrak{b}}(x,\zeta). \end{aligned} \tag{B.16}$$

B2. Wronskians of trajectories of the Schrödinger equation and their derivatives

In Appendix B.3 we will need Wronskians of up to 3 trajectories of the Schrödinger equation and their derivatives up to second order. Let $f_n := f(x,\zeta_n)$ for $n \in \{1,2,3\}$ be solutions of the Schrödinger Eq. (2) *for the same potential* $q = q(x)$:

$$\left(\frac{\partial^2}{\partial x^2} + q\right) f_n = \lambda_n f_n, \tag{B.17}$$

where $\lambda_n := (\mathrm{j}\zeta_n)^2$ to shorten notation. Let $\boldsymbol{f}_{n\mathrm{C}} := \mathbf{V}_{\mathrm{C}}(x,\zeta_n)\,f_n = \begin{bmatrix} f_n & \frac{\partial f_n}{\partial x} \end{bmatrix}^\top$. Then,

$$f_n = \begin{bmatrix} 1 & 0 \end{bmatrix}\boldsymbol{f}_{n\mathrm{C}}, \tag{B.18}$$

$$\frac{\partial f_n}{\partial x} = \begin{bmatrix} 0 & 1 \end{bmatrix}\boldsymbol{f}_{n\mathrm{C}}, \tag{B.19}$$

$$\frac{\partial^2 f_n}{\partial x^2} = (\lambda_n - q) f_n = \begin{bmatrix} \lambda_n - q & 0 \end{bmatrix}\boldsymbol{f}_{n\mathrm{C}}, \tag{B.20}$$

$$\frac{\partial^3 f_n}{\partial x^3} = -\frac{\mathrm{d}q}{\mathrm{d}x} f_n + (\lambda_n - q)\frac{\partial f_n}{\partial x} = \begin{bmatrix} -\frac{\mathrm{d}q}{\mathrm{d}x} & \lambda_n - q \end{bmatrix}\boldsymbol{f}_{n\mathrm{C}}, \tag{B.21}$$

$$W[f_1] = f_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}\boldsymbol{f}_{1\mathrm{C}}, \tag{B.22}$$

$$\frac{\partial}{\partial x} W[f_1] = \frac{\partial f_1}{\partial x} = \begin{bmatrix} 0 & 1 \end{bmatrix}\boldsymbol{f}_{1\mathrm{C}}, \tag{B.23}$$

$$\frac{\partial^2}{\partial x^2} W[f_1] = \frac{\partial^2 f_1}{\partial x^2} = \begin{bmatrix} (\lambda_1 - q) & 0 \end{bmatrix}\boldsymbol{f}_{1\mathrm{C}}, \tag{B.24}$$

$$W[f_1; f_2] = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} \\ f_2 & \frac{\partial f_2}{\partial x} \end{vmatrix} = \begin{bmatrix} -\frac{\partial f_1}{\partial x} & f_1 \end{bmatrix}\boldsymbol{f}_{2\mathrm{C}}, \tag{B.25}$$

$$\frac{\partial}{\partial x} W[f_1; f_2] = \begin{vmatrix} f_1 & \frac{\partial^2 f_1}{\partial x^2} \\ f_2 & \frac{\partial^2 f_2}{\partial x^2} \end{vmatrix} = \begin{vmatrix} f_1 & (\lambda_1 - q) f_1 \\ f_2 & (\lambda_2 - q) f_2 \end{vmatrix} = \begin{vmatrix} f_1 & \lambda_1 f_1 \\ f_2 & \lambda_2 f_2 \end{vmatrix} = \begin{bmatrix} (\lambda_2 - \lambda_1) f_1 & 0 \end{bmatrix}\boldsymbol{f}_{2\mathrm{C}}, \tag{B.26}$$

$$\begin{aligned}\frac{\partial^2}{\partial x^2} W[f_1; f_2] &= \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} W[f_1; f_2]\right) = \frac{\partial}{\partial x}\left(\begin{bmatrix}(\lambda_2-\lambda_1)f_1 & 0\end{bmatrix}\boldsymbol{f}_{2\mathrm{C}}\right)\\ &= \begin{bmatrix}(\lambda_2-\lambda_1)\frac{\partial f_1}{\partial x} & 0\end{bmatrix}\boldsymbol{f}_{2\mathrm{C}} + \begin{bmatrix}(\lambda_2-\lambda_1)f_1 & 0\end{bmatrix}\begin{bmatrix}0 & 1\\ \lambda_2-q & 0\end{bmatrix}\boldsymbol{f}_{2\mathrm{C}} = (\lambda_2-\lambda_1)\begin{bmatrix}\frac{\partial f_1}{\partial x} & f_1\end{bmatrix}\boldsymbol{f}_{2\mathrm{C}}\end{aligned} \tag{B.27}$$

$$\begin{aligned}W[f_1; f_2; f_3] &= \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \frac{\partial^2 f_1}{\partial^2 x}\\ f_2 & \frac{\partial f_2}{\partial x} & \frac{\partial^2 f_2}{\partial^2 x}\\ f_3 & \frac{\partial f_3}{\partial x} & \frac{\partial^2 f_3}{\partial^2 x}\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & (\lambda_1-q)f_1\\ f_2 & \frac{\partial f_2}{\partial x} & (\lambda_2-q)f_2\\ f_3 & \frac{\partial f_3}{\partial x} & (\lambda_3-q)f_3\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \lambda_1 f_1\\ f_2 & \frac{\partial f_2}{\partial x} & \lambda_2 f_2\\ f_3 & \frac{\partial f_3}{\partial x} & \lambda_3 f_3\end{vmatrix} =\\ &\left[\begin{vmatrix}\frac{\partial f_1}{\partial x} & \lambda_1 f_1\\ \frac{\partial f_2}{\partial x} & \lambda_2 f_2\end{vmatrix} + \lambda_3\begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x}\\ f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} - \begin{vmatrix} f_1 & \lambda_1 f_1\\ f_2 & \lambda_2 f_2\end{vmatrix}\right]\boldsymbol{f}_{3\mathrm{C}} = \left[\begin{vmatrix}(\lambda_3-\lambda_1)f_1 & \frac{\partial f_1}{\partial x}\\ (\lambda_3-\lambda_2)f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} \quad (\lambda_1-\lambda_2)f_1 f_2\right]\boldsymbol{f}_{3\mathrm{C}},\end{aligned} \tag{B.28}$$

$$\begin{aligned}\frac{\partial}{\partial x} W[f_1; f_2; f_3] &= \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \frac{\partial^3 f_1}{\partial^3 x}\\ f_2 & \frac{\partial f_2}{\partial x} & \frac{\partial^3 f_2}{\partial^3 x}\\ f_3 & \frac{\partial f_3}{\partial x} & \frac{\partial^3 f_3}{\partial^3 x}\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & -\frac{\mathrm{d}q}{\mathrm{d}x}f_1 + (\lambda_1-q)\frac{\partial f_1}{\partial x}\\ f_2 & \frac{\partial f_2}{\partial x} & -\frac{\mathrm{d}q}{\mathrm{d}x}f_2 + (\lambda_2-q)\frac{\partial f_2}{\partial x}\\ f_3 & \frac{\partial f_3}{\partial x} & -\frac{\mathrm{d}q}{\mathrm{d}x}f_3 + (\lambda_3-q)\frac{\partial f_3}{\partial x}\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \lambda_1\frac{\partial f_1}{\partial x}\\ f_2 & \frac{\partial f_2}{\partial x} & \lambda_2\frac{\partial f_2}{\partial x}\\ f_3 & \frac{\partial f_3}{\partial x} & \lambda_3\frac{\partial f_3}{\partial x}\end{vmatrix}\\ &= \left[\begin{vmatrix}\frac{\partial f_1}{\partial x} & \lambda_1\frac{\partial f_1}{\partial x}\\ \frac{\partial f_2}{\partial x} & \lambda_2\frac{\partial f_2}{\partial x}\end{vmatrix} \quad \lambda_3\begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x}\\ f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} - \begin{vmatrix} f_1 & \lambda_1\frac{\partial f_1}{\partial x}\\ f_2 & \lambda_2\frac{\partial f_2}{\partial x}\end{vmatrix}\right]\boldsymbol{f}_{3\mathrm{C}} = \left[(\lambda_2-\lambda_1)\frac{\partial f_1}{\partial x}\frac{\partial f_2}{\partial x} \quad \begin{vmatrix} f_1 & (\lambda_3-\lambda_1)\frac{\partial f_1}{\partial x}\\ f_2 & (\lambda_3-\lambda_2)\frac{\partial f_2}{\partial x}\end{vmatrix}\right]\boldsymbol{f}_{3\mathrm{C}}.\end{aligned} \tag{B.29}$$

B3. Dressing method in C basis

We make use of the results of Appendix B.2 to derive the dressing method in C basis for the 1-fold Crum transform in Appendix B.3.1, and for the 2-fold Crum transform Appendix B.3.2.

B3.1. Dressing method in C basis for $N = 1$ (Darboux transform)

Starting from Theorem 1, making use of the expressions in Appendix B.2 we derive the dressing method for the Darboux transform in C basis, (21), (22) and (23), as follows.

$$f^{(1)}(x,\zeta) = \frac{W[\vartheta_1(x); f^{(0)}(x,\zeta)]}{W[\vartheta_1(x)]} = \frac{\left[-\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x) \quad \vartheta_1(x)\right]\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\vartheta_1(x)}, \tag{B.30}$$

$$\begin{aligned}\frac{\partial}{\partial x} f^{(1)}(x,\zeta) &= \frac{\partial}{\partial x}\frac{W[\vartheta_1(x); f^{(0)}(x,\zeta)]}{W[\vartheta_1(x)]}\\ &= \frac{W[\vartheta_1(x)]\frac{\partial W[\vartheta_1(x); f^{(0)}(x,\zeta)]}{\partial x} - \frac{\mathrm{d}W[\vartheta_1(x)]}{\mathrm{d}x}W[\vartheta_1(x), f^{(0)}(x,\zeta)]}{\left(W[\vartheta_1(x)]\right)^2}\\ &= \frac{\left(\vartheta_1(x)\begin{bmatrix}\left((\mathrm{j}\zeta)^2-\kappa_1^2\right)\vartheta_1(x) & 0\end{bmatrix} - \left(\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\right)\begin{bmatrix}-\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x) & \vartheta_1(x)\end{bmatrix}\right)\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\left(\vartheta_1(x)\right)^2}\\ &= \begin{bmatrix}(\mathrm{j}\zeta)^2-\kappa_1^2+\left(\frac{\mathrm{d}\vartheta_1(x)}{\mathrm{d}x}\Big/\vartheta_1(x)\right)^2 & -\left(\frac{\mathrm{d}\vartheta_1(x)}{\mathrm{d}x}\Big/\vartheta_1(x)\right)\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta).\end{aligned} \tag{B.31}$$

Eqs. (B.30) and (B.31) imply the trajectory mapping in (21) with the dressing matrix in (23). Again from Theorem 1 we obtain the potential update:

$$\Delta q_{(0)}^{(1)}(x) = q^{(1)}(x) - q^{(0)}(x) = 2\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\left(W[\vartheta_1(x)]\right) = 2\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\left(\vartheta_1(x)\right)$$

$$
\begin{aligned}
&= 2\frac{\mathrm{d}}{\mathrm{d}x}\frac{\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)}{\vartheta_1(x)} = 2\frac{\big(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\vartheta_1(x)\big)\vartheta_1(x) - \big(\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\big)^2}{\big(\vartheta_1(x)\big)^2}\\
&= 2\frac{\big(\kappa_1^2 - q^{(0)}(x)\big)\big(\vartheta_1(x)\big)^2 - \big(\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\big)^2}{\big(\vartheta_1(x)\big)^2}\\
&= -2\,q^{(0)}(x) + 2\kappa_1^2 - 2\Big(\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\Big/\,\vartheta_1(x)\Big)^2.
\end{aligned}
\tag{B.32}
$$

This potential update is equivalent to (22).

B3.2. Dressing method in C basis for $N = 2$

Starting from Theorem 1, making use of the expressions in Appendix B.2 we derive the dressing method for the 2-fold Crum transform in C basis. This result is not used directly in the main text, but serves as an intermediate step in the derivation in Appendix B.4.2 of the dressing method for the 2-fold Crum transform in E basis, which is part of the proposed algorithm.

$$
f^{(2)}(x,\zeta) = \frac{W[\vartheta_1(x);\vartheta_2(x);f^{(0)}(x,\zeta)]}{W[\vartheta_1(x);\vartheta_2(x)]}
\tag{B.33}
$$

$$
= \frac{\begin{bmatrix}\begin{vmatrix}((\mathrm{j}\zeta)^2-\kappa_1^2)\,\vartheta_1(x) & \frac{\partial}{\partial x}\vartheta_1(x)\\ ((\mathrm{j}\zeta)^2-\kappa_2^2)\,\vartheta_2(x) & \frac{\partial}{\partial x}\vartheta_2(x)\end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\begin{vmatrix}\vartheta_1(x) & \frac{\partial}{\partial x}\vartheta_1(x)\\ [1pt]\vartheta_2(x) & \frac{\partial}{\partial x}\vartheta_2(x)\end{vmatrix}}
$$

$$
= (\mathrm{j}\zeta)^2\begin{bmatrix}1 & 0\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \frac{\begin{bmatrix}-\begin{vmatrix}\kappa_1^2\,\vartheta_1(x) & \frac{\partial}{\partial x}\vartheta_1(x)\\ \kappa_2^2\,\vartheta_2(x) & \frac{\partial}{\partial x}\vartheta_2(x)\end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\begin{vmatrix}\vartheta_1(x) & \frac{\partial}{\partial x}\vartheta_1(x)\\ \vartheta_2(x) & \frac{\partial}{\partial x}\vartheta_2(x)\end{vmatrix}}
$$

$$
= (\mathrm{j}\zeta)^2\begin{bmatrix}1 & 0\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \frac{\begin{bmatrix}-\begin{vmatrix}\kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}(x)\\ \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}(x)\end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\begin{vmatrix}\boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x)\end{vmatrix}},
$$

$$
\big(W[\vartheta_1(x);\vartheta_2(x)]\big)^2\frac{\partial}{\partial x}f^{(2)}(x,\zeta) = \big|\boldsymbol{\vartheta}_{1\mathrm{C}}(x) \quad \boldsymbol{\vartheta}_{2\mathrm{C}}(x)\big|^2\frac{\partial}{\partial x}f^{(2)}(x,\zeta)
$$

$$
= \begin{vmatrix} W[\vartheta_1(x);\vartheta_2(x)] & W[\vartheta_1(x);\vartheta_2(x);f^{(0)}(x,\zeta)]\\ \frac{\mathrm{d}}{\mathrm{d}x}W[\vartheta_1(x);\vartheta_2(x)] & \frac{\partial}{\partial x}W[\vartheta_1(x);\vartheta_2(x);f^{(0)}(x,\zeta)]\end{vmatrix}
$$

$$
= \begin{vmatrix}\vartheta_1(x) & \frac{\partial\,\vartheta_1(x)}{\partial x}\\ \vartheta_2(x) & \frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix}\begin{bmatrix}(\kappa_2^2-\kappa_1^2)\frac{\partial\,\vartheta_1(x)}{\partial x}\frac{\partial\,\vartheta_2(x)}{\partial x} & \begin{vmatrix}\vartheta_1(x) & ((\mathrm{j}\zeta)^2-\kappa_1^2)\frac{\partial\,\vartheta_1(x)}{\partial x}\\ \vartheta_2(x) & ((\mathrm{j}\zeta)^2-\kappa_2^2)\frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix}\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots
$$

$$
(\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\begin{bmatrix}\begin{vmatrix}((\mathrm{j}\zeta)^2-\kappa_1^2)\,\vartheta_1(x) & \frac{\partial\,\vartheta_1(x)}{\partial x}\\ ((\mathrm{j}\zeta)^2-\kappa_2^2)\,\vartheta_2(x) & \frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)
$$

$$
= (\mathrm{j}\zeta)^2\begin{vmatrix}\vartheta_1(x) & \frac{\partial\,\vartheta_1(x)}{\partial x}\\ \vartheta_2(x) & \frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix}^2\begin{bmatrix}(\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\begin{vmatrix}\vartheta_1(x) & \frac{\partial\,\vartheta_1(x)}{\partial x}\\ \vartheta_2(x) & \frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix}^{-1} & 1\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots
$$

$$
\begin{vmatrix}\vartheta_1(x) & \frac{\partial\,\vartheta_1(x)}{\partial x}\\ \vartheta_2(x) & \frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix}\begin{bmatrix}(\kappa_2^2-\kappa_1^2)\frac{\partial\,\vartheta_1(x)}{\partial x}\frac{\partial\,\vartheta_2(x)}{\partial x} & -\begin{vmatrix}\vartheta_1(x) & \kappa_1^2\frac{\partial\,\vartheta_1(x)}{\partial x}\\ \vartheta_2(x) & \kappa_2^2\frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix}\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots
$$

$$
(\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\begin{bmatrix}-\begin{vmatrix}\kappa_1^2\,\vartheta_1(x) & \frac{\partial\,\vartheta_1(x)}{\partial x}\\ \kappa_2^2\,\vartheta_2(x) & \frac{\partial\,\vartheta_2(x)}{\partial x}\end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\end{bmatrix}\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)
$$

$$
\begin{aligned}
&= (\mathrm{j}\zeta)^2 \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}^2 \begin{bmatrix} \dfrac{(\kappa_1^2-\kappa_2^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x)}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}} & 1 \end{bmatrix} \boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots \\
&\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix} \begin{bmatrix} (\kappa_2^2-\kappa_1^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) & -\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix} \end{bmatrix} \boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots \\
&\begin{bmatrix} (\kappa_2^2-\kappa_1^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) \begin{vmatrix} \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) \end{bmatrix} \boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) \\
&= (\mathrm{j}\zeta)^2 \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}^2 \begin{bmatrix} \dfrac{(\kappa_1^2-\kappa_2^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x)}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}} & 1 \end{bmatrix} \boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots \\
&= \begin{bmatrix} (\kappa_2^2-\kappa_1^2) \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) - \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}^2(x) - \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) \end{vmatrix} & \cdots \\ (\kappa_1^2-\kappa_2^2)^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) - \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix} \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix} \end{bmatrix} \boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta). \qquad \text{(B.34)}
\end{aligned}
$$

Hence, from (B.33) and (B.34):

$$
\boldsymbol{C}_{\mathrm{C}(0)}^{(2)}(x,\zeta) = (\mathrm{j}\zeta)^2 \begin{bmatrix} 1 & 0 \\ \dfrac{(\kappa_1^2-\kappa_2^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x)}{w(x)} & 1 \end{bmatrix} + \cdots \qquad \text{(B.35)}
$$

$$
\frac{1}{w^2(x)} \begin{bmatrix} -w(x) \begin{vmatrix} \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix} & \cdots \\ (\kappa_2^2-\kappa_1^2) \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) - \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) - \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}2}^2(x) \end{vmatrix} & \cdots \end{bmatrix} \qquad \text{(B.36)}
$$

$$
\begin{bmatrix} -w(x)(\kappa_2^2-\kappa_1^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) \\ (\kappa_2^2-\kappa_1^2)^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) - w(x) \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix} \end{bmatrix},
$$

where $w(x) := W[\vartheta_1(x);\vartheta_2(x)] = \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}$.

The potential update is

$$
\begin{aligned}
\Delta q_{(0)}^{(2)}(x) &= q^{(2)}(x) - q^{(0)}(x) = 2\frac{\mathrm{d}^2}{\mathrm{d}x^2} \ln\left(W[\vartheta_1(x);\vartheta_2(x)]\right) = 2\frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}x} W[\vartheta_1(x);\vartheta_2(x)] \\
&= 2\frac{\dfrac{\mathrm{d}^2}{\mathrm{d}x^2} W[\vartheta_1(x);\vartheta_2(x)]}{W[\vartheta_1(x);\vartheta_2(x)]} - 2\left(\frac{\dfrac{\mathrm{d}}{\mathrm{d}x} W[\vartheta_1(x);\vartheta_2(x)]}{W[\vartheta_1(x);\vartheta_2(x)]}\right)^2 \\
&= \frac{2(\kappa_2^2-\kappa_1^2)\left(\vartheta_1(x)\dfrac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} + \dfrac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\vartheta_2(x)\right)}{w(x)} - 2\left(\frac{(\kappa_2^2-\kappa_1^2)\,\vartheta_1(x)\,\vartheta_2(x)}{w(x)}\right)^2 \\
&= \frac{2(\kappa_2^2-\kappa_1^2)\left(\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) + \boldsymbol{\vartheta}_{1\mathrm{C}2}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\right)}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}} - 2\left(\frac{(\kappa_2^2-\kappa_1^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x)}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}}\right)^2. \\
&= 2(\kappa_2^2-\kappa_1^2)\frac{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}^2(x) \end{vmatrix} - (\kappa_2^2-\kappa_1^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x)}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}^2} \\
&= 2(\kappa_2^2-\kappa_1^2)\frac{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) - \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}^2(x) - \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) \end{vmatrix}}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}^2}. \qquad \text{(B.37)}
\end{aligned}
$$

B4. Dressing method in E basis

We make use of the results of Appendix B.3 to derive the dressing method in E basis for the 1-fold Crum transform in Appendix B.4.1, and for the 2-fold Crum transform Appendix B.4.2. These results are used in Section 4.

B4.1. Dressing method in E basis for $N = 1$ (Darboux transform)

Starting from (21), (22) and (23), we can derive the dressing method for the Darboux transform in E basis, the case $N = 1$ of (30) and (31) in Section 4.1, as follows. For convenience, rewrite (23) using (B.15) and $\sigma_1(x) := \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \big/ \vartheta_1(x)$ as

$$\begin{aligned}\boldsymbol{C}^{(1)}_{\mathrm{C}(0)}(x,\zeta) &= \begin{bmatrix} -\sigma_1(x) & 1 \\ (\sigma_1(x)-\kappa_1)(\sigma_1(x)+\kappa_1)+(\mathrm{j}\zeta)^2 & -\sigma_1(x)\end{bmatrix} \\ &\equiv \boldsymbol{A}_{\mathrm{C}}(x,\zeta)\Big|_{q(x)=\kappa_1^2-\sigma_1^2(x)} - \sigma_1(x)\begin{bmatrix}1&0\\0&1\end{bmatrix}.\end{aligned} \tag{B.38}$$

Then use (B.7), (B.10), (B.12) and (B.14) to find

$$\begin{aligned}\boldsymbol{C}^{(1)}_{\mathrm{E}(0)}(x,\zeta) &= \boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\,\boldsymbol{C}^{(1)}_{\mathrm{C}(0)}(x,\zeta)\,\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) \\ &= \boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\,\boldsymbol{A}_{\mathrm{C}}(x,\zeta)\Big|_{q(x)=\kappa_1^2-\sigma_1^2(x)}\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) - \sigma_1(x)\,\boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\begin{bmatrix}1&0\\0&1\end{bmatrix}\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) \\ &= \boldsymbol{A}_{\mathrm{E}}(x,\zeta)\Big|_{q(x)=\kappa_1^2-\sigma_1^2(x)} - \Big(\frac{\partial}{\partial x}\boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\Big)\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) - \sigma_1(x)\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ &= \frac{\kappa_1^2-\sigma_1^2(x)}{2\mathrm{j}\zeta}\begin{bmatrix}1 & \exp(2\mathrm{j}\zeta x)\\ -\exp(-2\mathrm{j}\zeta x) & -1\end{bmatrix} + \mathrm{j}\zeta\begin{bmatrix}-1&0\\0&1\end{bmatrix} - \sigma_1(x)\begin{bmatrix}1&0\\0&1\end{bmatrix}.\end{aligned} \tag{B.39}$$

Subsequently, we need to express $\sigma_1(x)$ as a function of $\boldsymbol{\vartheta}_{1\mathrm{E}}(x)$.

$$\begin{aligned}\boldsymbol{\vartheta}_{n\mathrm{C}}(x) &= \boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\mathrm{j}\kappa_n)\,\boldsymbol{\vartheta}_{n\mathrm{E}}(x) = \begin{bmatrix}\mathrm{e}^{\kappa_n x} & \mathrm{e}^{-\kappa_n x}\\ \kappa_n\mathrm{e}^{\kappa_n x} & -\kappa_n\mathrm{e}^{-\kappa_n x}\end{bmatrix}\begin{bmatrix}\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\\ \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}\\ \kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} - \kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}\end{bmatrix},\ \text{so}\end{aligned} \tag{B.40}$$

$$\sigma_n(x) = \frac{\boldsymbol{\vartheta}_{n\mathrm{C}2}(x)}{\boldsymbol{\vartheta}_{n\mathrm{C}1}(x)} = \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} - \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}; \tag{B.41}$$

$$\begin{aligned}\kappa_n + \sigma_n(x) &= \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}} + \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} - \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}} \\ &= \frac{2\kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}};\end{aligned} \tag{B.42}$$

$$\begin{aligned}\kappa_n - \sigma_n(x) &= \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}} - \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} - \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}} \\ &= \frac{2\kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}}.\end{aligned} \tag{B.43}$$

Substitution in (B.39) and (B.32) results in

$$\begin{aligned}\boldsymbol{C}^{(1)}_{\mathrm{E}(0)}(x,\zeta) = (\mathrm{j}\zeta)^{-1}\overbrace{\frac{2\kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{E}2}(x)}{\big(\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}\big)^2}\begin{bmatrix}1 & \mathrm{e}^{2\mathrm{j}\zeta x}\\ -\mathrm{e}^{-2\mathrm{j}\zeta x} & -1\end{bmatrix} + \cdots}^{=\boldsymbol{M}^{(1)}_{-1}(x,\zeta)} \\ \underbrace{\mathrm{j}\zeta\begin{bmatrix}-1&0\\0&1\end{bmatrix}}_{=\boldsymbol{M}^{(1)}_{1}(x,\zeta)} + \underbrace{\kappa_1\frac{\boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_1 x}}{\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_1 x}}\begin{bmatrix}1&0\\0&1\end{bmatrix}}_{=\boldsymbol{M}^{(1)}_{0}(x,\zeta)},\end{aligned} \tag{B.44}$$

$$\Delta q^{(1)}_{(0)}(x) = 2(\kappa_1-\sigma_1(x))(\kappa_1+\sigma_1(x)) - 2\,q^{(0)}(x) = \frac{8\kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{E}2}(x)}{\big(\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\,\mathrm{e}^{-\kappa_n x}\big)^2} - 2\,q^{(0)}(x). \tag{B.45}$$

This is equivalent to the equations in Section 4.1 for the case $N = 1$.

B4.2. Dressing method in E basis for $N = 2$

For the dressing method in E basis for $N = 2$, we find the trajectory mapping by applying (B.10) with (B.12) to (B.35). Let us first express (B.35) as

$$\boldsymbol{C}^{(2)}_{\mathrm{C}(0)}(x,\zeta) = (\mathrm{j}\zeta)^2 \begin{bmatrix} 1 & 0 \\ \dfrac{a}{w(x)} & 1 \end{bmatrix} + \frac{1}{w^2(x)} \begin{bmatrix} 2b\,w(x) & a\,w(x) \\ 2c & 2d \end{bmatrix}, \quad \text{where} \tag{B.46}$$

$$a := (\kappa_1^2 - \kappa_2^2)\, \boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\, \boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\,, \tag{B.47}$$

$$b := -\frac{1}{2} \begin{vmatrix} \kappa_1^2\, \boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \kappa_2^2\, \boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix}, \tag{B.48}$$

$$c := \frac{1}{2}(\kappa_2^2 - \kappa_1^2) \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\, \boldsymbol{\vartheta}_{1\mathrm{C}2}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) - \kappa_1^2\, \boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\, \boldsymbol{\vartheta}_{2\mathrm{C}2}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}^2(x) - \kappa_2^2\, \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) \end{vmatrix}, \tag{B.49}$$

$$d := \frac{1}{2}(\kappa_2^2 - \kappa_1^2)^2\, \boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x)\, \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) - \frac{1}{2} w(x) \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \kappa_1^2\, \boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \kappa_2^2\, \boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix}, \tag{B.50}$$

$$w(x) := W[\vartheta_1(x); \vartheta_2(x)] = \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}. \tag{B.51}$$

Applying (B.10) with (B.12) then gives

$$\begin{aligned} \boldsymbol{C}^{(2)}_{\mathrm{E}(0)}(x,\zeta) = (\mathrm{j}\zeta)\, \overbrace{\frac{a}{w(x)} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}}^{=\boldsymbol{M}_1^{(2)}(x,\zeta)} + (\mathrm{j}\zeta)^{-1} \overbrace{\frac{1}{w^2(x)} \begin{bmatrix} -c & -c\,\mathrm{e}^{2\mathrm{j}\zeta x} \\ c\,\mathrm{e}^{-2\mathrm{j}\zeta x} & c \end{bmatrix}}^{=\boldsymbol{M}_{-1}^{(2)}(x,\zeta)} + \cdots \\ \underbrace{\frac{1}{w^2(x)} \begin{bmatrix} b\,w(x) + d & (b\,w(x) - d)\mathrm{e}^{2\mathrm{j}\zeta x} \\ (b\,w(x) - d)\mathrm{e}^{-2\mathrm{j}\zeta x} & b\,w(x) + d \end{bmatrix}}_{=\boldsymbol{M}_0^{(2)}(x,\zeta)} + (\mathrm{j}\zeta)^2 \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{=\boldsymbol{M}_2^{(2)}(x,\zeta)}. \end{aligned} \tag{B.52}$$

Substitution of (B.40) results in

$$a = \left(\kappa_1^2 - \kappa_2^2\right)\left(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x}\right)\left(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x}\right), \tag{B.53}$$

$$c = 2(\kappa_2^2 - \kappa_1^2)\kappa_1\kappa_2 \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}1}^2(x)\,\mathrm{e}^{2\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}2}^2(x)\,\mathrm{e}^{-2\kappa_1 x} & \kappa_1\, \boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\, \boldsymbol{\vartheta}_{1\mathrm{E}2}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{E}1}^2(x)\,\mathrm{e}^{2\kappa_2 x} - \boldsymbol{\vartheta}_{2\mathrm{E}2}^2(x)\,\mathrm{e}^{-2\kappa_2 x} & \kappa_2\, \boldsymbol{\vartheta}_{2\mathrm{E}1}(x)\, \boldsymbol{\vartheta}_{2\mathrm{E}2}(x) \end{vmatrix}, \tag{B.54}$$

$$w(x) = \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x} & \kappa_1\left(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x}\right) \\ \boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x} & \kappa_2\left(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} - \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x}\right) \end{vmatrix}, \tag{B.55}$$

$$b\,w(x) - d = 2(\kappa_1^2 - \kappa_2^2) \begin{vmatrix} (\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x})^2 & \kappa_1^2\, \boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\, \boldsymbol{\vartheta}_{1\mathrm{E}2}(x) \\ (\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x})^2 & \kappa_2^2\, \boldsymbol{\vartheta}_{2\mathrm{E}1}(x)\, \boldsymbol{\vartheta}_{2\mathrm{E}2}(x) \end{vmatrix}, \tag{B.56}$$

$$b\,w(x) + d = \frac{1}{2}\left(a^2 - (\kappa_1^2 + \kappa_2^2)\, w^2(x)\right). \tag{B.57}$$

The equations for the trajectory mapping in Section 4.1 for the case $N = 2$ then follow by substitution in (B.52). Finally, we find the potential update for that case by substituting (B.40) in (B.37):

$$\begin{aligned} \Delta q^{(2)}_{(0)}(x) &= 2(\kappa_2^2 - \kappa_1^2) \frac{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}^2(x) - \kappa_1^2\, \boldsymbol{\vartheta}_{1\mathrm{C}1}^2(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}^2(x) - \kappa_2^2\, \boldsymbol{\vartheta}_{2\mathrm{C}1}^2(x) \end{vmatrix}}{\left| \boldsymbol{\vartheta}_{1\mathrm{C}}(x) \quad \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \right|^2} \\ &= 8(\kappa_1^2 - \kappa_2^2) \frac{\begin{vmatrix} (\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x})^2 & \kappa_1^2\, \boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\, \boldsymbol{\vartheta}_{1\mathrm{E}2}(x) \\ (\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x})^2 & \kappa_2^2\, \boldsymbol{\vartheta}_{2\mathrm{E}1}(x)\, \boldsymbol{\vartheta}_{2\mathrm{E}2}(x) \end{vmatrix}}{\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x} & \kappa_1\left(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x}\right) \\ \boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x} & \kappa_2\left(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} - \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x}\right) \end{vmatrix}^2}. \end{aligned} \tag{B.58}$$

References

[1] Korteweg DJ, De Vries G. On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. London Edinb Dublin Philos Mag J Sci Series 5 1895;39(240):422–43. doi:10.1080/14786449508620739.
[2] Hammack JL, Segur H. The Korteweg–de Vries equation and water waves. Part 2. Comparison with experiments. J Fluid Mech 1974;65(2):289–314. doi:10.1017/s002211207400139x.
[3] Brühl M, Becker M. Analysis of subaerial landslide data using nonlinear Fourier transform based on Korteweg-de Vries equation (KdV-NFT). J Earthq Tsunami 2018;12(02):1840002. doi:10.1142/s179343111840002x.
[4] Lannes D. Modeling shallow water waves. Nonlinearity 2020;33(5):R1–R57. doi:10.1088/1361-6544/ab6c7c.
[5] Ak T, Saha A, Dhawan S, Kara AH. Investigation of Coriolis effect on oceanic flows and its bifurcation via geophysical Korteweg–de Vries equation. Numer Methods Partial DifferEqu 2020. doi:10.1002/num.22469.
[6] Misra J, Patra M. A study of solitary waves in a tapered aorta by using the theory of solitons. Comput Math Appl 2007;54(2):242–54. doi:10.1016/j.camwa.2006.12.025.
[7] Abdou M, Hendi A, Alanzi HK. New exact solutions of KdV equation in an elastic tube filled with a variable viscosity fluid. Stud Nonlinear Sci 2012;3(2):62–8. doi:10.5829/idosi.sns.2012.3.2.247. https://idosi.org/sns/3(2)12/2.pdf
[8] Crighton DG. Applications of KdV. Acta Appl Math 1995;39(1-3):39–67. doi:10.1007/bf00994625.
[9] Temnov VV, Klieber C, Nelson KA, Thomay T, Knittel V, Leitenstorfer A, et al. Femtosecond nonlinear ultrasonics in gold probed with ultrashort surface plasmons. Nat Commun 2013;4(1). doi:10.1038/ncomms2480. Art. no. 1468
[10] Ricketts DS, Ham D. Electrical Solitons. CRC Press; 2011. doi:101201/9781315217802. ISBN 978-1-4398-2980-6
[11] Gardner CS, Greene JM, Kruskal MD, Miura RM. Method for solving the Korteweg-deVries equation. Phys Rev Lett 1967;19(19):1095–7. doi:10.1103/physrevlett.19.1095.
[12] Lax PD. Integrals of nonlinear equations of evolution and solitary waves. Commun Pure Appl Math 1968;21(5):467–90. doi:10.1002/cpa.3160210503.
[13] Ablowitz MJ, Kaup DJ, Newell AC, Segur H. The inverse scattering transform - Fourier analysis for nonlinear problems. Stud Appl Math 1974;53(4):249–315. doi:10.1002/sapm1974534249.
[14] Wahls S, Chimmalgi S, Prins PJ. FNFT: a software library for computing nonlinear Fourier transforms. J Open Source Softw 2018;3(23):597. doi:10.21105/joss.00597.
[15] Chimmalgi S, Prins PJ, Wahls S. Fast nonlinear Fourier transform algorithms using higher order exponential integrators. IEEE Access 2019;7:145161–76. doi:10.1109/access.2019.2945480.
[16] Prins PJ, Wahls S. Soliton phase shift calculation for the Korteweg–de Vries equation. IEEE Access 2019;7:122914–30. doi:10.1109/access.2019.2932256.
[17] Span A, Aref V, Bülow H, ten Brink S. Successive eigenvalue removal for multi-soliton spectral amplitude estimation. J Lightwave Technol 2020. doi:10.1109/jlt.2020.2994156.
[18] Chekhovskoy I, Medvedev SB, Vaseva IA, Sedov EV, Fedoruk MP. Introducing phase jump tracking – a fast method for eigenvalue evaluation of the direct Zakharov-Shabat problem. Commun Nonlinear Sci Numer Simulat 2021;96:105718. doi:10.1016/j.cnsns.2021.105718.
[19] Ablowitz MJ, Kodama Y. Note on asymptotic solutions of the Korteweg-de Vries equation with solitons. Stud Appl Math 1982;66(2):159–70. doi:10.1002/sapm1982662159.
[20] Osborne AR. Non-linear Fourier analysis for the infinite-interval Korteweg–de Vries equation I: an algorithm for the direct scattering transform. J Comput Phys 1991;94:284–313. doi:10.1016/0021-9991(91)90223-8.
[21] Christov I. Internal solitary waves in the ocean: analysis using the periodic, inverse scattering transform. Math Comput Simul 2009;80(1):192–201. doi:10.1016/j.matcom.2009.06.005.
[22] Brühl M., Ujvary S., Barranco I., Prins P.J., Wahls S., Liu P.L.-F.. Comparative analysis of bore propagation using conventional linear and KdV-based nonlinear Fourier transform; 2021 In preparation.
[23] Deift P, Trubowitz E. Inverse scattering on the line. Commun Pure Appl Math 1979;32(2):121–251. doi:10.1002/cpa.3160320202.
[24] Chadan K, Sabatier PC. Inverse problems in quantum scattering theory. 2nd (ed). Berlin, Heidelberg: Springer; 1989. doi:10.1007/978-3-642-83317-5. ISBN 978-3-642-83319-9
[25] Sacks P, Shin J. Computational methods for some inverse scattering problems. Appl. Math. Comput. 2009;207(1):111–23. doi:10.1016/j.amc.2008.10.033.
[26] Chimmalgi S. Improved fast inverse nonlinear Fourier transform for multi-solitons: a discrete Darboux based approach. Delft University of Technology; 2017. Master's thesis. URL http://resolver.tudelft.nl/uuid:adf4d14a-fecd-400f-a1be-3e6c32d23a06
[27] Gelash AA, Agafontsev DS. Strongly interacting soliton gas and formation of rogue waves. Phys Rev E 2018;98(4). doi:10.1103/physreve.98.042210.
[28] Neugebauer G, Meinel R. General *N*-soliton solution of the AKNS class on arbitrary background. Phys Lett A 1984;100(9):467–70. doi:10.1016/0375-9601(84)90827-2.
[29] Ablowitz MJ. Integrability and nonlinear waves. In: Kevrekidis PG, Cuevas-Maraver J, Saxena A, editors. Emerging Frontiers in nonlinear science. Springer International Publishing; 2020. p. 161–84. doi:10.1007/978-3-030-44992-6_7.
[30] Prins PJ, Wahls S. Higher order exponential splittings for the fast non-linear Fourier transform of the Korteweg-de Vries equation. In: 2018 IEEE International conference on acoustics, speech and signal processing (ICASSP); 2018. p. 4524–8. doi:10.1109/ICASSP.2018.8461708.
[31] Ablowitz MJ, Segur H. Solitons and the inverse scattering transform. Philadelphia, PA, USA: SIAM; 1981. doi:101137/19781611970883.
[32] Pöschel J, Trubowitz E. Inverse spectral theory. Pure and applied mathematics. Elsevier; 1987. doi:10.1016/s0079-8169(08)x6138-0. ISBN 978-0-12-563040-5
[33] Lamb GL. Elements of soliton theory. Wiley; 1980. ISBN 0-471-04559-4.
[34] Van Der Mee C, Seatzu S, Theis D. Structured matrix algorithms for inverse scattering on the line. Calcolo 2007;44(2):59–87. doi:10.1007/s10092-007-0129-9.
[35] Trogdon T, Olver S, Deconinck B. Numerical inverse scattering for the Korteweg–de Vries and modified Korteweg–de Vries equations. Physica D 2012;241(11):1003–25. doi:10.1016/j.physd.2012.02.016.
[36] Matveev VB, Salle MA. Darboux transformations and solitons. Springer-Verlag Berlin Heidelberg New York London Paris Tokyo Hong Kong Barcelona Budapest; 1991. doi:101007/978-3-662-00922-2. ISBN 3-540-50660-8
[37] Gu C, Hu H, Zhou Z. Darboux transformations in integrable systems; vol. 26 of mathematical physics studies. Springer Science & Business Media; 2005. doi:101007/1-4020-3088-6.
[38] Crum MM. Associated Sturm–Liouville systems. Q J Math 1955;6(1):121–7. doi:10.1093/qmath/6.1.121.
[39] Levi D, Ragnisco O, Sym A. Dressing method vs. classical Darboux transformation. Il Nuovo Cimento B Series 11 1984;83(1):34–42. doi:10.1007/bf02723762.
[40] Kasman A. Glimpses of soliton theory: the algebra and geometry of nonlinear PDEs, vol 54. American Mathematical Soc; 2010. ISBN: 978-0-8218-5245-3.
[41] Higham NJ. Accuracy and stability of numerical algorithms, vol 80. 2 ed. SIAM; 2002. doi:10.1137/1.9780898718027. ISBN 0-89871-521-0
[42] Zettl A. Sturm-Liouville theory. No. 121 in Mathematical surveys and monographs. American Mathematical Soc.; 2005. doi:10.1090/surv/121. ISBN 978-1-4704-1348-4
[43] García-Ferrero MÁ, Gómez-Ullate D. Oscillation theorems for the Wronskian of an arbitrary sequence of eigenfunctions of Schrödinger's equation. Lett. Math. Phys. 2015;105(4):551–73. doi:10.1007/s11005-015-0751-4.
[44] Gesztesy F, Simon B, Teschl G. Spectral deformations of one-dimensional Schrödinger operators. J. d'Anal. Math. 1996;70(1):267–324. doi:10.1007/bf02820446.

[45] Wahlquist HD, Estabrook FB. Bäcklund transformation for solutions of the Korteweg–de Vries equation. Phys Rev Lett 1973;31(23):1386–90. doi:10.1103/physrevlett.31.1386.
[46] Levitan BM. Inverse Sturm-Liouville problems. Utrecht: VNU Science Press; 1987. doi:101515/9783110941937. ISBN 978-3-11-046031-5
[47] Samsonov BF. New possibilities for supersymmetry breakdown in quantum mechanics and second-order irreducible Darboux transformations. Phys Lett A 1999;263(4-6):274–80. doi:10.1016/s0375-9601(99)00736-7.
[48] Adler VÉ. A modification of Crum's method. Theor Math Phys 1994;101(3):1381–6. doi:10.1007/bf01035458.
[49] Karpman V, Maslov E. Perturbation theory for solitons. JETP 1977;73:537–59.
[50] Zakharov VE, Faddeev LD. Korteweg-de Vries equation: a completely integrable Hamiltonian system. Funct Anal Appl 1972;5(4):280–7. doi:10.1007/bf01086739.
[51] Boffetta G, Osborne AR. Computation of the direct scattering transform for the nonlinear Schroedinger equation. J Comput Phys 1992;102(2):252–64. doi:10.1016/0021-9991(92)90370-e.
[52] Trefethen LN. Is Gauss quadrature better than Clenshaw–Curtis? SIAM Rev 2008;50(1):67–87. doi:10.1137/060659831.