AN APPLICATION OF ORTHOGONAL POLYNOMIALS IN NONLINEAR WAVE THEORY

EVAN S. ALLHANDS

ABSTRACT. We attempt to reproduce the results from a recent paper entitled "Efficient method for calculating the eigenvalues of the Zakharov-Shabat system" by S. Cui and Z. Wang [9]. The relevant theory of orthogonal polynomials, and their application to approximation theory, is presented prior to exposition of the method. We apply the method to the test problems in [9] and compare our results with those presented in [9]. Our results reveal spectral leakage at the boundaries which is not addressed in [9]. Potential shortcomings of this method and possible solutions are identified and discussed as directions for future research.

1. INTRODUCTION

In this manuscript, we consider a recently proposed numerical method for approximating the spectrum of the AKNS operator governing the Zakharov-Shabat system. The spectrum of an operator yields important information regarding the behavior and stability of solutions to dynamical systems [11]. Given that it is seldom possible to compute the spectrum analytically, numerical methods are often deployed for spectrum approximation [11]. The method investigated in this manuscript is a Chebyshev collocation method, as presented in [9]. Chebyshev collocation methods, sometimes referred to as *pseudospectral methods*, leverage the properties of orthogonal polynomials as a basis for function approximation [7]. Orthogonal polynomials are of critical importance in approximation theory and they are an important tool in studying otherwise intractable problems [7, 15, 6]. Orthogonal polynomials have a wide range of applications from numerical analysis to quantum mechanics and more.

Before we proceed, we state the notational conventions we adopt in this manuscript.

Remark 1.1 (On notation). We use a^* to denote complex conjugate of a scalar $a \in \mathbb{C}$. We use lowercase bold letters $\mathbf{u}$ to denote a column vector, $\mathbf{u} \in \mathbb{C}^{n\times 1}$ and uppercase bold letters $\mathbf{A}$ to denote matrices $\mathbf{A} \in \mathbb{C}^{n\times n}$, $n > 1$. $\mathbf{A}^*$ and $\mathbf{u}^*$ denote entry-wise complex conjugation without the transpose.

2. ORTHOGONAL POLYNOMIALS

In this section we introduce the notion of a family of orthogonal polynomials. Let $\{p_n(x)\}_{n=0}^{\infty}$ be a family of polynomials such that $p_n(x)$ is of degree exactly n, that is,

$$\lim_{|x|\to\infty} \frac{p_n(x)}{k_n x^n} = 1 \tag{2.1}$$

Date: August 3, 2023.

for some constant $k_n \neq 0$. Given a continuous positive function $w(x)$ on a possibly infinite interval (a, b), we define the (complex) inner product with weight w by

$$\langle f, g \rangle := \int_a^b f(x) g(x)^* w(x) \, \mathrm{d}x \tag{2.2}$$

for functions f, g that are complex-valued. For functions that are real-valued, we may define the real inner product by dropping the complex conjugate on $g(x)^*$ in (2.2). The inner product (2.2) induces the weighted L^2 norm

$$\|f\|_{L^2_w(a,b)} := \sqrt{\langle f, f \rangle}. \tag{2.3}$$

We say that $\{p_n(x)\}_{n=0}^\infty$ is orthogonal with respect to weight w if

$$\langle p_j, p_k \rangle = 0 \qquad \text{for } j \neq k. \tag{2.4}$$

Note that given the weight w, the orthogonal polynomials $p_n(x)$ with respect to w are not unique since we can multiply each polynomial $p_n(x)$ by a nonzero constant c_n and the resulting family of polynomials $q_n(x) := c_n p_n(x)$ would also be orthogonal with respect to w. However, it can be shown using orthogonality that specifying the leading coefficient k_n is enough to define $p_n(x)$ uniquely. A consequence of this is the following.

Proposition 2.1 (Uniqueness). *If $p_k(x)$ and $q_k(x)$ are orthogonal with respect to w to all polynomials of lower degree in $\{p_n(x)\}_{n=0}^\infty$, then $q_n(x) = C p_n(x)$ for some constant C.*

This lets us introduce two different normalizations of orthogonal polynomials.

Definition 2.2 (Monic orthogonal polynomials). Monic orthogonal polynomials with respect to weight w are those with $k_n = 1$ in (2.1). We denote the family of monic orthogonal polynomials by $\{\bar{p}_n\}_{n=0}^\infty$.

Note that monic orthogonal polynomials with respect to weight w are unique.

Definition 2.3 (Orthonormal polynomials). Orthonormal polynomials $\{q_n(x)\}_{n=0}^\infty$ are those that are orthogonal with respect to weight w and have unit norm: $\|q_n\|_{L^2_w} = 1$. Note that this corresponds to $h_n := \langle q_n, q_n \rangle = 1$ for the norming constant h_n.

Polynomials that are orthonormal with respect to weight w are not uniquely determined since we may multiply polynomials by -1 and preserve the orthonormality condition.

2.1. **Construction of orthogonal polynomials.** Orthogonal polynomials can be constructed by applying the Gram-Schmidt orthogonalization algorithm to the monomials $\{1, x, x^2, \ldots, x^n, \ldots\}$ starting with $p_0(x) := 1$ and using the weighted inner product (2.2). More precisely, we may construct orthogonal polynomials by

$$p_0(x) := 1, \tag{2.5}$$

$$p_n(x) := x^n - \sum_{k=0}^{n-1} \frac{\langle x^n, p_k \rangle}{\langle p_k, p_k \rangle} p_k(x), \quad n \geq 1. \tag{2.6}$$

This results in the so-called "standard" orthogonal polynomials with respect to weight w.

2.2. **Recurrence formulae.** Every family of orthogonal polynomials satisfy a three-term recurrence relation.

Theorem 2.4 (Theorem 3.2.1 in [20]). *Suppose that $\{p_n(x)\}_{n=0}^{\infty}$ is a family of orthogonal polynomials with respect to weight w. Then there exists constants a_n, $b_n \neq 0$, c_n such that*

$$xp_0(x) = b_0 p_0(x) + a_0 p_1(x) \tag{2.7}$$

$$xp_n(x) = c_n p_{n-1}(x) + b_n p_n(x) + a_n p_{n+1}(x), \qquad n = 1, 2, 3, \dots. \tag{2.8}$$

Proof. First, note that $xp_0(x)$ is of degree 1. The assertion follows (2.7) follows because $\{p_0, p_1\}$ provides a basis for all polynomials of degree less than or equal to 1.

To prove (2.8), observe that $\langle xf(x), g(x)\rangle = \langle f(x), xg(x)\rangle$ for any two functions $f, g \in L^2_w(a, b)$ and let $r_m(x)$ be an arbitrary polynomial of degree $m < n-1$. Then

$$\langle xp_n(x), r_m(x)\rangle = \langle p_n(x), xr_m(x)\rangle = 0 \tag{2.9}$$

since $xr_m(x)$ has degree less than n, hence it is in the span of $\{p_0, p_1, \dots, p_{n-1}\}$, and $p_n(x)$ is orthogonal to this collection of polynomials. Now, $xp_n(x)$ has degree $n+1$, so it can be written as

$$xp_n(x) = \sum_{k=0}^{n+1} \alpha_k p_k(x) \tag{2.10}$$

for some constants $\alpha_0, \alpha_1, \dots, \alpha_{n+1}$. Taking inner products with $p_k(x)$ on both sides of the above identity and solving for α_k gives

$$\alpha_k = \frac{\langle xp_n(x), p_k(x)\rangle}{\langle p_k(x), p_k(x)\rangle}, \quad k = 0, 1, 2, \dots, n+1. \tag{2.11}$$

But (2.9) implies that $\alpha_k = 0$ for all $k = 0, 1, 2, \dots, n-2$. Therefore, (2.10) reduces to

$$xp_n(x) = \alpha_{n+1} p_{n+1}(x) + \alpha_n p_n(x) + \alpha_{n-1} p_{n-1}(x). \tag{2.12}$$

This finishes the proof. □

2.3. **Classical orthogonal polynomials.** Classical orthogonal polynomials consist of three different families of polynomials that are orthogonal with respect to different weights supported on different intervals. They are due to Hermite, Laguerre, and Jacobi, given in Table 3.

Name	Interval (a, b)	Weight $w(x)$	Standard polynomial	Parameter constraint
Hermite	$(-\infty, \infty)$	e^{-x^2}	$H_n(x)$	
Laguerre	$(0, \infty)$	$x^\alpha\, \mathrm{e}^{-x}$	$L_n^{(\alpha)}(x)$	$\alpha > -1$
Jacobi	$(-1, 1)$	$(1-x)^\alpha (1+x)^\beta$	$P_n^{(\alpha,\beta)}(x)$	$\alpha, \beta > -1$

TABLE 1. Classical orthogonal polynomials.

2.4. **Special cases of Jacobi polynomials.** There are special cases of Jacobi polynomials corresponding to different choices of the parameters α and β in the Jacobi weight $w(x) = (1-x)^\alpha(1+x)^\beta$. These polynomials are used in applications and approximation theory quite often and they have their own names. These polynomials are given in Table 2.

Name	Jacobi parameters α, β	Weight $w(x)$	Standard polynomial
Legendre	$0, 0$	1	$P_n(x)$
Chebyshev (first kind)	$-\frac{1}{2}, -\frac{1}{2}$	$\frac{1}{\sqrt{1-x^2}}$	$T_n(x)$
Chebyshev (second kind)	$\frac{1}{2}, \frac{1}{2}$	$\sqrt{1-x^2}$	$U_n(x)$
Ultraspherical	$\lambda - \frac{1}{2}, \lambda - \frac{1}{2}$	$\left(1-x^2\right)^{\lambda-1/2}$, for $\lambda \neq 0$	$C_n^{(\lambda)}(x)$

Table 2. Special cases of Jacobi polynomials.

A good reference for the parameters, weights, intervals, and values of other defining quantities for orthogonal polynomials is [16, §18.3 and Table 18.3.1], see also Szegö's book [20].

In the remainder of this manuscript we will work with Chebyshev polynomials of the first kind.

3. Chebyshev polynomials and approximation

As was introduced in Table 2, Chebyshev polynomials (of the first kind) $T_n(x)$ are orthogonal on the interval $(-1, 1)$ with respect to the weight function

$$w(x) = (1-x^2)^{-1/2}, \qquad x \in [-1, 1]. \tag{3.1}$$

3.1. **Some properties of Chebyshev polynomials.** The leading coefficients k_n of the standard Chebyshev polynomials $T_n(x)$ are given by

$$k_n = \begin{cases} 1 & n = 0, \\ 2^{n-1} & n > 0, \end{cases} \tag{3.2}$$

see [16, Table 18.3.1]. Remarkably, these polynomials have a simple closed form expression.

Proposition 3.1. *For $n \geq 0$, we have $T_n(x) = \cos(n \arccos(x))$.*

Proof. The proof will rely on uniqueness. Let $\tilde{T}_n(x) := \cos(n \arccos(x))$. We first show that $\tilde{T}_n(x)$ is a polynomial. We note that $\tilde{T}_0(x) = 1$ and $\tilde{T}_1(x) = x$. For $\tilde{T}_2(x)$, we let $\theta = \arccos(x)$ and find that

$$\tilde{T}_2(x) = \cos(2\theta) = \cos(\theta)^2 - \sin(\theta)^2 = 2\cos(\theta)^2 - 1 = 2x^2 - 1, \tag{3.3}$$

which shows that $\tilde{T}_2(x)$ is also a polynomial. Again using the sum formula $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$, we find that

$$\tilde{T}_{n+1}(x) = \cos((n+1)\theta) = \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta), \tag{3.4}$$

$$\tilde{T}_{n-1}(x) = \cos((n+1)y) = \cos(n\theta)\cos(\theta) + \sin(n\theta)\sin(\theta). \tag{3.5}$$

Adding these formulae and substituting $\cos(\theta) = x$ results in the recursion relation

$$\tilde{T}_{n+1}(x) = 2x\tilde{T}_n(x) - \tilde{T}_{n-1}(x). \tag{3.6}$$

This implies that each $\tilde{T}_n(x)$ is a polynomial since $\tilde{T}_0(x)$ and $\tilde{T}_1(x)$ are polynomials. Next we show that $\{\tilde{T}_n(x)\}_{n=0}^{\infty}$ are orthogonal with respect to (3.1). Again using $x = \cos(\theta)$ and $\mathrm{d}x = -\sin(\theta)$, we have for $m \neq n$

$$\begin{aligned}\int_{-1}^{1} \frac{\tilde{T}_n(x)\tilde{T}_m(x)}{\sqrt{1-x^2}}\,\mathrm{d}x &= -\int_{\pi}^{0} \cos(n\theta)\cos(m\theta)\,\mathrm{d}\theta = \frac{1}{4}\int_{0}^{\pi} (\,\mathrm{e}^{\mathrm{i}m\theta} + \mathrm{e}^{-\mathrm{i}m\theta})(\,\mathrm{e}^{\mathrm{i}n\theta} + \mathrm{e}^{-\mathrm{i}n\theta})\,\mathrm{d}\theta \\ &= \frac{1}{4}\int_{0}^{\pi} \mathrm{e}^{\mathrm{i}(m+n)\theta} + \mathrm{e}^{\mathrm{i}(m-n)\theta} + \mathrm{e}^{-\mathrm{i}(m-n)\theta} + \mathrm{e}^{-\mathrm{i}(m+n)\theta}\,\mathrm{d}\theta = 0\end{aligned} \tag{3.7}$$

since $m \pm n \in \mathbb{Z}$. Finally, recalling that $\tilde{T}_0(x) = 1$ and $\tilde{T}_1(x) = x$, the recursion relation (3.8) implies the leading coefficient of $\tilde{T}_n(x)$ is 2^{n-1} for $n \geq 2$. Since this matches with (3.2), we conclude by uniqueness that $\tilde{T}_n(x) = T_n(x)$. $\square$

A by product of our proof above is the special case of the three-term recurrence relation stated in Theorem 2.4 for the Chebyshev polynomials.

Proposition 3.2. *$T_n(x)$ satisfies*

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \qquad n \geq 1. \tag{3.8}$$

The recurrence formula (3.8) provides a fast method to construct these polynomials and evaluate them numerically. Another property concerns the norming constants h_n of the Chebyshev polynomials.

Proposition 3.3. *We have*

$$h_n = \langle T_n(x), T_n(x)\rangle = \int_{-1}^{1} T_n(x)^2 w(x)\,\mathrm{d}x = \begin{cases} \pi, & n = 0, \\ \frac{\pi}{2} & n > 0. \end{cases} \tag{3.9}$$

The proof is by direct computation as in (3.7).

3.2. **Chebyshev series.** Chebyshev series expansions are an example of an orthogonal polynomial expansion, which are similar to expansions of periodic functions in Fourier series. Expansions of continuous functions in other orthogonal polynomial families are also possible. Chebyshev series expansions for an arbitrary continuous function $f : [-1, 1] \to \mathbb{R}$ are well-understood and much literature has been produced on the subject (see [7, 8, 13, 15, 20]). A Chebyshev series expansion of a continuous function $f\colon [-1, 1] \to \mathbb{R}$ is given by

$$f(x) = \sum_{i=0}^{\infty} c_i T_i(x) \tag{3.10}$$

provided the sum converges uniformly and absolutely for $x \in [-1, 1]$. Here $c_i \in \mathbb{R}$, and $T_i(x)$ is a Chebyshev polynomial of the first kind as defined above. Taking the inner product as in (2.2) with

$T_k(x)$, (3.10) yields

$$\langle f(x), T_k(x)\rangle = \sum_{i=0}^{\infty} c_i \langle T_i(x), T_k(x)\rangle = c_k \langle T_k(x), T_k(x)\rangle \tag{3.11}$$

by orthogonality. Equation (3.11) provides an explicit formula for the coefficients in the expansion,

$$c_k = \frac{\langle f(x), T_k(x)\rangle}{\langle T_k(x), T_k(x)\rangle} \tag{3.12}$$

which has the integral representation

$$c_k = \frac{N_k}{\pi} \int_{-1}^{1} f(x) T_k(x) w(x)\,\mathrm{d}x, \tag{3.13}$$

where $N_0 = 1$ and $N_k = 2$ for $k \geq 1$.

3.3. **Approximation with Chebyshev polynomials.** In practice one can use (3.10) to approximate smooth functions f on $[-1, 1]$ by a finite truncation of (3.10):

$$f(x) \approx \breve{f}(x) := \sum_{k=0}^{n-1} c_k T_k(x), \tag{3.14}$$

The error of a partial sum approximation is bounded by the sum of the absolute values of every neglected coefficient [7]. The accuracy of the approximation (3.14) is typically dictated by the smoothness properties of f. As the number of derivatives f has increases, the decay rate of the coefficients c_k as $k \to \infty$ increases, making the truncation a better approximation.

Proposition 3.4 (From [7], Theorem 6)**.** *If $f(x) \approx \breve{f}(x)$ as defined in (3.14), then an upper bound of the error is*

$$E(x) = |f(x) - \breve{f}(x)| \leq \sum_{k=n}^{\infty} |c_k| \tag{3.15}$$

for $x \in [-1, 1]$.

Proof. The proof follows from the fact that Chebyshev polynomials are bounded by 1 such that $|T_n(x)| \leq 1$. Thus the nth term is bounded by $|c_n|$ and the sum of bounds of the neglected terms yields the upper bound for the error. □

3.4. **Discretization of the derivative operator.** Approximations of the form (3.14) for functions become essential in solving differential equations numerically. As such, it is useful to have a linear relationship between the Chebyshev series coefficients $\{c_k\}_{k=1}^{n}$ of a smooth function f and the Chebyshev series coefficients $\{d_k\}_{k=1}^{n}$ of its derivative $f'(x)$ which is also assumed to be sufficiently smooth. Suppose we have

$$f(x) = \sum_{k=0}^{\infty} c_k T_k(x) \quad \text{and} \quad f'(x) = \sum_{k=0}^{\infty} d_k T_k(x). \tag{3.16}$$

The key fact is the following recurrence relationship due to Clenshaw.

Proposition 3.5 ([8])**.** $2kc_k = d_{k-1} - d_{k+1}$ *for $k \geq 1$.*

The proof will make us of the following lemma from [8].

Lemma 3.6. *Integration of Chebyshev polynomials results in the following identities*

$$\int T_k(x)\,\mathrm{d}x = \begin{cases} T_1(x) & \text{for } k=0 \\ \frac{1}{4}T_2(x) & \text{for } k=1 \\ \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1} - \frac{T_{k-1}(x)}{k-1}\right] & \text{for } k>1. \end{cases} \tag{3.17}$$

Proof of Proposition 3.5. By term-by-term integration of (3.16), we have

$$f(x) = \sum_{k=0}^{\infty} d_k \int T_k(x)\,\mathrm{d}x \tag{3.18}$$

which can be written as

$$\sum_{k=0}^{\infty} c_k T_k(x) = \sum_{k=0}^{\infty} d_k \int T_k(x)\,\mathrm{d}x \tag{3.19}$$

by replacing f with its Chebyshev expansion. Then, by (3.17) and orthogonality, it follows that

$$\begin{aligned} c_0 &= 0 \\ 2kc_k &= (d_{k-1} - d_{k+1}) \quad \text{for } k \geq 1. \end{aligned} \tag{3.20}$$

□

The relationship in (3.20) provides a clue that a suitable (linear) map should exist. However, it defines a relationship for coefficients c_k of the expansion of f in terms of the coefficients d_k of the expansion of f' — while the relation we seek should give d_k in terms of c_k. The following proposition establishes this reversed relationship.

Proposition 3.7. *The kth coefficient d_k of the Chebyshev expansion of f' is given in terms of the coefficients c_k of the Chebyshev expansion of f by*

$$d_k = 2\sum_{i=0}^{\infty}(k+2i+1)c_{k+2i+1} \quad \text{for } k = 0, 1, 2, \ldots. \tag{3.21}$$

Proof. We write copies of the identity (3.20) by shifting the indices by 2 to have

$$\begin{aligned} 2(k+1)c_{k+1} &= d_k - d_{k+2} \\ 2(k+3)c_{k+3} &= d_{k+2} - d_{k+4} \\ 2(k+5)c_{k+5} &= d_{k+4} - d_{k+6} \\ &\vdots \end{aligned} \tag{3.22}$$

Taking an infinite sum (assuming the coefficients decay sufficiently rapidly), we make use of the telescoping behavior of the right-hand side to arrive at

$$2\sum_{i=0}^{\infty}(k+2i+1)c_{k+2i+1} = d_k. \tag{3.23}$$

□

This establishes the existence of an (infinite-dimensional) linear transformation that maps $\{c_k\}_{k=1}^{\infty}$ to $\{d_k\}_{k=1}^{\infty}$. Indeed, evaluation of (3.21) for sequential values of k yields the following system

$$\begin{aligned}
\frac{1}{2}d_0 &= [c_1 + 3c_3 + 5c_5 + \dots] \\
d_1 &= 2[2c_2 + 4c_4 + 6c_6 + \dots] = [4c_2 + 8c_4 + 12c_6 + \dots] \\
d_2 &= 2[3c_3 + 5c_5 + 7c_7 + \dots] = [6c_3 + 10c_5 + 14c_7 + \dots] \\
d_3 &= 2[4c_4 + 6c_6 + 8c_8 + \dots] = [8c_4 + 12c_6 + 16c_8 + \dots] \\
&\vdots
\end{aligned} \tag{3.24}$$

Now, we can make use of infinite-dimensional linear algebra to have an infinite-dimensional matrix representation of the system of equation (3.24). To this end, we introduce the operator

$$\mathcal{D} = \begin{pmatrix} 0 & 1 & 0 & 3 & 0 & 5 & 0 & 7\dots \\ & 0 & 4 & 0 & 8 & 0 & 12 & 0\dots \\ & & 0 & 6 & 0 & 10 & 0 & 14\dots \\ & & & 0 & 8 & 0 & 12 & 0\dots \\ & & & & 0 & 10 & 0 & 14\dots \\ & & & & & \ddots & \ddots & \ddots \end{pmatrix} \tag{3.25}$$

If we represent the coefficients $\{c_k\}_{k=1}^{\infty}$ and $\{d_k\}_{k=1}^{\infty}$ as infinite column vectors

$$\mathbf{c}_\infty := \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} \quad \text{and} \quad \mathbf{d}_\infty := \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \end{bmatrix}, \tag{3.26}$$

the system of (infinitely many) equations (3.23) reads simply

$$\mathbf{d}_\infty = \mathcal{D}\mathbf{c}_\infty. \tag{3.27}$$

The following proposition will be useful when developing the method later in the manuscript.

Proposition 3.8. *Multiplication of a row-vector of basis polynomials by the differentiation operator $\mathcal{D}$ yields the derivatives of the basis polynomials such that*

$$\mathbf{T}'(x) = \mathbf{T}(x)\mathcal{D}. \tag{3.28}$$

Proof. The proof follows by recognition that differentiation of a Chebyshev series expansion of f yields $f'(x) = \mathbf{T}'(x)\mathbf{c}_\infty$. However, we also have shown above that $f'(x) = \mathbf{T}(x)\mathbf{d}_\infty$. Since $\mathbf{d}_\infty = \mathcal{D}\mathbf{c}_\infty$, it follows that $\mathbf{T}(x)\mathcal{D}\mathbf{c}_\infty = \mathbf{T}'(x)\mathbf{c}_\infty$. Hence, $\mathbf{T}(x)\mathcal{D} = \mathbf{T}'(x)$. $\square$

Remark 3.9. If one is to work with an approximation obtained via truncated Chebyshev series (3.14), a suitable truncation of $\mathcal{D}$ maps the truncated coefficients $\mathbf{c}$ to the truncated coefficients $\mathbf{d}$, but the accuracy of the resulting approximation

$$f'(x) \approx \sum_{k=1}^{n-1} d_k T_k(x) \tag{3.29}$$

need not be as good as that of the approximation (3.14).

3.5. **Discretization of the multiplication operator.** We now consider the Chebyshev series expansion for a product of two arbitrary smooth functions on $[-1,1]$

$$f(x)=\sum_{j=0}^{\infty}\alpha_j T_j(x) \quad \text{and} \quad g(x)=\sum_{j=0}^{\infty}\beta_j T_j(x), \tag{3.30}$$

and the Chebysehv series for their product:

$$f(x)g(x)=\sum_{j=0}^{\infty}\tau_j T_j(x). \tag{3.31}$$

The perspective here is that for a fixed function $g(x)$ whose Chebyshev coefficients β_j we know, we should be able to express the coefficients τ_j of its product with a given $f(x)$ as the image of a linear transformation $\mathcal{M}[g]$ acting on the coefficients α_j of the Chebyshev series for $f(x)$. In other words, we look for an infinite dimensional matrix $\mathcal{M}[g]$ such that

$$\boldsymbol{\tau}_\infty=\mathcal{M}[g]\boldsymbol{\alpha}_\infty, \tag{3.32}$$

where the entries of $\mathcal{M}[g]$ given in terms of the Chebyshev coefficients $\{\beta_j\}_{j=1}^{\infty}$ of the fixed function g, and as before, $\boldsymbol{\tau}_\infty$ and $\boldsymbol{\alpha}_\infty$ stand for the infinite-dimensional column vector representations of the Chebyshev coefficients of fg and f, respectively. In this case, $\mathcal{M}[g]$ provides a discretization of the multiplication operator by the function g. This is indeed possible, as explained in, for example, [17]. From [17], by [4, Proposition 2.1], the explicit formulae for τ_k is:

$$\tau_k=\begin{cases}\alpha_0\beta_0+\frac{1}{2}\sum_{j=1}^{\infty}\alpha_j\beta_j & \text{for } k=0,\\ \frac{1}{2}\sum_{j=0}^{k-1}\alpha_{k-1}\beta_j+\alpha_0\beta_k+\frac{1}{2}\sum_{j=1}^{\infty}\alpha_j\beta_{j+k}+\frac{1}{2}\sum_{j=0}^{\infty}\alpha_{j+k}\beta_j & \text{for } k\geq 1.\end{cases} \tag{3.33}$$

Thus, as explained in [17], a multiplication operator, denoted by $\mathcal{M}[g]$, transforms the coefficients of the expansion of f in to the coefficients for the product fg. The infinite dimensional matrix $\mathcal{M}[g]$ is given by

$$\mathcal{M}[g]=\frac{1}{2}\left[\begin{pmatrix}2\beta_0 & \beta_1 & \beta_2 & \beta_3 & \cdots\\ \beta_1 & 2\beta_0 & \beta_1 & \beta_2 & \ddots\\ \beta_2 & \beta_1 & 2\beta_0 & \beta_1 & \ddots\\ \beta_3 & \beta_2 & \beta_1 & 2\beta_0 & \ddots\\ \vdots & \ddots & \ddots & \ddots & \ddots\end{pmatrix}+\begin{pmatrix}0 & 0 & 0 & 0 & \cdots\\ \beta_1 & \beta_2 & \beta_3 & \beta_4 & \iddots\\ \beta_2 & \beta_3 & \beta_4 & \beta_5 & \iddots\\ \beta_3 & \beta_4 & \beta_5 & \beta_6 & \iddots\\ \vdots & \iddots & \iddots & \iddots & \iddots\end{pmatrix}\right]. \tag{3.34}$$

4. Application: the Nonlinear Schrödinger Equation

The focusing nonlinear Schrödinger equation (NLS)

$$\mathrm{i}q_t+\frac{1}{2}q_{xx}+|q|^2q=0 \tag{4.1}$$

for a complex-valued field $q=q(x,t)$ is a dispersive nonlinear wave equation that models propagation of waves in various physical contexts, from motion of water waves [24] to beam propagation in optical fibers [14]. This equation is known to be an integrable system [1, 25] in the sense that it arises as the compatibility condition of the following systems of linear differential equations of a *Lax pair*:

$$\frac{\partial \mathbf{w}}{\partial x} = \mathbf{U}\mathbf{w}, \qquad \mathbf{U} := \begin{bmatrix} -\mathrm{i}\lambda & q(x,t) \\ -q(x,t)^* & \mathrm{i}\lambda \end{bmatrix}, \tag{4.2}$$

$$\frac{\partial \mathbf{w}}{\partial t} = \mathbf{V}\mathbf{w}, \qquad \mathbf{V} := \begin{bmatrix} -\mathrm{i}\lambda^2 + \mathrm{i}\frac{1}{2}|q(x,t)|^2 & \lambda q(x,t) + \mathrm{i}\frac{1}{2}q_x(x,t) \\ -\lambda q(x,t)^* + \mathrm{i}\frac{1}{2}q_x(x,t)^* & \mathrm{i}\lambda^2 - \mathrm{i}\frac{1}{2}|q(x,t)|^2 \end{bmatrix} \tag{4.3}$$

where $\mathbf{w} = \mathbf{w}(\lambda; x, t) \in \mathbb{C}^2$ is an auxiliary unknown column-vector, and λ is a spectral parameter. This has the following meaning. Assuming that λ is independent of (x, t), the compatibility condition $\mathbf{w}_{xt}(\lambda; x, t) = \mathbf{w}_{tx}(\lambda; x, t)$ holds if and only if (4.1) holds. Note that enforcing $\mathbf{w}_{xt}(\lambda; x, t) = \mathbf{w}_{tx}(\lambda; x, t)$ results in the condition

$$\frac{\partial \mathbf{U}}{\partial t} - \frac{\partial \mathbf{V}}{\partial x} + [\mathbf{U}, \mathbf{V}] = \mathbf{0}, \tag{4.4}$$

which is sometimes called a "zero-curvature condition".

The focusing NLS equation also admits a special kind of explicit solitary wave solutions, named *solitons*, given by

$$q_{\mathrm{s}}(x,t) := 2\beta\, \mathrm{e}^{-2\mathrm{i}(\alpha x + (\alpha^2 - \beta^2)t + \phi)}\, \mathrm{sech}(2\beta(x - x_0 + 2\alpha t)), \tag{4.5}$$

where α, $\beta > 0$, x_0, and ϕ are real-valued parameters. This solution is a traveling wave, propagating with velocity -2α (propagating right if $\alpha < 0$ and propagating left if $\alpha > 0$) and amplitude $2\beta > 0$. The profile of $q_{\mathrm{s}}(x,t)$ is localized exponentially in space since the sech-shaped envelope in (4.5) decays exponentially fast as $x \to \pm\infty$ for any time $t \in \mathbb{R}$. Note that the width of the profile is inversely proportional to the amplitude. The location of the peak of $|q_{\mathrm{s}}(x, t = 0)|$ is given by $x = x_0$ and the parameter ϕ indicates the amount of initial phase-shift in the carrier wave (the exponential factor in (4.5)). While changing x_0 or ϕ does not have any effect on the shape and speed of the soliton, changing α and β changes the shape and speed of the soliton. Thus, we may say that the soliton solution is essentially characterized by the real-valued parameters α and $\beta > 0$.

Using techniques relating to integrability of the focusing NLS equation, it was proved in [5] that the solution of the initial-value problem for (4.1) on the full line $-\infty < x < \infty$ with rapidly decaying initial data $q_0(x) = q(x, 0)$ resolves into a sum of $N \in \mathbb{Z}_{\geq 0}$ (if any) solitons plus dispersive waves of size proportional to $t^{-1/2}$ that decay to $q = 0$ in the supremum norm as time $t \to \infty$. Remarkably, the soliton content of the long-time development can be read off from the initial data. The differential equation (4.2) reads

$$\frac{\partial}{\partial x} w_1(\lambda; x, t) = -\mathrm{i}\lambda w_1(\lambda; x, t) + q(x,t) w_2(\lambda; x, t), \tag{4.6}$$

$$\frac{\partial}{\partial x} w_2(\lambda; x, t) = -q(x,t)^* w_1(\lambda; x, t) + \mathrm{i}\lambda w_2(\lambda; x, t) \tag{4.7}$$

multiplying the former equation by i and the latter by $-\mathrm{i}$ and rearranging yields

$$\mathrm{i}\frac{\partial}{\partial x}w_1(\lambda;x,t) - \mathrm{i}q(x,t)w_2(\lambda;x,t) = \lambda w_1(\lambda;x,t), \tag{4.8}$$

$$-\mathrm{i}\frac{\partial}{\partial x}w_2(\lambda;x,t) - \mathrm{i}q(x,t)^*w_1(\lambda;x,t) = \lambda w_2(\lambda;x,t) \tag{4.9}$$

This can be rewritten as

$$\begin{bmatrix} \mathrm{i}\frac{\partial}{\partial x} & -\mathrm{i}q(x,t) \\ -\mathrm{i}q(x,t)^* & -\mathrm{i}\frac{\partial}{\partial x} \end{bmatrix} \mathbf{w}(\lambda;x,t) = \lambda \mathbf{w}(\lambda;x,t), \tag{4.10}$$

which is the spectral problem $\mathcal{U}\mathbf{w} = \lambda\mathbf{w}$ for the differential operator

$$\mathcal{U} := \begin{bmatrix} \mathrm{i}\frac{\partial}{\partial x} & -\mathrm{i}q \\ -\mathrm{i}q^* & -\mathrm{i}\frac{\partial}{\partial x} \end{bmatrix} \tag{4.11}$$

acting on column-vector-valued functions of (x,t). It is known that the eigenvalues of $\mathcal{U}$ are finitely many and simple for a rapidly decaying generic function $x \mapsto q(x,t)$ [5], and they come in complex-conjugate pairs. As it turns out, the soliton content of the above-mentioned long-time development of rapidly decaying initial data $q_0(x)$ for the focusing NLS equation (4.1) is characterized by the eigenvalues of the operator $\mathcal{U}$ with $q = q_0$. More precisely, to each complex-conjugate pair of eigenvalue

$$\lambda_k = \alpha_k + \mathrm{i}\beta_k, \quad \lambda_k^* = \alpha_k - \mathrm{i}\beta_k, \qquad \alpha_k \in \mathbb{R}, \quad \beta_k > 0, \tag{4.12}$$

there corresponds a soliton (4.5) with $\alpha = \alpha_k$ and $\beta = \beta_k$ emerging in the long-time development of the solution $q(x,t)$ of the initial-value problem with $q(x,0) = q_0(x)$. Thus, the speed of each soliton is determined by the real part of the eigenvalue and the amplitude (and width) of the soliton is determined by the imaginary part of the eigenvalue.

In its wider context, the eigenvalues of $\mathcal{U}$ are a part of what is called the *scattering data* associated with q: $q(x,t)$ can be reconstructed from eigenvalues of $\mathcal{U}$ along with so-called norming constants and reflection coefficient of $\mathcal{U}$. This is called the inverse-scattering transform (IST) method associated with the focusing NLS equation (4.1). It provides a nonlinear analogue of the Fourier transform method to solve the initial-value problem for linear partial differential equations.

In the remainder of this manuscript we focus on computing the eigenvalues of $\mathcal{U}$ by using the Chebyshev approximation theory we covered in the preceding sections. This is an important ingredient of the numerical IST methods associated with the NLS equation [22, 23].

4.1. **A method to compute the discrete spectrum of the AKNS operator.** To approximate the spectrum of the operator $\mathcal{U}$, as defined in §3, we proceed with developing the numerical method outlined in [9]. We will use Chebyshev extreme points as collocation points since unequally spaced collocation points with increased density near the boundaries reduces the risk of wild oscillations near the boundaries (Runge's Phenomenon [15], [7]). Since Chebyshev polynomials are defined only for values in $[-1,1]$, we will make use of an invertible mapping, $H : \mathbb{R} \to [-1,1]$, in order to extend our Chebyshev theory to functions defined on the full line. We will begin by defining our collocation points to be Chebyshev extreme points defined in the usual manner before revisiting the material in the preceding sections with extension to functions defined on the full line.

Definition 4.1. Let the set $\{\gamma_j\}_{j=0}^{n-1}$ be the set of Chebyshev extreme points such that

$$\gamma_j = -\cos\left(\frac{j\pi}{n-1}\right),\ j = 0, 1, 2, \dots, n, \tag{4.13}$$

which yields

$$\begin{aligned}
\gamma_0 &= -\cos(0) = -1,\\
\gamma_1 &= -\cos\left(\frac{\pi}{n-1}\right),\\
\gamma_2 &= -\cos\left(\frac{2\pi}{n-1}\right),\\
&\vdots,\\
\gamma_j &= -\cos\left(\frac{j\pi}{n-1}\right),\\
&\vdots,\\
\gamma_{n-1} &= -\cos\left(\frac{(n-1)\pi}{n-1}\right) = 1.
\end{aligned} \tag{4.14}$$

We will refer to these points as collocation points and represent them as a column vector, $\boldsymbol{\gamma} \in \mathbb{R}^{n\times 1}$:

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-1} \end{bmatrix}. \tag{4.15}$$

We observe that for each $\gamma_j \in \{\gamma\}_{j=0}^{n-1}$, $\gamma_j \in [-1, 1]$ by definition. Thus, from (3.14), an approximation of a function f evaluated at a Chebyshev extreme point interpolates f such that for sufficiently large $n \in \mathbb{N}$ we have

$$f(\gamma_j) = \sum_{k=0}^{n-1} c_k T_k(\gamma_j) \tag{4.16}$$

where c_k is the coefficient of the polynomial and $T_k(\gamma_j)$ is the kth Chebyshev Polynomial of the first kind. In vector notation (4.16) can be represented as

$$f(\gamma_j) = \mathbf{T}(\gamma_j)\mathbf{c} \tag{4.17}$$

where $\mathbf{T}(\gamma_j) \in \mathbb{R}^{1\times n}$ is a row vector of Chebyshev polynomials at γ_j; and, $\mathbf{c} \in \mathbb{R}^{n\times 1}$ is a column vector whose entries are the coefficients of the expansion in (4.16). Evaluation of (4.17) at each

Chebyshev collocation point in $\{\gamma_j\}_{j=0}^{n-1}$ results in the following system

$$\begin{aligned} f(\gamma_0) &= \mathbf{T}(\gamma_0)\mathbf{c} \\ f(\gamma_1) &= \mathbf{T}(\gamma_1)\mathbf{c} \\ f(\gamma_2) &= \mathbf{T}(\gamma_2)\mathbf{c} \\ &\vdots \\ f(\gamma_{n-1}) &= \mathbf{T}(\gamma_{n-1})\mathbf{c}. \end{aligned} \tag{4.18}$$

As the vector $\mathbf{c}$ is independent of γ, we can represent the above system by an equivalent matrix equation,

$$f(\vec{\gamma}) = \mathbf{T}(\vec{\gamma})\mathbf{c} \tag{4.19}$$

where $f(\vec{\gamma}) \in \mathbb{R}^{n\times 1}$ is a column vector containing the values of f at the Chebyshev collocation points, $\mathbf{T}(\vec{\gamma}) \in \mathbb{R}^{n\times n}$ is a matrix whose rows are the Chebyshev polynomials for γ_j, and $\mathbf{c}$ is defined as above. The matrix $\mathbf{T}$ will play a pivotal role in many of the approximations necessary for the numerical method.

Proposition 4.2. *The column vector* $\mathbf{c}$ *can be computed by*

$$\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}) = \mathbf{c}. \tag{4.20}$$

Proof. The proof follows from (4.19) by left-hand multiplication of $\mathbf{T}^{-1}(\vec{\gamma})$. Note that $\mathbf{T}(\vec{\gamma})$ is invertible by orthogonality of the columns of $\mathbf{T}(\vec{\gamma})$ under the inner product. □

This provides a convenient method for approximating a function f by its values at the Chebyshev extreme points.

Proposition 4.3. *An arbitrary function* $f : [-1, 1] \to \mathbb{R}$ *is approximated by*

$$f(x) \approx \mathbf{T}(x)\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}). \tag{4.21}$$

Proof. The proof follows by left-hand multiplication of (4.20) by $\mathbf{T}(x) \in \mathbb{R}^{1\times n}$. Doing so yields

$$\mathbf{T}(x)\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}) = \mathbf{T}(x)\mathbf{c}. \tag{4.22}$$

but, $\mathbf{T}(x)\mathbf{c} \approx f(x)$ by (3.14) above. Hence,

$$f(x) \approx \mathbf{T}(x)\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}). \tag{4.23}$$

□

Equation (4.23) defines an approximating polynomial in x. Approximation of the derivative of f is obtained in a similar manner, but by use of the differentiation matrix developed in §3.4 above.

Proposition 4.4. *The derivative of* f *is approximated from the coefficients of* f *by*

$$f'(x) \approx \mathbf{T}(x)\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}). \tag{4.24}$$

Proof. By differentiation of (4.23), we have

$$\begin{aligned} f'(x) &\approx \mathbf{T}'(x)\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}) \\ &= \mathbf{T}(x)\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}) \text{ by §3.4 above.} \end{aligned} \tag{4.25}$$

Thus, we have

$$f'(x) \approx \mathbf{T}(x)\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma})f(\vec{\gamma}) \tag{4.26}$$

for $x \in [-1, 1]$. □

For approximating a function $g : \mathbb{R} \to \mathbb{R}$ using Chebyshev series, we rely on the following map:

$$H(x) = \tanh(ax) \tag{4.27}$$

where $a \in (0, 1)$ is a parameter that allows us to distribute Chebyshev points according to the behavior of the function being approximated (see discussion on 'Rapid Change Interval' in [9]). For the general development of the methodology, we will neglect the discussion of the parameter a until a later section.

Proposition 4.5. *An approximation of an arbitrary function $g : \mathbb{R} \to \mathbb{R}$ in a Chebyshev basis is given by*

$$g(x) \approx \mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})g(H^{-1}(\vec{\gamma})). \tag{4.28}$$

Proof. Let $g : \mathbb{R} \to \mathbb{R}$ be an arbitrary function, and let $H : \mathbb{R} \to (-1, 1)$ be the invertible map, $H(x) = \tanh(x)$. We again demand interpolation such that the values of g at the inverse-mapped values of x be equal to the values of g at the Chebyshev extreme points. Thus we enforce the constraint that $x = H^{-1}(\gamma_j)$. Doing so yields

$$g(\vec{x}) = g(H^{-1}(\vec{\gamma})) = \mathbf{T}(\vec{\gamma})\mathbf{c} \tag{4.29}$$

for some vector of coefficients $\mathbf{c}$. By (4.20) above, the coefficients can be computed by

$$\mathbf{T}^{-1}(\vec{\gamma})g(H^{-1}(\vec{\gamma})) = \mathbf{c}. \tag{4.30}$$

To obtain the approximating Chebyshev polynomial, we multiply by the row-vector $\mathbf{T}(H(x))$,

$$\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})g(H^{-1}(\vec{\gamma})) = \mathbf{T}(H(x))\mathbf{c}. \tag{4.31}$$

Observing that $\mathbf{T}(H(x))\mathbf{c} \approx g(x)$, we have

$$\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})g(H^{-1}(\vec{\gamma})) \approx g(x). \tag{4.32}$$

□

Proposition 4.6. *An approximation of $g' : \mathbb{R} \to \mathbb{R}$ in a Chebyshev basis is given by*

$$g'(x) \approx H_x(x)\mathbf{T}(H(x))\mathcal{D}\mathbf{T}^{-1}(\gamma)g(H^{-1}(\vec{\gamma})) \tag{4.33}$$

Proof. The proof follows by differentiation of (4.32) and the chain rule. □

By substitution of (4.28) and (4.33) in the Zakharov-Shabat system,

$$\begin{bmatrix} i\frac{\partial}{\partial x} & -iq \\ -iq^* & -i\frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = ik \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{4.34}$$

we have:

(4.35)

$$-i \begin{bmatrix} -H_x(x)\mathbf{T}(H(x))\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) & \mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})q(H^{-1}(\vec{\gamma}))\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma}) \\ \mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})q^*(H^{-1}(\vec{\gamma}))\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma}) & H_x(x)\mathbf{T}(H(x))\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) \end{bmatrix} \begin{bmatrix} w_1(H^{-1}(\vec{\gamma})) \\ w_2(H^{-1}(\vec{\gamma})) \end{bmatrix}$$
$$= ik \begin{bmatrix} \mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})w_1(H^{-1}(\vec{\gamma})) \\ \mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})w_2(H^{-1}(\vec{\gamma})) \end{bmatrix}$$

Proposition 4.7. *The product of $-H_x(x)\mathbf{T}(H(x))\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma})w_1(H^{-1}(\vec{\gamma}))$ is an approximation of the derivative of $w_1(x)$ for a single given x-value.*

Proof. The proof follows by direct computation via (4.33). □

Proposition 4.8. *The product of $\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})q(H^{-1}(\vec{\gamma}))\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})w_2(H^{-1}(\vec{\gamma}))$ is an approximation of the product of two functions by their Chebyshev expansions.*

Proof. The proof is a direct consequence of Proposition 4.5. □

Proposition 4.9. *The product of $ik\mathbf{T}(H(x))\mathbf{T}^{-1}(\vec{\gamma})w_1(H^{-1}(\vec{\gamma}))$ is an approximation of $w_1(x)$ scaled by ik.*

Proof. The proof follows by direct computation as outlined in (4.32). □

The system in (4.35) is an approximation of the Zakharov-Shabat system for a single value of x, the solution of which corresponds to a single spectral value, λ_j in

$$\mathcal{U}\mathbf{w} = \lambda\mathbf{w} \tag{4.36}$$

as in §4. To approximate the spectrum of the AKNS operator, which is to say all $\lambda \in \{\lambda_j\}_{j=1}^n$, we evaluate (4.35) at n Chebyshev collocation points. Thus, with $\vec{x} = H^{-1}(\vec{\gamma})$, we have

(4.37)

$$-i \begin{bmatrix} -H_x(H^{-1}(\vec{\gamma}))\mathbf{T}(\vec{\gamma})\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) & \mathbf{T}(\vec{\gamma})\mathbf{T}^{-1}(\vec{\gamma})qH^{-1}(\vec{\gamma}))\mathbf{T}(\vec{\gamma}))\mathbf{T}^{-1}(\vec{\gamma}) \\ \mathbf{T}(\vec{\gamma})\mathbf{T}^{-1}(\vec{\gamma})q^*(H^{-1}(\vec{\gamma}))\mathbf{T}(\vec{\gamma})\mathbf{T}^{-1}(\vec{\gamma}) & H_x(H^{-1}(\vec{\gamma}))\mathbf{T}(\vec{\gamma})\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) \end{bmatrix} \begin{bmatrix} w_1(H^{-1}(\vec{\gamma}) \\ w_2(H^{-1}(\vec{\gamma})) \end{bmatrix}$$
$$= ik \begin{bmatrix} \mathbf{T}(\vec{\gamma})\mathbf{T}^{-1}(\vec{\gamma})w_1(H^{-1}(\vec{\gamma})) \\ \mathbf{T}(\vec{\gamma})\mathbf{T}^{-1}(\vec{\gamma})w_2(H^{-1}(\vec{\gamma})) \end{bmatrix}.$$

Since, $\mathbf{T}(\vec{\gamma})\mathbf{T}^{-1}(\vec{\gamma})$ is the identity operator I, we can represent (4.37) as

(4.38)

$$-i \begin{bmatrix} -\text{diag}[H_x(H^{-1}(\vec{\gamma}))]\mathbf{T}(\vec{\gamma})\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) & \text{diag}[q(H^{-1}(\vec{\gamma}))] \\ \text{diag}[q^*(H^{-1}(\vec{\gamma}))] & \text{diag}[H_x(H^{-1}(\vec{\gamma}))]\mathbf{T}(\vec{\gamma})\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) \end{bmatrix} \begin{bmatrix} w_1(H^{-1}(\vec{\gamma})) \\ w_2(H^{-1}(\vec{\gamma})) \end{bmatrix} = ik \begin{bmatrix} w_1(H^{-1}(\vec{\gamma})) \\ w_2(H^{-1}(\vec{\gamma})) \end{bmatrix}$$

where the evaluations of H_x, q, and q^* are placed on a diagonal to preserve the integrity of the system of equations. Thus, the operator $\mathcal{U}$ can be approximated by

$$i\begin{bmatrix} -\mathrm{diag}[H_x(H^{-1}(\vec{\gamma}))]\mathbf{T}(\vec{\gamma})\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) & \mathrm{diag}[q(H^{-1}(\vec{\gamma}))] \\ \mathrm{diag}[q^*(H^{-1}(\vec{\gamma}))] & \mathrm{diag}[H_x(H^{-1}(\vec{\gamma}))]\mathbf{T}(\vec{\gamma})\mathcal{D}\mathbf{T}^{-1}(\vec{\gamma}) \end{bmatrix} \tag{4.39}$$

and is an approximation of the AKNS operator appearing in (4.34). The 1,1, and 2,2 blocks approximate the derivative of a given function in a mapped Chebyshev basis, while the 1,2 and 2,1 blocks approximate the products of a given function with a potential function q or it's complex conjugate, q^*. Computation of the eigenvalues can be obtained by an eigenvalue solver, such as the QR Factorization, for example.

4.2. **Selecting a value for the parameter a.** We now return to our consideration of the fine-tuning parameter, a, referenced earlier in the manuscript. For the mapping function $H(x) = \tanh(ax)$, the parameter a impacts the interval $[-x, x]$ in which H changes most rapidly. Selecting a near 0 results in a wider interval than a value nearer 1, as is visible in Figure 1. For our method,

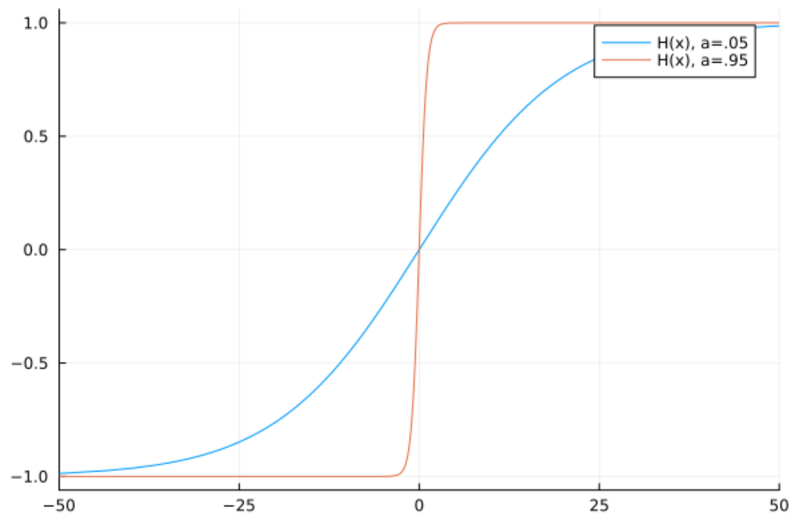

FIGURE 1.

the parameter a should be selected such that the intervals of $H(x)$ and the potential function $q(x)$ share the same"change" interval. Distributing a sufficient number of collocation points in the correct interval is essential for accuracy [9]. From [9], this is done by solving

$$\tanh(ax) = \alpha \quad \text{for } \alpha = 1 - \epsilon \tag{4.40}$$

for some $\epsilon > 0$ and a value $x = \xi$ corresponding to the interval $[-\xi, \xi]$ of change of the potential function q. Simple algebraic manipulation reveals

$$a = \frac{\tanh^{-1}(1-\epsilon)}{\xi}. \tag{4.41}$$

In order to align the intervals of q and H we demand that $H', q' \to 0$ as $x \to \pm\xi$. The desired accuracy informs the threshold for the interval. That is to say that one may choose to increase or decrease the interval such that $q'(\xi)$ is within some desired tolerance of zero.

Let us consider the potential function $q(x) = 1.8\,\mathrm{sech}(x)$. Choosing a tolerance of $10e{-}10$, we might choose the interval $[-25, 25]$ since $q'(\pm 25) = \mp 4.99966e - 11$. Taking $\epsilon = .01$ yields $a = .105866$, and the graph of both H and q show that their intervals of change are aligned as in Figure 2.

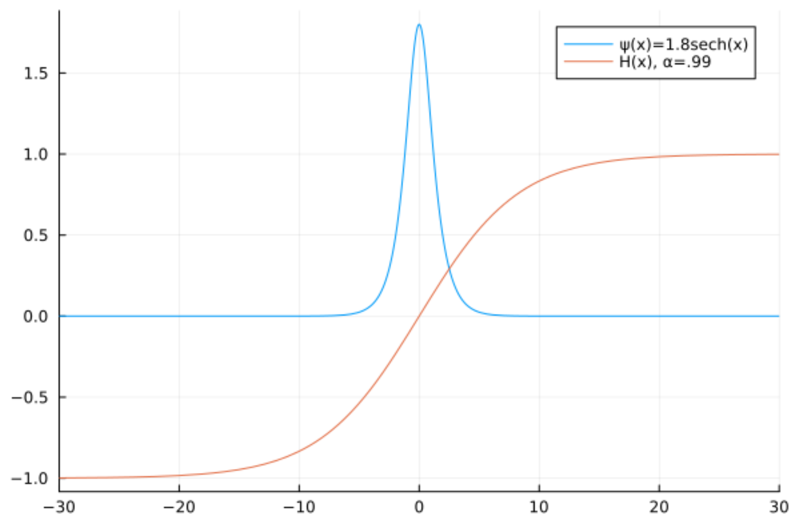

FIGURE 2.

5. EXAMPLES

All numerical examples reported are run on a MacBook Pro (2019) with a 2.6 GHz 6-Core Intel Core i7 processor, 16 GB 2400 MHz DDR4 memory running MacOS 13.4.1 (c). All code was written in Julia (v 1.8.1) and is included in Appendix A.

5.1. **Exact soliton.** The solitonic potential $q_s(x) = e^{-ix}\operatorname{sech}(x)$ has the single discrete eigenvalue $\lambda_1 = \frac{1}{2} + \frac{1}{2}i$ in the upper half plane, with its complex conjugate appearing in the lower half plane [9]. Settting $n = 200$ and a=0.1 per [9], yields the discrete spectrum value of $0.4999999999999999 + 0.5000000000000012i$ and $0.4999999999999999 - 0.5000000000000012i$) as in Figure 4.

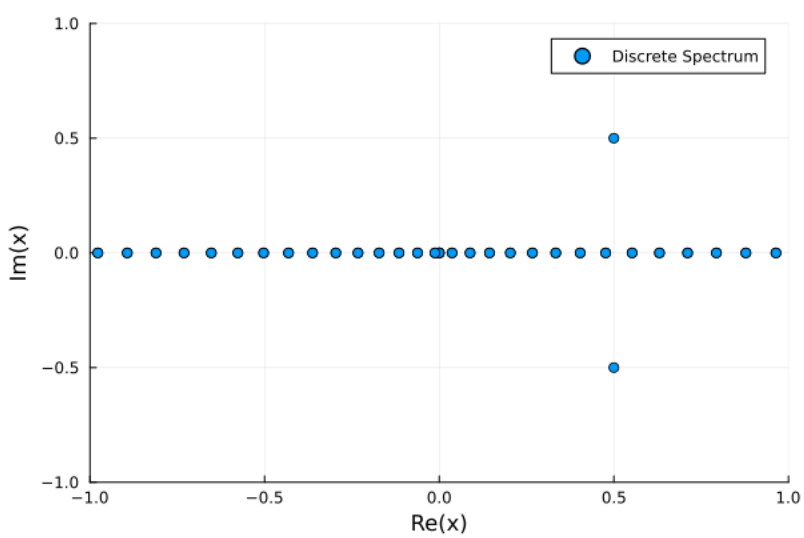

FIGURE 3. $q_s(x) = e^{-ix}\operatorname{sech}(x)$

5.2. **Satsuma-Yajima soliton ensemble.** The method is applied to the Zakharov-Shabat system with Satsuma-Yajima potential function $q(x) = 1.8\,\mathrm{sech}(x)$ which has four discrete eigenvalues: $\lambda_1 = 1.3i, \lambda_2 = .3i, \bar{\lambda}_1 = -1.3$, and $\bar{\lambda}_2 = -.3i$ [9]. Using 200 collocation points and setting $a = .15$ per [9] yields:

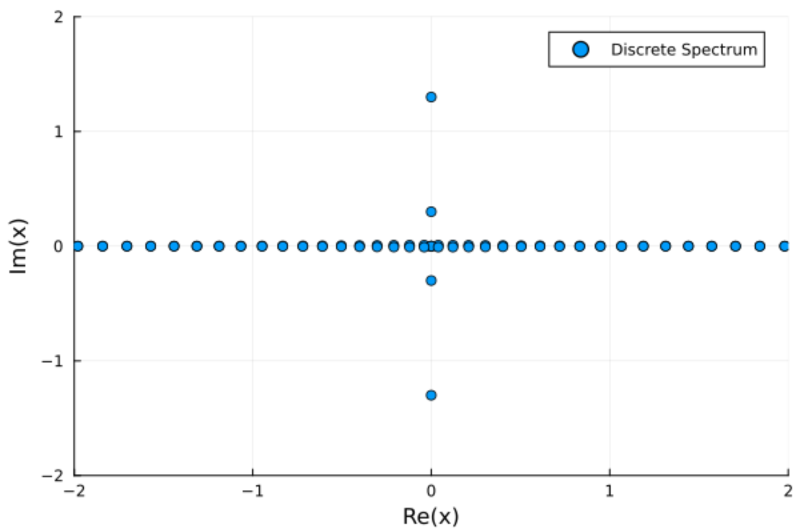

FIGURE 4. $q(x) = 1.8\,\mathrm{sech}(x)$

λ_1	$0.0 + 1.2999999999999927i$
λ_2	$0.0 + 0.2999999999999981i$
$\bar{\lambda}_1$	$0.0 - 1.2999999999999927i$
$\bar{\lambda}_2$	$0.0 - 0.2999999999999981i$

TABLE 3. Approximate discrete spectrum

6. DISCUSSION

The results above match the results presented in [9]. However, plotting all spectrum without restricting the plot range reveals spectral leakage at the boundaries, as in Figure 6. Varying the value of the fine-tuning parameter a minimizes the leakage but reduces the accuracy in the approximation of the discrete spectrum. Some consideration was given to the particular algorithms used for computing the eigenvalues, but the use of two different algorithms produced identical results. While this is not conclusive evidence that the algorithm is not involved, these results do suggest that the spectral leakage may be a shortcoming of the particular mapping function H. Thus, one area for future exploration is to search for a more suitable mapping function.
Another approach to computing the spectra is to utilize Hill's method outlined in [10]. This method involves expanding a suitable function in a Fourier series and solving the $(2N + 1)$-dimensional

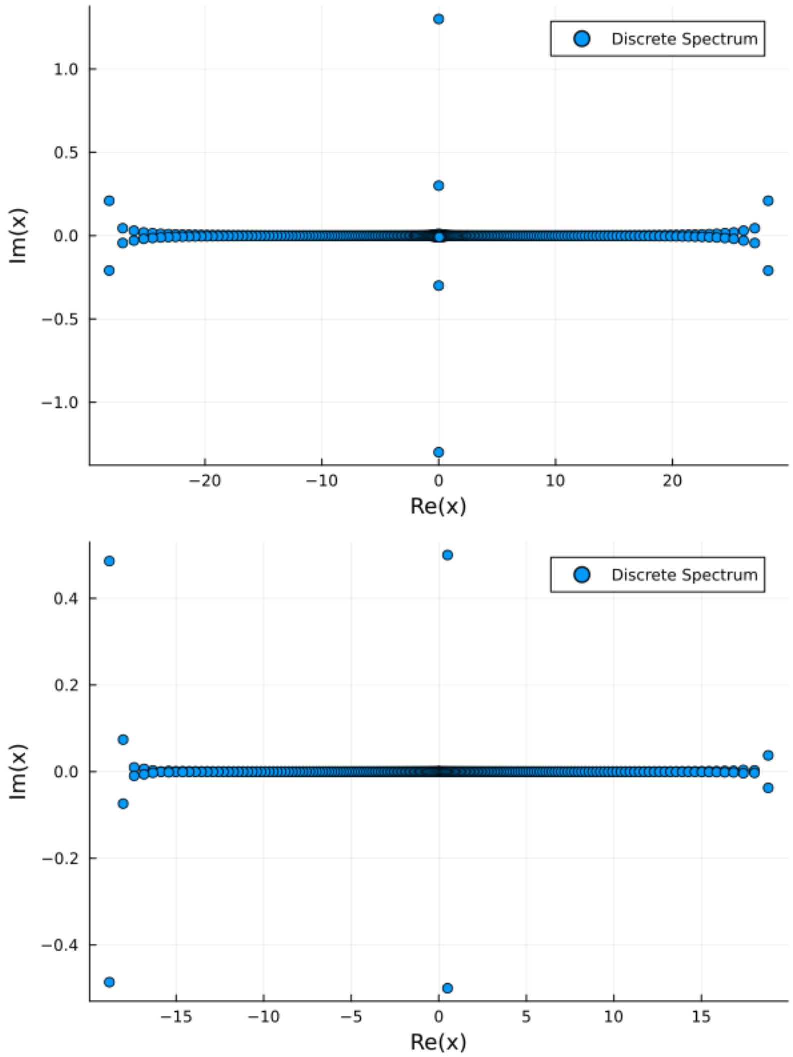

FIGURE 5.

matrix equation

$$\hat{\mathscr{L}}_N(\mu_k)\hat{\phi}_N = \lambda_N \hat{\phi}_N \tag{6.1}$$

where N is the number of Fourier modes and μ_k are evenly spaced between 0 and 2π. Solving this system for each eigenvalue yields a set of eigenvalues which is the approximate spectrum of the operator $\mathscr{L}$.

Another approach, which also utilizes Chebyshev series, can be found in [17]. This method utilizes ultraspherical polynomials to construct a discretized AKNS operator, but we believe that the Chebyshev approximation theory developed earlier in this manuscript enables an implementation of this method without converting to ultraspherical polynomials. That is to say that the AKNS operator

$$\begin{bmatrix} i\frac{\partial}{\partial x} & -iq(x) \\ -iq^*(x) & -i\frac{\partial}{\partial x} \end{bmatrix} \tag{6.2}$$

should have a suitable representation built from the differentiation and multiplication operators from §3.4, 3.5 respectively. This is an area for future exploration as several open questions regarding this method remain, including how the introduction of a suitable mapping function might impact this method.

7. Appendix A: Code

The following code has dependencies contained in the following libraries: LinearAlgebra, Plots, BlockArrays.

```
using LinearAlgebra
using Plots
using BlockArrays
```

The following functions are used in the computations of the above results.

```
function TMatrix(y)
    """ Returns the matrix of chebyshev polynomials evaluated at each point y in [-1,1].
    This function assumes a vector-valued y."""
    n=length(y) # set n equal to the length of the vector y
    C(k,y)=cos(k*acos(y)) # k-th Chebyshev polynomial at y
    T=zeros(n,n) # initialize an empty matrix
    u=range(1,n,step=1) # create range for use in array comp below
    T=[T[w,z]=C(w-1,y[z]) for w in u, z in u] # evaluate C for each entry of the matrix
    return Transpose(T) # return the transpose of T
end
```

```
function ChebPoints(n)
    """This function accepts an integer n, and returns n Chebyshev (extreme) collocation points."""
    r=range(1,step=1, length=n) # create range for use in list comprehension below
    t=[cos(((n-i)/(n-1))*pi) for i in r] # compute Chebyshev points and store in a vector
    return t
end
```

```
function Tinv(u)
    """This function accepts a vector of Chebyshev extreme points and returns
    the inverse of T by solving against the Identity matrix."""
    Tee=TMatrix(u)
    Tinv=Tee\I
    return Tinv
end
```

```
function BuildD(n)
    """This function accepts an integer input, and returns
    the D matrix which maps coefficients of the chebyshev approximation
    of a function to the coefficients for the derivative of the approximation
    of the function."""
    # Build matrix of weights, [1,2,2,...2]
    w=2*ones(n)
    w[1,1]=1
    W=Diagonal(w) # matrix of weights

    # Build D
    B=ones(n,n) # initialize a matrix of ones
    r=range(1,n,step=1) # create a range to use as column indices in Z
    Z=Diagonal(r.-1) # diagonal of column index minus 1
    D=UpperTriangular(Transpose(Z*B)) # creates an upper triangular matrix where
   each entry is its column minus 1
    D=Transpose(Transpose(D)*W)
    # place zeros in the matrix, D, based on column and row positions
    # such that odd columns have zeros in the odd rows,
    # and, even columns have zeros in the even rows.
    for j in r, k in r
        if j%2==0 && k%2==0
            D[j,k]=0
        end
        if j%2==1 && k%2==1
            D[j,k]=0
        end
    end
    return D
end
```

```
function BlockIt(A,B)
    """ This function takes two matrices A and B and returns the
    approximated Dirac/AKNS operator U."""
    C=conj(B)
    U=mortar(reshape([-A,C,B,A],2,2))
    return U
end
```

```
function BuildHankel(n,v)
    """ Build the Hankel operator for mult operator as in Olver/Townsend.
    n is the size of the system, v is a vector of coefficients of size 2n"""
    m=(2n-1)# coefficient indices for almost Hankel Operator
    Hank=zeros(n,n)
    r=range(1,n,step=1)
    for j in r, k in r
```

```
        if j+k-2<m+1
            Hank[j,k]=v[j+k-1]
        else
            Hank[j,k]=0
        end
    end
    Hank=vcat(Transpose(zeros(n)), Honk[2:n,:]) #replace first row with zeros
    return Hank
end
```

```
function BuildToeplitz(n,v)
    """Build the Toeplitz operator for mult operator as in olver/Townsend."""
    r=range(1,n,step=1)
    arr=range(2,n,step=1)
    Toeplitz=zeros(n,n)
    Toeplitz=[Toeplitz[j,k]=v[abs(k-j)+1] for j in r, k in r]+Diagonal(v[1]*ones(n
   ))
end
```

```
function BuildM(n,g)
    """ Build the multiplication matrix as in Olver/Townsend. This function does
   not
    include the tanh mapping for potential functions defined on the real line."""
    # get coefficients for function g
    length=2n
    y=ChebPoints(2n)
    F=GetF(y)
    coeffs=F*g.(y)

    Toeplitz=BuildToeplitz(n,coeffs)
    Hankel=BuildHankel(n,coeffs)
    M=.5*(Toeplitz+Hankel)
    return M
end
```

```
function BuildMReal(n,g)
    """Builds multiplication operator as in Olver/Townsend, and
    accounts for functions defined on the real line."""
    # get coefficients for function g
    length=2n
    y=ChebPoints(2n)
    F=GetF(y)
    coeffs=F*g.(h.(y)) # includes tanh mapping

    Toeplitz=BuildToeplitz(n,coeffs)
    Hankel=BuildHankel(n,coeffs)
```

```
    M=.5*(Toeplitz+Hankel)
    return M
end
```

For $q(x) = 1.8\,\mathrm{sech}(x)$:

```
q(x)=1.8*sech(x) # define potential function q
H(x)=tanh.(aa*x) # Maps real line to unit interval
h(z)=atanh.(z)/aa # Maps unit interval to real line
dH(x)=aa*sech.(aa*x)^2 # derivative of map function for chain rule
aa=.15 # fine-tuning parameter a
n=200 # number of collocation points

b=ChebPoints(n) # collocation points
D=BuildD(n) # differentiation matrix
T=TMatrix(b) # T matrix
F=Tinv(b) # inverse of T matrix
A=Diagonal(dH.(h.(b)))*tee*D*F  # 1,1 block
B=Diagonal(q.(h.(b))) # 2,2 block
U=BlockIt(A,B) # build AKNS operator
E=schur(U) # compute Schur decomposition

scatter(-im*E.values,label="Discrete Spectrum") # plot eigenvalues
```

For $q(x) = e^{-ix}\,\mathrm{sech}(x)$:

```
q(x)=exp(-im*x)*sech(x) # define potential function q
H(x)=tanh.(aa*x) # Maps real line to unit interval
h(z)=atanh.(z)/aa # Maps unit interval to real line
dH(x)=aa*sech.(aa*x)^2 # derivative of map function for chain rule
aa=.1 # fine-tuning parameter a
n=200 # number of collocation points

b=ChebPoints(n) # collocation points
D=BuildD(n) # differentiation matrix
T=TMatrix(b) # T matrix
F=Tinv(b) # inverse of T matrix
A=Diagonal(dH.(h.(b)))*tee*D*F  # 1,1 block
B=Diagonal(q.(h.(b))) # 2,2 block
U=BlockIt(A,B) # build AKNS operator
E=schur(U) # compute Schur decomposition

scatter(-im*E.values,label="Discrete Spectrum") # plot eigenvalues
```

References

[1] M. J. Ablowitz, D. J. Kaup, A. C. Newell, and H. Segur, "The inverse scattering transform-Fourier analysis for nonlinear problems," *Studies in Applied Mathematics,* **53** no. 4, 249–315, 1974.

[2] A. Amirhossein, R. Corless, M. Madhusoodan "Differentiation Matrices for Univariate Polynomials," *ArXiv:1809.05769v1*, 2018.

[3] C. Ballew and T. Trogdon "A Riemann-Hilbert Approach to Computing the Inverse Spectral Map For Measures Supported on Disjoint Intervals.," *ArXiv:2302.12930v1*, 2023.

[4] G. Baszenski, and M. Tasche, "Fast Polynomial Multiplication and Convolutions Related to the Discrete Cosine Transform," *Elsevier Science inc.*, 1997.

[5] M. Borghese, R. Jenkins, and K. D. T.-R. McLaughlin, "Long time asymptotic behavior of the focusing nonlinear Schrödinger equation," *Annales de L'Institut Henri Poincaré: Analyse Non Linéaire* **35**, 887–920, 2018.

[6] D. Borthwick, "Spectral Theory: Basic Concepts and Applications," *Springer Graduate Texts in Mathematics,*, 2020.

[7] J. P. Boyd,"Chebyshev and Fourier Spectral Methods," *Dover Publications, inc.*, 2000.

[8] C. W. Clenshaw, "Chebyshev Series for Mathematical Functions," *Mathematical Tables Volume 5* , 1962.

[9] S. Cui and Z. Wang, "Efficient method for calculating the eigenvalues of the Zakharov-Shabat system", `ArXiv:2212.13490`, 2023.

[10] B. Deconinck and J. N. Kutz, "Computing spectra of linear operators using the Floquet-Fourier-Hill method," *Elsevier, Journal of Computational Physics,***219**,296-321, 2006.

[11] B. Deconinck, F. Kiyak, J. D. Carter and J. N. Kutz, " SpectrUW: a laboratory for the numerical exploration of spectra of linear operators," 2006.

[12] P. Deift and E. Trubowitz, "Inverse scattering on the line," *Comm. Pure Appl. Math.* **32** no. 2, 121–252, 1979.

[13] T. Koornwinder, "Orthogonal polynomials, a short introduction," *arXiv:1303.2825v2 [math.CA] 11 Nov 2021*, 2021.

[14] B. Malomed, "Nonlinear Schrödinger Equations," in Alwyn Scott (ed.), *Encyclopedia of Nonlinear Science*, New York: Routledge, pp. 639–643, 2005.

[15] J. Mason, and D. Handsomb, "Chebyshev Polynomials," *Chapman & Hall/CRC*, 2003.

[16] F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, and B. V. Saunders, eds., NIST Digital Library of Mathematical Functions, http://dlmf.nist.gov/, version 1.0.17, 2017.

[17] S. Olver and A. Townsend, "A Fast and Well-Conditioned Spectral Method," *arXiv:1202.1347v2 [math.NA] 15 Aug 2012*, 2012.

[18] D. H. Peregrine, "Water waves, nonlinear Schrödinger equations and their solutions," *J. Austral. Math. Soc. Ser. B* **25**, 16–43, 1983.

[19] M. Sezer, and M. Kaynak, "Chebyshev polynomial solutions of linear differential equations," *Int. J. Math. Educ. Sci. Technol. Volume 27 No. 4* , 607–618, 1996.

[20] G. Szegő, "Orthogonal Polynomials," *American Mathematical Society Colloquium Publications Volume XXIII* , 1939.

[21] L. Trefethen, "Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations," `unpublished text,` 1996.

[22] T. Trogdon and S. Olver, "Numerical inverse scattering for the focusing and defocusing nonlinear Schrodinger equations," *Proceedings of the Royal Society A* **469** no. 2149, 20120330–20120330, 2012.

[23] T. Trogdon, "Scattering and inverse scattering for the AKNS system: A rational function approach," *Studies in Applied Mathematics* **147** no. 4, 1443–1480, 2021.

[24] V. E. Zakharov, "Stability of periodic waves of finite amplitude on the surface of a deep fluid," *Journal of Applied Mechanics and Technical Physics* **9** no. 2, 190–194. Originally in: *Zhurnal Prikdadnoi Mekhaniki i Tekhnicheskoi Fiziki* **9** no. 2, 86–94, 1968.

[25] V. E. Zakharov and A. B. Shabat, "Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media," *Journal of Experimental and Theoretical Physics* **34** no. 1, p. 62, 1972.

Email address: `allhanes@mail.uc.edu`