Reliable numerical algorithms for the Non-linear Fourier Transform of the KdV equation

Prins, Peter J.

DOI
10.4233/uuid:171ba94a-e8f4-4969-b6ed-912d4f334968

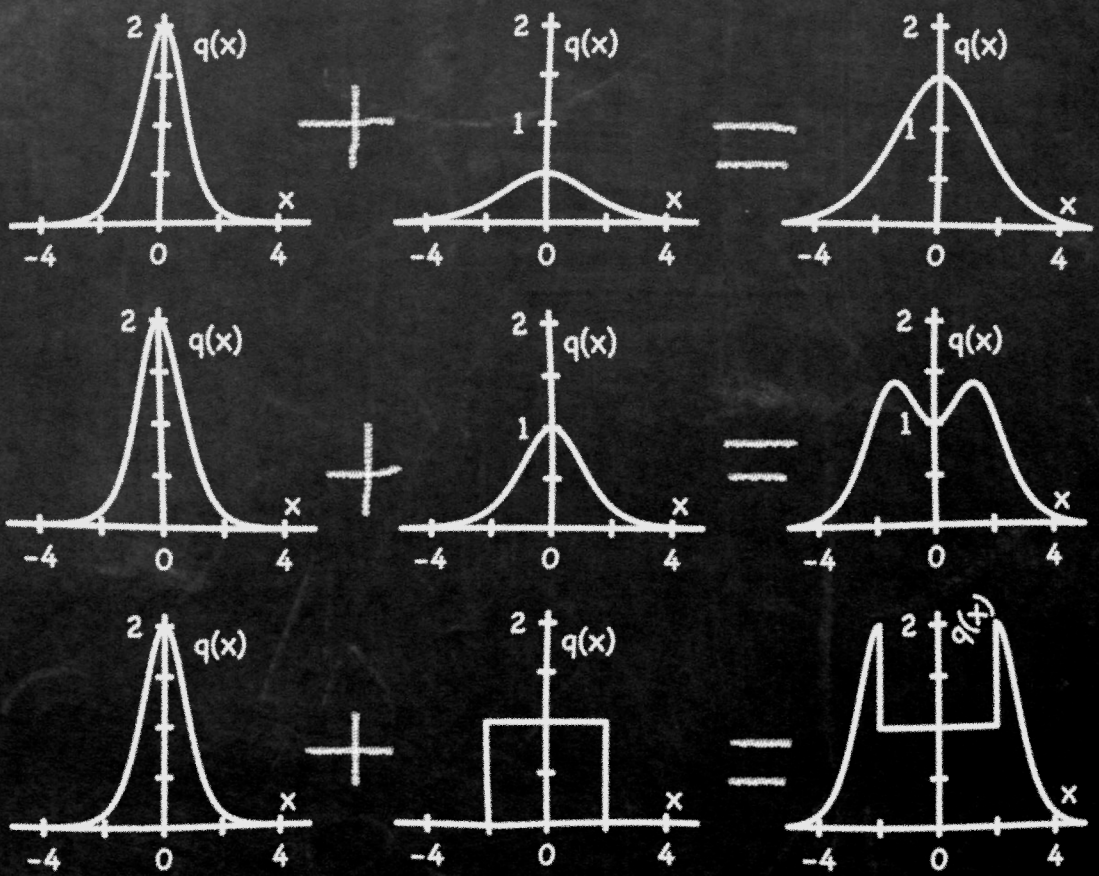
Publication date
2022

Document Version
Final published version

Citation (APA)
Prins, P. J. (2022). *Reliable numerical algorithms for the Non-linear Fourier Transform of the KdV equation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:171ba94a-e8f4-4969-b6ed-912d4f334968

Reliable numerical algorithms for the Non-linear Fourier Transform of the KdV equation

Peter J. Prins



Reliable numerical algorithms for the Non-linear Fourier Transform of the KdV equation

Reliable numerical algorithms for the Non-linear Fourier Transform of the KdV equation

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 28 april 2022 om 15:00 uur

door

Petrus Johannes PRINS

Master of Science in Electrical Engineering, Universiteit Twente, Nederland,
Master of Science in Systems and Control, Technische Universiteit Delft, Nederland,
geboren te Amsterdam, Nederland.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Dr.-Ing. S. Wahls,	Technische Universiteit Delft, promotor
Prof.dr.ir. M. Verhaegen,	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
Prof.dr. G. El,	Northumbria University, UK
Prof.dr. B. Deconinck,	University of Washington, USA
Prof.dr. J.M. Thijssen,	Technische Universiteit Delft
Prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft
Reservelid:	
Dr. M. Mazo Espinosa,	Technische Universiteit Delft

Dit project heeft van de Europese Onderzoeksraad (ERC) financiële middelen ontvangen uit hoofde van Horizon 2020 — het programma voor onderzoek en innovatie van de Europese Unie (subsidieovereenkomst nr. 716669).

Keywords: Signal processing algorithms, Non-linear Fourier Transform (NFT), Korteweg–de Vries equation (KdV), Schrödinger equation, Water wave, Soliton, Norming constant, Exponential splittings, Dressing method, Darboux transform, Crum transform.

Printed by: Drukkerij Haveka

Front & Back: Background: Chalkboard Background, Karin Dalziel (CC BY-NC 2.0). Foreground: 'Addition' of KdV-NFT spectra, Peter J. Prins. ul/ml/bl: $q(x) = 2\operatorname{sech}^2(x)$, um: $q(x) = \frac{1}{2}\operatorname{sech}^2\left(\frac{x}{2}\right)$, ur: $q(x) = \frac{3}{2}\operatorname{sech}^2\left(\frac{x}{2}\right)$, mm: $q(x) = \operatorname{sech}^2\left(\frac{x}{\sqrt{2}}\right)$, mr: $q(x) = \frac{1+\cosh(2x)+2\cosh(\sqrt{2}x)}{(\gamma_+\cosh(\gamma_- x)+\gamma_-\cosh(\gamma_+ x))^2}$, $\gamma_\pm = 1 \pm \frac{1}{\sqrt{2}}$, bm: $q(x) = [\![|x| < 2]\!]$, br: $q(x) = [\![|x| < 2]\!] + [\![|x| \geq 2]\!]\, 2\operatorname{sech}^2(|x| - 2)$, where u=upper, m=middle, b=bottom, l=left, r=right.

ISBN 978-94-6384-320-1

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Contents

Summary

Research question

The topic of this dissertation is the numerical computation of the forward and inverse Non-linear Fourier Transform (NFT) for the Korteweg–de Vries equation (KdV), for sampled signals that decay sufficiently fast on both sides. With NFTs certain non-linear Partial Differential Equations (PDEs) can be solved in a way that is analogous to solving linear Ordinary Differential Equations (ODEs) and PDEs by means of the ordinary Fourier transform. Similarly to the linear Fourier transform, NFTs can be used to analyse, synthesise, filter and predict signals. Existing numerical NFT algorithms suffer from either or both a limited accuracy or a long computation time, which limit the usability of the KdV-NFT for engineering problems. In this dissertation we develop new algorithms that achieve a higher accuracy or require a shorter computation time.

Design methods

We implemented existing numerical algorithms in Mathworks Matlab in floating point arithmetic to analyse their behaviour. Thereafter we designed new algorithms that avoid the undesirable behaviour of the existing algorithms. We demonstrated the improvements by means of benchmark tests. Furthermore we implemented some of the new algorithms in the programming language C in the Fast Non-linear Fourier Transform (FNFT) software library.

Results

We have developed algorithms to compute the continuous KdV-NFT spectrum and the eigenvalues and norming constants of the discrete KdV-NFT spectrum. Furthermore we developed an algorithm to compute the contribution of the discrete spectrum to the inverse KdV-NFT. The continuous KdV-NFT spectrum can now be computed with a fast algorithm at a comparable error tolerance to the Non-linear Schrödinger Equation (NSE)-NFT. That means that the computational complexity has been reduced from $O(D^2)$ to $O(D(\log(D))^2)$, where D is the number of samples, without a significant deterioration of the accuracy. The eigenvalues of the discrete KdV-NFT spectrum can now be computed reliably and more efficiently than before. The norming constants can now be computed in all known cases without the anomalous errors that were observed for older algorithms. That means an improvement of the accuracy by several orders of magnitude. The contribution of the inverse KdV-NFT can now be computed for discrete spectra with three to seven times as many eigenvalues in comparison to previously available algorithms.

Conclusions and applications

The KdV can be used as a model for nearly linear wave phenomena that propagate in one direction. These are found in a plethora of physical applications. The algorithms that we presented in this dissertation can be used for the analysis, synthesis, filtering and prediction of sampled data from such systems. Their higher accuracy and/or shorter computation time thus brings the KdV-NFT a step closer to the engineering practice.

Samenvatting

Onderzoeksvraag

Dit proefschrift gaat over de numerieke berekening van de voorwaartse en inverse niet-recht-evenredige Fourier transformatie (NFT) voor de Korteweg–De-Vriesvergelijking (KdV). Met NFTs kunnen bepaalde niet-recht-evenredige partiële differentiaalvergelijkingen opgelost worden op een wijze die vergelijkbaar is met het oplossen van recht evenredige gewone en partiële differentiaalvergelijkingen met behulp van de gewone Fourier transformatie. Evenals de gewone Fouriertransformatie kan een NFT gebruikt worden voor het analyseren, synthetiseren, filteren en voorspellen van signalen. Bestaande numerieke NFT algoritmen kennen een beperkte nauwkeurigheid en/of vergen een lange rekentijd, waardoor de bruikbaarheid van de KdV-NFT voor technische vraagstukken beperkt is. In dit proefschrift ontwikkelen we nieuwe algoritmen die een hogere mate van nauwkeurigheid realiseren danwel een kortere rekentijd vergen.

Ontwerpmethoden

We hebben bestaande numerieke algoritmen geïmplementeerd in Mathworks Matlab op basis van zwevendekommagetallen om het gedrag daarvan te analyseren. Vervolgens hebben we nieuwe algoritmen ontworpen die het onwenselijke gedrag van de bestaande algoritmen vermijden. De verbeteringen hebben we aangetoond door middel van benchmarkonderzoeken. Enkele van de ontwikkelde algoritmen hebben we tevens uitgevoerd in de programmeertaal C in de FNFT softwarebibliotheek.

Resultaten

we hebben algoritmen ontwikkeld om het continue KdV-NFT spectrum en de eigenwaarden en de normeringsconstanten van het discrete KdV-NFT spectrum te berekenen. Tevens hebben we een algoritme ontwikkeld om de bijdrage van het discrete spectrum aan de inverse KdV-NFT te berekenen. Het continue KdV-NFT spectrum kan nu worden berekend met een snel algoritme binnen een vergelijkbare foutmarge. Dat houdt in dat de complexiteitsgraad van $O\left(D^2\right)$ is teruggebracht tot slechts $O\left(D(\log(D))^2\right)$, waarin D staat voor het aantal bemonsteringen, zonder noemenswaardige verslechtering van de nauwkeurigheid. De eigenwaarden van het discrete KdV-NFT spectrum kunnen nu betrouwbaar berekend worden en efficiënter dan voorheen. De normeringsconstanten kunnen nu, voor zover bekend, in alle gevallen berekend worden zonder de abnormale fouten die in oudere algoritmen optraden. Dat betekent dat we de nauwkeurigheid hebben verbeterd met meerdere ordes van grootte. De bijdrage van de inverse KdV-NFT kan nu worden berekend voor discrete spectra met drie- tot zevenmaal zoveel eigenwaarden in vergelijking tot voorheen beschikbare algoritmen.

Conclusie en toepassingen

De KdV vergelijking kan gebruikt worden als model voor bijna recht evenredige golfverschijnselen die zich in één richting bewegen. Deze treft men aan in een veelheid aan praktische toepassingsgebieden. De algoritmen die we hebben gepresenteerd in dit proefschrift kunnen gebruikt worden voor het analyseren, synthetiseren, filteren en voorspellen van bemonsterde signalen die afkomstig zijn van dergelijke systemen. De hogere nauwkeurigheid en kortere rekentijd van deze algoritmen brengt de KdV-NFT derhalve een stap dichter bij de technische praktijk.

1 Introduction

'I believe I shall best introduce this phænomenon by describing the circumstances of my own first acquaintance with it. I was observing the motion of a boat which was rapidly drawn along a narrow channel by a pair of horses when the boat suddenly stopped — not so the mass of water in the channel which it had put in motion; it accumulated round the prow of the vessel in a state of violent agitation, then suddenly leaving it behind, rolled forward with great velocity, assuming the form of a large solitary elevation, a rounded, smooth and well-defined heap of water, which continued its course along the channel apparently without change of form or diminution of speed. I followed it on horseback and overtook it still rolling on at a rate of some eight or nine miles an hour, preserving its original figure some thirty feet long and a foot and a half in height. Its height gradually diminished, and after a chase of one or two miles I lost it in the windings of the channel. Such, in the month of August 1834, was my first chance interview with that singular and beautiful phænomenon which I have called the Wave of Translation, a name which it now very generally bears; which I have since found to be an important element in almost every case of fluid resistance, and ascertained to be the type of that great moving elevation of the sea, which, with the regularity of a planet, ascends our rivers and rolls along our shores.'

John Scott Russell [120]

1.1. The history of solitary waves and the Korteweg–de Vries equation (KdV)

The Scottish ship designer John Scott Russell was the first to recognise the unusual behaviour of a solitary wave, a single hump of water that travels in shallow water without changing its shape. His observation led him to conduct a series of experiments in a wave tank that he set up in his backyard. He found that the speed of a solitary wave is proportional to its amplitude. Unfortunately, the publication of his results [120] was not well received. Russell's

1

observations could not be explained by the wave equations that were known at that time. This led in particular the mathematical physicists George Biddell Airy and George Gabriel Stokes to dismiss Russells observations. [5, 74]

It took several decades before the appropriate mathematical model for Russell's observation was discovered. Although an equivalent representation was published first by Joseph Valentin Boussinesq [18] (as cited in [94, §2]) the model was named after Diederik Korteweg and Gustav de Vries as the Korteweg–de Vries equation (KdV)[1] [76]:

$$\frac{\partial q(x,t)}{\partial t} + 6\,q(x,t)\frac{\partial q(x,t)}{\partial x} + \frac{\partial^3 q(x,t)}{\partial x^3} = 0. \tag{1.1}$$

(Equation (1.1) is a normalised form of the Korteweg–de Vries equation (KdV) in which t represents the normalised time, x represents the normalised position with respect to a reference frame that moves with a certain constant speed and $q(x,t)$ represents the normalised free surface elevation. The normalisation can be found in Chapter 4.) Korteweg and De Vries assumed in their derivation that the fluid is incompressible and irrotational and that the waves are long-crested, long but not too tall with respect to the water depth, and travel in one direction [76]. Although the KdV is a non-linear Partial Differential Equation (PDE), Korteweg and De Vries found a set of solutions that were describable with an exact formula [5, Eq. (1.1.6)]:

$$q(x,t) = 2k^2 \operatorname{sech}^2\big(k(x - 4k^2 t) - \varphi\big), \tag{1.2}$$

where $\operatorname{sech}^2(\theta) := 4\left(\mathrm{e}^{-\theta} + \mathrm{e}^{\theta}\right)^{-2}$, φ is a real valued free parameter, and k is a positive free parameter. Equation (1.2) describes a wave with a constant symmetric shape with a single crest and no trough that moves at a constant speed: a *solitary wave*. This is illustrated in Fig. 1.1. The solitary wave solutions of the KdV were in perfect agreement with Russell's observations from half a century earlier: A single hump of water that travels without changing its shape, at a constant speed that has an affine relation to its amplitude. [74]

Again half a century later, Martin Kruskal and Norman Zabusky conducted simulations on a digital computer. They made new remarkable observations about the solitary waves of the KdV: If two spatially separated solutions of the form of (1.2) with a different amplitude are superposed such that the taller (and thus faster) wave can catch up with the smaller one, both waves re-emerge after the collision in their original shape. Furthermore, they[2] found that every other localized hump of water would split into one or more waves of the form of (1.2), plus some radiation. Because this behaviour reminded them of particles, they introduced the term *soliton*. [5, 74]

Shortly thereafter Gardner, Greene, Kruskal, and Miura [53] found a method to solve the initial value problem for the KdV: the *scattering transform*. (Later this transform would

[1] By the spelling rules of Dutch names, an infix like *de* must be capitalized whenever it is used without being preceded by another part of that same person's name, such as a first name [131]. Therefore, the correct spelling is Korteweg–De Vries equation (KDV), with an upper case D. Nevertheless, we will adopt the unfortunately firmly established lower case d in the name of the equation and its acronym.

[2] In [74] this second observation is attributed to both Kruskal and Zabusky, with reference to [162]. However, this observation is not reported in the latter. In [53], which was co-authored by Kruskal, the same observation seems to be attributed to an at that time unpublished work of Zabusky alone, without further specification.

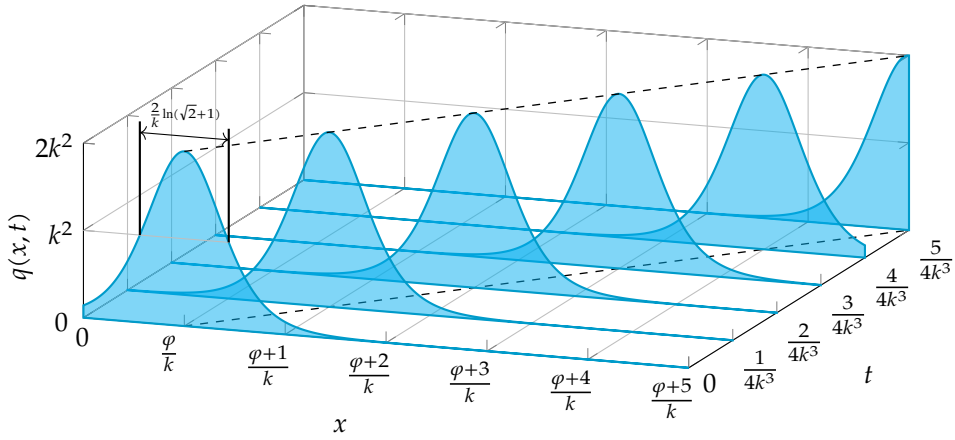


Figure 1.1: Schematic illustration of the solitary wave solution (1.2) of the KdV (1.1). The surface profile $q(x,t)$ is plotted at discrete time instants t. The soliton has an amplitude of $2k^2$, a wave length at half amplitude of $\frac{2}{k}\ln\left(\sqrt{2}+1\right) \approx \frac{1}{3k}$ at any fixed time instant t and travels at a constant speed of $4k^2$. The wave duration at half amplitude is $\frac{1}{2k^3}\ln\left(\sqrt{2}+1\right) \approx \frac{1}{12k^3}$ at any reference frame fixed position x (not shown). The parameter φ determines the offset with respect to the chosen (x,t) frame.

acquire its second name: the Non-linear Fourier Transform (NFT), which expresses the analogy of the method to the Fourier transform for linear PDEs and Ordinary Differential Equations (ODEs).) Their key observation was that the discrete eigenvalues λ of the Schrödinger equation,

$$\left(\frac{\partial^2}{\partial x^2} + q(x,t)\right) f(x,t) = \lambda f(x,t), \tag{1.3}$$

are constant if the potential $q(x,t)$ evolves according to the KdV. In analogy to the linear Fourier transform, they could compute from initial data $q(x,t_0)$ at any fixed time t_0 a spectrum of which the 'magnitude' part is constant and of which the 'phase' part evolves trivially as a function of the time t. The data $q(x,t_0)$ can then be retrieved from an integral equation that is known as the Gel'fand–Levitan–Marchenko (GLM) equation. A year later, Lax [81] found the underpinning mathematical structure that linked the KdV to the Schrödinger equation, a structure that explained the observation by Gardner, Greene, Kruskal, and Miura [53]. Lax found that the KdV could be written as [45, 81]

$$\frac{\partial \mathsf{L}(x,t)}{\partial t} + \mathsf{L}(x,t)\,\mathsf{A}(x,t) - \mathsf{A}(x,t)\,\mathsf{L}(x,t) = 0, \tag{1.4}$$

where the differential operators L and A are

$$\mathsf{L}(x,t) := \frac{\partial^2}{\partial x^2} + q(x,t), \tag{1.5}$$

$$\mathsf{A}(x,t) := -4\frac{\partial^3}{\partial x^3} - 6\,q(x,t)\frac{\partial}{\partial x} - 3\frac{\partial q(x,t)}{\partial x}. \tag{1.6}$$

Here it should be noted that the Schrödinger equation, (1.3), can be written as $\mathsf{L}(x,t)f(x,t) = \lambda f(x,t)$. If an equation can be written as a Lax pair (1.4) and solved by the NFT, we call it *(Lax-)integrable*. The KdV is not the only integrable PDE. Zakharov and Shabat [163] were the first to demonstrate that another physically significant PDE, the Non-linear Schrödinger Equation (NSE), is also integrable. Ablowitz, Kaup, Newell, and Segur [4] (*'the AKNS paper'*) developed a method to find a wide class of integrable equations and write their scattering transform in a unified way. [5, 74]

Despite early pioneering work on the numerical computation of NFTs by Ablowitz and Ladik [3], for a few decades the NFT remained mainly a tool that allowed mathematicians to find and analyse exact solutions of integrable PDEs. Around the start of the third millennium more researchers became interested in bringing the scattering transform to the engineering practice. To that end algorithms had to be developed that use sampled initial data $q(x,t_0)$ rather than functions [17, 19, 101, 102, 114, 140, 143, 160]. This enabled the use of the NFT as an analysis tool that reveals the physical structure of measured data from non-linear systems [23, 25, 103–105, 108, 135, 144, 166]. However, the early numerical methods were slow: Their computation times scaled quadratically in the number of samples. This motivated the development of Fast Non-linear Fourier Transforms (FNFTs) [150–152]. These scale almost linearly in the number of samples. Many recent publications, as well as this dissertation, present various improvements on aspects of the numerical NFT in computation time and accuracy, e.g. [29, 32, 50, 60, 90–92, 110–112, 132, 148].

1.2. Applications of the KdV equation

Like linear Fourier analysis applies to systems that are adequately modelled by a linear PDE or a linear ODE, NFT analysis applies to systems that are adequately modelled by the KdV. Thus, before we continue with a brief introduction to the KdV-NFT, we discuss which kind of systems can be modelled by the KdV. These systems have a few aspects in common with the shallow water wave system for which the KdV was first derived. These aspects regard both the physical setup and the state of the dynamics that appear therein.

1. The KdV describes a wave phenomenon in *one spatial dimension*. Therefore either there must be some kind of wave guide, such as a canal or natural guiding force, or the waves need to be planar. Furthermore the waves may only travel *forwards*. Physical parameters that influence the propagation of the wave (e.g. bathymetry, wave impedance) should remain the same along the direction of propagation.

2. The KdV typically arises in *nearly linear systems* that start to show non-linear behaviour as the amplitude of the wave phenomenon is increased.

3. The KdV describes relatively *long waves*. Even in the presence of some additional short waves, it remains an adequate model for long waves.

4. The KdV describes a *lossless* system. Therefore it does not apply to significantly lossy systems or to forced systems where energy is being added to the propagating waves.

Despite these limitations, the KdV was derived and/or experimentally verified for many applications. We mention some of these applications below.

1.2.1. Applications in fluid dynamics

Surface gravity waves in a fluid

Surface gravity waves are the weakly non-linear waves at a fluid's surface for which the KdV was first derived. As mentioned earlier in this introduction, the first experimental verification was [120]. Numerous studies thereafter have confirmed the applicability of the KdV, e.g. [26, 61, 62, 104, 108, 129, 139]. Although the KdV is a model for long waves, it was experimentally confirmed by Hammack and Segur [61] that the presence of disturbance in the form of shorter waves does not invalidate the KdV model, because these shorter wave components decay rapidly. The dynamics for which the KdV is the most adequate model may be found in canals and near shorelines. The KdV is frequently encountered in the literature as a model for tsunami waves. However, it has been questioned if the KdV is the most appropriate model in this situation, because the Earth's seas are too short to allow a tsunami wave to form solitons [35, 36, 57, 88].

Internal gravity waves in a stratified fluid

Stratified fluids consist of two or more layers that stay on top of each other without mixing. This can for example be due to a difference in chemical substance or temperature. Waves may occur at the boundary between the layers and these waves are known as internal gravity waves. The KdV is in some cases an adequate model for this type of waves [166]. The KdV has for example been proposed as a model for the internal waves of the thermocline along the equator[3] [34]. Research into internal waves in seas is amongst other things important for offshore platforms, because internal waves can reach damaging amplitudes, even while the surface waves are modest [37, §3].

Waves of an incompressible fluid in an elastic tube

The KdV has been derived for the dynamics of waves of an incompressible fluid in an elastic tube [79]. As such, the KdV has been proposed as a model for blood pressure waves in arteries [158].

Pressure waves in bubbly flows

If a fluid is mixed with small bubbles of gas, the result is a compressible substance. The (acoustic) pressure waves that can travel through this substance can be modelled by the KdV [78, 98, 155].

Conduits of buoyant fluid in a more viscous fluid

If a fluid is covered by a fluid of higher density and viscosity, it can buoyantly rise up. Thereby it forms blobs which are adequately described as solitons of the KdV [65]. This kind of transport of magma from the Earth's interior towards the surface is hypothesised as a cause of volcanic activity [37, §6].

1.2.2. Applications outside fluid dynamics

Non-linear electric transmission lines

Electric transmission lines can be designed to show the non-linear behaviour that is described by the KdV. These non-linear transmission lines have applications in electronic circuit design [119].

[3] The equator acts as a natural wave guide because of the Coriolis force due to the rotation of the earth.

Atmospheric Rossby waves

Rossby waves occur in the atmosphere of a planet due to its rotation. They are a major factor to the climate. These waves too exhibit behaviour that can be modelled by the KdV [117], [37, §7].

Plasma physics

The KdV has been derived as a model for hydromagnetic plasma waves [52] as well as for ion-acoustic waves in a cold plasma [154], [37, §8].

Non-linear ultrasonics

Accoustic pulses that travel through a crystalline gold layer were reported in [137] to be in excellent agreement with the KdV model.

Traffic flow

Besides many systems in physics, the KdV model appears to be applicable to certain human behaviour as well: The KdV was found to be a suitable model for the density of traffic flow [64, 96].

1.3. The Nonlinear Fourier Transform (NFT) method

Below, we give a brief introduction to the NFT method with respect to the KdV. The reader is referred to other sources for derivations and mathematical details, e.g. [5, 10, 43, 46, 79, 100, 128].

The KdV-NFT can be used to solve the KdV (1.1), subject to an initial condition $q(x, t_0)$: a spatial signal at a fixed time $t = t_0$. It cannot be used to solve the KdV (1.1), subject to a boundary condition $q(x_0, t)$: a time signal at a fixed position $x = x_0$. However, a problem for which the (1.1) is an appropriate model, can typically be described to the same accuracy by a KdV in which x and t change roles. See, e.g. [72, §15]. This other KdV is not mathematically equivalent, but describes a very similar evolution of $q(x, t)$. Naturally, the KdV-NFT can be applied to this other KdV to solve problems with a boundary condition $q(x_0, t)$. Hence depending on the purpose, in all that follows one could interpret x as the time coordinate and t as the space coordinate.

Among the ordinary Fourier methods a distinction can be made between Fourier series and the Fourier transform. The first applies to periodic input signals. That is, there exists a constant L such that $q(x + L, t_0) \equiv q(x, t_0)$. The second applies to vanishing input signals. That is, signals that decay sufficiently fast as $|x| \to \infty$. The same distinction can be made for NFT algorithms. In this dissertation we only look at algorithms for *vanishing input signals*. More precisely, these signals must satisfy

$$\int_{-\infty}^{\infty} |q(x, t_0)| \, (1 + |x|) \, \mathrm{d}x < \infty. \tag{1.7}$$

Signals that satisfy (1.7) belong to the *Faddeev class* [48]. For convenience we furthermore assume that

$$\lim_{|x| \to \infty} q(x, t_0) = 0. \tag{1.8}$$

The latter condition excludes some pathological signals that would otherwise complicate the exposition. In practice, when dealing with sampled input data, the input $q(x, t_0)$ is often truncated to some finite window $x \in [X_-, X_+]$, beyond which there is no data available. The truncated signal satisfies (1.8) trivially. Finally, we only consider *real valued* signals $q(x, t)$ in this dissertation. Complex valued signals $q(x, t)$ are meaningless in the majority (if not all) of the real world applications of the KdV and this restriction enables more efficient numerical algorithms.

As mentioned before, the KdV-NFT spectrum of a vanishing input signal is computed from the Schrödinger equation (1.3). We start by considering two sets of standard *trajectories* $f(x, t_0)$ that are defined by the Schrödinger equation and a set of boundary conditions at $x \to \pm\infty$ respectively. By virtue of (1.8) the Schrödinger equation reduces to

$$\frac{\partial^2}{\partial x^2} f(x, t_0) = \lambda f(x, t_0) \quad \text{as} \quad x \to \pm\infty. \tag{1.9}$$

Hence, we can define specific trajectories $f = \phi, f = \bar{\phi}, f = \bar{\psi}$ and $f = \psi$ that satisfy the respective boundary conditions

$$\begin{cases} \phi(x, t_0, \zeta) \to \exp(-\mathrm{j}\zeta x) \\ \bar{\phi}(x, t_0, \zeta) \to \exp(+\mathrm{j}\zeta x) \end{cases} \quad \text{as } x \to -\infty, \tag{1.10}$$

$$\begin{cases} \bar{\psi}(x, t_0, \zeta) \to \exp(-\mathrm{j}\zeta x) \\ \psi(x, t_0, \zeta) \to \exp(+\mathrm{j}\zeta x) \end{cases} \quad \text{as } x \to +\infty, \tag{1.11}$$

where $\zeta := \mathrm{j}\sqrt{\lambda}$ and $\mathrm{j} := \sqrt{-1}$. These particular trajectories are known as the *Jost solutions* and depend (for other x) on the input data through the Schrödinger equation. By Abel's identity [2, p. 22] the Wronskian of any two trajectories of (1.3) is independent of x. Thus, by evaluating the Wronskian of the Jost solutions, we can establish that the Jost solutions are linearly independent within both sets for all x, if and only if $\zeta \neq 0$:

$$\mathrm{W}[\phi(x, t_0, \zeta); \bar{\phi}(x, t_0, \zeta)] = \lim_{x \to -\infty} \begin{vmatrix} \phi(x, t_0, \zeta) & \bar{\phi}(x, t_0, \zeta) \\ \frac{\partial}{\partial x} \phi(x, t_0, \zeta) & \frac{\partial}{\partial x} \bar{\phi}(x, t_0, \zeta) \end{vmatrix} = 2\mathrm{j}\zeta, \tag{1.12}$$

$$\mathrm{W}[\bar{\psi}(x, t_0, \zeta); \psi(x, t_0, \zeta)] = \lim_{x \to \infty} \begin{vmatrix} \bar{\psi}(x, t_0, \zeta) & \psi(x, t_0, \zeta) \\ \frac{\partial}{\partial x} \bar{\psi}(x, t_0, \zeta) & \frac{\partial}{\partial x} \psi(x, t_0, \zeta) \end{vmatrix} = 2\mathrm{j}\zeta. \tag{1.13}$$

Since both sets of Jost solutions are linearly independent trajectories of the Schrödinger equation, a second order ODE, every trajectory of the Schrödinger equation can be parametrised as a linear combination of either of these sets. The particular linear combination that relates the two sets of Jost solutions defines the KdV-NFT spectrum. We write

$$\phi(x, t_0, \zeta) \equiv a(\zeta)\, \bar{\psi}(x, t_0, \zeta) + b(\zeta, t_0)\, \psi(x, t_0, \zeta), \tag{1.14}$$

$$\bar{\phi}(x, t_0, \zeta) \equiv \bar{b}(\zeta, t_0)\, \bar{\psi}(x, t_0, \zeta) + \bar{a}(\zeta)\, \psi(x, t_0, \zeta), \tag{1.15}$$

where the scattering parameters are defined as

$$a(\zeta) := \frac{1}{2\mathrm{j}\zeta} \mathrm{W}[\phi(x,t_0,\zeta);\psi(x,t_0,\zeta)], \tag{1.16}$$

$$b(\zeta,t_0) := \frac{1}{2\mathrm{j}\zeta} \mathrm{W}[\bar{\psi}(x,t_0,\zeta);\phi(x,t_0,\zeta)], \tag{1.17}$$

$$\bar{b}(\zeta,t_0) := \frac{1}{2\mathrm{j}\zeta} \mathrm{W}[\bar{\phi}(x,t_0,\zeta);\psi(x,t_0,\zeta)], \tag{1.18}$$

$$\bar{a}(\zeta) := \frac{1}{2\mathrm{j}\zeta} \mathrm{W}[\bar{\psi}(x,t_0,\zeta);\bar{\phi}(x,t_0,\zeta)]. \tag{1.19}$$

The KdV-NFT spectrum consists of two parts: A *continuous spectrum* and a *discrete spectrum*. The continuous spectrum is formed by the *reflection coefficient*

$$R(\zeta,t_0) := \frac{b(\zeta,t_0)}{a(\zeta)}, \tag{1.20}$$

where ζ is on the real line but not zero. The discrete spectrum consists of a countable number of *eigenvalues* ζ_n ($n \in \{1,2,\dots,N\}$) and corresponding *norming constants* $b(\zeta_n,t_0)$, defined by

$$\mathcal{D}(t_0) := \{(\zeta_n, b(\zeta_n,t_0)) \,|\, a(\zeta_n) = 0\}. \tag{1.21}$$

It can be shown that all eigenvalues are on the positive imaginary axis and that all eigenvalues are simple (i.e. their multiplicity cannot be higher than one) [79, p. 50–53]. Each eigenvalue in the spectrum is associated with a soliton. If that soliton appears separated from other wave components, then it satisfies (1.2) where $k = |\zeta_n|$ determines its speed, amplitude, length and duration.

The time propagation of the scattering parameters and the time propagation of the spectrum are easy to compute. The parameter $a(\zeta)$ is independent of time and $b(\zeta,t)$ evolves as [53], [5, §1.4]

$$b(\zeta,t) = b(\zeta,t_0)\exp\bigl(-(2\mathrm{j}\zeta)^3(t-t_0)\bigr). \tag{1.22}$$

This determines the time evolution of both the continuous spectrum $R(\zeta,t)$ and the discrete spectrum $\mathcal{D}(t)$. The norming constants are thus not constant in time, despite their name. Apart from (1.22), t can be regarded as a constant parameter in the forward and inverse NFT. Therefore it is often dropped from the notation. However, in this introduction we will write t_0 explicitly to show which variables depend on time, but can be regarded as a constant.

A few remarks are in place regarding different definitions of the KdV-NFT spectrum that appear in the literature. Firstly, when Gardner, Greene, Kruskal, and Miura [53] introduced the scattering transform, they used the variable b for the reflection coefficient and the variable a for the so-called transmission coefficient $T(\zeta) := 1/a(\zeta)$. The influential AKNS paper however used a and b as presented here [4, Appendix 3]. Therefore both conventions can be encountered in the literature. Secondly, in [4] both $\phi(x,t_0,\zeta)$ and $b(\zeta,t_0)$ are defined as above for the KdV, but with an opposite sign for all other PDEs. Thirdly, in the discrete

spectrum the norming constants $b(\zeta_n, t_0)$ are sometimes replaced by the residues of the (meromorphic continuation of the) reflection coefficient [5, §1.3]

$$c(\zeta_n, t_0) := \frac{b(\zeta_n, t_0)}{a'(\zeta_n)}, \quad \text{where} \quad a'(\zeta) := \frac{\mathrm{d}}{\mathrm{d}\zeta} a(\zeta). \tag{1.23}$$

Lastly, the reflection coefficient used here is the so-called right reflection coefficient. One could instead make use of the left reflection coefficient, which is obtained by expressing the Jost solution $\psi(x, t_0, \zeta)$ as a linear combination of $\phi(x, t_0, \zeta)$ and $\bar{\phi}(x, t_0, \zeta)$ instead of the other way around.

The inverse KdV-NFT is the opposite problem: Given a spectrum $(R(\zeta, t_0), \mathcal{D}(t_0))$, find the corresponding signal $q(x, t_0)$. This problem is mathematically described by the GLM integral equation. Because this part of the theory is not required within this dissertation, we refer the interested reader to e.g. [5, 40, 53]. In Chapter 5 a partial implementation of the inverse KdV-NFT is discussed, namely the *Crum transform*. By the Crum transform one can add eigenvalues to (or remove eigenvalues from) a signal with a known spectrum. With the knowledge that $(0, \emptyset)$ is the spectrum of the trivial signal $q(x, t_0) \equiv 0$, the Crum transform can thus be used to compute the inverse KdV-NFT for every spectrum of the form $(0, \mathcal{D}(t_0))$. Furthermore, the Crum transform can be used in conjunction with an algorithm that computes the signal for arbitrary spectra of the form $(R(\zeta, t_0), \emptyset)$ to become a complete inverse KdV-NFT algorithm. The Crum transform is further introduced in Chapter 5.

1.4. Numerical computation of the NFT

For numerical computations the Schrödinger equation (1.3) is typically rewritten as a *system of first order ODEs*. Thereto one defines an operator

$$\mathbf{V}(x, \zeta) := \begin{bmatrix} v_{11}(x, \zeta) + v_{12}(x, \zeta) \frac{\partial}{\partial x} \\ v_{21}(x, \zeta) + v_{22}(x, \zeta) \frac{\partial}{\partial x} \end{bmatrix}, \tag{1.24}$$

where

$$\begin{vmatrix} v_{11}(x, \zeta) & v_{12}(x, \zeta) \\ v_{22}(x, \zeta) & v_{22}(x, \zeta) \end{vmatrix} \neq 0, \tag{1.25}$$

such that (1.3) can be rewritten as

$$\frac{\partial}{\partial x} \boldsymbol{f}(x, t_0, \zeta) = \boldsymbol{A}(x, t_0, \zeta) \boldsymbol{f}(x, t_0, \zeta), \tag{1.26}$$

where the *vector valued trajectory* is

$$\boldsymbol{f}(x, t_0, \zeta) := \mathbf{V}(x, \zeta) f(x, t_0, \zeta). \tag{1.27}$$

The simplest choice for the operator $\mathbf{V}(x, \zeta)$ is

$$\mathbf{V}_{\mathrm{C}}(x, \zeta) := \begin{bmatrix} 1 \\ \frac{\partial}{\partial x} \end{bmatrix}. \tag{1.28}$$

Other common choices for the operator $\mathbf{V}(x,\zeta)$ can be found in Appendix 5.B.1. It is readily verified that

$$\boldsymbol{A}_{\mathrm{C}}(x,t_0,\zeta) = \begin{bmatrix} 0 & 1 \\ (\mathrm{j}\zeta)^2 - q(x,t_0) & 0 \end{bmatrix} \tag{1.29}$$

and that the boundary conditions for the Jost solutions can be written as

$$\boldsymbol{\Phi}_{\mathrm{C}}(x,t_0,\zeta) := \begin{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x,t_0,\zeta) & \bar{\boldsymbol{\phi}}_{\mathrm{C}}(x,t_0,\zeta) \end{bmatrix} \tag{1.30}$$

$$\to \begin{bmatrix} \exp(-\mathrm{j}\zeta x) & \exp(\mathrm{j}\zeta x) \\ -\mathrm{j}\zeta\exp(-\mathrm{j}\zeta x) & \mathrm{j}\zeta\exp(\mathrm{j}\zeta x) \end{bmatrix} \quad \text{as } x \to -\infty, \tag{1.31}$$

$$\boldsymbol{\Psi}_{\mathrm{C}}(x,t_0,\zeta) := \begin{bmatrix} \bar{\boldsymbol{\psi}}_{\mathrm{C}}(x,t_0,\zeta) & \boldsymbol{\psi}_{\mathrm{C}}(x,t_0,\zeta) \end{bmatrix} \tag{1.32}$$

$$\to \begin{bmatrix} \exp(-\mathrm{j}\zeta x) & \exp(\mathrm{j}\zeta x) \\ -\mathrm{j}\zeta\exp(-\mathrm{j}\zeta x) & \mathrm{j}\zeta\exp(\mathrm{j}\zeta x) \end{bmatrix} \quad \text{as } x \to +\infty. \tag{1.33}$$

By juxtaposing the Jost solutions in the matrices $\boldsymbol{\Phi}_{\mathrm{C}}(x,t_0,\zeta)$ and $\boldsymbol{\Psi}_{\mathrm{C}}(x,t_0,\zeta)$ respectively, (1.14) can be written compactly as

$$\boldsymbol{\Phi}_{\mathrm{C}}(x,t_0,\zeta) \equiv \boldsymbol{\Psi}_{\mathrm{C}}(x,t_0,\zeta)\,\boldsymbol{S}(\zeta,t_0), \tag{1.34}$$

where the *scattering matrix* is defined as

$$\boldsymbol{S}(\zeta,t_0) := \begin{bmatrix} a(\zeta) & \bar{b}(\zeta,t_0) \\ b(\zeta,t_0) & \bar{a}(\zeta) \end{bmatrix}. \tag{1.35}$$

Since the Jost solutions $\bar{\boldsymbol{\psi}}_{\mathrm{C}}(x,t_0,\zeta)$ and $\boldsymbol{\psi}_{\mathrm{C}}(x,t_0,\zeta)$ are linearly independent, the scattering parameters can be computed as

$$\boldsymbol{S}(\zeta,t_0) = \boldsymbol{\Psi}_{\mathrm{C}}^{-1}(x,t_0,\zeta)\,\boldsymbol{\Phi}_{\mathrm{C}}(x,t_0,\zeta) \tag{1.36}$$

at any position x (where $\cdot^{-1}$ denotes matrix inversion), or as

$$\begin{bmatrix} a(\zeta) \\ b(\zeta,t_0) \end{bmatrix} = \boldsymbol{\Psi}_{\mathrm{C}}^{-1}(x,t_0,\zeta)\,\boldsymbol{\Phi}_{\mathrm{C}}(x,t_0,\zeta)\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \boldsymbol{\Psi}_{\mathrm{C}}^{-1}(x,t_0,\zeta)\,\boldsymbol{\phi}_{\mathrm{C}}(x,t_0,\zeta) \tag{1.37}$$

to save computation time if the barred scattering parameters are not required.

To be able to evaluate (1.36) or (1.37), either or both of the sets of Jost solutions will have to be propagated to the same position x, the *matching point*, by numerically integrating the Schrödinger equation (1.26). The computational complexity of a numerical integrator typically scales linearly in D, the number of x samples. For the computation of the continuous spectrum $R(\zeta,t_0)$, the computation needs to be repeated for every desired ζ sample. If the number of ζ samples is chosen equal to D, the overall complexity of the computation of the continuous spectrum is thus $O(D^2)$. Wahls and Poor [150] introduced the FNFT. Instead of repeating the computation for every desired ζ sample, they first computed a polynomial in $\exp(\mathrm{j}\zeta)$ that approximates $R(\zeta,t_0)$. Thereafter they evaluated this polynomial for every desired sample of ζ. They showed that the overall this algorithm to compute the continuous spectrum could be implemented with a complexity of only $O\left(D\,(\log(D))^2\right)$. The

polynomial approximation can be obtained by means of exponential splitting, applied to the AKNS representation of the Schrödinger equation. The order of the exponential splitting scheme needs to be high enough to obtain a sufficiently accurate approximation. It turned out that computations for the KdV-FNFT are considerably more sensitive to inaccuracies in this polynomial approximation than computations for the FNFT for the NSE, for which the FNFT was first developed. Therefore in Chapter 2 a set of higher order splitting schemes is presented, which satisfies the specific requirements of the FNFT.

From (1.21) it can be seen that the computation of the eigenvalues in the discrete spectrum is a root finding problem: Find all ζ such that $a(\zeta) = 0$. In general this computation cannot be performed in closed form. Therefore one typically resorts to iterative root finding techniques, such as Newton–Raphson (NR). Iterative root finding has (for the purpose of the NFT) two difficulties: It is not known in advance how many roots there are to be found, and iterative root finders need for every root an initial guess that is within the basin of attraction of that root for that root finder. The FNFT [150] mitigates these difficulties by approximating $a(\zeta)$ as a polynomial in $\exp(\mathrm{j}\zeta)$. The roots of a polynomial can be found from a finite eigenvalue problem. However, as mentioned before, their method was primarily developed for the NSE. When the FNFT is adapted to the KdV, the resulting approximation of $a(\zeta)$ is unfortunately not a polynomial in $\exp(\mathrm{j}\zeta)$. (The approximation of $a(\zeta)$ also contains factors $(\mathrm{j}\zeta)^{\pm 1}$ outside an exponent.) Regarding the discrete spectrum the KdV has nonetheless two advantages over the NSE. Firstly, whereas the eigenvalues of the NSE can be anywhere in the upper half of the complex plane, those of the KdV are confined to the positive imaginary axis. Secondly, the Schrödinger equation is an example of a Sturm–Liouville (SL) equation. Therefore one can make use of SL oscillation theory to compute the number of roots in advance and to implement an iterative algorithm that finds each of them efficiently. Such an algorithm is presented in Chapter 3.

Once the set of eigenvalues $\{\zeta_n\}$ is known, it may seem like the computation of the norming constants $b(\zeta_n, t_0)$ is a matter of evaluating (1.37) for each of the eigenvalues. However, this computation is numerically highly unstable. For the NSE-NFT Aref [12] and Hari and Kschischang [63] developed a method to compute the norming constants in a way that cancels the dominant source of error. The main idea is to propagate the Jost solutions from both sides to a matching point $x = X_0$. It turns out that the norming constants can be computed to a much higher precision at that matching point, provided that the matching point is chosen well and that $a(\zeta_n) = 0$ is exploited to cancel certain terms in the computation. In Chapter 4 this method has been adapted to the KdV-NFT. Furthermore, a new and better criterion is presented to select the matching point X_0, which applies to the KdV-NFT as well as to the NSE-NFT. It is shown that with this criterion the norming constant can be computed without any significant error.

The forward KdV-NFT by itself can be used as a signal analysis tool for systems that are adequately described by the KdV. Often it is also desirable to be able to predict the future dynamics of such a system. Therefore one needs to be able to compute the inverse KdV-NFT. 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 discrete spectrum in a second stage. See, e.g., [40, Sect. 3], [28, Chap. xvii.3.2], or [122, Sect. 4.2]. We already mentioned thath this second stage is known as the Crum transform. The Crum transform can be computed in closed form, but already for a mod-

est number of eigenvalues it reaches the range and accuracy limits of floating point arithmetic. In Chapter 5 a numerical algorithm for the Crum transform is presented that carefully mitigates these sources of error. In comparison to the state of the art this algorithm can therefore process three to seven times as many eigenvalues in ordinary floating point arithmetic. The algorithm in Chapter 5 can be used as an inverse KdV-FNFT for spectra for which $R(\zeta, t_0) = 0$ for all ζ, or as the second stage of an inverse KdV-NFT algorithm when $R(\zeta, t_0) \neq 0$. The computation of the first stage, i.e. the inverse KdV-NFT of a spectrum of the form $(R(\zeta, t_0), \mathcal{D}(t_0)) = \left(R(\zeta, t_0) \neq 0, \emptyset\right)$ is discussed elsewhere, e.g., [28, 40, 122].

1.5. Outline of the dissertation

The topic of this dissertation is the numerical computation of the forward and inverse NFT for the KdV, for signals with vanishing boundary conditions. Existing algorithms suffer from either or both a limited accuracy or a long computation time. That limits the usability of the KdV-NFT for engineering problems. In this dissertation new algorithms are developed that achieve a higher accuracy or require a shorter computation time.

1.5.1. Chapter 2: Fast computation of the continuous KdV spectrum

The FNFT makes use of a polynomial approximation to speed up the computation of the NFT spectrum. This approximation is obtained from a numerical method that is known as exponential splitting. It appears that when the FNFT is (naively) applied to the KdV, the error in the spectrum due to this approximation is high in comparison to the FNFT for other integrable PDEs. Therefore in Chapter 2 we present new exponential splitting schemes that satisfy the specific requirements of the FNFT and lead to a higher order approximation. It is shown that this leads to a lower error in the computed spectrum. The presented schemes can be applied to the FNFT with respect to the KdV and other integrable PDEs, such as the NSE.

1.5.2. Chapter 3: Reliable computation of the eigenvalues of the discrete KdV spectrum

There exist two main approaches to compute the eigenvalues of the discrete KdV spectrum. The first is based on solving a matrix eigenproblem and the second, known as shooting, uses an integrator in combination with a root finder. Both approaches have strengths an weaknesses. In Chapter 3 we present a shooting algorithm that improves on existing shooting algorithms. A general purpose gradient based iterative root finding algorithm such as NR needs a suitable starting value for each eigenvalue. However, it is typically not even known exactly how many eigenvalues a spectrum contains.[4] On the other hand, a carefully designed specialized iterative root finder for SL equations can reliably compute the number of eigenvalues and find each of them. However, such root finders typically rely heavily on bisection and therefore require more iterations.

The shooting algorithm that we present in Chapter 3 combines the reliability of a root finder for SL equations with the lower number of iterations of NR. It is shown that the presented algorithm is more robust to numerical errors and therefore more reliable than other

[4] There do exist nevertheless bounds on the number of eigenvalues, see [127, §1 and §3].

shooting algorithms that are currently available. Furthermore it is shown that it typically saves computation time compared to the conventional bisection based approaches.

1.5.3. Chapter 4: Accurate computation of the norming constants of the discrete KdV spectrum

In theory, the norming constants of the discrete spectrum can be computed analogously to the continuous spectrum once the eigenvalues are known. However, this computation is numerically highly unstable. The norming constants can be computed to a much higher precision by making use of the bidirectional algorithm. This algorithm was originally developed for the NSE-NFT and in Chapter 4 it is adapted for the first time for the KdV-NFT. Furthermore, Chapter 4 presents a significant improvement to the bidirectional algorithm that also applies to other NFTs. That is, for the bidirectional algorithm the Jost solutions have to be propagated from both sides to a suitable matching point. Previously proposed criteria would not always result in a suitable matching point. In those cases the error in the computed norming constant would still be significant. In Chapter 4 a new criterion is proposed to select the matching point. In numerical examples, the new criterion is found to perform reliably even in cases where existing algorithms break down.

1.5.4. Chapter 5: Accurate inverse NFT computation for the discrete KdV spectrum

When computing the inverse KdV-NFT, the contribution of the discrete spectrum can be computed in closed form by making use of the Crum transform. However, a naive numerical implementation of the Crum transform quickly reaches the limitations of floating point arithmetic, both regarding the accuracy and the range of representable numbers. In Chapter 5 a numerical algorithm for the Crum transform is presented that carefully mitigates these sources of error. It is demonstrated that this new algorithm is considerably more accurate in floating point arithmetic than benchmark algorithms found in the literature: It can compute the inverse NFT at the same error tolerance when the discrete spectrum contains between three and seven times as many eigenvalues.

2

Fast computation of the continuous KdV spectrum

Non-linear Fourier Transforms (NFTs) enable the analysis of signals governed by certain non-linear evolution equations in a way that is analogous to how the conventional Fourier transform is used to analyse linear wave equations. Recently, fast numerical algorithms have been derived for the numerical computation of certain NFTs. In this chapter, we are primarily concerned with fast NFTs with respect to the Korteweg-de Vries equation (KdV), which describes e.g. the evolution of waves in shallow water. We find that in the KdV case, the fast NFT can be more sensitive to numerical errors caused by an exponential splitting. We present higher order splittings that reduce these errors and are compatible with the fast NFT. Furthermore we demonstrate for the NSE case that using these splittings can make the accuracy of the fast NFT match that of the conventional NFT.

2.1. Introduction

The Korteweg–de Vries equation (KdV) for a function $q = q(x,t)$,

$$q_t + 6qq_x + q_{xxx} = 0, \tag{2.1}$$

where the subscripts x and t denote partial derivatives, is a well known non-linear differential equation. It describes a large class of nearly hyperbolic mathematical systems, including water waves, lattice waves, and hydromagnetic and ion-acoustic waves in a plasma [70, 154, 161]. The initial value problem for the KdV can be solved with a Non-linear Fourier Transform (NFT) in a way that is similar to the way Fourier solved the heat equation with the linear Fourier transform [53]. Like the linear Fourier transform, NFTs can be used to analyse data. The KdV NFT has for example been used to analyse water waves [25, 33, 106].

Parts of this chapter have been published as P. J. Prins and S. Wahls. "Higher Order Exponential Splittings for the Fast Non-Linear Fourier Transform of the Korteweg-De Vries Equation". In: *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2018, pp. 4524–4528. DOI: 10.1109/ICASSP.2018.8461708,

(NFTs are also known as scattering transforms in the literature.) Another evolution equation that can be solved using NFTs is the Non-linear Schrödinger Equation (NSE) [4, 81, 163]. It has recently attracted attention for applications in fibre-optic communication [145]. Unfortunately, most naive numerical implementations of NFTs have a computational complexity of at least $O(D^2)$, making it unattractive for large numbers of samples D. For the NSE, recently a Fast Non-linear Fourier Transform (FNFT), an algorithm with a computational complexity of $O(D\log^2 D)$, has been introduced [150].[1] Formally, the approach used for the NSE can also be used to obtain an FNFT for the KdV. However, we observed unexpectedly large numerical errors when we tried this. The FNFT algorithm requires an exponential splitting to become fast. (The matrix exponential $\mathrm{e}^{\boldsymbol{A}} \equiv \exp(\boldsymbol{A}) := \sum_{k=0}^{\infty} \boldsymbol{A}^k/k!$ in general does not satisfy $\exp((\boldsymbol{A}+\boldsymbol{B})\varepsilon) = \exp(\boldsymbol{A}\varepsilon)\exp(\boldsymbol{B}\varepsilon)$. Exponential splittings approximate $\exp((\boldsymbol{A}+\boldsymbol{B})\varepsilon)$ up to an error of $O(\varepsilon^{n+1})$, where n is the order of accuracy.) We found that removing the exponential splitting made these errors disappear. A splitting with a higher order of accuracy can reduce the error, but most spittings known in the literature unfortunately do not have the special structure that is needed for the FNFT. In this chapter, we therefore present higher order splittings that have this structure and can thus be used to obtain FNFTs with reduced errors. More precisely, we are interested in n-th order accurate exponential splitting schemes of the general form

$$\mathrm{e}^{(\boldsymbol{A}+\boldsymbol{B})\varepsilon} = \sum_{k=0}^{k_{\max}} \gamma_k \prod_{j=0}^{j_{\max,k}} \mathrm{e}^{\alpha_{j,k}\boldsymbol{A}\varepsilon}\mathrm{e}^{\beta_{j,k}\boldsymbol{B}\varepsilon} + O\left(\varepsilon^{n+1}\right). \tag{2.2}$$

As will be explained in Section 2.3, if all $\alpha_{j,k}$ (or all $\beta_{j,k}$) are rational numbers, we obtain a suitable form for the FNFT. Two widely used splittings that fulfil these conditions are the symmetrically weighted sequential splitting and the symmetric Strang splitting, which are both second order accurate [133, 134]. Suzuki [136] presented an algorithm to find the parameters for splittings with any desired order of accuracy, of the form of (2.2) with $k_{\max} = 0$, but this algorithm results in irrational or even complex coefficients and is not suitable for our purpose. We remark that the stability of a splitting scheme can be guaranteed if all the parameters $\alpha_{j,k}$, $\beta_{j,k}$ and γ_k in (2.2) are real and non-negative. Unfortunately, then the order of accuracy is two at most [58, 130]. Higher order schemes may be stable, but that needs to be determined on a per application basis.

The chapter is structured as follows. In Section 2.2, we summarise the part of the theory behind the (F)NFTs that is relevant for the purpose of this paper. In Section 2.3, we establish a sufficient condition for a splitting to be suitable for the FNFT and present several higher order exponential splitting schemes that fulfil this condition. We demonstrate the value of these splitting schemes numerically in Section 2.4. The paper is concluded in Section 2.5.

2.2. Preliminaries

In this section we describe the numerical calculation of the reflection coefficient of the Schrödinger equation. This is an intermediate step in the calculation of the NFT of a potential $q(x;t_0)$ that evolves according to a suitable non-linear differential equation, like the KdV or the NSE. We will omit the dependence on the fixed time t_0.

[1] To be precise: This complexity holds for the so-called reflection coefficient when $\zeta \in \mathbb{R}$ (see (2.3)), the case we consider in this chapter.

The following differential equation is the basis for the (F)NFT:

$$v_x(x,\zeta) = \boldsymbol{C}(x,\zeta)\, v(x,\zeta) = \begin{bmatrix} -\mathrm{j}\zeta & q(x) \\ r(x) & \mathrm{j}\zeta \end{bmatrix} v(x,\zeta) \text{ [4]}, \tag{2.3}$$

where j is the imaginary unit. For the KdV the boundary condition

$$\lim_{x\to-\infty} v(x,\zeta)\exp(\mathrm{j}\zeta x) = \begin{bmatrix} 2\mathrm{j}\zeta \\ 1 \end{bmatrix} \tag{2.4}$$

is used and $r(x) \equiv -1$ is chosen in (2.3) [4].[2] This choice reduces (2.3) to $v_{2xx} + (\zeta^2 + q(x))v_2 = 0$: the Schrödinger equation associated with (2.1) [4]. In this standard form of the KdV all variables have been made unit-less by normalization. Note that (2.3) is a generalised scattering problem [22, Sec. 6]. Similar problems appear in many other signal processing applications [21].

The NFT consists of multiple parts, of which we only need the so-called reflection coefficient $R(\zeta)$ in this chapter. It is defined in terms of the solution of (2.3) and (2.4), denoted as

$$v(x,\zeta) \to \frac{1}{T(\zeta)} \begin{bmatrix} 2\mathrm{j}\zeta \mathrm{e}^{-\mathrm{j}\zeta x} \\ \mathrm{e}^{-\mathrm{j}\zeta x} + R(\zeta)\mathrm{e}^{\mathrm{j}\zeta x} \end{bmatrix} \text{ as } x \to \infty \quad \text{[4]}. \tag{2.5}$$

Hence,

$$R(\zeta) = \lim_{x\to\infty} \left(2\mathrm{j}\zeta \frac{v_2(x,\zeta)}{v_1(x,\zeta)} - 1 \right) \mathrm{e}^{-2\mathrm{j}\zeta x}. \tag{2.6}$$

The quantity $T(\zeta)$ is called the transmission coefficient.

As in for example [101], we apply two approximations to calculate (2.6) numerically:

1. The interval $x \in (-\infty, \infty)$ is replaced by $x \in [L_-, L_+]$, with $L_\pm$ 'close' to $\pm\infty$. The approximation is exact if $q(x) = 0$ for all $x \notin [L_-, L_+]$.

2. The potential $q(x)$ is approximated by a piecewise constant function, a staircase $\hat{q}(x)$. That is, the interval $[L_-, L_+]$ is divided into D subintervals of width $\varepsilon = (L_+ - L_-)/D$, and in each of these subintervals $\hat{q}(x) = q_i := q(x_i)$, where x_i is the midpoint of the i-th subinterval. (In the general case, the same is done for $r(x)$. For the KdV it is already constant.)

Because of Approximation 2, (2.3) becomes a linear differential equation for each subinterval, which is solved as

$$v(x_i + \varepsilon/2, \zeta) = \boldsymbol{G}(x_i, \zeta)\, v(x_i - \varepsilon/2, \zeta), \tag{2.7}$$

where

$$\boldsymbol{G}(x_i, \zeta) := \mathrm{e}^{\boldsymbol{C}(x_i,\zeta)\varepsilon}. \tag{2.8}$$

[2] We remark that the NSE is obtained for the boundary condition $\lim_{x\to-\infty} v(x,\zeta)\exp(\mathrm{j}\zeta x) = [1\ 0]^\top$ and $r(x) \equiv \pm q^*(x)$, where the star denotes the complex conjugate [4].

Continuity of $\boldsymbol{v}(x,\zeta)$ at each of the boundaries of the subintervals allows us to write

$$\boldsymbol{v}(L_+,\zeta) = \boldsymbol{H}(\zeta)\,\boldsymbol{v}(L_-,\zeta), \tag{2.9}$$

where

$$\boldsymbol{H}(\zeta) \coloneqq \boldsymbol{G}(x_D,\zeta)\,\boldsymbol{G}(x_{D-1},\zeta)\cdots\boldsymbol{G}(x_2,\zeta)\,\boldsymbol{G}(x_1,\zeta). \tag{2.10}$$

We substitute the result of (2.9) with the boundary condition $\boldsymbol{v}(L_-,\zeta) = \left[\begin{smallmatrix}2\mathrm{j}\zeta\\1\end{smallmatrix}\right]\exp(-\mathrm{j}\zeta L_-)$ in (2.6), to approximate the reflection coefficient as[3]

$$\hat{R}(\zeta) = \left(2\mathrm{j}\zeta\,\frac{2\mathrm{j}\zeta\,H_{21}(\zeta)+H_{22}(\zeta)}{2\mathrm{j}\zeta\,H_{11}(\zeta)+H_{12}(\zeta)} - 1\right)\mathrm{e}^{-2\mathrm{j}\zeta L_+}. \tag{2.11}$$

The calculation of $\boldsymbol{H}(\zeta)$ according to (2.10) requires $D-1$ matrix multiplications for each value of ζ. To reduce the computational complexity, Wahls and Poor [150, 151] proposed to approximate the entries of $\boldsymbol{G}(x_i,\zeta)$ by rational functions with respect to a transformed variable $z(\zeta)$. Then, after evaluation of (2.10) with fast tree-wise polynomial multiplication, the entries of $\boldsymbol{H}(\zeta)$ are rational approximations as well, which can be evaluated using a fast polynomial evaluation method for each desired value of ζ. This is the idea behind the FNFT. In this chapter, we express this third approximation as follows:

3. Find a rational approximation

$$\begin{aligned}\hat{\boldsymbol{G}}(x_i,\zeta) &= \sum_{k=-m}^{m} \boldsymbol{M}_{k+m}(q_i,r_i)\,(z(\zeta))^k \\ &= (z(\zeta))^{-m}\sum_{k=0}^{2m}\boldsymbol{M}_k(q_i,r_i)\,(z(\zeta))^k,\end{aligned} \tag{2.12}$$

for some set of matrix valued coefficients $\boldsymbol{M}_k(q_i,r_i)$, and substitute it in (2.10) to find

$$\hat{\boldsymbol{H}}(\zeta) = (z(\zeta))^{-d}\sum_{k=0}^{2d}\boldsymbol{N}_k(q_i,r_i)\,(z(\zeta))^k. \tag{2.13}$$

This result can be used in (2.11).

How can a rational approximation like (2.12) be obtained? — Feced, Zervas, and Muriel [49] simplified (2.8) by applying a symmetric Strang splitting (see (2.18)) with $\boldsymbol{C}(x_i,\zeta) = \boldsymbol{A}(\zeta)+\boldsymbol{B}(x_i)$, where

$$\boldsymbol{A}(\zeta) \coloneqq \begin{bmatrix}-\mathrm{j}\zeta & 0\\ 0 & \mathrm{j}\zeta\end{bmatrix};\quad \boldsymbol{B}(x_i) \coloneqq \begin{bmatrix}0 & q(x_i)\\ r(x_i) & 0\end{bmatrix}. \tag{2.14}$$

The result of their approximation, which expresses $\boldsymbol{G}(x_i,\zeta)$ as a product of matrices that depend either only on x_i or only on ζ, is rational in $z(\zeta) \coloneqq \exp(-\mathrm{j}\zeta\epsilon/2)$ and can be written like (2.12). This can for example be seen from Lemma 1 in Section 2.3.

[3] While the paper [110] on which this chapter is based was being reviewed we noted that we could use $\bar{R}(-\zeta) = R(\zeta)$ in [4, p. 299], in order to obtain an alternative to (2.11): $\hat{R}(\zeta) = H_{12}(-\zeta)\,(2\mathrm{j}\zeta\,H_{11}(-\zeta)-H_{12}(-\zeta))^{-1}\,\mathrm{e}^{-2\mathrm{j}\zeta L_+}$. This equation appears to be less sensitive to numerical errors in $\boldsymbol{H}(\zeta)$ than (2.11).

2.3. Higher order splittings

Modification of the NSE FNFT for the KdV seems straightforward: Just use $r(x_i) \equiv -1$ instead of $r(x_i) \equiv \pm q^*(x_i)$ in (2.14), as described in Section 2.2. However, as we will see in the numerical examples in Section 2.4, then the error in $\hat{R}$ caused by Approximation 3 can be much higher than what one would expect from the NSE case.[3] To reduce this error (without reducing the step size ε), we want to use splitting schemes with a higher order of accuracy. Lemma 1 below states that a splitting scheme of the form of (2.2) is suitable for the FNFT if all parameters $\alpha_{j,k}$ are positive rational numbers. (We will omit the dependencies on ζ and x_i from here.)

Lemma 1 *If $\boldsymbol{G}$ is approximated by a splitting scheme of the form of (2.2) with $\boldsymbol{A}$ and $\boldsymbol{B}$ defined in (2.14) and $\sum_j \alpha_{j,k} = 1\ \forall k$ and $0 < \alpha_{j,k} \in \mathbb{Q}\ \forall j,k$, then the approximation can be written as (2.12) with $2m \in \mathbb{N}$.*

Proof Write $\alpha_{j,k}$ for all j and k as an irreducible fraction. Let $2m$ be the least common multiple of their denominators and rewrite

$$\mathrm{e}^{\alpha_{j,k}\boldsymbol{A}\varepsilon} = \mathrm{e}^{-\mathrm{j}\zeta\varepsilon\alpha_{j,k}} \begin{bmatrix} 1 & 0 \\ 0 & \exp(\mathrm{j}\zeta\varepsilon/m) \end{bmatrix}^{2m\alpha_{j,k}}. \tag{2.15}$$

Define $z := \exp(\mathrm{j}\zeta\epsilon/m)$. Equation (2.2) should hold for any arbitrary step size ε. For $\epsilon = 0$ it yields $\sum_k \gamma_k = 1$, so substitution of (2.15) in (2.2) results in

$$\hat{\boldsymbol{G}} = z^{-m} \sum_{k=0}^{k_{\max}} \gamma_k \prod_{j=0}^{j_{\max,k}} \begin{bmatrix} 1 & 0 \\ 0 & z^{2m\alpha_{j,k}} \end{bmatrix} \mathrm{e}^{\beta_{j,k}\boldsymbol{B}\varepsilon}. \tag{2.16}$$

Since $2m\alpha_{j,k} \in \mathbb{N}\ \forall j,k$, (2.16) can be written as (2.12). ∎

The following splitting schemes are suitable for the FNFT, because each of them fulfils the requirements of Lemma 1. The same holds for the dual schemes that can be obtained by changing every $\boldsymbol{A}$ to a $\boldsymbol{B}$ and vice versa.

$$\mathrm{e}^{\boldsymbol{C}\varepsilon} = \mathrm{e}^{\boldsymbol{A}\varepsilon}\mathrm{e}^{\boldsymbol{B}\varepsilon} + O(\varepsilon^2); \tag{2.17}$$

$$\mathrm{e}^{\boldsymbol{C}\varepsilon} = \mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon}\mathrm{e}^{\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon} + O(\varepsilon^3); \tag{2.18}$$

$$\mathrm{e}^{\boldsymbol{C}\varepsilon} = \tfrac{9}{8}\mathrm{e}^{\frac{1}{3}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{2}{3}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{2}{3}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{1}{3}\boldsymbol{B}\varepsilon} - \tfrac{1}{8}\mathrm{e}^{\boldsymbol{A}\varepsilon}\mathrm{e}^{\boldsymbol{B}\varepsilon} + O(\varepsilon^4); \tag{2.19}$$

$$\mathrm{e}^{\boldsymbol{C}\varepsilon} = \tfrac{4}{3}\mathrm{e}^{\frac{1}{4}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{4}\boldsymbol{A}\varepsilon} - \tfrac{1}{3}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon}\mathrm{e}^{\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon} + O(\varepsilon^5); \tag{2.20}$$

$$\mathrm{e}^{\boldsymbol{C}\varepsilon} = \tfrac{625}{384}\mathrm{e}^{\frac{1}{5}\boldsymbol{A}\varepsilon}\left(\mathrm{e}^{\frac{2}{5}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{2}{5}\boldsymbol{A}\varepsilon}\right)^2\mathrm{e}^{\frac{1}{5}\boldsymbol{B}\varepsilon} - \tfrac{81}{128}\mathrm{e}^{\frac{1}{3}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{2}{3}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{2}{3}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{1}{3}\boldsymbol{B}\varepsilon} + \ldots \tfrac{1}{192}\mathrm{e}^{\boldsymbol{A}\varepsilon}\mathrm{e}^{\boldsymbol{B}\varepsilon} + O(\varepsilon^6); \tag{2.21}$$

$$\mathrm{e}^{\boldsymbol{C}\varepsilon} = \tfrac{81}{40}\mathrm{e}^{\frac{1}{6}\boldsymbol{A}\varepsilon}\left(\mathrm{e}^{\frac{1}{3}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{3}\boldsymbol{A}\varepsilon}\right)^2\mathrm{e}^{\frac{1}{3}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{6}\boldsymbol{A}\varepsilon} - \tfrac{16}{15}\mathrm{e}^{\frac{1}{4}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{4}\boldsymbol{A}\varepsilon} + \ldots \tfrac{1}{24}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon}\mathrm{e}^{\boldsymbol{B}\varepsilon}\mathrm{e}^{\frac{1}{2}\boldsymbol{A}\varepsilon} + O(\varepsilon^7); \tag{2.22}$$

$$e^{C\varepsilon} = \frac{117649}{46080} e^{\frac{1}{7}A\varepsilon}\left(e^{\frac{2}{7}B\varepsilon}e^{\frac{2}{7}A\varepsilon}\right)^3 e^{\frac{1}{7}B\varepsilon} - \frac{15625}{9216} e^{\frac{1}{5}A\varepsilon}\left(e^{\frac{2}{5}B\varepsilon}e^{\frac{2}{5}A\varepsilon}\right)^2 e^{\frac{1}{5}B\varepsilon} + \dots \quad (2.23)$$
$$\frac{729}{5120} e^{\frac{1}{3}A\varepsilon}e^{\frac{2}{3}B\varepsilon}e^{\frac{2}{3}A\varepsilon}e^{\frac{1}{3}B\varepsilon} - \frac{1}{9216} e^{A\varepsilon}e^{B\varepsilon} + O(\varepsilon^8);$$

$$e^{C\varepsilon} = \frac{1024}{315} e^{\frac{1}{8}A\varepsilon}\left(e^{\frac{1}{4}B\varepsilon}e^{\frac{1}{4}A\varepsilon}\right)^3 e^{\frac{1}{4}B\varepsilon}e^{\frac{1}{8}A\varepsilon} - \frac{729}{280} e^{\frac{1}{6}A\varepsilon}\left(e^{\frac{1}{3}B\varepsilon}e^{\frac{1}{3}A\varepsilon}\right)^2 e^{\frac{1}{3}B\varepsilon}e^{\frac{1}{6}A\varepsilon} + \dots \quad (2.24)$$
$$\frac{16}{45} e^{\frac{1}{4}A\varepsilon}e^{\frac{1}{2}B\varepsilon}e^{\frac{1}{2}A\varepsilon}e^{\frac{1}{2}B\varepsilon}e^{\frac{1}{4}A\varepsilon} - \frac{1}{360} e^{\frac{1}{2}A\varepsilon}e^{B\varepsilon}e^{\frac{1}{2}A\varepsilon} + O(\varepsilon^9).$$

Equations (2.17) and (2.18) are well-known, as sequential splitting and symmetric Strang splitting [134] respectively. Equation (2.19) is reported in [27]. Equation (2.20) is known as the Strang-Richardson scheme (see e.g. [41]). We remark that only the symmetric Strang splitting has been used for FNFTs so far. We could not find the other schemes in the literature, so they may be new. We derived these as follows. We imposed the number of terms $k_{\max}$ and factors $j_{\max,k}$ and the order of accuracy n in (2.2). In case n is even we set $\beta_{j_{\max,k},k} = 0\ \forall k$. (In words: Every term in an even order accurate splitting has to start and end with an exponential of $\boldsymbol{A}$.) Then we replaced every matrix exponential by its Taylor series expansion and used algebraic computational software (Wolfram Mathematica) to find the parameter values $\alpha_{j,k}$, $\beta_{j,k}$ and γ_k by equating the appropriate left and right hand side terms. That is, terms that contain the same product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. These values appear to be unique, with respect to the particular choices for $k_{\max}$, $j_{\max,k}$ and n corresponding to (2.17) to (2.24) respectively. We do not know whether these splittings are part of a family that extends up to arbitrary order of accuracy; neither do we have direct formulas for the parameters.

Remark 1 Equations (2.12) and (2.16) for $2m \in \mathbb{N}$ are rational in $\sqrt{z}$, because the denominator z^m is a positive integer power of $\sqrt{z}$. Yet, the numerator is a polynomial in z (with matrix-valued coefficients). □

Remark 2 If the conditions for Lemma 1 hold, except for the positivity of $\alpha_{j,k}$, we can still obtain a rational approximation that is suitable for the FNFT. The constructive proof becomes more complicated in that case and since Lemma 1 suffices for the purpose of this chapter, we favoured the less general, but clearer result. □

Remark 3 The constructive proof of Lemma 1 is based on finding a number m such that $2m\alpha_{j,k} \in \mathbb{Z}\ \forall j,k$. One could hypothesise that such a number m may still exist when the condition $\alpha_{j,k} \in \mathbb{Q}$ does not hold. However, if $2m\alpha_{j,k} \in \mathbb{Z}\ \forall j,k$ and $\sum_j \alpha_{j,k} = 1$ it follows that $2m \in \mathbb{Z}$ and $\alpha_{j,k} \in \mathbb{Q}\ \forall j,k$: $1 = \sum_j \alpha_{j,k} \Rightarrow 2m = \sum_j 2m\alpha_{j,k} \in \mathbb{Z} \Rightarrow \alpha_{j,k} \in \mathbb{Q}\ \forall j,k$. □

2.4. Numerical examples

In this section we demonstrate the application of the proposed splitting schemes for two potential functions $q(x)$ for the KdV. As a comparison, we also include an example for the NSE. For these examples the reflection coefficient is known analytically; the respective equations provide the ground truth. For the KdV examples, we divide the interval $x \in [-16, 16]$ in $D = 256$ sections of width $\varepsilon = 1/8$ and sample $q(x)$ at the midpoints of each section to obtain a staircase approximation. (The interval is large enough to make Approximation 1

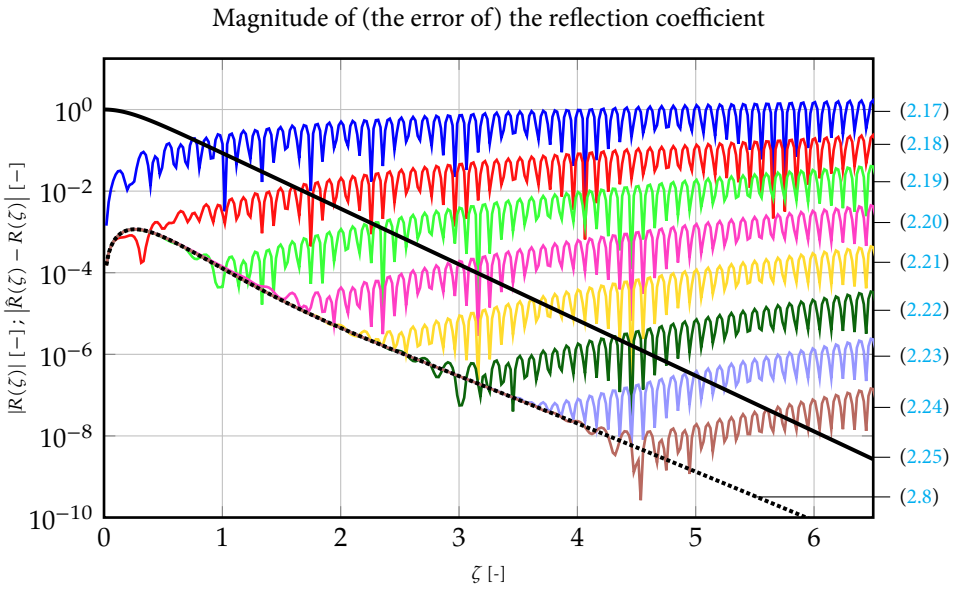


Figure 2.1: The black solid curve with label (2.25) shows the magnitude of the reflection coefficient of a squared hyperbolic secant potential for the KdV. All other other curves show the magnitude of the *error* of a numerical approximation of the reflection coefficient. The numbers between brackets refer to the corresponding equation: (2.8): error without splitting (due to staircase approximation), (2.17)–(2.24): error with splitting.

negligible for the chosen potentials.) We apply each of the proposed splitting schemes ((2.17) to (2.24)) as well as the un-split matrix exponential ((2.8)), and calculate the error of the approximated reflection coefficient (compared to the ground truth) with (2.10) and (2.11). We omit the transmission coefficient as well as the dual splitting schemes with the roles of $\boldsymbol{A}$ and $\boldsymbol{B}$ reversed, but we remark that the shown results are representative.

2.4.1. KdV with a squared hyperbolic secant potential

Consider the potential function $q(x) = \mathring{q}\,\mathrm{sech}^2(x)$ with $\mathring{q} = 9$. The reflection coefficient for this function is given by

$$R(\zeta) = \frac{\cos(\pi\delta)}{\pi} \cdot \frac{\Gamma(\mathrm{j}\zeta)}{\Gamma(-\mathrm{j}\zeta)} \Gamma\left(\tfrac{1}{2} - \mathrm{j}\zeta + \delta\right) \Gamma\left(\tfrac{1}{2} - \mathrm{j}\zeta - \delta\right), \tag{2.25}$$

where $\delta := \sqrt{\mathring{q} + 1/4}$ and Γ is the gamma function [79].

The magnitude of this reflection coefficient is shown in Fig. 2.1, as well as the magnitude of the error for the various numerical approximations. We see that the error due to Approximation 2 is below approximately 1% for each value of ζ. Above some frequency, Approximation 3 becomes the dominant error source for the splitting schemes. One can reduce this error and increase this frequency by choosing a splitting scheme with a higher order of accuracy.

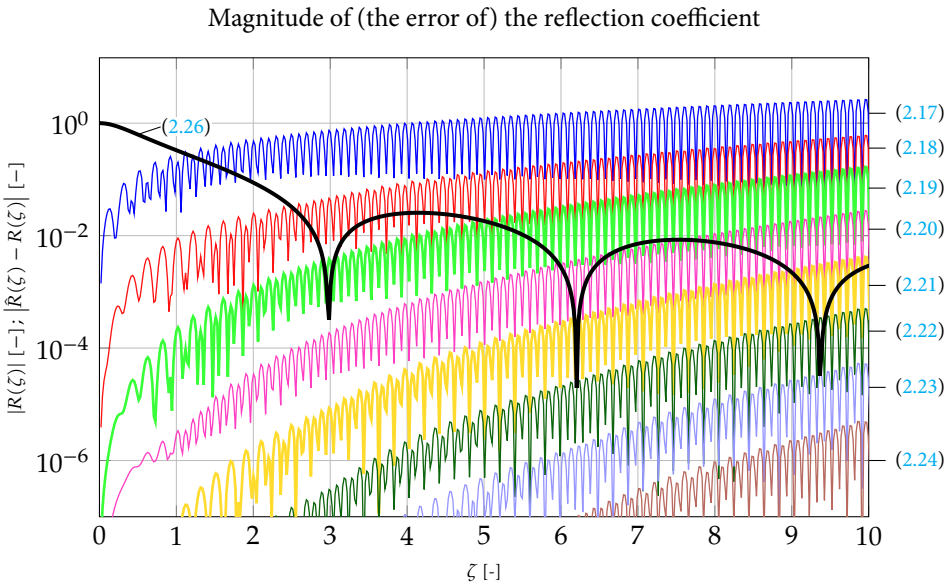


Figure 2.2: The black solid curve with label (2.26) shows the magnitude of the reflection coefficient of a rectangular potential for the KdV. All other curves show the magnitude of the *error* of a numerical approximation of the reflection coefficient. The numbers between brackets refer to the corresponding equation: (2.17)–(2.24): error due to splitting. The error without splitting (2.8) is zero here, because this potential equals its staircase approximation exactly.

2.4.2. KdV with a Rectangular potential

The rectangular potential $q(x) = \mathring{q} = 1$ for $|x| < l/2 = 1/2$ and $q(x) = 0$ else, is represented exactly by its staircase approximation for the x grid we chose. Hence, the errors due to Approximations 1 and 2 will be zero in this case. The reflection coefficient for this function is given by

$$R(\zeta) = \frac{\mathrm{j}\gamma_- \sin(\varsigma l) \exp(-\mathrm{j}\zeta l)}{\cos(\varsigma l) - \mathrm{j}\gamma_+ \sin(\varsigma l)}, \tag{2.26}$$

where $\varsigma := \sqrt{\mathring{q} + \zeta^2}$ and $\gamma_\pm := \frac{1}{2}\left(\varsigma/\zeta \pm \zeta/\varsigma\right)$ [101].

The magnitude of this reflection coefficient is shown in Fig. 2.2, as well as the magnitude of the error for the various numerical approximations. The error due to Approximation 2 is zero, as expected. The error due to Approximation 3 is reduced by choosing a splitting scheme with a higher order of accuracy, which increases the frequency up to which the approximated reflection coefficient is accurate.

2.4.3. NSE with a hyperbolic secant potential

As a comparison, we have included a numerical example for the NSE. We take the potential function $q(x) = \mathring{q}\,\mathrm{j}\,\mathrm{sech}(x)$ with $\mathring{q} = 5.5$ and $r(x) = -q^*(x)$. For this example we need a larger interval to prevent significant truncation errors (Approximation 1) and choose $x \in$

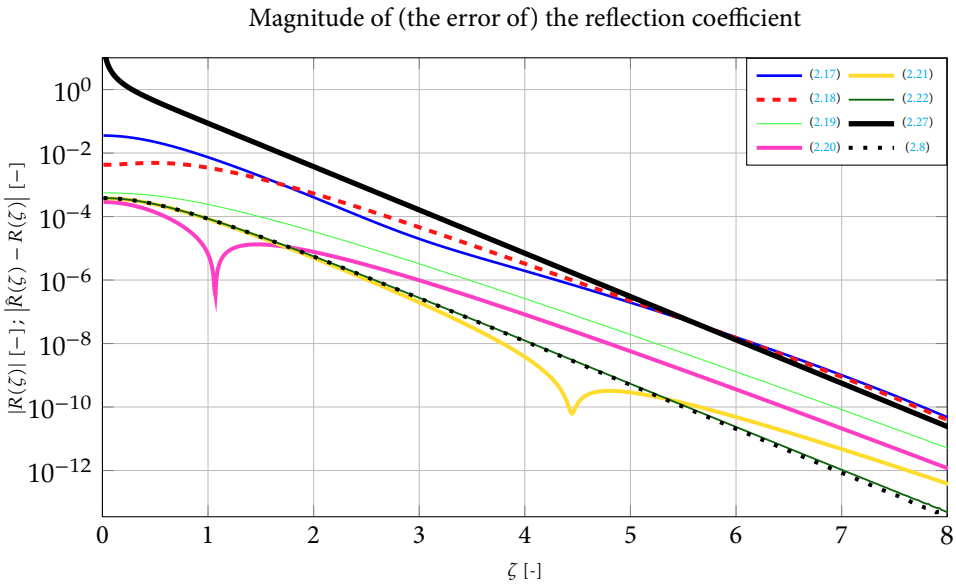


Figure 2.3: The black solid curve with label (2.27) shows the magnitude of the reflection coefficient of a hyperbolic secant potential for the NSE.All other curves show the magnitude of the *error* of a numerical approximation of the reflection coefficient. The numbers between brackets refer to the corresponding equation: (2.8): error without splitting (due to staircase approximation), (2.17)–(2.22): error with splitting.

$[-32, 32]$ while keeping $\varepsilon = 1/8$ the same, so $D = 512$. The ground truth reflection coefficient for this example is given by

$$R(\zeta) = \frac{\mathrm{j}\sin(\pi\mathring{q})}{\cosh(\pi\zeta)} \cdot \frac{\Gamma\left(\frac{1}{2} - \mathrm{j}\zeta + \mathring{q}\right)\Gamma\left(\frac{1}{2} - \mathrm{j}\zeta - \mathring{q}\right)}{\Gamma^2\left(\frac{1}{2} - \mathrm{j}\zeta\right)}, \tag{2.27}$$

where Γ is the gamma function [124].

The magnitude of this reflection coefficient is shown in Fig. 2.3, as well as the error for some of the proposed numerical approximations. Different from the KdV examples, all approximation errors decay for higher frequencies. Although the absolute error is low at high frequencies for every splitting scheme, the relative error for the first and second order scheme is high. We see that the error can be reduced significantly by using one of the schemes between third and sixth order accuracy, (2.19) to (2.22) respectively. The seventh and eighth order accurate scheme ((2.23) and (2.24)) are omitted in Fig. 2.3, because the result is on this scale indistinguishable from the error without splitting, (2.8). That is, the error due to Approximation 3 in this example becomes negligible compared to the error due to Approximation 2, which means that by choosing a splitting scheme of sufficient order, the FNFT can reach the same accuracy as any other NFT that uses a staircase approximation of the potential.

2.5. Conclusion and discussion

We have presented several exponential splitting schemes that can be incorporated into the Fast Non-linear Fourier Transforms and investigated their performance for two examples for the KdV NFT and one example for the NSE NFT, respectively. The presented higher order splitting schemes allow to increase the numerical accuracy of the NFT without having to decrease the step size (which might not always be feasible in applications).

Since the calculation time of the FNFT depends partly on the degrees of the rational approximations, one would like to obtain the maximum accuracy for a certain degree. Our method results in rational expressions, but their degrees are not monotonically increasing in the order of accuracy of the splitting schemes. For example, (2.21) with an order of accuracy of five results in a higher degree than (2.22) with an order of accuracy of six. This obscures the trade-off between the calculation time of the FNFT and its accuracy.

Remark 4 (Post-review) The numerical errors in the calculation of $\hat{\boldsymbol{H}}(\zeta)$ with the FNFT lead to much larger errors in the reflection coefficient in the KdV case compared to the NSE case. While the paper [110] on which this chapter is based was being reviewed, we noted that an alternative to (2.11) exists, as mentioned in Footnote 3. When we use it, the two cases show comparable error behaviour: Fig. 2.1 then looks very similar to Fig. 2.3, with the results from the highest order splitting schemes overlaying the error without splitting. This shows that also in the KdV case FNFTs can reach the same accuracies as the conventional NFT based on a staircase approximation of the potential. □

3

Reliable computation of the eigenvalues of the discrete KdV spectrum

We propose a numerical algorithm that computes the eigenvalues of the Korteweg–de Vries equation (KdV) from sampled input data with vanishing boundary conditions. It can be used as part of the Non-linear Fourier Transform (NFT) for the KdV. The algorithm that we propose makes use of Sturm–Liouville (SL) oscillation theory to guaranty that all eigenvalues are found. In comparison to similar algorithms that are currently available, we show that our algorithm is more robust to numerical errors and therefore more reliable. Furthermore we show that our root finding algorithm, which is based on the Newton–Raphson (NR) algorithm, typically saves computation time compared to the conventional approaches that rely heavily on bisection.

3.1. Introduction

The Korteweg–de Vries equation (KdV) is a well known non-linear Partial Differential Equation (PDE). It can serve as a model for a wide variety of nearly hyperbolic, weakly non-linear processes, such as surface waves in shallow water [24, 25, 61, 76, 106], internal waves in stratified fluids [166], acoustic waves in metals [137], electrical waves in transmission lines [119], traffic flow [64] and pressure waves in fluids [1, 95, 155]. See also [37] for a survey of some (more) applications of the KdV. The normalized form of the KdV which we consider is

$$\frac{\partial}{\partial t} q(x,t) + 6\, q(x,t) \frac{\partial}{\partial x} q(x,t) + \frac{\partial^3}{\partial x^3} q(x,t) = 0. \tag{3.1}$$

Parts of this chapter are in preparation for publication as P. J. Prins and S. Wahls. "Reliable computation of the eigenvalues of the discrete KdV spectrum". Manuscript submitted for publication.

Equation (3.1) can be mapped to the various physical forms of the KdV with dimensional variables and coefficients by means of affine transformations of q, x and t [46, §1.2]. For simplicity, we can think of x and t as position and time respectively.

The KdV is the prototypical example of a Lax-integrable PDE. By that, we mean that its initial value problem can be solved with a technique that is called scattering transform or Non-linear Fourier Transform (NFT). It parallels the use of the ordinary Fourier transform for linear PDEs: The temporal evolution of $q(x,t)$ is hard to compute, but one can transform it back and forth to a so-called spectrum, of which the evolution is simple to compute [5, Sect. 1.4], [53]. The price to pay is the calculation of the direct and inverse NFT. At first, the NFT was an analytical method that allowed mathematicians to compute exact solutions of the KdV and other Lax-integrable PDEs [4, 53, 81, 163]. Later, algorithms were developed to use the NFT in numerical computations, when only a sampled input signal is available [3, 17, 19, 101, 102, 114, 140, 143, 160]. This enabled the use of the NFT as an analysis tool that reveals the physical structure of measured data from non-linear systems [23, 25, 103–105, 108, 135, 144, 166]. However, the early numerical methods were slow: Their computation times scaled quadratically in the number of samples. This motivated the development of fast numerical NFTs [150–152]. These scale almost linearly in the number of samples. Many recent publications present various improvements on aspects of the numerical NFT in computation time and accuracy, e.g. [29, 32, 50, 60, 90–92, 110–112, 132, 148].

Before we can zoom in on the specific aspect that we address in this chapter, we have to outline the mathematical problem that defines the NFT for the KdV, and introduce some terminology. The KdV-NFT spectrum of a signal $q(x,t)$ can be obtained at any fixed time $t = t_0$ from the one-dimensional Schrödinger equation[1,2]

$$\left(\frac{\partial^2}{\partial x^2} + q(x,t_0)\right) f(x,\kappa,t_0) = \kappa^2 f(x,\kappa,t_0). \tag{3.2}$$

The input signal $q(x,t_0)$ is also called the *potential*, because of its role in the Schrödinger equation (3.2) in the context of quantum mechanics. We call any signal $f(x,\kappa,t_0)$ that satisfies (3.2) a *trajectory* of the potential $q(x,t_0)$. Here, we are concerned with real-valued potentials $q(x,t)$ that evolve according to the KdV (3.1) and furthermore satisfy the *vanishing boundary condition*

$$\int_{-\infty}^{\infty} |q(x,t_0)|\,(1+|x|)\,\mathrm{d}x < \infty \qquad \text{and} \qquad \lim_{|x|\to\infty} q(x,t_0) = 0. \tag{3.3}$$

The KdV-NFT spectrum of such a potential consists of two parts: A *continuous spectrum* and a *discrete spectrum*. The continuous spectrum describes a wave continuum called radiation. We will discuss its computation later, in Chapter 2. The discrete spectrum describes a countable number of wave components called solitons. It consists of *eigenvalues* and *norming constants*, one of each for every soliton. The eigenvalues are the values $\kappa = K_n > 0$ for which there exists a trajectory $f(x,\kappa,t_0)$ with finite energy[3] such that the Schrödinger

[1] The reason for this is that the Schrödinger equation (3.2) together with the appropriate evolution equation forms a Lax-pair that constitutes the KdV [4, 53, 81, 163].

[2] In most related literature one uses the parameter $\zeta \equiv \mathrm{j}\kappa$, where $\mathrm{j} := \sqrt{-1}$. Nevertheless, we use κ in this chapter since it simplifies the exposition and keeps all computations in the real domain.

[3] The energy of $f(x)$ is defined as $\int_{-\infty}^{\infty} |f(x)|^2\,\mathrm{d}x$.

equation (3.2) is satisfied. This trajectory $f(x, K_n, t_0)$ is the corresponding *eigenfunction*. Each eigenfunction is unique up to a scalar factor. The norming constants can be obtained from the eigenfunctions, as we will discuss later, in Chapter 4. In this chapter we address the computation of the eigenvalues.

In the literature two approaches can be found for the computation of the eigenvalues [82, §2.1], [160, §IV]:

The first approach is to use a *finite dimensional approximation* of (3.2) that turns its eigenproblem into a (large) matrix eigenproblem. Collocation methods (e.g. [44, 157]), rational approximations (e.g. [150]), finite difference methods (e.g. [82, §2.2.1]), and Hill's method [39, 67], [142, §8.2.1], [143, §5.1] belong to the first class. The computational complexity of these methods is at best $O(D^2)$, where D is the number of degrees of freedom of the discretization. The accuracy of these methods quickly deteriorates for faster oscillating trajectories [82, Chap. 2].[4]

The second approach is known as the *shooting* approach. That is, one reduces the boundary value problem first to an initial value problem by keeping just one boundary condition. A free parameter, κ in our case, is introduced to make the initial value problem well-defined. Then one verifies with an initial value solver if the remainder of the boundary conditions is also satisfied. This procedure is iterated in a root finder that tries different values of the parameter until the boundary value problem is solved. Shooting methods can be implemented with a computational complexity of only $O(DNP)$, where D is the number of samples, N is the number of eigenvalues and P is average required number of iterations per eigenvalue.

Basic implementations of the shooting approach cannot guarantee global convergence, so they may not find every eigenvalue. However, global convergence can be guaranteed by combining a shooting method with Sturm–Liouville (SL) oscillation theory. (This theory applies because the Schrödinger equation (3.2) is a specific example of a SL equation.) In short: According to the SL theory, the number of zero-crossings of the trajectory $f(x, \kappa, t_0)$ at a fixed value κ as x runs from $-\infty$ to $+\infty$, reveals the number of eigenvalues that is greater than κ. To make use of this information, we need to track the zero-crossings of the trajectory. From the Schrödinger equation Prüfer [116] derived a non-linear Ordinary Differential Equation (ODE) for the *phase* of the trajectory: the Prüfer equation. The number of zero-crossings follows trivially from the phase. The shooting method that relies on the integration of the Prüfer equation has become known as the Prüfer method [15, 97]. However, the Prüfer equation is a stiff system which is hard to integrate [82, §2.1]. Alternatively, one can integrate the Schrödinger equation itself, which is simpler, and meanwhile count the number of zero-crossings [69, 102, 115]. In this chapter we propose an algorithm that also uses this method for the computation of the eigenvalues.

The challenge within this method is to count the zero-crossings of the trajectory in a numerically robust way. Any missed or doubly counted zero-crossing can cause a significantly wrong result. Even causes for a miscount that may seem pathological at first sight, are surprisingly likely to occur in practice, because the root counting procedure is repeated D (samples) times N (eigenvalues) times P (iterations) times, and because the eigenfunctions that we search are themselves corner cases of the computation. Osborne [102] proposed to compute the number of zero-crossings by counting sign changes of the trajectory from sample to sample. However, Christov [33] observed numerical instabilities when he tried to

[4] If (3.2) and (3.3) are observed, $f(x, K_n, t_0)$ oscillates faster as K_n is closer to zero.

	Osborne [102]	Matslise 2.0 [85]	Proposed
Input	Samples	Function	Samples
Boundary at $\pm\infty$	No	Only as input	Yes
Order	$O(\varepsilon^2)$	$O(\varepsilon^{18})$	$O(\varepsilon^4)$
Root finder	Bisection	Bisection to bracket, refinement with NR	NR if possible, bisection for a new starting point

Table 3.1: Characteristics of two SL based algorithms to compute the eigenvalues of the Schrödinger equation found in the literature, and the proposed algorithm. Matslise 2.0 allows the specification of boundary conditions at $\pm\infty$, but replaces $\pm\infty$ internally by finite value before the computation.

use Osbornes algorithm on certain simulation data. Pruess and Fulton [115, §4] and Ixaru, Meyer, and Berghe [69, §5.2] proposed a more rigorous counter, to cover the possibility that there is more than one zero-crossing between two consecutive samples. However, their approach may lead to a miscount if a zero-crossing occurs close to the boundary between two integration steps. In Section 3.3.2, we will clarify this vulnerability and explain how the algorithm that we propose mends it.

Another aspect on which we improve on SL based shooting methods that are known in the literature, is the root finder. The use of SL theory provides upper and lower bounds on the eigenvalues. This information is most easily incorporated into a *bracketing* root finder, such as bisection. (A bracket consists of an upper and lower bound of a certain root.) Unfortunately, bisection has only linear convergence. Some algorithms speed up the convergence by using a two-stage approach: First, bisection is applied to obtain a (sufficiently tight) bracket for each root, which contains no other roots. Second, a root finder with a faster convergence is applied to refine the localization of the eigenvalue. The second stage could for example use regula falsi [33] or Newton–Raphson (NR) [69, 85, 102]. These algorithms are thus still limited to linear convergence in the first stage. Algorithms that apply an *open* root finder such as NR in the second stage then face the problem of finding an initial guess for which the iterations do not jump out of the just computed bracket. The algorithm that we propose in this chapter uses a different approach. Since we typically need to compute all of the eigenvalues for the NFT, there is no need to isolate the eigenvalues upfront and to specify which one to localize next. Instead, we let the NR procedure converge to *any* eigenvalue. Thereafter, we select a new starting point and let NR converge to another eigenvalue. Meanwhile, we update the brackets of all the eigenvalues on every iteration of the NR procedure, based on SL theory. If the next NR iteration is not within the bracket of any of the eigenvalues, we switch to a new starting point within one of the brackets. This ensures that every iteration increases our knowledge about the location of at least one of the eigenvalues.

We briefly mention some other relevant aspects that distinguish different root finding algorithms. These aspects are summarized in Table 3.1 for the proposed algorithm and for the two algorithms which we use as benchmark algorithms.

- For practical applications of the KdV-NFT we need an algorithm that takes a sampled input signal. Some algorithms that were developed for other purposes, such as Matslise [85] require a functional description of the input signal instead. In our benchmark comparison we will work around this issue by fitting a Fourier series to the data.

- Some algorithms were developed for finding eigenvalues of the Schrödinger equation on a finite interval. The KdV-NFT for potentials that satisfy the vanishing boundary condition (3.3) requires boundary conditions at infinity. Not all algorithms support this. Matslise [85] allows the specification of boundary conditions at $\pm\infty$, but replaces $\pm\infty$ internally by a finite value before the computation. That means that not only the potential is truncated, but also the trajectories. In the NFT literature it is common practice to exploit the fact that trajectories of the Schrödinger equation can be written as the sum of two exponentials outside the support of the truncated potential. Therefore, no truncation of the trajectories is required. We also apply this in the proposed algorithm. We adapt this aspect of Osbornes algorithm [102] accordingly for the benchmark comparison in this chapter.

- There exist many integrators that can be used to integrate the Schrödinger equation. If the potential is (piecewise) sufficiently smooth, integrators of higher order are more accurate, but computationally more expensive per integration step. Matslise [85] uses an 18th order integrator, but severely reduces the number of integration steps to trade a part of the accuracy gain for computational cost. We propose to use a specific fourth order integrator (see Section 3.3.4), because it allows for an accurate computation of the zero-crossings of the trajectory (see (3.18)).

The rest of this chapter is organized as follows. In Section 3.2 we mention a few important aspects of the eigenvalues of the KdV and introduce two variables that play a fundamental role in the proposed algorithm to find the eigenvalues. These variables are the scattering parameter $a(\kappa)$ and the accounting function $s(-\infty, \infty, \kappa)$. In Section 3.3 we show how to compute these parameters numerically at a given value of κ. In Section 3.4 we propose a root finder that finds the eigenvalues by sampling these parameters, with a lower computational cost than bisection. In Section 3.5 we evaluate the proposed algorithm by comparing it with other methods on six different example signals. The chapter is concluded in Section 3.6.

3.2. Preliminaries

Recall that the problem we aim to solve in this chapter is the following. Given a uniformly sampled potential $q(x, t_0)$ that satisfies (3.3), find all the eigenvalues. We will search for the eigenvalues with a shooting method. That is, choose a value κ, determine if κ is an eigenvalue of $q(x, t_0)$ and repeat until all the eigenvalues are localized. In this section we will explain two strategies to determine if κ is an eigenvalue: The basic method and the one that makes use of SL oscillation theory. We will also mention the numerical advantages and disadvantages of both strategies. In Section 3.3 we will explain how to perform the numerical computations for these strategies. In Section 3.4 we will integrate these strategies in one algorithm that combines the advantages of both.

3.2.1. Notation

Matrices have an upper case symbol in bold weight (e.g. $\boldsymbol{A}$), vectors have a lower case symbol in bold weight (e.g. $\boldsymbol{\phi}$), scalars have a normal weight (e.g. x). Constants have an upright style (e.g. π), real or imaginary variables have a slanted style (e.g. x), booleans are printed in a typewriter font (e.g. $\mathtt{f}$). We will use super scripts $^{\mathtt{l}}$ and $^{\mathtt{u}}$ for respectively the lower and

upper boundary value of an interval. $O(\,)$ is used as the Landau 'big-O' order symbol. The symbol ':=' denotes a definition. The symbol '$\leftarrow$' means that the left hand side gets the value of the right hand side (at that point in an algorithm). The symbol '$\wedge$' denotes the logical AND operation. The notation '$|x|$' means the absolute value of x. The notation '$\lfloor x \rceil$' means rounding x towards the nearest integer,'$\lceil x \rceil$' means rounding x towards the nearest greater or equal integer, '$\lfloor x \rfloor$' means rounding x towards the nearest lesser or equal integer. We will make extensive use of the *Iverson bracket* for piecewise expressions: $[\![\mathtt{b}]\!] := 1$ if `b` is `true`, 0 otherwise. By convention, $x[\![\mathtt{false}]\!] = 0$, even if x is infinite or undefined. Single square brackets [ ] are used for the composition of vectors and matrices. Finally, we write exp( ) for the natural scalar or matrix exponential function: $\exp(\boldsymbol{A}) \equiv \mathrm{e}^{\boldsymbol{A}} := \sum_{i=0}^{\infty} \boldsymbol{A}^i / i!$.

3

3.2.2. Eigenvalues of the Korteweg–de Vries equation

As mentioned in Section 3.1, the eigenvalues of the KdV are obtained from the Schrödinger equation (3.2). It can be shown that if $q(x,t)$ is real and evolves according to the KdV (3.1), then the eigenvalues are constant and isolated (with multiplicity one) [5, 79], and that all eigenvalues[5] satisfy $0 < K_n^2 < \sup_x q(x,t)$ (for all t) [127, p. 732]. We will index the eigenvalues such that $0 < K_1 < K_2 < \cdots < K_N < \sqrt{\sup_x q(x,t)}$, where $N \geqslant 0$ is the number of eigenvalues. Since the eigenvalues are constant, we can simplify the notation by dropping the dependence on the arbitrary fixed time $t = t_0$ of other variables.

3.2.3. Scattering parameter $a(\kappa)$

Given any potential $q(x)$ that satisfies the vanishing boundary condition (3.3), it is readily verified that as $|x| \to \infty$ all trajectories of the Schrödinger equation (3.2) can be parametrized as a linear combination of $\exp(\pm\kappa x)$. We can thus define special trajectories that vanish as $x \to \pm\infty$ respectively. These trajectories are known as Jost solutions and satisfy the boundary conditions

$$\lim_{x\to-\infty} \phi(x,\kappa)\exp(-\kappa x) = 1; \tag{3.4}$$

$$\lim_{x\to+\infty} \psi(x,\kappa)\exp(+\kappa x) = 1. \tag{3.5}$$

Since the eigenfunctions have finite energy, they must vanish both as $x \to -\infty$ and as $x \to \infty$. Therefore every eigenfunction must satisfy both (3.4) and (3.5) up to an arbitrary constant factor: $\phi(x,K_n) \propto f(x,K_n) \propto \psi(x,K_n)$. We can use this insight to find the eigenvalues according to the shooting method. That is, we solve (3.2) for the boundary condition (3.4) at several values of κ. Then we check if $\phi(x,\kappa)$ satisfies (3.5) up to a scalar factor. To simplify this check, one defines the scattering parameter $a(\kappa)$ as follows.

$$a(\kappa) := \frac{\mathrm{W}[\psi,\phi]}{2\kappa}, \tag{3.6}$$

where

$$\mathrm{W}[\psi,\phi] := \psi(x,\kappa)\frac{\partial\,\phi(x,\kappa)}{\partial x} - \phi(x,\kappa)\frac{\partial\,\psi(x,\kappa)}{\partial x}. \tag{3.7}$$

[5] From (3.3) it follows that $\sup_x q(x,t) \geq 0$. If $\sup_x q(x,t) = 0$, there are no eigenvalues, so the discrete spectrum is an empty set.

The Wronskian $W[\psi, \phi]$, also known as the mismatch function [82, Eq. 2.14], vanishes if and only if the trajectories $\phi(x, \kappa)$ and $\psi(x, \kappa)$ are proportional. Hence, the eigenvalues K_n are the values κ for which $a(\kappa) = 0$. Usually one searches for the eigenvalues with a root finder that also makes use of the gradient $a'(\kappa) := \frac{\partial}{\partial \kappa} a(\kappa)$, for example NR. Initially, both a lower and an upper bound on the eigenvalues are known, see Section 3.2.2. These bounds can be used to guess suitable starting values for the root finder.

It is unreliable to compute the eigenvalues from samples of $a(\kappa)$ and $a'(\kappa)$ only, because it remains unknown how many eigenvalues there are. Suppose we have two adjacent, non-zero samples $a(\kappa_1)$ and $a(\kappa_2)$, we can only infer the parity of the number of eigenvalues between κ_1 and κ_2: The parity is odd if $a(\kappa_1)\, a(\kappa_2) < 0$ and even if $a(\kappa_1)\, a(\kappa_2) > 0$. In the odd case there must be at least one eigenvalue between these samples, which is useful information. However, in the even case the number of eigenvalues between these samples could be zero as well as any other even number. No matter how many samples are taken, there is always a possibility that one or more pairs of eigenvalues are missed. In practice, we would have to evaluate $a(\kappa)$ on a very fine κ-grid and hope that odd means one and even means zero. In that manner we can never be sure that we have localized all the eigenvalues. Additional information needs to be collected to ensure that no eigenvalues are missed. An attractive source of additional information is the accounting function, which we discuss next.

3.2.4. Accounting function

Since the Schrödinger equation (3.2) is an example of an SL equation, SL oscillation theory applies. From that theory, it is known that the number of solutions of $\phi(x, \kappa_0) = 0$ for fixed κ_0 and finite real x is equal to the number of eigenvalues that is greater than κ_0 [165, Thms. 2.6.2 & 10.12.1.(4)]. We call these solutions zero-crossings for short.[6] Let us define the accounting function $s\big(x^{\mathrm{l}}, x^{\mathrm{u}}, \kappa\big)$ as the number of zero-crossings of $\phi(x, \kappa)$ in the open interval $x \in (x^{\mathrm{l}}, x^{\mathrm{u}})$ at a fixed value κ. Then $s(-\infty, \infty, \kappa)$ is equal to the number of eigenvalues that is greater than κ. Since all eigenvalues are positive, the total number of eigenvalues is thus given by $N = s(-\infty, \infty, 0)$. Since all the eigenvalues are smaller than $\sqrt{\sup_x q(x)}$, we know a priori that $s(-\infty, \infty, \kappa) = 0$ for $\kappa \geq \sqrt{\sup_x q(x)}$. When κ is increased from zero to $\sqrt{\sup_x q(x)}$, the value of the accounting function $s(-\infty, \infty, \kappa)$ is decremented by one whenever κ equals an eigenvalue. We can thus localize the eigenvalues by searching for the steps in $s(-\infty, \infty, \kappa)$. The value $s(-\infty, \infty, \kappa)$ for κ just above or just below each localized eigenvalue K_n, reveals the index n of that eigenvalue.

The advantage compared to searching for zero-crossings of $a(\kappa)$ is the following. Recall that if we have two adjacent, non-zero samples $a(\kappa_1)$ and $a(\kappa_2)$, we can only infer whether the number of eigenvalues between κ_1 and κ_2 is odd (if $a(\kappa_1)\, a(\kappa_2) < 0$) or even (if $a(\kappa_1)\, a(\kappa_2) > 0$). On the other hand, if we know $s(-\infty, \infty, \kappa_1)$ and $s(-\infty, \infty, \kappa_2)$, their difference reveals not just the parity of the number of eigenvalues between κ_1 and κ_2, but the number itself. There is no risk of missing any closely spaced eigenvalues if the κ-grid is too coarse.

[6] All these zeros are crossings, because if it were the case that $\phi(x_0, \kappa_0) = 0$ and $\frac{\partial}{\partial x} \phi(x, \kappa_0)|_{x=x_0} = 0$ at the same fixed position x_0, then (3.2) implies $\phi(x, \kappa_0) = 0\ \forall x$, which violates (3.4). Hence $\phi(x_0, \kappa_0) = 0 \Rightarrow \frac{\partial}{\partial x} \phi(x, \kappa_0)|_{x=x_0} \neq 0$, implying that $\phi(x, \kappa_0)$ must change sign at $x = x_0$.

Of course, the accounting function is only useful if we can reliably evaluate it in a numerical computation. To that end we have to detect the zero-crossings of $\phi(x, \kappa_0)$, where κ_0 is fixed. That may seem like a similar problem as detecting the roots of $a(\kappa)$: If we would evaluate $\phi(x, \kappa_0)$ on an x-grid and count the sign changes, we might miss pairs of zero-crossings between two adjacent samples. However, the Schrödinger equation allows us to reliably detect even multiple zero-crossings of $\phi(x, \kappa_0)$ between samples. Since $\phi(x, \kappa_0)$ is a trajectory of the Schrödinger equation (3.2), the 'speed' at which it oscillates (the slope of the Prüfer phase) is controlled by the potential $q(x)$. If the potential is well-behaved between samples, the oscillation is also well-behaved. For the proposed algorithm we will choose a *reconstruction* from the given samples, for which the Schrödinger equation has a piecewise analytic solution. This solution allows us to compute the number of zero-crossings piece by piece. We will discuss this further in Section 3.3.

The downside of using the accounting function is that it does not have a gradient to help finding the eigenvalues. Therefore in Section 3.4 we will apply a NR root finder on the scattering parameter $a(\kappa)$ and its gradient $a'(\kappa)$, and evaluate the accounting function in parallel in order to bracket each eigenvalue.

3.3. Integration of the Schrödinger equation

In this section we discuss the numerical computation of the scattering parameter $a(\kappa)$ and the accounting function $s(-\infty, \infty, \kappa)$ from the potential $q(x)$. In Section 3.4 we will incorporate these computations in an algorithm that finds the eigenvalues of the KdV efficiently and accurately.

3.3.1. The sampled and reconstructed potential

We assume that we do not know the true potential $q(x)$, but only a finite number of samples on a uniform x-grid. We are going to define a reconstruction of the potential, for which we can integrate the Schrödinger equation while keeping track of the number of zero-crossings. We denote the number of samples by D and the step size by ε. That is, if x_1 is the first grid point, then all the grid points are given by $x_d = x_1 + (d-1)\varepsilon$, where $d \in \{1, 2, \dots, D\}$ and the known potential samples are $q_d := q(x_d)$. For notational convenience, we define around each grid point an interval $(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}) := (x_d - \frac{\varepsilon}{2}, x_d + \frac{\varepsilon}{2})$. Indeed, for $d < D$ it follows that $x_{d+1}^{\mathrm{l}} = x_d^{\mathrm{u}}$.

We will compute $a(\kappa)$ and $s(-\infty, \infty, \kappa)$ first for the simplest reconstruction of the potential from the samples q_d. That is, we use a piecewise constant reconstruction, $\hat{q}(x)$, by the midpoint rule. The known potential samples give no information about $q(x)$ for $x \notin (x_1^{\mathrm{l}}, x_D^{\mathrm{u}})$. Therefore we set in the reconstruction $\hat{q}(x) = 0$ for $x \notin (x_1^{\mathrm{l}}, x_D^{\mathrm{u}})$. Hence,

$$\hat{q}(x) := \sum_{d=1}^{D} q_d [\![x_d^{\mathrm{l}} < x < x_d^{\mathrm{u}}]\!]. \tag{3.8}$$

(Please refer to Section 3.2.1 for the meaning of the Iverson bracket $[\![\ ]\!]$.) The eigenvalues of $\hat{q}(x)$ approximate those of $q(x)$ up to an error term $O(\varepsilon^2)$. After demonstrating the computation for $\hat{q}(x)$, we discuss in Section 3.3.4 how to upgrade the method such that the error term reduces to $O(\varepsilon^4)$ for sufficiently smooth potentials.

3.3.2. Numerical computation of the scattering parameter $a(\kappa)$ and the accounting function

In numerical computations, it is convenient to replace (3.6) by an algebraic expression. We do that by defining vector valued trajectories as

$$\boldsymbol{f}_{\mathrm{C}}(x,\kappa) := \begin{bmatrix} f(x,\kappa) \\ \frac{\partial}{\partial x} f(x,\kappa) \end{bmatrix}. \tag{3.9}$$

Then (3.6) is equivalent to the algebraic expression

$$a(\kappa) \equiv \frac{1}{2}\, \boldsymbol{\psi}_{\mathrm{C}}^{\top}(x,\kappa) \begin{bmatrix} 0 & \kappa^{-1} \\ -\kappa^{-1} & 0 \end{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x,\kappa). \tag{3.10}$$

Both (3.6) and (3.10) hold for all x. Therefore, $a(\kappa)$ can be computed at any *matching point* x for which both $\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)$ and $\boldsymbol{\psi}_{\mathrm{C}}(x,\kappa)$ are known. For simplicity, we will evaluate (3.10) at $x = x_D^{\mathrm{u}}$ in the analysis that follows.[7] Therefore we need to propagate $\phi(x,\kappa)$ as defined in (3.4) from $-\infty$ to x_D^{u}. We do that by exactly solving the Schrödinger equation (3.2) in every constant piece of the reconstruction of the potential, (3.8). Likewise, we propagate $\psi(x,\kappa)$ as defined in (3.5) back from ∞ to x_D^{u}.

Furthermore, we evaluate the accounting function. We will do that by summing the zero-crossings of $\phi(x,\kappa)$ in every piecewise constant interval of the reconstructed potential. That is,

$$s(-\infty,\infty,\kappa) = s\big(-\infty, x_1^{\mathrm{l}}, \kappa\big) + \left(\sum_{d=1}^{D} s\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big) \right) + s\big(x_D^{\mathrm{u}}, \infty, \kappa\big). \tag{3.11}$$

When we evaluate the accounting function (3.11) numerically, we must be very careful with the treatment of the number zero. Firstly, to determine the sign of $\phi(x,\kappa)$ near one of its zero-crossings, we must use the same intermediate value $\phi\big(x_d^{\mathrm{u}},\kappa\big)$ for the computation of $s\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big)$ as for the computation of $\phi\big(x_{d+1}^{\mathrm{u}},\kappa\big)$ and further. Otherwise the sign of $\phi\big(x_d^{\mathrm{u}},\kappa\big)$ may differ between the two computation paths. We will come back to this in (3.18) and the discussion thereafter. Secondly, we must be careful to obtain a correct result in case any intermediate value $\phi\big(x_d^{\mathrm{u}},\kappa\big)$ equals exactly 0. We found that the simplest treatment is to *consider zero as a positive number*. That means that we count the crossings at 0^-, between zero and the smallest representable negative number. Since 0^- has no representation in finite precision, these crossings can numerically never lie exactly on an x grid point. This allowed us to write (3.11) as a summation of zero crossings in open rather than closed intervals.

We have to be careful with the case $\lim_{x\to\infty} \phi(x,\kappa) = 0$. Even if $\phi(x,\kappa)$ approaches zero from below as $x \to \infty$, we must not count this as a zero crossing. This might sound obvious at this point, but later, namely in (3.20), it will lead to one strict inequality '> 0', whereas we need '≥ 0' everywhere else, in accordance with the treatment of zero as a positive number. This exception is essential for the working of the algorithm when $a(K_n)$ is

[7] If the numerical representation of the trajectories causes an overflow during the computation, a different choice of x is a possible workaround. However, one could also solve it by rescaling the trajectory by a suitable non-zero scalar factor c, i.e. $\boldsymbol{f}_{\mathrm{C}}(x,\kappa) \leftarrow c\,\boldsymbol{f}_{\mathrm{C}}(x,\kappa)$, and look for the roots of $c\,a(\kappa)$), and/or, if $\kappa \neq 0$, by choosing a more suitable basis for the representation of the trajectories. Cf. Section 5.3.

numerically equal to zero for any eigenvalue. It can be verified that $\lim_{x\to\infty}\phi(x,\kappa)=0$ indicates that κ is an eigenvalue and that the eigenfunctions of K_N, K_{N-2}, … approach zero from above as $x\to\infty$, whereas those of K_{N-1}, K_{N-3}, … approach from below. By never counting $\lim_{x\to\infty}\phi(x,\kappa)=0$ as a zero-crossing, we obtain a consistent behaviour of the accounting function at its steps. Namely, $s(-\infty,\infty,K_n):=\lim_{\kappa\downarrow K_n}s(-\infty,\infty,\kappa)\equiv\lim_{\kappa\uparrow K_n}s(-\infty,\infty,\kappa)-1$, where $\downarrow$ denotes the limit from above and $\uparrow$ denotes the limit from below.

The lower tail: $x\in(-\infty,x_1^{\mathrm{l}})$

In the interval $x\in(-\infty,x_1^{\mathrm{l}})$ we know that $\phi(x,\kappa)$ satisfies the Schrödinger equation (3.2) and the boundary condition (3.4). The reconstructed potential $\hat{q}(x)$ is zero in this interval. It is readily verified that the solution in this interval is $\phi(x,\kappa)\equiv\exp(\kappa x)$. Hence at the boundary of the next interval we find

$$\boldsymbol{\phi}_{\mathrm{C}}(x_1^{\mathrm{l}},\kappa)=\begin{bmatrix}1\\ \kappa\end{bmatrix}\exp(\kappa x_1^{\mathrm{l}}). \tag{3.12}$$

The Jost solution $\phi(x,\kappa)$ has no zero-crossings in this interval, so $s\big(-\infty,x_1^{\mathrm{l}},\kappa\big)=0$.

The support: $x\in(x_1^{\mathrm{l}},x_D^{\mathrm{u}})$

The computations in this interval are similar to those for the periodic boundary condition algorithm of Osborne [102], but improve on it by a more robust computation of the accounting function. This interval consists of D adjacent subintervals $(x_d^{\mathrm{l}},x_d^{\mathrm{u}})$. In each of these subintervals the potential is constant. Therefore the Schrödinger equation (3.2) in the d-th subinterval simplifies to $\frac{\partial^2}{\partial x^2}\phi(x,\kappa)=\kappa^2-q_d$. The vector-valued Jost solution thus satisfies

$$\frac{\partial}{\partial x}\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)=\boldsymbol{A}_{\mathrm{C}}(q_d,\kappa)\,\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa), \tag{3.13}$$

where

$$\boldsymbol{A}_{\mathrm{C}}(q_d,\kappa):=\begin{bmatrix}0 & 1\\ \kappa^2-q_d & 0\end{bmatrix}. \tag{3.14}$$

By solving (3.13) subject to the boundary condition at $x=x_d^{\mathrm{l}}$, we find at $x=x_d^{\mathrm{u}}=x_{d+1}^{\mathrm{l}}$ that

$$\boldsymbol{\phi}_{\mathrm{C}}(x_d^{\mathrm{u}},\kappa)=\boldsymbol{H}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa)\,\boldsymbol{\phi}_{\mathrm{C}}(x_d^{\mathrm{l}},\kappa), \tag{3.15}$$

where the change of state matrix

$$\boldsymbol{H}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa)=\exp(\varepsilon\,\boldsymbol{A}_{\mathrm{C}}(q_d,\kappa))=\begin{bmatrix}\cos(\gamma\varepsilon) & \varepsilon\,\operatorname{sinc}(\gamma\varepsilon)\\ -\gamma\sin(\gamma\varepsilon) & \cos(\gamma\varepsilon)\end{bmatrix}, \tag{3.16}$$

where

$$\gamma:=\sqrt{q_d-\kappa^2},\qquad\qquad \operatorname{sinc}(\gamma\varepsilon):=\frac{\sin(\gamma\varepsilon)}{\gamma\varepsilon}[\![\gamma\varepsilon\neq 0]\!]+[\![\gamma\varepsilon=0]\!].$$

We remark that γ is either real or imaginary, but $\boldsymbol{H}_{\mathrm{C}}\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big)$ is always real.

To count the number of zero-crossings of $\phi(x,\kappa)$ in the d-th subinterval, it is not sufficient in general to look only at the signs of $\phi\big(x_d^{\mathrm{u}},\kappa\big)$ and $\phi\big(x_d^{\mathrm{l}},\kappa\big)$ as in [102]. The simplest way to see this, is by looking at the structure of the closed form solution of $\phi(x,\kappa)$ in the d-th subinterval. That is,

$$\phi(x,\kappa) = \begin{cases} c_1 \exp\Big(-x\sqrt{\kappa^2 - q_d}\Big) + c_2 \exp\Big(x\sqrt{\kappa^2 - q_d}\Big) & q_d - \kappa^2 < 0, \\ c_1 + c_2 x & q_d - \kappa^2 = 0, \\ c_1 \sin\Big(c_2 + x\sqrt{q_d - \kappa^2}\Big) & q_d - \kappa^2 > 0; \end{cases} \tag{3.17}$$

where c_1 and c_2 are real constants. If $q_d - \kappa^2 \leq 0$ (the non-oscillatory case) we see from (3.17) that the number of zero-crossings in the d-th interval is either zero or one. Under that condition the number of zero-crossings can be determined reliably by comparing the signs of $\phi\big(x_d^{\mathrm{l}},\kappa\big)$ and $\phi\big(x_d^{\mathrm{u}},\kappa\big)$. However, the oscillatory case $q_d - \kappa^2 > 0$ requires a more careful computation, because there could be more than one zero-crossing in the d-th interval. In the oscillatory case we need to look at the propagation of the phase of the sine in (3.17) across the d-th interval, and compute the number of zero-crossings accordingly. If $0 < \varepsilon^2(q_d - \kappa^2) < \pi^2$, the number of zero-crossings is still at most one. Hence, both ways of counting $s\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big)$ are valid in this domain. Sign comparison is computationally cheaper, but switching between the two computations exactly at $\varepsilon^2(q_d - \kappa^2) = \pi^2$ is numerically not robust. Therefore we choose to switch at $\varepsilon^2(q_d - \kappa^2) = 3^2$. Hence, we count the number of zero-crossings in the interval $(x_d^{\mathrm{l}}, x_d^{\mathrm{u}})$ as

$$s\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big) = \begin{cases} \Big|\llbracket \phi\big(x_d^{\mathrm{u}},\kappa\big) \geq 0 \rrbracket - \llbracket \phi\big(x_d^{\mathrm{l}},\kappa\big) \geq 0 \rrbracket\Big| & (\varepsilon\gamma)^2 < 9, \\ \llbracket \phi\big(x_d^{\mathrm{u}},\kappa\big) \geq 0 \rrbracket - \llbracket \phi\big(x_d^{\mathrm{l}},\kappa\big) \geq 0 \rrbracket + 2\left\lfloor \frac{\varepsilon\gamma - \theta\big(x_d^{\mathrm{u}}\big) + \theta\big(x_d^{\mathrm{l}}\big)}{2\pi} \right\rceil & (\varepsilon\gamma)^2 \geq 9; \end{cases} \tag{3.18}$$

where $\gamma = \sqrt{q_d - \kappa^2}$, $\theta(x) := \operatorname{atan2}\Big(\gamma\,\phi(x,\kappa), \frac{\partial}{\partial x}\phi(x,\kappa)\Big)$, and '$\lfloor\ \rceil$' means 'round to the nearest integer'. The four-quadrant arctangent is defined by

$$\operatorname{atan2}(y,z) := \begin{cases} \operatorname{atan}\big(\frac{y}{z}\big) + \pi\llbracket z < 0 \rrbracket\big(\llbracket y \geq 0 \rrbracket - \llbracket y < 0 \rrbracket\big) & z \neq 0, \\ \frac{\pi}{2}\big(\llbracket y \geq 0 \rrbracket - \llbracket y < 0 \rrbracket\big) & z = 0 \wedge y \neq 0, \\ \text{undefined} & z = 0 \wedge y = 0. \end{cases} \tag{3.19}$$

The values that are needed to evaluate (3.18) are thus q_d, κ, $\boldsymbol{\phi}_{\mathrm{C}}\big(x_d^{\mathrm{l}},\kappa\big)$ and $\boldsymbol{\phi}_{\mathrm{C}}\big(x_d^{\mathrm{u}},\kappa\big)$ as computed with (3.15). Note that it is important to follow the definitions above carefully when either $\phi\big(x_d^{\mathrm{l}},\kappa\big) = 0$, or $\phi\big(x_d^{\mathrm{u}},\kappa\big) = 0$. That is, zero counts as a positive number.

Let us highlight the merit of (3.18) in comparison to the literature. When we use the accounting function $s(-\infty,\infty,\kappa)$ in the search for eigenvalues, we sample it at different values of κ. The accounting function is a non-increasing staircase function of κ. This fact is implicitly exploited while bracketing the eigenvalues. Therefore, if a numerical evaluation of the accounting function is off by only ± 1, the perception of the accounting function will be very different. The error will often remain unnoticed because the implicit assumption

of a non-increasing staircase prevents the algorithm from taking suitable samples for that. Hence, *it is essential to count every zero-crossing exactly once.* As mentioned before, Osborne [102] relies on the cheap sign check in the upper line of (3.18), for every value of $(\varepsilon\gamma)^2$. This may lead to a miscount if $\pi^2 \leq (\varepsilon\gamma)^2 < (2\pi)^2$ and will surely lead to a miscount if $(\varepsilon\gamma)^2 \geq (2\pi)^2$. Pruess and Fulton [115, §4], Ixaru, Meyer, and Berghe [69, §5.2], and Ledoux and Van Daele [84, §4.2] distinguish between the non-oscillatory case $(\varepsilon\gamma)^2 \leq 0$ where the cheap sign check suffices, and the oscillatory case $(\varepsilon\gamma)^2 > 0$ where they all use a more expensive computation. However, the step size ε of a sampled input signal is usually small compared to the fastest oscillation in any of the trajectories in practice, in order to have a sufficiently accurate representation of that signal. Therefore, most of the oscillatory case samples will be in the interval $0 < (\varepsilon\gamma)^2 < 9$, for which we can safely use the cheap sign check. *Our computation is thus more efficient in this respect.*

Furthermore, the oscillatory case computation that is used by [115, §4] is not robust when a zero-crossing occurs close to x_d^{u}. The problem is that the sign of $\phi\big(x_d^{\mathrm{u}}, \kappa\big)$ is implicitly computed twice: First from $\phi\big(x_d^{\mathrm{l}}, \kappa\big)$ and $\varepsilon\gamma$ to count the zeros in the dth subinterval and second with (3.15) to obtain the initial condition for the next subinterval, $\phi\big(x_{d+1}^{\mathrm{l}}, \kappa\big)$. If there is a zero-crossing near x_d^{u}, then due to numerical inaccuracies one computation may end up just above zero while the other ends up just below zero. The consequence is then that this zero-crossing is counted either twice or not at all. We instead compute $\phi\big(x_d^{\mathrm{u}}, \kappa\big) = \phi\big(x_{d+1}^{\mathrm{l}}, \kappa\big)$ once, with (3.15). As indicated in the bottom case of (3.18), we only use $\varepsilon\gamma$ to count the number of full oscillations (with two zero-crossings each) that remains after accounting for the initial phase angle $\theta\big(x_d^{\mathrm{l}}\big)$ and final phase angle $\theta\big(x_d^{\mathrm{u}}\big)$. *This ensures that with our computation a zero-crossing near $x_d^{\mathrm{u}} = x_{d+1}^{\mathrm{l}}$ is always counted either in the dth or in the $(d+1)$th subinterval.*

The computation for the oscillatory case that is used by [69, §5.2] and Ledoux and Van Daele [84, §4.2] also makes use of $\phi\big(x_d^{\mathrm{u}}, \kappa\big)$, but it has another vulnerability. Their idea is to use $\phi\big(x_d^{\mathrm{u}}, \kappa\big)$ to add a small correction to the phase propagation $\varepsilon\gamma$. However the way their equations handle the branch cuts of the (single variable) arctangent function is not numerically robust. These branch cuts occur in their case (but in our notation) when $\frac{\partial}{\partial x}\,\phi\big(x_d^{\mathrm{l}}, \kappa\big) = 0$ and when $\frac{\partial}{\partial x}\,\phi\big(x_d^{\mathrm{u}}, \kappa\big) = 0$. The compensation for these branch cuts takes place when $\vartheta\big(x_d^{\mathrm{u}}\big) - \big(\varepsilon\gamma + \vartheta\big(x_d^{\mathrm{l}}\big)\big) - \pi\big\lfloor \frac{1}{\pi}\big(\varepsilon\gamma + \vartheta\big(x_d^{\mathrm{l}}\big)\big)\big\rceil = \pm\frac{\pi}{2}$, where $\vartheta(x) := \operatorname{atan}\Big(\gamma\,\phi(x,\kappa)\Big/\big(\frac{\partial}{\partial x}\,\phi(x,\kappa)\big)\Big)$. Since this requires *different* comparisons between floating point numbers, rounding errors will very likely cause glitches near the branch cuts. Equation (3.18) instead uses $\phi\big(x_d^{\mathrm{u}}, \kappa\big)$ in essence to compute a 'signed parity' of the number of zero-crossings and then adds as the third term a 'correction' for the number of full cycles based on $\varepsilon\gamma$ minus a correction for the angles $\theta\big(x_d^{\mathrm{u}}\big) \in (-\pi, \pi]$ and $\theta\big(x_d^{\mathrm{l}}\big) \in (-\pi, \pi]$.[8] The merit is that our branch cut compensation always checks, up to multiplication by γ, the sign of the *same* floating point number that causes the branch cut.[9] Hence, if $\phi\big(x_d^{\mathrm{l}}, \kappa\big)$ is perturbed near zero, then $\theta\big(x_d^{\mathrm{l}}\big)$ and $[\![\phi\big(x_d^{\mathrm{l}}, \kappa\big) \geq 0]\!]$ in (3.18) are guaranteed to flip simultaneously. Likewise, if $\phi\big(x_d^{\mathrm{u}}, \kappa\big)$ is perturbed near zero, then $\theta\big(x_d^{\mathrm{u}}\big)$ and $[\![\phi\big(x_d^{\mathrm{u}}, \kappa\big) \geq 0]\!]$ are

[8] That is, the number between the rounding brackets $\lfloor\,\rceil$ in (3.18) should be an integer already, up to the effects of numerical error.

[9] Numerical multiplication by γ will not change the sign, since $\gamma > 3/\varepsilon \gg 0$, unless the data is unrealistically poorly normalized.

guaranteed to flip simultaneously. *This ensures that with our computation, no glitches occur near the branch cuts of the four-quadrant arctangent.*

The higher tail: $x \in (x_D^{\mathrm{u}}, \infty)$

In the interval $(x_D^{\mathrm{u}}, \infty)$ the reconstructed potential is zero. We first solve the Schrödinger equation (3.2) for the boundary condition $\boldsymbol{\phi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa)$ in order to find the number of zero-crossings. Since $q(x) = 0 \leq \kappa^2$, we see from (3.17) that the number of zero-crossings in this interval is either zero or one. This number can be computed by comparing the sign of $\phi(x_D^{\mathrm{u}}, \kappa)$ to the sign of $\lim_{x\to\infty} \phi(x, \kappa)$ as follows.

$$\begin{aligned} s(x_D^{\mathrm{u}}, \infty, \kappa) = {} & \left[\!\left[\begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) < 0 \wedge \begin{bmatrix} \kappa & 1 \end{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) > 0 \right]\!\right] + \ldots \\ & \left[\!\left[\begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) \geq 0 \wedge \begin{bmatrix} \kappa & 1 \end{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) < 0 \right]\!\right]. \end{aligned} \tag{3.20}$$

As discussed earlier in this section, the condition $\begin{bmatrix} \kappa & 1 \end{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) > 0$ is the only exception we must make to counting zero as a positive number, to obtain a consistent behaviour of the accounting function at the eigenvalues.

Next, we need to compute $\boldsymbol{\psi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa)$, so that we can compute $a(\kappa)$ from (3.10) at $x = x_D^{\mathrm{u}}$. We thus solve the Schrödinger equation (3.2) with a potential of zero for the boundary condition (3.5). It is readily verified that the solution in this interval is $\psi(x, \kappa) \equiv \exp(-\kappa x)$. Hence we obtain at $x = x_D^{\mathrm{u}}$

$$\boldsymbol{\psi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) = \begin{bmatrix} 1 \\ -\kappa \end{bmatrix} \exp(-\kappa x_D^{\mathrm{u}}). \tag{3.21}$$

3.3.3. Numerical computation of the gradient of the scattering parameter $a(\kappa)$

In order to find the roots of $a(\kappa)$, we will make use of the gradient $a'(\kappa) := \frac{\mathrm{d}}{\mathrm{d}\kappa} a(\kappa)$. If we take the derivative of (3.10) with respect to κ, we find

$$\begin{bmatrix} a(\kappa) \\ a'(\kappa) \end{bmatrix} \equiv \frac{1}{2} \begin{bmatrix} \boldsymbol{\psi}_{\mathrm{C}}^{\top}(x, \kappa) & \begin{bmatrix} 0 & 0 \end{bmatrix} \\ \frac{\partial}{\partial\kappa} \boldsymbol{\psi}_{\mathrm{C}}^{\top}(x, \kappa) & \boldsymbol{\psi}_{\mathrm{C}}^{\top}(x, \kappa) \end{bmatrix} \begin{bmatrix} 0 & \kappa^{-1} & 0 & 0 \\ -\kappa^{-1} & 0 & 0 & 0 \\ 0 & -\kappa^{-2} & 0 & \kappa^{-1} \\ \kappa^{-2} & 0 & -\kappa^{-1} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{\mathrm{C}}(x, \kappa) \\ \frac{\partial}{\partial\kappa} \boldsymbol{\phi}_{\mathrm{C}}(x, \kappa) \end{bmatrix}, \tag{3.22}$$

where we have used the convenient formulation of the scalar derivative of matrix products found in [17]. For the computation of $a'(\kappa)$ we need to extend the equations in Section 3.3.2 as follows. The derivatives with respect to κ of (3.12) and (3.21) are respectively

$$\frac{\partial}{\partial\kappa} \boldsymbol{\phi}_{\mathrm{C}}(x_1^{\mathrm{l}}, \kappa) = \begin{bmatrix} 1 \\ \kappa \end{bmatrix} x_1^{\mathrm{l}} \exp(\kappa x_1^{\mathrm{l}}); \tag{3.23}$$

$$\frac{\partial}{\partial\kappa} \boldsymbol{\psi}_{\mathrm{C}}(x_D^{\mathrm{u}}, \kappa) = \begin{bmatrix} -1 \\ \kappa \end{bmatrix} x_D^{\mathrm{u}} \exp(-\kappa x_D^{\mathrm{u}}). \tag{3.24}$$

Then $\frac{\partial}{\partial\kappa}\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)$ needs to be propagated from $x = x_1^{\mathrm{l}}$ to $x = x_D^{\mathrm{u}}$. We will do that according to [17, 104]. That is, we augment (3.13) to

$$\frac{\partial}{\partial x}\begin{bmatrix}\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)\\ \frac{\partial}{\partial\kappa}\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)\end{bmatrix} = \widetilde{\boldsymbol{A}}_{\mathrm{C}}(q_d,\kappa)\begin{bmatrix}\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)\\ \frac{\partial}{\partial\kappa}\boldsymbol{\phi}_{\mathrm{C}}(x,\kappa)\end{bmatrix}, \tag{3.25}$$

where

$$\widetilde{\boldsymbol{A}}_{\mathrm{C}}(q_d,\kappa) := \begin{bmatrix}\boldsymbol{A}_{\mathrm{C}}(q_d,\kappa) & \begin{bmatrix}0&0\\0&0\end{bmatrix}\\ \frac{\partial}{\partial\kappa}\boldsymbol{A}_{\mathrm{C}}(q_d,\kappa) & \boldsymbol{A}_{\mathrm{C}}(q_d,\kappa)\end{bmatrix} = \begin{bmatrix}0&1&0&0\\ \kappa^2-q_d&0&0&0\\ 0&0&0&1\\ 2\kappa&0&\kappa^2-q_d&0\end{bmatrix}. \tag{3.26}$$

Analogous to (3.15) it follows that

$$\begin{bmatrix}\boldsymbol{\phi}_{\mathrm{C}}(x_d^{\mathrm{u}},\kappa)\\ \frac{\partial}{\partial\kappa}\boldsymbol{\phi}_{\mathrm{C}}(x_d^{\mathrm{u}},\kappa)\end{bmatrix} = \widetilde{\boldsymbol{H}}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa)\begin{bmatrix}\boldsymbol{\phi}_{\mathrm{C}}(x_d^{\mathrm{l}},\kappa)\\ \frac{\partial}{\partial\kappa}\boldsymbol{\phi}_{\mathrm{C}}(x_d^{\mathrm{l}},\kappa)\end{bmatrix}, \tag{3.27}$$

where the augmented change of state matrix

$$\widetilde{\boldsymbol{H}}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa) := \exp\bigl(\varepsilon\,\widetilde{\boldsymbol{A}}_{\mathrm{C}}(q_d,\kappa)\bigr) = \begin{bmatrix}\boldsymbol{H}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa) & \begin{bmatrix}0&0\\0&0\end{bmatrix}\\ \frac{\partial}{\partial\kappa}\boldsymbol{H}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa) & \boldsymbol{H}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa)\end{bmatrix}$$

$$= \begin{bmatrix}\cos(\gamma\varepsilon) & \varepsilon\operatorname{sinc}(\gamma\varepsilon) & 0 & 0\\ -\gamma\sin(\gamma\varepsilon) & \cos(\gamma\varepsilon) & 0 & 0\\ \kappa\varepsilon^2\operatorname{sinc}(\gamma\varepsilon) & \frac{\kappa\varepsilon}{\gamma^2}\bigl(\operatorname{sinc}(\gamma\varepsilon)-\cos(\gamma\varepsilon)\bigr) & \cos(\gamma\varepsilon) & \varepsilon\operatorname{sinc}(\gamma\varepsilon)\\ \kappa\varepsilon(\operatorname{sinc}(\gamma\varepsilon)+\cos(\gamma\varepsilon)) & \kappa\varepsilon^2\operatorname{sinc}(\gamma\varepsilon) & -\gamma\sin(\gamma\varepsilon) & \cos(\gamma\varepsilon)\end{bmatrix}. \tag{3.28}$$

A root finder that makes use of the gradient usually requires fewer iterations, but each iteration is computationally costlier. In order to determine if it pays off in this case to use the gradient, we need to determine if the first outweighs the latter. In Table 3.2 we have therefore counted the number of FLoating point OPerations (FLOPs) to compute respectively the two by two matrix $\boldsymbol{H}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa)$ and the matrix vector multiplication in (3.15), or the four by four matrix $\widetilde{\boldsymbol{H}}_{\mathrm{C}}(x_d^{\mathrm{l}},x_d^{\mathrm{u}},\kappa)$ and the matrix vector multiplication in (3.27). Since the number of repetitions of this part of the computation scales proportionally to the (usually large) number of samples D, this part dominates the computational cost of evaluating (3.10) or (3.22) respectively. For the basic operations in Table 3.2 we assumed the number of FLOPs that was estimated in [68, p. 5]. In practice FLOP counts vary between different implementations, programming languages, compilers and hardware architectures. Therefore, these results should be treated as rough estimates. The results in Table 3.2 show that an evaluation of $a(\kappa)$ alone takes roughly $35D$ FLOPs plus overhead. The computation of $a'(\kappa)$ at the same value of κ takes roughly $24D$ FLOPs extra. Hence, we estimate that every iteration of a gradient based root finder is roughly 70 % more expensive than an iteration of a gradient free root finder.

3.3.4. Upgrade to fourth order accuracy

In Section 3.3.2 we approximated the scattering parameter $a(\kappa)$ of a potential $q(x)$. Thereto we used an exact computation (in infinite precision) for the reconstructed potential $\hat{q}(x)$,

Calculation	Operation(s)	FLOPs
$\underline{\gamma^2} \leftarrow \underline{q_d} - \underline{\kappa} \cdot \underline{\kappa}$	$1 \times$ and $1 \pm$	2
$\underline{\gamma} \leftarrow \sqrt{\underline{\gamma^2}}$	$1\ \surd$	4
$\underline{\gamma\varepsilon} \leftarrow \underline{\gamma} \cdot \underline{\varepsilon}$	$1 \times$	1
$\underline{\cos(\gamma\varepsilon)} \leftarrow \cos\big(\underline{\gamma\varepsilon}\big)$	1 cos	8
$\underline{\sin(\gamma\varepsilon)} \leftarrow \sin\big(\underline{\gamma\varepsilon}\big)$	1 sin	8
$\underline{\varepsilon\,\mathrm{sinc}(\gamma\varepsilon)} \leftarrow \underline{\sin(\gamma\varepsilon)} \,/\, \underline{\gamma}$	$1 \div$	4
$\underline{-\gamma\sin(\gamma\varepsilon)} \leftarrow -\underline{\gamma} \cdot \underline{\sin(\gamma\varepsilon)}$	$1 \times$ and $1 \pm$	2
Right-multiply a 2×2 matrix by a vector	$4 \times$ and $2 \pm$	6
Total FLOPs per κ-sample per x-sample to compute $a(\kappa)$:		35
$\underline{\kappa\varepsilon} \leftarrow \underline{\kappa} \cdot \underline{\varepsilon}$	$1 \times$	1
$\underline{\kappa\varepsilon^2\,\mathrm{sinc}(\gamma\varepsilon)} \leftarrow \underline{\kappa\varepsilon} \cdot \underline{\varepsilon\,\mathrm{sinc}(\gamma\varepsilon)}$	$1 \times$	1
$\underline{\kappa\varepsilon\,\mathrm{sinc}(\gamma\varepsilon)} \leftarrow \underline{\kappa} \cdot \underline{\varepsilon\,\mathrm{sinc}(\gamma\varepsilon)}$	$1 \times$	1
$\underline{\kappa\varepsilon\cos(\gamma\varepsilon)} \leftarrow \underline{\kappa\varepsilon} \cdot \underline{\cos(\gamma\varepsilon)}$	$1 \times$	1
$\kappa\varepsilon(\mathrm{sinc}(\gamma\varepsilon) + \cos(\gamma\varepsilon)) \leftarrow \underline{\kappa\varepsilon\,\mathrm{sinc}(\gamma\varepsilon)} + \underline{\kappa\varepsilon\cos(\gamma\varepsilon)}$	$1 \pm$	1
$\underline{\frac{\kappa\varepsilon}{\gamma^2}\big(\mathrm{sinc}(\gamma\varepsilon) - \cos(\gamma\varepsilon)\big)} \leftarrow \big(\underline{\kappa\varepsilon\,\mathrm{sinc}(\gamma\varepsilon)} - \underline{\kappa\varepsilon\cos(\gamma\varepsilon)}\big) \,/\, \underline{\gamma^2}$	$1 \pm$ and $1 \div$	5
Right-multiply a 2×4 matrix by a vector	$8 \times$ and $6 \pm$	14
Extra FLOPs per κ-sample per x-sample to compute also the gradient $a'(\kappa)$:		24

Table 3.2: Count of the amount of FLOPs that is required per x-sample to compute $a(\kappa)$ at one sample κ, and the extra amount for computing also the gradient $a'(\kappa)$. The underlined expressions should be interpreted as a single variable with a known quantity. Initially, only the potential sample q_d, the spatial step size ε, and the spectral parameter κ are known. The upper part of the table regards the computation of $\boldsymbol{H}_{\mathrm{C}}\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big)$ according to (3.16), and the multiplication in (3.15). This appears to take 35 FLOPs. The lower part of the table regards the extra operations for the computation of $\widetilde{\boldsymbol{H}}_{\mathrm{C}}\big(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\big)$ according to (3.28), and the multiplication in (3.27). Once the values of the upper part of the table are known, this appears to take 24 additional FLOPs. Hence, the whole computation of $a(\kappa)$, $s(-\infty, \infty, \kappa)$ and $a'(\kappa)$ takes 59 FLOPs per x-sample, per κ sample. The overhead, that does not depend on the number of samples, is ignored. The number of FLOPs that are required for each basic operation, are obtained from [68, p. 5]. In practice FLOP counts vary between different implementations, programming languages, compilers and hardware architectures. Therefore, these results should be treated as rough estimates.

defined in (3.8). This piecewise constant reconstruction enabled the relatively simple computation in (3.15) and (3.16). With respect to the true potential $q(x)$, (3.15) implements the exponential midpoint rule, which is also known as $\mathrm{CF}_1^{[2]}$.[10] The exponential midpoint rule provides an approximation of order two in the step size [16, p. 244]. Consequently, we obtain from (3.10) $a(\kappa) + O(\varepsilon^3)$. The same considerations and error order apply to the computation of the gradient $a'(\kappa)$ according to (3.22) in Section 3.3.3.

If the potential is sufficiently smooth, a more accurate reconstruction of $a(\kappa)$ and $a'(\kappa)$ can be obtained by using a higher order integrator. In particular, we will use the fourth order integrator $\mathrm{CF}_2^{[4]}$ [16, Eq. 12]:

$$\boldsymbol{H}_{\mathrm{C}}\left(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\right) := \exp\left(\frac{\varepsilon}{2}\boldsymbol{A}_{\mathrm{C}}(\check{q}_{2d}, \kappa)\right)\exp\left(\frac{\varepsilon}{2}\boldsymbol{A}_{\mathrm{C}}(\check{q}_{2d-1}, \kappa)\right), \tag{3.29}$$

where

$$\begin{bmatrix}\check{q}_{2d-1}\\ \check{q}_{2d}\end{bmatrix} := \frac{1}{2\sqrt{3}}\begin{bmatrix}\sqrt{3}+2 & \sqrt{3}-2\\ \sqrt{3}-2 & \sqrt{3}+2\end{bmatrix}\begin{bmatrix}q\left(x_d^{\mathrm{m}} - \varepsilon/(2\sqrt{3})\right)\\ q\left(x_d^{\mathrm{m}} + \varepsilon/(2\sqrt{3})\right)\end{bmatrix}. \tag{3.30}$$

To obtain the non-equispaced samples of $q(x)$ that are required in (3.30), we use band limited interpolation, as proposed in [32]. Since this particular non-equispaced grid consists of two equispaced grids, the interpolation requires only three[11] Fast Fourier Transform (FFT) computations, with a complexity of $O(D\log(D))$, and is thus computationally cheap.

We see that the computation of $\boldsymbol{H}_{\mathrm{C}}\left(x_d^{\mathrm{l}}, x_d^{\mathrm{u}}, \kappa\right)$ in (3.29) has the same structure as the one in (3.16), for two adjacent steps of step size $\varepsilon/2$. That is, the approximation of $a(\kappa)$ and $a'(\kappa)$ that we obtain from the $\mathrm{CF}_2^{[4]}$ integrator are (in infinite precision) the exact results for the preprocessed potential

$$\begin{aligned}\check{q}(x) &:= \sum_{d=1}^{D}\left(\check{q}_{2d-1}[\![x_d^{\mathrm{l}} < x < x_d]\!] + \check{q}_{2d}[\![x_d < x < x_d^{\mathrm{u}}]\!]\right)\\ &= \sum_{d=1}^{2D}\check{q}_d[\![\check{x}_d^{\mathrm{l}} < x < \check{x}_d^{\mathrm{u}}]\!],\end{aligned} \tag{3.31}$$

where $\check{x}_d^{\mathrm{l}} := x_1^{\mathrm{l}} + (d-1)\frac{\varepsilon}{2}$ and $\check{x}_d^{\mathrm{u}} := \check{x}_d^{\mathrm{l}} + \frac{\varepsilon}{2}$.

Since we can interpret the approximations by the $\mathrm{CF}_2^{[4]}$ integrator as the exact results for a real piecewise constant potential, the computations of the accounting function in Section 3.3.2 still apply. We only have to use the preprocessed potential samples $\check{q}_d(x)$, half the step size ε, and double the number of samples D.

It is natural to ask if this approach can be extended to integrators of orders above four. Unfortunately, higher order CF integrators require complex coefficients or negative step sizes

[10] CF stands for commutator-free quasi-Magnus exponential integrators. The superscript number denotes the order of accuracy in the step size ε. The subscript denotes the number of matrix exponentials per step.

[11] If it is permissible to shift the truncation window from $x \in [x_1^{\mathrm{l}}, x_D^{\mathrm{u}}]$ to $x \in [x_1^{\mathrm{l}} \pm \varepsilon/(2\sqrt{3}), x_D^{\mathrm{u}} \pm \varepsilon/(2\sqrt{3})]$, the number of required Fast Fourier Transform (FFT) operations can be reduced to two, by using (instead of two shifted grids) the original grid on which the samples are known together with one grid that is shifted by $\mp\varepsilon/\sqrt{3}$.

[11, 16]. Therefore these break the aforementioned interpretability on which our approach relies. Integrators that assume a piecewise polynomial approximation of the potential lead to more complicated piecewise expressions for the trajectory than (3.17). Thus in (3.18) the phase propagation changes from $\varepsilon\gamma$ to $\varepsilon\gamma + O(\varepsilon^2)$, cf. [84]. If these higher order terms amount to π or more, then (3.18) will no longer count the correct number of zero crossings. Especially when some of the accuracy gain of such a higher order integrator is traded against a larger step size ε, these higher order terms might become significant. Dealing with this effect in a numerically robust way will need further investigation.

3.4. Algorithm to compute the eigenvalues

In this section we present the algorithm that we propose for computing the eigenvalues of the KdV. We have seen in Section 3.2 that the spectral parameter $a(\kappa)$ and the accounting function $s(-\infty, \infty, \kappa)$ both contain the full information on the eigenvalues K_n. Numerically, we can only compute these functions for one sample of κ at a time. Therefore we need a strategy to choose these samples and a system to accumulate the information that we obtain at each new sample. We start with the latter in Section 3.4.1. In Sections 3.4.2 and 3.4.3 we proceed with respectively an existing and the proposed strategy to choose the samples κ. The existing strategy, bisection, both serves as a benchmark algorithm in Section 3.5 and as a stepping stone towards the exposition of the proposed algorithm.

3.4.1. Bounds on the eigenvalues

Our aim is to find the eigenvalues K_n from a numerical algorithm. More precisely, we first want to determine the number of eigenvalues N. Thereto we compute $N \leftarrow s(-\infty, \infty, 0)$. Next, we want to find each eigenvalue K_n with $n \in \{1, 2, \dots, N\}$ up some user-selectable tolerance Δ, with respect to the eigenvalues of the discretized potential. That is, we want to find N lower bounds K_n^{l} and N upper bounds K_n^{u}, such that $\hat{K}_n \in (K_n^{\mathrm{l}}, K_n^{\mathrm{u}})$ and $K_n^{\mathrm{u}} - K_n^{\mathrm{l}} \leq \Delta$ for all $n \in \{1, 2, \dots, N\}$. The interval $(K_n^{\mathrm{l}}, K_n^{\mathrm{u}})$ is called a *bracket* (of the nth eigenvalue). Furthermore, we store the value of the scattering parameter $a(\kappa)$ at all bounds. In the end we will use those residuals to select for each eigenvalue a best guess between the lower and upper bound, based on a minimal residual criterion. The bounds together with the residuals form our current knowledge about the eigenvalues. The initial bounds are the same for all eigenvalues: $K_n^{\mathrm{l}} \leftarrow 0$ and $K_n^{\mathrm{u}} \leftarrow \sqrt{\max(0, \max_d \check{q}_d)}$. (See Section 3.2.2 and (3.31).) If $\max_d \check{q}_d \leq 0$, so all samples are non-positive, it follows immediately that the discrete spectrum is an empty set. In that case there is no need to do any other computations than this simple check.

To increase our knowledge about the eigenvalues we proceed as follows. We select a value for κ and then evaluate the scattering parameter $\alpha \leftarrow a(\kappa)$ and the accounting function $\varsigma \leftarrow s(-\infty, \infty, \kappa)$. In Fig. 3.1 we show how we use these results to update our knowledge about the eigenvalues. Since the accounting function signifies the number of greater eigenvalues, its value ς tells us that $K_n < \kappa$ for $n \in \{1, 2, \dots N - \varsigma\}$. Therefore, κ is an upper bound on this subset of the eigenvalues. If this upper bound is tighter than the previously known upper bound, we overwrite it. Similarly κ is a lower bound on K_n for $n \in \{N - \varsigma + 1, \dots, N - 1, N\}$. If this bound is tighter than the previously known lower bound, we overwrite it.

If $\alpha = 0$, we know that κ is an eigenvalue. In that case $\phi(x, \kappa) \to 0$ as $x \to \infty$, but

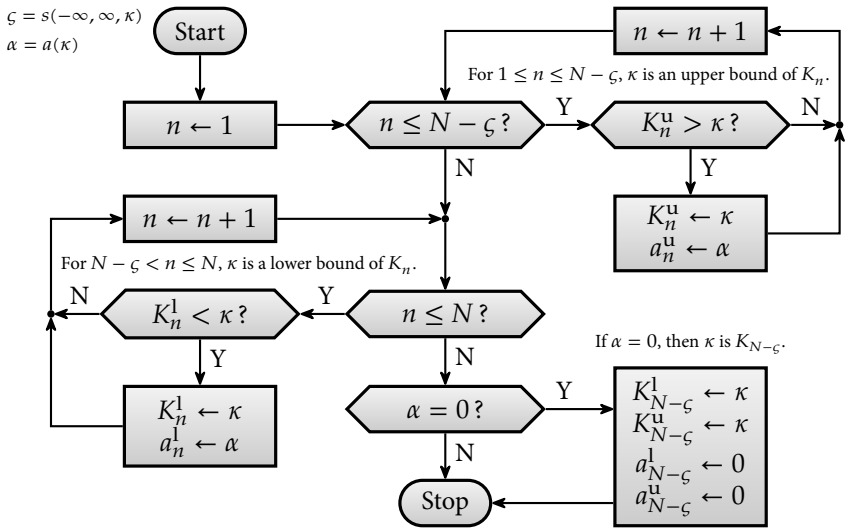


Figure 3.1: Flow chart of the update of the brackets $(K_n^{\mathrm{l}}, K_n^{\mathrm{u}})$ and the respective residuals a_n^{l} and a_n^{u} at the interval boundaries. This flow chart is the specification of the lower left block in Fig. 3.2 and block 6 in Fig. 3.3. This update algorithm makes use of the total number of eigenvalues N, the current sample κ, the spectral parameter at this sample $\varsigma = s(-\infty, \infty, \kappa)$, and the residual at this sample $\alpha = a(\kappa)$. Every bracket is checked either in the upper loop for a tightening of the upper bound, or in the lower loop for a tightening of the lower bound. The exceptional case $\alpha = 0$ indicates that κ is an eigenvalue. This gets a special treatment, in the lower right block.

as emphasized in Section 3.3.2, we do not count this limit as a zero-crossing. Therefore $\phi(x, K_{N-\varsigma})$ has ς zero-crossings. Thus, if $\alpha = 0$ we infer that $\kappa = K_{N-\varsigma}$. We store this conclusion by setting $K_{N-\varsigma}^{\mathrm{l}} \leftarrow \kappa$ and $K_{N-\varsigma}^{\mathrm{u}} \leftarrow \kappa$.[12] The special case $K_n^{\mathrm{l}} = K_n^{\mathrm{u}}$ should thus be considered as a closed rather than an open interval.

In order to shrink a particular interval $(K_n^{\mathrm{l}}, K_n^{\mathrm{u}})$, the sample κ must lie in this interval. Hence, if κ does not lie between the currently known bounds of any eigenvalue, the computations $\alpha \leftarrow a(\kappa)$ and $\varsigma \leftarrow s(-\infty, \infty, \kappa)$ will not result in any progress regarding our knowledge about the eigenvalues. We say that κ is proper if and only if $\kappa \in \bigcup_{n=1}^{N} (K_n^{\mathrm{l}}, K_n^{\mathrm{u}})$. We use the properness of κ as a conditional in the proposed algorithm to prevent such idle iterations.

3.4.2. Bisection

Osborne [102] proposed to search for the eigenvalues by means of bisection. An implementation of that is shown in Fig. 3.2. In addition to the variables that are explained in Section 3.4.1, we keep track of a boolean array $\{\mathfrak{b}_n\}$ that stores if the n-th eigenvalue is localized. It is thus updated as $\mathfrak{b}_n \leftarrow (K_n^{\mathrm{u}} - K_n^{\mathrm{l}} \leq \Delta)$, for $n \in \{1, 2, \ldots, N\}$. As long as there are eigenvalues left to localize, we set κ by bisecting one of their intervals. After computing $\alpha \leftarrow a(\kappa)$ and

[12] In the flowchart in Fig. 3.1, $K_{N-\varsigma}^{\mathrm{u}}$ is already set to κ before reaching the conditional $\alpha = 0$. Therefore technically only the lower bound $K_{N-\varsigma}^{\mathrm{l}}$ still has to be overwritten in the lower right block. The current representation is chosen for conceptual clarity.

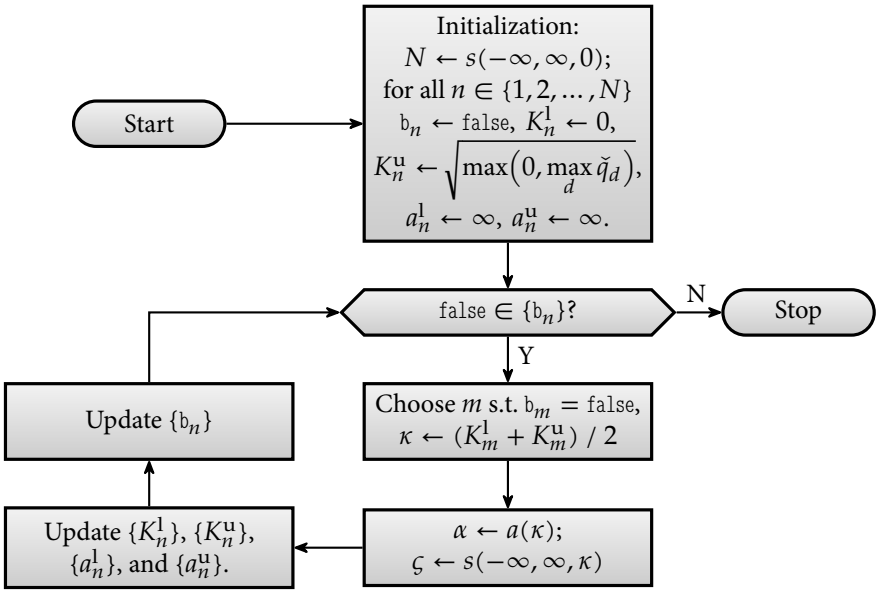


Figure 3.2: Top level flow chart of the bisection benchmark algorithm.

$\varsigma \leftarrow s(-\infty, \infty, \kappa)$, we update the bounds of all the eigenvalues as described in Section 3.4.1 and Fig. 3.1. Then we update $\{\mathtt{b}_n\}$ and repeat. Finally, we select as best guess for each eigenvalue the bound with the lowest residual. That is, $\hat{K}_n := K_n^{\mathrm{l}} [\![|a_n^{\mathrm{l}}| < |a_n^{\mathrm{u}}|]\!] + K_n^{\mathrm{u}} [\![|a_n^{\mathrm{l}}| \geq |a_n^{\mathrm{u}}|]\!]$.

3.4.3. Proposed algorithm

Bisection is a simple technique, but it has only linear convergence. Christov [33] proposed to use an Illinois type *regula falsi* root-finder instead, to speed up the convergence. However, this technique requires initial brackets that contain exactly one eigenvalue, for which he resorted to a grid search. Root finding based on $a(\kappa)$ and $a'(\kappa)$ with the NR technique enjoys a quadratic convergence. However, an open root-finder like NR is more difficult to combine with SL oscillation theory than a bracketing root-finder like bisection. First, one has to find initial guesses within the basins of attraction of every eigenvalue rather than an initial bracket. Second, one can only establish convergence to the nth eigenvalue if the corresponding unit step of the accounting function is sampled both below and above the eigenvalue at a distance smaller than the tolerance Δ. A possible strategy is to apply a two-step procedure: First use bisection to find a reasonably tight bracket for each eigenvalue and refine thereafter with a NR procedure [69, 83, 85]. However, this combines the disadvantages of both techniques: The slow convergence of bisection and the difficulty of keeping NR iterations within a bracket.

We instead propose a hybrid algorithm that combines quadratic convergence (for most iterations) with the guarantee that all eigenvalues are localized. In short, we follow the NR procedure, unless its update results in an improper value κ. (See Section 3.4.1 for the definition of improper.) Only in that case and at the start we apply one bisection step to obtain a

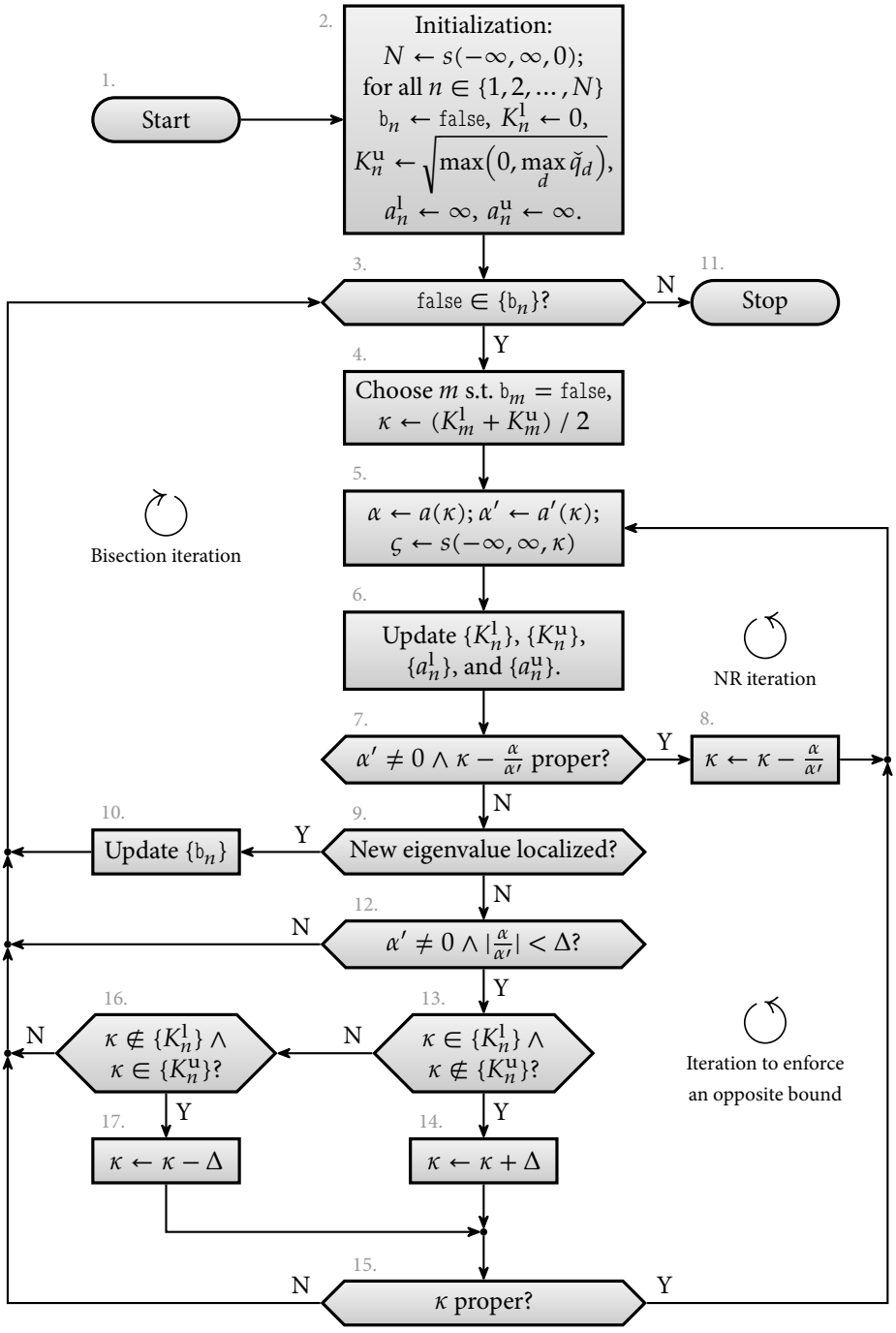


Figure 3.3: Top level flow chart of the proposed algorithm.

new initial value for the NR procedure. The algorithm stops when each eigenvalue is localized. Hence we do not select a priori the order in which we search for the eigenvalues. Thus we avoid the difficulty of finding an initial guess in the basin of attraction of that eigenvalue. Instead we allow the NR procedure to converge to any eigenvalue. If it happens to have started within the basin of attraction of an eigenvalue that is already localized, it will update to an improper value of κ, typically already in the first iteration. Then we use a bisection step instead.

A flowchart of the proposed algorithm is shown in Fig. 3.3. The blocks 1 up to and including 11 form the basis of the algorithm. Let us describe these blocks first. Thereafter we will explain the purpose and working of blocks 12 up to and including 17.

1. The input of the proposed algorithm is the array of samples $\check{q}_d$ that results from (3.30). Furthermore, the tolerance on the eigenvalues, Δ, must be provided. One should keep in mind that this is the maximum deviation of the returned eigenvalues with respect to the eigenvalues of the discretized potential. The error that is caused by the sampling and reconstruction of the potential cannot be reduced by choosing Δ smaller and smaller.

2. The initialization is the same as for the bisection algorithm. See Sections 3.4.1 and 3.4.2.

3. While there are eigenvalues left to localize, continue searching.

4. We select any unlocalized eigenvalue and bisect its currently known bracket. It makes little difference which of the remaining brackets we choose to bisect. We choose the lowest m for which $\mathtt{b}_m = \mathtt{false}$.

5. At the current value κ we evaluate $\alpha \leftarrow a(\kappa)$, $\alpha' \leftarrow a'(\kappa)$, and the accounting function $\varsigma \leftarrow s(-\infty, \infty, \kappa)$. This is the most computationally expensive step, so we will count every time the algorithm enters this block as one iteration.

6. We update the brackets of all eigenvalues as described in Section 3.4.1 and Fig. 3.1.

7.–8. If κ is not a stationary point, we can compute the NR update of κ. That is, $\kappa \leftarrow \kappa - \alpha \,/\, \alpha'$. If the update succeeds, the NR descend continues from the updated value κ. However, if κ would become improper after this update (see Section 3.4.1), we break the NR cycle before the actual update.

9.–10. We consider an eigenvalue as localized if $K_n^{\mathrm{u}} - K_n^{\mathrm{l}} \leq \Delta$. If that condition is met for any eigenvalue n whereas $\mathtt{b}_n$ is still $\mathtt{false}$, we set $\mathtt{b}_n$ to $\mathtt{true}$ and return. Please note that this is not necessarily the eigenvalue m that was selected in block 4, and that in rare cases we may find more than one eigenvalue at once.

11. When all eigenvalues are localized up to an interval of length Δ at most, we stop searching. For each eigenvalue $n \in \{1, 2, \dots, N\}$ we return the bound with the smallest residual as the best guess. That is, $\hat{K}_n := K_n^{\mathrm{l}} [\![|a_n^{\mathrm{l}}| < |a_n^{\mathrm{u}}|]\!] + K_n^{\mathrm{u}} [\![|a_n^{\mathrm{l}}| \geq |a_n^{\mathrm{u}}|]\!]$.

If we would only use the part of the algorithm that is described in blocks 1 up to and including 11, we would already be able to localize all eigenvalues reliably. However it has a

flaw that makes it fall back to linear convergence. This is because the NR algorithm tends to converge to a root monotonically: either from above or from below. Let us consider the spectral parameter $a(\kappa)$ around a root K_n. More precisely, we look at a neighbourhood of K_n where $a(\kappa)$ is either convex or concave. In the convex case, every tangent line to $a(\kappa)$ in this neighbourhood crosses zero at $\kappa > K_n$. This crossing represents the NR update. That means that once κ enters this neighbourhood, all following updates satisfy $\kappa > K_n$. Thus, ignoring finite precision effects, the algorithm converges monotonically to K_n from above. Upon termination, we will thus have found a tight upper bound of that root. However, since we require that the bracket $(K_n^{\mathrm{l}}, K_n^{\mathrm{u}})$ is smaller than Δ before we consider an eigenvalue localized, we also need a sufficiently tight lower bound. Unfortunately, this lower bound can only be found with bisection, since every nearby lower NR iteration would immediately jump over the already known upper bound. *Mutadis mutandis*, the same happens if $a(\kappa)$ is concave in a neighbourhood of K_n. Therefore, the algorithm as a whole falls back on linear convergence.[13]

The blocks 12 up to and including 17 in Fig. 3.3 are added to avoid this. The principle is that if the NR descend terminates on a tight upper bound, we decrement κ by Δ to enforce a tight enough lower bound. Vice versa, we increment κ by Δ in case of a tight lower bound. In detail:

12. When the NR cycle terminated without localizing an eigenvalue, there are two possibilities. Firstly, we might not have started within the basin of attraction of any eigenvalue that was left to localize. In that case we need to return, to find a new starting point by bisection. Secondly, due to numerical effects, the NR update will always become improper near a root of $a(\kappa)$. When that happens, we have localized one tight bound of that root. In that case we want to enforce the opposite bound, as explained before. To distinguish between these two cases, we look at the magnitude of the NR update that would have taken place.

13.–17. If the value of κ became a lower bound for one or more eigenvalues (in block 6), we expect at this point that this bound is tight for at least one of these eigenvalues. In order to guarantee that this eigenvalue is localized, we try to put an upper bound at $\kappa + \Delta$. We expect $\kappa + \Delta$ to be proper in this case, but we test it nevertheless (in block 15), to guard against unforeseen behaviour. Similarly, if he value of κ became an upper bound for one or more eigenvalues, we expect at this point that this bound is tight for at least one of these eigenvalues. In order to guarantee that this eigenvalue is localized, we try to put a lower bound at $\kappa - \Delta$. We do not give any special treatment to the rare case in which κ is both a lower bound of one eigenvalue and an upper bound of another.

3.5. Examples

In this section we will demonstrate the proposed algorithm. We select six different vanishing potentials and compute their eigenvalues with the proposed algorithm and with three

[13] Thanks to finite precision effects, the NR descend sometimes finds a sufficiently small interval anyway. That happens if one of the final updates numerically jumps over the root. However, we want to enforce quadratic convergence in the majority of the cases.

benchmark algorithms. We first compare the accuracy that the algorithms achieve as a function of the tolerance Δ. Next, we compare in detail the accuracy and computational cost at an ideal fixed value Δ.

3.5.1. Benchmark algorithms

We compare the proposed algorithm to two versions of the bisection algorithm that was described in Section 3.4.2 and Fig. 3.2. The difference between the two versions lies in the reconstruction of the potential. By using the reconstruction in (3.8) we obtain the second order version. We will refer to this benchmark algorithm as Alg. B2. By using the reconstruction in (3.30) and (3.31) we obtain the fourth order version. We refer to this benchmark algorithm as Alg. B4. Both Algs. B2 and B4 use the *proposed* root counter that is described in Section 3.3.2. We will also test one of the examples with Alg. B2s. That algorithms is equal to Alg. B2, except that it uses the root counter that was proposed by Osborne [102]. This particular example shows the problem with that root counter. We remark that we cannot use the complete algorithm from [102] as a benchmark, since we must at least adapt it to data with vanishing instead of periodic boundary conditions.

Furthermore, we compare the proposed algorithm to Matslise 2.0 [85].[14] We refer to this eighteenth order benchmark algorithm as Alg. MS18. This algorithm needs an analytic expression of the input signal $q(x)$, whereas our algorithm requires samples. Therefore we use band limited interpolation. That is, we pass an expression of the form

$$q_{\mathrm{MS}}(x) := -\left(\alpha_0 + \sum_{d=1}^{\lfloor D/2 \rfloor} \alpha_d \cos\left(\frac{2\pi d}{\varepsilon D}(x - x_1)\right) + \cdots \right. \\ \left. \sum_{d=1}^{\lceil D/2 \rceil - 1} \beta_d \sin\left(\frac{2\pi d}{\varepsilon D}(x - x_1)\right)\right) [\![x > x_1^{\mathrm{l}}]\!][\![x < x_D^{\mathrm{u}}]\!], \tag{3.32}$$

for which we calculate the coefficients α_d and β_d with the FFT of the sampled input signal. The minus sign is required due to the different parametrisation of the Schrödinger equation. Matslise does not make use of the step size ε of the data, but automatically chooses a mesh. Matslise only allows us to specify a tolerance for the squares of the eigenvalues, not for the eigenvalues themselves. We use the same tolerance Δ as for the other algorithms, but point out that it has to be interpreted differently for Matslise. We also pass to Matslise the jump points $x = x_1^{\mathrm{l}}$ and $x = x_D^{\mathrm{u}}$. We obtain the eigenvalues as $K_n = \sqrt{-E_n}$, where E_n are the squared eigenvalues that are returned by Matslise.

In this benchmark, the proposed method is compared with other shooting methods. A comparison with finite dimensional approximation methods would be more involved due to several characteristics that differ from the shooting methods. It is therefore beyond the scope of this work.

3.5.2. Example potentials

We will compute the eigenvalues of the six potentials that are shown in Fig. 3.4. The exact description of these potentials is as follows.

[14] We considered Matslise 3.0/Pyslise [14] as well, but this package is currently still under development [13]. In particular, we found that boundary conditions at infinity are not yet supported.

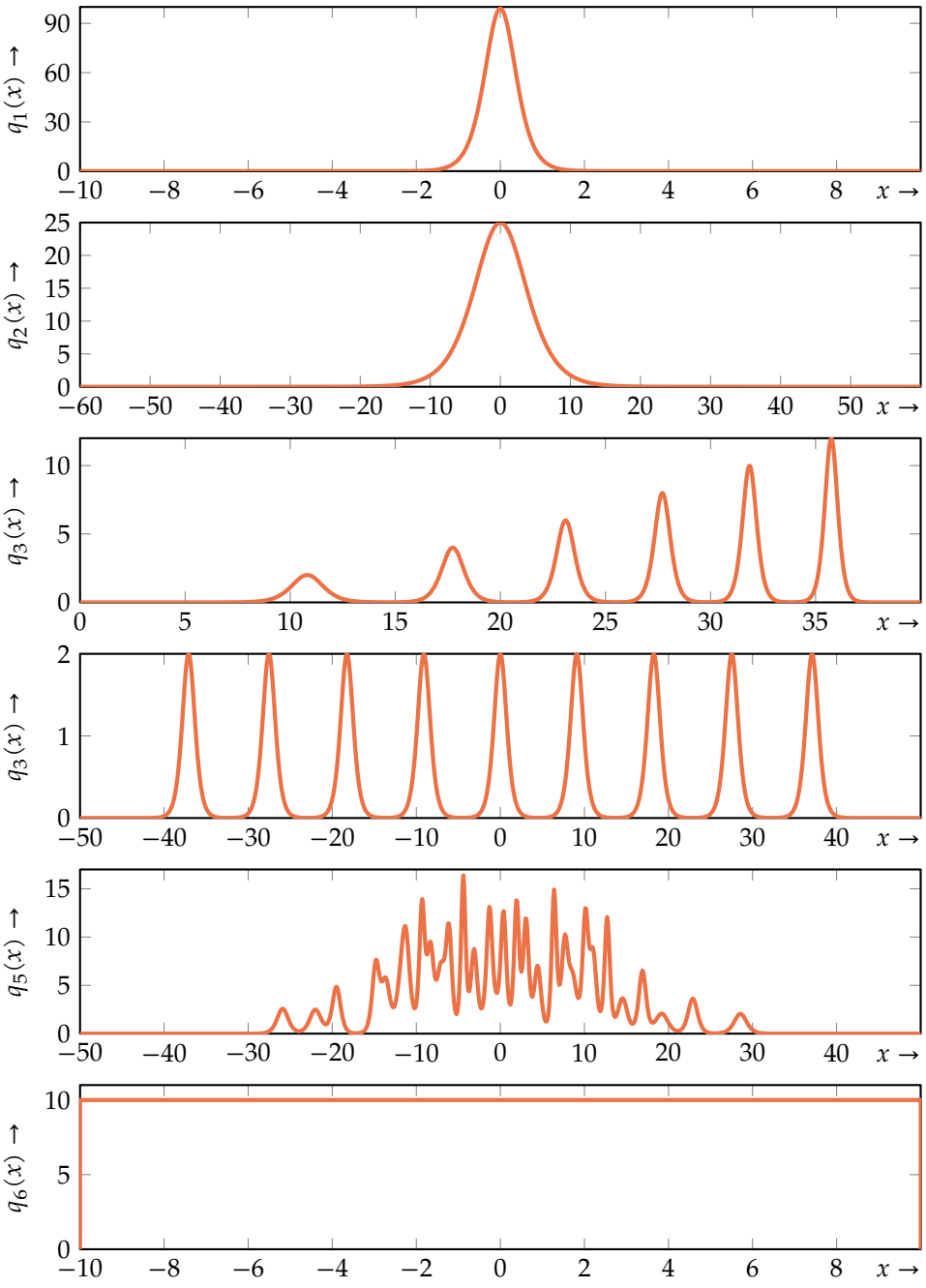


Figure 3.4: Potentials of which the eigenvalues are calculated to demonstrate the proposed algorithm. See the main text for their definitions.

$q_1(x)$ We define $q_1(x) := 99\,\mathrm{sech}^2(2x)$. This is a non-reflectionless[15] potential with five eigenvalues: $K_1 = 1, K_2 = 3, K_3 = 5, K_4 = 7$, and $K_5 = 9$ [79, §2.5].

$q_2(x)$ We define $q_2(x) := 24.99\,\mathrm{sech}^2(x/5)$. This is a non-reflectionless potential with 25 eigenvalues: $K_n = 0.2n - 0.1$ for $n \in \{1, 2, \dots, 25\}$ [79, §2.5].

$q_3(x)$ The reflectionless potential $q_3(x)$ shows a typical far-field pattern with six separated solitons. We selected its eigenvalues as $K_n := \sqrt{n}$, where $n \in \{1, 2, \dots 6\}$. The norming constants are set to $b(K_n) := (-1 \times 10^{12})^n$. The required samples of this potential are computed numerically with the Crum transform algorithm of Prins and Wahls [112].

$q_4(x)$ The reflectionless potential $q_4(x)$ forms a wave train with nearly equal eigenvalues. The eigenvalues are selected as $K_n := 1 + (n - 5)10^{-4}$, where $n \in \{1, 2, \dots 9\}$. The norming constants are set to $b(K_n) := -(-1)^n$. The required samples of this potential are computed numerically with the same Crum transform algorithm [112].

$q_5(x)$ Potential $q_5(x)$ is a reflectionless potential with thirty eigenvalues. The eigenvalues are selected as $\{K_n\} := \{2 - \cos(i) \mid i \in \{1, 2, \dots 30\}\}$. We sort this set of eigenvalues such that $K_1 < K_2 < \cdots < K_{30}$, Then, the norming constants are set to $b(K_n) := (-1)^n 10^{6\sin(n)}$, where n is the index after sorting. The required samples of this potential are once again computed numerically with the Crum transform algorithm from [112].

$q_6(x)$ This is a rectangular potential which is defined as $q_6(x) := 10\,[\![-10 \le x \le 10]\!]$. It is non-reflectionless and it has 21 eigenvalues. There exists no closed form expression to compute these eigenvalues. However, they can be approximated to any desired finite precision with e.g. (5.61) on Page 107.

For numerical processing, all potentials are truncated to the respective intervals shown in Fig. 3.4. Potentials $q_1(x)$ till $q_5(x)$ are sampled on a uniform grid of 10^4 samples on this interval, such that for example $x_1^{\mathrm{l}} = -10$ and $x_D^{\mathrm{u}} = 10$ for potential $q_1(x)$. Potential $q_6(x)$ is sampled with only two samples, at $x = -5$ and $x = 5$ respectively.

3.5.3. Error measures

Since we know the true eigenvalues of each of the example potentials, we can calculate the error of the computed eigenvalues. In some cases the numerical algorithms return a different number of eigenvalues than the true number of eigenvalues. We will always compare every true eigenvalue to the computed eigenvalue with the same number of zero-crossings of the eigenfunction. Thus, the greatest true eigenvalue is compared to the greatest computed eigenvalue etcetera. If the number of computed eigenvalues $\hat{N}$ is larger than the true number of eigenvalues N, we compute a separate error measure that compares the spurious $\hat{N} - N$ smallest eigenvalues to zero. The thought behind this is that an artificial eigenvalue

[15] A reflectionless potential is a potential of which the continuous KdV-NFT spectrum is zero, such that the signal only consists of the (possibly interacting) solitions that are represented by the eigenvalues.

of (nearly) zero corresponds to a part of the spectrum with (nearly) zero mass, momentum, and energy. We thus define

$$\underset{n\leq N}{\mathrm{RMS}}\{\hat{K}_{n+\hat{N}-N}-K_n\} := \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{K}_{n+\hat{N}-N}-K_n)^2}, \tag{3.33}$$

$$\underset{n>N}{\mathrm{RMS}}\{\hat{K}_{n+\hat{N}-N}\} := \sqrt{\frac{1}{\hat{N}-N}\sum_{n=1}^{\hat{N}-N}\hat{K}_n^2}, \quad \text{if } \hat{N}>N. \tag{3.34}$$

3

In addition, we can indicate the error by the residual of the spectral parameter $a(\kappa)$ at the computed eigenvalues. This error measure does not account for discretization errors. The main benefit is that it can even be calculated if the true eigenvalues are not known, which is in practice typically the case. Again, we calculate this error for the N largest eigenvalues and the spurious $\hat{N}-N$ smallest eigenvalues separately:

$$\underset{n\leq N}{\mathrm{RMS}}\{a(\hat{K}_n)\} := \sqrt{\frac{1}{N}\sum_{n=1}^{N}\big(a(\hat{K}_{n+\hat{N}-N})\big)^2}, \tag{3.35}$$

$$\underset{n>N}{\mathrm{RMS}}\{a(\hat{K}_n)\} := \sqrt{\frac{1}{\hat{N}-N}\sum_{n=1}^{\hat{N}-N}\big(a(\hat{K}_n)\big)^2}, \quad \text{if } \hat{N}>N. \tag{3.36}$$

Unfortunately, the Matslise benchmark algorithm (MS18) does not provide information on the residual.

3.5.4. The effect of the tolerance

Both the proposed algorithm and the benchmark algorithms contain a tolerance parameter Δ to select the desired accuracy. However, they respond very differently to the setting of Δ. This can be seen in Fig. 3.5. In that figure we compare for the potential $q_1(x)$ the proposed algorithm to the benchmark algorithms for different settings of Δ.

The top right panel of Fig. 3.5 shows the achieved error $\mathrm{RMS}_{n\leq N}\{\hat{K}_{n+\hat{N}-N}-K_n\}$ as a function of the tolerance Δ. The similar graphs for the example potentials $q_2(x)$ till $q_6(x)$ are shown in Fig. 3.6. We see that for high values of the tolerance Δ the bisection based benchmark algorithms (Algs. B2 and B4) achieve an error $\mathrm{RMS}_{n\leq N}\{\hat{K}_{n+\hat{N}-N}-K_n\}$ just below the tolerance Δ. When the tolerance is reduced, both of these algorithms hit an error floor when the error due to the reconstruction of the potential becomes the dominant source of error. These error floors depend on the potential as well as on the sampling interval and truncation. The fourth order error floor of benchmark Alg. B4 is in most cases lower than the second order error floor of benchmark Alg. B2.[16] The proposed algorithm, Alg. NR4, achieves the fourth order error floor for a much wider range of settings of the tolerance Δ. The reason is that if the NR descend is successful, it terminates at either a tight upper or lower bound. The opposite bound is enforced to ensure that the bound is tight enough,

[16] For example potential $q_6(x)$ the second and fourth order error floor are equal, because a rectangular potential is a truncated constant. For example potential $q_4(x)$ the error floor is equal for all algorithms. We suspect that the computation of the samples of $q_4(x)$ is the dominant source of error in that example, due to the close separation of the eigenvalues.

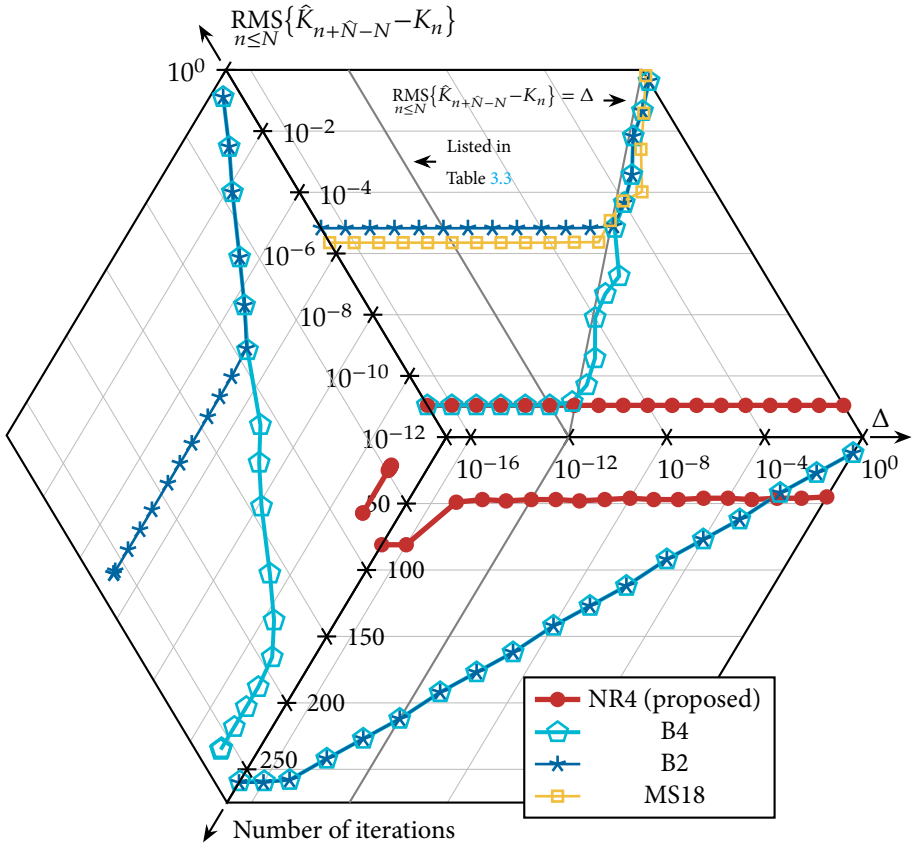


Figure 3.5: The plots show a comparison between the proposed algorithm and the three benchmark algorithms for the example potential $q_1(x)$. The top right panel shows the achieved accuracy as a function of the chosen tolerance Δ. The bottom right panel shows the number of iterations that is required, as a function of the chosen tolerance Δ. The left panel shows the resulting trade-off curve. For Alg. MS18 no information is available on the number of iterations.

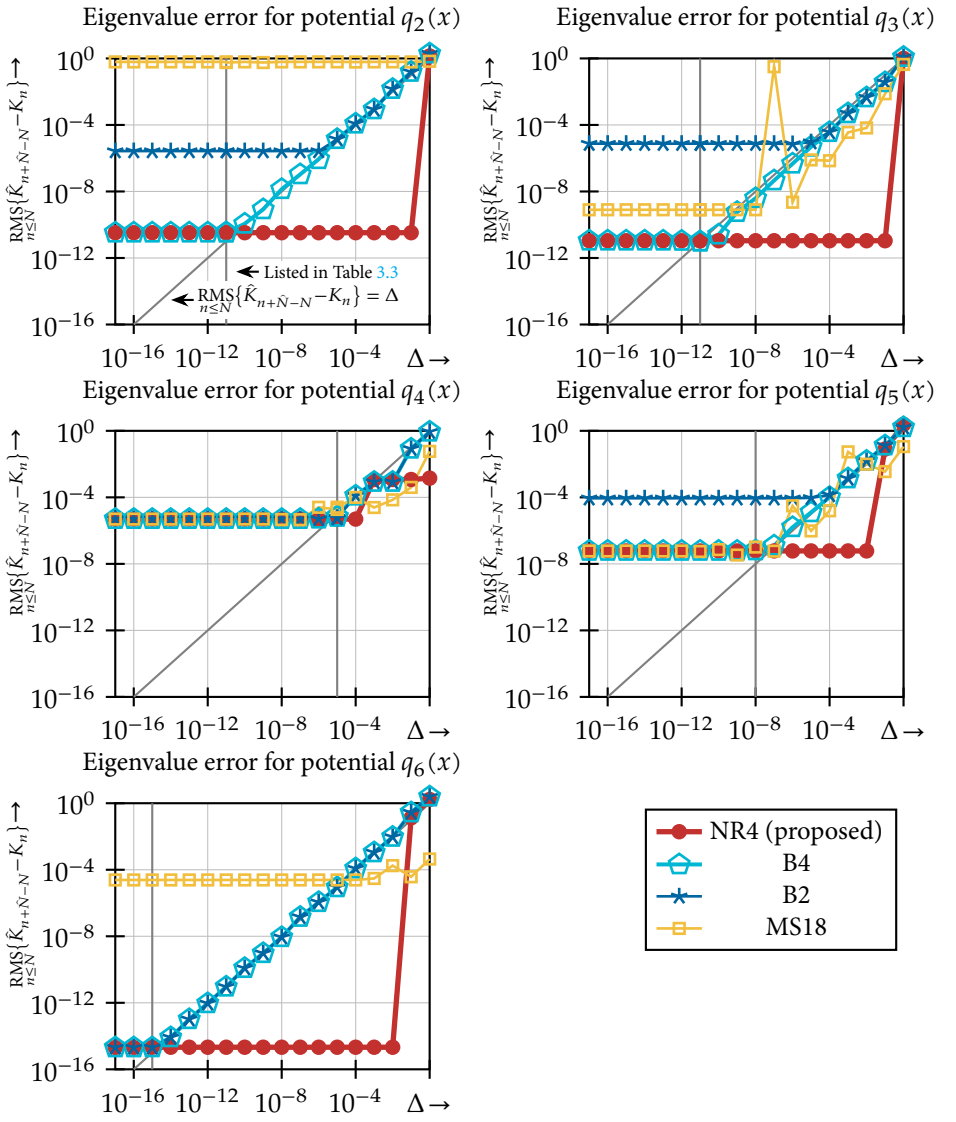


Figure 3.6: The plots show the achieved accuracy as a function of the chosen tolerance Δ for the example potentials $q_2(x)$ till $q_6(x)$ when using the proposed algorithm and the three benchmark algorithms respectively.

but in practice it is the termination point of the NR descend that determines the returned estimates $\hat{K}_n$. If the allowable tolerance is very high, the error of the proposed algorithm may be closer to the tolerance. In those cases one or more NR descends jumped out of the range of proper values κ, upon which the algorithm checks if the bracket is below the tolerance. (See block 9 in Fig. 3.3.) If a high tolerance is allowed, that may already be the case, even if the NR descend did not converge. On the other hand, if we select an extremely low tolerance, machine precision effects can show up. Among these examples this effect is only visible for the potential $q_1(x)$, in Fig. 3.5 for $\Delta = 1 \times 10^{-16}$ and $\Delta = 1 \times 10^{-17}$. In these cases the tolerance is so small that after the termination of the NR descend, the update to enforce an opposite bound (blocks 17 and 18 in Fig. 3.3) is too small to find an opposite bound immediately. It is thus advisable not to choose Δ lower than 1×10^{-15} in double precision computations.

The invariant behaviour of the proposed algorithm, Alg. NR4, with respect to a wide range of tolerances Δ is a practical advantage: We do not have to think much about an appropriate setting of Δ, since we obtain the maximum accuracy in the same number of iterations for any reasonable setting. For computations in double precision arithmetic, we would simply select a default value of $\Delta = 1 \times 10^{-15}$.

Algorithm MS18 does not make use of the samples that the other algorithms use. Instead it selects its own mesh based on the variation of the potential and the user input tolerance [85, §2.3]. Please recall that Matslise's tolerance is a tolerance on the squares of the eigenvalues, and therefore $\mathrm{RMS}_{n\leq N}\{\hat{K}_{n+\hat{N}-N} - K_n\} < \Delta$ does not have to hold. Nevertheless, we would expect that as the tolerance Δ is tightened, the result keeps getting more accurate (at the cost of computation time) until the truncation of the potential becomes the dominant source of error. Since the other algorithms suffer likewise from this truncation error, the error floor of the fourth order algorithms (Algs. B4 and NR4) should be achievable for the eighteenth order Matslise algorithm as well. Nevertheless, we see in Figs. 3.5 and 3.6 that the error floor of Alg. MS18 can be much higher than the fourth order error floor. Furthermore, we see for some of the examples that the error curve is not a monotonically non-decreasing function of the tolerance, but shows local maxima instead. We also noticed that these error curves can change significantly if the computation is repeated with a different version of Matlab, or if the potential is obtained from a slightly different number of samples before band limited interpolation. According to the current developers, Matslise 3.0/Pyslise uses a more conservative error estimate and a finer mesh than its predecessor [14]. Therefore we surmise that the next version, if it allows boundary conditions at infinity, will return stabler and more accurate results.

In the bottom right panel of Fig. 3.5, we see the effect of the tolerance on the number of iterations. For the bisection based algorithms, Algs. B2 and B4, the number of iterations increases proportional to $\log(1/\Delta)$ and is the same for both versions. The proposed algorithm, Alg. NR4, needs roughly the same number of iterations for every user-selected tolerance Δ.[17] Unfortunately we do not have access to the number of iterations (and computational cost per iteration) of Alg. MS18.

The left panel of Fig. 3.5 shows the trade-off curve between accuracy and number of iterations that results from the other two panels. We will discuss it in the next subsection.

[17] If the error is not at the fourth order error floor for high Δ, such as in Fig. 3.6, the algorithm terminates earlier. Hence in that case the number of iterations also drops.

3.5.5. Comparison of error and computational cost

The left panel of Fig. 3.5 shows the trade-off curve between accuracy and number of iterations that results from the other two panels. We see that the proposed algorithm, Alg. NR4, achieves the best achievable accuracy of the fourth order benchmark algorithm, Alg. B4, but at a much lower number of iterations. Even if we take into account that the iterations of the proposed algorithm are 70 % more expensive (see Section 3.3.3), the proposed algorithm remains computationally cheaper.

Hereafter, we want to compare the proposed algorithm to the benchmark algorithms in more detail, considering computational cost and several kinds of error. For a fair comparison it would be preferable to fix the error and compare the computational cost, or to fix the computational cost and compare the errors. That is unfortunately not possible. We can only fix the tolerance. We set it close to the optimal value for the benchmark algorithm B4. That is, close to the maximum value Δ for which the error $\mathrm{RMS}_n\{\hat{K}_{n+\hat{N}-N} - K_n\}$ is at the error floor. For example in Fig. 3.5 we see that this is $\Delta = 1\times10^{-12}$ for potential $q_1(x)$. The values for the other potentials can be read likewise from Fig. 3.6. A consequence of this choice is that the eigenvalue errors of Algs. NR4 and B4 will be both on the fourth order error floor, so approximately the same. We will furthermore have to keep in mind that Alg. B2 would be capable of achieving the same error as in this test at less computational cost, by selecting a higher tolerance. The same likely holds for Alg. MS18, but we cannot verify that since we have no information on its computational cost for these examples.

In Table 3.3 we show the results at this tolerance for the proposed algorithm, Alg. NR4, as well as for the three benchmark algorithms, Algs. B2, B4 and MS18. Besides the eigenvalue error that we have already seen in Figs. 3.5 and 3.6, it also shows the residual error, the errors which correspond to spurious eigenvalues where applicable and the iteration counts. Algorithm B2s is equal to Alg. B2, except for the root counter. It uses instead of (3.18) always the simple sign check, as in the upper case of (3.18). Algorithm B2s is thus as close as possible to the algorithm that was proposed in [102], except for the necessary adaptation to vanishing boundary conditions. We will discuss the results from this algorithm in Section 3.5.6.

Computational cost

Although we optimized the tolerance for the B4 algorithm, Table 3.3 shows that the proposed algorithm requires only about a fourth of the number of iterations of the bisection based algorithms for most example potentials. Even when we take into account that iterations of the proposed algorithm are approximately 70 % more expensive than those of B4 (see Section 3.3.3) and those of Alg. B2 are 50 % cheaper than those of B4, the proposed algorithm remains computationally cheapest. Indeed, Alg. B2 would have achieved approximately the same error at a lower number of computations if a higher tolerance were selected. However, the optimal choice for the tolerance is in practice unknown.

The proposed algorithm is typically computationally cheaper than bisection, but Table 3.3 shows one exception, namely potential $q_4(x)$. The eigenvalues of $q_4(x)$ are clustered together and while searching for the first eigenvalue, the bisection algorithms find a tight bracket for the whole cluster. Therefore the other eight eigenvalues can be localized in few extra iterations. Algorithm NR4 on the other hand converges first with a long series of NR iterations from below to the smallest eigenvalue. For the remainder of the cluster this only gives a reasonably tight lower bound. It finds the upper bound of the cluster with another

Alg.	$\hat{N}$	Iter.	$\underset{n\le N}{\mathrm{RMS}}\{\hat{K}_{n+\hat{N}-N}-K_n\}$	$\underset{n\le N}{\mathrm{RMS}}\{a(\hat{K}_{n+\hat{N}-N})\}$	$\underset{n>N}{\mathrm{RMS}}\{\hat{K}_n\}$	$\underset{n>N}{\mathrm{RMS}}\{a(\hat{K}_n)\}$
$q(x)=q_1(x)$	$(N=5)$,	$\Delta=1\times10^{-12}$				
NR4	N	47	1.09×10^{-11}	2.47×10^{-16}		
B4	N	212	1.09×10^{-11}	3.89×10^{-14}		
B2	N	212	6.70×10^{-6}	4.43×10^{-14}		
MS18	N	$\star$	2.26×10^{-6}	$\star$		
$q(x)=q_2(x)$	$(N=25)$,	$\Delta=1\times10^{-11}$				
NR4	N	202	3.32×10^{-11}	2.54×10^{-19}		
B4	N	881	3.38×10^{-11}	1.33×10^{-12}		
B2	N	881	2.68×10^{-6}	1.05×10^{-12}		
MS18	$N+3$	$\star$	6.19×10^{-1}	$\star$	3.68×10^{-1}	$\star$
$q(x)=q_3(x)$	$(N=6)$,	$\Delta=1\times10^{-11}$				
NR4	N	52	1.09×10^{-11}	9.89×10^{-21}		
B4	N	219	9.69×10^{-12}	2.26×10^{-15}		
B2	$N+1$	256	7.47×10^{-6}	3.62×10^{-16}	3.05×10^{-5}	5.50×10^{-8}
MS18	N	$\star$	7.93×10^{-10}	$\star$		
$q(x)=q_4(x)$	$(N=9)$,	$\Delta=1\times10^{-5}$				
NR4	$N+1$	188	4.93×10^{-6}	1.06×10^{-46}	6.45×10^{-5}	3.82×10^{-2}
B4	$N+1$	77	6.90×10^{-6}	1.05×10^{-36}	6.47×10^{-5}	4.21×10^{-2}
B2	$N+1$	77	5.77×10^{-6}	1.99×10^{-36}	1.03×10^{-4}	6.63×10^{-3}
MS18	N	$\star$	2.57×10^{-5}	$\star$		
$q(x)=q_5(x)$	$(N=30)$,	$\Delta=1\times10^{-8}$				
NR4	N	230	6.01×10^{-8}	1.49×10^{-36}		
B4	N	707	6.42×10^{-8}	1.30×10^{-28}		
B2	$N+1$	733	8.79×10^{-5}	6.14×10^{-29}	5.35×10^{-4}	6.85×10^{-6}
MS18	N	$\star$	1.02×10^{-7}	$\star$		
$q(x)=q_6(x)$	$(N=21)$,	$\Delta=1\times10^{-15}$				
NR4	N	258	2.13×10^{-15}	1.70×10^{-23}		
B4	N	1002	2.07×10^{-15}	1.73×10^{-18}		
B2	N	1002	2.07×10^{-15}	2.17×10^{-18}		
B2s	$N-20$	52				
MS18	N	$\star$	2.44×10^{-5}	$\star$		

Table 3.3: Results of finding the eigenvalues of potentials $q_1(x)$, ..., $q_6(x)$ (see Fig. 3.4) with four versions of an automatic eigenvalue finder: Algs. NR4 (proposed), B2, B4 and MS18. Additionally, the results of a fifth algorithm, Alg. B2s, are shown for the potential $q_6(x)$. Algorithm B2s uses a simple sign check root counter instead of (3.18) and is otherwise equal to Alg. B2. Table cells of which the value cannot be computed (because the computed number of eigenvalues $\hat{N}$ leads to an invalid expression) are left blank. The symbol $\star$ means that the data are missing that are needed to compute the value in that cell.

long series of NR iterations that converges to the highest eigenvalue. Both of these series appear to be atypical cases in which the NR algorithm converges rather slowly.

Spurious eigenvalues

When an algorithm returns more eigenvalues than there should be, we refer to the surplus of eigenvalues as *spurious eigenvalues*. More precisely, we consider the $\hat{N} - N$ smallest eigenvalues, where $\hat{N}$ is the numerically calculated number of eigenvalues, as spurious. Those are the eigenvalues for which the highest numbers of zero-crossings are detected.

Regarding potential $q_4(x)$ we see in Table 3.3 that all algorithms except for Alg. MS18 return one spurious eigenvalue. Algorithm B2 furthermore returns one spurious eigenvalue for the potentials $q_3(x)$ and $q_5(x)$. In all cases this eigenvalue is close to zero: The eigenvalue is typically one order of magnitude larger than the *error* in the true eigenvalues. This is typical for reflectionless potentials. The reason is that the scattering parameter $a(\kappa)$ of the reconstructed potential indeed has an extra root compared to the true potential. However, this is not problematic in practice, since a near-zero eigenvalue represents in the KdV-NFT spectrum a component of near-zero 'mass', 'momentum' and 'energy'. More precisely, these are proportional to respectively the first, third and fifth power of the eigenvalue [47, §3.1], [164, §3]. Therefore, if we would compute the inverse KdV-NFT of a spectrum, the presence of a very small spurious eigenvalue will have little influence on the resulting potential.

For potential $q_2(x)$ we see in Table 3.3 that Alg. MS18 returns three spurious eigenvalues, while the other algorithms find the correct number of eigenvalues. The spurious and non-spurious eigenvalue errors of Alg. MS18 are also very high in this case. This is because Alg. MS18 finds the true eigenvalues to reasonable precision, plus some spurious eigenvalues *between* the true eigenvalues. The index of some of the smallest true eigenvalues is shifted by these spurious eigenvalues. This would be expected behaviour for a root finder that does not use SL oscillation theory. However, this theory makes it possible to index every single eigenvalue without knowing the others, by counting the number of zero-crossings of its eigenfunction [85]. Therefore, this behaviour is remarkable for a root finder that is based on SL theory.

Residual errors

We have no information on the residual error of Alg. MS18, so we can compare the proposed algorithm, Alg. NR4, only to the bisection based benchmark Algorithms B2 and B4. When we look at the residual errors of the non-spurious eigenvalues ($\underset{n\leq N}{\mathrm{RMS}}\{a(\hat{K}_{n+\hat{N}-N})\}$), we see that the proposed algorithm, Alg. NR4, achieves in all cases a significantly lower error than the bisection Algorithms B2 and B4. Nevertheless, the eigenvalue error $\underset{n\leq N}{\mathrm{RMS}}\{\hat{K}_{n+\hat{N}-N}-K_n\}$ of the proposed algorithm is similar to the eigenvalue error of Alg. B4. This shows that the proposed algorithm gets significantly closer to the eigenvalues of the *reconstructed* potential (see (3.31)), but not closer to the eigenvalues of the *true* potential. Hence, the error of the proposed search algorithm is negligible compared to the discretization error itself.

3.5.6. Numerical robustness

The potential $q_6(x)$ is included to demonstrate two aspects of the numerical robustness of the computation of the accounting function that was presented in Section 3.3.2.

Firstly, the potential $q_6(x)$ demonstrates that the simple sign check (the case $(\varepsilon\gamma)^2 < 9$ in (3.18) alone, cf. Osborne [102]), is not sufficient. We included in this example Alg. B2s, which is equal to Alg. B2, except that Alg. B2s uses the simple sign check as a root counter, instead of (3.18) as a whole. Since there are only two samples plus the higher tail in this case, Alg. B2s can count maximally three sign changes in total, whereas the correct accounting function should range to $N = 21$. As we see in Table 3.3, it detects only one zero crossing at $\kappa = 0$, so that only $\hat{N} = 1$ eigenvalue is returned. (That one numerically computed eigenvalue differs 2.66×10^{-16} from the highest eigenvalue in this spectrum.) It needs no explanation that the inverse KdV-NFT will be significantly different from the original potential if 20 of the 21 eigenvalues are missing from its discrete spectrum. Indeed, the potential $q_6(x)$ is deliberately constructed as an extreme case, but still it illustrates why the case $(\varepsilon\gamma)^2 \geq 9$ in (3.18) is needed.

Secondly, the potential $q_6(x)$ demonstrates the robustness of counting zero-crossings of the trajectory $\phi(x)$ near a boundary $x = x_d^{\mathrm{u}}$. Because of the even symmetry of $q_6(x)$, 10 out of its 21 eigenfunctions are odd symmetric and thus have a zero-crossing at $x = 0$. For Algs. NR4, B2 and B4 that is at the boundary between two piecewise constant sections of the reconstructed potential. (Since Matslise chooses its own grid, we cannot effectively present Alg. MS18 with this challenge.) With the low number of samples we put this zero-crossing also numerically as close as possible to $x = 0$ and maximise the risk of counting this zero-crossing as two or not at all. We see from the results in Table 3.3 that Algs. NR4, B2 and B4 each find all eigenvalues, so we conclude that the computation indeed appears robust against this source of error.

The results of $q_6(x)$ furthermore show that the eigenvalue error does not improve when using a fourth order algorithm (Algs. NR4 and B4) rather than a second order algorithm (Alg. B2). This was to be expected, since in this particular case both approximations result in the exact true potential.

3.6. Conclusion

In this chapter we presented a numerical algorithm for the computation of eigenvalues of the KdV-NFT spectrum from sampled data with vanishing boundaries. The proposed algorithm uses a shooting approach with an NR-based root-finder. Nevertheless, because we make use of SL oscillation theory, we can guarantee that our algorithm finds all the eigenvalues. This theory is in the literature usually combined with a bisection-based root-finder, because a bracketing root-finder is the natural choice for the type of information that SL oscillation theory provides. However, NR typically converges faster. Therefore we designed an algorithm that combines SL oscillation theory with an NR-based root-finder. We have demonstrated that our algorithm indeed typically saves computation time compared to a bisection based root-finder. Furthermore, we have shown that for sampled data with vanishing boundaries, our algorithm is more robust to numerical rounding errors than other shooting algorithms that are currently available. Therefore, our algorithm is more reliable.

4

Accurate computation of the norming constants of the discrete KdV spectrum

Several non-linear fluid mechanical processes, such as wave propagation in shallow water, are known to generate solitons: localized waves of translation. Solitons are often hidden in a wave packet at the beginning and only reveal themselves in the far field. With a special signal processing technique known as the Non-linear Fourier Transform (NFT), solitons can be detected and characterized before they emerge. In this chapter we present a new algorithm aimed at computing the phase shift of solitons in processes governed by the Korteweg–de Vries equation (KdV) equation. In numerical examples, the new algorithm is found to perform reliably even in cases where existing algorithms break down.

4.1. Introduction

The Korteweg–de Vries equation (KdV) is a non-linear partial differential equation that describes for example the propagation of long internal waves [33] as well as long unidirectional surface waves in shallow[1] water [76]. A dimensional form of the KdV is

$$\eta_{\tilde{t}} + c\eta_{\tilde{x}} + \alpha\eta\eta_{\tilde{x}} + \beta\eta_{\tilde{x}\tilde{x}\tilde{x}} = 0, \tag{4.1}$$

where $\eta = \eta(\tilde{x}, \tilde{t})$ [m] is the (free) surface elevation as a function of the time $\tilde{t}$ [s] and the position in $\tilde{x}$ [m] in lab coordinates and where subscripts denote partial derivatives. In the

Parts of this chapter have been published as P. J. Prins and S. Wahls. "Soliton Phase Shift Calculation for the Korteweg–De Vries Equation". In: *IEEE Access* 7 (1 Sept. 12, 2019), pp. 122914–122930. ISSN: 2169-3536. DOI: 10.1109/access.2019.2932256, under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/). Minimal changes in text and layout have been made.

[1] Shallow: The depth is less than 0.22 times the wavelength [25].

case of free surface waves, the coefficients of (4.1) are

$$c = \sqrt{gh}, \qquad \alpha = \frac{3c}{2h}, \qquad \beta = \frac{ch^2}{6},$$

where g [m/s^2] is the gravitational acceleration, h [m] the still water depth, and c [m/s] the phase celerity of the wave [25, 101]. By the changes of variables [61]

$$t = \frac{c}{6h}\tilde{t}, \qquad x = \frac{\tilde{x} - c\tilde{t}}{h}, \qquad q(x,t) = \frac{3}{2h}\,\eta(\tilde{x}(x,t), \tilde{t}(t)),$$

we obtain a standard form of the KdV, to wit

$$q_t + 6qq_x + q_{xxx} = 0. \tag{4.2}$$

All variables in this standard form are unit-less and real. The position x in (4.2) is expressed with respect to a frame that moves with the phase celerity c, which cancels one term compared to (4.1).

A peculiarity of the KdV is that it evolves any wave packet into a parade of $N \geq 0$ solitons. (If there are no solitons, $N = 0$, the wave is fully dispersive and will vanish over time [5, §1.7.c].) Solitons are waves of translation with the shape of a squared hyperbolic secant,

$$\operatorname{sech}^2(\theta) = \left(\tfrac{1}{2}\,\mathrm{e}^{-\theta} + \tfrac{1}{2}\,\mathrm{e}^{\theta}\right)^{-2}. \tag{4.3}$$

After a sufficiently long time t,

$$q(x,t) \approx \sum_{n=1}^{N} \mathring{q}_n \operatorname{sech}^2(k_n x - \omega_n t - \varphi_n), \tag{4.4}$$

where $\mathring{q}_n = 2k_n^2$ and $\omega_n = 4k_n^3$ [125, Eq. 17], [126, p. 83, Eq. 3.3], [7, Eq. 2.20a]. The amplitude, wavelength and celerity of a soliton are thus coupled. We call the regime where (4.4) holds the far field. Otherwise we are speaking of the near field. The near field can in general not be described as a linear superposition of wave components, due to the non-linear interaction between them.

Nevertheless, from near field data we can obtain the free parameters that describe the far field – the generalized wave numbers k_n and the phase shifts φ_n – long before individual solitons start to separate from a wave packet. We do that with the so-called scattering transform. It is a generalization of the common Fourier transform and therefore nowadays often called Non-linear Fourier Transform (NFT) in the literature [24]. The generalized wave numbers and phase shifts of the solitons are represented in the NFT spectrum by the so-called eigenvalues ζ_{0n} and norming constants $b(\zeta_{0n})$, as will be explained later, and the NFT enables us to calculate them from the normalised free surface at any fixed time. For brevity and to conform to the parlance of the NFT, we will hereafter call the normalized free surface at a fixed time, $q(x, t_0)$, the potential. It has been shown that the NFT can extract features of water waves that remain hidden with linear methods of signal analysis. We remark that most of the recent research assumes periodic waves [23–25, 106, 107], whereas in this chapter we consider wave packets [61, 62, 101, 110, 114].

The main motivation of this chapter is that whereas the numerical computation of the soliton amplitude, wavelength and celerity from near field data is a solved problem, the numerical computation of the phase shifts φ_n that appear in (4.4) is surprisingly hard. The difficulty lies in the numerical computation of the aforementioned norming constants $b(\zeta_{0n})$ [55]. This is the problem we deal with in this chapter. There are few publications that deal with the numerical aspects of the computation of the norming constants for the KdV-NFT. The paper [114] reports numerical results, but does not disclose how the phase shifts were calculated numerically. Neither does its companion paper [101]. In [142, §8.2] an algorithm to calculate norming constants is proposed. We will discuss it briefly at the end of Section 4.3.3, after the required terminology has been established. The authors apply their algorithm to calculate the norming constants of the potential $q(x,0) = 2.4\,\mathrm{sech}^2(x)$ and the result is used to demonstrate the convergence of an inverse NFT algorithm that they propose. As far as the calculation of the norming constants is concerned, this potential is a relatively easy case, because it has a single crest, at $x = 0$, and contains only two solitons (cf. Example 3 in Section 4.4.4). It is not clear at the moment how well this method performs on difficult examples. In [55] one of the possible ways to compute the norming constants is analysed and it is concluded that it is necessary to use a high-precision arithmetic, except for the norming constant of the soliton with the lowest amplitude. Recently two closely related algorithms have been proposed independently to address the numerical computation of the norming constants for the NFT that solves the Non-linear Schrödinger Equation (NSE) [12, 63]. In this chapter we derive a new algorithm to compute the norming constants of the KdV-NFT that builds on the ideas of [12, 63]. Our key contribution is that we utilize not one but two different estimators of the norming constant, which greatly improves the reliability of the algorithm.

The chapter is organized as follows. In Section 4.2 we recapitulate the relevant parts of the NFT for the KdV. In Section 4.3 we present and motivate our new algorithm to calculate the norming constants; we validate it with numerical examples in Section 4.4 and show that it is significantly more reliable than a reapplication of the ideas of [12, 63] for the KdV. This chapter ends with a conclusion in Section 4.5. We have furthermore included several appendices with supporting information.

4.2. Preliminaries

4.2.1. Notation

In this chapter we typeset variables in italic and constants upright. The constants include j as the imaginary unit, e as Euler's number and π as the ratio of a circle's circumference to its diameter. For vectors we use lower case bold ($\boldsymbol{v}$); for matrices upper case bold ($\boldsymbol{A}$) and $\boldsymbol{I}$ denotes the 2×2 identity matrix. Scalars are typeset in lower or upper case regular (q, X_0), where for elements of a matrix two subscripts denote the row and column in that order (A_{ij}). For operators we use a sans-serif upright font (V). A hat on top of a variable ($\hat{q}$) means an estimation or approximation of that variable. The set $\mathbb{R}$ is the set of real numbers; $\mathbb{I}$ is the set of imaginary numbers; $\mathbb{C}$ is the set of complex numbers; other sets are denoted with a capital in calligraphic font ($\mathcal{X}$). The symbol := means that the left hand side is defined by the right hand side; the symbol $\propto$ means 'is proportional to'. We use $\mathcal{O}$ as the 'big-O' Landau order symbol. We reserve log for the base 10 logarithm and use ln for the natural logarithm,

with base e. The function exp denotes the exponential with base e, so $\exp(\theta) := e^{\theta}$.

4.2.2. Non-Linear Fourier Transform for wave packets

In this subsection, we survey the mathematical background of the NFT for wave packets. By wave packets we mean a localized real potential $q(x, t_0)$ with so-called vanishing boundary conditions. Formally,[2]

$$\begin{cases} \lim_{|x|\to\infty} q(x, t_0) = 0; \\ \int_{-\infty}^{\infty} (1 + |x|)\ |q(x, t_0)|\ \mathrm{d}x < \infty, \end{cases} \tag{4.5}$$

for some fixed time t_0. In this chapter we deal with the case that the function $q(x, t)$ furthermore satisfies the KdV (4.2). As a magic step (refer to [4, 81] for the explanation) we use $q(x, t)$ as a time-varying potential in the Schrödinger eigenvalue problem:

$$\left(\frac{\partial^2}{\partial x^2} + q(x, t)\right) f(x, \zeta, t) = (\mathrm{j}\zeta)^2 \cdot f(x, \zeta, t). \tag{4.6}$$

Then (4.6) has two important types of solutions. Firstly, for all $\zeta \in \mathbb{R}\backslash\{0\}$ there exist solutions for which the eigenfunctions f are power signals,[3] and they comprise the so-called continuous spectrum. This part of the NFT spectrum decays over time; the corresponding wave components dwindle as a dispersive wave train [7]. The continuous spectrum is thus of interest for the near-field of a wave, but not in the scope of this chapter. Secondly, there is a finite number $N \geq 0$ of distinct solutions for which the eigenfunctions f are energy signals.[4] These solutions comprise the so-called discrete spectrum. The values of ζ that are part of the discrete spectrum are called eigenvalues and we will denote them as ζ_0, or when we refer to a specific eigenvalue as ζ_{01}, ζ_{02}, etcetera. It can be shown that all eigenvalues of (4.6) are non-zero imaginary numbers in the upper half plane [4, p. 251], [114, Eq. 3.21], [127, §3]. When we write hereafter ζ, we mean a non-zero number that is either real or imaginary unless explicitly indicated otherwise.

It is the discrete spectrum that has our focus in this chapter, for the reason that every eigenvalue corresponds to one separated soliton in the far field. The discrete spectrum can be subdivided into two parts. The first part consists of the eigenvalues, which can be shown to remain constant while the wave evolves [53]. That is, from the near field up to and including the far field. Specifically, in (4.4), $\mathrm{j}k_n = \zeta_{0n}$, so an eigenvalue tells us the amplitude, wavelength and celerity of the corresponding soliton in the far field and as the solitons evolve further into the far field we can observe that those indeed remain constant. However, we can calculate the eigenvalues with the NFT at any fixed $t = t_0$.

The second part of the discrete spectrum contains information about the eigenfunctions. This part of the spectrum does evolve over time – in an easy to compute way – and contains

[2] If the KdV is normalized otherwise, such that $q(x, t_0) \to \breve{h}$ as $|x| \to \infty$ for some finite constant $\breve{h}$, then (4.2) is equivalent to $q'_t + 6q'q'_{x'} + q'_{x'x'x'} = 0$, where $x' := x - 6\breve{h}t$ and $q' := q - \breve{h} \to 0$ as $|x'| \to \infty$. If the transformed potential q' satisfies (4.5), we can proceed with this scaled KdV.

[3] $f(x)$ is a power signal $\Leftrightarrow \lim_{X\to\infty} \frac{1}{2X} \int_{-X}^{X} |f(x)|^2\ \mathrm{d}x < \infty$.

[4] $f(x)$ is an energy signal $\Leftrightarrow \int_{-\infty}^{\infty} |f(x)|^2\ \mathrm{d}x < \infty$.

the additional information we need, to calculate the phase shifts φ_n in (4.4). By the virtue of (4.5), as $|x| \to \infty$ the Schrödinger eigenvalue problem (4.6) reduces to

$$\frac{\partial^2}{\partial x^2} f(x,\zeta,t) = (\mathrm{j}\zeta)^2 \cdot f(x,\zeta,t), \tag{4.7}$$

and hence every eigenfunction of (4.6) reduces in this limit to a linear combination of the functions $\exp(\pm \mathrm{j}\zeta x)$ [79, §2.8]. In particular, we can look for one set of eigenfunctions that satisfies

$$\begin{cases} \phi(x,\zeta,t) \to \exp(-\mathrm{j}\zeta x), \\ \bar{\phi}(x,\zeta,t) \to \exp(\mathrm{j}\zeta x) \end{cases} \quad \text{as } x \to -\infty \tag{4.8}$$

and another set of eigenfunctions that satisfies

$$\begin{cases} \bar{\psi}(x,\zeta,t) \to \exp(-\mathrm{j}\zeta x), \\ \psi(x,\zeta,t) \to \exp(\mathrm{j}\zeta x) \end{cases} \quad \text{as } x \to \infty. \tag{4.9}$$

These sets of eigenfunctions are known as the Jost solutions and it should be noted that they are also uniquely defined for all other x by (4.6). Each of the two sets of Jost solutions forms a linearly independent basis for the eigenfunctions of (4.6) and they are related as

$$\begin{bmatrix} \phi(x,\zeta,t) & \bar{\phi}(x,\zeta,t) \end{bmatrix} = \begin{bmatrix} \bar{\psi}(x,\zeta,t) & \psi(x,\zeta,t) \end{bmatrix} \boldsymbol{S}(\zeta,t), \tag{4.10}$$

where

$$\boldsymbol{S}(\zeta,t) := \begin{bmatrix} a(\zeta) & \bar{b}(\zeta,t) \\ b(\zeta,t) & \bar{a}(\zeta) \end{bmatrix} \in \mathbb{C}^{2\times 2}, \tag{4.11}$$

where the scattering parameters $b(\zeta,t) \equiv \bar{b}(-\zeta,t)$ and $a(\zeta) \equiv \bar{a}(-\zeta)$ are implicitly defined by (4.10); their explicit definitions can be found in Appendix 4.A. The parameter $a(\zeta)$ depends neither on x nor on t; the parameter $b(\zeta,t)$ does not depend on x, but evolves over time as [5, Eq. (1.4.10)]

$$b(\zeta,t) = b(\zeta,0)\exp(8\mathrm{j}\zeta^3 t). \tag{4.12}$$

Although the scattering matrix $\boldsymbol{S}(\zeta,t)$ is in general complex valued, it is real valued when ζ is an imaginary number, so for the discrete spectrum in particular $\boldsymbol{S}(\zeta_0,t) \in \mathbb{R}^{2\times 2}$. Another important property of the scattering matrix is [4, Eq. A3.4][79, Eqs. 2.8.10–2.8.11]

$$\det(\boldsymbol{S}(\zeta,t)) = a(\zeta)\,\bar{a}(\zeta) - b(\zeta,t)\,\bar{b}(\zeta,t) \equiv 1. \tag{4.13}$$

It can be shown that the set of eigenvalues of (4.6) can be determined from the scattering matrix as [4, Eq. 1.7b],[79, §2.8]

$$\{\zeta_0 \mid \mathrm{Im}(\zeta_0) > 0 \ \wedge\ a(\zeta_0) = 0\} \subset \mathbb{I}. \tag{4.14}$$

Associated to each eigenvalue ζ_{0n} is a norming constant that is given by $b(\zeta_{0n},t)$, which is in fact not a constant, but a quantity that evolves over time according to (4.12). With this

norming constant we can finally calculate the phase shifts φ_n in (4.4) as [125, Eq. 25a],[126, p. 84, Eq. (4.2a)],[7, Eq. 2.20b]

$$\varphi_n = \frac{1}{2}\ln\left(\frac{b(\zeta_{0n},t)}{\mathrm{j}\,a'(\zeta_{0n})}\right) - 4\mathrm{j}\zeta_{0n}^3 t - \frac{1}{2}\ln(-2\mathrm{j}\zeta_{0n}) + \sum_{p=n+1}^{N}\ln\left(\frac{\zeta_{0p}-\zeta_{0n}}{\zeta_{0p}+\zeta_{0n}}\right), \tag{4.15}$$

where $|\zeta_{01}| < |\zeta_{02}| < \ldots < |\zeta_{0N}|$ is required as the order of the eigenvalues and where

$$a'(\zeta_{0n}) := \left.\frac{\mathrm{d}\,a(\zeta)}{\mathrm{d}\zeta}\right|_{\zeta=\zeta_{0n}} \in \mathbb{I}. \tag{4.16}$$

Among the quantities required in (4.15), the norming constants are especially hard to calculate numerically.[5] We will substantiate this claim in Section 4.3. The main goal of this chapter is to still calculate these norming constants accurately, to be able to calculate the phase shifts.

Not only do the norming constants contain information about the position of the solitons in the far field, but also of the evolution of the wave in the near field. The calculation of the latter is however more involved due to the non-linear interaction of the wave components. In the general case one needs to compute the inverse NFT, e.g. [127]; for a pure soliton potential (i.e. if $b(\zeta,t_0) = 0\ \forall\zeta \in \mathbb{R}\backslash\{0\}$) that may contain significant non-linear interactions between the solitons, simpler methods exist, such as the one described in Appendix 4.F and the reference therein.

In the remainder of this chapter we drop the dependence on $t = t_0$ from our notation.

Remark 5 (Sign inconsistency in [4])
The definitions used in this chapter for both $\bar{\phi}(x,\zeta)$ and $\bar{b}(\zeta)$ differ from [4, §III] by a minus sign, in order to be consistent with [4, Appendix 3]. Appendix 4.E, where we talk about the Non-linear Schrödinger Equation (NSE), is an exception. □

4.2.3. Calculation of the scattering matrix

The mathematical definition of the NFT is not convenient for numerical calculations. Therefore the Schrödinger eigenvalue problem, (4.6), is typically rewritten as a system of first order differential equations,

$$\frac{\partial}{\partial x}\boldsymbol{v}(x,\zeta) = \underbrace{\boldsymbol{A}(x,\zeta)}_{\in\mathbb{C}^{2\times 2}}\boldsymbol{v}(x,\zeta), \text{ where } \boldsymbol{v}(x,\zeta) := \mathbf{V}(\zeta)f(x,\zeta) \in \mathbb{C}^{2\times 1}, \tag{4.17}$$

and $\mathbf{V}(\zeta)$ is a column vector of length two of operators. In literature different choices are made for $\mathbf{V}(\zeta)$. We say that each suitable choice results in an analytically equivalent calculation, expressed in a different basis for the state vector $\boldsymbol{v}(x,\zeta)$. In this chapter we make the special choice

$$\mathbf{V}(\zeta) = \mathbf{V}_{\mathrm{S}}(\zeta) := \frac{1}{2\mathrm{j}\zeta}\begin{bmatrix}\mathrm{j}\zeta - \frac{\partial}{\partial x}\\ \mathrm{j}\zeta + \frac{\partial}{\partial x}\end{bmatrix}, \tag{4.18}$$

[5] One can for example calculate ζ_{0n} and $a'(\zeta_{0n})$ as in [17].

for which

$$\boldsymbol{A}(x,\zeta) = \boldsymbol{A}_{\mathrm{S}}(x,\zeta) = \begin{bmatrix} -\mathrm{j}\zeta + \frac{q(x)}{2\mathrm{j}\zeta} & \frac{q(x)}{2\mathrm{j}\zeta} \\ -\frac{q(x)}{2\mathrm{j}\zeta} & \mathrm{j}\zeta - \frac{q(x)}{2\mathrm{j}\zeta} \end{bmatrix}, \tag{4.19}$$

that simplifies the exposition. We will refer to this choice as the S basis. In Appendix 4.C we show how our results extend to alternative choices found in literature. For the Jost solutions with respect to the S basis we let

$$\boldsymbol{\Phi}_{\mathrm{S}}(x,\zeta) := \mathbf{V}_{\mathrm{S}}(\zeta) \begin{bmatrix} \phi(x,\zeta) & \bar{\phi}(x,\zeta) \end{bmatrix} = \mathbf{V}_{\mathrm{S}}(\zeta) \begin{bmatrix} \bar{\psi}(x,\zeta) & \psi(x,\zeta) \end{bmatrix} \boldsymbol{S}(\zeta) \tag{4.20}$$

by (4.10). Then we left-multiply (4.20) by

$$\boldsymbol{E}(\zeta x) := \begin{bmatrix} \exp(\mathrm{j}\zeta x) & 0 \\ 0 & \exp(-\mathrm{j}\zeta x) \end{bmatrix} \tag{4.21}$$

and take the limits for $x \to \mp\infty$ respectively to obtain

$$\lim_{x\to-\infty} \boldsymbol{E}(\zeta x)\, \boldsymbol{\Phi}_{\mathrm{S}}(x,\zeta) = \boldsymbol{I}, \tag{4.22}$$

$$\lim_{x\to\infty} \boldsymbol{E}(\zeta x)\, \boldsymbol{\Phi}_{\mathrm{S}}(x,\zeta) = \boldsymbol{S}(\zeta), \tag{4.23}$$

where we used (4.8) and (4.9). It is useful to note that $\boldsymbol{\Phi}_{\mathrm{S}}(x,\zeta)$ is invertible for all x, because $\det(\boldsymbol{\Phi}_{\mathrm{S}}(x,\zeta)) = \mathrm{W}[\phi;\bar{\phi}] = 2\mathrm{j}\zeta \neq 0$ by (4.51) in Appendix 4.A.

If we are dealing with a potential that satisfies (4.5) and is furthermore zero outside some window (X_-, X_+), the boundary conditions defining the Jost solutions [(4.8) and (4.9)] hold for all $x \geq X_+$ or all $x \leq X_-$ respectively. This allows us to replace $x \to -\infty$ in all the previous equations by $x = X_-$ and $x \to \infty$ by $x = X_+$. If we then multiply (4.23) from the left by $\boldsymbol{E}(-\zeta X_-)$ and from the right by (4.22) we find

$$\boldsymbol{\Phi}_{\mathrm{S}}(X_+,\zeta) = \boldsymbol{H}_{\mathrm{S}}(X_-,X_+,\zeta)\, \boldsymbol{\Phi}_{\mathrm{S}}(X_-,\zeta), \tag{4.24}$$

where

$$\boldsymbol{H}_{\mathrm{S}}(X_-,X_+,\zeta) := \boldsymbol{E}(-\zeta X_+)\, \boldsymbol{S}(\zeta)\, \boldsymbol{E}(\zeta X_-). \tag{4.25}$$

That is, $\boldsymbol{H}_{\mathrm{S}}(X_-,X_+,\zeta)$ is a state transition matrix of (4.17), a matrix that defines a bijective linear mapping from every initial state vector $\boldsymbol{v}_{\mathrm{S}}(X_-,\zeta)$ to its corresponding final state vector $\boldsymbol{v}_{\mathrm{S}}(X_+,\zeta)$. The calculation of the scattering matrix $\boldsymbol{S}(\zeta)$ can then be done by numerically evaluating (4.17) to find $\boldsymbol{H}_{\mathrm{S}}(X_-,X_+,\zeta)$, after which the scattering matrix is found according to (4.25).

Remark 6 (Why not the ubiquitous AKNS basis?)
An alternative choice of basis results in the so-called Ablowitz–Kaup–Newell–Segur (AKNS) system [4]. The AKNS system is popular for two reasons. Firstly it is a framework that covers multiple evolution equations, among which the KdV and the NSE. Secondly, for most of these evolution equations, the AKNS system is the special choice of basis that simplifies the exposition, because (4.22) and (4.23) hold for these evolution equations in that basis,

cf. [4, Eqs. 3.1 and 3.3] respectively. However, for the KdV the special choice appears to be the S basis instead. Indeed, [4, Eqs. A3.2 and A3.3] for the KdV in the AKNS basis differ from (4.22) and (4.23) respectively, in the sense that they implicitly use the change of basis matrix from the S basis to the AKNS basis, see (4.75) in Appendix 4.C. It is hence by the virtue of the S basis that the exposition in this chapter parallels the one for e.g. the NSE in the AKNS basis. □

4.3. Bidirectional algorithm

Recall from the introduction that the eigenvalues ζ_0 of a potential – a normalized free surface at any fixed time – signify the amplitude, wavelength and celerity of solitons, but contain no information about their phase shifts in the far field or their location in general. Therefore we need for each eigenvalue a second parameter that encodes this information: the norming constant $b(\zeta_0)$.

It is notoriously hard to calculate norming constants numerically. Let us shortly outline the reason for this issue. The eigenfunction $f(x,\zeta_0)$ should by definition be an energy signal. Therefore $f(x,\zeta_0)$ must be bounded[6] as $x \to \pm\infty$. For the KdV $\mathrm{j}\zeta_0 < 0$, so the Jost solutions $\bar{\phi}(x,\zeta_0)$ and $\bar{\psi}(x,\zeta_0)$ are by definition unbounded, see (4.8) and (4.9). Hence we can express the eigenfunction as a scalar multiple of the remaining, bounded Jost solutions:

$$f(x,\zeta_0) \propto \phi(x,\zeta_0) = b(\zeta_0)\,\psi(x,\zeta_0) \tag{4.26}$$

by (4.10) and (4.14). However, (4.26) will in general not hold exactly in a numerical calculation. Instead, an eigenfunction is typically represented like

$$\hat{\phi}(x,\zeta_0) = \underbrace{\hat{a}(\zeta_0)}_{\approx 0}\,\underbrace{\bar{\psi}(x,\zeta_0)}_{\text{unbounded}} + \hat{b}(\zeta_0)\,\psi(x,\zeta_0). \tag{4.27}$$

The Jost solution $\bar{\psi}(x,\zeta_0)$ grows exponentially as $x \to \infty$, so when $\hat{a}(\zeta_0)$ is small but not exactly zero, $\hat{\phi}(x,\zeta_0)$ grows exponentially as $x \to \infty$. Although this is in principle an error in $\hat{a}(\zeta_0)$, it will make the calculation of the norming constant $b(\zeta_0)$ ill-conditioned, as illustrated in more detail in Appendix 4.D. For a rigorous analysis of this issue, we refer to Gelash and Mullyadzhanov [55]. However, we remark that their conclusion that 'a high precision arithmetic is required to exclude these errors' disregards the possibility of using a different computation than the one they analysed. In this chapter we demonstrate that the need for high precision arithmetic is overcome by the algorithm that we propose.

The bidirectional algorithm is a numerical method to calculate the norming constants that lessens the influence of the aforementioned numerical errors significantly. It was published by Hari and Kschischang [63] and simultaneously discovered as the forward-backward method by Aref [12]. The key idea is to evaluate the norming constant $b(\zeta_0)$ not at a boundary of the window (X_-, X_+), but at a point $x = X_0$ in between, and enforce $\hat{a}(\zeta_0) = 0$ in the numerical calculation. The algorithm was originally developed and presented for use with the Zakharov–Shabat (ZS) system (albeit with a change of variables), which is equivalent to the AKNS system for the NSE. In Section 4.3.1 we extend this method to make it usable for the KdV and we formulate an overdetermined equation for the calculation of norming

[6] Definition: A function $f(x)$ is bounded on a set $\mathcal{X}$ if and only if there exists a number B such that $|f(x)| \leq B < \infty$ for all $x \in \mathcal{X}$.

constants $b(\zeta_0)$. We show in Appendix 4.E that the equations found in [12, 63] respectively for the calculation of norming constants for the NSE are actually the two different halves of this overdetermined equation. In Section 4.3.3 we propose a new criterion for the selection of the matching point X_0 that is observed to minimize the estimation error of the norming constant $\hat{b}(\zeta_0)$ by utilizing the overdetermined equation as a whole. By minimizing the estimation error of the norming constant, we also minimize the estimation error of the phase shift that is calculated from the norming constant. We summarize the criteria found in literature in Section 4.3.3 as well, but (as we will show in Section 4.4) our criterion leads to considerably more reliable estimates of the norming constants than the existing criteria.

4.3.1. Bidirectional algorithm for the KdV equation

We assume again that the potential resembles a wave packet of finite length: The potential satisfies (4.5) and we can choose a window $(X_-, X_+) \subset \mathbb{R}$ such that the potential $q(x)$ is zero outside this window. Let us cut the potential in a point $X_0 \in (X_-, X_+)$ in a left and a right part:

$$q_{\mathrm{L}}(x) := \begin{cases} q(x) & x \in (X_-, X_0), \\ 0 & \text{otherwise;} \end{cases} \tag{4.28}$$

$$q_{\mathrm{R}}(x) := \begin{cases} q(x) & x \in (X_0, X_+), \\ 0 & \text{otherwise.} \end{cases} \tag{4.29}$$

For the state transition matrices from X_- to X_0 and from X_0 to X_+ it holds by (4.24) that

$$\boldsymbol{H}_{\mathrm{S}}(X_-, X_+, \zeta) = \boldsymbol{R}_{\mathrm{S}}(\zeta)\, \boldsymbol{L}_{\mathrm{S}}(\zeta), \tag{4.30}$$

where

$$\boldsymbol{L}_{\mathrm{S}}(\zeta) := \boldsymbol{H}_{\mathrm{S}}(X_-, X_0, \zeta) \tag{4.31}$$

and

$$\boldsymbol{R}_{\mathrm{S}}(\zeta) := \boldsymbol{H}_{\mathrm{S}}(X_0, X_+, \zeta) \tag{4.32}$$

can be calculated from the potentials $q_{\mathrm{L}}(x)$ and $q_{\mathrm{R}}(x)$ respectively. Then by (4.25),

$$\begin{aligned} \boldsymbol{S}(\zeta) &= \boldsymbol{E}(\zeta X_+)\, \boldsymbol{H}_{\mathrm{S}}(X_-, X_+, \zeta)\, \boldsymbol{E}(-\zeta X_-) \\ &= \boldsymbol{E}(\zeta X_+)\, \boldsymbol{R}_{\mathrm{S}}(\zeta)\, \underbrace{\boldsymbol{E}(-\zeta X_0)\, \boldsymbol{E}(\zeta X_0)}_{=\boldsymbol{I}}\, \boldsymbol{L}_{\mathrm{S}}(\zeta)\, \boldsymbol{E}(-\zeta X_-) = \boldsymbol{S}_{\mathrm{R}}(\zeta)\, \boldsymbol{S}_{\mathrm{L}}(\zeta), \end{aligned} \tag{4.33}$$

where $\boldsymbol{S}_{\mathrm{R}}(\zeta)$ and $\boldsymbol{S}_{\mathrm{L}}(\zeta)$ are the scattering matrices for the respective potentials $q_{\mathrm{R}}(x)$ and $q_{\mathrm{L}}(x)$. By (4.13) and Cramer's rule,

$$\boldsymbol{S}_{\mathrm{R}}^{-1}(\zeta) = \begin{bmatrix} \bar{a}_{\mathrm{R}}(\zeta) & -\bar{b}_{\mathrm{R}}(\zeta) \\ -b_{\mathrm{R}}(\zeta) & a_{\mathrm{R}}(\zeta) \end{bmatrix}. \tag{4.34}$$

We multiply (4.33) for $\zeta = \zeta_0$ from the left by $\boldsymbol{S}_{\mathrm{R}}^{-1}(\zeta_0)$ and from the right by $\begin{bmatrix} 1 & 0 \end{bmatrix}^{\top}$ to find after substitution of $a(\zeta_0) = 0$ that

$$\begin{bmatrix} -\bar{b}_{\mathrm{R}}(\zeta_0) \\ a_{\mathrm{R}}(\zeta_0) \end{bmatrix} b(\zeta_0) = \begin{bmatrix} a_{\mathrm{L}}(\zeta_0) \\ b_{\mathrm{L}}(\zeta_0) \end{bmatrix}, \tag{4.35}$$

which is an overdetermined equation from which $b(\zeta_0)$ is to be solved. If $\boldsymbol{S}_{\mathrm{R}}(\zeta_0)$ and $\boldsymbol{S}_{\mathrm{L}}(\zeta_0)$ are the exact scattering matrices of any potential $\hat{q}(x)$ at an exact eigenvalue thereof, (4.35) will be consistent. Hence, an inconsistency in (4.35) indicates a numerical error that is not due to e.g. discretisation of the potential. The extend to which (4.35) is consistent appears to depend heavily on the choice of the matching point X_0, an observation we exploit in Section 4.3.3 to formulate a new criterion for this choice. To facilitate the discussion thereof, we use the two rows of (4.35) to define two separate estimators for the norming constant:

$$\hat{b}_1(\zeta_0) := -a_{\mathrm{L}}(\zeta_0) \,/\, \bar{b}_{\mathrm{R}}(\zeta_0), \tag{4.36}$$
$$\hat{b}_2(\zeta_0) := b_{\mathrm{L}}(\zeta_0) \,/\, a_{\mathrm{R}}(\zeta_0), \tag{4.37}$$

which depend implicitly on X_0 via (4.28) and (4.29).

In the derivation above we have not posed any restriction on the way ζ_0, (the first row of) $\boldsymbol{S}_{\mathrm{R}}(\zeta_0)$, and (the first column of) $\boldsymbol{S}_{\mathrm{L}}(\zeta_0)$ are calculated numerically. The bidirectional algorithm is hence independent on the numerical method by which (4.17) is solved. This may for example be an exponential integrator method (e.g. [16, 32, 93, 101]) or a collocation method (e.g. [156, §2.4.3], [143],[142, §8.2]). Furthermore we are free to choose the basis for this calculation. In the S basis (4.35) becomes, after left-multiplication by $\mathrm{e}^{\mathrm{j}\zeta X_-}\boldsymbol{E}(-\zeta X_0)$,

$$\begin{bmatrix} -R_{\mathrm{S}12}(\zeta_0) \\ R_{\mathrm{S}11}(\zeta_0) \end{bmatrix} b(\zeta_0)\mathrm{e}^{\mathrm{j}\zeta(X_-+X_+)} = \begin{bmatrix} L_{\mathrm{S}11}(\zeta_0) \\ L_{\mathrm{S}21}(\zeta_0) \end{bmatrix}. \tag{4.38}$$

With the aid of Appendix 4.C we could readily express (4.35) in any other basis we may wish to use. For example in the ubiquitous AKNS basis we obtain after left-multiplying (4.35) by $\mathrm{e}^{\mathrm{j}\zeta X_-}\boldsymbol{E}(-\zeta X_0)$ the equivalent expression

$$\begin{bmatrix} -\frac{R_{\mathrm{A}12}(\zeta_0)}{2\mathrm{j}\zeta_0} \\ R_{\mathrm{A}11}(\zeta_0) + \frac{R_{\mathrm{A}12}(\zeta_0)}{2\mathrm{j}\zeta_0} \end{bmatrix} b(\zeta_0)\mathrm{e}^{\mathrm{j}\zeta(X_-+X_+)} = \begin{bmatrix} L_{\mathrm{A}11}(\zeta_0) + \frac{L_{\mathrm{A}12}(\zeta_0)}{2\mathrm{j}\zeta_0} \\ -L_{\mathrm{A}11}(\zeta_0) - \frac{L_{\mathrm{A}12}(\zeta_0)}{2\mathrm{j}\zeta_0} + 2\mathrm{j}\zeta_0 L_{\mathrm{A}21}(\zeta_0) + L_{\mathrm{A}22}(\zeta_0) \end{bmatrix}, \tag{4.39}$$

which demonstrates that the use of the AKNS basis for the KdV is possible at the cost of more complicated equations compared to the S basis.

In Appendix 4.E we link the formulation of (4.35) to the previous work on the bidirectional algorithm, which was solely aimed at calculations for the NSE in the AKNS basis. In short, both [63] and [12] develop the bidirectional algorithm as described in this chapter, but [63] finds only (4.36), whereas [12] finds only (4.37) as the estimator for the norming constant. They furthermore use different criteria to select the matching point, as we will discuss in Section 4.3.3. However, since we are proposing a criterion that aims to minimize the error in the phase shift, we need to discuss first – in the next subsection – how this error is affected by an error in the norming constant.

4.3.2. Phase shift error

We have seen that we can calculate the phase shifts φ_n appearing in (4.4) from the norming constants $b(\zeta_{0n})$ with (4.15). From a numerical estimation of the norming constant,

$\hat{b}(\zeta_{0n})$, we can thus calculate an estimation of the phase shift, $\hat{\varphi}_n$. It is readily verified that the estimation error in the phase shift satisfies

$$\hat{\varphi}_n - \varphi_n = \tfrac{1}{2}\ln\big(\hat{b}(\zeta_{0n}) \,/\, b(\zeta_{0n})\big), \tag{4.40}$$

when all other variables that appear in (4.15) remain the same. Hence an additive error in the phase shift is directly related to a multiplicative error in the norming constant.

We must further consider what happens when this multiplicative error is negative. (We will see in Section 4.4 that this occurs commonly for calculations of the norming constant according to the benchmark algorithms.) In that case (4.40) evaluates to a complex number, which is a meaningless result. Empirically, when we reconstruct a potential from an NFT spectrum that is modified by flipping the sign of one or more norming constants, we obtain a completely different potential. Hence, a proper measure for the phase shift error is

$$\mathcal{E}(\hat{b}, b; \zeta_0) := \begin{cases} \frac{1}{2}\ln\big(\hat{b}(\zeta_0)/b(\zeta_0)\big) & \hat{b}(\zeta_0)/b(\zeta_0) > 0, \\ \infty & \hat{b}(\zeta_0)/b(\zeta_0) \leq 0. \end{cases} \tag{4.41}$$

Errors in numerical algorithms for the calculation of the NFT emerge for example due to discretisation of the potential, where there is a trade-off between the error and the required number of computations. For an exponential integrator method the required number of computations depends on the number of samples D and the relative error in the result is typically of the order $O(D^{-p})$ for some positive integer p. We remark that then the error measure defined in (4.41) converges at the same rate, i.e.

$$\left|\frac{\hat{b}(\zeta_0) - b(\zeta_0)}{b(\zeta_0)}\right| = O(D^{-p}) \qquad \Longleftrightarrow \qquad |\mathcal{E}(\hat{b}, b; \zeta_0)| = O(D^{-p}), \tag{4.42}$$

which can be shown by Taylor expansion.

4.3.3. Choice of matching point

We have shown in Section 4.3.1 how the bidirectional algorithm can be used for the calculation of the norming constants for the KdV. Two questions are left to answer in this subsection: How should we choose the matching point X_0 and how do we find the optimal estimate of a norming constant from the overdetermined equation (4.35)? In this subsection we summarize the existing criteria for choosing the matching point that are known in the literature for the NSE and we propose a new criterion. In Section 4.4 we will demonstrate with numerical examples that the existing criteria are not suitable for arbitrary vanishing potentials, whereas the criterion we propose provides the most accurate estimate of the norming constant in every case. By following the proposed criterion the error between the two estimates – (4.36) and (4.37) respectively – will generally become negligible compared to the discretisation error, in which case we can simply select any of the two as the numerical norming constant.

Suppose that the potential $q(x)$ is known on a grid $x \in \{X_- + m\varepsilon - \frac{1}{2}\varepsilon \mid m \in \{1, 2, \dots, D\}\}$, where the step size $\varepsilon := (X_+ - X_-)/D$. The natural candidates for the matching point are the points exactly in between: $X_0 \in \mathcal{X}_0 := \{X_- + m\varepsilon \mid m \in \{1, 2, \dots, D-1\}\}$. We propose to select the matching point that both minimizes the relative error between the

two estimates of the norming constant ($\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ as defined in (4.36) and (4.37) respectively) and minimizes the additive error of the phase shifts that can be calculated from these. That is,

$$X_{0\,\text{proposed}} = \arg\min_{X_0 \in \mathcal{X}_0} \left| \mathcal{E}(\hat{b}_1, \hat{b}_2; \hat{\zeta}_0) \right|, \tag{4.43}$$

with $\mathcal{E}$ as defined in (4.41). The intuition behind this choice is that when the numerical error is small, the two estimates are close to each other. Conversely, when either of the two suffers from a large numerical error, the two estimates are probably far apart.

In the literature three other criteria are known to select the matching point. These were all proposed for the calculation of norming constants with respect to the NSE, but we will use them as benchmarks for the calculation with respect to the KdV anyway, as there are to our best knowledge no such criteria for the KdV.

- The *Hari criterion* [63, Algorithm 1] can be summarized as follows: Evaluate for each eigenvalue ζ_0 the scattering parameter $a_\text{L}(\zeta_0)$ as a function of the matching point X_0. Then select the matching point as

$$X_{0\,\text{Hari}} = \arg\min_{X_0 \in \mathcal{X}_0} |a_\text{L}(\zeta_0) - 0.5|. \tag{4.44}$$

- The *Aref criterion* [12, §III.B] for $X_- = -X_+$ is

$$X_{0\,\text{Aref}} = \arg\min_{x \in \mathcal{X}_0} |q(x)| \exp(-2\text{j}\zeta_0 |x|). \tag{4.45}$$

  In case $X_- \neq -X_+$, two options are given in [12, §III.A]: Shift the potential such that (X_-, X_+) shifts to $(\frac{X_+ - X_-}{-2}, \frac{X_+ - X_-}{2})$ and correct for this space translation afterwards, or pad the potential with zero on one side to enlarge the window until $X_- = -X_+$. The choice between these two affects the outcome of (4.45). We will refer to these criteria as the Aref criterion with potential shift or support extension respectively.

- The *1-norm criterion*, used by the Fast Non-linear Fourier Transform (FNFT) software library [152], is

$$X_{0\,\text{1-norm}} = \arg\min_{X_0 \in \mathcal{X}_0} \left| \|q_\text{L}(x)\|_1 - \|q_\text{R}(x)\|_1 \right|. \tag{4.46}$$

Apart from the bidirectional algorithm one could also estimate the norming constant with a naive calculation, directly from the scattering matrix:

$$\hat{b}_\text{naive}(\zeta_0) = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{S}(\zeta_0) \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{4.47}$$

We treat the naive calculation hereafter in the framework of the bidirectional algorithm by choosing

$$X_{0\,\text{naive}} = X_+ \;\Rightarrow\; \hat{b}_\text{naive}(\zeta_0) \equiv \hat{b}_2(\zeta_0), \tag{4.48}$$

where we have used that the scattering matrix $\boldsymbol{S}_{\mathrm{R}}(\zeta_0)$ of $q_{\mathrm{R}}(x) \equiv 0$ is the identity matrix. Note that $\hat{b}_1(\zeta_0)$ is undetermined for $X_{0\,\mathrm{naive}}$.

We remark that the use of a matching point alone does not define the bidirectional algorithm. Some algorithms in the literature (e.g. [142, §8.2.1]) integrate (4.17) separately from X_- to X_0 and from X_+ backwards to X_0 to calculate $\boldsymbol{L}_{\mathfrak{b}}(\zeta)$ and $\boldsymbol{R}_{\mathfrak{b}}(\zeta)$ respectively (where $\mathfrak{b}$ is their basis of choice) and then calculate $\boldsymbol{S}(\zeta)$ from the product of $\boldsymbol{L}_{\mathfrak{b}}(\zeta)$ and $\boldsymbol{R}_{\mathfrak{b}}(\zeta)$, cf. (4.33). Since $a(\zeta_0) = 0$ is not exploited in that case, such algorithms are not examples of the bidirectional algorithm.

4.4. Numerical examples

In this section we compare the numerical calculation of the norming constants according to the bidirectional algorithm with the proposed criterion to the benchmark criteria listed in Section 4.3.3. We demonstrate that the bidirectional algorithm with the proposed criterion computes the correct norming constants even in difficult examples where all other criteria fail. In each example we start with a potential $q(x)$ of which the norming constants are analytically known. Then we calculate the norming constants numerically according to our proposed criterion as well as each of the benchmark criteria and compare the results.

4.4.1. Example setup

For each example we approximate the potential $q(x)$ as a piecewise constant function $\hat{q}(x)$ that is 0 outside a window $x \in (X_-, X_+)$. The step size is a constant $\varepsilon := (X_+ - X_-)/D$, where D is the number of samples. Each step has the same value as the potential at the midpoint of the step, so $\hat{q}(x_m) = q(x_m)$ for all $x_m = X_- + m\varepsilon - \frac{1}{2}\varepsilon$, where $m \in \{1, 2, \dots, D\}$. This approximation of the potential introduces a relative error in the spectrum proportional to D^{-2} [16, 17], thus by (4.42) we expect the error according to the error measure defined in (4.41) to be of the order $O(D^{-2})$.[7]

For the approximated potential $\hat{q}(x)$ we find the eigenvalues numerically according the algorithm described in [17, §4] where we take $\boldsymbol{U}(q, \varepsilon) = \exp(\varepsilon \boldsymbol{A}_{\mathrm{S}}(x, \zeta))$ [see (4.19)] to do the calculation in the S basis for the KdV. Then for each eigenvalue we calculate the two norming constant estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ according to (4.36) and (4.37) respectively at every matching point candidate $X_0 \in \mathcal{X}_0$. Finally we find the matching point according to the proposed criterion, (4.43), as well as to the benchmark criteria, (4.44) to (4.46) and (4.48) respectively.[8] We will report both estimates of the norming constant for each of the matching point criteria (except for the naive computation), even though their respective sources make use of only one (see Appendix 4.E).

We display the results for each example in two different ways. Firstly, we choose a low number of samples D for which still the analytically known number of eigenvalues can be found. We vary the matching point X_0 and plot for every eigenvalue against that the error between the two norming constants estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ as well as the error between both of these estimates and the analytically known norming constant. Secondly, we vary the

[7] Since a piecewise constant interpolation leads (with a suitable x-grid) to an exact representation of a rectangular potential, i.e. $\hat{q}(x) \equiv q(x)$, such an example shows atypical results for all criteria and is therefore not included.

[8] In case any of the criteria does not have a unique global minimum, we choose the leftmost (lowest) matching point among the global minima.

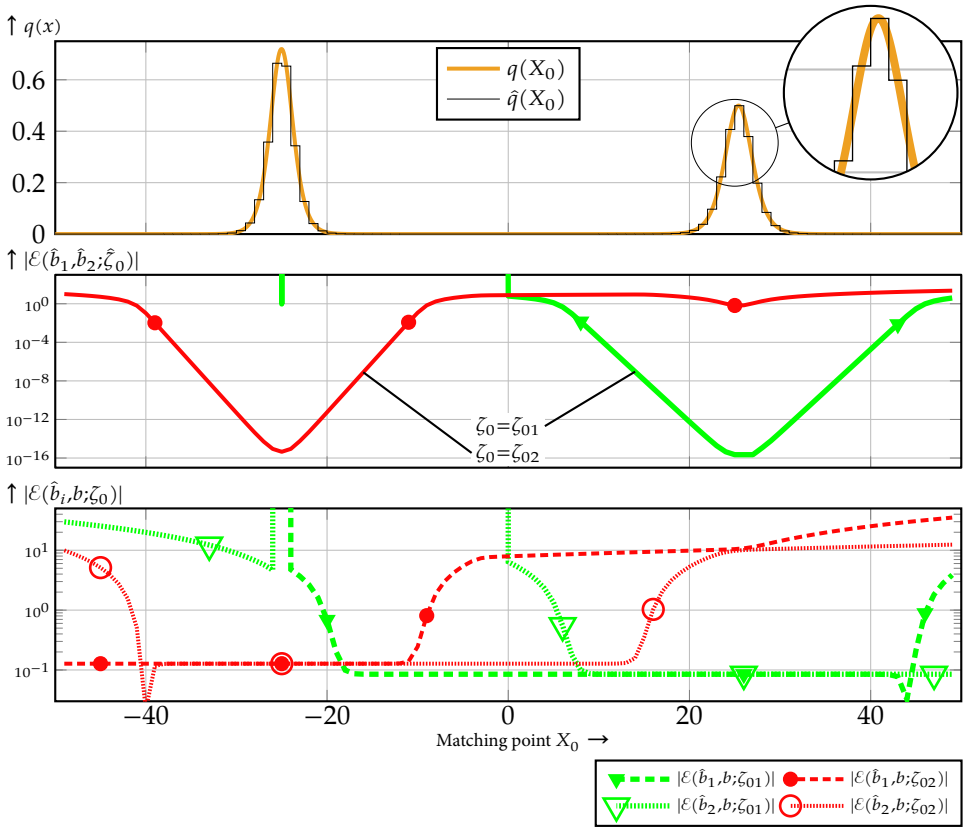


Figure 4.1: Potential $q(x)$ of 'Example 1: Two separated solitons' and the piecewise constant approximation $\hat{q}(x)$ thereof with $D = 100$ samples (top), the phase shift error between norming constant estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ as a function of the matching point X_0 (middle), and the phase shift error of these two estimates compared to the true norming constant $b(\zeta_0)$ (bottom). All three plots have the same scale on the horizontal axis.

number of samples D and plot for every eigenvalue against that the error between both of these estimates and the analytically known norming constant, where the matching point is chosen according to the proposed criterion and each of the benchmark criteria respectively. We plot these matching points as well against the number of samples.

4.4.2. Example 1: Two separated solitons

For this example we choose a pure soliton potential with two eigenvalues: $\zeta_{01} = 0.5\mathrm{j}$ with norming constant $b(\zeta_{01}) = -10^{10}$ and $\zeta_{02} = 0.6\mathrm{j}$ with $b(\zeta_{02}) = 10^{-12}$. The resulting potential consists of two well separated solitons as shown in Fig. 4.1 (top), with the soliton corresponding to ζ_{01} at $x \approx 25$ and the one corresponding to ζ_{02} at $x \approx -25$. We calculated this potential with the algorithm from [99], with an essential numerical improvement described in Appendix 4.F.

For the numerical calculation of the NFT we approximate this potential (initially) with

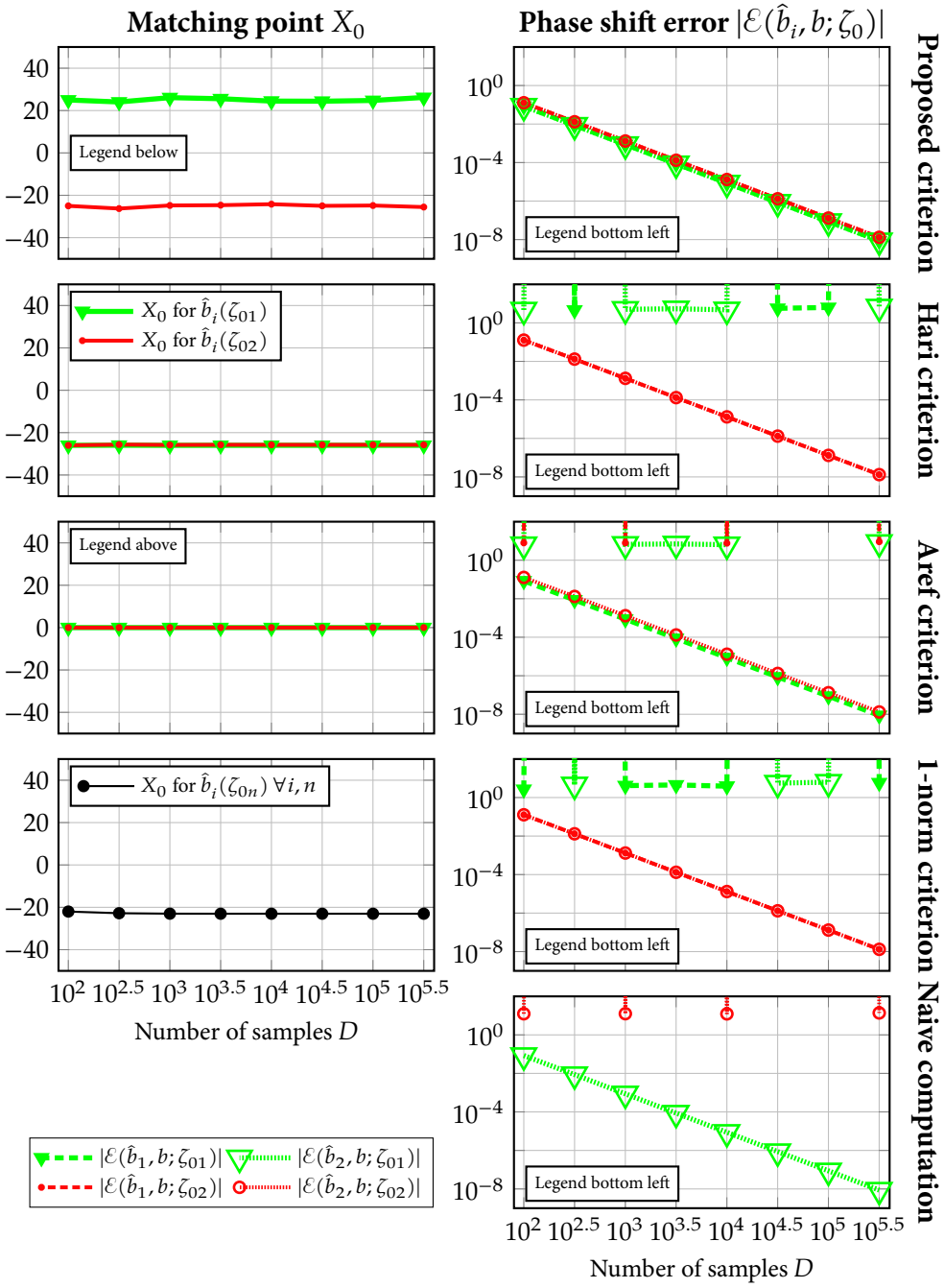


Figure 4.2: Matching point for the bidirectional algorithm and resulting phase shift errors for different numbers of samples for 'Example 1: Two separated solitons', following the proposed criterion (top row) and three benchmark criteria (second to fourth row); phase shift error with the naive computation (bottom right).

a coarse grid of $D = 100$ samples in the interval $x \in (X_-, X_+) = (-50, 50)$ as shown in Fig. 4.1 (top). Fig. 4.1 (middle) shows the error between the two estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ for all matching point candidates $X_0 \in \mathcal{X}_0$. (Here and later, errors above a certain threshold are not shown as they can become very large.) These errors attain a minimum when X_0 is at the location of the corresponding soliton. Hence, at this location (4.35) is most consistent. Away from the soliton this error becomes several orders of magnitude larger. Motivated by this observation we want to know if the numerical error is minimized by choosing the matching point for each norming constant according the proposed criterion. In Fig. 4.1 (bottom) we plot therefore the error between both respective estimates and the ground truth and indeed we see that the numerical error of both estimates for both solitons is minimal when we choose the matching point X_0 according to the proposed criterion.[9] The main difference compared to Fig. 4.1 middle is that the error reaches an error floor in the vicinity of the soliton, thereby forming bathtub shaped curves rather than V-shaped curves. This error floor is caused by the approximation of the potential by a piecewise constant function and can be lowered by reducing the step size. This is shown in Fig. 4.2, where we plot (as a function of the number of samples D) X_0 according to each of the criteria as well as the error of the resulting norming constant estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ compared to the true norming constant $b(\zeta_0)$ for both of the eigenvalues ζ_0. It can be seen that the bidirectional algorithm with the proposed criterion is the only one for which the error decay is consistently for all estimates $\mathcal{O}(D^{-2})$ – a factor 100 per decade – as expected. The other criteria find norming constants with an error that shows no convergence and that is several orders of magnitude larger for one of the eigenvalues or for one of the estimators, $\hat{b}_1(\zeta_0)$ or $\hat{b}_2(\zeta_0)$. We remark that curves that leave the graphing area vertically indicate that the neighbouring data point corresponds to an estimate of the norming constant with the opposite sign, yielding infinite error by (4.41). The reason why the benchmark criteria perform like this, is that the solitons in the potential are separated from each other. The benchmark criteria select the matching point for all eigenvalues either near one of the two solitons or right in the middle, which are clearly no suitable matching points in every case if we look at Fig. 4.1 (bottom). Hence, even for this simple potential the proposed criterion is the only one that results in only reliable estimates for the norming constants.

4.4.3. Example 2: Six partially overlapping solitons

In this example we construct a pure soliton potential with six eigenvalues, $\zeta_{0n} = 0.1nj$ for $n \in \{1, 2, \dots, 6\}$, and norming constants $b(\zeta_{0n}) = (-1)^n \exp(8n(-1.01)^n)$. The resulting potential is calculated as in Example 1 and consists of two clusters of three overlapping solitons each: The solitons for odd n cluster at $x \approx -45$, those for even n cluster at $x \approx +45$.

For the numerical calculation of the NFT we approximate this potential with a coarse grid of $D = 316$ steps in the interval $x \in (-160, 160)$. The potential and its approximation are shown in Fig. 4.3 (first plot). The error between the two estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_{2'}(\zeta_0)$ are shown in Fig. 4.1 (second plot). Again we see that for all norming constants the global

[9] Sharp dips below the error floor, such as in Fig. 4.1 (bottom) for $|\mathcal{E}(\hat{b}_2, b; \zeta_{02})|$ at $X_0 = -40$ are cases where the error in the estimate of the norming constant coincidentally cancels the error due to the approximation of the potential. Since the norming constant is a real number this is not unlikely to happen at some matching point candidate, but there is no way to find that point without knowledge of the true norming constant and exploit this effect.

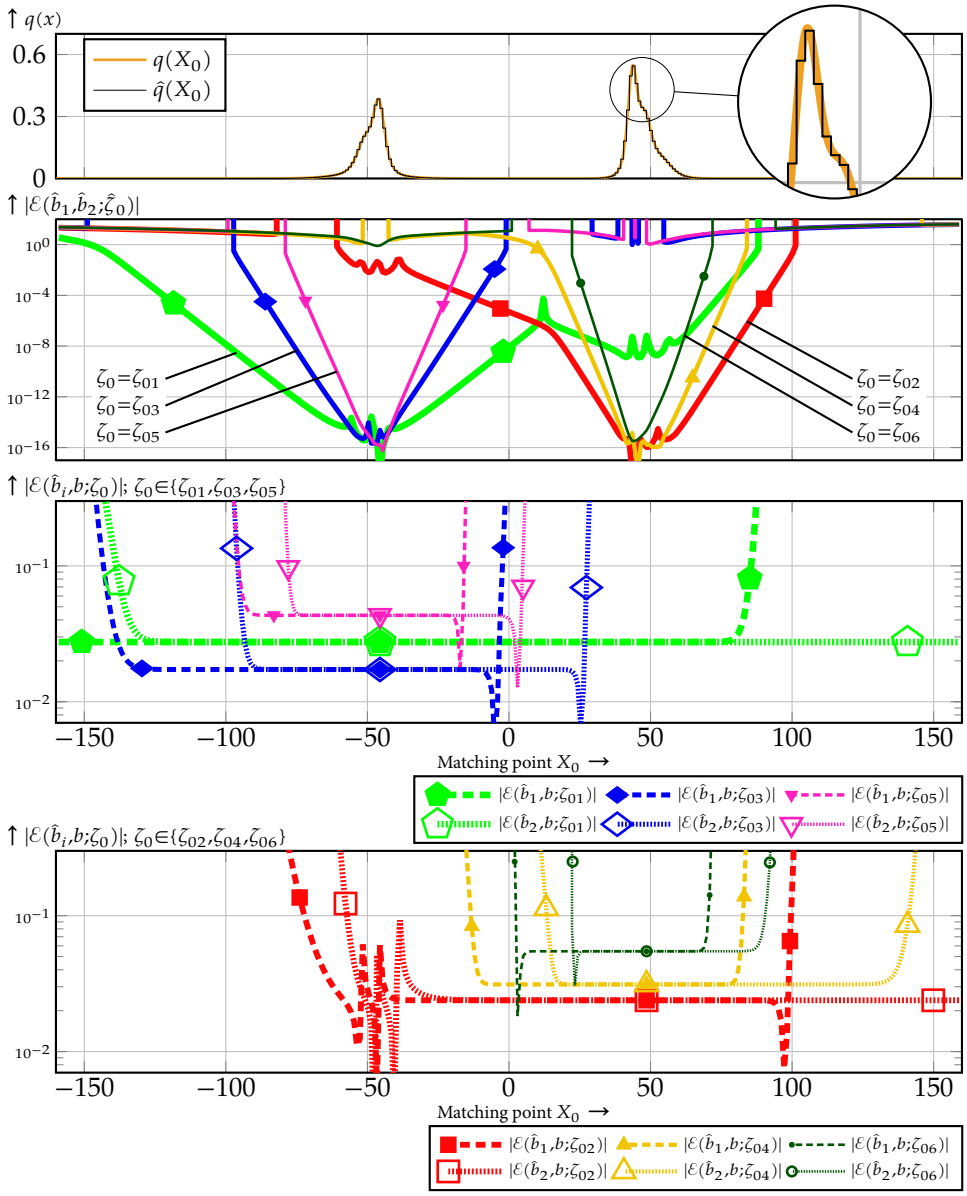


Figure 4.3: Potential $q(x)$ of 'Example 2: Six partially overlapping solitons' and the piecewise constant approximation $\hat{q}(x)$ thereof with $D = 316$ samples (first plot), the phase shift error between norming constant estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ as a function of the matching point X_0 (second plot), and the phase shift error of these two estimates compared to the true norming constant $b(\zeta_0)$ (last two plots). All four plots have the same scale on the horizontal axis.

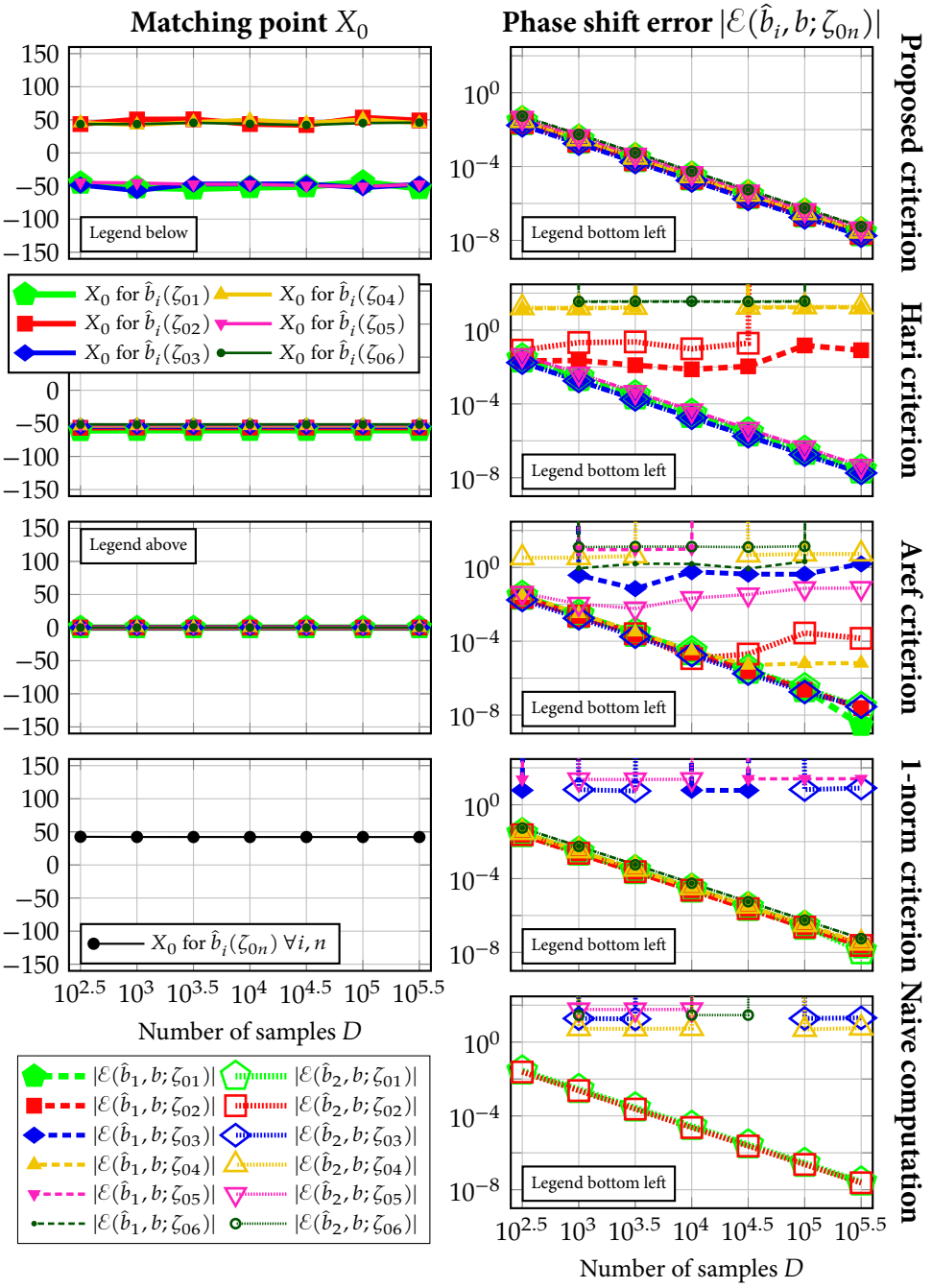


Figure 4.4: Matching point for the bidirectional algorithm and resulting phase shift errors for different numbers of samples for 'Example 2: Six partially overlapping solitons', following the proposed criterion (top row) and three benchmark criteria (second to fourth row); phase shift error with the naive computation (bottom right).

minimum of this error is for X_0 in the vicinity of the corresponding soliton. In Fig. 4.3 (last two plots) we show for each eigenvalue the error between both respective estimates and the ground truth. Similar to Example 1 we see that the numerical error of both estimates for all solitons is at a minimum when we follow the proposed criterion. Due to the larger window (X_-, X_+) we can recognise the bathtub shapes of the curves better than in Example 1, and see that the ones for $\hat{b}_1(\zeta_0)$ are shifted to the left compared to the corresponding soliton, whereas the ones for $\hat{b}_2(\zeta_0)$ are shifted to the right. Furthermore we see that the bottoms of the bathtubs become more narrow as the corresponding eigenvalue increases in magnitude, implying that choosing a good matching point becomes more important as the energy of the corresponding soliton increases.

We can lower the bottoms of the bathtubs, and thereby the achievable error of the estimate of the norming constant, by increasing the number of samples, as shown in Fig. 4.4. Again we observe that all of the benchmark criteria return several estimates of the norming constant with an error that does not decay consistently with an increase of the number of samples, or is even high throughout. The proposed algorithm in contrast returns estimates of the norming constant of which the error decays neatly at the expected rate of $O(D^{-2})$ for every eigenvalue and for both estimates thereof.

4.4.4. Example 3: Potential with non-zero continuous spectrum

For the last example we start from the potential $q'(x) = (35/64)\operatorname{sech}^2(x/4)$. It has a non-zero continuous spectrum, meaning that it is not a pure soliton potential. This potential has three eigenvalues, $\zeta_{0n} = (2n-1)\mathrm{j}/8$ for $n \in \{1,2,3\}$, with norming constants $b'(\zeta_{0n}) = (-1)^{n+1}$ [79, §5.2]. For this example we translate this potential to the right by 12π to obtain $q(x) := q'(x - 12\pi) = (35/64)\operatorname{sech}^2(x/4 - 3\pi)$. We approximate this potential with a piecewise constant function with (initially) only $D = 32$ steps in the interval $x \in (0, 80)$. This approximation $\hat{q}(x)$ and the potential $q(x)$ itself are shown in Fig. 4.5 (top). The resulting truncation is deliberately asymmetric with respect to the axis of symmetry of the potential.[10] By Lemma 2 in Appendix 4.B we can calculate that the norming constants of $q(x)$ are $b(\zeta_{0n}) = (-1)^{n+1}\exp(-24\pi\mathrm{j}\zeta_{0n})$.

In Fig. 4.5 (middle) we show the error between the two estimates of each norming constant. We see that all three curves show a global minimum near the axis of symmetry of the potential and in Fig. 4.5 (bottom) we show the errors between these two estimates and the true norming constants, which have a wide global minimum near the axis of symmetry of the potential. This minimum is quite high because of the limited number of samples.

When we increase the number of samples, we obtain the results shown in Fig. 4.6. The Hari criterion, the 1-norm criterion, and the proposed criterion select a matching point near the axis of symmetry of the potential and the errors of all their estimates of the norming constant decay at the expected rate of a factor 100 per decade, $O(D^{-2})$. The flooring that is seen in all these cases from around $D = 10^5$ samples is because the error due to the truncation to $x \in (0, 80)$ becomes dominant compared to the error due to the piecewise constant approximation itself and could hence be removed by enlarging the window (X_-, X_+). For

[10] We do this because otherwise the staircase approximation $\hat{q}(x)$ would be even symmetric around $x = 12\pi$ as well. Since for every even symmetric potential all norming constants are ± 1 by Corollary 1 in Appendix 4.B, we would obtain no error in the norming constant due to the staircase approximation at any number of samples, except for due to the error in $\hat{\zeta}_0$.

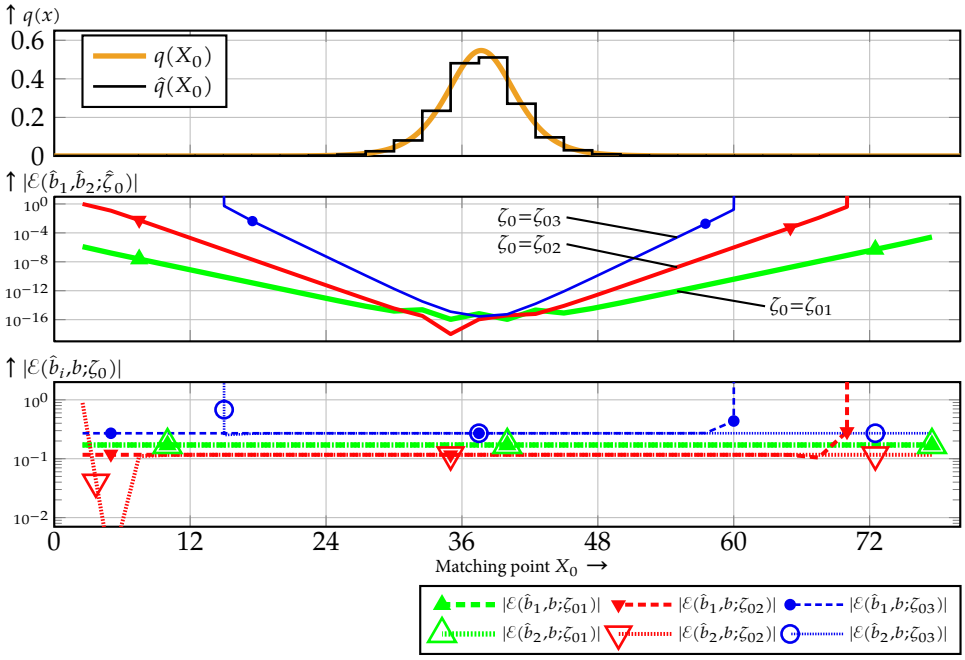


Figure 4.5: Potential $q(x)$ of 'Example 3: Potential with non-zero continuous spectrum' and the piecewise constant approximation $\hat{q}(x)$ thereof with $D = 32$ samples (top), the phase shift error between norming constant estimates $\hat{b}_1(\zeta_0)$ and $\hat{b}_2(\zeta_0)$ as a function of the matching point X_0 (middle), and the phase shift error of these two estimates compared to the true norming constant $b(\zeta_0)$ (bottom). All three plots have the same scale on the horizontal axis.

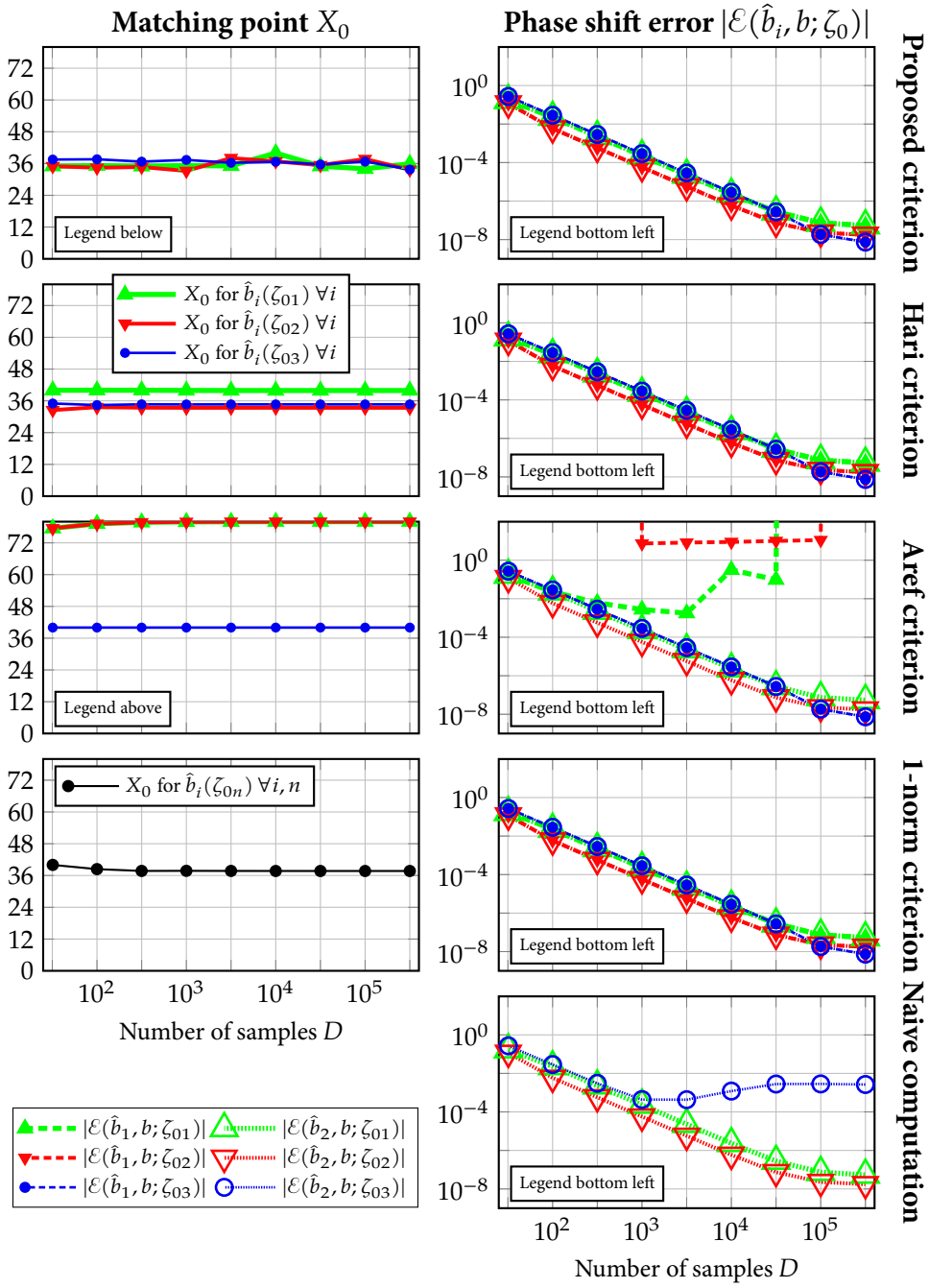


Figure 4.6: Matching point for the bidirectional algorithm and resulting phase shift errors for different numbers of samples for 'Example 3: Potential with non-zero continuous spectrum', following the proposed criterion (top row) and three benchmark criteria (second to fourth row); phase shift error with the naive computation (bottom right).

this example even the naive computation shows the expected error decay for two of the three norming constants, the third one floors at a higher level. The Aref criterion[11] selects matching points for two of the eigenvalues at the far right end of the potential, which results in two estimates of the norming constants with a non-decaying error. From this example we see that in the case of a potential where the solitons are all clustered together, the proposed criterion performs equally well as the best among the benchmark criteria.

4.5. Conclusion

In this chapter we investigated how to calculate the phase shifts of solitons in the far field of any wave packet that evolves according to the Korteweg–de Vries equation (KdV) equation. For that we in particular need to compute the so-called norming constant of each soliton. The naive method to compute norming constants is however known to be unreliable. We adapted the bidirectional algorithm, which was originally designed for the computation of norming constants for the Non-linear Schrödinger Equation (NSE), to the KdV. Furthermore we proposed a new criterion to select the matching point required for this algorithm. The criterion is based on the observation that the bidirectional algorithm actually provides two estimates of each norming constant. The proposed criterion is to select the matching point that minimizes the difference between these two estimates, which is observed to minimize also their distance to the true norming constant. We demonstrated with three examples that the proposed method performs at least as good as existing algorithms, and often several orders of magnitude better.

Our method was implemented in the FNFT software library [152] and has been applied to real-world data [26].

4.A. Definition of the scattering parameters of the KdV equation

By Abel's identity [2, p. 22], the Wronskian of any two eigenfunctions f_1 and f_2 of (4.6) is independent of x:

$$\mathrm{W}[y_1(x); y_2(x)] := y_1 y_{2x} - y_{1x} y_2; \tag{4.49}$$

$$\frac{\partial}{\partial x} \mathrm{W}[y_1(x); y_2(x)] \equiv 0. \tag{4.50}$$

Consequently, we may evaluate Wronskians of the Jost solutions, (4.8) and (4.9), at any convenient value x:

$$\mathrm{W}[\phi; \bar{\phi}] = \lim_{x \to -\infty} \mathrm{W}[\phi; \bar{\phi}] = 2\mathrm{j}\zeta; \tag{4.51}$$

$$\mathrm{W}[\bar{\psi}; \psi] = \lim_{x \to \infty} \mathrm{W}[\bar{\psi}; \psi] = 2\mathrm{j}\zeta. \tag{4.52}$$

[11] If there is a matching point X_0 for which $q(X_0) = 0$, then it minimizes (4.45). Consequently the Aref criterion with support extension returns a matching point in the zero padding of the potential, which does not give up to par results. Therefore we use the Aref criterion with potential shift in this example.

Because these Wronskians are non-zero for $\zeta \neq 0$, both sets of Jost solutions are linearly independent for $\zeta \neq 0$. To relate these two sets, define the scattering parameters as

$$a(\zeta) := \frac{\mathrm{W}[\phi(x,\zeta,t);\psi(x,\zeta,t)]}{2\mathrm{j}\zeta}, \tag{4.53}$$

$$b(\zeta,t) := \frac{\mathrm{W}[\bar{\psi}(x,\zeta,t);\phi(x,\zeta,t)]}{2\mathrm{j}\zeta}, \tag{4.54}$$

$$\bar{a}(\zeta) := \frac{\mathrm{W}[\bar{\psi}(x,\zeta,t);\bar{\phi}(x,\zeta,t)]}{2\mathrm{j}\zeta} \equiv a(-\zeta), \tag{4.55}$$

$$\bar{b}(\zeta,t) := \frac{\mathrm{W}[\bar{\phi}(x,\zeta,t)\,\psi(x,\zeta,t)]}{2\mathrm{j}\zeta} \equiv b(-\zeta,t). \tag{4.56}$$

such that (4.10) holds. The scattering parameters $a(\zeta)$ and $\bar{a}(\zeta)$ do not depend on t [4, Eq. 3.8].

4.B. Properties of the scattering parameters of the KdV equation

Lemma 2 (Space translation) *If the scattering matrix of a potential $q(x)$ is $\boldsymbol{S}(\zeta)$ as in (4.11), then the scattering matrix of the potential $q'(x) \equiv q(x - x_0)$ is*

$$\begin{aligned}\boldsymbol{S}'(\zeta) &= \boldsymbol{E}(\zeta x_0)\,\boldsymbol{S}(\zeta)\,\boldsymbol{E}(-\zeta x_0) \\ &= \begin{bmatrix} a(\zeta) & \bar{b}(\zeta)\exp(2\mathrm{j}\zeta x_0) \\ b(\zeta)\exp(-2\mathrm{j}\zeta x_0) & \bar{a}(\zeta) \end{bmatrix},\end{aligned} \tag{4.57}$$

where $\boldsymbol{E}(\zeta x)$ is defined in (4.21).

Proof The scattering problem for $q'(x)$ is equivalent to the scattering problem for $q(x')$ with $x' := x - x_0$. The Jost solutions in the translated coordinate are

$$\phi(x,\zeta) \equiv \phi(x' + x_0,\zeta) \equiv \phi(x',\zeta)\exp(-\mathrm{j}\zeta x_0), \tag{4.58}$$

$$\bar{\phi}(x,\zeta) \equiv \bar{\phi}(x' + x_0,\zeta) \equiv \bar{\phi}(x',\zeta)\exp(\mathrm{j}\zeta x_0), \tag{4.59}$$

$$\bar{\psi}(x,\zeta) \equiv \bar{\psi}(x' + x_0,\zeta) \equiv \bar{\psi}(x',\zeta)\exp(-\mathrm{j}\zeta x_0), \tag{4.60}$$

$$\psi(x,\zeta) \equiv \psi(x' + x_0,\zeta) \equiv \psi(x',\zeta)\exp(\mathrm{j}\zeta x_0). \tag{4.61}$$

Equation (4.57) follows from filling these out in the definitions of the scattering parameters, (4.53) to (4.56). ■

Remark 7 (Change of x-coordinate frame)
Although $b(\zeta)$ and $\bar{b}(\zeta)$ do not depend on x (see Appendix 4.A), $b(\zeta)$ and $\bar{b}(\zeta)$ do change when we change from x to a translated coordinate $x' := x - x_0$, as we see in Lemma 2. For the continuous spectrum ζ is real and then this change affects $\angle\, b(\zeta)$ and $\angle\, \bar{b}(\zeta)$, which is analogous to a phase shift of the ordinary Fourier transform under a translation of the space coordinate. For the discrete spectrum ζ_0 is imaginary, and then this change affects $|b(\zeta_0)|$ and $|\bar{b}(\zeta_0)|$ instead. □

Lemma 3 (Space reversal) *If the scattering matrix of a potential $q(x)$ is $\boldsymbol{S}(\zeta)$ as in (4.11), then the scattering matrix of the potential $q'(x) \equiv q(-x)$ is*

$$\boldsymbol{S}'(\zeta) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \boldsymbol{S}^{-1}(\zeta) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} a(\zeta) & -b(\zeta) \\ -\bar{b}(\zeta) & \bar{a}(\zeta) \end{bmatrix}. \tag{4.62}$$

Proof The scattering problem for $q'(x)$ is equivalent to the scattering problem for $q(x')$ with $x' := -x$. The Jost solutions in the mirrored coordinate are

$$\phi(x,\zeta) \equiv \psi(x',\zeta), \tag{4.63}$$
$$\bar{\phi}(x,\zeta) \equiv \bar{\psi}(x',\zeta), \tag{4.64}$$
$$\bar{\psi}(x,\zeta) \equiv \bar{\phi}(x',\zeta), \tag{4.65}$$
$$\psi(x,\zeta) \equiv \phi(x',\zeta). \tag{4.66}$$

Equation (4.62) follows from filling these out in the definitions of the scattering parameters, (4.53) to (4.56). ■

Corollary 1 (Even symmetric potential) *For a potential that is even symmetric, i.e. $q(x) \equiv q(-x)$, by Lemma 3*

$$b(\zeta) \equiv -\bar{b}(\zeta). \tag{4.67}$$

Furthermore, since $a(\zeta_0) = 0$ for every eigenvalue ζ_0, from (4.13) and (4.67) norming constants of even symmetric potentials satisfy $b(\zeta_0) = \pm 1$. □

4.C. Other bases for solving the Schrödinger eigenvalue problem

The derivations in Chapter 4 for the S basis can be translated into other bases by means of similarity transformations. Let $\mathfrak{b}$ indicate any such basis, then the basis dependent variables are related to the S basis as

$$\mathbf{V}_\mathfrak{b}(x,\zeta) = \boldsymbol{T}_\mathrm{S}^\mathfrak{b}(\zeta)\, \mathbf{V}_\mathrm{S}(x,\zeta), \tag{4.68}$$
$$v_\mathfrak{b}(x,\zeta) = \boldsymbol{T}_\mathrm{S}^\mathfrak{b}(\zeta)\, v_\mathrm{S}(x,\zeta), \tag{4.69}$$
$$\boldsymbol{\Phi}_\mathfrak{b}(x,\zeta) = \boldsymbol{T}_\mathrm{S}^\mathfrak{b}(\zeta)\, \boldsymbol{\Phi}_\mathrm{S}(x,\zeta), \tag{4.70}$$
$$\boldsymbol{A}_\mathfrak{b}(x,\zeta) = \boldsymbol{T}_\mathrm{S}^\mathfrak{b}(\zeta)\, \boldsymbol{A}_\mathrm{S}(x,\zeta)\, \boldsymbol{T}_\mathfrak{b}^\mathrm{S}(\zeta), \tag{4.71}$$
$$\boldsymbol{H}_\mathfrak{b}(x_1,x_2,\zeta) = \boldsymbol{T}_\mathrm{S}^\mathfrak{b}(\zeta)\, \boldsymbol{H}_\mathrm{S}(x_1,x_2,\zeta)\, \boldsymbol{T}_\mathfrak{b}^\mathrm{S}(\zeta), \tag{4.72}$$

where

$$\boldsymbol{T}_\mathfrak{b}^\mathrm{S}(\zeta) = \left(\boldsymbol{T}_\mathrm{S}^\mathfrak{b}(\zeta)\right)^{-1}. \tag{4.73}$$

Hereafter we discuss some bases found in literature.

4.C.1. AKNS basis

The AKNS system for the KdV [4] is found with the choice

$$\mathbf{V}_\mathrm{A}(\zeta) := \begin{bmatrix} \mathrm{j}\zeta - \frac{\partial}{\partial x} \\ 1 \end{bmatrix} \qquad \Rightarrow \qquad \boldsymbol{A}_\mathrm{A}(x,\zeta) = \begin{bmatrix} -\mathrm{j}\zeta & q(x) \\ r(x) & \mathrm{j}\zeta \end{bmatrix}, \tag{4.74}$$

where $r(x) \equiv -1$. The transformation matrices that relate the AKNS basis for the KdV to the S basis are

$$\boldsymbol{T}_{\mathrm{S}}^{\mathrm{A}}(\zeta) = \begin{bmatrix} 2\mathrm{j}\zeta & 0 \\ 1 & 1 \end{bmatrix}, \qquad \boldsymbol{T}_{\mathrm{A}}^{\mathrm{S}}(\zeta) = \frac{1}{2\mathrm{j}\zeta} \begin{bmatrix} 1 & 0 \\ -1 & 2\mathrm{j}\zeta \end{bmatrix}. \tag{4.75}$$

We remark that (4.22) and (4.23) expressed in the AKNS basis are equivalent to [4, Eqs. A3.1 and A3.2] respectively. That shows that our definitions are consistent with [4, Appendix 3], in accordance with the claim in Remark 5.

By setting $r(x)$ differently, the AKNS system can be used for other non-linear differential equations [4]. In particular, the choice $r(x) = \pm q^*(x)$ results in a system for the NSE, which is also known as the ZS system. In this appendix we mean by AKNS the AKNS system with $r(x) \equiv -1$ and refer to the NSE version as ZS.

A variant of the AKNS system is

$$\mathbf{V}_{\bar{\mathrm{A}}}(\zeta) := \mathbf{V}_{\mathrm{A}}(-\zeta) \Rightarrow \boldsymbol{A}_{\bar{\mathrm{A}}}(x,\zeta) = \boldsymbol{A}_{\mathrm{A}}(x,-\zeta), \tag{4.76}$$

and is for the KdV related the S basis by the transformation matrices

$$\boldsymbol{T}_{\mathrm{S}}^{\bar{\mathrm{A}}}(\zeta)] \begin{bmatrix} 0 & -2\mathrm{j}\zeta \\ 1 & 1 \end{bmatrix}, \qquad \boldsymbol{T}_{\bar{\mathrm{A}}}^{\mathrm{S}}(\zeta) = \frac{1}{2\mathrm{j}\zeta} \begin{bmatrix} 1 & 2\mathrm{j}\zeta \\ -1 & 0 \end{bmatrix}. \tag{4.77}$$

This basis leads for the KdV to a numerically more accurate calculation of the continuous spectrum [110, Footnote 3].

4.C.2. Companion basis

Another choice results in a companion system:

$$\mathbf{V}_{\mathrm{C}}(\zeta) := \begin{bmatrix} 1 \\ \frac{\partial}{\partial x} \end{bmatrix} \qquad \Rightarrow \qquad \boldsymbol{A}_{\mathrm{C}}(x,\zeta) = \begin{bmatrix} 0 & 1 \\ (\mathrm{j}\zeta)^2 - q(x) & 0 \end{bmatrix}. \tag{4.78}$$

The transformation matrices that relate this basis to the S basis are

$$\boldsymbol{T}_{\mathrm{S}}^{\mathrm{C}}(\zeta) = \begin{bmatrix} 1 & 1 \\ -\mathrm{j}\zeta & \mathrm{j}\zeta \end{bmatrix}, \qquad \boldsymbol{T}_{\mathrm{C}}^{\mathrm{S}}(\zeta) = \frac{1}{2\mathrm{j}\zeta} \begin{bmatrix} \mathrm{j}\zeta & -1 \\ \mathrm{j}\zeta & 1 \end{bmatrix}. \tag{4.79}$$

The advantage of this basis is that it only requires computations on real numbers for both the discrete spectrum ($\zeta \in \mathbb{I}$) and the continuous spectrum ($\zeta \in \mathbb{R}$), whereas the other bases in this appendix need complex arithmetic for the continuous spectrum. This advantage is employed by e.g. [106, §17.5.1][17, Eq. 5.3].

4.C.3. Osborne basis

The following choice leads to a close relative of the S basis:

$$\mathbf{V}_{\mathrm{O}}(\zeta) := \frac{1}{2\mathrm{j}\zeta} \begin{bmatrix} \frac{\partial}{\partial x} - \mathrm{j}\zeta \\ \frac{\partial}{\partial x} + \mathrm{j}\zeta \end{bmatrix} \qquad \Rightarrow \qquad \boldsymbol{A}_{\mathrm{O}}(x,\zeta) = \begin{bmatrix} -\mathrm{j}\zeta + \frac{q(x)}{2\mathrm{j}\zeta} & -\frac{q(x)}{2\mathrm{j}\zeta} \\ \frac{q(x)}{2\mathrm{j}\zeta} & \mathrm{j}\zeta - \frac{q(x)}{2\mathrm{j}\zeta} \end{bmatrix}. \tag{4.80}$$

The transformation matrices are given by

$$\boldsymbol{T}_{\mathrm{S}}^{\mathrm{O}}(\zeta) = \boldsymbol{T}_{\mathrm{O}}^{\mathrm{S}}(\zeta) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.81}$$

This basis is implicitly used in Osborne [101] and Provenzale and Osborne [114], although with all matrix equations transposed compared to this appendix.

4.D. Naive computation of the norming constant

A naive numerical computation of the norming constant of a certain eigenvalue ζ_0 would use (4.10) to compute the scattering matrix $\boldsymbol{S}(\zeta_0)$ which contains the norming constant $b(\zeta_0)$. However, this computation is ill-conditioned. As an illustration thereof we will add here a particular small perturbation to this computation and show that this has a large effect on the result.

Let us consider a potential $q(x)$ that is zero for all x outside a window (X_-, X_+) and with at least one eigenvalue ζ_0 for which we have numerically computed the scattering matrix $\hat{\boldsymbol{S}}(\hat{\zeta}_0)$, in which $\hat{a}(\hat{\zeta}_0) \neq 0$. Now we perturb the potential near $x = X_+$ as

$$q_\mu(x) = \begin{cases} q(X_+) + \mu & X_+ - \varepsilon < x < X_+, \\ q(x) & \text{otherwise}, \end{cases} \tag{4.82}$$

where we assume the potential to be approximately constant for $x \in (X_+ - \varepsilon, X_+)$. Then the scattering matrix of the perturbed potential becomes

$$\hat{\boldsymbol{S}}_\mu(\hat{\zeta}_0) = \boldsymbol{E}(\hat{\zeta}_0 X_+)\, \boldsymbol{G}_{\mathrm{S}\mu}(\hat{\zeta}_0)\, \boldsymbol{E}(\hat{\zeta}_0 X_+)\, \hat{\boldsymbol{S}}(\hat{\zeta}_0), \tag{4.83}$$

where

$$\boldsymbol{G}_{\mathrm{S}\mu}(\hat{\zeta}_0) := \boldsymbol{H}_{\mathrm{S}\mu}(X_+ - \varepsilon, X_+, \hat{\zeta}_0)\, (\boldsymbol{H}_{\mathrm{S}}(X_+ - \varepsilon, X_+, \hat{\zeta}_0))^{-1} \tag{4.84}$$

first steps back, thereby cancelling the unperturbed potential $q(x)$, then steps forward with the perturbed potential $q_\mu(x)$. Using that for a locally constant potential

$$\boldsymbol{H}_{\mathrm{S}}(X_+ - \varepsilon, X_+, \hat{\zeta}_0) = \exp(\varepsilon \boldsymbol{A}_{\mathrm{S}}(X_+, \hat{\zeta}_0)) \tag{4.85}$$

and likewise for the perturbed potential, it can be shown that

$$\boldsymbol{G}_{\mathrm{S}\mu}(\hat{\zeta}_0) = \begin{bmatrix} 1-\delta & -\delta \\ \delta & 1+\delta \end{bmatrix} + O(\varepsilon^2), \tag{4.86}$$

where

$$\delta := \mu\varepsilon \,/\, (-2\mathrm{j}\hat{\zeta}_0). \tag{4.87}$$

Finally after filling out the first order approximation of (4.86) in (4.83), we find

$$\frac{\hat{b}_\mu(\hat{\zeta}_0)}{\hat{b}(\hat{\zeta}_0)} \approx 1 + \delta \cdot \left(\frac{\hat{a}(\hat{\zeta}_0)}{\hat{b}(\hat{\zeta}_0)} \exp(-2\mathrm{j}\hat{\zeta}_0 X_+) + 1 \right). \tag{4.88}$$

The numerical error in the computed norming constant is typically large because of the exponential factor in (4.88). As an example suppose that we have an even symmetric potential, then $b(\zeta_0) = \pm 1$ for all eigenvalues (see Corollary 1 in Appendix 4.B). Then if $|\delta\hat{a}(\hat{\zeta}_0)| \approx 10^{-15}$ with $\delta \ll 0.1$, the estimation error according to (4.88) is already 10% for $-\mathrm{j}\zeta_0 X_+ \approx 16$.

It may seem as if we could apply a change of variables $x \to x + x_0$ to make $\left(\hat{a}(\hat{\zeta}_0) / \hat{b}(\hat{\zeta}_0)\right) \exp(-2\mathrm{j}\hat{\zeta}_0 X_+)$ arbitrarily small by lowering X_+. However, application of Lemma 2 in Appendix 4.B shows that such a change of variables leaves this quantity unchanged.

4.E. Link with previous work on the non-linear Schrödinger equation

Hari and Kschischang [63] use for the NSE a variant of the ZS basis in which any state transition matrix on an interval that covers all non-zero parts af the potential equals the scattering matrix. To see this let us write [4, Eqs. 3.1 & 3.3] as [cf. (4.22) and (4.23)]

$$\lim_{x\to-\infty} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{E}(\zeta x) \begin{bmatrix} \boldsymbol{\phi}(x,\zeta) & \bar{\boldsymbol{\phi}}(x,\zeta) \end{bmatrix} = \boldsymbol{I}, \tag{4.89}$$

$$\lim_{x\to\infty} \boldsymbol{E}(\zeta x) \begin{bmatrix} \boldsymbol{\phi}(x,\zeta) & \bar{\boldsymbol{\phi}}(x,\zeta) \end{bmatrix} = \begin{bmatrix} a(\zeta) & \bar{b}(\zeta) \\ b(\zeta) & -\bar{a}(\zeta) \end{bmatrix}. \tag{4.90}$$

The variables are changed in Hari and Kschischang [63, §IIIA] such that

$$\boldsymbol{\Phi}_{\mathrm{H}}(x,\zeta) := \boldsymbol{E}(\zeta x) \begin{bmatrix} \boldsymbol{\phi}(x,\zeta) & \bar{\boldsymbol{\phi}}(x,\zeta) \end{bmatrix}. \tag{4.91}$$

Then for a potential that is zero for all x outside an interval (X_-, X_+) it follows that [cf. (4.24)]

$$\boldsymbol{\Phi}_{\mathrm{H}}(X_+,\zeta) = \boldsymbol{H}_{\mathrm{H}}(X_-, X_+, \zeta)\, \boldsymbol{\Phi}_{\mathrm{H}}(X_-,\zeta), \tag{4.92}$$

where

$$\boldsymbol{H}_{\mathrm{H}}(X_-, X_+, \zeta) = \begin{bmatrix} a(\zeta) & -\bar{b}(\zeta) \\ b(\zeta) & \bar{a}(\zeta) \end{bmatrix}. \tag{4.93}$$

Hence propagating $\begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$ forward up till the matching point results in $\begin{bmatrix} a_{\mathrm{L}}(\zeta_0) & b_{\mathrm{L}}(\zeta_0) \end{bmatrix}^{\mathsf{T}}$, whereas propagating $\begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathsf{T}}$ backward up till the matching point results in $\begin{bmatrix} \bar{b}_{\mathrm{R}}(\zeta_0) & a_{\mathrm{R}}(\zeta_0) \end{bmatrix}^{\mathsf{T}}$. Finally $b(\zeta_0)$ is calculated using only the first element of both results:

$$\hat{b}(\zeta_0) = a_{\mathrm{L}}(\zeta_0) \,/\, \bar{b}_{\mathrm{R}}(\zeta_0), \tag{4.94}$$

which is $\hat{b}_1(\zeta_0)$, the estimator according to the first row of (4.35), with the sign difference explained in Remark 5.

Aref [12] uses the same basis as Hari and Kschischang [63], but calculates the norming constant as

$$b(\zeta_0) = \frac{S_{\mathrm{L}\,21}(\zeta_0)}{S_{\mathrm{R}\,11}(\zeta_0)} = \frac{b_{\mathrm{L}}(\zeta_0)}{a_{\mathrm{R}}(\zeta_0)}. \tag{4.95}$$

This is $\hat{b}_2(\zeta_0)$, the estimate that only makes use of is the second row of (4.35).

4.F. Generation of a multisoliton potential for the KdV equation

For the generation of the multisoliton potential in Section 4.4.2, we made use of an algorithm from [99]. However, we observed that the calculation was only well-conditioned near the centre of the solitons. In order to use it for the 'tails' of the solitons, we adapted the algorithm as described below.

4.F.1. Simplification of the determinant equation

Consider from [99] the unnumbered equation between (22) and (23). For this equation to be valid, the denominator matrix must be invertible. Hence, dividing the two determinants is equivalent to taking the determinant of the product between the inverse of the denominator matrix (hereafter $\boldsymbol{D}$) and the numerator matrix (hereafter $\boldsymbol{N}$). Since these two matrices are equal except for the last column, the aforementioned product (hereafter $\boldsymbol{Q}$) has a particular structure that considerably simplifies taking its determinant:

$$\boldsymbol{Q} \coloneqq \boldsymbol{D}^{-1}\boldsymbol{N} = \begin{bmatrix} 1 & 0 & \cdots & 0 & * \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & * \\ 0 & \cdots & 0 & 1 & * \\ 0 & \cdots & 0 & 0 & c \end{bmatrix}, \tag{4.96}$$

where $*$ denotes a number that is not of interest. Hence, $|\boldsymbol{N}|/|\boldsymbol{D}| = c$ and this quantity c can be found by solving $\boldsymbol{D}\boldsymbol{y} = \boldsymbol{n}$ for $\boldsymbol{y}$ with a suitable linear solver, where $\boldsymbol{n}$ is the last column of $\boldsymbol{N}$ and c is the last element of $\boldsymbol{y}$.

4.F.2. Scaling of the matrix equation

Consider again from [99] the unnumbered equation between (22) and (23) and the alternative calculation thereof described above. We observed that these equations become badly scaled in the 'tails' of the potential of the KdV, because half of the β_n parameters tend to ∞. This can be solved by reformulation of the equation in terms of the α_n parameters, i.e. by left-multiplying both $\boldsymbol{N}$ and $\boldsymbol{D}$ by a diagonal matrix $\operatorname{diag}(\alpha_1, \alpha_2, \ldots, \alpha_{2N}) = \operatorname{diag}(\beta_1^{-1}, \beta_2^{-1}, \ldots, \beta_{2N}^{-1})$. This results in a condition number of the denominator matrix $\boldsymbol{D}$ that tends to a constant as $|x| \to \infty$. We remark that in some cases this scaling results in a worse condition number near the solitons centres. For such cases we compare the condition numbers between the original scaling and the one described here for each potential sample and choose the best conditioned one for the calculation of that potential sample.

5

Accurate inverse NFT computation for the discrete KdV spectrum

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 $O(N^2)$, 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 $O(N^3)$ and are often even less accurate for other reasons. Our algorithm has a complexity of $O(N^2)$. 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.

5.1. 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 [9, 24, 61, 76, 80]; blood pressure waves in arteries [1, 95]; internal waves in the ocean, Rosby waves in the atmosphere, plasma waves, acoustic waves [37, 137]; and electrical waves in non-linear transmission lines [119].

Parts of this chapter are accepted for publication as P. J. Prins and S. Wahls. "An accurate $O(N^2)$ floating point algorithm for the Crum transform of the KdV equation". In: *Communications in Nonlinear Science and Numerical Simulation* 102.105782 (Nov. 2021), pp. 1–25. ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2021.105782,

From a mathematical point of view, the KdV has gained fame as the first example of a non-linear PDE that is *(Lax-)integrable* [53, 81]. That is, there exist a direct and an inverse Non-linear Fourier Transform (NFT) (the latter is also known as Inverse Scattering Transform (IST) [4]) 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 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., [29, 32, 111, 132, 142, 152]. 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 [7, 24, 26, 33, 61, 101].

In this chapter, we are concerned with the inverse KdV-NFT for signals that satisfy vanishing boundary conditions (defined later by (5.1)). The spectrum of such a signal consists in general of two parts: a continuous spectrum and a discrete spectrum. The discrete spectrum consists in its turn of $N \geq 0$ eigenvalues. The inverse KdV-NFT reconstructs the signal from its spectrum. This can be expressed analytically as a Gel'fand–Levitan–Marchenko (GLM) integral equation [5, Eqs. (1.3.37a–c)]. Some numerical approaches are based on solving this equation, e.g. [122]. Another possibility is to rewrite the inverse KdV-NFT as a Riemann–Hilbert problem and then use a numerical solver for Riemann–Hilbert problems, e.g. [141–143]. For a closely related computation, namely of the inverse NFT with respect to the Nonlinear Schrödinger Equation (NSE), Vasylchenkova [147, §3.3] reported several numerical difficulties with both of these approaches. Yousefi and Kschischang [159] also considered a Riemann–Hilbert formulation for the inverse NSE-NFT and found that the system would occasionally become ill-conditioned. It is not clear to which extend the numerical difficulties for the computation of the inverse NSE-NFT also apply to algorithms for the computation of the inverse KdV-NFT. The accuracy of the algorithm in [142, 143] is bounded uniformly in time and space for any given spectrum and chosen order of the used Chebyshev approximation, but the possible influence of the 'difficulty' of the spectrum on the accuracy is not discussed. The algorithm in [141] makes it possible to use a high or variable precision arithmetic to trade computational cost for accuracy. If sufficient computational resources are available, this makes the accuracy of the spectrum itself the limiting factor in the accuracy of the reconstructed signal. Sacks and Shin [122, §4.2] point at exponentially growing terms in the computation of the inverse KdV-NFT as a possible cause of numerical difficulties. They suggest that it is convenient from a computational point of view to split the computation in two parts, cf. [40, Sect. 3], [28, Chap. xvii.3.2]. First the inverse transform of the (suitably pre-compensated) continuous spectrum is computed. The N eigenvalues of the discrete spectrum are added in a second stage, which is also known as the *Crum transform*.

The troublesome exponentially growing terms become part of the computation of the Crum transform. However [122, §4.2] argues that the two-stage approach makes it simpler to mitigate their effect on the accuracy. With this approach in mind, we present in this chapter an algorithm to compute the Crum transform accurately in floating point arithmetic.

From an analytic perspective, the Crum transform for the KdV can be carried out with an exact direct computation, 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 [30, Chap. 5] studied a Crum transform with respect to the 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 [54] 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 chapter, 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 [40, 122]. The intermediate results generated by this sequential 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 5.3.2.) A non-sequential algorithm for the KdV Crum transform that is free of artificial singularities was presented in [99]. 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 [111, Sect. IV/App. F], we proposed some modifications to the algorithm of [99] 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 [26], 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 chapter 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, without using any (functional) approximation. 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 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 chapter, although some of the underlying ideas can be carried over.

We structured the remainder of this chapter as follows. In Section 5.2 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 chapter. In Section 5.3, 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 5.4. The advantages of the new algorithm are illustrated with numerical examples in Section 5.5. The chapter is concluded in Section 5.6. Lengthy proofs and derivations of some of the results presented in this chapter are put in appendices, to improve the readability of the main text.

5.2. Preliminaries

5.2.1. 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.

5.2.2. 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 5.2.3. For a comprehensive introduction to the NFT we refer to [6] 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

$$\lim_{|x|\to\infty} q(x,t_0) = 0 \qquad \text{and} \qquad \int_{-\infty}^{\infty} |q(x,t_0)|\,(1+|x|)\,\mathrm{d}x < \infty. \tag{5.1}$$

To find $q(x,t)$ for any t, we first compute the NFT spectrum of the initial condition [110, 111]. Then we propagate this spectrum forward or backward in time by means of simple formulas

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

[53], [5, 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]

$$\left(\frac{\partial^2}{\partial x^2} + q(x,t_0)\right) f(x,\zeta,t_0) = (\mathrm{j}\zeta)^2 f(x,\zeta,t_0), \tag{5.2}$$

where

$$\mathrm{j} := \sqrt{-1}. \tag{5.3}$$

We call signals $f(x,\zeta,t_0)$ that fulfil (5.2) *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

$$\phi(x,\pm\zeta,t_0) \to \mathrm{e}^{\mp\mathrm{j}\zeta x} \text{ as } x \to -\infty; \qquad \psi(x,\pm\zeta,t_0) \to \mathrm{e}^{\pm\mathrm{j}\zeta x} \text{ as } x \to \infty. \tag{5.4}$$

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]

$$\phi(x,\zeta,t_0) \equiv a(\zeta)\,\psi(x,-\zeta,t_0) + b(\zeta,t_0)\,\psi(x,\zeta,t_0). \tag{5.5}$$

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

$$R(\zeta,t_0) := b(\zeta,t_0)/a(\zeta) \in \mathbb{C} \qquad \forall\zeta \in \mathbb{R} \quad \{0\}. \tag{5.6}$$

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

$$\mathcal{D}(t_0) := \{(\gamma_m, b(\mathrm{j}\gamma_m,t_0)) \in \mathbb{R}_{>0} \times \mathbb{R}\backslash\{0\} \mid a(\mathrm{j}\gamma_m) = 0\}. \tag{5.7}$$

To simplify the notation, we omit the dependence on time in the remainder of this chapter. 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 (5.5) reduces to

$$\phi(x,\mathrm{j}\gamma_m) \equiv b(\mathrm{j}\gamma_m)\,\psi(x,\mathrm{j}\gamma_m). \tag{5.8}$$

It can be shown that all eigenvalues lie on the upper half of the imaginary axis, i.e. $\gamma_m > 0$ for all $m \in \mathbb{N}_1^M = \{1,2,\dots,M\}$, where M is the cardinality of the discrete spectrum. Furthermore, all eigenvalues are known to be simple [79, pp. 50–53]. Throughout this chapter 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, 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

[2] Equation (5.2) establishes a close connection between the NFT and quantum scattering, cf. [28].

[3] Trajectories of (5.2) are to be understood as solutions in a weak sence [109, pp. 1–2], so henceforward equations involving trajectories of (5.2) only need to hold almost everywhere.

[4] See e.g. [111, App. A] for the explicit definitions of $a(\zeta)$ and $b(\zeta,t_0)$.

1. reformulating the Schrödinger equation (5.2) as a Gel'fand–Levitan–Marchenko integral equation,
2. solving an associated Riemann–Hilbert problem, and
3. transferring the Schrödinger equation into the time domain and exploiting causality principles.

See e.g. [28, Chap. xvii] for an exhaustive survey and [122, 142, 143, 146] for some recently proposed numerical methods.

As already mentioned in Section 5.1, 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 chapter, 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.

5

5.2.3. Crum transform

In this subsection, we briefly review the Crum transform. Detailed derivations can be found e.g. in [40, Sect. 3], [89], or [59, Chap. 1].

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

$$\mathrm{W}[g_1(x); \dots; g_N(x)] := \begin{vmatrix} g_1(x) & \frac{\mathrm{d}}{\mathrm{d}x} g_1(x) & \cdots & (\frac{\mathrm{d}}{\mathrm{d}x})^{N-1} g_1(x) \\ \vdots & \vdots & & \vdots \\ g_N(x) & \frac{\mathrm{d}}{\mathrm{d}x} g_N(x) & \cdots & (\frac{\mathrm{d}}{\mathrm{d}x})^{N-1} g_N(x) \end{vmatrix}. \tag{5.9}$$

Crum [38] used Wronskians to construct new solutions of the Schrödinger equation (5.2) 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) [40, Thm. 6], [89, Sect. 2.1] *Let $f(x,\zeta) = f^{(0)}(x,\zeta)$ be any trajectory of (5.2) for a potential $q(x) = q^{(0)}(x)$ that satisfies (5.1). Let $f(x,\zeta_n) = \vartheta_n(x)$ for $n \in \mathbb{N}_1^N = \{1, 2, \dots, N\}$ be N specific trajectories of (5.2) for $q(x) = q^{(0)}(x)$ and $\zeta = \zeta_n$. Then*

$$f^{(N)}(x,\zeta) = \mathsf{C}_{(0)}^{(N)} f^{(0)}(x,\zeta) := \frac{\mathrm{W}[\vartheta_1(x); \vartheta_2(x); \dots; \vartheta_N(x); f^{(0)}(x,\zeta)]}{\mathrm{W}[\vartheta_1(x); \vartheta_2(x); \dots; \vartheta_N(x)]} \tag{5.10}$$

is a trajectory of (5.2) for the potential

$$q^{(N)}(x) = q^{(0)}(x) + 2\frac{\mathrm{d}^2}{\mathrm{d}x^2} \ln(\,\mathrm{W}[\vartheta_1(x); \vartheta_2(x); \dots; \vartheta_N(x)]). \tag{5.11}$$

□

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 [89, Sect. 2.1]. The Crum transform can thus be implemented in one shot, adding all N eigenvalues at once (e.g. [89, Sect. 2.1], [40, Thm. 6]); or sequentially, by repeated application of the Darboux transform (e.g., [59, Chap. 1], [40, Thm. 2], [28, Chap. XVII.3.2], [122, Sect. 4.2]). (In Section 5.3.1, 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, (5.9), 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, (5.10) implies that the Crum transform maps all of the seed trajectories to zero, i.e.

$$\mathsf{C}_{(0)}^{(N)}\, \vartheta_n(x) \equiv 0 \qquad \forall n \in \mathbb{N}_1^N. \tag{5.12}$$

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

$$\vartheta_n(x) = \phi^{(0)}(x, \mathrm{j}\kappa_n) - (-1)^N \beta_n\, \psi^{(0)}(x, \mathrm{j}\kappa_n), \qquad \beta_n \neq 0, \qquad \kappa_n > 0, \tag{5.13}$$

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 [40, Thm. 6], [89, eq. (2.2.10)]

$$\phi^{(N)}(x,\zeta) = \left(\prod_{n=1}^{N} \tfrac{1}{\kappa_n - \mathrm{j}\zeta}\right) \mathsf{C}_{(0)}^{(N)}\, \phi^{(0)}(x,\zeta), \qquad \mathrm{j}\zeta \notin \{\kappa_1, \kappa_2, \dots, \kappa_n\}; \tag{5.14}$$

$$\psi^{(N)}(x,\zeta) = \left(\prod_{n=1}^{N} \tfrac{-1}{\kappa_n - \mathrm{j}\zeta}\right) \mathsf{C}_{(0)}^{(N)}\, \psi^{(0)}(x,\zeta), \qquad \mathrm{j}\zeta \notin \{\kappa_1, \kappa_2, \dots, \kappa_n\}. \tag{5.15}$$

Using (5.12) to (5.15) and the linearity of $\mathsf{C}_{(0)}^{(N)}$, we find

$$0 \equiv \mathsf{C}_{(0)}^{(N)}\, \vartheta_n(x) \equiv \mathsf{C}_{(0)}^{(N)} \left(\phi^{(0)}(x, \mathrm{j}\kappa_n) - (-1)^N \beta_n \psi^{(0)}(x, \mathrm{j}\kappa_n)\right) \equiv \tag{5.16}$$

$$\left(\mathsf{C}_{(0)}^{(N)} \phi^{(0)}(x, \mathrm{j}\kappa_n)\right) - \beta_n \left((-1)^N\, \mathsf{C}_{(0)}^{(N)} \psi^{(0)}(x, \mathrm{j}\kappa_n)\right) \propto \phi^{(N)}(x, \mathrm{j}\kappa_n) - \beta_n \psi^{(N)}(x, \mathrm{j}\kappa_n)$$

$$\Rightarrow \quad \phi^{(N)}(x, \mathrm{j}\kappa_n) \equiv \beta_n \psi^{(N)}(x, \mathrm{j}\kappa_n), \tag{5.17}$$

where '$\propto$' denotes proportionality. By comparing (5.17) with (5.8) 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 (5.13), the effect of the Crum transform on the discrete spectrum (5.7) is

$$\left\{\left(\gamma_m^{(N)}, b^{(N)}(\mathrm{j}\gamma_m^{(N)})\right)\right\} = \left\{\left(\gamma_m^{(0)}, (-1)^N b^{(0)}(\mathrm{j}\gamma_m^{(0)})\right)\right\} \cup \left\{(\kappa_n, \beta_n)\,\middle|\, n \in \mathbb{N}_1^N\right\}, \tag{5.18}$$

[5] The Crum transform can also be used to remove eigenvalues [40, Thm. 3], but in this chapter we only discuss the use of the Crum transform to add eigenvalues.

[6] To see this, let $\mathsf{C}_{(0)}^{(N)}$ operate on both sides of (5.8) and then substitute (5.14) and (5.15).

where $\cup$ denotes the union of sets. The effect on the continuous spectrum (5.6) is[7]

$$R^{(N)}(\zeta) = \left(\prod_{n=1}^{N} \frac{\kappa_n - \mathrm{j}\zeta}{\kappa_n + \mathrm{j}\zeta}\right) R^{(0)}(\zeta). \tag{5.19}$$

5.2.4. Dressing method

For numerical computations the Schrödinger equation (5.2) is typically rewritten as a system of first order ordinary differential equations. Thereto one defines an operator

$$\mathbf{V}(x,\zeta) := \begin{bmatrix} v_{11}(x,\zeta) + v_{12}(x,\zeta)\frac{\partial}{\partial x} \\ v_{21}(x,\zeta) + v_{22}(x,\zeta)\frac{\partial}{\partial x} \end{bmatrix}, \tag{5.20}$$

where

$$\begin{vmatrix} v_{11}(x,\zeta) & v_{12}(x,\zeta) \\ v_{22}(x,\zeta) & v_{22}(x,\zeta) \end{vmatrix} \neq 0, \tag{5.21}$$

such that (5.2) can be rewritten as

$$\tfrac{\partial}{\partial x} f(x,\zeta) = A(x,\zeta) f(x,\zeta), \tag{5.22}$$

where

$$f(x,\zeta) := \mathbf{V}(x,\zeta) f(x,\zeta) \in \mathbb{C}^{2\times 1}. \tag{5.23}$$

The simplest choice for the operator $\mathbf{V}(x,\zeta)$ is

$$\mathbf{V}_{\mathrm{C}}(x,\zeta) := \begin{bmatrix} 1 \\ \frac{\partial}{\partial x} \end{bmatrix}. \tag{5.24}$$

Other common choices for the operator $\mathbf{V}(x,\zeta)$ have been discussed in [111, App. C], see also Appendix 5.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 [86]. Let

$$f(x,\zeta) := \mathbf{V}(x,\zeta) f(x,\zeta) \tag{5.25}$$

and

$$\boldsymbol{\vartheta}_n(x) := \mathbf{V}(x,\zeta)\, \vartheta_n(x) \tag{5.26}$$

denote the vector-valued versions of the trajectories in Theorem 1. The Crum transform can then be expressed as

$$q^{(N)}(x) = q^{(0)}(x) + \Delta q^{(N)}_{(0)}(x) \tag{5.27}$$

[7] To see this, let $\mathsf{C}^{(N)}_{(0)}$ operate on both sides of (5.5), use the linearity of $\mathsf{C}^{(N)}_{(0)}$ and then substitute (5.14) and (5.15). See also [40, Thm. 6].

and

$$f^{(N)}(x,\zeta) = \boldsymbol{C}_{(0)}^{(N)}(x,\zeta)\, f^{(0)}(x,\zeta), \tag{5.28}$$

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

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

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 $\boldsymbol{\vartheta}_n(x)$ are independent linear combinations of $\vartheta_n(x)$ and $\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_n(x)$ by (5.20). 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 equation (5.2) [40, p. 177–178]. (See also Appendix 5.B.2.) Therefore, the computation of $\Delta q_{(0)}^{(N)}(x)$ and $\boldsymbol{C}_{(0)}^{(N)}(x,\zeta)$ 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

$$\Delta q_{(0)}^{(1)}(x) = -2\,q^{(0)}(x) + 2\frac{(\kappa_1\,\vartheta_1(x) - \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x})(\kappa_1\,\vartheta_1(x) + \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x})}{\vartheta_1^2(x)} \tag{5.29}$$

and

$$\boldsymbol{C}_{\mathrm{C}(0)}^{(1)}(x,\zeta) = \begin{bmatrix} -\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} / \vartheta_1(x) & 1 \\ (\mathrm{j}\zeta)^2 - \frac{(\kappa_1\,\vartheta_1(x) - \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x})(\kappa_1\,\vartheta_1(x) + \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x})}{\vartheta_1^2(x)} & -\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} / \vartheta_1(x) \end{bmatrix}, \tag{5.30}$$

where

$$\vartheta_1(x) = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) \qquad \text{and} \qquad \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x). \tag{5.31}$$

The derivation of (5.29) and (5.30) is shown in Appendix 5.B.3.

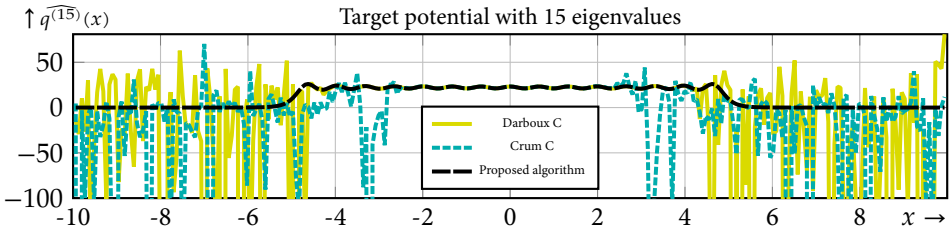


Figure 5.1: Target potential with 15 eigenvalues, as sequentially computed with (5.27), (5.29) and (5.30) (——, Darboux C), by the dressing method for the same operator for $N = 15$ (- - - -, Crum C), and by the proposed algorithm (— — —) respectively, all in 64 bit floating point arithmetic. The graph 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 5.5.3. 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.

5.3. 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. 5.1, where 15 eigenvalues were added to a zero background potential. The first two lines in Fig. 5.1 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 (5.27), (5.29) and (5.30), 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 5.4 of this chapter. 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 (5.29) and (5.30) 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 5.4.

5.3.1. 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.

One might intuitively expect that adding eigenvalues sequentially accumulates more numerical error than adding them all at once. However, Fig. 5.1 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 $O(N^3)$ FLoating point

OPerations (FLOPs), because for each fixed x one either needs to solve a square linear system of size N, or calculate N-th order determinants. On the other hand, the computational complexity of adding N eigenvalues sequentially is only in the order of $O(N^2)$ FLOPs (e.g., [28, 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.

5.3.2. Poles in an intermediate potential

The Crum transform of a background potential that satisfies the vanishing boundary conditions (5.1) 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 (5.1). This happens if the Wronskian of the seed trajectories in (5.10) and (5.11) vanishes at one or more points x cf. [59, Sect. 1.1.2], [73, 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. 5.2. 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 (5.1). However, when the larger eigenvalue is added first, the intermediate potential has a pole (cf. [5, eq. (3.4.10)], [73, 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 [66, 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 (SL) oscillation theory that the eigenfunctions of (5.2) 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 [165, Thm. 10.12.1-(4)], [51, 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 (5.4) and $\phi(x, \zeta)$ changes sign $M - m$ times before approaching $+\infty$, where $\psi(x, \zeta) > 0$ by (5.4), (5.8) implies

$$\operatorname{sign}(b(\mathrm{j}\gamma_m)) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^M, \tag{5.32}$$

where

$$\gamma_1 < \gamma_2 < \ldots < \gamma_M. \tag{5.33}$$

If (5.32) 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 (5.32). When we use the

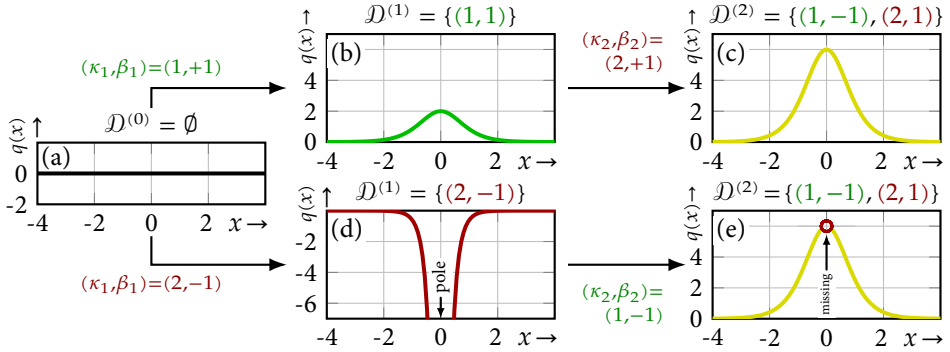


Figure 5.2: Calculation of the target potential with spectrum $R(\zeta) \equiv 0$, $\mathcal{D}^{(2)} = \{(1,-1),(2,1)\}$ from a zero background potential by adding the eigenvalues in different orders with Darboux transforms. 1) Smallest first: From (a) — $q^{(0)}(x) \equiv 0$, to (b) — $q^{(1)}(x) = 2\,\mathrm{sech}^2(x)$, 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).

5

Crum transform to add an odd number of eigenvalues, the norming constants of the background potential all change sign according to (5.18). 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 [56, p. 269], [149, p. 1388]. Its norming constant must be positive in the target spectrum, cf. [87, Sect. 6.6], [149, 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 (5.32), cf. [123].

Rule 3. When removing one or more eigenvalues, those eigenvalues must satisfy [8, Thm.]. (No additional condition on the norming constants is needed here, because these satisfy (5.32) 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 5.4 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.

5.3.3. Catastrophic cancellation for large $|x|$

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. 5.1. 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 $\boldsymbol{C}^{(N)}_{\mathrm{C}(0)}(x,\zeta)$ (e.g. (5.29) and (5.30)) one encounters factors of the form $(\kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x))$. When the seed trajectories are chosen as in (5.13), one can verify with (5.4) 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 $(\kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x))$ 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 (5.20) it follows that

$$\kappa_n\,\vartheta_n(x) \mp \frac{\mathrm{d}}{\mathrm{d}x}\,\vartheta_n(x) = \tag{5.34}$$

$$\left.\frac{\begin{bmatrix} -\mathrm{j}\zeta\, v_{22}(x,\zeta) \mp v_{21}(x,\zeta) & \mathrm{j}\zeta\, v_{12}(x,\zeta) \pm v_{11}(x,\zeta) \end{bmatrix} \boldsymbol{\vartheta}_n(x)}{\begin{vmatrix} v_{11}(x,\zeta) & v_{12}(x,\zeta) \\ v_{21}(x,\zeta) & v_{22}(x,\zeta) \end{vmatrix}}\right|_{\zeta=\mathrm{j}\kappa_n}.$$

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 (5.29) 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 5.3.2 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.

5.3.4. Seed trajectories growing exponentially as $|x| \to \infty$

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 (5.13), as was already mentioned in Section 5.3.3. Therefore, they tend to grow larger than the largest representable number in IEEE 754 double precision floating point numbers (`realmax` $\approx \mathrm{e}^{710}$ [66, 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 5.4) is

[8] From (5.4), (5.11) and (5.13) one can verify that $q^{(N)}(x) - q^{(0)}(x) \propto \exp(\mp 2\kappa_1 x)$ as $x \to \pm\infty$.

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 5.3.3, we choose[9]

$$\mathbf{V}_{\mathrm{E}}(x,\zeta) := \frac{1}{2\mathrm{j}\zeta}\begin{bmatrix} \exp(\mathrm{j}\zeta x)(\mathrm{j}\zeta - \frac{\partial}{\partial x}) \\ \exp(-\mathrm{j}\zeta x)(\mathrm{j}\zeta + \frac{\partial}{\partial x}) \end{bmatrix}. \tag{5.35}$$

In Appendix 5.B.1 we relate $\mathbf{V}_{\mathrm{E}}(x,\zeta)$ to other operators one encounters in the literature.

5.3.5. Seed trajectories remaining exponentially large or small

The operator in (5.35) 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 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 [111, App. B]. In that case we can see from (5.4), (5.13) and (5.35) 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 $ab \neq 0$), (5.10) and (5.11) are independent of η_n. Since the dressing method is an exact reformulation of the Crum transform, the same must hold for (5.27), 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

$$h_n(x) = 2^{-\left\lfloor \frac{1}{2}\,\mathrm{lb}|[\,1\;\,0\,]\,\boldsymbol{\vartheta}_n(x)| + \frac{1}{2}\,\mathrm{lb}|[\,0\;\,1\,]\,\boldsymbol{\vartheta}_n(x)| \right\rceil}, \tag{5.36}$$

where

$$\mathrm{lb}(a) = \log_2(a), \tag{5.37}$$

$$\lfloor\cdot\rceil = \mathrm{round}(\cdot). \tag{5.38}$$

[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.

[10] In IEEE 754 double precision floating point numbers `realmin` $\approx \mathrm{e}^{-708}$ is the smallest representable number above zero [66, Sect. 2.1].

5.3.6. 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 (5.30) could be calculated from $\boldsymbol{\vartheta}_{1\mathrm{E}}(x)$ as

$$-\frac{\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)}{\vartheta_1(x)} = \frac{\kappa_1\left[-\exp(\kappa_1 x) \quad \exp(-\kappa_1 x)\right]\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}{\left[\exp(\kappa_1 x) \quad \exp(-\kappa_1 x)\right]\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}, \tag{5.39}$$

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:

$$-\frac{\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)}{\vartheta_1(x)} = \begin{cases} \dfrac{\kappa_1\left[-\exp(2\kappa_1 x) \quad 1\right]\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}{\left[\exp(2\kappa_1 x) \quad 1\right]\boldsymbol{\vartheta}_{1\mathrm{E}}(x)} & x \le 0, \\ \dfrac{\kappa_1\left[-1 \quad \exp(-2\kappa_1 x)\right]\boldsymbol{\vartheta}_{1\mathrm{E}}(x)}{\left[1 \quad \exp(-2\kappa_1 x)\right]\boldsymbol{\vartheta}_{1\mathrm{E}}(x)} & x > 0. \end{cases} \tag{5.40}$$

5.3.7. 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 $\frac{\mathrm{d}\,\vartheta_n(x)}{\mathrm{d}x} / \vartheta_n(x)$.[11] See e.g. (5.120) and (5.127). 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 (5.11) 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 n-th seed trajectory vanishes, $\frac{\mathrm{d}\,\vartheta_n(x)}{\mathrm{d}x} / \vartheta_n(x)$ 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.

5.3.8. 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.

[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 $2N+2$.

5.4. Proposed algorithm

In this section, we present our new algorithm for computing Crum transforms for the KdV. The algorithm is based on the error analysis from Section 5.3.

5.4.1. 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 (5.22) using the operator $\mathbf{V}_{\mathrm{E}}$ defined in (5.35). In Appendix 5.B.4, it is shown that the N-fold Crum transform with $N \in \{1,2\}$ takes the form

$$q^{(N)}(x) = (-1)^N q^{(0)}(x) + \left[\begin{smallmatrix} 4 & 0 \end{smallmatrix}\right] \boldsymbol{M}^{(N)}_{N-2}(x,0) \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]; \tag{5.41}$$

$$f^{(N)}_{\mathrm{E}}(x,\zeta) = \boldsymbol{C}^{(N)}_{\mathrm{E}(0)} f^{(0)}_{\mathrm{E}}(x,\zeta), \tag{5.42}$$

where

$$\boldsymbol{C}^{(N)}_{\mathrm{E}(0)} = \sum_{m=-1}^{N} (\mathrm{j}\zeta)^m \boldsymbol{M}^{(N)}_m(x,\zeta). \tag{5.43}$$

In order to evaluate (5.41) and (5.42), 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 $\boldsymbol{\vartheta}_{1\mathrm{E}1} := \left[\begin{smallmatrix} 1 & 0 \end{smallmatrix}\right] \boldsymbol{\vartheta}_{1\mathrm{E}}(x)$ and $\boldsymbol{\vartheta}_{1\mathrm{E}2} := \left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right] \boldsymbol{\vartheta}_{1\mathrm{E}}(x)$. We propose to calculate the matrices as

$$\boldsymbol{M}^{(1)}_{-1}(x,\zeta) = \begin{cases} \frac{2\kappa_1^2 \boldsymbol{\vartheta}_{1\mathrm{E}1} \boldsymbol{\vartheta}_{1\mathrm{E}2}}{w_-^2(\boldsymbol{\vartheta}_1)} \left[\begin{smallmatrix} \mathrm{e}^{2\kappa_1 x} & \mathrm{e}^{2(\mathrm{j}\zeta+\kappa_1)x} \\ -\mathrm{e}^{-2(\mathrm{j}\zeta-\kappa_1)x} & -\mathrm{e}^{2\kappa_1 x} \end{smallmatrix}\right] & x \le 0, \\ \frac{2\kappa_1^2 \boldsymbol{\vartheta}_{1\mathrm{E}1} \boldsymbol{\vartheta}_{1\mathrm{E}2}}{w_+^2(\boldsymbol{\vartheta}_1)} \left[\begin{smallmatrix} \mathrm{e}^{-2\kappa_1 x} & \mathrm{e}^{2(\mathrm{j}\zeta-\kappa_1)x} \\ -\mathrm{e}^{-2(\mathrm{j}\zeta+\kappa_1)x} & -\mathrm{e}^{-2\kappa_1 x} \end{smallmatrix}\right] & x > 0; \end{cases} \tag{5.44}$$

$$\boldsymbol{M}^{(1)}_{0}(x,\zeta) = \begin{cases} \frac{\kappa_1 \left(\boldsymbol{\vartheta}_{1\mathrm{E}2} - \boldsymbol{\vartheta}_{1\mathrm{E}1} \mathrm{e}^{2\kappa_1 x}\right)}{w_-(\boldsymbol{\vartheta}_1)} \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] & x \le 0, \\ \frac{\kappa_1 \left(\boldsymbol{\vartheta}_{1\mathrm{E}2} \mathrm{e}^{-2\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}1}\right)}{w_+(\boldsymbol{\vartheta}_1)} \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] & x > 0; \end{cases} \tag{5.45}$$

$$\boldsymbol{M}^{(1)}_{1}(x,\zeta) = \left[\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right]; \tag{5.46}$$

where

$$w_\pm(\boldsymbol{\vartheta}_1) := \mathrm{W}[\boldsymbol{\vartheta}_1(x)] \mathrm{e}^{\mp\kappa_1 x} = \boldsymbol{\vartheta}_{1\mathrm{E}1} \mathrm{e}^{(1\mp1)\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2} \mathrm{e}^{-(1\pm1)\kappa_1 x}. \tag{5.47}$$

The derivation of these expressions can be found in Appendix 5.B.4.

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 (5.41) and (5.42). 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), $\boldsymbol{\vartheta}_{2\mathrm{E}1} := \left[\begin{smallmatrix} 1 & 0 \end{smallmatrix}\right] \boldsymbol{\vartheta}_{2\mathrm{E}}(x)$, and $\boldsymbol{\vartheta}_{2\mathrm{E}2} := \left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right] \boldsymbol{\vartheta}_{2\mathrm{E}}(x)$. We propose to calculate these matrices as follows.

$$\boldsymbol{M}^{(2)}_{-1}(x,\zeta) = \begin{cases} \frac{2\kappa_1\kappa_2(\kappa_2^2-\kappa_1^2)}{w^2(\boldsymbol{\vartheta}_1,\boldsymbol{\vartheta}_2)} \left[\begin{smallmatrix} m_-(x,0) & m_-(x,\zeta) \\ -m_-(x,-\zeta) & -m_-(x,0) \end{smallmatrix}\right] & x \le 0, \\ \frac{2\kappa_1\kappa_2(\kappa_2^2-\kappa_1^2)}{w_+^2(\boldsymbol{\vartheta}_1,\boldsymbol{\vartheta}_2)} \left[\begin{smallmatrix} m_+(x,0) & m_+(x,\zeta) \\ -m_+(x,-\zeta) & -m_+(x,0) \end{smallmatrix}\right] & x > 0; \end{cases} \tag{5.48}$$

$$\boldsymbol{M}_0^{(2)}(x,\zeta) = \begin{cases} \frac{2(\kappa_2^2-\kappa_1^2)}{w_-^2(\vartheta_1,\vartheta_2)} \begin{bmatrix} p_-(x) & s_-(x,\zeta) \\ s_-(x,-\zeta) & p_-(x) \end{bmatrix} - \frac{\kappa_2^2+\kappa_1^2}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x \le 0, \\ \frac{2(\kappa_2^2-\kappa_1^2)}{w_+^2(\vartheta_1,\vartheta_2)} \begin{bmatrix} p_+(x) & s_+(x,\zeta) \\ s_+(x,-\zeta) & p_+(x) \end{bmatrix} - \frac{\kappa_2^2+\kappa_1^2}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x > 0; \end{cases} \tag{5.49}$$

$$\boldsymbol{M}_1^{(2)}(x,\zeta) = \begin{cases} \frac{(\kappa_2^2-\kappa_1^2)(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{2\kappa_1 x}+\boldsymbol{\vartheta}_{1\mathrm{E}2})(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{2\kappa_2 x}+\boldsymbol{\vartheta}_{2\mathrm{E}2})}{w_-(\vartheta_1,\vartheta_2)} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} & x \le 0, \\ \frac{(\kappa_2^2-\kappa_1^2)(\boldsymbol{\vartheta}_{1\mathrm{E}1}+\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-2\kappa_1 x})(\boldsymbol{\vartheta}_{2\mathrm{E}1}+\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-2\kappa_2 x})}{w_+(\vartheta_1,\vartheta_2)} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} & x > 0; \end{cases} \tag{5.50}$$

$$\boldsymbol{M}_2^{(2)}(x,\zeta) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \tag{5.51}$$

where

$$w_\pm(\vartheta_1,\vartheta_2) := \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x}+\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x} & \kappa_1\left(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x}-\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x}\right) \\ \boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x}+\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x} & \kappa_2\left(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x}-\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x}\right) \end{vmatrix}, \tag{5.52}$$

$$m_\pm(x,\zeta) := \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}1}^2\mathrm{e}^{(2\mp2)\kappa_1 x}-\boldsymbol{\vartheta}_{1\mathrm{E}2}^2\mathrm{e}^{-(2\pm2)\kappa_1 x} & \kappa_1\boldsymbol{\vartheta}_{1\mathrm{E}1}\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_1)x} \\ \boldsymbol{\vartheta}_{2\mathrm{E}1}^2\mathrm{e}^{(2\mp2)\kappa_2 x}-\boldsymbol{\vartheta}_{2\mathrm{E}2}^2\mathrm{e}^{-(2\pm2)\kappa_2 x} & \kappa_2\boldsymbol{\vartheta}_{2\mathrm{E}1}\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_2)x} \end{vmatrix}, \tag{5.53}$$

$$p_\pm(x) := \tfrac{1}{4}(\kappa_2^2-\kappa_1^2)(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x}+\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x})^2 \times \cdots (\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x}+\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x})^2, \tag{5.54}$$

$$s_\pm(x,\zeta) := \begin{vmatrix} \kappa_1^2\boldsymbol{\vartheta}_{1\mathrm{E}1}\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_1)x} & \left(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_1 x}+\boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_1 x}\right)^2 \\ \kappa_2^2\boldsymbol{\vartheta}_{2\mathrm{E}1}\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{2(\mathrm{j}\zeta\mp\kappa_2)x} & \left(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{(1\mp1)\kappa_2 x}+\boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-(1\pm1)\kappa_2 x}\right)^2 \end{vmatrix}. \tag{5.55}$$

The derivations of these expressions can also be found in Appendix 5.B.4.

Referring back to Section 5.3.6, we emphasize that for numerical reasons the analytically equivalent expressions for the cases $x \le 0$ and $x > 0$ should not be merged.

5.4.2. 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 matrices $\boldsymbol{M}_m^{(\widetilde{N})}(x,\zeta)$ in the algorithm should be calculated with the equations in Section 5.4.1 after dropping the tildes from $\tilde{\kappa}_1$, $\tilde{\kappa}_2$, $\widetilde{\boldsymbol{\vartheta}}_{1\mathrm{E}}(x)$, $\widetilde{\boldsymbol{\vartheta}}_{2\mathrm{E}}(x)$ 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 (5.14), (5.15) and (5.18).

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 4 and 5 in Appendix 5.A. ∎

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 (5.22) for the operator $\mathbf{V}_{\mathrm{E}}$, cf. (5.96).

1: **inputs:** $q^{(0)}(x)$, N, $\{\kappa_n,\, b^{(N)}(\mathrm{j}\kappa_n),\, \boldsymbol{\phi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n),\, \boldsymbol{\psi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n)\}\ \forall n \in \mathbb{N}_1^N$
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 $\lim_{x\to-\infty}\boldsymbol{\phi}_{\mathrm{E}}^{(0)} = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\ \forall\kappa_n$
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 $\lim_{x\to\infty}\boldsymbol{\psi}_{\mathrm{E}}^{(0)} = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\ \forall\kappa_n$
5: **for all** x **do** {
6: $\quad q(x) \leftarrow q^{(0)}(x)$;
7: $\quad$ **for all** $n = 1$ **to** N **do** {
8: $\quad\quad \boldsymbol{\vartheta}_{n\mathrm{E}}(x) \leftarrow \boldsymbol{\phi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n) - (-1)^N\, b^{(N)}(\mathrm{j}\kappa_n)\, \boldsymbol{\psi}_{\mathrm{E}}^{(0)}(x,\mathrm{j}\kappa_n)$;
9: $\quad$ }
10: $\quad$ **while** $N > 0$ **do** {
11: $\quad\quad \widetilde{N} \leftarrow 1$;
12: $\quad\quad \tilde{\kappa}_1 \leftarrow \kappa_N$;
13: $\quad\quad \widetilde{\boldsymbol{\vartheta}}_{1\mathrm{E}}(x) \leftarrow 2^{-\left\lfloor \frac{1}{2}\mathrm{lb}\lvert[\,1\ 0\,]\,\boldsymbol{\vartheta}_{N\mathrm{E}}(x)\rvert + \frac{1}{2}\mathrm{lb}\lvert[\,0\ 1\,]\,\boldsymbol{\vartheta}_{N\mathrm{E}}(x)\rvert \right\rfloor}\,\boldsymbol{\vartheta}_{N\mathrm{E}}(x)$;
14: $\quad\quad$ **if** N is even **then** {
15: $\quad\quad\quad \widetilde{N} \leftarrow 2$;
16: $\quad\quad\quad \tilde{\kappa}_2 \leftarrow \kappa_{(N-1)}$;
17: $\quad\quad\quad \widetilde{\boldsymbol{\vartheta}}_{2\mathrm{E}}(x) \leftarrow 2^{-\left\lfloor \frac{1}{2}\,\mathrm{lb}\lvert[\,1\ 0\,]\,\boldsymbol{\vartheta}_{(N-1)\mathrm{E}}(x)\rvert + \frac{1}{2}\,\mathrm{lb}\lvert[\,0\ 1\,]\,\boldsymbol{\vartheta}_{(N-1)\mathrm{E}}(x)\rvert \right\rfloor}\,\boldsymbol{\vartheta}_{(N-1)\mathrm{E}}(x)$;
18: $\quad\quad$ }
19: $\quad\quad q(x) \leftarrow (-1)^{\widetilde{N}}\, q(x) + [\,4\ 0\,]\,\boldsymbol{M}_{\widetilde{N}-2}^{(\widetilde{N})}(x,0)\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$;
20: $\quad\quad$ **if** $q^{(0)}(x) \equiv 0$ **then** {
21: $\quad\quad\quad q(x) \leftarrow \max(q(x),0)$;
22: $\quad\quad$ }
23: $\quad\quad N \leftarrow N - \widetilde{N}$;
24: $\quad\quad$ **for** $n = 1$ **to** N **do**
25: $\quad\quad\quad \boldsymbol{\vartheta}_{n\mathrm{E}}(x) \leftarrow \sum_{m=-1}^{\widetilde{N}} (-\kappa_n)^m\, \boldsymbol{M}_m^{(\widetilde{N})}(x,\mathrm{j}\kappa_n)\, \boldsymbol{\vartheta}_{n\mathrm{E}}(x)$;
26: $\quad\quad$ }
27: $\quad$ }
28: $\quad q^{(N)}(x) \leftarrow q(x)$;
29: }
30: **output:** $q^{(N)}(x)$

The conventional method of implementing the Crum transform as a chain of N Darboux transforms does not have this property. (See Section 5.3.2.)

The computational complexity of the proposed algorithm is $O(N^2)$. To see this, note that the mapping of the remaining seed trajectories to the next potential in Line 25 of Algorithm 1 dominates the complexity for large N. This mapping is executed $(2\lceil N/2\rceil - 2) + (2\lceil N/2\rceil - 4) + \ldots + 4 + 2 + 0 = \lceil N/2\rceil^2 - \lceil N/2\rceil = O(N^2)$ times.

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

5.5. 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 (5.29) and (5.30), where each step adds the lowest of the remaining target eigenvalues. We will refer to this $O(N^2)$ algorithm as Darboux C (——). The second benchmark algorithm is the one-shot Crum transform presented in [99], with a complexity of $O(N^3)$. 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 [111, Sect. IV/App. F]. The complexity of Neugebauer modified is still $O(N^3)$.

5.5.1. 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

$$E_0 := \left\| \widehat{q^{(N)}} - q^{(N)} \right\|_2 \;\Big/\; \left\| q^{(N)} \right\|_2 , \tag{5.56}$$

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

$$E_p := \left| \frac{\int_{-\infty}^{\infty} (\widehat{q^{(N)}}(x))^p - (q^{(0)}(x))^p \,\mathrm{d}x}{\int_{-\infty}^{\infty} (q^{(N)}(x))^p - (q^{(0)}(x))^p \,\mathrm{d}x} - 1 \right|, \qquad p \in \{1, 2\}, \tag{5.57}$$

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 KdV to evaluate the denominator in (5.57) directly from the spectrum. The first two of these conserved quantities are [5, Sect.

[12] The analytic sensitivity of the potential to perturbations of the spectrum is discussed elsewhere [71].

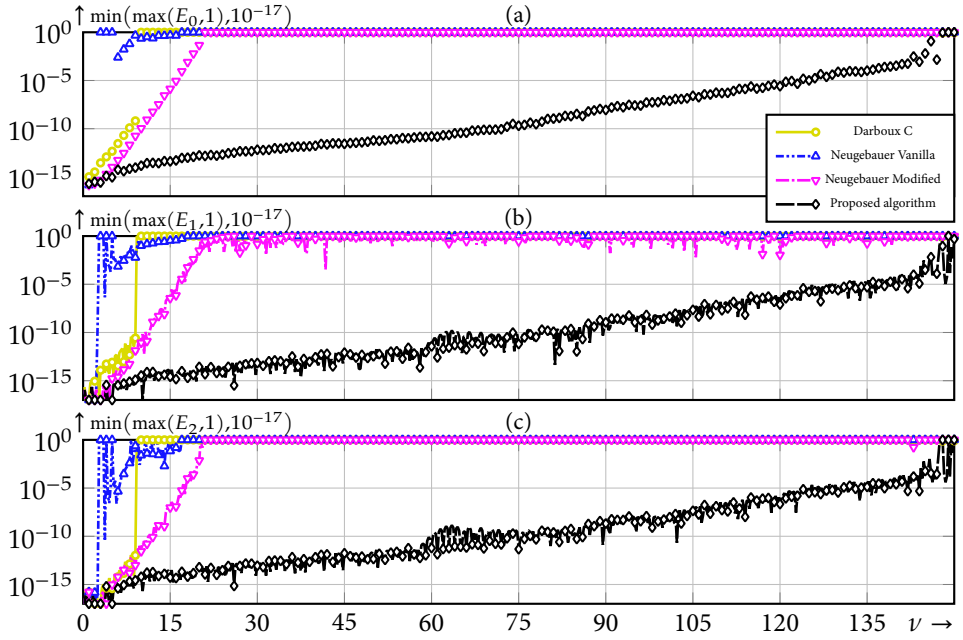


Figure 5.3: Results example 'The discrete spectrum of a squared secant hyperbolic potential' (Section 5.5.2). The target potentials, consisting of $N = \lceil \nu \rceil$ eigenvalues, were computed with three benchmark algorithms and the proposed one. The errors E_1 and E_2 were calculated with (5.60) for $\nu \in \{\frac{1}{4}, \frac{2}{4}, \dots, 150\}$ and are shown as lines (——/-·-·-/-·-·-/——). Furthermore E_0, E_1, and E_2 were calculated with (5.56) and (5.57) for $\nu \in \mathbb{N}_1^{150}$ and are shown as markers (○/△/▽/◇).

1.6], [164]

$$\int_{-\infty}^{\infty} q(x)\,\mathrm{d}x = \tfrac{1}{\pi}\int_{-\infty}^{\infty} \ln\bigl(1-|R(\zeta)|^2\bigr)\,\mathrm{d}\zeta + 4\sum_{m=1}^{M}\gamma_m; \tag{5.58}$$

$$\int_{-\infty}^{\infty} \bigl(q(x)\bigr)^2\,\mathrm{d}x = -\tfrac{4}{\pi}\int_{-\infty}^{\infty} \zeta^2 \ln\bigl(1-|R(\zeta)|^2\bigr)\,\mathrm{d}\zeta + \tfrac{16}{3}\sum_{m=1}^{M}\gamma_m^3. \tag{5.59}$$

Since the eigenvalues of the background potential are invariant under the Crum transform, as is $|R(\zeta)|$ for ζ on the real line by (5.19), it follows from (5.57) to (5.59) that

$$E_p = \left| \frac{(2p-1)\int_{-\infty}^{\infty} \bigl(\widehat{q^{(N)}}(x)\bigr)^p - \bigl(q^{(0)}(x)\bigr)^p\,\mathrm{d}x}{4^p \sum_{n=1}^{N} \kappa_n^{2p-1}} - 1 \right|, \qquad p \in \{1,2\}. \tag{5.60}$$

This formula enables us to evaluate E_1 and E_2 even if the true target potential is unknown. In Section 5.5.2, we will demonstrate that E_1 and E_2 are highly correlated with E_0.

5.5.2. 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 $(\kappa_n, b^{(\lceil \nu \rceil)}(\mathrm{j}\kappa_n)) = (\nu - \lceil \nu \rceil + n, (-1)^{\lceil \nu \rceil - n})$, where $n \in \mathbb{N}_1^{\lceil \nu \rceil}$ and ν is varied from $\frac{1}{4}$ to

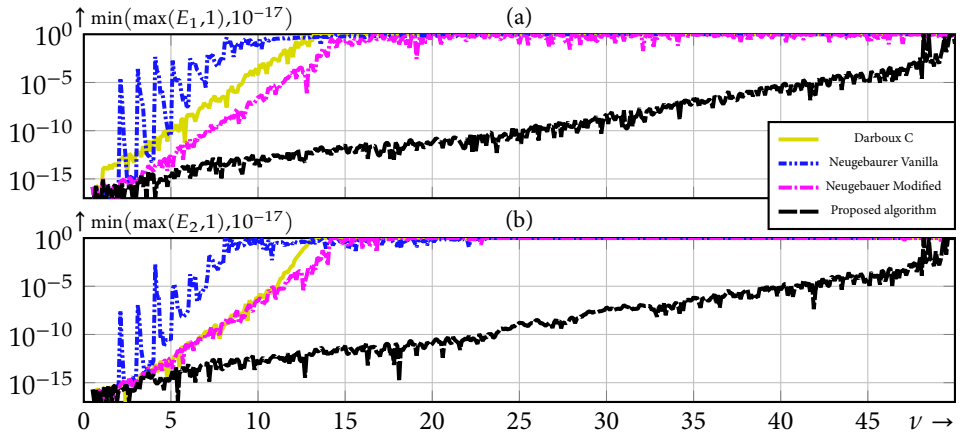


Figure 5.4: Results example 'The discrete spectrum of a rectangular potential' (Section 5.5.3). The target potentials, consisting of $N = \lceil \nu \rceil$ eigenvalues, were computed with three benchmark algorithms (—/-·-·-/-··-··-) and the proposed one (——). The errors E_1 and E_2 were calculated with (5.60) for $\nu \in \{0.5, 0.6, \ldots, 50\}$.

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 (5.60) 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 $q^{(\nu)}(x) = \nu(\nu+1)\,\mathrm{sech}^2(x)$ [79, Sect. 2.5]. In those cases, we additionally evaluate E_0 with (5.56) and evaluate E_1 and E_2 also using (5.57), again making use of trapezoidal integration. The results are shown in Fig. 5.3. 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 (5.60) is indeed equivalent to the error calculation according to (5.57), and that E_1 and E_2 are highly correlated with E_0.

5.5.3. 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 $q_{\mathrm{rect}}(x) := \left(\frac{\pi\nu}{\ell}\right)^2$ 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 [101, 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

$$\frac{\sqrt{(\pi\nu)^2 - (\ell\kappa_n)^2}}{\ell\kappa_n} - \frac{\ell\kappa_n}{\sqrt{(\pi\nu)^2 - (\ell\kappa_n)^2}} = 2\cot\left(\sqrt{(\pi\nu)^2 - (\ell\kappa_n)^2}\right) \tag{5.61}$$

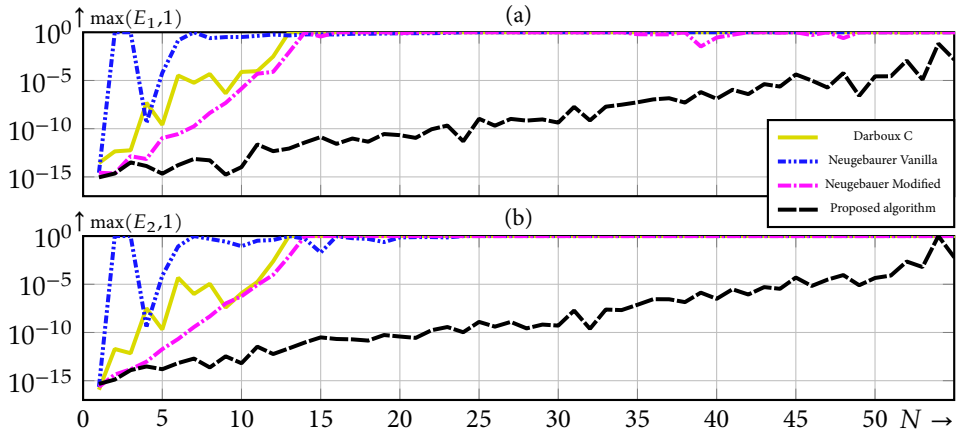


Figure 5.5: Results of example 'Non-zero background potential' (Section 5.5.4). The target potentials, with $2+N$ eigenvalues, were computed with three benchmark algorithms (——/·-·-·/-·-·-) and the proposed one (— —). The errors E_1 and E_2 were calculated with (5.60) for $N \in \mathbb{N}_1^{55}$.

for which

$$\pi\sqrt{\max(\,0\,,\,\nu^2-(N-n+1)^2\,)} < \ell\kappa_n < \pi\sqrt{\nu^2-(N-n)^2}.$$

Since $q_{\text{rect}}(x)$ 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 [111, Cor. 1] and (5.32). We vary ν from 0.5 to 50 in steps of 0.1 and evaluate the errors E_1 and E_2 according to (5.60)[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. 5.4. 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.

5.5.4. 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., [17, 152].) 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 (5.18) and (5.32)). Instead, we chose to add $2\lfloor N/2\rfloor$ eigenvalues

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

in this interval. If N is odd, the remaining target eigenvalue must be larger than $\mathrm{j}\gamma_2$. We achieve this by choosing

$$\left\{\kappa_n \middle| n \in \mathbb{N}_1^N\right\} = \left\{\tfrac{K_2+K_1}{2} + \tfrac{K_2-K_1}{N}\left(p+\tfrac{1}{2}\right) \middle| p \in \mathbb{Z}\right\} \cap (K_1, K_2], \tag{5.62}$$

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 n-th 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 (5.32).

The trapezoidal approximation of the integral in the error criterion (5.60) converged only very slowly in this example. Therefore, we instead approximated the integral in (5.60) with Clanshaw–Curtis quadrature [138, 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. 5.5. 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 5.3.2.)

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}.

5.6. Conclusion

The Crum transform plays an important role in many inverse Non-linear Fourier Transform (NFT) methods. In this chapter, we have proposed a new algorithm for the numerical computation of the Crum transform for the Korteweg–de Vries equation (KdV). The numerical accuracy of the new algorithm is much better than that of previous Crum transform 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.

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

5.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 4 *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,

$$(5.32) \Rightarrow \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_m^{(N)})) = (-1)^{M+N-m} \quad \forall m \in \mathbb{N}_1^{M+N}, \tag{5.63}$$

$$(5.32) \Rightarrow \operatorname{sign}(b^{(0)}(\mathrm{j}\gamma_m^{(0)})) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^{M}, \tag{5.64}$$

$$(5.18) \text{ and } (5.64) \Rightarrow \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_m^{(0)})) = (-1)^{M+N-m} \quad \forall m \in \mathbb{N}_1^{M}, \tag{5.65}$$

$$(5.63) \text{ and } (5.65) \Rightarrow \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_{M+N}^{(N)})) \neq \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_{M}^{(0)})), \tag{5.66}$$

$$(5.66) \Rightarrow \gamma_{M+N}^{(N)} \neq \gamma_M^{(0)} \Rightarrow \gamma_{M+N}^{(N)} = \kappa_N > \max_m \gamma_m^{(0)}. \tag{5.67}$$

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

$$(5.67) \Rightarrow \gamma_m^{(1)} = \begin{cases} \kappa_N & m = M+1, \\ \gamma_m^{(0)} & m \in \mathbb{N}_1^M \end{cases} \tag{5.68}$$

$$(5.68) \Rightarrow \operatorname{sign}(b^{(1)}(\mathrm{j}\gamma_m^{(1)})) = \begin{cases} \operatorname{sign}(b^{(1)}(\mathrm{j}\kappa_N)) & m = M+1, \\ \operatorname{sign}(b^{(1)}(\mathrm{j}\gamma_m^{(0)})) & m \in \mathbb{N}_1^M \end{cases} \tag{5.69}$$

$$(5.18) \text{ and } (5.67) \Rightarrow = \begin{cases} \operatorname{sign}(b^{(1)}(\mathrm{j}\gamma_{M+N}^{(N)})) & m = M+1, \\ -\operatorname{sign}(b^{(0)}(\mathrm{j}\gamma_m^{(0)})) & m \in \mathbb{N}_1^M \end{cases} \tag{5.70}$$

$$(5.18) \text{ and } (5.64) \Rightarrow = \begin{cases} \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_{M+N}^{(N)})) & m = M+1, \\ -(-1)^{M-m} & m \in \mathbb{N}_1^M \end{cases} \tag{5.71}$$

$$(5.63) \Rightarrow = (-1)^{(M+1)-m} \quad \forall m \in \mathbb{N}_1^{M+1}. \tag{5.72}$$

From (5.32) and (5.72) it follows that $q^{(1)}(x)$ is absolutely integrable. If the discrete spectrum of the background potential is empty, the proof reduces to (5.68) to (5.72), where $\mathbb{N}_1^M = \mathbb{N}_1^0$ is an empty set. ∎

Lemma 5 *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,

$$(5.32), N \text{ even} \Rightarrow \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_m^{(N)})) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^{M+N}, \tag{5.73}$$

$$(5.32) \Rightarrow \operatorname{sign}(b^{(0)}(\mathrm{j}\gamma_m^{(0)})) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^M, \tag{5.74}$$

$$(5.18), (5.74) \Rightarrow \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_m^{(0)})) = (-1)^{M-m} \quad \forall m \in \mathbb{N}_1^M, \tag{5.75}$$

$$(5.73), (5.75) \Rightarrow \begin{cases} |\{\kappa_n \mid \gamma_m^{(0)} < \kappa_n < \gamma_{m+1}^{(0)}\}| \mod 2 = 0 & \forall m \in \mathbb{N}_1^{M-1}, \\ |\{\kappa_n \mid \kappa_n > \gamma_M^{(0)}\}| \mod 2 = 0. \end{cases} \tag{5.76}$$

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

$$(5.76) \Rightarrow \gamma_m^{(N)} = \begin{cases} \gamma_{m-N}^{(0)} & m \in \mathbb{N}_{M+N-\mu+1}^{M+N}, \\ \kappa_{m+\mu-M} & m \in \mathbb{N}_{M+N-\mu-1}^{M+N-\mu}, \\ \star & m \in \mathbb{N}_1^{M+N-\mu-2}, \end{cases} \tag{5.77}$$

where $\star$ is not of interest and where $\mu := |\{\gamma_m^{(0)} \mid \gamma_m^{(0)} > \kappa_N\}| \in \mathbb{N}_1^M$.

$$(5.73), (5.77) \Rightarrow \operatorname{sign}(b^{(N)}(\mathrm{j}\kappa_{m+\mu-M})) = \operatorname{sign}(b^{(N)}(\mathrm{j}\gamma_m^{(N)})) = (-1)^{M-m}, \quad \text{for } m \in \mathbb{N}_{M+N-\mu-1}^{M+N-\mu}; \tag{5.78}$$

$$(5.76) \Rightarrow \gamma_m^{(2)} = \begin{cases} \gamma_{m-2}^{(0)} & m \in \mathbb{N}_{M-\mu+3}^{M+2}, \\ \kappa_{m+\mu-M} & m \in \mathbb{N}_{M-\mu+1}^{M-\mu+2}, \\ \gamma_m^{(0)} & m \in \mathbb{N}_1^{M-\mu}; \end{cases} \tag{5.79}$$

$$(5.79) \Rightarrow \operatorname{sign}(b^{(2)}(\mathrm{j}\gamma_m^{(2)})) = \begin{cases} \operatorname{sign}(b^{(2)}(\gamma_{m-2}^{(0)})) & m \in \mathbb{N}_{M-\mu+3}^{M+2}, \\ \operatorname{sign}(b^{(2)}(\kappa_{m+\mu-M})) & m \in \mathbb{N}_{M-\mu+1}^{M-\mu+2}, \\ \operatorname{sign}(b^{(2)}(\gamma_m^{(0)})) & m \in \mathbb{N}_1^{M-\mu}; \end{cases} \tag{5.80}$$

$$(5.18), N \text{ even} \Rightarrow = \begin{cases} \operatorname{sign}(b^{(0)}(\gamma_{m-2}^{(0)})) & m \in \mathbb{N}_{M-\mu+3}^{M+2}, \\ \operatorname{sign}(b^{(N)}(\kappa_{m+\mu-M})) & m \in \mathbb{N}_{M-\mu+1}^{M-\mu+2}, \\ \operatorname{sign}(b^{(0)}(\gamma_m^{(0)})) & m \in \mathbb{N}_1^{M-\mu}; \end{cases} \tag{5.81}$$

$$(5.74), (5.78) \Rightarrow = (-1)^{(M+2)-m} \quad \forall m \in \mathbb{N}_1^{M+2}. \tag{5.82}$$

From (5.32) and (5.82) it follows that $q^{(2)}(x)$ is absolutely integrable. If the discrete spectrum of the background potential is empty, the proof reduces to (5.77) to (5.82), 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. ∎

5.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 5.B.1 we show how to translate between different representations of the Schrödinger eigenvalue problem. In Appendix 5.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 5.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 5.B.4, we translate the results from Appendix 5.B.3 using those from Appendix 5.B.1 to the numerically advantageous representation in which the dressing method is used in the proposed algorithm. (See Section 5.4.)

5.B.1. 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 equation (5.2) as a system of first order equations (5.22) and also for the formulation of the dressing method (5.27). 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 [111]. Hence we consider

$$\mathbf{V}_{\mathrm{C}}(x,\zeta) := \begin{bmatrix} 1 \\ \frac{\partial}{\partial x} \end{bmatrix}, \tag{5.83}$$

$$\mathbf{V}_{\mathrm{S}}(x,\zeta) := \frac{1}{2\mathrm{j}\zeta} \begin{bmatrix} \mathrm{j}\zeta - \frac{\partial}{\partial x} \\ \mathrm{j}\zeta + \frac{\partial}{\partial x} \end{bmatrix}, \tag{5.84}$$

$$\mathbf{V}_{\mathrm{E}}(x,\zeta) := \frac{1}{2\mathrm{j}\zeta} \begin{bmatrix} \mathrm{e}^{\mathrm{j}\zeta x}(\mathrm{j}\zeta - \frac{\partial}{\partial x}) \\ \mathrm{e}^{-\mathrm{j}\zeta x}(\mathrm{j}\zeta + \frac{\partial}{\partial x}) \end{bmatrix}. \tag{5.85}$$

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 [111, 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 [111, eqs. (70)–(75)] to

$$\mathbf{V}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\mathbf{V}_{\mathfrak{a}}(x,\zeta), \tag{5.86}$$

$$f_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,f_{\mathfrak{a}}(x,\zeta), \tag{5.87}$$

$$\boldsymbol{\Phi}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{\Phi}_{\mathfrak{a}}(x,\zeta), \tag{5.88}$$

$$\boldsymbol{A}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{A}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta) + \left(\tfrac{\partial}{\partial x}\,\boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right)\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta) \tag{5.89}$$

(see (5.98)),

$$\boldsymbol{H}_{\mathfrak{b}}(x_1,x_2,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x_2,\zeta)\,\boldsymbol{H}_{\mathfrak{a}}(x_1,x_2,\zeta)\,\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x_1,\zeta), \tag{5.90}$$

$$\text{where } \boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta) = \left(\boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right)^{-1}, \tag{5.91}$$

and for the dressing method we add

$$\boldsymbol{C}_{\mathfrak{b}}(x,\zeta) = \boldsymbol{T}_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\,\boldsymbol{C}_{\mathfrak{a}}(x,\zeta)\,\boldsymbol{T}_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta). \tag{5.92}$$

From (5.83) to (5.85) one can find

$$\boldsymbol{T}_{\mathrm{S}}^{\mathrm{E}}(x,\zeta) = \begin{bmatrix} \mathrm{e}^{\mathrm{j}\zeta x} & 0 \\ 0 & \mathrm{e}^{-\mathrm{j}\zeta x} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{T}_{\mathrm{E}}^{\mathrm{S}}(x,\zeta) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}\zeta x} & 0 \\ 0 & \mathrm{e}^{\mathrm{j}\zeta x} \end{bmatrix}; \tag{5.93}$$

$$T_{\mathrm{C}}^{\mathrm{E}}(x,\zeta) = \frac{1}{2\mathrm{j}\zeta}\begin{bmatrix} \mathrm{j}\zeta \mathrm{e}^{\mathrm{j}\zeta x} & -\mathrm{e}^{\mathrm{j}\zeta x} \\ \mathrm{j}\zeta \mathrm{e}^{-\mathrm{j}\zeta x} & \mathrm{e}^{-\mathrm{j}\zeta x} \end{bmatrix} \quad \Rightarrow \quad T_{\mathrm{E}}^{\mathrm{C}}(x,\zeta) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}\zeta x} & \mathrm{e}^{\mathrm{j}\zeta x} \\ -\mathrm{j}\zeta \mathrm{e}^{-\mathrm{j}\zeta x} & \mathrm{j}\zeta \mathrm{e}^{\mathrm{j}\zeta x} \end{bmatrix}. \tag{5.94}$$

Using (5.89), (5.93), and

$$A_{\mathrm{S}}(x,\zeta) = \frac{q(x)}{2\mathrm{j}\zeta}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} - \begin{bmatrix} \mathrm{j}\zeta & 0 \\ 0 & -\mathrm{j}\zeta \end{bmatrix} \qquad [111, \text{eq. (20)}] \tag{5.95}$$

$$\text{we find } A_{\mathrm{E}}(x,\zeta) = \frac{q(x)}{2\mathrm{j}\zeta}\begin{bmatrix} 1 & \exp(2\mathrm{j}\zeta x) \\ -\exp(-2\mathrm{j}\zeta x) & -1 \end{bmatrix}, \tag{5.96}$$

which we use in Algorithm 1, Lines 3 and 4. Finally, from (5.89), (5.94) and (5.96) we find

$$A_{\mathrm{C}}(x,\zeta) = \begin{bmatrix} 0 & 1 \\ (\mathrm{j}\zeta)^2 - q(x) & 0 \end{bmatrix}, \tag{5.97}$$

which is, like it should be, the same result as what can be found directly from (5.2), (5.22) and (5.83).

The derivation of (5.89) is as follows.

$$\begin{aligned} \frac{\partial}{\partial x} f_{\mathfrak{b}}(x,\zeta) &= \frac{\partial}{\partial x}\left(T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta) f_{\mathfrak{a}}(x,\zeta)\right) \\ &= T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\left(\tfrac{\partial}{\partial x} f_{\mathfrak{a}}(x,\zeta)\right) + \left(\tfrac{\partial}{\partial x} T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right) f_{\mathfrak{a}}(x,\zeta) \\ &= \left(T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta) A_{\mathfrak{a}}(x,\zeta) + \left(\tfrac{\partial}{\partial x} T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right)\right) f_{\mathfrak{a}}(x,\zeta) \\ &= \underbrace{\left(T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta) A_{\mathfrak{a}}(x,\zeta) T_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta) + \left(\tfrac{\partial}{\partial x} T_{\mathfrak{a}}^{\mathfrak{b}}(x,\zeta)\right) T_{\mathfrak{b}}^{\mathfrak{a}}(x,\zeta)\right)}_{=A_{\mathfrak{b}}(x,\zeta)} f_{\mathfrak{b}}(x,\zeta). \end{aligned} \tag{5.98}$$

5.B.2. Wronskians of trajectories of the Schrödinger equation and their derivatives

In Appendix 5.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 equation (5.2) *for the same potential* $q = q(x)$:

$$(\tfrac{\partial^2}{\partial x^2} + q) f_n = \lambda_n f_n, \tag{5.99}$$

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,

$$f_n = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{f}_{n\mathrm{C}}, \tag{5.100}$$

$$\frac{\partial f_n}{\partial x} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{f}_{n\mathrm{C}}, \tag{5.101}$$

$$\frac{\partial^2 f_n}{\partial x^2} = (\lambda_n - q) f_n = \begin{bmatrix} \lambda_n - q & 0 \end{bmatrix} \boldsymbol{f}_{n\mathrm{C}}, \tag{5.102}$$

$$\frac{\partial^3 f_n}{\partial x^3} = -\frac{\mathrm{d}q}{\mathrm{d}x} f_n + (\lambda_n - q)\frac{\partial f_n}{\partial x} = \begin{bmatrix} -\frac{\mathrm{d}q}{\mathrm{d}x} & \lambda_n - q \end{bmatrix} \boldsymbol{f}_{n\mathrm{C}}, \tag{5.103}$$

$$\mathrm{W}[f_1] = f_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{f}_{1\mathrm{C}}, \tag{5.104}$$

$$\frac{\partial}{\partial x}\mathrm{W}[f_1] = \frac{\partial f_1}{\partial x} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{f}_{1\mathrm{C}}, \tag{5.105}$$

$$\frac{\partial^2}{\partial x^2}\,\mathrm{W}[f_1] = \frac{\partial^2 f_1}{\partial x^2} = \begin{bmatrix}(\lambda_1 - q) & 0\end{bmatrix} \boldsymbol{f}_{1\mathrm{C}}, \tag{5.106}$$

$$\mathrm{W}[f_1;f_2] = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} \\ f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} = \begin{bmatrix} -\frac{\partial f_1}{\partial x} & f_1\end{bmatrix} \boldsymbol{f}_{2\mathrm{C}}, \tag{5.107}$$

$$\begin{aligned}\frac{\partial}{\partial x}\,\mathrm{W}[f_1;f_2] &= \begin{vmatrix} f_1 & \frac{\partial^2 f_1}{\partial x^2} \\ f_2 & \frac{\partial^2 f_2}{\partial x^2}\end{vmatrix} = \begin{vmatrix} f_1 & (\lambda_1 - q) f_1 \\ f_2 & (\lambda_2 - q) f_2\end{vmatrix} = \begin{vmatrix} f_1 & \lambda_1 f_1 \\ f_2 & \lambda_2 f_2\end{vmatrix} \\ &= \begin{bmatrix}(\lambda_2 - \lambda_1) f_1 & 0\end{bmatrix} \boldsymbol{f}_{2\mathrm{C}}, \end{aligned} \tag{5.108}$$

$$\begin{aligned}\frac{\partial^2}{\partial x^2}\,\mathrm{W}[f_1;f_2] &= \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\,\mathrm{W}[f_1;f_2]\right) = \frac{\partial}{\partial x}\left(\begin{bmatrix}(\lambda_2 - \lambda_1) f_1 & 0\end{bmatrix} \boldsymbol{f}_{2\mathrm{C}}\right) \\ &= \begin{bmatrix}(\lambda_2 - \lambda_1)\frac{\partial f_1}{\partial x} & 0\end{bmatrix} \boldsymbol{f}_{2\mathrm{C}} + \begin{bmatrix}(\lambda_2 - \lambda_1) f_1 & 0\end{bmatrix} \begin{bmatrix} 0 & 1 \\ \lambda_2 - q & 0\end{bmatrix} \boldsymbol{f}_{2\mathrm{C}} \\ &= (\lambda_2 - \lambda_1) \begin{bmatrix}\frac{\partial f_1}{\partial x} & f_1\end{bmatrix} \boldsymbol{f}_{2\mathrm{C}}, \end{aligned} \tag{5.109}$$

$$\begin{aligned}\mathrm{W}[f_1;f_2;f_3] &= \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \frac{\partial^2 f_1}{\partial^2 x} \\ f_2 & \frac{\partial f_2}{\partial x} & \frac{\partial^2 f_2}{\partial^2 x} \\ f_3 & \frac{\partial f_3}{\partial x} & \frac{\partial^2 f_3}{\partial^2 x}\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & (\lambda_1 - q) f_1 \\ f_2 & \frac{\partial f_2}{\partial x} & (\lambda_2 - q) f_2 \\ f_3 & \frac{\partial f_3}{\partial x} & (\lambda_3 - q) f_3\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \lambda_1 f_1 \\ f_2 & \frac{\partial f_2}{\partial x} & \lambda_2 f_2 \\ f_3 & \frac{\partial f_3}{\partial x} & \lambda_3 f_3\end{vmatrix} \\ &= \left[\begin{vmatrix}\frac{\partial f_1}{\partial x} & \lambda_1 f_1 \\ \frac{\partial f_2}{\partial x} & \lambda_2 f_2\end{vmatrix} + \lambda_3 \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} \\ f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} \quad - \begin{vmatrix} f_1 & \lambda_1 f_1 \\ f_2 & \lambda_2 f_2\end{vmatrix}\right] \boldsymbol{f}_{3\mathrm{C}} \\ &= \left[\begin{vmatrix}(\lambda_3 - \lambda_1) f_1 & \frac{\partial f_1}{\partial x} \\ (\lambda_3 - \lambda_2) f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} \quad (\lambda_1 - \lambda_2) f_1 f_2\right] \boldsymbol{f}_{3\mathrm{C}}, \end{aligned} \tag{5.110}$$

$$\begin{aligned}\frac{\partial}{\partial x}\,\mathrm{W}[f_1;f_2;f_3] &= \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \frac{\partial^3 f_1}{\partial^3 x} \\ f_2 & \frac{\partial f_2}{\partial x} & \frac{\partial^3 f_2}{\partial^3 x} \\ f_3 & \frac{\partial f_3}{\partial x} & \frac{\partial^3 f_3}{\partial^3 x}\end{vmatrix} = \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & -\frac{\mathrm{d}q}{\mathrm{d}x} f_1 + (\lambda_1 - q)\frac{\partial f_1}{\partial x} \\ f_2 & \frac{\partial f_2}{\partial x} & -\frac{\mathrm{d}q}{\mathrm{d}x} f_2 + (\lambda_2 - q)\frac{\partial f_2}{\partial x} \\ f_3 & \frac{\partial f_3}{\partial x} & -\frac{\mathrm{d}q}{\mathrm{d}x} f_3 + (\lambda_3 - q)\frac{\partial f_3}{\partial x}\end{vmatrix} \\ &= \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} & \lambda_1 \frac{\partial f_1}{\partial x} \\ f_2 & \frac{\partial f_2}{\partial x} & \lambda_2 \frac{\partial f_2}{\partial x} \\ f_3 & \frac{\partial f_3}{\partial x} & \lambda_3 \frac{\partial f_3}{\partial x}\end{vmatrix} \\ &= \left[\begin{vmatrix}\frac{\partial f_1}{\partial x} & \lambda_1 \frac{\partial f_1}{\partial x} \\ \frac{\partial f_2}{\partial x} & \lambda_2 \frac{\partial f_2}{\partial x}\end{vmatrix} \quad \lambda_3 \begin{vmatrix} f_1 & \frac{\partial f_1}{\partial x} \\ f_2 & \frac{\partial f_2}{\partial x}\end{vmatrix} - \begin{vmatrix} f_1 & \lambda_1 \frac{\partial f_1}{\partial x} \\ f_2 & \lambda_2 \frac{\partial f_2}{\partial x}\end{vmatrix}\right] \boldsymbol{f}_{3\mathrm{C}} \\ &= \left[(\lambda_2 - \lambda_1)\frac{\partial f_1}{\partial x}\frac{\partial f_2}{\partial x} \quad \begin{vmatrix} f_1 & (\lambda_3 - \lambda_1)\frac{\partial f_1}{\partial x} \\ f_2 & (\lambda_3 - \lambda_2)\frac{\partial f_2}{\partial x}\end{vmatrix}\right] \boldsymbol{f}_{3\mathrm{C}}. \end{aligned} \tag{5.111}$$

5.B.3. Dressing method in C basis

In this section we make use of the results of Appendix 5.B.2 to derive the dressing method in C basis for the 1- and 2-fold Crum transform.

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

Starting from Theorem 1, making use of the expressions in Appendix 5.B.2 we derive the dressing method for the Darboux transform in C basis, (5.27), (5.29) and (5.30), as follows.

$$f^{(1)}(x,\zeta) = \frac{\mathrm{W}[\vartheta_1(x); f^{(0)}(x,\zeta)]}{\mathrm{W}[\vartheta_1(x)]} = \frac{\left[-\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x) \quad \vartheta_1(x)\right] \boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\vartheta_1(x)}, \tag{5.112}$$

$$\begin{aligned}
\tfrac{\partial}{\partial x} f^{(1)}(x,\zeta) &= \frac{\partial}{\partial x}\frac{\mathrm{W}[\vartheta_1(x); f^{(0)}(x,\zeta)]}{\mathrm{W}[\vartheta_1(x)]} \qquad (5.113)\\
&= \frac{\mathrm{W}[\vartheta_1(x)]\frac{\partial \mathrm{W}[\vartheta_1(x); f^{(0)}(x,\zeta)]}{\partial x} - \frac{\mathrm{d}\,\mathrm{W}[\vartheta_1(x)]}{\mathrm{d}x}\mathrm{W}[\vartheta_1(x), f^{(0)}(x,\zeta)]}{\left(\mathrm{W}[\vartheta_1(x)]\right)^2}\\
&= \frac{\left(\vartheta_1(x)\left[\begin{smallmatrix}(\mathrm{j}\zeta)^2-\kappa_1^2)\,\vartheta_1(x) & 0\end{smallmatrix}\right] - \left(\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\right)\left[\begin{smallmatrix}-\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x) & \vartheta_1(x)\end{smallmatrix}\right]\right)\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\left(\vartheta_1(x)\right)^2}\\
&= \left[(\mathrm{j}\zeta)^2 - \kappa_1^2 + \left(\tfrac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \big/ \vartheta_1(x)\right)^2 \quad -\left(\tfrac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \big/ \vartheta_1(x)\right)\right]\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta).
\end{aligned}$$

Equations (5.112) and (5.113) imply the trajectory mapping in (5.27) with the dressing matrix in (5.30). Again from Theorem 1 we obtain the potential update:

$$\begin{aligned}
\Delta q_{(0)}^{(1)}(x) &= q^{(1)}(x) - q^{(0)}(x) = 2\tfrac{\mathrm{d}^2}{\mathrm{d}x^2}\ln(\mathrm{W}[\vartheta_1(x)]) = 2\tfrac{\mathrm{d}^2}{\mathrm{d}x^2}\ln(\vartheta_1(x))\\
&= 2\frac{\mathrm{d}}{\mathrm{d}x}\frac{\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)}{\vartheta_1(x)} = 2\frac{\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\vartheta_1(x)\right)\vartheta_1(x) - \left(\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\right)^2}{\left(\vartheta_1(x)\right)^2}\\
&= 2\frac{(\kappa_1^2 - q^{(0)}(x))\left(\vartheta_1(x)\right)^2 - \left(\frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\right)^2}{\left(\vartheta_1(x)\right)^2}\\
&= -2\,q^{(0)}(x) + 2\kappa_1^2 - 2\left(\tfrac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \big/ \vartheta_1(x)\right)^2. \qquad (5.114)
\end{aligned}$$

This potential update is equivalent to (5.29).

Dressing method in C basis for $N = 2$

Starting from Theorem 1, making use of the expressions in Appendix 5.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 5.B.4 of the dressing method for the 2-fold Crum transform in E basis, which is part of the proposed algorithm.

$$\begin{aligned}
f^{(2)}(x,\zeta) &= \frac{\mathrm{W}[\vartheta_1(x); \vartheta_2(x); f^{(0)}(x,\zeta)]}{\mathrm{W}[\vartheta_1(x); \vartheta_2(x)]} \qquad (5.115)\\
&= \frac{\left[\left|\begin{smallmatrix}((\mathrm{j}\zeta)^2-\kappa_1^2)\,\vartheta_1(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\\ ((\mathrm{j}\zeta)^2-\kappa_2^2)\,\vartheta_2(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_2(x)\end{smallmatrix}\right| \quad (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x)\right]\boldsymbol{f}_{\mathrm{C}}^{(0)}(x,\zeta)}{\left|\begin{smallmatrix}\vartheta_1(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x)\\ \vartheta_2(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_2(x)\end{smallmatrix}\right|}
\end{aligned}$$

5

$$= (\mathrm{j}\zeta)^2 [\,1\;0\,] f_{\mathrm{C}}^{(0)}(x,\zeta) + \frac{\left[-\begin{vmatrix} \kappa_1^2\,\vartheta_1(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x) \\ \kappa_2^2\,\vartheta_2(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_2(x) \end{vmatrix} \quad (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x) \right] f_{\mathrm{C}}^{(0)}(x,\zeta)}{\begin{vmatrix} \vartheta_1(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_1(x) \\ \vartheta_2(x) & \frac{\mathrm{d}}{\mathrm{d}x}\vartheta_2(x) \end{vmatrix}}$$

$$= (\mathrm{j}\zeta)^2 [\,1\;0\,] f_{\mathrm{C}}^{(0)}(x,\zeta) + \frac{\left[-\begin{vmatrix} \kappa_1^2\,\vartheta_{1\mathrm{C}1}(x) & \vartheta_{1\mathrm{C}2}(x) \\ \kappa_2^2\,\vartheta_{2\mathrm{C}1}(x) & \vartheta_{2\mathrm{C}2}(x) \end{vmatrix} \quad (\kappa_1^2-\kappa_2^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x) \right] f_{\mathrm{C}}^{(0)}(x,\zeta)}{\begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}},$$

$$\left(\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]\right)^2 \frac{\partial}{\partial x} f^{(2)}(x,\zeta) = \begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}^2 \frac{\partial}{\partial x} f^{(2)}(x,\zeta) \tag{5.116}$$

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

$$= \begin{vmatrix} \vartheta_1(x) & \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \vartheta_2(x) & \frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix} \left[\begin{bmatrix} (\kappa_2^2-\kappa_1^2)\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} & \begin{vmatrix} \vartheta_1(x) & ((\mathrm{j}\zeta)^2-\kappa_1^2)\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \vartheta_2(x) & ((\mathrm{j}\zeta)^2-\kappa_2^2)\frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix} \end{bmatrix} f_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots \right.$$

$$\left. (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x) \begin{bmatrix} \begin{vmatrix} ((\mathrm{j}\zeta)^2-\kappa_1^2)\,\vartheta_1(x) & \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ ((\mathrm{j}\zeta)^2-\kappa_2^2)\,\vartheta_2(x) & \frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x) \end{bmatrix} f_{\mathrm{C}}^{(0)}(x,\zeta) \right]$$

$$= (\mathrm{j}\zeta)^2 \begin{vmatrix} \vartheta_1(x) & \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \vartheta_2(x) & \frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix}^2 \left[(\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x) \begin{vmatrix} \vartheta_1(x) & \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \vartheta_2(x) & \frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix}^{-1} \quad 1 \right] f_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots$$

$$\left. \begin{vmatrix} \vartheta_1(x) & \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \vartheta_2(x) & \frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix} \left[\begin{bmatrix} (\kappa_2^2-\kappa_1^2)\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x}\frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} & -\begin{vmatrix} \vartheta_1(x) & \kappa_1^2\frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \vartheta_2(x) & \kappa_2^2\frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix} \end{bmatrix} f_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots \right.$$

$$\left. (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x) \begin{bmatrix} -\begin{vmatrix} \kappa_1^2\,\vartheta_1(x) & \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} \\ \kappa_2^2\,\vartheta_2(x) & \frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} \end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_1(x)\,\vartheta_2(x) \end{bmatrix} f_{\mathrm{C}}^{(0)}(x,\zeta) \right]$$

$$= (\mathrm{j}\zeta)^2 \begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}^2 \left[\frac{(\kappa_1^2-\kappa_2^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x)}{\begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}} \quad 1 \right] f_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots$$

$$\left. \begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix} \left[\begin{bmatrix} (\kappa_2^2-\kappa_1^2)\,\vartheta_{1\mathrm{C}2}(x)\,\vartheta_{2\mathrm{C}2}(x) & -\begin{vmatrix} \vartheta_{1\mathrm{C}1}(x) & \kappa_1^2\,\vartheta_{1\mathrm{C}2}(x) \\ \vartheta_{2\mathrm{C}1}(x) & \kappa_2^2\,\vartheta_{2\mathrm{C}2}(x) \end{vmatrix} \end{bmatrix} f_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots \right.$$

$$\left. \begin{bmatrix} (\kappa_2^2-\kappa_1^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x) & \begin{vmatrix} \kappa_1^2\,\vartheta_{1\mathrm{C}1}(x) & \vartheta_{1\mathrm{C}2}(x) \\ \kappa_2^2\,\vartheta_{2\mathrm{C}1}(x) & \vartheta_{2\mathrm{C}2}(x) \end{vmatrix} & (\kappa_1^2-\kappa_2^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x) \end{bmatrix} f_{\mathrm{C}}^{(0)}(x,\zeta) \right]$$

$$= (\mathrm{j}\zeta)^2 \begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}^2 \left[\frac{(\kappa_1^2-\kappa_2^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x)}{\begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}} \quad 1 \right] f_{\mathrm{C}}^{(0)}(x,\zeta) + \cdots$$

$$\left[(\kappa_2^2-\kappa_1^2) \begin{vmatrix} \vartheta_{1\mathrm{C}1}(x)\,\vartheta_{1\mathrm{C}2}(x) & \vartheta_{1\mathrm{C}2}^2(x) - \kappa_1^2\,\vartheta_{1\mathrm{C}1}^2(x) \\ \vartheta_{2\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}2}(x) & \vartheta_{2\mathrm{C}2}^2(x) - \kappa_2^2\,\vartheta_{2\mathrm{C}1}^2(x) \end{vmatrix} \quad \cdots \right.$$

$$\left. (\kappa_1^2-\kappa_2^2)^2\,\vartheta_{1\mathrm{C}1}^2(x)\,\vartheta_{2\mathrm{C}1}^2(x) - \begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix} \begin{vmatrix} \vartheta_{1\mathrm{C}1}(x) & \kappa_1^2\,\vartheta_{1\mathrm{C}2}(x) \\ \vartheta_{2\mathrm{C}1}(x) & \kappa_2^2\,\vartheta_{2\mathrm{C}2}(x) \end{vmatrix} \right] f_{\mathrm{C}}^{(0)}(x,\zeta).$$

Hence, from (5.115) and (5.116):

$$\boldsymbol{C}^{(2)}_{\mathrm{C}(0)}(x,\zeta) = (\mathrm{j}\zeta)^2 \begin{bmatrix} 1 & 0 \\ \frac{(\kappa_1^2-\kappa_2^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x)}{w(x)} & 1 \end{bmatrix} + \cdots \tag{5.117}$$

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

$$\cdots \quad \begin{matrix} -w(x)(\kappa_2^2-\kappa_1^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x) \\ (\kappa_2^2-\kappa_1^2)^2\,\vartheta^2_{1\mathrm{C}1}(x)\,\vartheta^2_{2\mathrm{C}1}(x) - w(x) \begin{vmatrix} \vartheta_{1\mathrm{C}1}(x) & \kappa_1^2\,\vartheta_{1\mathrm{C}2}(x) \\ \vartheta_{2\mathrm{C}1}(x) & \kappa_2^2\,\vartheta_{2\mathrm{C}2}(x) \end{vmatrix} \end{matrix} \Bigg],$$

$$\text{where } w(x) := \mathrm{W}[\vartheta_1(x);\vartheta_2(x)] = \begin{vmatrix} \vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x) \end{vmatrix}. \tag{5.118}$$

The potential update is

$$\begin{aligned}
\Delta q^{(2)}_{(0)}(x) = q^{(2)}(x) - q^{(0)}(x) &= 2\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln(\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]) = 2\frac{\mathrm{d}}{\mathrm{d}x}\frac{\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]}{\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]} \\
&= 2\frac{\frac{\mathrm{d}^2}{\mathrm{d}x^2}\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]}{\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]} - 2\left(\frac{\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]}{\mathrm{W}[\vartheta_1(x);\vartheta_2(x)]}\right)^2 \\
&= \frac{2(\kappa_2^2-\kappa_1^2)\left(\vartheta_1(x)\frac{\mathrm{d}\,\vartheta_2(x)}{\mathrm{d}x} + \frac{\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(\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}2}(x) + \vartheta_{1\mathrm{C}2}(x)\,\vartheta_{2\mathrm{C}1}(x)\right)}{\begin{vmatrix}\vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x)\end{vmatrix}} + \cdots \\
&\qquad - 2\left(\frac{(\kappa_2^2-\kappa_1^2)\,\vartheta_{1\mathrm{C}1}(x)\,\vartheta_{2\mathrm{C}1}(x)}{\begin{vmatrix}\vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x)\end{vmatrix}}\right)^2. \\
&= 2(\kappa_2^2-\kappa_1^2)\frac{\begin{vmatrix}\vartheta^2_{1\mathrm{C}1}(x) & \vartheta^2_{1\mathrm{C}2}(x) \\ \vartheta^2_{2\mathrm{C}1}(x) & \vartheta^2_{2\mathrm{C}2}(x)\end{vmatrix} - (\kappa_2^2-\kappa_1^2)\,\vartheta^2_{1\mathrm{C}1}(x)\,\vartheta^2_{2\mathrm{C}1}(x)}{\begin{vmatrix}\vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x)\end{vmatrix}^2} \\
&= 2(\kappa_2^2-\kappa_1^2)\frac{\begin{vmatrix}\vartheta^2_{1\mathrm{C}1}(x) & \vartheta^2_{1\mathrm{C}2}(x) - \kappa_1^2\,\vartheta^2_{1\mathrm{C}1}(x) \\ \vartheta^2_{2\mathrm{C}1}(x) & \vartheta^2_{2\mathrm{C}2}(x) - \kappa_2^2\,\vartheta^2_{2\mathrm{C}1}(x)\end{vmatrix}}{\begin{vmatrix}\vartheta_{1\mathrm{C}}(x) & \vartheta_{2\mathrm{C}}(x)\end{vmatrix}^2}.
\end{aligned} \tag{5.119}$$

5.B.4. Dressing method in E basis

in this section we make use of the results of Appendix 5.B.3 to derive the dressing method in E basis for the 1- and 2-fold Crum transform. These results are used in Section 5.4.

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

Starting from (5.27), (5.29) and (5.30), we can derive the dressing method for the Darboux transform in E basis, the case $N = 1$ of (5.41) and (5.42) in Section 5.4.1, as follows. For

convenience, rewrite (5.30) using (5.97) and $\sigma_1(x) := \frac{\mathrm{d}\,\vartheta_1(x)}{\mathrm{d}x} / \vartheta_1(x)$ as

$$\begin{aligned}\boldsymbol{C}^{(1)}_{\mathrm{C}(0)}(x,\zeta) &= \begin{bmatrix} -\sigma_1(x) & 1 \\ (\sigma_1(x)-\kappa_1)\,(\sigma_1(x)+\kappa_1)+(\mathrm{j}\zeta)^2 & -\sigma_1(x)\end{bmatrix} \\ &\equiv \boldsymbol{A}_{\mathrm{C}}(x,\zeta)\big|_{q(x)=\kappa_1^2-\sigma_1^2(x)} - \sigma_1(x)\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]. \end{aligned} \tag{5.120}$$

Then use (5.89), (5.92), (5.94) and (5.96) to find

$$\begin{aligned}\boldsymbol{C}^{(1)}_{\mathrm{E}(0)}(x,\zeta) &= \boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\,\boldsymbol{C}^{(1)}_{\mathrm{C}(0)}(x,\zeta)\,\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) \\ &= \boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\,\boldsymbol{A}_{\mathrm{C}}(x,\zeta)\big|_{q(x)=\kappa_1^2-\sigma_1^2(x)}\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) - \sigma_1(x)\,\boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) \\ &= \boldsymbol{A}_{\mathrm{E}}(x,\zeta)\big|_{q(x)=\kappa_1^2-\sigma_1^2(x)} - \left(\tfrac{\partial}{\partial x}\,\boldsymbol{T}^{\mathrm{E}}_{\mathrm{C}}(x,\zeta)\right)\boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\zeta) - \sigma_1(x)\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right] \\ &= \frac{\kappa_1^2-\sigma_1^2(x)}{2\mathrm{j}\zeta}\begin{bmatrix}1 & \exp(2\mathrm{j}\zeta x)\\ -\exp(-2\mathrm{j}\zeta x) & -1\end{bmatrix} + \mathrm{j}\zeta\begin{bmatrix}-1&0\\0&1\end{bmatrix} - \sigma_1(x)\begin{bmatrix}1&0\\0&1\end{bmatrix}. \end{aligned} \tag{5.121}$$

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

$$\begin{aligned}\boldsymbol{\vartheta}_{n\mathrm{C}}(x) &= \boldsymbol{T}^{\mathrm{C}}_{\mathrm{E}}(x,\mathrm{j}\kappa_n)\,\boldsymbol{\vartheta}_{n\mathrm{E}}(x) = \begin{bmatrix}\mathrm{e}^{\kappa_n x} & \mathrm{e}^{-\kappa_n x}\\ \kappa_n\mathrm{e}^{\kappa_n x} & -\kappa_n\mathrm{e}^{-\kappa_n x}\end{bmatrix}\begin{bmatrix}\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\\ \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}\\ \kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} - \kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}\end{bmatrix}, \text{ so} \end{aligned} \tag{5.122}$$

$$\sigma_n(x) = \frac{\boldsymbol{\vartheta}_{n\mathrm{C}2}(x)}{\boldsymbol{\vartheta}_{n\mathrm{C}1}(x)} = \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} - \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}; \tag{5.123}$$

$$\begin{aligned}\kappa_n + \sigma_n(x) &= \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}} + \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} - \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}} \\ &= \frac{2\kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}; \end{aligned} \tag{5.124}$$

$$\begin{aligned}\kappa_n - \sigma_n(x) &= \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}} - \kappa_n\frac{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} - \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}} \\ &= \frac{2\kappa_n\,\boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}{\boldsymbol{\vartheta}_{n\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{n\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}}. \end{aligned} \tag{5.125}$$

Substitution in (5.114) and (5.121) results in

$$\begin{aligned}\boldsymbol{C}^{(1)}_{\mathrm{E}(0)}(x,\zeta) = (\mathrm{j}\zeta)^{-1}\overbrace{\frac{2\kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{E}2}(x)}{\left(\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}\right)^2}\begin{bmatrix}1 & \mathrm{e}^{2\mathrm{j}\zeta x}\\ -\mathrm{e}^{-2\mathrm{j}\zeta x} & -1\end{bmatrix}}^{=\boldsymbol{M}^{(1)}_{-1}(x,\zeta)} + \cdots \\ \underbrace{\mathrm{j}\zeta\begin{bmatrix}-1&0\\0&1\end{bmatrix}}_{=\boldsymbol{M}^{(1)}_{1}(x,\zeta)} + \underbrace{\kappa_1\frac{\boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\mathrm{e}^{-\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\mathrm{e}^{\kappa_1 x}}{\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\mathrm{e}^{-\kappa_1 x}}\begin{bmatrix}1&0\\0&1\end{bmatrix}}_{=\boldsymbol{M}^{(1)}_{0}(x,\zeta)}, \end{aligned} \tag{5.126}$$

$$\Delta q^{(1)}_{(0)}(x) = 2(\kappa_1 - \sigma_1(x))(\kappa_1 + \sigma_1(x)) - 2\,q^{(0)}(x) \tag{5.127}$$

$$= \frac{8\kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{E}2}(x)}{\left(\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\mathrm{e}^{\kappa_n x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}(x)\mathrm{e}^{-\kappa_n x}\right)^2} - 2\,q^{(0)}(x).$$

This is equivalent to the equations in Section 5.4.1 for the case $N = 1$.

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 (5.92) with (5.94) to (5.117). Let us first express (5.117) as

$$\boldsymbol{C}^{(2)}_{\mathrm{C}(0)}(x,\zeta) = (\mathrm{j}\zeta)^2 \begin{bmatrix} 1 & 0 \\ \frac{a}{w(x)} & 1 \end{bmatrix} + \frac{1}{w^2(x)} \begin{bmatrix} 2b\,w(x) & a\,w(x) \\ 2c & 2d \end{bmatrix}, \quad \text{where} \tag{5.128}$$

$$a := (\kappa_1^2 - \kappa_2^2)\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x), \tag{5.129}$$

$$b := -\frac{1}{2}\begin{vmatrix} \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix}, \tag{5.130}$$

$$c := \frac{1}{2}(\kappa_2^2 - \kappa_1^2)\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) & \boldsymbol{\vartheta}^2_{1\mathrm{C}2}(x) - \kappa_1^2\,\boldsymbol{\vartheta}^2_{1\mathrm{C}1}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) & \boldsymbol{\vartheta}^2_{2\mathrm{C}2}(x) - \kappa_2^2\,\boldsymbol{\vartheta}^2_{2\mathrm{C}1}(x) \end{vmatrix}, \tag{5.131}$$

$$d := \frac{1}{2}(\kappa_2^2 - \kappa_1^2)^2\,\boldsymbol{\vartheta}^2_{1\mathrm{C}1}(x)\,\boldsymbol{\vartheta}^2_{2\mathrm{C}1}(x) - \frac{1}{2}\,w(x)\begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}1}(x) & \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{C}2}(x) \\ \boldsymbol{\vartheta}_{2\mathrm{C}1}(x) & \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{C}2}(x) \end{vmatrix}, \tag{5.132}$$

$$w(x) := \mathrm{W}[\vartheta_1(x);\vartheta_2(x)] = \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{C}}(x) & \boldsymbol{\vartheta}_{2\mathrm{C}}(x) \end{vmatrix}. \tag{5.133}$$

Applying (5.92) with (5.94) then gives

$$\boldsymbol{C}^{(2)}_{\mathrm{E}(0)}(x,\zeta) = (\mathrm{j}\zeta)\overbrace{\frac{a}{w(x)}\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}}^{=\boldsymbol{M}^{(2)}_1(x,\zeta)} + (\mathrm{j}\zeta)^{-1}\overbrace{\frac{1}{w^2(x)}\begin{bmatrix} -c & -c\,\mathrm{e}^{2\mathrm{j}\zeta x} \\ c\,\mathrm{e}^{-2\mathrm{j}\zeta x} & c \end{bmatrix}}^{=\boldsymbol{M}^{(2)}_{-1}(x,\zeta)} + \cdots \tag{5.134}$$

$$\underbrace{\frac{1}{w^2(x)}\begin{bmatrix} b\,w(x)+d & (b\,w(x)-d)\mathrm{e}^{2\mathrm{j}\zeta x} \\ (b\,w(x)-d)\mathrm{e}^{-2\mathrm{j}\zeta x} & b\,w(x)+d \end{bmatrix}}_{=\boldsymbol{M}^{(2)}_0(x,\zeta)} + (\mathrm{j}\zeta)^2\underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{=\boldsymbol{M}^{(2)}_2(x,\zeta)}.$$

Substitution of (5.122) results in

$$a = (\kappa_1^2 - \kappa_2^2)(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x})(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x}), \tag{5.135}$$

$$c = 2(\kappa_2^2 - \kappa_1^2)\kappa_1\kappa_2\begin{vmatrix} \boldsymbol{\vartheta}^2_{1\mathrm{E}1}(x)\mathrm{e}^{2\kappa_1 x} - \boldsymbol{\vartheta}^2_{1\mathrm{E}2}(x)\mathrm{e}^{-2\kappa_1 x} & \kappa_1\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{E}2}(x) \\ \boldsymbol{\vartheta}^2_{2\mathrm{E}1}(x)\mathrm{e}^{2\kappa_2 x} - \boldsymbol{\vartheta}^2_{2\mathrm{E}2}(x)\mathrm{e}^{-2\kappa_2 x} & \kappa_2\,\boldsymbol{\vartheta}_{2\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{E}2}(x) \end{vmatrix}, \tag{5.136}$$

$$w(x) = \begin{vmatrix} \boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x} & \kappa_1\,(\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} - \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x}) \\ \boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x} & \kappa_2\,(\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} - \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x}) \end{vmatrix}, \tag{5.137}$$

$$b\,w(x) - d = 2(\kappa_1^2 - \kappa_2^2)\begin{vmatrix} (\boldsymbol{\vartheta}_{1\mathrm{E}1}\mathrm{e}^{\kappa_1 x} + \boldsymbol{\vartheta}_{1\mathrm{E}2}\mathrm{e}^{-\kappa_1 x})^2 & \kappa_1^2\,\boldsymbol{\vartheta}_{1\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{1\mathrm{E}2}(x) \\ (\boldsymbol{\vartheta}_{2\mathrm{E}1}\mathrm{e}^{\kappa_2 x} + \boldsymbol{\vartheta}_{2\mathrm{E}2}\mathrm{e}^{-\kappa_2 x})^2 & \kappa_2^2\,\boldsymbol{\vartheta}_{2\mathrm{E}1}(x)\,\boldsymbol{\vartheta}_{2\mathrm{E}2}(x) \end{vmatrix}, \tag{5.138}$$

$$b\,w(x) + d = \tfrac{1}{2}\left(a^2 - (\kappa_1^2 + \kappa_2^2)\,w^2(x)\right). \tag{5.139}$$

The equations for the trajectory mapping in Section 5.4.1 for the case $N = 2$ then follow by substitution in (5.134). Finally, we find the potential update for that case by substituting

(5.122) in (5.119):

$$\begin{aligned}\Delta q^{(2)}_{(0)}(x) &= 2(\kappa_2^2-\kappa_1^2)\frac{\begin{vmatrix}\vartheta^2_{1C1}(x) & \vartheta^2_{1C2}(x)-\kappa_1^2\,\vartheta^2_{1C1}(x)\\ \vartheta^2_{2C1}(x) & \vartheta^2_{2C2}(x)-\kappa_2^2\,\vartheta^2_{2C1}(x)\end{vmatrix}}{\left|\vartheta_{1C}(x)\quad \vartheta_{2C}(x)\right|^2}\\ &= 8(\kappa_1^2-\kappa_2^2)\frac{\begin{vmatrix}(\vartheta_{1E1}e^{\kappa_1 x}+\vartheta_{1E2}e^{-\kappa_1 x})^2 & \kappa_1^2\,\vartheta_{1E1}(x)\,\vartheta_{1E2}(x)\\ (\vartheta_{2E1}e^{\kappa_2 x}+\vartheta_{2E2}e^{-\kappa_2 x})^2 & \kappa_2^2\,\vartheta_{2E1}(x)\,\vartheta_{2E2}(x)\end{vmatrix}}{\begin{vmatrix}\vartheta_{1E1}e^{\kappa_1 x}+\vartheta_{1E2}e^{-\kappa_1 x} & \kappa_1\,(\vartheta_{1E1}e^{\kappa_1 x}-\vartheta_{1E2}e^{-\kappa_1 x})\\ \vartheta_{2E1}e^{\kappa_2 x}+\vartheta_{2E2}e^{-\kappa_2 x} & \kappa_2\,(\vartheta_{2E1}e^{\kappa_2 x}-\vartheta_{2E2}e^{-\kappa_2 x})\end{vmatrix}^2}.\end{aligned} \tag{5.140}$$

6

Conclusions and recommendations

6.1. Conclusions

The goal of the project that is documented in this dissertation was to design better algorithms for the numerical computation of the forward and inverse Non-linear Fourier Transform (NFT) for the Korteweg–de Vries equation (KdV), for sampled signals that decay sufficiently fast on both sides. For most parts of the numerical NFT the existing algorithms were not accurate enough to process realistic data in ordinary double precision floating point arithmetic. In this dissertation more accurate algorithms have been developed for these parts. That is, algorithms that achieve a sufficient level of accuracy without resorting to higher precision arithmetic and without increasing the order of the computational complexity compared to the fastest existing algorithms. For the computation of the continuous spectrum numerical algorithms were available that were accurate enough, but these had an undesirably high computational cost. In this dissertation a faster yet barely less accurate algorithm has been developed for this computation. Naturally the algorithms that are developed in this dissertation build on the work of other authors. The remainder of this section is divided between different parts of the numerical KdV-NFT. For each part the state of the art before this project is summarized, followed by a description of the algorithm that is presented in this dissertation and the results that are achieved with this new algorithm.

6.1.1. Computation of the continuous KdV-NFT spectrum

State of the art before this project

Prior to this project the only numerical NFT algorithms for sampled data that were specifically aimed at the KdV were due to Alfred R Osborne and his co-workers [17, 101, 114]. The principle was to use a piecewise constant interpolation of the input samples. In each piecewise constant interval the Schrödinger equation was solved exactly. This solution was represented as a set of 2×2 matrices, one for each sample. Finally the spectrum of the full input signal was obtained from the product of all these matrices. It was also known that some computation time could be saved by, essentially, multiplying the matrix product by suitable vector. Performed in the right order, this reduced the computation from D matrix matrix

multiplications to $D+1$ matrix vector multiplications, where D is the number of input samples. (This however does not save half of the computation time, since the calculation of the matrix entries takes more computation time than the matrix multiplication thereafter.) In any case, the whole procedure had to be repeated for every desired sample of the spectrum. If the number of output samples was chosen equal to the number of input samples, the overall computational complexity was $O\left(D^2\right)$.

Many numerical NFT developments, both before and during this project, were primarily aimed at the NFT with respect to the Non-linear Schrödinger Equation (NSE). There are many similarities between the NSE-NFT and the KdV-NFT, but also a few differences that make porting NSE-NFT developments to the KdV-NFT harder than it seems at first sight. A particularly important development was the Fast Non-linear Fourier Transform (FNFT) [150]. This algorithm also represented the solution of (the NSE equivalent of) the Schrödinger equation as a 2×2 matrix for each sample, but instead of filling out the value of the 'non-linear frequencies' sample, it was left as the variable in a polynomial that approximated the true function. After multiplying all the matrices a 2×2 matrix of polynomials resulted. These were then evaluated for each desired spectral sample. By using fast algorithms for both the matrix multiplications and the evaluation of the polynomials, the overall computational complexity was only $O\left(D\left(\log(D)\right)^2\right)$. At the start of this project the NSE-FNFT was already available as part of the FNFT software library [152]. The FNFT had not been demonstrated yet for the KdV, but it was expected to require only minor changes, since the KdV fits into the Ablowitz–Kaup–Newell–Segur (AKNS) framework [4] on which the NSE-FNFT was based.

Algorithm presented in this dissertation

The initial attempts to adapt the FNFT algorithm to the KdV resulted in unexpectedly large errors. Only for very low 'non-linear frequencies' did the result reasonably match with the theoretic spectrum. At greater 'non-linear frequencies' the result was dominated by an error floor with a high pass spectrum. To achieve an acceptable accuracy, a set of higher order polynomial approximation schemes for the FNFT is developed in this dissertation. This is a set of exponential splitting schemes with rational real coefficients, as required for use in the FNFT. The best known example of a splitting scheme is Strang splitting, which is of order two. In the literature suitable splitting schemes were only found up till an order of four. In this dissertation four more schemes are added, of orders five, six, seven and eight respectively. Every increase of the order of the splitting scheme by one results in a reduction of the error floor by roughly one order of magnitude and thus increases the domain of 'non-linear frequencies' for which the KdV-FNFT is usable. The new splitting schemes are now also used for the NSE-FNFT. Because the NSE-FNFT already showed a lower error floor and because that error floor had a low-pass characteristic, the improvements are less dramatic than in the KdV case. Nevertheless an example is shown in which the new splitting scheme of order eight reduces the noise floor by a factor between 100 and 1000, to become almost indistinguishable from the error floor of the slow NSE-NFT, which is solely due to the numerical integrator.

Later in this project it is discovered that the numerical error in the KdV-FNFT is related to the asymmetric way in which the KdV is encapsulated in the AKNS framework on which the FNFT is based. As a result, the KdV-FNFT shows a similar error behaviour as the NSE-

FNFT for negative 'non-linear frequencies'. Hence the high errors are avoided by essentially computing the spectrum first for negative 'non-linear frequencies', and then exploit symmetries to compute from that the spectrum for positive 'non-linear frequencies'. This procedure is equivalent to a different encapsulation of the KdV in the AKNS framework.

Both the new higher order exponential splitting schemes and the favourable encapsulation have been implemented in the FNFT software library, which is freely available on the internet for commercial and non-commercial use under the terms of the GNU's not Unix (GNU) General Public Licence (GPL) version 2 [152]. Hence as a result of the work in this dissertation the continuous KdV-NFT spectrum can now be computed with a ready to use piece of software at a computational complexity of only $O\left(D\left(\log(D)\right)^2\right)$.

6.1.2. Computation of the eigenvalues of the discrete KdV-NFT spectrum

State of the art before this project

The eigenvalues of the KdV are actually eigenvalues of the Schrödinger equation. The numerical computation of these eigenvalues had been researched in different fields. Unfortunately there had been seemingly little cross fertilisation between these different fields. Scientific publications in one field mostly cited other publications within the same field and often different terminology and notation were used. Also the assumptions on the input signal often differed. From a mathematical perspective the input signal was typically considered to be a function whereas for the applications of the NFT typically only sampled data is available. Also, for the KdV-NFT for input signals with vanishing boundaries, the Schrödinger equation must be solved on the line (with boundary conditions at infinity), whereas other fields are more interested in solutions on an interval (with boundary conditions at the finite interval's boundaries).

There were two main approaches to compute eigenvalues of the KdV and Schrödinger equation: Finite dimensional approximations and shooting methods. The first worked by approximating the continuous eigenvalue problem by a matrix eigenvalue problem. The computational cost of these algorithms scaled at best quadratically in the number of degrees of freedom and their accuracy typically deteriorated quickly for faster oscillating eigenfunctions [82, Chap. 2]. Furthermore some of these algorithms required samples on a specific grid that did not necessarily coincide with the available data samples. The second approach worked by integrating the Schrödinger equation for a sequence candidate eigenvalues that were selected with an iterative root finder, such as Newton–Raphson (NR). In the NFT literature it was known how to evaluate samples of the objective function of which the roots are the KdV eigenvalues, and its gradient [17]. The main difficulty of the shooting approach was the selection of suitable initial guesses for the algorithm such that every eigenvalue would be found. There are a few applications for which initial guesses are available: If the NFT is used for communication over a non-linear fibre optical transmission line modelled by the NSE or over a non-linear electrical transmission line modelled by the KdV, the possible eigenvalues may be fixed in a modulation scheme. However, for the majority of applications such knowledge is not available. The NSE-FNFT mitigated this problem by using a finite dimensional approximation to obtain initial guesses followed by a shooting method for refinement. From a mathematical perspective, the Schrödinger equation was regarded as an example of a

Sturm–Liouville (SL) equation. In this field many algorithms had been developed that were based on SL oscillation theory. This theory relates the ordinal number (index) of each eigenvalue to the number of zero-crossings of the corresponding eigenfunction. This knowledge was exploited in an iterative root finder. See e.g. [82] and the references therein. Also in the NFT literature an algorithm had been proposed that essentially made use of SL oscillation theory [102], seemingly unaware of the prior existence of similar methods in other fields of research.

Algorithm presented in this dissertation

In this dissertation an SL type algorithm is proposed that is fine tuned for the requirements of the KdV-NFT. The algorithm thus makes use of sampled input data and boundary conditions at infinity (for which the input is approximated as zero outside the sampled interval). In contrast to existing SL type algorithms, the proposed algorithm simultaneously makes use of the gradient of the objective function and the discrete information based on SL oscillation theory. It is shown that the use of SL information because of its discrete nature requires special attention to the behaviour of the algorithm at discontinuities, such as increments of the oscillation counter. It is demonstrated that the proposed algorithm is, for sampled input data with vanishing boundary conditions, more reliable than the considered state of the art benchmark algorithm. Furthermore it typically saves computation time by requiring fewer iterations than existing algorithms that rely more heavily on bisection.

6.1.3. Computation of the norming constants of the discrete KdV-NFT spectrum

State of the art before this project

It was known that the direct computation of the norming constants of the discrete (NSE-) NFT spectrum was numerically highly unstable. Hari and Kschischang [63] and Aref [12] had independently developed a more stable method, which they named the the bi-directional algorithm or the forward-backward method respectively. They had proposed for the NSE-NFT to numerically solve the NSE equivalent of the Schrödinger equation in the AKNS framework (respectively a slight modification thereof) from two sides. That is, they had formulated certain boundary conditions at plus and minus infinity respectively. From there the trajectory was computed up to a central matching point. At the matching point they had each found an equation that expressed the norming constant $b(\zeta_n)$ as an affine function in the scattering parameter $a(\zeta_n)$. The coefficients of this affine function depended on the values of both trajectories at the matching point. Since $a(\zeta_n)$ should be zero at an eigenvalue, they could cancel this term from the equation. They had found that the remaining term was a numerically reliable estimator of the norming constant, if the matching point was chosen well. They had both proposed a criterion to select this matching point.

Algorithm presented in this dissertation

The direct computation of the norming constants of the discrete KdV-NFT spectrum shows the same numerical instability as the similar computation for the NSE-NFT. However, from the exposition in [12, 63] it is not immediately clear how the bi-directional algorithm can be generalised for the KdV. In the AKNS framework, the scattering parameters $a(\zeta_n)$ and $b(\zeta)$ of the KdV-NFT are not as directly related to the value of trajectories at a matching point as

in the case of the NSE-NFT. Namely, the scattering parameters are in [4, Eqs. (A3.2)–(A3.3)] found as the coefficients in a particular parametrization of the trajectory at plus infinity, that results from a particular boundary condition at minus infinity. It is not explained therein why the parametrization and boundary condition should look like this. Therefore it is not obvious what the boundary condition at plus infinity should be for the purpose of the bi-directional algorithm. Furthermore it is a tedious task to find for the KdV-NFT an equation that expresses the norming constant as a function of the scattering parameter $a(\zeta_n)$ and the values of certain trajectories of the AKNS representation of the Schrödinger equation at the matching point. The solution is not even unique, since in theory $a(\zeta_n)$ should be exactly equal to zero. In this dissertation the problem is solved by expressing the Schrödinger equation not in the ubiquitous AKNS basis, but in another system of first order Ordinary Differential Equations (ODEs) which is called S-basis in this dissertation. These representations can be transformed into one another, in most cases by left and right multiplication by suitable transformation matrices. (It is however not always a similarity transformation, because the transformation matrices can depend on x.) The S-basis has for the KdV-NFT the property that the scattering parameters are related to its trajectories in a simple way. Hence once the Schrödinger equation is transformed into the S-basis, application of the bi-directional algorithm becomes simple. One obtains an equation that expresses the norming constants as an affine function in the scattering parameter $a(\zeta_n)$, of which the coefficients depend on the values of both trajectories in S-basis at the matching point. If desired, this expression can be transformed back into the AKNS basis after cancelling the term that contains the factor $a(\zeta_n)$. Hence this dissertation has contributed to the generalisation of the bi-directional algorithm for different NFTs.

Moreover, the bidirectional algorithm itself is improved in this dissertation by developing a better criterion for the selection of the matching point. At the midpoint between every x sample two different estimates of the norming constant are computed. The point where the difference between these estimates is the smallest is selected as the matching point. The new criterion works even in cases where the previously proposed criteria break down. In these cases the new criterion improves the accuracy by several orders of magnitude. This improvement mostly shows for signals of which the energy is not concentrated around one location. Up till today no example is known in which the new criterion does not completely cancel the numerical instability and leaves only the ordinary discretization error of the integrator. Until such an example is found, the accuracy of the numerical computation of the norming constants should be considered as a solved problem.

The bi-directional algorithm with the matching point criterion as proposed in this dissertation is implemented in the FNFT software library [152].

6.1.4. Computation of the inverse NFT from the discrete KdV-NFT spectrum

State of the art before this project

Before this project there existed several numerical algorithms for the computation of the inverse KdV–NFT that consider both the continuous and the discrete spectrum. Among these were algorithms that are based on solving the Gel'fand–Levitan–Marchenko (GLM) equation, e.g. [122] or on solving an equivalent Riemann–Hilbert problem, e.g. [142, 143]. Several authors had discussed a two-stage approach in which the (suitably pre-compensated)

continuous spectrum is transformed first and the discrete spectrum is accounted for thereafter by means of Darboux or Crum transforms, e.g. [122, §4.2], [40, Sect. 3], [28, Chap. xvii.3.2]. This approach was considered attractive for numerical computations [122, §4.2].

The Darboux transform adds (or removes) one eigenvalue to (or from) a known signal spectrum pair [40, 59]. Starting from the trivial signal spectrum pair $q(x) = 0 \Leftrightarrow (R(\zeta), \mathcal{D}) = (0, \emptyset)$, the signal corresponding to any discrete spectrum can be computed by adding the eigenvalues one by one with the Darboux transform. The Crum transform is a generalisation of the Darboux transform that adds (or removes) any number of eigenvalues to (or from) a known signal spectrum pair [38]. It was known how to exploit the Schrödinger equation to compute the Darboux or Crum transform without computing higher order derivatives [40, pp. 177–178]. An implementation of the Crum transform for Partial Differential Equations (PDEs) that fit in the AKNS framework had been presented in [99]. Numerical implementations of these algorithms had not been discussed in the literature.

Algorithm presented in this dissertation

When Darboux or Crum transform algorithms are straightforwardly implemented in ordinary double precision floating point arithmetic, the numerical properties are poor. Typically the error in the computed signal becomes unacceptably high already for only a few eigenvalues. Therefore these algorithms are not reliable enough for numerical computations in practice. In this dissertation the numerical properties of these algorithms are carefully examined. A new algorithm is developed that mitigates these errors and is thus optimized for computations in ordinary floating point arithmetic. Examples indicate that the new algorithm can process discrete spectra with three to seven times as many eigenvalues at the same error tolerance in comparison to the earlier existing algorithms. Hence the Crum transform can now be computed numerically in ordinary double precision floating point arithmetic for a larger range of spectra than before.

6.2. Recommendations

6.2.1. Inverse KdV-NFT for arbitrary spectra

In this dissertation a numerical algorithm is proposed for the inverse KdV-NFT for the discrete spectrum, which is optimized for floating point arithmetic. If this algorithm is complemented with a likewise optimized algorithm to compute the inverse KdV-NFT for an arbitrary continuous spectrum (but empty discrete spectrum) the inverse KdV-NFT could be computed accurately for arbitrary spectra in floating point arithmetic. It should be investigated if such a complementary algorithm can be based on an existing optimized algorithm to compute the inverse NSE-NFT for an arbitrary continuous spectrum, and if the result is sufficiently accurate. Alternatively such a complementary algorithm could be obtained by optimizing an existing algorithm for the inverse KdV-NFT for arbitrary spectra such as [122, 141, 142]. The trade-off between accuracy and computational cost of the resulting two-stage algorithm should be compared quantitatively to the original versions of these algorithms.

6.2.2. NFT for the Kaup–Broer equation

One of the physical processes that can be modelled by the KdV are free surface waves in shallow water. However, the KdV applies only if the waves all travel in the same direction.

Often reflected waves are of practical interest, for example to study the effectiveness of structures that are made to reflect waves. For waves in the linear regime methods exist to separate the incoming and reflected wave. This boils down to identifying parameters in a linear wave model and relies on superposition. In shallow water conditions axiomatically no linear model applies and hence any separation by means of fitting parameters in a linear model is meaningless. Waves that travel in two directions in shallow water conditions can be modelled by a Boussinesq-like equation known as the Kaup–Broer equation (KB) [20, 75]. This was experimentally verified in a wave flume for a head-on collision of two solitons and for an overtaking interaction of two solitons [118]. As one would expect, by assuming that waves travel in one direction only, the KB can be simplified to the KdV [42, §5.1]. Like the KdV the KB is integrable in the sense of the NFT [75, 77, 121]. It is of interest for reflection analysis of shallow water waves to have a numerical KB-NFT. The KB fits in the AKNS framework [121, Eq. (94)]. However, it shows the same entry -1 in the dynamics matrix as the KdV counterpart. In the KdV case the entry -1 was the cause of many of the numerical difficulties of the KdV-NFT in comparison to the NSE-NFT. Hence it is worth investigating numerical algorithms for the KB-NFT and develop numerically reliable ready to use algorithms.

Bibliography

[1] M. Abdou, A. Hendi, and H. K. Alanzi. "New Exact Solutions of KdV Equation in an Elastic Tube Filled with a Variable Viscosity Fluid". In: *Studies in Nonlinear Sciences* 3.2 (2012), pp. 62–68. ISSN: 2221-3910. DOI: 10.5829/idosi.sns.2012.3.2.247. URL: https://idosi.org/sns/3(2)12/2.pdf (cit. on pp. 25, 87).

[2] N. H. Abel. "Ueber einige bestimmte Integrale". In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 2 (Jan. 1827), pp. 22–30. DOI: 10.1515/crll.1827.2.22 (cit. on pp. 7, 80).

[3] M. J. Ablowitz and J. F. Ladik. "A Nonlinear Difference Scheme and Inverse Scattering". In: *Studies in Applied Mathematics* 55.3 (Sept. 1976), pp. 213–229. DOI: 10.1002/sapm1976553213 (cit. on pp. 4, 26).

[4] M. J. Ablowitz, D. J. Kaup, A. C. Newell, and H. Segur. "The Inverse Scattering Transform - Fourier Analysis for Nonlinear Problems". In: *Studies in Applied Mathematics* 53.4 (Dec. 1974), pp. 249–315. DOI: 10.1002/sapm1974534249 (cit. on pp. 4, 8, 16–18, 26, 62–66, 81–83, 85, 88, 122, 125).

[5] M. J. Ablowitz and H. Segur. *Solitons and the Inverse Scattering Transform*. Philadelphia, PA, USA: SIAM, Jan. 1981. DOI: 10.1137/1.9781611970883 (cit. on pp. 2, 4, 6, 8, 9, 26, 30, 60, 63, 88, 91, 97, 105).

[6] M. J. Ablowitz. "Integrability and Nonlinear Waves". In: *Emerging Frontiers in Nonlinear Science*. Ed. by P. G. Kevrekidis, J. Cuevas-Maraver, and A. Saxena. Springer International Publishing, 2020, pp. 161–184. DOI: 10.1007/978-3-030-44992-6_7 (cit. on p. 90).

[7] M. J. Ablowitz and Y. Kodama. "Note on Asymptotic Solutions of the Korteweg-de Vries Equation with Solitons". In: *Studies in Applied Mathematics* 66.2 (Apr. 1982), pp. 159–170. DOI: 10.1002/sapm1982662159 (cit. on pp. 60, 62, 64, 88).

[8] V. É. Adler. "A modification of Crum's method". In: *Theoretical and Mathematical Physics* 101.3 (1994), pp. 1381–1386. DOI: 10.1007/bf01035458 (cit. on p. 98).

[9] T. Ak, A. Saha, S. Dhawan, and A. H. Kara. "Investigation of Coriolis effect on oceanic flows and its bifurcation via geophysical Korteweg–de Vries equation". In: *Numerical Methods for Partial Differential Equations* (2020). DOI: 10.1002/num.22469 (cit. on p. 87).

[10] T. Aktosun. "Inverse Scattering Transform, KdV, and Solitons". In: *Current Trends in Operator Theory and its Applications*. Birkhäuser Basel, 2004, pp. 1–22. DOI: 10.1007/978-3-0348-7881-4_1 (cit. on p. 6).

[11] A. Alvermann, H. Fehske, and P. B. Littlewood. "Numerical time propagation of quantum systems in radiation fields". In: *New Journal of Physics* 14.105008 (Oct. 2012), pp. 1–22. DOI: 10.1088/1367-2630/14/10/105008 (cit. on p. 41).

[12] V. Aref. *Control and Detection of Discrete Spectral Amplitudes in Nonlinear Fourier Spectrum*. May 2016. arXiv: 1605.06328v1 [math.NA] (cit. on pp. 11, 61, 66–68, 70, 85, 124).

[13] T. Baeyens. *Welcome to the documentation of Pyslise*. URL: https://matslise.ugent.be (visited on 07/02/2021) (cit. on p. 47).

[14] T. Baeyens and M. Van Daele. "The fast and accurate computation of eigenvalues and eigenfunctions of time-independent one-dimensional Schrödinger equations". In: *Computer Physics Communications* 258 (Aug. 26, 2020), p. 107568. DOI: 10.1016/j.cpc.2020.107568 (cit. on pp. 47, 53).

[15] P. B. Bailey, W. N. Everitt, and A. Zettl. "Algorithm 810: The SLEIGN2 Sturm-Liouville Code". In: *ACM Transactions on Mathematical Software* 27.2 (June 2001), pp. 143–192. DOI: 10.1145/383738.383739 (cit. on p. 27).

[16] S. Blanes, F. Casas, and M. Thalhammer. "High-order commutator-free quasi-Magnus exponential integrators for non-autonomous linear evolution equations". In: *Computer Physics Communications* 220 (Nov. 2017), pp. 243–262. DOI: 10.1016/j.cpc.2017.07.016 (cit. on pp. 40, 41, 68, 71).

[17] G. Boffetta and A. R. Osborne. "Computation of the Direct Scattering Transform for the Nonlinear Schroedinger Equation". In: *Journal of Computational Physics* 102.2 (Oct. 1992), pp. 252–264. DOI: 10.1016/0021-9991(92)90370-e (cit. on pp. 4, 26, 37, 38, 64, 71, 83, 108, 121, 123).

[18] J. Boussinesq. "Théorie des ondes et des remous qui se propagent le long d'un canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement pareilles de la surface au fond". In: *Journal de mathématiques pures et appliquées* 17.2 (1872), pp. 55–108 (cit. on p. 2).

[19] J. K. Brenne and J. Skaar. "Design of Grating-Assisted Codirectional Couplers With Discrete Inverse-Scattering Algorithms". In: *Journal of lightwave technology* 21.1 (2003), pp. 254–263. DOI: 10.1109/jlt.2003.808648 (cit. on pp. 4, 26).

[20] L. Broer. "Approximate equations for long water waves". In: *Applied Scientific Research* 31.5 (1975), pp. 377–395. DOI: 10.1007/BF00418048 (cit. on p. 127).

[21] A. Bruckstein and T. Kailath. "An Inverse Scattering Framework for Several Problems in Signal Processing". In: *IEEE ASSP Magazine* 4.1 (1987), pp. 6–20. DOI: 10.1109/massp.1987.1165567 (cit. on p. 17).

[22] A. M. Bruckstein, B. C. Levy, and T. Kailath. "Differential methods in inverse scattering". In: *SIAM Journal on Applied Mathematics* 45.2 (1985), pp. 312–335. DOI: 10.21236/ada458729 (cit. on p. 17).

[23] M. Brühl. "Direct and inverse nonlinear Fourier transform based on the KortewegdeVries equation (KdV-NFT). A spectral analysis of nonlinear surface waves in shallow water". PhD thesis. Technischen Universität Carolo-Wilhelmina, Braunschweig, Nov. 2013. URL: https://nbn-resolving.org/urn:nbn:de:gbv:084-14112709287 (cit. on pp. 4, 26, 60).

[24] M. Brühl and M. Becker. "Analysis of Subaerial Landslide Data Using Nonlinear Fourier Transform Based on Korteweg-deVries Equation (KdV-NFT)". In: *Journal of Earthquake and Tsunami* 12.02 (Apr. 2018), p. 1840002. DOI: 10.1142/s179343111840002x (cit. on pp. 25, 60, 87, 88).

[25] M. Brühl and H. Oumeraci. "Analysis of long-period cosine-wave dispersion in very shallow water using nonlinear Fourier transform based on KdV equation". In: *Applied Ocean Research* 61 (Dec. 2016), pp. 81–91. DOI: 10.1016/j.apor.2016.09.009 (cit. on pp. 4, 15, 25, 26, 59, 60).

[26] M. Brühl, P. J. Prins, S. Ujvary, I. Barranco, S. Wahls, and P. L.-F. Liu. "Comparative analysis of bore propagation using conventional linear and KdV-based nonlinear Fourier transform". In: *Wave Motion* 111 (2022), p. 102905. ISSN: 0165-2125. DOI: 10.1016/j.wavemoti.2022.102905 (cit. on pp. 5, 80, 88, 89, 147).

[27] S. Z. Burstein and A. A. Mirin. "Third Order Difference Methods for Hyperbolic Equations". In: *Journal of Computational Physics* 5.3 (June 1970), pp. 547–571. ISSN: 0021-9991. DOI: 10.1016/0021-9991(70)90080-x (cit. on p. 20).

[28] K. Chadan and P. C. Sabatier. *Inverse Problems in Quantum Scattering Theory*. 2nd ed. Springer, Berlin, Heidelberg, 1989. ISBN: 978-3-642-83319-9. DOI: 10.1007/978-3-642-83317-5 (cit. on pp. 11, 12, 88, 91–93, 97, 126).

[29] I. Chekhovskoy, S. Medvedev, I. Vaseva, E. Sedov, and M. P. Fedoruk. "Introducing phase jump tracking-a fast method for eigenvalue evaluation of the direct Zakharov-Shabat problem". In: *Communications in Nonlinear Science and Numerical Simulation* 96 (May 2021), p. 105718. DOI: 10.1016/j.cnsns.2021.105718 (cit. on pp. 4, 26, 88).

[30] S. Chimmalgi. "Improved Fast Inverse Nonlinear Fourier Transform for Multi-solitons: A Discrete Darboux Based Approach". MA thesis. Delft University of Technology, Aug. 31, 2017. URL: http://resolver.tudelft.nl/uuid:adf4d14a-fecd-400f-a1be-3e6c32d23a06 (cit. on p. 89).

[31] S. Chimmalgi, P. J. Prins, and S. Wahls. "Nonlinear Fourier Transform Algorithm Using a Higher Order Exponential Integrator". In: *Advanced Photonics 2018 (BGPP, IPR, NP, NOMA, Sensors, Networks, SPPCom, SOF)*. OSA, 2018. DOI: 10.1364/sppcom.2018.spm4g.5 (cit. on p. 147).

[32] S. Chimmalgi, P. J. Prins, and S. Wahls. "Fast Nonlinear Fourier Transform Algorithms Using Higher Order Exponential Integrators". In: *IEEE Access* 7 (2019), pp. 145161–145176. DOI: 10.1109/access.2019.2945480 (cit. on pp. 4, 26, 40, 68, 88, 147).

[33] I. Christov. "Internal solitary waves in the ocean: Analysis using the periodic, inverse scattering transform". In: *Mathematics and Computers in Simulation* 80.1 (Sept. 2009), pp. 192–201. DOI: 10.1016/j.matcom.2009.06.005 (cit. on pp. 15, 27, 28, 43, 59, 88).

[34] A. Constantin and R. I. Ivanov. "Equatorial Wave–Current Interactions". In: *Communications in Mathematical Physics* 370.1 (June 2019), pp. 1–48. DOI: 10.1007/s00220-019-03483-8 (cit. on p. 5).

[35] A. Constantin. "On the propagation of tsunami waves, with emphasis on the tsunami of 2004". In: *Discrete & Continuous Dynamical Systems - B* 12.3 (2009), pp. 525–537. DOI: 10.3934/dcdsb.2009.12.525 (cit. on p. 5).

[36] A. Constantin. "On the relevance of soliton theory to tsunami modelling". In: *Wave Motion* 46.6 (Sept. 2009), pp. 420–426. DOI: 10.1016/j.wavemoti.2009.05.002 (cit. on p. 5).

[37] D. G. Crighton. "Applications of KdV". In: *Acta Applicandae Mathematicae* 39.1-3 (June 1995), pp. 39–67. DOI: 10.1007/bf00994625 (cit. on pp. 5, 6, 25, 87).

[38] M. M. Crum. "Associated Sturm–Liouville systems". In: *The Quarterly Journal of Mathematics* 6.1 (1955), pp. 121–127. ISSN: 0033-5606. DOI: 10.1093/qmath/6.1.121 (cit. on pp. 92, 126).

[39] B. Deconinck and J. N. Kutz. "Computing spectra of linear operators using the Floquet–Fourier–Hill method". In: *Journal of Computational Physics* 219.1 (Nov. 2006), pp. 296–321. DOI: 10.1016/j.jcp.2006.03.020 (cit. on p. 27).

[40] P. Deift and E. Trubowitz. "Inverse Scattering on the Line". In: *Communications on Pure and Applied Mathematics* 32.2 (Mar. 1979), pp. 121–251. DOI: 10.1002/cpa.3160320202 (cit. on pp. 9, 11, 12, 88, 89, 92–95, 126).

[41] S. Descombes. "Convergence of a splitting method of high order for reaction-diffusion systems". In: *Mathematics of Computation* 70.236 (July 2000), pp. 1481–1502. DOI: 10.1090/s0025-5718-00-01277-1 (cit. on p. 20).

[42] M. W. Dingemans. "Water wave propagation over uneven bottoms". PhD thesis. Delft University of Technology, Nov. 1994. URL: http://resolver.tudelft.nl/uuid:67580088-62af-4c6f-b32e-b3940584e5d2 (cit. on p. 127).

[43] P. G. Drazin and R. S. Johnson. *Solitons: An introduction*. Vol. 2. Cambridge university press, 1989 (cit. on p. 6).

[44] T. A. Driscoll, N. Hale, and L. N. Trefethen, eds. *Chebfun Guide. For Chebfun version 5*. 1st ed. 2014. URL: http://www.chebfun.org/docs/guide/chebfun_guide.pdf (cit. on p. 27).

[45] B. Dubrovin. *Integrable Systems and Riemann Surfaces Lecture Notes (preliminary version)*. 2009. URL: https://people.sissa.it/~dubrovin/rsnleq_web.pdf (cit. on p. 3).

[46] W. Eckhaus and A. M. van Harten. *The Inverse Scattering Transformation and the Theory of Solitons*. Vol. 50. Elsevier, 1981. ISBN: 0 444 86166 1 (cit. on pp. 6, 26).

[47] G. El. "Korteweg – de Vries equation: solitons and undular bores". In: *Solitary Waves in Fluids, Advances in Fluid Mechanics*. Ed. by R. Grimshaw. Vol. 47. WIT Press, UK, 2007. Chap. 2, pp. 19–54 (cit. on p. 56).

[48] L. D. Faddeev. "Properties of the S-matrix of the one-dimensional Schrödinger equation". In: *American Mathematical Society Translations, Series 2* 65 (1967), pp. 139–166. DOI: 10.1090/trans2/065/04 (cit. on p. 6).

[49] R. Feced, M. N. Zervas, and M. A. Muriel. "An Efficient Inverse Scattering Algorithm for the Design of Nonuniform Fiber Bragg Gratings". In: *IEEE Journal of Quantum Electronics* 35.8 (1999), pp. 1105–1115. DOI: 10.1109/3.777209 (cit. on p. 18).

[50] L. Fermo, C. van der Mee, and S. Seatzu. "A numerical method to compute the scattering solution for the KdV equation". In: *Applied Numerical Mathematics* 149 (Mar. 2020), pp. 3–16. DOI: 10.1016/j.apnum.2019.07.001 (cit. on pp. 4, 26).

[51] M. Á. García-Ferrero and D. Gómez-Ullate. "Oscillation Theorems for the Wronskian of an Arbitrary Sequence of Eigenfunctions of Schrödinger's Equation". In: *Letters in Mathematical Physics* 105.4 (Feb. 2015), pp. 551–573. DOI: 10.1007/s11005-015-0751-4 (cit. on p. 97).

[52] C. S. Gardner and G. Morikawa. "Similarity in the asymptotic behavior of collision-free hydromagnetic waves and water waves". 1960 (cit. on p. 6).

[53] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura. "Method for solving the Korteweg-deVries equation". In: *Physical Review Letters* 19.19 (Nov. 1967), pp. 1095–1097. DOI: 10.1103/physrevlett.19.1095 (cit. on pp. 2, 3, 8, 9, 15, 26, 62, 88, 91).

[54] A. A. Gelash and D. S. Agafontsev. "Strongly interacting soliton gas and formation of rogue waves". In: *Physical Review E* 98.4 (Oct. 2018). DOI: 10.1103/physreve.98.042210 (cit. on p. 89).

[55] A. Gelash and R. Mullyadzhanov. "Anomalous errors of direct scattering transform". In: *Physical Review E* 101.5 (May 2020). DOI: 10.1103/physreve.101.052206 (cit. on pp. 61, 66).

[56] F. Gesztesy, B. Simon, and G. Teschl. "Spectral deformations of one-dimensional Schrödinger operators". In: *Journal d'Analyse Mathématique* 70.1 (Dec. 1996), pp. 267–324. DOI: 10.1007/bf02820446 (cit. on p. 98).

[57] A. Geyer and R. Quirchmayr. "Shallow water equations for equatorial tsunami waves". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2111 (Dec. 2017), p. 20170100. DOI: 10.1098/rsta.2017.0100 (cit. on p. 5).

[58] D. Goldman and T. J. Kaper. "Nth-Order Operator Splitting Schemes and Nonreversible Systems". In: *SIAM Journal on Numerical Analysis* 33.1 (Feb. 1996), pp. 349–367. DOI: 10.1137/0733018 (cit. on p. 16).

[59] C. Gu, H. Hu, and Z. Zhou. *Darboux Transformations in Integrable Systems. Theory and their Applications to Geometry*. Vol. 26. Mathematical Physics Studies. Springer Science & Business Media, 2005. ISBN: 1-4020-3088-6. DOI: 10.1007/1-4020-3088-6 (cit. on pp. 92, 93, 97, 126).

[60] A. Gudko, A. Gelash, and R. Mullyadzhanov. "High-order numerical method for scattering data of the Korteweg—De Vries equation". In: *Journal of Physics: Conference Series*. Vol. 1677. IOP Publishing. IOP Publishing, Nov. 2020, p. 012011. DOI: 10.1088/1742-6596/1677/1/012011 (cit. on pp. 4, 26).

[61] J. L. Hammack and H. Segur. "The Korteweg–de Vries equation and water waves. Part 2. Comparison with experiments". In: *Journal of Fluid Mechanics* 65.2 (Aug. 1974), pp. 289–314. DOI: 10.1017/s002211207400139x (cit. on pp. 5, 25, 60, 87, 88).

[62] J. L. Hammack and H. Segur. "The Korteweg–de Vries equation and water waves. Part 3. Oscillatory waves". In: *Journal of Fluid Mechanics* 84.2 (Jan. 1978), pp. 337–358. DOI: 10.1017/s0022112078000208 (cit. on pp. 5, 60).

[63] S. Hari and F. R. Kschischang. "Bi-Directional Algorithm for Computing Discrete Spectral Amplitudes in the NFT". In: *Journal of Lightwave Technology* 34.15 (Aug. 2016), pp. 3529–3537. DOI: 10.1109/jlt.2016.2577702 (cit. on pp. 11, 61, 66–68, 70, 85, 124).

[64] L. Hattam. "KdV cnoidal waves in a traffic flow model with periodic boundaries". In: *Physica D: Nonlinear Phenomena* 348 (June 2017), pp. 44–53. DOI: 10.1016/j.physd.2017.02.010 (cit. on pp. 6, 25).

[65] K. R. Helfrich and J. A. Whitehead. "Solitary waves on conduits of buoyant fluid in a more viscous fluid". In: *Geophysical & Astrophysical Fluid Dynamics* 51.1-4 (1990), pp. 35–52. DOI: 10.1080/03091929008219850 (cit. on p. 5).

[66] N. J. Higham. *Accuracy and stability of numerical algorithms*. 2nd ed. Vol. 80. SIAM, Jan. 2002. ISBN: 0-89871-521-0. DOI: 10.1137/1.9780898718027 (cit. on pp. 97, 99, 100).

[67] G. W. Hill. "On the part of the motion of the lunar perigee which is a function of the mean motions of the sun and moon". In: *Acta mathematica* 8.1 (1886), pp. 1–36. DOI: 10.1007/BF02417081 (cit. on p. 27).

[68] J. E. Huss and J. A. Pennline. *A comparison of five benchmarks*. Technical Memorandum 19870008008. Cleveland, OH, United States: NASA Lewis Research Center, Feb. 1, 1987 (cit. on pp. 38, 39).

[69] L. Ixaru, H. D. Meyer, and G. V. Berghe. "CP methods for the Schrödinger equation revisited". In: *Journal of Computational and Applied Mathematics* 88.2 (Mar. 1997), pp. 289–314. DOI: 10.1016/s0377-0427(97)00218-5 (cit. on pp. 27, 28, 36, 43).

[70] A. Jeffrey and T. Kakutani. "Weak Nonlinear Dispersive Waves: A Discussion Centered Around the Korteweg–De Vries Equation". In: *SIAM Review* 14.4 (Oct. 1972), pp. 582–643. DOI: 10.1137/1014101 (cit. on p. 15).

[71] V. Karpman and E. Maslov. "Perturbation theory for solitons". In: *JETP* 73 (1977), pp. 537–559 (cit. on p. 105).

[72] V. I. Karpman. *Non-Linear Waves in Dispersive Media: International Series of Monographs in Natural Philosophy*. Vol. 71. Elsevier, 1974 (cit. on p. 6).

[73] A. Kasman. *Glimpses of soliton theory: the algebra and geometry of nonlinear PDEs*. Vol. 54. American Mathematical Soc., 2010. ISBN: 978-0-8218-5245-3 (cit. on p. 97).

[74] A. Kasman. "A Brief History of Solitons and the KDV Equation". In: *Current Science* 115.8 (Oct. 2018), p. 1486. DOI: 10.18520/cs/v115/i8/1486-1496 (cit. on pp. 2, 4).

[75] D. J. Kaup. "A Higher-Order Water-Wave Equation and the Method for Solving It". In: *Progress of Theoretical Physics* 54.2 (Aug. 1975), pp. 396–408. DOI: 10.1143/ptp.54.396 (cit. on p. 127).

[76] D. J. Korteweg and G. de Vries. "On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science: Series 5* 39.240 (May 1895), pp. 422–443. DOI: 10.1080/14786449508620739 (cit. on pp. 2, 25, 59, 87).

[77] B. Kupershmidt. "Mathematics of dispersive water waves". In: *Communications in Mathematical Physics* 99.1 (1985), pp. 51–73. DOI: 10.1007/BF01466593 (cit. on p. 127).

[78] V. Kuznetsov, V. Nakoryakov, B. Pokusaev, and I. Shreiber. "Propagation of perturbations in a gas-liquid mixture". In: *Journal of Fluid Mechanics* 85.1 (1978), pp. 85–96. DOI: doi.org/10.1017/s0022112078000543 (cit. on p. 5).

[79] G. L. Lamb. *Elements of soliton theory*. Wiley, 1980. ISBN: 0-471-04559-4 (cit. on pp. 5, 6, 8, 21, 30, 49, 63, 77, 91, 107).

[80] D. Lannes. "Modeling shallow water waves". In: *Nonlinearity* 33.5 (Mar. 2020), R1–R57. DOI: 10.1088/1361-6544/ab6c7c (cit. on p. 87).

[81] P. D. Lax. "Integrals of nonlinear equations of evolution and solitary waves". In: *Communications on Pure and Applied Mathematics* 21.5 (1968), pp. 467–490. DOI: 10.1002/cpa.3160210503 (cit. on pp. 3, 16, 26, 62, 88).

[82] V. Ledoux. "Study of Special Algorithms for solving Sturm-Liouville and Schrödinger Equations". PhD thesis. Ghent University, 2007. DOI: 1854/6145 (cit. on pp. 27, 31, 123, 124).

[83] V. Ledoux, M. V. Daele, and G. V. Berghe. "Matslise: A matlab package for the numerical solution of Sturm-Liouville and Schrödinger equations". In: *ACM Transactions on Mathematical Software (TOMS)* 31.4 (2005), pp. 532–554. DOI: 10.1145/1114268.1114273 (cit. on p. 43).

[84] V. Ledoux and M. Van Daele. "Solution of Sturm–Liouville problems using modified Neumann schemes". In: *SIAM Journal on Scientific Computing* 32.2 (Jan. 2010), pp. 563–584. DOI: 10.1137/090758398 (cit. on pp. 36, 41).

[85] V. Ledoux and M. Van Daele. "MATSLISE 2.0: A Matlab Toolbox for Sturm-Liouville Computations". In: *ACM Transactions on Mathematical Software (TOMS)* 42.4 (2016), pp. 1–18. DOI: 10.1145/2839299 (cit. on pp. 28, 29, 43, 47, 53, 56).

[86] D. Levi, O. Ragnisco, and A. Sym. "Dressing method vs. classical Darboux transformation". In: *Il Nuovo Cimento B Series 11* 83.1 (Sept. 11, 1984), pp. 34–42. DOI: 10.1007/bf02723762 (cit. on p. 94).

[87] B. M. Levitan. *Inverse Sturm-Liouville Problems*. VNU Science press, Utrecht, Dec. 1, 1987. ISBN: 978-3-11-046031-5. DOI: 10.1515/9783110941937 (cit. on p. 98).

[88] P. A. Madsen, D. R. Fuhrman, and H. A. Schäffer. "On the solitary wave paradigm for tsunamis". In: *Journal of Geophysical Research: Oceans* 113.C12 (Dec. 2008). DOI: 10.1029/2008jc004932 (cit. on p. 5).

[89] V. B. Matveev and M. A. Salle. *Darboux transformations and solitons*. Springer-Verlag Berlin Heidelberg New York London Paris Tokyo Hong Kong Barcelona Budapest, 1991. ISBN: 3-540-50660-8. DOI: 10.1007/978-3-662-00922-2 (cit. on pp. 92, 93).

[90] S. Medvedev, I. Chekhovskoy, I. Vaseva, and M. Fedoruk. *Fast Computation of the Direct Scattering Transform by Fourth Order Conservative Multi-Exponential Scheme*. Sept. 29, 2019. arXiv: 1909.13228v1 [math.NA] (cit. on pp. 4, 26).

[91] S. Medvedev, I. Chekhovskoy, I. Vaseva, and M. Fedoruk. "Fast sixth-order algorithm based on the generalized Cayley transform for the Zakharov-Shabat system associated with nonlinear Schrodinger equation". In: *Journal of Computational Physics* (Oct. 2021), p. 110764. DOI: 10.1016/j.jcp.2021.110764 (cit. on pp. 4, 26).

[92] S. Medvedev, I. Vaseva, I. Chekhovskoy, and M. Fedoruk. "Exponential fourth order schemes for direct Zakharov-Shabat problem". In: *Optics Express* 28.1 (Dec. 2019), p. 20. DOI: 10.1364/OE.377140 (cit. on pp. 4, 26).

[93] S. Medvedev, I. Vaseva, I. Chekhovskoy, and M. Fedoruk. "Numerical algorithm with fourth-order accuracy for the direct Zakharov-Shabat problem". In: *Optics Letters* 44.9 (Apr. 2019), pp. 2264–2267. DOI: 10.1364/ol.44.002264 (cit. on p. 68).

[94] J. W. Miles. "The Korteweg-de Vries equation: a historical essay". In: *Journal of fluid mechanics* 106 (1981), pp. 131–147. DOI: 10.1017/S0022112081001559 (cit. on p. 2).

[95] J. Misra and M. Patra. "A study of solitary waves in a tapered aorta by using the theory of solitons". In: *Computers & Mathematics with Applications* 54.2 (July 2007), pp. 242–254. DOI: 10.1016/j.camwa.2006.12.025 (cit. on pp. 25, 87).

[96] M. Muramatsu and T. Nagatani. "Soliton and kink jams in traffic flow with open boundaries". In: *Physical Review E* 60.1 (July 1999), pp. 180–187. DOI: 10.1103/physreve.60.180 (cit. on p. 6).

[97] NAG Library Code Contributors. *NAG Library Routine Document d02kdf*. Tech. rep. The Numerical Algorithms Group Ltd, Oxford, UK, 2017. URL: https://www.nag.com/numeric/fl/nagdoc_fl26/pdf/d02/d02kdf.pdf (cit. on p. 27).

[98] V. E. Nakoryakov, B. G. Pokusaev, and I. R. Shreiber. *Wave propagation in gas-liquid media*. CRC Press, 1993 (cit. on p. 5).

[99] G. Neugebauer and R. Meinel. "General N-soliton solution of the AKNS class on arbitrary background". In: *Physics Letters A* 100.9 (Feb. 1984), pp. 467–470. DOI: 10.1016/0375-9601(84)90827-2 (cit. on pp. 72, 86, 89, 105, 126).

[100] S. Novikov, S. Manakov, L. Pitaevskii, and V. Zakharov. *Theory of Solitons: The Inverse Scattering Method*. Springer Science & Business Media, 1984 (cit. on p. 6).

[101] A. R. Osborne. "Non-linear Fourier Analysis for the Infinite-Interval Korteweg–de Vries equation I: An algorithm for the Direct Scattering Transform". In: *Journal of Computational Physics* 94 (June 1991), pp. 284–313. DOI: 10.1016/0021-9991(91)90223-8 (cit. on pp. 4, 17, 22, 26, 60, 61, 68, 84, 88, 107, 121).

[102] A. R. Osborne. "Automatic algorithm for the numerical inverse scattering transform of the Korteweg–de Vries equation". In: *Mathematics and Computers in Simulation* 37.4 (1994), pp. 431–450. ISSN: 0378-4754. DOI: 10.1016/0378-4754(94)00029-8 (cit. on pp. 4, 26–29, 34–36, 42, 47, 54, 57, 124).

[103] A. R. Osborne and M. Petti. "Numerical inverse-scattering-transform analysis of laboratory-generated surface wave trains". In: *Physical Review E* 47.2 (Feb. 1993), pp. 1035–1037. DOI: 10.1103/physreve.47.1035 (cit. on pp. 4, 26).

[104] A. R. Osborne and M. Petti. "Laboratory-generated, shallow-water surface waves: Analysis using the periodic, inverse scattering transform". In: *Physics of Fluids* 6.5 (May 1994), pp. 1727–1744. DOI: 10.1063/1.868235 (cit. on pp. 4, 5, 26, 38).

[105] A. R. Osborne, E. Segre, G. Boffetta, and L. Cavaleri. "Soliton basis states in shallow-water ocean surface waves". In: *Physical Review Letters* 67.5 (July 1991), pp. 592–595. DOI: 10.1103/physrevlett.67.592 (cit. on pp. 4, 26).

[106] A. R. Osborne. *Nonlinear ocean waves and the inverse scattering transform*. 1st ed. Vol. 97. International Geophysics. Academic Press, 2010. ISBN: 978-0-12-528629-9 (cit. on pp. 15, 25, 60, 83).

[107] A. R. Osborne. "Nonlinear Fourier Methods for Ocean Waves". In: *Procedia IUTAM* 26 (2018), pp. 112–123. DOI: 10.1016/j.piutam.2018.03.011 (cit. on p. 60).

[108] A. Osborne, M. Petti, G. Liberatore, and L. Cavaleri. "Nonlinear Fourier analysis of laboratory generated, broad-banded surface waves". In: *Procs. Intl. Conference on Computer Modelling in Ocean Engng., held Venice, 19-23 Sept. 1988*. 1988, pp. 99–105 (cit. on pp. 4, 5, 26).

[109] J. Pöschel and E. Trubowitz. *Inverse Spectral Theory*. Pure and Applied Mathematics. Elsevier, 1987. ISBN: 978-0-12-563040-5. DOI: 10.1016/s0079-8169(08)x6138-0 (cit. on p. 91).

[110] P. J. Prins and S. Wahls. "Higher Order Exponential Splittings for the Fast Non-Linear Fourier Transform of the Korteweg-De Vries Equation". In: *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2018, pp. 4524–4528. DOI: 10.1109/ICASSP.2018.8461708 (cit. on pp. 4, 15, 18, 24, 26, 60, 83, 90, 147).

[111] P. J. Prins and S. Wahls. "Soliton Phase Shift Calculation for the Korteweg–De Vries Equation". In: *IEEE Access* 7 (1 Sept. 12, 2019), pp. 122914–122930. ISSN: 2169-3536. DOI: 10.1109/access.2019.2932256 (cit. on pp. 4, 26, 59, 88–91, 94, 100, 105, 108, 112, 113, 147).

[112] P. J. Prins and S. Wahls. "An accurate $O(N^2)$ floating point algorithm for the Crum transform of the KdV equation". In: *Communications in Nonlinear Science and Numerical Simulation* 102.105782 (Nov. 2021), pp. 1–25. ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2021.105782 (cit. on pp. 4, 26, 49, 87, 147).

[113] P. J. Prins and S. Wahls. "Reliable computation of the eigenvalues of the discrete KdV spectrum". Manuscript submitted for publication. (cit. on pp. 25, 147).

[114] A. Provenzale and A. R. Osborne. "Nonlinear Fourier analysis for the infinite-interval Korteweg–de Vries equation II : Numerical tests of the direct scattering transform". In: *Journal of Computational Physics* 94.2 (June 1991), pp. 314–351. DOI: 10.1016/0021-9991(91)90224-9 (cit. on pp. 4, 26, 60–62, 84, 121).

[115] S. Pruess and C. T. Fulton. "Mathematical software for Sturm-Liouville problems". In: *ACM Transactions on Mathematical Software (TOMS)* 19.3 (1993), pp. 360–376. DOI: 10.1145/155743.155791 (cit. on pp. 27, 28, 36).

[116] H. Prüfer. "Neue Herleitung der Sturm-Liouvilleschen Reihenentwicklung stetiger Funktionen". In: *Mathematische Annalen* 95.1 (Dec. 1926), pp. 499–518. DOI: 10.1007/bf01206624 (cit. on p. 27).

[117] L. Redekopp. "On the theory of solitary Rossby waves". In: *Journal of Fluid Mechanics* 82.4 (1977), pp. 725–745. DOI: 10.1017/s0022112077000950 (cit. on p. 6).

[118] I. Redor, E. Barthélemy, H. Michallet, M. Onorato, and N. Mordant. "Experimental evidence of a hydrodynamic soliton gas". In: *Physical Review Letters* 122.21 (2019), p. 214502. DOI: 10.1103/PhysRevLett.122.214502 (cit. on p. 127).

[119] D. S. Ricketts and D. Ham. *Electrical solitons. Theory, Design, and Applications*. CRC Press, 2011. ISBN: 978-1-4398-2980-6. DOI: 10.1201/9781315217802 (cit. on pp. 5, 25, 87).

[120] J. S. Russell. "Report on waves". In: *Report of the fourteenth meeting of the British association for the advancement of science*. John Murray, London, 1845, pp. 311–390 (cit. on pp. 1, 5).

[121] R. L. Sachs. "On the integrable variant of the Boussinesq system: Painlevé property, rational solutions, a related many-body system, and equivalence with the AKNS hierarchy". In: *Physica D: Nonlinear Phenomena* 30.1-2 (Feb. 1988), pp. 1–27. DOI: 10.1016/0167-2789(88)90095-4 (cit. on p. 127).

[122] P. Sacks and J. Shin. "Computational methods for some inverse scattering problems". In: *Applied mathematics and computation* 207.1 (Jan. 2009), pp. 111–123. DOI: 10.1016/j.amc.2008.10.033 (cit. on pp. 11, 12, 88, 89, 92, 93, 125, 126).

[123] B. F. Samsonov. "New possibilities for supersymmetry breakdown in quantum mechanics and second-order irreducible Darboux transformations". In: *Physics Letters A* 263.4-6 (Dec. 1999), pp. 274–280. DOI: 10.1016/s0375-9601(99)00736-7 (cit. on p. 98).

[124] J. Satsuma and N. Yajima. "Initial Value Problems of One-Dimensional Self-Modulation of Nonlinear Waves in Dispersive Media". In: *Progress of Theoretical Physics Supplement* 55 (1974), pp. 284–306. DOI: 10.1143/ptps.55.284 (cit. on p. 23).

[125] P. Schuur. "Multisoliton phase shifts in the case of a nonzero reflection coefficient". In: *Physics Letters A* 102.9 (June 1984), pp. 387–392. DOI: 10.1016/0375-9601(84)91061-2 (cit. on pp. 60, 64).

[126] P. C. Schuur. *Asymptotic analysis of soliton problems: an inverse scattering approach*. Ed. by A. Dold and B. Eckmann. Vol. 1232. Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg, 1986. ISBN: 0-387-17203-3 (cit. on pp. 60, 64).

[127] H. Segur. "The Korteweg-de Vries equation and water waves. Solutions of the equation. Part 1". In: *Journal of Fluid Mechanics* 59.4 (Aug. 1973), pp. 721–736. DOI: 10.1017/s0022112073001813 (cit. on pp. 12, 30, 62, 64).

[128] H. Segur. *Solitons and the Inverse Scattering Transform*. Tech. rep. New Jersey, U.S.A.: Aeronautical Research Associates of Princeton, inc., Jan. 1980. DOI: 10.21236/ada088023 (cit. on p. 6).

[129] H. Segur and J. Hammack. "Soliton models of long internal waves". In: *Journal of Fluid Mechanics* 118 (1982), pp. 285–304. DOI: 10.1017/s0022112082001086 (cit. on p. 5).

[130] Q. Sheng. "Solving Linear Partial Differential Equations by Exponential Splitting". In: *IMA Journal of Numerical Analysis* 9.2 (1989), pp. 199–212. DOI: 10.1093/imanum/9.2.199 (cit. on p. 16).

[131] N. van der Sluis. *Hoofdletters in namen: Nynke van der Sluis / Nynke Van der Sluis*. May 2011. URL: https://onzetaal.nl/taaladvies/hoofdletters-namen-nederland (visited on 07/16/2021) (cit. on p. 2).

[132] A. Span, V. Aref, H. Bülow, and S. ten Brink. "Successive Eigenvalue Removal for Multi-Soliton Spectral Amplitude Estimation". In: *Journal of Lightwave Technology* (May 11, 2020). ISSN: 1558-2213. DOI: 10.1109/jlt.2020.2994156 (cit. on pp. 4, 26, 88).

[133] G. Strang. "Accurate Partial Difference Methods I: Linear Cauchy Problems". In: *Archive for Rational Mechanics and Analysis* 12.1 (Jan. 1963), pp. 392–402. DOI: 10.1007/bf00281235 (cit. on p. 16).

[134] G. Strang. "On the construction and comparison of difference schemes". In: *SIAM Journal on Numerical Analysis* 5.3 (1968), pp. 506–517. DOI: 10.1137/0705041 (cit. on pp. 16, 20).

[135] S. Sugavanam, M. K. Kopae, J. Peng, J. E. Prilepsky, and S. K. Turitsyn. "Analysis of laser radiation using the Nonlinear Fourier transform". In: *Nature communications* 10.1 (2019), pp. 1–10. DOI: 10.1038/s41467-019-13265-4 (cit. on pp. 4, 26).

[136] M. Suzuki. "General theory of fractal path integrals with applications to many-body theories and statistical physics". In: *Journal of Mathematical Physics* 32.2 (Feb. 1991), pp. 400–407. DOI: 10.1063/1.529425 (cit. on p. 16).

[137] V. V. Temnov, C. Klieber, K. A. Nelson, T. Thomay, V. Knittel, A. Leitenstorfer, D. Makarov, M. Albrecht, and R. Bratschitsch. "Femtosecond nonlinear ultrasonics in gold probed with ultrashort surface plasmons". In: *Nature Communications* 4.1 (Feb. 2013). Art. no. 1468. DOI: 10.1038/ncomms2480 (cit. on pp. 6, 25, 87).

[138] L. N. Trefethen. "Is Gauss Quadrature Better than Clenshaw–Curtis?" In: *SIAM Review* 50.1 (Jan. 2008), pp. 67–87. DOI: 10.1137/060659831 (cit. on p. 109).

[139] S. Trillo, G. Deng, G. Biondini, M. Klein, G. Clauss, A. Chabchoub, and M. Onorato. "Experimental Observation and Theoretical Description of Multisoliton Fission in Shallow Water". In: *Physical Review Letters* 117.14 (Sept. 2016). DOI: 10.1103/physrevlett.117.144102 (cit. on p. 5).

[140] T. Trogdon and S. Olver. "Numerical inverse scattering for the focusing and defocusing nonlinear Schrödinger equations". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 469.2149 (Nov. 2012), p. 20120330. DOI: 10.1098/rspa.2012.0330 (cit. on pp. 4, 26).

[141] T. Trogdon. "Scattering and inverse scattering for the AKNS system: A rational function approach". In: *Studies in Applied Mathematics* 147.4 (2021), pp. 1443–1480. DOI: 10.1111/sapm.12434 (cit. on pp. 88, 126).

[142] T. Trogdon and S. Olver. *Riemann–Hilbert Problems, Their Numerical Solution, and the Computation of Nonlinear Special Functions*. Society for Industrial and Applied Mathematics, Dec. 2015. ISBN: 978-1-611974-19-5. DOI: 10.1137/1.9781611974201 (cit. on pp. 27, 61, 68, 71, 88, 92, 125, 126).

[143] T. Trogdon, S. Olver, and B. Deconinck. "Numerical inverse scattering for the Korteweg–de Vries and modified Korteweg–de Vries equations". In: *Physica D: Nonlinear Phenomena* 241.11 (June 2012), pp. 1003–1025. DOI: 10.1016/j.physd.2012.02.016 (cit. on pp. 4, 26, 27, 68, 88, 92, 125).

[144] S. K. Turitsyn, I. S. Chekhovskoy, and M. P. Fedoruk. "Nonlinear Fourier transform for analysis of optical spectral combs". In: *Physical Review E* 103.2 (Feb. 2021). DOI: 10.1103/physreve.103.1020202 (cit. on pp. 4, 26).

[145] S. K. Turitsyn, J. E. Prilepsky, S. T. Le, S. Wahls, L. L. Frumin, M. Kamalian, and S. A. Derevyanko. "Nonlinear Fourier transform for optical data processing and transmission: advances and perspectives". In: *Optica* 4.3 (Feb. 2017), p. 307. DOI: 10.1364/optica.4.000307 (cit. on p. 16).

[146] C. van der Mee, S. Seatzu, and D. Theis. "Structured matrix algorithms for inverse scattering on the line". In: *Calcolo* 44.2 (June 2007), pp. 59–87. DOI: 10.1007/s10092-007-0129-9 (cit. on p. 92).

[147] A. Vasylchenkova. "Nonlinear Fourier Transform in application to long-haul optical communications". PhD thesis. Aston University, 2020 (cit. on p. 88).

[148] A. Vasylchenkova, D. Salnikov, D. Karaman, O. G. Vasylchenkov, and J. E. Prilepskiy. "Fixed-point realization of fast nonlinear Fourier transform algorithm for FPGA implementation of optical data processing". In: *Nonlinear Optics and Applications XII*. Ed. by M. Bertolotti, A. V. Zayats, and A. M. Zheltikov. Vol. 11770. International Society for Optics and Photonics. SPIE, 2021, pp. 111–120. DOI: 10.1117/12.2588735 (cit. on pp. 4, 26).

[149] H. D. Wahlquist and F. B. Estabrook. "Bäcklund Transformation for Solutions of the Korteweg–de Vries Equation". In: *Physical Review Letters* 31.23 (Dec. 1973), pp. 1386–1390. DOI: 10.1103/physrevlett.31.1386 (cit. on p. 98).

[150] S. Wahls and H. V. Poor. "Introducing the fast nonlinear Fourier transform". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. May 2013, pp. 5780–5784. DOI: 10.1109/ICASSP.2013.6638772 (cit. on pp. 4, 10, 11, 16, 18, 26, 27, 122).

[151] S. Wahls and H. V. Poor. "Fast Numerical Nonlinear Fourier Transforms". In: *IEEE Transactions on Information Theory* 61.12 (Dec. 2015), pp. 6957–6974. ISSN: 0018-9448. DOI: 10.1109/TIT.2015.2485944 (cit. on pp. 4, 18, 26).

[152] S. Wahls, S. Chimmalgi, and P. J. Prins. "FNFT: A Software Library for Computing Nonlinear Fourier Transforms". In: *Journal of Open Source Software* 3.23 (Mar. 2018), p. 597. DOI: 10.21105/joss.00597 (cit. on pp. 4, 26, 70, 80, 88, 108, 122, 123, 125, 147).

[153] S. Wahls, S. Chimmalgi, and P. J. Prins. "Wiener-Hopf Method for b-Modulation". In: *Optical Fiber Communication Conference (OFC) 2019*. OSA, 2019. DOI: 10.1364/ofc.2019.w2a.50 (cit. on p. 147).

[154] H. Washimi and T. Taniuti. "Propagation of Ion-Acoustic Solitary Waves of Small Amplitude". In: *Physical Review Letters* 17.19 (Nov. 1966), pp. 996–998. DOI: 10.1103/PhysRevLett.17.996 (cit. on pp. 6, 15).

[155] L. V. Wijngaarden. "Evolving Solitons in Bubbly Flows". In: *KdV '95*. Springer Netherlands, 1995, pp. 507–516. DOI: 10.1007/978-94-011-0017-5_29 (cit. on pp. 5, 25).

[156] J. Yang. *Nonlinear Waves in Integrable and Nonintegrable Systems*. Philadelphia, PA, USA: SIAM, 2010. DOI: 10.1137/1.9780898719680 (cit. on p. 68).

[157] W. Yang and A. C. Peet. "The collocation method for bound solutions of the Schrödinger equation". In: *Chemical physics letters* 153.1 (1988), pp. 98–104. DOI: 10.1016/0009-2614(88)80139-8 (cit. on p. 27).

[158] S. Yomosa. "Solitary Waves in Large Blood Vessels". In: *Journal of the Physical Society of Japan* 56.2 (Feb. 1987), pp. 506–520. DOI: 10.1143/jpsj.56.506 (cit. on p. 5).

[159] M. I. Yousefi and F. R. Kschischang. "Information transmission using the nonlinear Fourier transform, Part III: Spectrum modulation". In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 4346–4369. DOI: 10.1109/TIT.2014.2321155 (cit. on p. 88).

[160] M. I. Yousefi and F. R. Kschischang. "Information Transmission Using the Nonlinear Fourier Transform, Part II: Numerical Methods". In: *IEEE Transactions on Information Theory* 60.7 (July 2014), pp. 4329–4345. DOI: 10.1109/tit.2014.2321151 (cit. on pp. 4, 26, 27).

[161] N. J. Zabusky and C. J. Galvin. "Shallow-water waves, the Korteweg-deVries equation and solitons". In: *Journal of Fluid Mechanics* 47.04 (June 1971), p. 811. DOI: 10.1017/s0022112071001393 (cit. on p. 15).

[162] N. J. Zabusky and M. D. Kruskal. "Interaction of "Solitons" in a Collisionless Plasma and the Recurrence of Initial States". In: *Physical Review Letters* 15.6 (Aug. 1965), pp. 240–243. DOI: 10.1103/physrevlett.15.240 (cit. on p. 2).

[163] V. E. Zakharov and A. B. Shabat. "Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media". In: *Soviet physics JETP* 34.1 (1972), p. 62 (cit. on pp. 4, 16, 26).

[164] V. E. Zakharov and L. D. Faddeev. "Korteweg-de Vries equation: A completely integrable Hamiltonian system". In: *Functional Analysis and Its Applications* 5.4 (1972), pp. 280–287. DOI: 10.1007/bf01086739 (cit. on pp. 56, 106).

[165] A. Zettl. *Sturm-Liouville theory*. Mathematical surveys and monographs 121. American Mathematical Soc., 2005. ISBN: 978-1-4704-1348-4. DOI: 10.1090/surv/121 (cit. on pp. 31, 97).

[166] W. Zimmerman and G. Haarlemmer. "Internal gravity waves: Analysis using the periodic, inverse scattering transform". In: *Nonlinear Processes in Geophysics* 6.1 (1999), pp. 11–26. DOI: 10.5194/npg-6-11-1999 (cit. on pp. 4, 5, 25, 26).

Acronyms

AKNS	Ablowitz–Kaup–Newell–Segur.
CPU	Central Processing Unit.
FFT	Fast Fourier Transform.
FLOP	FLoating point OPeration.
FNFT	Fast Non-linear Fourier Transform.
GLM	Gel'fand–Levitan–Marchenko.
GNU	GNU's not Unix.
GPL	General Public Licence.
IST	Inverse Scattering Transform.
KB	Kaup–Broer equation.
KdV	Korteweg–de Vries equation.
NaN	Not a Number.
NFT	Non-linear Fourier Transform.
NR	Newton–Raphson.
NSE	Non-linear Schrödinger Equation.
ODE	Ordinary Differential Equation.
PDE	Partial Differential Equation.
SL	Sturm–Liouville.
ZS	Zakharov–Shabat.

Acknowledgements

Ah, here you are! I didn't expect you to reach this chapter so soon. Did you like this dissertation so far? – Although my name is on the cover I'm not the only one to take the credits for it.

First of all I want to thank my promotors: Sander Wahls for acquiring the funding, for giving me the opportunity to pursue a PhD at the TU Delft and for his patient guidance and thorough feedback; Michel Verhaegen for his trust and support and for providing a pleasant working atmosphere. Furthermore I want to thank the other members of the doctoral committee for their time invested in reviewing this dissertation and for their helpful comments: prof.dr. El, prof.dr. Deconinck, prof.dr. Thijssen, prof.dr.ir. Heemink and dr. Mazo Espinosa.

During my first year of research I was very lucky to get an office mate who became like an informal mentor to me, with a remarkable talent for cheering me up by putting setbacks in perspective. Thank you very much, Elisabeth. Later I shared an office with Shrinivas, who contributed to this dissertation with several sparring sessions about our research topic. Although his library seemed disorganised to me, he could always remember exactly which reference contained the needed information. I'm also thankful to Markus, for proofreading a draft of the paper on which Chapter 4 is based.

Besides research an important and structure providing part of my schedule consisted of meetings to promote interdisciplinary cooperation and reflection on the world in a broad sense, otherwise known as coffee and lunch breaks. I'm grateful to all participants in different periods and frequencies: Yasin, Wiegert, Vinod, Tijmen, Thea, Sina, Shrinivas, Roger, Reinier, Pieter, Pascal, Paolo, Niloofar, Nikolaos, Markus, Maarten, Lukasz, Leonoor, Laurens, Katerina, Jacques, Jacopo, Iurie, Hai, Huichen, Elisabeth, Dean, Coen, Barbara, Baptiste.

This dissertation builds on the hard and valuable work of other scientists before me. Without the slightest doubt, I would not have been able to write this dissertation without the many authors listed in the bibliography and many more on which their work is based. In another sense I would not have been able to write this dissertation without the work of Donald E. Knuth (TEX), Leslie Lamport (LATEX), and many package contributors after them.

I want to thank Gürol and Miya for their work and attitude that provided me a valuable learning experience and I want to thank the trainers who work for the graduate school for changing my view on transferable skills.

Finally I'm grateful to my family for their support and belief in me. That concerns in particular Rianne, whose unconditional love and encouragement made this journey possible. I feel privileged to have her by my side.

Peter Prins
Leiden, March 2022

List of Publications

9. P. J. Prins and S. Wahls. "Reliable computation of the eigenvalues of the discrete KdV spectrum". Manuscript submitted for publication.

8. M. Brühl, P. J. Prins, S. Ujvary, I. Barranco, S. Wahls, and P. L.-F. Liu. "Comparative analysis of bore propagation using conventional linear and KdV-based nonlinear Fourier transform". In: *Wave Motion* 111 (2022), p. 102905. ISSN: 0165-2125. DOI: 10.1016/j.wavemoti.2022.102905

7. P. J. Prins and S. Wahls. "An accurate $O(N^2)$ floating point algorithm for the Crum transform of the KdV equation". In: *Communications in Nonlinear Science and Numerical Simulation* 102.105782 (Nov. 2021), pp. 1–25. ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2021.105782

6. S. Chimmalgi, P. J. Prins, and S. Wahls. "Fast Nonlinear Fourier Transform Algorithms Using Higher Order Exponential Integrators". In: *IEEE Access* 7 (2019), pp. 145161–145176. DOI: 10.1109/access.2019.2945480

5. P. J. Prins and S. Wahls. "Soliton Phase Shift Calculation for the Korteweg–De Vries Equation". In: *IEEE Access* 7 (1 Sept. 12, 2019), pp. 122914–122930. ISSN: 2169-3536. DOI: 10.1109/access.2019.2932256

4. S. Wahls, S. Chimmalgi, and P. J. Prins. "Wiener-Hopf Method for b-Modulation". In: *Optical Fiber Communication Conference (OFC) 2019*. OSA, 2019. DOI: 10.1364/ofc.2019.w2a.50

3. S. Chimmalgi, P. J. Prins, and S. Wahls. "Nonlinear Fourier Transform Algorithm Using a Higher Order Exponential Integrator". In: *Advanced Photonics 2018 (BGPP, IPR, NP, NOMA, Sensors, Networks, SPPCom, SOF)*. OSA, 2018. DOI: 10.1364/sppcom.2018.spm4g.5

2. S. Wahls, S. Chimmalgi, and P. J. Prins. "FNFT: A Software Library for Computing Nonlinear Fourier Transforms". In: *Journal of Open Source Software* 3.23 (Mar. 2018), p. 597. DOI: 10.21105/joss.00597

1. P. J. Prins and S. Wahls. "Higher Order Exponential Splittings for the Fast Non-Linear Fourier Transform of the Korteweg-De Vries Equation". In: *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2018, pp. 4524–4528. DOI: 10.1109/ICASSP.2018.8461708