Uniform and high-order practical implementation of Sturm–Liouville problems via Fer streamers

Alberto Gil C. P. Ramos*

December 11, 2015

Abstract

At present, Fer streamers are the only methods for Sturm–Liouville problems that possess error bounds which: *i*) hold uniformly over the entire eigenvalue range, and, *ii*) can attain arbitrary high-order. Following their truncation and discretization using Lie-algebraic techniques and multivariate oscillatory quadrature in (Ramos and Iserles, 2015, *Numerische Mathematik*, DOI: 10.1007/s00211-014-0695-0) and (Ramos, 2015*a*, Submitted), the current manuscript presents their practical implementation with uniform global orders 4, 7, 10 and 13. In particular, the practical implementation is shown to benefit from a 'reduced' Hall basis which leads to a decreased amount of linear algebra in the given approach.

In addition, the current paper is accompanied by supplementary material in the form of a MATLAB package, which realizes this practical implementation, and can be downloaded online.

1 Introduction

This paper is about the practical implementation of a new algorithm to approximate the eigenvalues and eigenfunctions $\{(\lambda_j, y_{\lambda_j})\}_{j\in\mathbb{Z}_0^+}$ of regular Sturm–Liouville problems in Liouville's normal form

$$-y''_{\lambda_j}(t) + q(t)y_{\lambda_j}(t) = \lambda_j y_{\lambda_j}(t), \quad t\in[a,b], \quad a,b,\lambda_j\in\mathbb{R}, \quad y_{\lambda_j}:[a,b]\to\mathbb{R}, \tag{1.1}$$

with continuous and piecewise analytic potentials $q:[a,b]\to[q_{\min},q_{\max}]\subseteq\mathbb{R}$ and self-adjoint separated boundary conditions

$$\begin{aligned} \alpha_1 y_{\lambda_j}(a) + \alpha_2 y'_{\lambda_j}(a) = 0, \quad \alpha_1,\alpha_2\in\mathbb{R}, \quad \alpha_1^2+\alpha_2^2>0,\\ \beta_1 y_{\lambda_j}(b) + \beta_2 y'_{\lambda_j}(b) = 0, \quad \beta_1,\beta_2\in\mathbb{R}, \quad \beta_1^2+\beta_2^2>0. \end{aligned} \tag{1.2}$$

These are classical problems in computational mathematics which lie on the interface between numerical analysis and spectral theory with important applications in physics and chemistry not least in the approximation of energy levels and wave functions of quantum systems (Pryce, 1993; Zettl, 2005).

The main feature that separates this new algorithm from every existing technique is that it is accompanied by error bounds which: *i*) hold uniformly for every eigenvalue, and, *ii*) can attain arbitrary high-order. The output is then an algorithm which is mathematically guaranteed to be uniformly precise throughout all orders of magnitude of the λ_j.

The mathematical analysis that guarantees such novel error bounds is centered on a new set of ideas based on Fer streamers (Ramos and Iserles, 2015; Ramos, 2015*a*), which revolve around Lie-algebraic techniques and multivariate oscillatory quadrature.

While these papers (Ramos and Iserles, 2015; Ramos, 2015*a*) focused on developing Fer streamers with uniform truncation and discretization error bounds, the current paper instead explains how to capitalize on a 'reduced' Hall basis to yield an efficient implementation, which has now been realized in the form of a MATLAB package, with uniform global orders 4, 7, 10 and 13, presented for the first time in this paper.

*Cambridge Centre for Analysis, Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Wilberforce Road, Cambridge CB3 0WA, United Kingdom; a.g.c.p.ramos@maths.cam.ac.uk

1.1 Uniform but low-order error bounds

The Piecewise Constant Method (PCM) (Pruess, 1973; Paine and de Hoog, 1980; Marletta and Pryce, 1992; Pruess and Fulton, 1993) is among the earliest techniques used to approximate the eigenvalues of regular Sturm–Liouville problems. True for this day, it remains one of the few techniques mathematically guaranteed to approximate every eigenvalue uniformly well.

The underlying principle of the PCM consists in two approximations. First, approximate the eigenvalues of the original problem with those of a 'new' problem, defined as the original problem except q is replaced by a piecewise constant interpolation $\tilde{q} : [a, b] \to \mathbb{R}$. Second, compute the eigenvalues of the 'new' problem. The motivation is two-fold: on one hand, the first approximation is easily controlled with perturbation techniques, and, on the other hand, unlike the original problem, the 'new' problem is numerically tractable, up to prescribed tolerance.

That the PCM is sure to approximate all eigenvalues equally well, has long since been established (Pruess, 1973; Paine and de Hoog, 1980). More concretely, for each numerical mesh $c_0 := a < c_1 < \cdots < c_{m-1} < c_m := b$, $h_k := c_{k+1} - c_k$, $h_{\max} := \max\{h_0, \ldots, h_{m-1}\}$ such that $\tilde{q}|_{(c_k, c_{k+1})}$ is constant and interpolates q at some point in $[c_k, c_{k+1}]$, there exist error bounds in the uniform regime

$$h_{\max} \to 0^+, \text{ uniformly w.r.t. } \lambda \in \mathbb{R}, \tag{1.3}$$

where the constants in the big $\mathcal{O}$ notation are bounded independently of $\lambda \in \mathbb{R}$, that provide a convergence rate $d_1 h_{\max}^1$, where $d_1 > 0$ does not depend on $\lambda \in \mathbb{R}$. Among them, (Pruess, 1973, Theorem 1) controls the relative error with an error bound that yields uniform order 1 in the sense of (1.3). Another classical result is (Paine and de Hoog, 1980, Corollary 3.1), which controls the absolute error with an error bound that gives uniform order 1, again with respect to (1.3).

The uniform character of the error bounds for the PCM makes it an algorithm guaranteed to approximate every eigenvalue, at the same price. However, the convergence rate is rather low, which in practice means fine meshes and many function evaluations of q in order to construct $\tilde{q}$, where a popular choice is $\tilde{q}(t) := q((c_k + c_{k+1})/2)$ for t in (c_k, c_{k+1}). As function evaluations of q can be of considerable cost in practice, this creates a problem.

For this reason, there has been a lot of effort to develop algorithms with high convergence rate. Unfortunately, as discussed in the next subsection, the new results along this line of research have lost the uniform property of the error bounds in favor of a high convergence rate limited to 'small' or 'large' eigenvalues, so called 'asymptotic' rate.

1.2 High-order but non-uniform error bounds

The concept of 'asymptotic' order valid for 'small' or 'large' eigenvalues, alluded to in the previous subsection, was introduced first in (Ixaru et al., 1997, 1999; Ixaru, 2000; Ledoux et al., 2004, 2005) to analyze the convergence of the Constant Perturbation Method (CPM). The guiding rule of the CPM consists of two truncations and one discretization. The first truncation is to approximate $q(c + ht)$, $t \in (0, 1)$ by the p-th degree polynomial of the Legendre series partial sum

$$\tilde{q}_p|_{(c,c+h)}(c + ht) := \sum_{j=0}^{p} q|_{(c,c+h)_j} P_j(2t - 1),$$

$$q|_{(c,c+h)_j} := (2j + 1) \int_0^1 q(c + ht) P_j(2t - 1) dt.$$

The second truncation is based on the PCM with $\lfloor \frac{2}{3} p \rfloor + 1$ corrections, rather than infinitely many. Together the two truncations form an approximation known as the CPM$[p, \lfloor \frac{2}{3} p \rfloor + 1]$. The ethos of the asymptotic order in (Ixaru et al., 1997) and references that follow, is to investigate the truncation error of the CPM$[p, \lfloor \frac{2}{3} p \rfloor + 1]$

in the asymptotic regimes:

$$\lambda \text{ fixed and } h \to 0^+, \tag{1.4}$$

$$h \text{ fixed and } \lambda \to +\infty. \tag{1.5}$$

The approach corresponds directly to an analysis based on Taylor series or on asymptotic expansions. The first then provides some information about the behavior of the approximations for 'small' eigenvalues whereas the second gives some insight into 'large' eigenvalues. Unfortunately, the underlying issue at play is that these asymptotic regimes are not well suited to study 'intermediary' eigenvalues, which require the control of (λ, h) instead of only either h (with fixed λ) or λ (with fixed h). Without being too precise, one of the difficulties here is that the power broker behind the scene is in fact

$$\begin{aligned} z_{\lambda,h} &:= \left(q|_{(c,c+h)_0} - \lambda\right) h^2, \\ \varpi_{\lambda,h} &:= 2\sqrt{-z_{\lambda,h}} = 2h\sqrt{\lambda - \frac{\int_c^{c+h} q\,(\xi)\,d\xi}{h}}, \end{aligned} \tag{1.6}$$

which appears, one way or another, as the argument of oscillatory functions (Ixaru et al., 1997, 1999; Ixaru, 2000; Ledoux et al., 2004, 2005). For instance, as (Ixaru et al., 1997, p. 294) puts it:

> " As λ is a free parameter (...) and we want to analyze the error behaviour at arbitrary λ, the whole range of $z_{\lambda,h}$'s has to be investigated. However, we can cover only two relevant extreme cases: $|z_{\lambda,h}|$ small and $z_{\lambda,h}$ large and negative. "

These extreme cases correspond precisely to the asymptotic regimes (1.4)–(1.5) (Ixaru et al., 1997, p. 295, p. 298). In particular, if one calls upon Taylor series in (1.4) as customary in the literature, then the factor λ and its powers populate the constants in the big $\mathcal{O}$ notation in the error bounds, making them useless for 'intermediary' or 'large' eigenvalues, unless one takes a prohibitively tiny step size, which is not an option in practice. Likewise, if one invokes asymptotic expansions in (1.5), one depends on $z_{\lambda,h} \ll -1$, making the error bounds unusable this time for 'small' or 'intermediary' eigenvalues. The truncation error of the CPM$[p, \lfloor \frac{2}{3}p \rfloor + 1]$ in these extreme cases is then controlled by (Ixaru et al., 1997, p. 294):

- $r_{\lambda,p} h^{2p+2}$ w.r.t. (1.4), where $\lim_{\lambda\to+\infty} r_{\lambda,p} = +\infty$ & $\lim_{\lambda\to+\infty} r_{\lambda,p+1}/r_{\lambda,p} = +\infty$, and,
- $s_p h^p/\sqrt{\lambda}$ w.r.t. (1.5).

Finally, as $q|_{(c,c+h)_j}$, $j \in \{0, 1, \ldots, p\}$ are in general unavailable, they are approximated by quadrature, which forms the discretization step. For this, $q(c + h\cdot)$ is evaluated at p Gauss points in $(0, 1)$ (Ledoux et al., 2004, p. 158) to form a quadrature with error h^{2p} for $j = 0$ and h^p for $j > 0$. Since $q|_{(c,c+h)_j}$ are always multiplied by h^2, this yields local error proportional to h^{p+2}, which caps the truncation bounds and yields the discretization bounds CPM$\{p + 2, p\}$ that behave as:

- $\tilde{r}_{\lambda,p} h^{p+2}$ w.r.t. (1.4), where $\lim_{\lambda\to+\infty} \tilde{r}_{\lambda,p} = +\infty$ & $\lim_{\lambda\to+\infty} \tilde{r}_{\lambda,p+1}/\tilde{r}_{\lambda,p} = +\infty$, and,
- $\tilde{s}_p h^p/\sqrt{\lambda}$ w.r.t. (1.5).

Since the introduction of the CPM (Ixaru et al., 1997, 1999; Ixaru, 2000; Ledoux et al., 2004, 2005), there exist now different techniques that also attain asymptotic high-order. These include the modified Neumann integral series (Iserles, 2004*a*; Ledoux and Daele, 2010), which, as (Degani, 2004; Degani and Schiff, 2006) point out, are closely related to the CPM. Yet another approach is that of the modified or right correction Magnus Lie-group/Lie-algebra integral series (Iserles, 2004*b*; Degani, 2004; Degani and Schiff, 2006; Ledoux et al., 2010). These again are centered around the asymptotic regimes (1.4)–(1.5) with error bounds limited to 'small' or 'large' eigenvalues.

Unlike the above results, the next subsection presents a new set of ideas based on Fer streamers (Ramos and Iserles, 2015; Ramos, 2015*a*) with which the whole range of $z_{\lambda,h}$'s is investigated. Unrestricted to the extreme cases described in the quote above, the output is then the first algorithm that possesses error bounds which can attain arbitrary high-order and hold uniformly for all 'small', 'intermediary' and 'large' eigenvalues.

1.3 Uniform and high-order error bounds

As mentioned above, unlike previous techniques (Ixaru et al., 1997, 1999; Ixaru, 2000; Ledoux et al., 2004, 2005; Iserles, 2004*a*; Ledoux and Daele, 2010; Iserles, 2004*b*; Degani, 2004; Degani and Schiff, 2006; Ledoux et al., 2010), Fer streamers (Ramos and Iserles, 2015; Ramos, 2015*a*) attain high-order without compromising the uniform property of the error bounds. Compared with the PCM, they are thus also mathematically guaranteed to be uniformly precise but are not restricted to low-order.

At the heart of this advancement, Fer streamers are based on a completely new approach via Fer expansions, which is more algebraic in nature.

On this note, one of the key points in Fer streamers is that they capitalize on the recursive nature of Fer expansions and exploit the low-dimensionality of a certain Lie-algebra to sum up the infinite sums in Fer expansions, in closed-form! This closed-form then makes it possible to bypass the approximation of each infinite sum in Fer expansions by its first partial sum — an approximation known to yield an error that grows with λ. As it turns out, by circumventing this approximation, Fer streamers lead to an entirely new truncation and discretization of Fer expansions, with error bounds that hold equally well for all λ.

Another key element in Fer streamers is that their analysis calls upon Taylor series only for bounded functions with bounded derivatives with bounds independent of λ and abandons asymptotic expansions altogether for $z_{\lambda,h} \ll -1$.

Because of all this, the asymptotic regimes (1.4)–(1.5) do not even appear in the analysis of Fer streamers. Instead, once every algebraic feature is taken into account, it is the uniform regime

$$h_{\max} \to 0^+, \text{ uniformly w.r.t. } \lambda \in [q_{\max} - h_{\max}^{-2}, +\infty), \tag{1.7}$$

that emerges naturally and it is with respect to (1.7) that Fer streamers derive error bounds, where the constants in the big $\mathcal{O}$ notation are bounded independently of $\lambda \in [q_{\max} - h_{\max}^{-2}, +\infty)$. In particular, given $p+1 \in \{4, 7, 10, 13, \ldots\}$, Fer streamers evaluate $q(a)$ and $q(c_k + h_k \cdot)$ at p points in $(0, 1]$ and yield total global error bounds $d_p h_{\max}^{p+1}$, with $d_p > 0$ independent of $\lambda \in [q_{\max} - h_{\max}^{-2}, +\infty)$.

1.4 Geometric integration

Apart from the different type of error bounds that separate these techniques, there is a geometric property left intact by some but not all, which pertains to the solution of the initial value problem

$$\boldsymbol{Y}'_\lambda(t) = \begin{bmatrix} 0 & 1 \\ q(t) - \lambda & 0 \end{bmatrix} \boldsymbol{Y}_\lambda(t), \quad t \in [a, b], \quad a, b, \lambda \in \mathbb{R}, \quad \boldsymbol{Y}_\lambda : [a, b] \to \mathbb{R}^{2\times 2}, \tag{1.8}$$

with the initial condition

$$\boldsymbol{Y}_\lambda(a) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{1.9}$$

This initial value problem appears in all of the above techniques, in transcendental characterizations (Zettl, 2005, p. 43–47) and in shooting methods (Pryce, 1993, p. 88–116), which can be used, together with a root-finding algorithm, to approximate the eigenvalues and the eigenfunctions of the boundary value problem (1.1)–(1.2), provided one can approximate the solution of the initial value problem (1.8)–(1.9) at $t = c_{k+1}$. In particular, since the matrix in (1.9) belongs to the Lie group

$$\mathrm{SL}(2, \mathbb{R}) := \left\{ \boldsymbol{U} \in \mathbb{R}^{2\times 2} : \det(\boldsymbol{U}) = 1 \right\}$$

and the matrix in (1.8) lies in the Lie algebra

$$\mathfrak{sl}(2, \mathbb{R}) := \left\{ \boldsymbol{V} \in \mathbb{R}^{2\times 2} : \mathrm{tr}(\boldsymbol{V}) = 0 \right\}$$

the solution possesses the geometric property

$$\boldsymbol{Y}_\lambda([a, b]) \subseteq \mathrm{SL}(2, \mathbb{R}).$$

Regarding the above methods, the Fer streamers approach (Ramos and Iserles, 2015; Ramos, 2015*a*) preserves this geometric feature as does the PCM (Pruess, 1973; Paine and de Hoog, 1980; Marletta and Pryce, 1992; Pruess and Fulton, 1993) and the modified or right correction Magnus (Iserles, 2004*b*; Degani, 2004; Degani and Schiff, 2006; Ledoux et al., 2010), but not the CPM (Ixaru et al., 1997, 1999; Ixaru, 2000; Ledoux et al., 2004, 2005) nor the modified Neumann (Iserles, 2004*a*; Ledoux and Daele, 2010).

1.5 Notation

Italic lower case letters denote scalars, boldface lower case letters denote column vectors and boldface upper case letters denote matrices, e.g., c, $\boldsymbol{c}$ and $\boldsymbol{C}$, respectively. The transpose of $\boldsymbol{C}$ is denoted by $\boldsymbol{C}^\top$. Given two matrices, with the same dimensions, $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{C}^{j_1 \times j_2}$, the Hadamard product between $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ is denoted by $\odot$ and is the unique element in $\mathbb{C}^{j_1 \times j_2}$ defined by

$$[\boldsymbol{M}_1 \odot \boldsymbol{M}_2]_{j_3,j_4} := [\boldsymbol{M}_1]_{j_3,j_4} [\boldsymbol{M}_2]_{j_3,j_4} .$$

2 Fer streamers

Having already a bird's-eye view of the novel approach to Sturm–Liouville problems via Fer streamers in Subsections 1.3 and 1.4, the current section summarizes the new set of ideas that surround Fer streamers, necessary for their practical implementation with uniform global orders 4, 7, 10 and 13 with respect to (1.7). For additional information, including truncation and discretization error bounds, the reader may wish to consult (Ramos and Iserles, 2015; Ramos, 2015*a*).

In a nutshell, the Fer streamers' approach sets out to approximate the solution of the initial value problem (1.8)–(1.9) at some point $t = c_{k+1}$. To this end, the approach commences from the well-known Fer expansions integral series summarized below in Theorem 2.1.

Definition 2.1. *Let* $\boldsymbol{X}, \boldsymbol{Y} \in \mathfrak{sl}(2,\mathbb{R})$, $l \in \mathbb{Z}^+$ *and* $t \in [c_k, c_{k+1}]$, *and define*

$$\begin{aligned}
\rho(\boldsymbol{X}) &:= 2\sqrt{-\det(\boldsymbol{X})},\\
e^{\boldsymbol{X}} &:= \cosh(\rho(\boldsymbol{X})/2) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\sinh(\rho(\boldsymbol{X})/2)}{\rho(\boldsymbol{X})/2} \boldsymbol{X},\\
\mathrm{ad}_{\boldsymbol{X}} \boldsymbol{Y} =: [\boldsymbol{X}, \boldsymbol{Y}] &:= \boldsymbol{X}\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X},\\
\boldsymbol{B}_{\lambda,0}(c_k, t) &:= \begin{bmatrix} 0 & 1 \\ q(t) - \lambda & 0 \end{bmatrix},\\
\boldsymbol{D}_{\lambda,0}(c_k, t) &:= (t - c_k) \begin{bmatrix} 0 & 1 \\ \frac{\int_{c_k}^t q(\xi) d\xi}{t - c_k} - \lambda & 0 \end{bmatrix},\\
\boldsymbol{B}_{\lambda,l}(c_k, t) &:= \sum_{j=1}^{\infty} (-1)^j \frac{j}{(j+1)!} \mathrm{ad}^j_{\boldsymbol{D}_{\lambda,l-1}(c_k,t)} \boldsymbol{B}_{\lambda,l-1}(c_k, t),\\
\boldsymbol{D}_{\lambda,l}(c_k, t) &:= \int_{c_k}^t \boldsymbol{B}_{\lambda,l}(c_k, \xi) d\xi.
\end{aligned}$$

Theorem 2.1 (Fer, 1958)**.** *The solution of the initial value problem* (1.8)–(1.9) *is given by the Fer expansions*

$$\begin{aligned}
\boldsymbol{F}_\lambda(c_k, c_{k+1}) &:= e^{\boldsymbol{D}_{\lambda,0}(c_k,c_{k+1})} e^{\boldsymbol{D}_{\lambda,1}(c_k,c_{k+1})} e^{\boldsymbol{D}_{\lambda,2}(c_k,c_{k+1})} \cdots,\\
\boldsymbol{Y}_\lambda(c_{k+1}) &= \boldsymbol{F}_\lambda(c_k, c_{k+1}) \cdots \boldsymbol{F}_\lambda(c_1, c_2) \boldsymbol{F}_\lambda(a, c_1).
\end{aligned}$$

With these integral series in mind, the Fer streamers' approach starts off by using the recursive nature of Fer expansions together with the low-dimensionality of $\mathfrak{sl}(2,\mathbb{R})$ to sum up the infinite sums in Definition 2.1 in closed-form, as presented below in Theorem 2.2. These closed-form expressions, named 'Fer streamers', turn out to be essential to flesh out the magnitude and behaviour of the terms in Fer expansions, required for their integration in practice. In particular, they yield the representation of the first Fer streamer presented in Remark 2.1 below, which is one of the cornerstones in (Ramos and Iserles, 2015; Ramos, 2015*a*).

Definition 2.2. *Let $z \in \mathbb{C}$, and define*

$$\varphi(z) := \frac{\cosh(z) - 1 - z\sinh(z)}{z^2}, \qquad \phi(z) := \frac{z\cosh(z) - \sinh(z)}{z^3}.$$

Theorem 2.2 (Ramos and Iserles, 2015)**.** *If $l \in \mathbb{Z}^+$ and $t \in [c_k, c_{k+1}]$, then the infinite sums appearing in the terms of the Fer expansions of the initial value problem* (1.8)–(1.9) *are given in closed-form by the 'Fer streamers'*

$$\begin{aligned}\boldsymbol{B}_{\lambda,l}(c_k,t) = {}& \varphi\left(\rho\left(\boldsymbol{D}_{\lambda,l-1}(c_k,t)\right)\right)\operatorname{ad}_{\boldsymbol{D}_{\lambda,l-1}(c_k,t)}\boldsymbol{B}_{\lambda,l-1}(c_k,t)\\ &+ \phi\left(\rho\left(\boldsymbol{D}_{\lambda,l-1}(c_k,t)\right)\right)\operatorname{ad}^2_{\boldsymbol{D}_{\lambda,l-1}(c_k,t)}\boldsymbol{B}_{\lambda,l-1}(c_k,t).\end{aligned}$$

Definition 2.3. *Let $\boldsymbol{X} \in \mathfrak{sl}(2,\mathbb{R})$ and $\boldsymbol{x} \in \mathbb{R}^{3\times 1}$, and define*

$$\begin{aligned}\boldsymbol{\pi}\big(\boldsymbol{X}\big) &:= \begin{bmatrix}[\boldsymbol{X}]_{1,1} & [\boldsymbol{X}]_{1,2} & [\boldsymbol{X}]_{2,1}\end{bmatrix}^\top, & \boldsymbol{\pi}^{-1}\big(\boldsymbol{\pi}\big(\boldsymbol{X}\big)\big) &= \boldsymbol{X},\\ \boldsymbol{\pi}^{-1}\big(\boldsymbol{x}\big) &:= \begin{bmatrix}[\boldsymbol{x}]_{1,1} & [\boldsymbol{x}]_{2,1}\\ [\boldsymbol{x}]_{3,1} & -[\boldsymbol{x}]_{1,1}\end{bmatrix}, & \boldsymbol{\pi}\big(\boldsymbol{\pi}^{-1}\big(\boldsymbol{x}\big)\big) &= \boldsymbol{x}.\end{aligned}$$

Remark 2.1 (Ramos and Iserles, 2015)**.** *As an example, since*

$$\boldsymbol{\pi}\left(\boldsymbol{B}_{\lambda,0}(c_k,t)\right) = \begin{bmatrix}0 & 1 & q(t)-\lambda\end{bmatrix}^\top$$

we have that

$$\rho\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right) = 2(t-c_k)\sqrt{\frac{\int_{c_k}^t q(\xi)d\xi}{t-c_k} - \lambda} \tag{2.1}$$

and Theorem 2.2 yields

$$\boldsymbol{\pi}\left(\boldsymbol{B}_{\lambda,1}(c_k,t)\right) = \begin{bmatrix}\varphi\left(\rho\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)\right)\frac{q(t)-\frac{\int_{c_k}^t q(\xi)d\xi}{t-c_k}}{t-c_k}(t-c_k)^2\\ -2\phi\left(\rho\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)\right)\frac{q(t)-\frac{\int_{c_k}^t q(\xi)d\xi}{t-c_k}}{t-c_k}(t-c_k)^3\\ \frac{1}{2}\phi\left(\rho\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)\right)\rho^2\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)\frac{q(t)-\frac{\int_{c_k}^t q(\xi)d\xi}{t-c_k}}{t-c_k}(t-c_k)\end{bmatrix}.$$

Remark 2.2. *As touched upon in Section 1,* (1.6)*, a quantity which intertwines λ and h, always appears in the numerical solution of Sturm–Liouville problems, one way or another, as the argument of oscillatory functions. Here, with Fer streamers, it relates to* (2.1) *as it takes the form*

$$\rho\left(\boldsymbol{D}_{\lambda,0}(c,c+h\sigma)\right) = i\sigma\varpi_{\lambda,h}\sqrt{\frac{\lambda - \frac{\int_c^{c+h\sigma} q(\xi)d\xi}{h\sigma}}{\lambda - \frac{\int_c^{c+h} q(\xi)d\xi}{h}}}, \quad \sigma \in [0,1].$$

More concretely, Remark 2.1 places it once again as the argument of oscillatory functions, this time those being

$$x \mapsto \varphi(ix), \quad x \mapsto -2\phi(ix), \quad x \mapsto \frac{1}{2}\phi(ix)(ix)^2.$$

In particular, if both $\varpi_{\lambda,h}$ and λ are large and positive then $\boldsymbol{B}_{\lambda,1}(c, c+h\sigma)$ is an highly oscillatory matrix function for $\sigma \in [0,1]$. Central to this work, this realization arises from having written the infinite series in Definition 2.1 in closed-form via Fer streamers in Theorem 2.2, since otherwise it would remain unnoticed.

With the closed-form from Remark 2.1 in hand, Definition 2.4 below introduces $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k t)$ as a means to expose the fine and coarse scales of $\boldsymbol{B}_{\lambda,1}(c_k, c_k + h_k t)$. These are made precise with Theorems 2.3, 2.4 and 2.5 below, which depict the magnitude and behaviour of $\boldsymbol{B}_{\lambda,1}(c_k, c_k + h_k t)$ through $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k t)$. In particular, the magnitude of $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k t)$ is $\mathcal{O}(1)$ with respect to (1.7) and the behaviour of $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k t)$ changes with $\lambda \in [q_{\max} - h_{\max}^{-2}, +\infty)$ and varies from:

$$\lambda \in [q_{\max} - h_{\max}^{-2}, q_{\min} - 1] \cup [q_{\max} + 1, q_{\max} + h_{\max}^{-2}], \tag{2.2}$$

$$\lambda \in [q_{\min} - 1, q_{\max} + 1], \tag{2.3}$$

$$\lambda \in [q_{\max} + h_{\max}^{-2}, +\infty). \tag{2.4}$$

In particular, $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k \cdot)$ is:

- mildly exponential or oscillatory in (2.2) as made clear in Theorem 2.3,
- well-behaved in (2.3) as made explicit in Theorem 2.4,
- mildly to highly oscillatory in (2.4) as made explicit in Theorem 2.5.

This magnitude and behaviour are important to form an approximation of the matrix $\boldsymbol{B}_{\lambda,1}(c_k, c_k + h_k \cdot)$, which is necessary for the practical implementation of Fer streamers.

Definition 2.4. *Depending on* (2.2)–(2.4)*, let $\boldsymbol{B}^{fine}_{\lambda,1}(c_k, c_k + h_k t)$ be the unique element in $\mathfrak{sl}(2,\mathbb{R})$ such that*

$$\begin{aligned}
&\boldsymbol{\pi}\big(\boldsymbol{B}_{\lambda,1}(c_k, c_k + h_k t)\big) \\
&=: \boldsymbol{\pi}\big(\boldsymbol{B}^{fine}_{\lambda,1}(c_k, c_k + h_k t)\big) \odot \begin{cases} h_k \begin{bmatrix} h_k & h_k^2 & 1 \end{bmatrix}^\top & \Leftarrow (2.2), (2.3), \\ h_k \begin{bmatrix} \frac{1}{\sqrt{\lambda - q(c_k)}} & \frac{1}{\lambda - q(c_k)} & 1 \end{bmatrix}^\top & \Leftarrow (2.4). \end{cases}
\end{aligned}$$

In addition, let also

$$\begin{aligned}
&\omega_{\lambda,1}(c_k, t) := 2\,(t - c_k)\,\sqrt{\lambda - q\,(c_k)}, && r_{\lambda,1}(c_k, t) := \sqrt{\frac{\lambda - \frac{\int_{c_k}^t q(\xi)d\xi}{t - c_k}}{\lambda - q\,(c_k)}}, \\
&\epsilon_{\lambda,1}(c_k, t) := \omega_{\lambda,1}(c_k, t)\,(r_{\lambda,1}(c_k, t) - 1)\,, && s_{\lambda,1}(c_k, t) := \omega_{\lambda,1}(c_k, t)\epsilon_{\lambda,1}(c_k, t).
\end{aligned}$$

Theorem 2.3 (Ramos, 2015*a*)**.** *If* (2.2) *then $\omega_{\lambda,1}(c_k, c_{k+1}) \in [0, 2i]$ or $\omega_{\lambda,1}(c_k, c_{k+1}) \in \left[0, 2\sqrt{2}\right]$ and*

$$\begin{aligned}
\boldsymbol{\pi}\big(\boldsymbol{B}^{fine}_{\lambda,1}(c_k, c_k + h_k t)\big) = &\frac{1 - \cos\left(\omega_{\lambda,1}(c_k, c_{k+1})t\right)}{\left(\omega_{\lambda,1}(c_k, c_{k+1})t\right)^2} \begin{bmatrix} t \\ 0 \\ 0 \end{bmatrix} \odot t \begin{bmatrix} [\boldsymbol{f}_{\lambda,1}(c_k, c_k + h_k t)]_{1,1} \\ 0 \\ 0 \end{bmatrix} \\
&+ \cos\left(\omega_{\lambda,1}(c_k, c_{k+1})t\right) \begin{bmatrix} t \\ t^2 \\ 1 \end{bmatrix} \odot t \begin{bmatrix} [\boldsymbol{f}_{\lambda,1}(c_k, c_k + h_k t)]_{2,1} \\ [\boldsymbol{f}_{\lambda,1}(c_k, c_k + h_k t)]_{4,1} \\ [\boldsymbol{f}_{\lambda,1}(c_k, c_k + h_k t)]_{6,1} \end{bmatrix}
\end{aligned}$$

$$+\frac{\sin\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\omega_{\lambda,1}(c_k,c_{k+1})t}\begin{bmatrix}t\\0\\1\end{bmatrix}\odot t\begin{bmatrix}[\boldsymbol{f}_{\lambda,1}(c_k,c_k+h_kt)]_{3,1}\\0\\[\boldsymbol{f}_{\lambda,1}(c_k,c_k+h_kt)]_{7,1}\end{bmatrix}$$
$$+\phi\left(i\cdot\omega_{\lambda,1}(c_k,c_{k+1})t\right)\begin{bmatrix}0\\t^2\\0\end{bmatrix}\odot t\begin{bmatrix}0\\[\boldsymbol{f}_{\lambda,1}(c_k,c_k+h_kt)]_{5,1}\\0\end{bmatrix} \tag{2.5}$$

where

$$\boldsymbol{f}_{\lambda,1}(c_k,t):=\frac{q(t)-\frac{\int_{c_k}^t q(\xi)d\xi}{t-c_k}}{t-c_k}\frac{1}{r_{\lambda,1}^2(c_k,t)}$$
$$\times\begin{bmatrix}1\\(r_{\lambda,1}(c_k,t)-1)\left(r_{\lambda,1}(c_k,t)\varphi\left(i\cdot\epsilon_{\lambda,1}(c_k,t)\right)-\frac{1-\cos(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}^2(c_k,t)}\right)\\-r_{\lambda,1}(c_k,t)\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right)-\frac{r_{\lambda,1}(c_k,t)-1}{r_{\lambda,1}(c_k,t)}\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right)\\-2\left(r_{\lambda,1}(c_k,t)-1\right)\left(\frac{(r_{\lambda,1}(c_k,t)-1)^2}{r_{\lambda,1}(c_k,t)}\phi\left(i\cdot\epsilon_{\lambda,1}(c_k,t)\right)+\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right)\\-\frac{2}{r_{\lambda,1}(c_k,t)}\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right)+r_{\lambda,1}(c_k,t)s_{\lambda,1}(c_k,t)\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right)\\\frac{1}{2}r_{\lambda,1}^2(c_k,t)\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right)-\frac{r_{\lambda,1}(c_k,t)-1}{r_{\lambda,1}(c_k,t)}\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right)\\-\frac{1}{2}r_{\lambda,1}(c_k,t)\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right)+r_{\lambda,1}(c_k,t)s_{\lambda,1}(c_k,t)\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right)\end{bmatrix}$$

is such that the derivatives $\boldsymbol{f}_{\lambda,1}^{(j)}(c_k,t)$ *can be bounded independently of* λ*.*

Theorem 2.4 (Ramos, 2015*a*)**.** *If* (2.3) *then*

$$\boldsymbol{\pi}\big(\boldsymbol{B}_{\lambda,1}^{fine}(c_k,c_k+h_kt)\big)=\begin{bmatrix}t\\t^2\\1\end{bmatrix}\odot t\cdot\begin{bmatrix}[\boldsymbol{\iota}_{\lambda,1}(c_k,c_k+h_kt)]_{1,1}\\[\boldsymbol{\iota}_{\lambda,1}(c_k,c_k+h_kt)]_{2,1}\\[\boldsymbol{\iota}_{\lambda,1}(c_k,c_k+h_kt)]_{3,1}\end{bmatrix} \tag{2.6}$$

where

$$\boldsymbol{\iota}_{\lambda,1}(c_k,t):=\frac{q(t)-\frac{\int_{c_k}^t q(\xi)d\xi}{t-c_k}}{t-c_k}\begin{bmatrix}\varphi\left(\sqrt{\rho^2\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)}\right)\\-2\phi\left(\sqrt{\rho^2\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)}\right)\\\frac{1}{2}\phi\left(\sqrt{\rho^2\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)}\right)\rho^2\left(\boldsymbol{D}_{\lambda,0}(c_k,t)\right)\end{bmatrix}$$

is such that the derivatives $\boldsymbol{\iota}_{\lambda,1}^{(j)}(c_k,t)$ *can be bounded independently of* λ*.*

Theorem 2.5 (Ramos, 2015*a*)**.** *If* (2.4) *then* $\omega_{\lambda,1}(c_k,c_{k+1})\in[1,+\infty)$ *and*

$$\boldsymbol{\pi}\big(\boldsymbol{B}_{\lambda,1}^{fine}(c_k,c_k+h_kt)\big)=\frac{1-\cos\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\omega_{\lambda,1}(c_k,c_{k+1})t}t\begin{bmatrix}[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{1,1}\\0\\0\end{bmatrix}$$
$$+\cos\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)t\begin{bmatrix}[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{2,1}\\[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{4,1}\\[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{6,1}\end{bmatrix}$$
$$+\frac{\sin\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\omega_{\lambda,1}(c_k,c_{k+1})t}t\begin{bmatrix}0\\[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{5,1}\\[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{7,1}\end{bmatrix}$$
$$+\sin\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)t\begin{bmatrix}[\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{3,1}\\0\\0\end{bmatrix} \tag{2.7}$$

where

$$\boldsymbol{g}_{\lambda,1}(c_k,t) := \frac{1}{2}\frac{q(t)-\frac{\int_{c_k}^{t} q(\xi)d\xi}{t-c_k}}{t-c_k}\frac{1}{r_{\lambda,1}^2(c_k,t)} \times \begin{bmatrix} 1 \\ (r_{\lambda,1}(c_k,t)-1)\frac{1-\cos(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)} - r_{\lambda,1}(c_k,t)\sin\left(\epsilon_{\lambda,1}(c_k,t)\right) \\ -r_{\lambda,1}(c_k,t)\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right) - \frac{r_{\lambda,1}(c_k,t)-1}{r_{\lambda,1}(c_k,t)}\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right) \\ \cos\left(\epsilon_{\lambda,1}(c_k,t)\right) - \frac{r_{\lambda,1}(c_k,t)-1}{r_{\lambda,1}(c_k,t)}\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)} \\ -\frac{1}{r_{\lambda,1}(c_k,t)}\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right) + r_{\lambda,1}(c_k,t)s_{\lambda,1}(c_k,t)\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right) \\ r_{\lambda,1}^2(c_k,t)\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right) - \frac{r_{\lambda,1}(c_k,t)-1}{r_{\lambda,1}(c_k,t)}\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right) \\ -r_{\lambda,1}(c_k,t)\left(\cos\left(\epsilon_{\lambda,1}(c_k,t)\right) + r_{\lambda,1}(c_k,t)s_{\lambda,1}(c_k,t)\frac{\sin(\epsilon_{\lambda,1}(c_k,t))}{\epsilon_{\lambda,1}(c_k,t)}\right) \end{bmatrix}$$

is such that the derivatives $\boldsymbol{g}_{\lambda,1}^{(j)}(c_k,t)$ *can be bounded independently of* λ*.*

With the magnitude and behaviour of $\boldsymbol{B}_{\lambda,1}^{\text{fine}}(c_k,c_k+h_kt)$ made explicit in Theorems 2.3–2.5, Definition 2.5 below forms an approximation $\tilde{\boldsymbol{B}}_{\lambda,1,\mathcal{T}_{l-1}}^{\text{fine}}(c_k,c_k+h_kt)$ designed to satisfy two requirements: Firstly, it is such that the difference $\boldsymbol{B}_{\lambda,1}^{\text{fine}}(c_k,c_k+h_kt) - \tilde{\boldsymbol{B}}_{\lambda,1,\mathcal{T}_{l-1}}^{\text{fine}}(c_k,c_k+h_kt)$ is uniformly small, with respect to (1.7). Secondly, it is such that the integrals that appear below in Definition 2.6 can be integrated exactly.

In essence, $\tilde{\boldsymbol{B}}_{\lambda,1,\mathcal{T}_{l-1}}^{\text{fine}}(c_k,c_k+h_kt)$ interpolates the slow varying parts of the matrix $\boldsymbol{B}_{\lambda,1}^{\text{fine}}(c_k,c_k+h_kt)$, which, as exposed in Theorems 2.3–2.5, depend on (2.2)–(2.4).

Definition 2.5. *Let the interpolation points* $\mathcal{T}_{l-1} \subseteq (0,1]$ *be as defined by:*

$$\begin{aligned}
\mathcal{S}_{11} &:= \{(t+1)/2 : U_{11}(t)=0\} =: \{u_1,u_2,\ldots,u_{11} : u_1<u_2<\cdots<u_{11}\} \subseteq (0,1)\\
\mathcal{S}_8 &:= \{u_1,u_2,u_4,u_5,u_6,u_8,u_9,u_{10}\} \subseteq \mathcal{S}_{11}\\
\mathcal{S}_5 &:= \{u_2,u_4,u_6,u_8,u_{10}\} = \{(t+1)/2 : U_5(t)=0\} \subseteq \mathcal{S}_8\\
\mathcal{S}_2 &:= \{u_4,u_8\} = \{(t+1)/2 : U_2(t)=0\} \subseteq \mathcal{S}_5\\
\mathcal{T}_{l-1} &:= \mathcal{S}_{l-2} \cup \{1\}, l \in \{4,7,10,13\}
\end{aligned}$$

where $U_j(t)$ *denotes the* j*-th Chebyshev polynomial of the second kind. In addition, define also*

$$\tilde{\boldsymbol{B}}_{\lambda,1,\mathcal{T}_{l-1}}^{fine}(c_k,c_k+h_kt) \in \mathfrak{sl}(2,\mathbb{R})$$

in each of (2.2)*,* (2.3) *and* (2.4) *by, respectively:*

- *the right hand side of* (2.5) *with* $t \mapsto [\boldsymbol{f}_{\lambda,1}(c_k,c_k+h_kt)]_{j,1}$ *replaced by polynomial interpolation at* $\mathcal{T}_{l-1}$*,*
- *the right hand side of* (2.6) *with* $t \mapsto [\boldsymbol{\iota}_{\lambda,1}(c_k,c_k+h_kt)]_{j,1}$ *replaced by polynomial interpolation at* $\mathcal{T}_{l-1}$*,*
- *the right hand side of* (2.7) *with* $t \mapsto [\boldsymbol{g}_{\lambda,1}(c_k,c_k+h_kt)]_{j,1}$ *replaced by polynomial interpolation at* $\mathcal{T}_{l-1}$*.*

With the machinery introduced so far, Definition 2.6 below sets out the key elements $\tilde{\boldsymbol{I}}_{\lambda,j,\mathcal{T}_{l-1}}^{\text{fine}}(c_k,c_{k+1})$ and $\tilde{\boldsymbol{I}}_{\lambda,j,\mathcal{T}_{l-1}}(c_k,c_{k+1})$, which emerge in the truncated and discretized flow and solution that appear at the end of this section in Definition 2.7 and Theorem 2.6. In particular, $\tilde{\boldsymbol{I}}_{\lambda,j,\mathcal{T}_{l-1}}(c_k,c_{k+1})$ are given by a rescaling of $\tilde{\boldsymbol{I}}_{\lambda,j,\mathcal{T}_{l-1}}^{\text{fine}}(c_k,c_{k+1})$, which, by construction, can be integrated exactly.

Definition 2.6. *Let*

$$\tilde{\boldsymbol{I}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_{k+1}) := \int_0^1 \tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt)dt, \tag{2.8}$$

$$\begin{aligned}\tilde{\boldsymbol{I}}^{fine}_{\lambda,2,\mathcal{T}_{l-1}}(c_k,c_{k+1}) := \int_0^1\int_0^{t_1} \big[&\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_2),\\ &\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_1)\big]dt_2dt_1,\end{aligned} \tag{2.9}$$

$$\begin{aligned}\tilde{\boldsymbol{I}}^{fine}_{\lambda,3,\mathcal{T}_{l-1}}(c_k,c_{k+1}) := \int_0^1\int_0^{t_1}\int_0^{t_1} \big[&\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_3),\\ &\big[\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_2),\\ &\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_1)\big]\big]dt_3dt_2dt_1,\end{aligned} \tag{2.10}$$

$$\begin{aligned}\tilde{\boldsymbol{I}}^{fine}_{\lambda,4,\mathcal{T}_{l-1}}(c_k,c_{k+1}) := \int_0^1\int_0^{t_1}\int_0^{t_1}\int_0^{t_1} \big[&\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_4),\\ &\big[\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_3),\\ &\big[\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_2),\\ &\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_1)\big]\big]\big]dt_4dt_3dt_2dt_1,\end{aligned} \tag{2.11}$$

$$\begin{aligned}\tilde{\boldsymbol{I}}^{fine}_{\lambda,5,\mathcal{T}_{l-1}}(c_k,c_{k+1}) := \int_0^1\int_0^{t_1}\int_0^{t_1}\int_0^{t_2} \big[&\big[\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_4),\\ &\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_2)\big],\\ &\big[\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_3),\\ &\tilde{\boldsymbol{B}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_k+h_kt_1)\big]\big]dt_4dt_3dt_2dt_1.\end{aligned} \tag{2.12}$$

In addition, according to (2.2)–(2.4)*, let* $\tilde{\boldsymbol{I}}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_{k+1}),\ldots,\tilde{\boldsymbol{I}}_{\lambda,5,\mathcal{T}_{l-1}}(c_k,c_{k+1}) \in \mathfrak{sl}(2,\mathbb{R})$ *be the unique elements which satisfy*

$$\begin{aligned}\boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big) &:= \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}^{fine}_{\lambda,1,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big)\\ &\quad\odot\begin{cases} h_k^2\begin{bmatrix}h_k & h_k^2 & 1\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ h_k^2\begin{bmatrix}\frac{1}{\sqrt{\lambda-q(c_k)}} & \frac{1}{\lambda-q(c_k)} & 1\end{bmatrix}^\top & \Leftarrow (2.4),\end{cases}\\ \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}_{\lambda,2,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big) &:= \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}^{fine}_{\lambda,2,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big)\\ &\quad\odot\begin{cases} h_k^5\begin{bmatrix}h_k & h_k^2 & 1\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ \frac{h_k^4}{\sqrt{\lambda-q\,(c_k)}}\begin{bmatrix}\frac{1}{\sqrt{\lambda-q(c_k)}} & \frac{1}{\lambda-q(c_k)} & 1\end{bmatrix}^\top & \Leftarrow (2.4),\end{cases}\\ \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}_{\lambda,3,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big) &:= \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}^{fine}_{\lambda,3,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big)\\ &\quad\odot\begin{cases} h_k^8\begin{bmatrix}h_k & h_k^2 & 1\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ \frac{h_k^6}{\lambda-q\,(c_k)}\begin{bmatrix}\frac{1}{\sqrt{\lambda-q(c_k)}} & \frac{1}{\lambda-q(c_k)} & 1\end{bmatrix}^\top & \Leftarrow (2.4),\end{cases}\end{aligned}$$

$$\boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}_{\lambda,4,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big) := \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}^{fine}_{\lambda,4,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big)$$
$$\odot \begin{cases} h_k^{11}\begin{bmatrix} h_k & h_k^2 & 1\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ \dfrac{h_k^8}{(\lambda-q(c_k))^{\frac{3}{2}}}\begin{bmatrix}\frac{1}{\sqrt{\lambda-q(c_k)}} & \frac{1}{\lambda-q(c_k)} & 1\end{bmatrix}^\top & \Leftarrow (2.4),\end{cases}$$
$$\boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}_{\lambda,5,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big) := \boldsymbol{\pi}\big(\tilde{\boldsymbol{I}}^{fine}_{\lambda,5,\mathcal{T}_{l-1}}(c_k,c_{k+1})\big)$$
$$\odot \begin{cases} h_k^{11}\begin{bmatrix} h_k & h_k^2 & 1\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ \dfrac{h_k^8}{(\lambda-q(c_k))^{\frac{3}{2}}}\begin{bmatrix}\frac{1}{\sqrt{\lambda-q(c_k)}} & \frac{1}{\lambda-q(c_k)} & 1\end{bmatrix}^\top & \Leftarrow (2.4).\end{cases}$$

In order to approximate the flow $\boldsymbol{F}_\lambda(c_k,c_{k+1})$ and solution $\boldsymbol{Y}_\lambda(c_{k+1})$ of the initial value problem (1.8)–(1.9) in Theorem 2.1, Definition 2.7 below introduces the truncated and discretized flow $\tilde{\tilde{\boldsymbol{F}}}_{\lambda,n}(c_k,c_{k+1})$ and solution $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$, together with the local $\boldsymbol{L}^{\text{total}}_{\lambda,n}(c_k,c_{k+1})$ and global $\boldsymbol{G}^{\text{total}}_{\lambda,n}(c_{k+1})$ errors that characterize each approximation. In particular, in line with Subsection 1.4, $\tilde{\tilde{\boldsymbol{F}}}_{\lambda,n}(c_k,c_{k+1})$ and $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ belong to the Lie group $\mathrm{SL}(2,\mathbb{R})$, whereas $\boldsymbol{L}^{\text{total}}_{\lambda,n}(c_k,c_{k+1})$ and $\boldsymbol{G}^{\text{total}}_{\lambda,n}(c_{k+1})$ lie in the Lie algebra $\mathfrak{sl}(2,\mathbb{R})$.

As testified in Theorem 2.6 below, Fer streamers then achieve global order $g\in\{4,7,10,13\}$, with respect to (1.7), by approximating $\boldsymbol{Y}_\lambda(c_{k+1})$ with $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ where $n=\log((g+2)/3)/\log(2)$.

Definition 2.7. *Let $n\in\{1,\log(3)/\log(2),2,\log(5)/\log(2)\}$, and define*

$$\begin{aligned}
\tilde{\tilde{\boldsymbol{F}}}_{\lambda,1}(c_k,c_{k+1}) &:= e^{\boldsymbol{D}_{\lambda,0}(c_k,c_{k+1})}e^{\tilde{\boldsymbol{I}}_{\lambda,1,\mathcal{T}_3}(c_k,c_{k+1})},\\
\tilde{\tilde{\boldsymbol{F}}}_{\lambda,\log(3)/\log(2)}(c_k,c_{k+1}) &:= e^{\boldsymbol{D}_{\lambda,0}(c_k,c_{k+1})}e^{\tilde{\boldsymbol{I}}_{\lambda,1,\mathcal{T}_6}(c_k,c_{k+1})}\\
&\quad\times e^{-\frac{1}{2}\tilde{\boldsymbol{I}}_{\lambda,2,\mathcal{T}_3}(c_k,c_{k+1})},\\
\tilde{\tilde{\boldsymbol{F}}}_{\lambda,2}(c_k,c_{k+1}) &:= e^{\boldsymbol{D}_{\lambda,0}(c_k,c_{k+1})}e^{\tilde{\boldsymbol{I}}_{\lambda,1,\mathcal{T}_9}(c_k,c_{k+1})}\\
&\quad\times e^{-\frac{1}{2}\tilde{\boldsymbol{I}}_{\lambda,2,\mathcal{T}_6}(c_k,c_{k+1})+\frac{1}{3}\tilde{\boldsymbol{I}}_{\lambda,3,\mathcal{T}_3}(c_k,c_{k+1})},\\
\tilde{\tilde{\boldsymbol{F}}}_{\lambda,\log(5)/\log(2)}(c_k,c_{k+1}) &:= e^{\boldsymbol{D}_{\lambda,0}(c_k,c_{k+1})}e^{\tilde{\boldsymbol{I}}_{\lambda,1,\mathcal{T}_{12}}(c_k,c_{k+1})}\\
&\quad\times e^{-\frac{1}{2}\tilde{\boldsymbol{I}}_{\lambda,2,\mathcal{T}_9}(c_k,c_{k+1})+\frac{1}{3}\tilde{\boldsymbol{I}}_{\lambda,3,\mathcal{T}_6}(c_k,c_{k+1})-\frac{1}{8}\tilde{\boldsymbol{I}}_{\lambda,4,\mathcal{T}_3}(c_k,c_{k+1})}\\
&\quad\times e^{-\frac{1}{8}\tilde{\boldsymbol{I}}_{\lambda,5,\mathcal{T}_3}(c_k,c_{k+1})},\\
\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1}) &:= \tilde{\tilde{\boldsymbol{F}}}_{\lambda,n}(c_k,c_{k+1})\cdots\tilde{\tilde{\boldsymbol{F}}}_{\lambda,n}(c_1,c_2)\tilde{\tilde{\boldsymbol{F}}}_{\lambda,n}(a,c_1),\\
\boldsymbol{L}^{total}_{\lambda,n}(c_k,c_{k+1}) &:= \log\big(\boldsymbol{F}_\lambda(c_k,c_{k+1})\tilde{\tilde{\boldsymbol{F}}}^{-1}_{\lambda,n}(c_k,c_{k+1})\big),\\
\boldsymbol{G}^{total}_{\lambda,n}(c_{k+1}) &:= \log\big(\boldsymbol{Y}_\lambda(c_{k+1})\tilde{\tilde{\boldsymbol{Y}}}^{-1}_{\lambda,n}(c_{k+1})\big).
\end{aligned}$$

Theorem 2.6 (Ramos, 2015*a*)**.** *If $n\in\{1,\log(3)/\log(2),2,\log(5)/\log(2)\}$, and (2.2), (2.3) or (2.4), then, in the uniform regime (1.7),*

$$\boldsymbol{\pi}\big(\boldsymbol{L}^{total}_{\lambda,n}(c_k,c_{k+1})\big) = h_{\max}^{3\cdot 2^n-1}\begin{cases}\begin{bmatrix}\mathcal{O}(h_{\max}) & \mathcal{O}\left(h_{\max}^2\right) & \mathcal{O}(1)\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ \begin{bmatrix}\frac{\mathcal{O}(1)}{\sqrt{\lambda-q_{\max}}} & \frac{\mathcal{O}(1)}{\lambda-q_{\max}} & \mathcal{O}(1)\end{bmatrix}^\top & \Leftarrow (2.4),\end{cases}$$

$$\boldsymbol{\pi}\big(\boldsymbol{G}^{total}_{\lambda,n}(c_{k+1})\big) = h_{\max}^{3\cdot 2^n-2}\begin{cases}\begin{bmatrix}\mathcal{O}(h_{\max}) & \mathcal{O}\left(h_{\max}^2\right) & \mathcal{O}(1)\end{bmatrix}^\top & \Leftarrow (2.2),(2.3),\\ \begin{bmatrix}\frac{\mathcal{O}(1)}{\sqrt{\lambda-q_{\max}}} & \frac{\mathcal{O}(1)}{\lambda-q_{\max}} & \mathcal{O}(1)\end{bmatrix}^\top & \Leftarrow (2.4).\end{cases}$$

As indicated in Definition 2.7 and Theorem 2.6, the Fer streamers approach with global order $g \in \{4, 7, 10, 13\}$, uses the polynomial interpolation in Definition 2.5 to approximate $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k t)$ with $\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_{l-1}}(c_k, c_k + h_k t)$, $l \in \{4, 7, 10, 13\} \cap [0, g]$ uniformly with respect to (1.7), which requires the data:

$$\{q(a)\} \cup \left(\bigcup_{k=0}^{m-1} \left\{ q(c_k + h_k t), \int_{c_k}^{c_k + h_k t} q(\xi) d\xi : t \in \mathcal{T}_{l-1} \right\} \right).$$

Since the antiderivative of the potential is usually unavailable in closed-form, one approximates, up to local order, the antiderivative data

$$\left\{ \int_{c_k}^{c_k + h_k t} q(\xi) d\xi : t \in \mathcal{T}_{l-1} \right\}$$

by the polynomial interpolation of $q(\xi)$ in $\xi \in [c_k, c_{k+1}]$ with the potential data

$$\{q(c_k)\} \cup \{q(c_k + h_k t) : t \in \mathcal{T}_{l-1}\}$$

and the exact integration of the result. Since $\mathcal{T}_3 \subseteq \mathcal{T}_6 \subseteq \mathcal{T}_9 \subseteq \mathcal{T}_{12}$ (c.f. Definition 2.5), to attain global order $p + 1 \in \{4, 7, 10, 13\}$, Fer streamers evaluate $q(a)$ and $q(c_k + h_k \cdot)$, $k \in \{0, \ldots, m-1\}$, at the p points in $\mathcal{T}_p$, in accordance with Subsection 1.3.

3 Practical implementation

Following the description of the uniform approximations provided by Fer streamers for the solution of the initial value problem (1.8)–(1.9) in the previous section, the present section now bridges between theoretical construction and practical implementation. In particular, Subsections 3.1–3.2 below examine the practical implementation of the truncated and discretized solution $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$, whereas Subsections 3.3–3.4 discuss the use of the computed data $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ to approximate the eigenvalues of the boundary value problem (1.1)–(1.2).

3.1 'Reduced' Hall basis for Fer streamers

In view of Definitions 2.6 and 2.7, the implementation of the truncated and discretized solution $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ boils down to the computation of $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,j,\mathcal{T}_{l-1}}(c_k, c_{k+1})$. More precisely, according to Theorem 2.6, to attain global order $g \in \{4, 7, 10, 13\}$, in the sense of (1.7), one may approximate $\boldsymbol{Y}_\lambda(c_{k+1})$ by $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ with $n = \log((g+2)/3)/\log(2)$, the implementation of which reduces to the computation of:

- $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_3}(c_k, c_{k+1})$, for $g = 4$,
- $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_6}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,2,\mathcal{T}_3}(c_k, c_{k+1})$, for $g = 7$,
- $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_9}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,2,\mathcal{T}_6}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,3,\mathcal{T}_3}(c_k, c_{k+1})$, for $g = 10$,
- $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_{12}}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,2,\mathcal{T}_9}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,3,\mathcal{T}_6}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,4,\mathcal{T}_3}(c_k, c_{k+1})$, $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,5,\mathcal{T}_3}(c_k, c_{k+1})$, for $g = 13$.

As discussed in this subsection, since, by construction, each integral (2.8)–(2.12) can be integrated exactly, the question then becomes how to achieve such computation with the least 'amount of linear algebra' and, by extension, computational time.

To minimize the length of this subsection while retaining its essential message, the discussion focuses on the reduction of the 'amount of linear algebra' for the terms with three interpolation points across global orders 4, 7, 10 and 13:

$$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,j,\mathcal{T}_3}(c_k, c_{k+1}), \quad j \in \{1,2,3,4,5\}, \tag{3.1}$$

which reveal the ins and outs also for the computation of the other terms.

Recalling the construction of $\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_{l-1}}(c_k, c_k + h_k t)$ in Definition 2.5, it is clear that, for each fixed numerical mesh, the computation of (2.8)–(2.12), depends on which interval (2.2), (2.3) or (2.4), λ lies on. As an example, with three interpolation points, by solving a linear system exactly, one can write:

$$\begin{aligned}
&\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_3}(c_k, c_k + h_k t)\\
&\left\{\begin{aligned}
&=: \frac{1-\cos\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)^2} t^2 \left(t^2 \mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t\mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3} + \mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,3,\mathcal{T}_3}\right)\\
&\quad + \cos\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right) t \Big(t^4 \mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t^3 \mathscr{D}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\\
&\qquad\qquad + t^2 \mathscr{G}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t\mathscr{E}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + \mathscr{C}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\Big)\\
&\quad + \frac{\sin\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\omega_{\lambda,1}(c_k,c_{k+1})t} t \Big(t^3 \mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,4,\mathcal{T}_3} + t^2 \mathscr{E}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3}\\
&\qquad\qquad + t\mathscr{E}^{\boldsymbol{f},c_k,h_k}_{\lambda,3,\mathcal{T}_3} + \mathscr{C}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3}\Big)\\
&\quad + \phi\left(i\cdot\omega_{\lambda,1}(c_k,c_{k+1})t\right) t^3 \left(t^2 \mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3} + t\mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,3,\mathcal{T}_3} + \mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,4,\mathcal{T}_3}\right), && \Leftarrow (2.2),\\
&=: t\left(t^4 \mathscr{B}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t^3 \mathscr{D}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t^2 \mathscr{G}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t\mathscr{E}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + \mathscr{C}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\right), && \Leftarrow (2.3),\\
&=: \frac{1-\cos\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\omega_{\lambda,1}(c_k,c_{k+1})t} t \left(t^2 \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t\mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,2,\mathcal{T}_3} + \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,3,\mathcal{T}_3}\right)\\
&\quad + \cos\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right) t \left(t^2 \mathscr{G}^{\boldsymbol{g},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t\mathscr{G}^{\boldsymbol{g},c_k,h_k}_{\lambda,2,\mathcal{T}_3} + \mathscr{G}^{\boldsymbol{g},c_k,h_k}_{\lambda,3,\mathcal{T}_3}\right)\\
&\quad + \frac{\sin\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right)}{\omega_{\lambda,1}(c_k,c_{k+1})t} t \left(t^2 \mathscr{F}^{\boldsymbol{g},c_k,h_k}_{\lambda,1,\mathcal{T}_3} + t\mathscr{F}^{\boldsymbol{g},c_k,h_k}_{\lambda,2,\mathcal{T}_3} + \mathscr{F}^{\boldsymbol{g},c_k,h_k}_{\lambda,3,\mathcal{T}_3}\right)\\
&\quad + \sin\left(\omega_{\lambda,1}(c_k,c_{k+1})t\right) t \left(t^2 \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,4,\mathcal{T}_3} + t\mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,5,\mathcal{T}_3} + \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,6,\mathcal{T}_3}\right), && \Leftarrow (2.4),
\end{aligned}\right.
\end{aligned}$$

where each group of matrices

$$\begin{aligned}
&\Big\{\mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,3,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{f},c_k,h_k}_{\lambda,4,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,1,\mathcal{T}_3},\\
&\quad \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,3,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,4,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,5,\mathcal{T}_3}, \mathscr{A}^{\boldsymbol{g},c_k,h_k}_{\lambda,6,\mathcal{T}_3}\Big\}, && (3.2)\\
&\Big\{\mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,3,\mathcal{T}_3}, \mathscr{B}^{\boldsymbol{f},c_k,h_k}_{\lambda,4,\mathcal{T}_3}, \mathscr{B}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\Big\}, && (3.3)\\
&\Big\{\mathscr{C}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{C}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{C}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\Big\}, && (3.4)\\
&\Big\{\mathscr{D}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{D}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\Big\}, && (3.5)\\
&\Big\{\mathscr{E}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{E}^{\boldsymbol{f},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{E}^{\boldsymbol{f},c_k,h_k}_{\lambda,3,\mathcal{T}_3}, \mathscr{E}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3}\Big\}, && (3.6)\\
&\Big\{\mathscr{F}^{\boldsymbol{g},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{F}^{\boldsymbol{g},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{F}^{\boldsymbol{g},c_k,h_k}_{\lambda,3,\mathcal{T}_3}\Big\}, && (3.7)\\
&\Big\{\mathscr{G}^{\boldsymbol{f},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{G}^{\boldsymbol{\iota},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{G}^{\boldsymbol{g},c_k,h_k}_{\lambda,1,\mathcal{T}_3}, \mathscr{G}^{\boldsymbol{g},c_k,h_k}_{\lambda,2,\mathcal{T}_3}, \mathscr{G}^{\boldsymbol{g},c_k,h_k}_{\lambda,3,\mathcal{T}_3}\Big\}, && (3.8)
\end{aligned}$$

possesses a certain structure. More concretely, if

$$\boldsymbol{E}_1 := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \boldsymbol{E}_2 := \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{E}_3 := \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

then (3.2)–(3.8) exhibit the following features:

$$\mathscr{A} \in \operatorname{span}\{\boldsymbol{E}_1\}, \qquad \mathscr{D} \in \operatorname{span}\{\boldsymbol{E}_1, \boldsymbol{E}_2\}, \tag{3.9}$$

$$\mathscr{B} \in \operatorname{span}\{\boldsymbol{E}_2\}, \qquad \mathscr{E} \in \operatorname{span}\{\boldsymbol{E}_1, \boldsymbol{E}_3\}, \tag{3.10}$$

$$\mathscr{C} \in \operatorname{span}\{\boldsymbol{E}_3\}, \qquad \mathscr{F} \in \operatorname{span}\{\boldsymbol{E}_2, \boldsymbol{E}_3\}, \qquad \mathscr{G} \in \operatorname{span}\{\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3\}. \tag{3.11}$$

Even though, as said before, the presentation focuses on the decrease of the 'amount of linear algebra' for the exact integration of (3.1) with three interpolation points, it is pertinent at this moment to inform the reader that the aforementioned representation of $\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_{l-1}}(c_k, c_k + h_k t)$ in terms of matrices of type $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$, $\mathscr{E}$, $\mathscr{F}$ and $\mathscr{G}$, made explicit above for three interpolation points, holds similarly for any number of points, given that, in general, by solving a linear system exactly, one can write $\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_{l-1}}(c_k, c_k + h_k t)$ as a linear combination of:

- $((l-1)+1)$ $\mathscr{A}$'s, $((l-1)+1)$ $\mathscr{B}$'s, 2 $\mathscr{C}$'s, 1 $\mathscr{D}$, $(l-1)$ $\mathscr{E}$'s and $((l-1)-2)$ $\mathscr{G}$'s, in (2.2),
- 1 $\mathscr{B}$, 1 $\mathscr{C}$, 1 $\mathscr{D}$, 1 $\mathscr{E}$ and $((l-1)-2)$ $\mathscr{G}$'s, in (2.3),
- $2(l-1)$ $\mathscr{A}$'s, $(l-1)$ $\mathscr{F}$'s and $(l-1)$ $\mathscr{G}$'s, in (2.4).

With this in mind, returning to the exact integration of (3.1), it is now convenient to aggregate the matrices in (3.2)–(3.8) according to each eigenvalue range (2.2), (2.3) or (2.4), and to denote them more generically by:

$$\Big\{\boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,1}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,2}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,3}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,4}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,5}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,6}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,7}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,8}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,9}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,10}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,11}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,12}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,13}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,14}, \boldsymbol{Z}^{\boldsymbol{f}}_{\lambda,15}\Big\}, \tag{3.12}$$

$$\Big\{\boldsymbol{Z}^{\boldsymbol{\iota}}_{\lambda,1}, \boldsymbol{Z}^{\boldsymbol{\iota}}_{\lambda,2}, \boldsymbol{Z}^{\boldsymbol{\iota}}_{\lambda,3}, \boldsymbol{Z}^{\boldsymbol{\iota}}_{\lambda,4}, \boldsymbol{Z}^{\boldsymbol{\iota}}_{\lambda,5}\Big\}, \tag{3.13}$$

$$\Big\{\boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,1}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,2}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,3}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,4}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,5}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,6}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,7}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,8}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,9}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,10}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,11}, \boldsymbol{Z}^{\boldsymbol{g}}_{\lambda,12}\Big\}. \tag{3.14}$$

Gauging upon the definition of (3.1) in (2.8)–(2.12), noting that the integrands of (3.1) are, respectively, 0, 1, 2, 3 and 3 commutators between $\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_3}(c_k, c_k + h_k\xi)$ evaluated at various ξ, while at the same time recalling that $\tilde{\boldsymbol{B}}^{\text{fine}}_{\lambda,1,\mathcal{T}_3}(c_k, c_k + h_k\xi)$ has been given above as a linear combination of $\boldsymbol{Z}$'s, where each scalar coefficient of each matrix $\boldsymbol{Z}$ is a function of ξ, it becomes clear that to integrate each (3.1), one must expand each commutator representation of each integrand via the bilinear properties of the commutator

$$\begin{aligned}
\big[\boldsymbol{Z}_{\lambda,j_1} + \boldsymbol{Z}_{\lambda,j_2}, \boldsymbol{Z}_{\lambda,j_3}\big] &= \big[\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_3}\big] + \big[\boldsymbol{Z}_{\lambda,j_2}, \boldsymbol{Z}_{\lambda,j_3}\big], \\
\big[\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_2} + \boldsymbol{Z}_{\lambda,j_3}\big] &= \big[\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_2}\big] + \big[\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_3}\big], \\
\big[c\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_2}\big] &= c\big[\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_2}\big], \quad c \in \mathbb{R}, \\
\big[\boldsymbol{Z}_{\lambda,j_1}, c\boldsymbol{Z}_{\lambda,j_2}\big] &= c\big[\boldsymbol{Z}_{\lambda,j_1}, \boldsymbol{Z}_{\lambda,j_2}\big], \quad c \in \mathbb{R},
\end{aligned}$$

to single out the scalar functions that require integration. This, of course, represents each integrand of (3.1) as a linear combination of the elements with, respectively, 0, 1, 2, 3 and 3 commutators in the free magma of each alphabet (3.12), (3.13) or (3.14), the size of which grows significantly with the size of the alphabet and the number of commutators, as depicted in Table 1 under "free magma".

The 'amount of linear algebra' mentioned above then relates to the number of commutators that result from such procedure.

Fortunately, there exist three mechanisms that can be used to decrease this amount of linear algebra. These are:

- Firstly, Free Lie algebra (FLA) techniques and Hall basis, which lead to fewer commutators via a systematic use of commutator identities such as: skew symmetry, Jacobi's identity, etc,
- Secondly, when collected in a Hall basis (which varies with the ordering of the alphabet), certain linear combinations between different integrands are then identically zero,
- Thirdly, when collected in a Hall basis (which depends on the ordering of the alphabet), certain linear combinations between different integrands integrate exactly to zero.

The remainder of this section then concerns a brief description of the savings achieved via these mechanisms, the first of which is well-known in the literature, whereas the second and third arise now from the practical implementation of Fer streamers.

The first mechanism above is well-known, an excellent reference being (Munthe–Kaas and Owren, 1999). In a nutshell, FLA techniques and Hall basis, diminish the number of commutators by judiciously invoking skew symmetry, Jacobi's identity and other relations:

$$\big[\boldsymbol{Z}_{\lambda,j_1},\boldsymbol{Z}_{\lambda,j_2}\big] = -\big[\boldsymbol{Z}_{\lambda,j_2},\boldsymbol{Z}_{\lambda,j_1}\big], \tag{3.15}$$

$$\begin{aligned} 0 = &\big[\boldsymbol{Z}_{\lambda,j_1},\big[\boldsymbol{Z}_{\lambda,j_2},\boldsymbol{Z}_{\lambda,j_3}\big]\big] \\ &+\big[\boldsymbol{Z}_{\lambda,j_2},\big[\boldsymbol{Z}_{\lambda,j_3},\boldsymbol{Z}_{\lambda,j_1}\big]\big] \\ &+\big[\boldsymbol{Z}_{\lambda,j_3},\big[\boldsymbol{Z}_{\lambda,j_1},\boldsymbol{Z}_{\lambda,j_2}\big]\big], \end{aligned} \tag{3.16}$$

$$\begin{aligned} \big[\big[\boldsymbol{Z}_{\lambda,j_1},\boldsymbol{Z}_{\lambda,j_3}\big],\big[\boldsymbol{Z}_{\lambda,j_2},\boldsymbol{Z}_{\lambda,j_4}\big]\big] = &-\big[\boldsymbol{Z}_{\lambda,j_4},\big[\boldsymbol{Z}_{\lambda,j_3},\big[\boldsymbol{Z}_{\lambda,j_2},\boldsymbol{Z}_{\lambda,j_1}\big]\big]\big] \\ &-\big[\boldsymbol{Z}_{\lambda,j_1},\big[\boldsymbol{Z}_{\lambda,j_4},\big[\boldsymbol{Z}_{\lambda,j_3},\boldsymbol{Z}_{\lambda,j_2}\big]\big]\big] \\ &-\big[\boldsymbol{Z}_{\lambda,j_2},\big[\boldsymbol{Z}_{\lambda,j_1},\big[\boldsymbol{Z}_{\lambda,j_4},\boldsymbol{Z}_{\lambda,j_3}\big]\big]\big] \\ &-\big[\boldsymbol{Z}_{\lambda,j_3},\big[\boldsymbol{Z}_{\lambda,j_2},\big[\boldsymbol{Z}_{\lambda,j_1},\boldsymbol{Z}_{\lambda,j_4}\big]\big]\big], \end{aligned} \tag{3.17}$$

to remove commutators that become redundant in light of such equalities. As illustrated in Table 1 under "Hall basis", the number of terms decreases substantially from the free magma to the Hall basis. On this point, it is important to note that the MATLAB package DiffMan (Engøet al., 1999) has been used to confirm the coefficient expansion of the various terms in the Hall basis, which have been used to sort up the data required for global order up to 13.

The second mechanism above originates from the observation that while it is unquestionable that expressing the commutators in a Hall basis results in a significant decrease of the amount of linear algebra, it is equally true that, by construction, a Hall basis does not take into account any structure that each letter '$\boldsymbol{Z}$' of each alphabet (3.12), (3.13) or (3.14) might possess, creating a chance for further reduction. In keeping with this train of thought, in view of (3.9)–(3.11), one realizes that the equalities

$$\big[\boldsymbol{E}_1,\boldsymbol{E}_2\big] = 2\boldsymbol{E}_2, \qquad \big[\boldsymbol{E}_1,\boldsymbol{E}_3\big] = -2\boldsymbol{E}_3, \qquad \big[\boldsymbol{E}_2,\boldsymbol{E}_3\big] = \boldsymbol{E}_1,$$

give rise to the relations (that are not captured by a Hall basis):

$$[\mathscr{A},\mathscr{A}] = \boldsymbol{0}, \tag{3.18}$$

$$[\mathscr{B},\mathscr{B}] = \boldsymbol{0}, \qquad [\mathscr{C},\mathscr{C}] = \boldsymbol{0}, \tag{3.19}$$

$$[\mathscr{A},[\mathscr{B},\mathscr{C}]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{E},\mathscr{E}]] = \boldsymbol{0}, \tag{3.20}$$

$$[\mathscr{B},[\mathscr{A},\mathscr{B}]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{A},\mathscr{C}]] = \boldsymbol{0}, \tag{3.21}$$

$$[\mathscr{B},[\mathscr{A},\mathscr{D}]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{A},\mathscr{E}]] = \boldsymbol{0}, \tag{3.22}$$

$$[\mathscr{B},[\mathscr{B},\mathscr{D}]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{C},\mathscr{E}]] = \boldsymbol{0}, \tag{3.23}$$

$$[\mathscr{A},[\mathscr{B},[\mathscr{A},\mathscr{C}]]] = \boldsymbol{0}, \qquad [\mathscr{B},[\mathscr{A},[\mathscr{B},\mathscr{G}]]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{A},[\mathscr{C},\mathscr{E}]]] = \boldsymbol{0}, \tag{3.24}$$

$$[\mathscr{A},[\mathscr{B},[\mathscr{A},\mathscr{E}]]] = \boldsymbol{0}, \qquad [\mathscr{B},[\mathscr{B},[\mathscr{B},\mathscr{C}]]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{A},[\mathscr{C},\mathscr{G}]]] = \boldsymbol{0}, \tag{3.25}$$

$$[\mathscr{A},[\mathscr{B},[\mathscr{A},\mathscr{G}]]] = \boldsymbol{0}, \qquad [\mathscr{B},[\mathscr{B},[\mathscr{B},\mathscr{E}]]] = \boldsymbol{0}, \qquad [\mathscr{C},[\mathscr{A},[\mathscr{E},\mathscr{E}]]] = \boldsymbol{0}, \tag{3.26}$$

$$[\mathscr{A},[\mathscr{B},[\mathscr{C},\mathscr{E}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{B},[\mathscr{B},\mathscr{G}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{C},[\mathscr{B},\mathscr{C}]]]=\mathbf{0}, \tag{3.27}$$

$$[\mathscr{A},[\mathscr{B},[\mathscr{E},\mathscr{E}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{D},[\mathscr{A},\mathscr{B}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{C},[\mathscr{C},\mathscr{D}]]]=\mathbf{0}, \tag{3.28}$$

$$[\mathscr{A},[\mathscr{C},[\mathscr{A},\mathscr{B}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{D},[\mathscr{A},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{C},[\mathscr{C},\mathscr{G}]]]=\mathbf{0}, \tag{3.29}$$

$$[\mathscr{A},[\mathscr{C},[\mathscr{A},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{D},[\mathscr{B},\mathscr{C}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{A},\mathscr{C}]]]=\mathbf{0}, \tag{3.30}$$

$$[\mathscr{A},[\mathscr{C},[\mathscr{A},\mathscr{G}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{D},[\mathscr{B},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{A},\mathscr{E}]]]=\mathbf{0}, \tag{3.31}$$

$$[\mathscr{A},[\mathscr{C},[\mathscr{B},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{D},[\mathscr{B},\mathscr{E}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{B},\mathscr{C}]]]=\mathbf{0}, \tag{3.32}$$

$$[\mathscr{B},[\mathscr{A},[\mathscr{A},\mathscr{B}]]]=\mathbf{0}, \quad [\mathscr{B},[\mathscr{D},[\mathscr{B},\mathscr{G}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{C},\mathscr{D}]]]=\mathbf{0}, \tag{3.33}$$

$$[\mathscr{B},[\mathscr{A},[\mathscr{A},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{A},[\mathscr{A},\mathscr{C}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{C},\mathscr{E}]]]=\mathbf{0}, \tag{3.34}$$

$$[\mathscr{B},[\mathscr{A},[\mathscr{B},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{A},[\mathscr{A},\mathscr{E}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{C},\mathscr{G}]]]=\mathbf{0}, \tag{3.35}$$

$$[\mathscr{B},[\mathscr{A},[\mathscr{B},\mathscr{E}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{A},[\mathscr{C},\mathscr{D}]]]=\mathbf{0}, \quad [\mathscr{C},[\mathscr{E},[\mathscr{E},\mathscr{E}]]]=\mathbf{0}, \tag{3.36}$$

$$[\mathscr{D}_{\lambda,j},[\mathscr{B},[\mathscr{C},\mathscr{D}_{\lambda,j}]]]=\mathbf{0}, \quad [\mathscr{D}_{\lambda,j},[\mathscr{B},[\mathscr{D}_{\lambda,j},\mathscr{G}]]]=\mathbf{0}, \quad [\mathscr{E}_{\lambda,j},[\mathscr{C},[\mathscr{D},\mathscr{E}_{\lambda,j}]]]=\mathbf{0}, \tag{3.37}$$

$$[\mathscr{D}_{\lambda,j},[\mathscr{B},[\mathscr{D}_{\lambda,j},\mathscr{E}]]]=\mathbf{0}, \quad [\mathscr{E}_{\lambda,j},[\mathscr{C},[\mathscr{B},\mathscr{E}_{\lambda,j}]]]=\mathbf{0}, \quad [\mathscr{E}_{\lambda,j},[\mathscr{C},[\mathscr{E}_{\lambda,j},\mathscr{G}]]]=\mathbf{0}, \tag{3.38}$$

$$[[\mathscr{A},\mathscr{B}],[\mathscr{A},\mathscr{B}]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{C}],[\mathscr{E},\mathscr{E}]]=\mathbf{0}, \quad [[\mathscr{B},\mathscr{C}],[\mathscr{B},\mathscr{C}]]=\mathbf{0}, \tag{3.39}$$

$$[[\mathscr{A},\mathscr{B}],[\mathscr{A},\mathscr{D}]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{D}],[\mathscr{A},\mathscr{D}]]=\mathbf{0}, \quad [[\mathscr{B},\mathscr{D}],[\mathscr{B},\mathscr{D}]]=\mathbf{0}, \tag{3.40}$$

$$[[\mathscr{A},\mathscr{B}],[\mathscr{B},\mathscr{D}]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{D}],[\mathscr{B},\mathscr{D}]]=\mathbf{0}, \quad [[\mathscr{C},\mathscr{E}],[\mathscr{C},\mathscr{E}]]=\mathbf{0}, \tag{3.41}$$

$$[[\mathscr{A},\mathscr{C}],[\mathscr{A},\mathscr{C}]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{E}],[\mathscr{A},\mathscr{E}]]=\mathbf{0}, \quad [[\mathscr{C},\mathscr{E}],[\mathscr{E},\mathscr{E}]]=\mathbf{0}, \tag{3.42}$$

$$[[\mathscr{A},\mathscr{C}],[\mathscr{A},\mathscr{E}]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{E}],[\mathscr{C},\mathscr{E}]]=\mathbf{0}, \quad [[\mathscr{E},\mathscr{E}],[\mathscr{E},\mathscr{E}]]=\mathbf{0}, \tag{3.43}$$

$$[[\mathscr{A},\mathscr{C}],[\mathscr{C},\mathscr{E}]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{E}],[\mathscr{E},\mathscr{E}]]=\mathbf{0}, \tag{3.44}$$

$$[[\mathscr{A},\mathscr{G}_{\lambda,j}],[\mathscr{A},\mathscr{G}_{\lambda,j}]]=\mathbf{0}, \tag{3.45}$$

$$[[\mathscr{B},\mathscr{E}_{\lambda,j}],[\mathscr{B},\mathscr{E}_{\lambda,j}]]=\mathbf{0}, \quad [[\mathscr{B},\mathscr{G}_{\lambda,j}],[\mathscr{B},\mathscr{G}_{\lambda,j}]]=\mathbf{0}, \quad [[\mathscr{C},\mathscr{D}_{\lambda,j}],[\mathscr{C},\mathscr{D}_{\lambda,j}]]=\mathbf{0}, \tag{3.46}$$

$$[[\mathscr{C},\mathscr{G}_{\lambda,j}],[\mathscr{C},\mathscr{G}_{\lambda,j}]]=\mathbf{0}, \tag{3.47}$$

$$[\mathscr{A},[\mathscr{F},\mathscr{F}]]=\mathbf{0}, \tag{3.48}$$

$$[\mathscr{A},[\mathscr{F},[\mathscr{A},\mathscr{F}]]]=\mathbf{0}, \quad [\mathscr{F}_{\lambda,j},[\mathscr{A},[\mathscr{A},\mathscr{F}_{\lambda,j}]]]=\mathbf{0}, \quad [[\mathscr{F},\mathscr{F}],[\mathscr{F},\mathscr{F}]]=\mathbf{0}, \tag{3.49}$$

$$[\mathscr{A},[\mathscr{F},[\mathscr{A},\mathscr{G}]]]=\mathbf{0}, \quad [\mathscr{F}_{\lambda,j},[\mathscr{A},[\mathscr{F}_{\lambda,j},\mathscr{G}]]]=\mathbf{0}, \quad [[\mathscr{A},\mathscr{F}_{\lambda,j}],[\mathscr{A},\mathscr{F}_{\lambda,j}]]=\mathbf{0}, \tag{3.50}$$

where each relation in (3.18)–(3.50) should be understood as a family of equalities. For instance, (3.18), (3.45) and (3.48) encode:

$$\begin{aligned}
[\mathscr{A}_{\lambda,j_1},\mathscr{A}_{\lambda,j_2}]&=\mathbf{0},\forall j_1,\forall j_2,\\
[[\mathscr{A}_{\lambda,j_1},\mathscr{G}_{\lambda,j}],[\mathscr{A}_{\lambda,j_2},\mathscr{G}_{\lambda,j}]]&=\mathbf{0},\forall j_1,\forall j_2,\forall j,\\
[\mathscr{A}_{\lambda,j_1},[\mathscr{F}_{\lambda,j_2},\mathscr{F}_{\lambda,j_3}]]&=\mathbf{0},\forall j_1,\forall j_2,\forall j_3.
\end{aligned}$$

In particular, for each fixed numerical mesh, their appropriateness varies according to where λ lies. In detail:

- (3.18)–(3.47) play a role in (2.2),
- (3.23) play a role in (2.3),
- (3.18), (3.45), (3.48)–(3.50) play a part in (2.4).

The idea is then to remove such terms from a Hall basis.

However, since the constraints (3.18)–(3.50) are not symmetric in (3.2)–(3.8) and the terms in a Hall basis (with more than one commutator) are not symmetric in the alphabets (3.12)–(3.14), it may be possible, at least in principle, that certain bijections between (3.2)–(3.8) and (3.12)–(3.14) yield less non-zero terms than others when considering (3.18)–(3.50) on top of a Hall basis.

To put it another way, since (3.18)–(3.50) do not distinguish between different $\mathscr{A}$'s, different $\mathscr{B}$'s or different $\mathscr{C}$'s, the question can be raised equivalently as to whether specific distinct bijections between,

respectively, (3.12), (3.13), (3.14) and

$$\Big\{\mathscr{A}^{\boldsymbol{f}},\mathscr{A}^{\boldsymbol{f}},\mathscr{A}^{\boldsymbol{f}},\mathscr{A}^{\boldsymbol{f}},\mathscr{B}^{\boldsymbol{f}},\mathscr{B}^{\boldsymbol{f}},\mathscr{B}^{\boldsymbol{f}},\mathscr{B}^{\boldsymbol{f}},\mathscr{C}^{\boldsymbol{f}},\mathscr{C}^{\boldsymbol{f}},\mathscr{D}^{\boldsymbol{f}},\mathscr{E}^{\boldsymbol{f}}_{\lambda,1},\mathscr{E}^{\boldsymbol{f}}_{\lambda,2},\mathscr{E}^{\boldsymbol{f}}_{\lambda,3},\mathscr{G}^{\boldsymbol{f}}\Big\}, \tag{3.51}$$

$$\Big\{\mathscr{B}^{\boldsymbol{\iota}},\mathscr{C}^{\boldsymbol{\iota}},\mathscr{D}^{\boldsymbol{\iota}},\mathscr{E}^{\boldsymbol{\iota}},\mathscr{G}^{\boldsymbol{\iota}}\Big\}, \tag{3.52}$$

$$\Big\{\mathscr{A}^{\boldsymbol{g}},\mathscr{A}^{\boldsymbol{g}},\mathscr{A}^{\boldsymbol{g}},\mathscr{A}^{\boldsymbol{g}},\mathscr{A}^{\boldsymbol{g}},\mathscr{A}^{\boldsymbol{g}},\mathscr{F}^{\boldsymbol{g}}_{\lambda,1},\mathscr{F}^{\boldsymbol{g}}_{\lambda,2},\mathscr{F}^{\boldsymbol{g}}_{\lambda,3},\mathscr{G}^{\boldsymbol{g}}_{\lambda,1},\mathscr{G}^{\boldsymbol{g}}_{\lambda,2},\mathscr{G}^{\boldsymbol{g}}_{\lambda,3}\Big\}, \tag{3.53}$$

yield less non-zero terms than others when considering (3.18)–(3.50) on top of a Hall basis.

This is indeed the case, and having this in mind, when faced with the task of decreasing the amount of linear algebra in the exact integration of (3.1), one should then search for a bijection from, respectively, (3.12)–(3.14) to (3.51)–(3.53), that minimizes the number of non-zero terms in a Hall basis when taking (3.18)–(3.50) into account.

The last equivalence is then particularly useful given that the number of distinct bijections between elements with repetition is often much smaller than the total number of bijections.

If even the number of distinct permutation is so large that it is not practical to search for a minimizer by brute-force, one might benefit from simulated annealing (Press et al., 2007, Section 10.12).

As recorded in Table 1 under "Reduced Hall basis with non-zero integrands", the second mechanism leads to substantial savings when compared with a vanilla Hall basis that does not take (3.18)–(3.50) into account.

Finally, the third mechanism above follows simply from the fact that, as stated, certain linear combinations between various integrands integrate exactly to zero. Thus, one should search such occurrences to further decrease the amount of linear algebra in the exact integration of (3.1), as depicted in Table 1 under "Reduced Hall basis with non-zero integrals".

To conclude, it is intriguing to observe in Table 1 that, when compared with the rest of the eigenvalue range (2.3)–(2.4), it is the intermediary regime (2.2), which, as discussed in Section 1, is not covered by alternative techniques, that requires the largest amount of linear algebra per evaluation of $\lambda \mapsto \tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,j,\mathcal{T}_{l-1}}(c_k, c_{k+1})$.

3.2 Self-adjoint basis and graded FLA

In passing, it is important to say at this point that, in principle, one could call upon yet another mechanism to further reduce the number of commutators and amount of linear algebra. This is the theory of the graded FLA introduced by the the self-adjoint basis put forth in (Munthe–Kaas and Owren, 1999, Subsection 4.a) and further discussed in (Iserles et al., 2000, Subsection 5.2), for settings without oscillatory behaviour.

While preparing the current manuscript, these ideas were implemented and tested on Fer streamers, in the eigenvalue range (2.4). However, it has been found that the graded FLA induced by the self-adjoint basis in essence destroys the advantages gained from the deliberate interpolation of $\boldsymbol{B}^{\text{fine}}_{\lambda,1}(c_k, c_k + h_k t)$ in Definition 2.5 at the right-boundary point, $t = 1$, that, as explained carefully in (Ramos, 2015*a*), reduces the quadrature error in many cases when high oscillation is present. In addition, by construction, the self-adjoint basis and its graded FLA in (Munthe–Kaas and Owren, 1999; Iserles et al., 2000) are advantageous when the step size h is close to 0, but, in practice, it often happens that Fer streamers work well even with h near to 1, so that one cannot reap the benefits from the self-adjoint basis, unless needlessly reducing h. Because of all this, although implemented, this graded FLA was discarded from the Fer streamers' MATLAB package that accompanies the current paper.

3.3 Shooting and root-finding methods

Having examined the practical implementation of $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ in Subsections 3.1–3.2, the following and subsequent subsubsections discuss the use of $\tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$ to approximate the eigenvalues of (1.1)–(1.2). Towards this end, as touched upon in Subsection 1.4, one employs shooting and root-finding methods.

			Eigenvalue range (2.2)				Eigenvalue range (2.3)				Eigenvalue range (2.4)			
			Free magma	Hall basis	'Reduced' Hall basis with non-zero integrands (3.18) up to (3.47)	'Reduced' Hall basis with non-zero integrals	Free magma	Hall basis	'Reduced' Hall basis with non-zero integrands (3.23)	'Reduced' Hall basis with non-zero integrals	Free magma	Hall basis	'Reduced' Hall basis with non-zero integrands (3.18), (3.45), (3.48) up to (3.50)	'Reduced' Hall basis with non-zero integrals
global order	4	$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_3}$	15	15	15	15	5	5	5	5	12	12	12	12
	7	$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_6}$	27	27	27	27	8	8	8	8	24	24	24	24
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,2,\mathcal{T}_3}$	225	105	92	92	25	10	10	10	144	66	51	49
	10	$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_9}$	39	39	39	39	11	11	11	11	36	36	36	36
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,2,\mathcal{T}_6}$	729	351	308	308	64	28	28	28	576	276	210	205
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,3,\mathcal{T}_3}$	3375	1120	≤ 412	≤ 412	125	40	38	34	1728	572	≤ 408	≤ 404
	13	$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,1,\mathcal{T}_{12}}$	51	51	51	51	14	14	14	14	48	48	48	48
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,2,\mathcal{T}_9}$	1521	741	650	650	121	55	55	55	1296	630	477	469
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,3,\mathcal{T}_6}$	19683	6552	≤ 5072	≤ 5072	512	168	166	154	13824	4600	≤ 3236	≤ 3201
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,4,\mathcal{T}_3}$	50625	12600	≤ 7899	≤ 7867	625	150	134	126	20736	5148	≤ 3069	≤ 3060
		$\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,5,\mathcal{T}_3}$	50625	5460	≤ 3444	≤ 3438	625	45	45	42	20736	2145	≤ 1182	≤ 1082

Table 1: Number of terms in each integrand of $\tilde{\boldsymbol{I}}^{\text{fine}}_{\lambda,j,\mathcal{T}_{l-1}}$, for each eigenvalue range and different global orders

3.3.1 Shooting via Prüfer's scaled representation

For shooting, the Fer streamers' MATLAB package employs a scaled Prüfer transformation (Pryce, 1993, Section 5) (Zettl, 2005, p. 81–87). More concretely, the shooting is based on modified versions of the stabilized shooting algorithm in (Pruess and Fulton, 1993, p. 364–367) and the scaled Prüfer transformation in (Ixaru et al., 1999, p. 263–265). As is well-known in the literature, the idea is ingenious: in its simplest form, an unscaled Prüfer transformation follows from representing $(y'_\lambda(t), y_\lambda(t))$ in polar coordinates:

$$\begin{aligned} y_\lambda(t) &=: r_\lambda(t)\sin(\theta_\lambda(t)) \\ y'_\lambda(t) &=: r_\lambda(t)\cos(\theta_\lambda(t)) \end{aligned}$$

which recasts

$$\begin{bmatrix} y_\lambda(t) \\ y'_\lambda(t) \end{bmatrix}' = \begin{bmatrix} 0 & 1 \\ q(t)-\lambda & 0 \end{bmatrix} \begin{bmatrix} y_\lambda(t) \\ y'_\lambda(t) \end{bmatrix}$$

into

$$\begin{cases} \theta'_\lambda(t) = \cos^2(\theta_\lambda(t)) + (\lambda - q(t))\sin^2(\theta_\lambda(t)) \\ r'_\lambda(t) = (1+q(t)-\lambda)\sin(\theta_\lambda(t))\cos(\theta_\lambda(t))r_\lambda(t) \end{cases}$$

and yields the seminal result:

Theorem 3.1 ((Pryce, 1993, Theorem 5.2), (Zettl, 2005, Theorem 4.6.2))**.** *Let $\alpha \in [0,\pi)$ and $\beta \in (0,\pi]$ be such that*

$$\begin{aligned} \tan(\alpha) &:= -\alpha_2/\alpha_1 \quad \text{if} \quad \alpha_1 \neq 0, \quad \text{and} \quad \alpha := \pi/2 \quad \text{if} \quad \alpha_1 = 0, \\ \tan(\beta) &:= -\beta_2/\beta_1 \quad \text{if} \quad \beta_1 \neq 0, \quad \text{and} \quad \beta := \pi/2 \quad \text{if} \quad \beta_1 = 0. \end{aligned}$$

Then each eigenvalue λ_j, $j \in \mathbb{Z}_0^+$, is the unique solution $\lambda = \lambda_j$ of the equation

$$\theta_\lambda(b) = \beta + j\pi, \tag{3.54}$$

where, for each $\lambda \in \mathbb{R}$, $t \mapsto \theta_\lambda(t)$ is the solution determined by the initial condition

$$\theta_\lambda(a) := \alpha. \tag{3.55}$$

Using similar procedures to (Pruess and Fulton, 1993, p. 364–367) and (Ixaru et al., 1999, p. 263–265), each function $\lambda \mapsto \theta_\lambda(c_{k+1})$ can be uniformly approximated via the uniform approximations $\lambda \mapsto \tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(a), \lambda \mapsto \tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_1), \ldots, \lambda \mapsto \tilde{\tilde{\boldsymbol{Y}}}_{\lambda,n}(c_{k+1})$. With an uniform approximation $\lambda \mapsto \tilde{\tilde{\theta}}_{\lambda,n}(c_{k+1})$ in hand, one can then approximate λ_j, the solution to (3.54) with initial condition (3.55), by $\tilde{\tilde{\lambda}}_{j,n}$, the solution to

$$\tilde{\tilde{\theta}}_{\lambda,n}(b) = \beta + j\pi, \tag{3.56}$$

with initial condition

$$\tilde{\tilde{\theta}}_{\lambda,n}(a) := \alpha.$$

Since, in general, one cannot solve (3.56), i.e., root-find $\lambda \mapsto (\tilde{\tilde{\theta}}_{\lambda,n}(b) - \beta - j\pi)$, in closed-form, instead, as discussed in the next subsubsection, one calls upon a root-finding algorithm, which approximates $\tilde{\tilde{\lambda}}_{j,n}$ by $\tilde{\tilde{\tilde{\lambda}}}_{j,n}$, up to prescribed tolerance.

3.3.2 Root-finding via Brent's method

As mentioned in the previous subsubsection, the Fer streamers' MATLAB package outputs an approximation $\tilde{\tilde{\tilde{\lambda}}}_{j,n}$ to $\tilde{\tilde{\lambda}}_{j,n}$, which is the result of applying a root-finding algorithm to $\lambda \mapsto (\tilde{\tilde{\theta}}_{\lambda,n}(b) - \beta - j\pi)$, up to stipulated tolerance. In particular, the Fer streamers' MATLAB package uses Brent's method (Brent, 2002, Section 4) for root-finding, rather than the more popular (and often faster) Newton's method, simply because, unlike with Newton's method, Brent's method always brackets a root and possesses robust convergence properties.

3.4 Heuristics for mesh selection and error estimation

The Fer streamers' MATLAB package uses a nested rule with uniform global orders $\{g_1, g_2, g_3\}$, where $g_1 < g_2 < g_3$. There exist two options in the current version: $\{g_1, g_2, g_3\} := \{4, 7, 10\}$ and $\{g_1, g_2, g_3\} := \{7, 10, 13\}$; the latter being the default. Since the interpolation points in Definition 2.5 are nested, i.e., $\mathcal{T}_3 \subseteq \mathcal{T}_6 \subseteq \mathcal{T}_9 \subseteq \mathcal{T}_{12}$, nested rules do not incur extra function evaluations of the potential q.

3.4.1 Mesh selection

The motivation for the mesh selection is straightforward: keep to a minimum the number of function evaluations of the potential q. In particular, the Fer streamers' MATLAB package employs a modified version of the mesh selection in (Ixaru et al., 1997, p. 305–306) and (Ledoux et al., 2010, p. 764–765).

In short, for each $[c_k, c_{k+1}^{\text{trial}}]$, the mesh selection is based on a local difference between Fer streamers with uniform local orders $\{g_2 + 1, g_3 + 1\}$, which is tested on

$$\lambda \in \left\{ q_{\min}, \frac{\int_{c_k}^{c_{k+1}^{\text{trial}}} q(\xi) d\xi}{c_{k+1}^{\text{trial}} - c_k}, q_{\max} \right\}, \tag{3.57}$$

where $q_{\min}$ only needs to be a lower bound for the minimum of the potential and $q_{\max}$ only needs to be an upper bound for its maximum. With the uniform guarantees from Theorem 2.6, the specific choice (3.57) is motivated by the magnitude and behaviour of the first Fer streamer in Remark 2.1, discussed at length in Section 2. Once computed, the numerical mesh remains unaltered from start to finish.

3.4.2 Error estimation

Having computed the mesh, the Fer streamers MATLAB package first brackets the eigenvalues with uniform global g_1. With these preliminary brackets, the package then runs with uniform global orders g_2 and g_3. The error estimation of the absolute error and relative error are then given by, respectively, the absolute and relative errors between

$$\tilde{\tilde{\lambda}}_{j,(\log((g_2+2)/3)/\log(2))} \text{ and } \tilde{\tilde{\lambda}}_{j,(\log((g_3+2)/3)/\log(2))}. \tag{3.58}$$

4 Calling the Fer streamers MATLAB package

The Fer streamers MATLAB package can be downloaded from (Ramos, 2015*c*), which accompanies the current paper (Ramos, 2015*b*). The root file `m_index.m` sets up the input, calls the main file `m_Fer_streamers.m` and provides the output.

4.1 Input

To run the main file `m_Fer_streamers.m`, the root file `m_index.m` sets up the input:

- `parameter`:

which serves to parameterize the Sturm–Liouville problem if necessary, otherwise set to empty,

- a, b, q, $q_{\min}$, $q_{\max}$, α_1, α_2, β_1, β_2:

which characterize the Sturm–Liouville problem, where $q_{\min}$ only needs to be a lower bound for the minimum of the potential and $q_{\max}$ only needs to be an upper bound for its maximum,

- `index_range_to_eigenvalues_or_eigenvalue_range_to_eigenvalues`, `range_min`, `range_max`:

that set up whether the eigenvalues should be computed according to their indices or values, and on which range, as well as,

- `error_absolute_or_relative`, `tol_stopping_criteria`:

that set up the tolerance and type of error that should be used for the stopping criteria.

4.2 Output

The main file `m_Fer_streamers.m` outputs:

- `all_t_and_q_at_t_pairs`:

a two column matrix that contains per line all evaluations $(t, q(t))$ used or discarded from start to finish,

- `eigenvalues_indices_absoluteErrors_relativeErrors`:

a four column matrix which collects per line the requested eigenvalues, their indices and an error estimation of the absolute and relative errors in their approximation, as described in Subsubsection 3.4.2.

5 Numerical results

To illustrate the Fer streamers' MATLAB package with nested uniform global orders $\{7, 10, 13\}$ (c.f., Subsection 3.4), the numerical results in this section describe its output on three Sturm–Liouville problems (5.1), (5.2) and (5.3) below, when set to approximate their first 500 eigenvalues:

- `index_range_to_eigenvalues_or_eigenvalue_range_to_eigenvalues` =`'index_range_to_eigenvalues'`, `range_min`=0, `range_max`=499,

up to prescribed absolute error with tolerance 10^{-8}:

- `error_absolute_or_relative`=`'absolute'`, `tol_stopping_criteria`=10^{-8}.

To cover different phenomena, this sections then examines the second Paine problem (Pryce, 1993, p. 281) defined by (1.1)–(1.2) together with:

$$\begin{gathered} a = 0, \quad b = \pi, \quad q(t) = \frac{1}{(t + 1/10)^2}, \quad q_{\min} \geq q(b) - 1, \quad q_{\max} = q(a), \\ y_\lambda(a) = y_\lambda(b) = 0, \quad \alpha_1 \neq 0, \quad \beta_1 \neq 0, \quad \alpha_2 = \beta_2 = 0, \end{gathered} \tag{5.1}$$

the Coffey–Evans problem (Pryce, 1993, p. 283) with `parameter`=β, defined by (1.1)–(1.2) together with:

$$\begin{gathered} \beta = 30, \quad a = -\frac{\pi}{2}, \quad b = \frac{\pi}{2}, \quad q(t) = -2\beta\cos(2t) + \beta^2\sin(2t)^2, \quad q_{\min} = -2\beta, \\ q_{\max} = \beta^2 + 1, \quad y_\lambda(a) = y_\lambda(b) = 0, \quad \alpha_1 \neq 0, \quad \beta_1 \neq 0, \quad \alpha_2 = \beta_2 = 0, \end{gathered} \tag{5.2}$$

as well as the Truncated Gelfand–Levitan problem (Pryce, 1993, p. 283) given by (1.1)–(1.2) combined with:

$$\begin{gathered} a = 0, \quad b = 100, \quad q(t) = \frac{32\cos(t)(\cos(t) + (2 + t)\sin(t))}{(4 + 2t + \sin(2t))^2}, \quad q_{\min} \geq -1, \quad q_{\max} \leq 2, \\ y_\lambda(a) + y'_\lambda(a) = 0, \quad y_\lambda(b) = 0, \quad \alpha_1 = \alpha_2 \neq 0, \quad \beta_1 \neq 0, \quad \beta_2 = 0. \end{gathered} \tag{5.3}$$

The numerical results in Figure 1 illustrate the output from Fer streamers' MATLAB package with nested uniform global orders $\{7, 10, 13\}$ and absolute error tolerance 10^{-8} for the first 500 eigenvalues of the Sturm–Liouville problems (5.1), (5.2) and (5.3). Apart from the number of function evaluations of the potential q used or discarded by Fer streamers, each plot displays the estimated absolute/relative error by Fer streamers as defined in Subsubsection 3.4.2 together with the actual absolute/relative error when compared with a reference solution, computed with MATSLISE's package (Ledoux et al., 2005). Absent circles mean that the

estimated error by Fer streamers is equal to zero, i.e., that (3.58) with $g_2 = 10$ and $g_3 = 13$ are identical in machine precision. In particular, one can see that the estimated error by Fer streamers is in-line with the prescribed absolute error tolerance 10^{-8}, being often quite conservative.

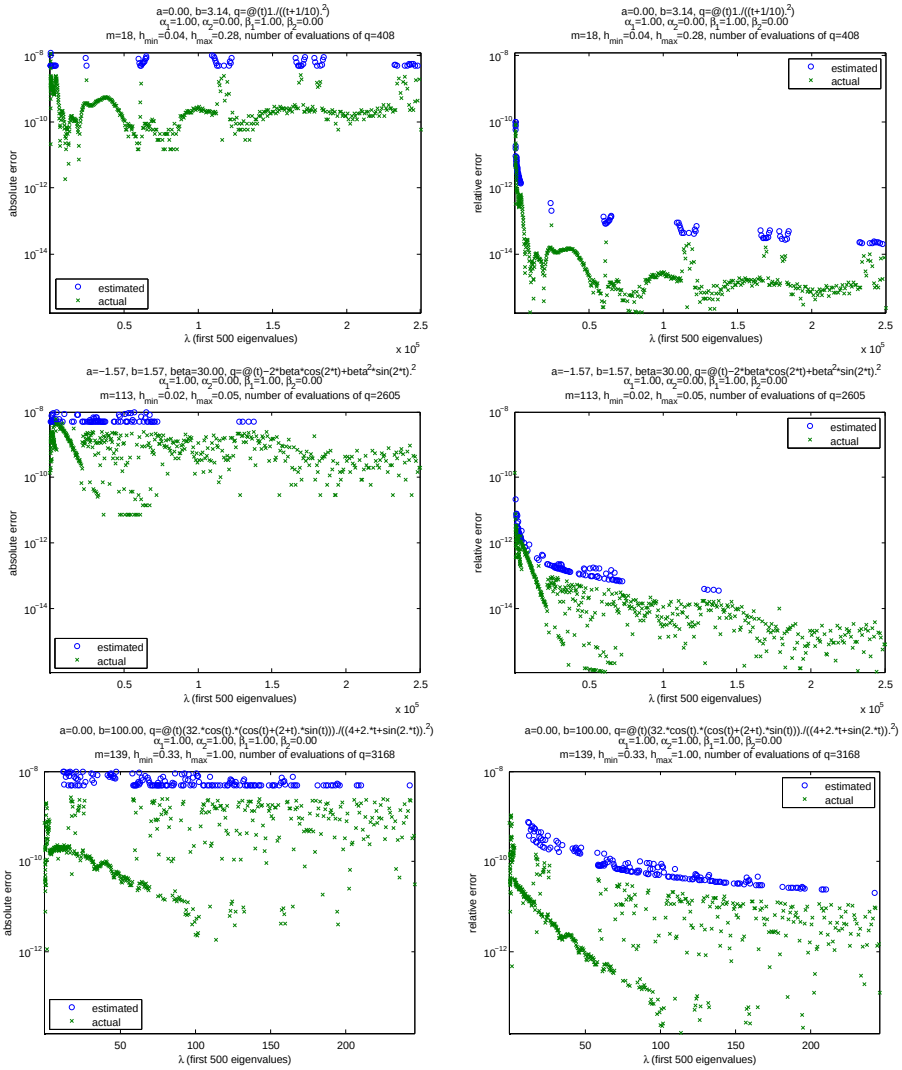

Figure 1: Absolute error (left) and relative error (right) with Fer streamers' MATLAB package with nested uniform global orders $\{7, 10, 13\}$ and absolute error tolerance 10^{-8} for the first 500 eigenvalues of the second Paine problem (5.1) (top), the Coffey–Evans problem (5.2) (middle) and the truncated Gelfand–Levitan problem (5.3) (bottom). Apart from the number of evaluations of q used or discarded by Fer streamers, each plot displays the estimated error by Fer streamers as defined in Subsubsection 3.4.2 together with the actual error when compared with a reference solution. Absent circles signify that the estimated error by Fer streamers equals zero, i.e., that (3.58) with $g_2 = 10$ and $g_3 = 13$ coincide up to machine precision.

Acknowledgments

The author would like to thank Arieh Iserles for useful comments and suggestions. This work was supported by Fundação para a Ciência e a Tecnologia, Portugal through the fellowship SFRH/BD/71692/2010.

References

Brent, R. (2002), *Algorithms for minimization without derivatives*, reprint edn, Dover, New York.

Degani, I. (2004), RCMS - Right Correction Magnus Schemes for Oscillatory ODEs, and Cubature Formulae and Commuting Extensions, PhD thesis, Weizmann institute of science, dept. of mathematics.

Degani, I. and Schiff, J. (2006), 'RCMS: Right Correction Magnus Series approach for oscillatory ODEs', *Journal of Computational and Applied Mathematics* **193**(2), 413–436.

Engø, K., Marthinsen, A. and Munthe–Kaas, H. Z. (1999), DiffMan — an object oriented MATLAB toolbox for solving differential equations on manifolds, Technical report, Department of Computer Science, University of Bergen, Norway. Available at: (http://www.diffman.no/).

Fer, F. (1958), 'Résolution del l'equation matricielle $\dot{U} = pU$ par produit infini d'exponentielles matricielles', *Bulletin de la Classe des Sciences Académie Royale de Belgique* **44**, 818–829.

Iserles, A. (2004*a*), 'On the method of Neumann series for highly oscillatory equations', *BIT Numerical Mathematics* **44**(3), 473–488.

Iserles, A. (2004*b*), 'On the numerical quadrature of highly-oscillating integrals I: Fourier transforms', *IMA Journal of Numerical Analysis* **24**(3), 365–391.

Iserles, A., Munthe–Kaas, H. Z., Nørsett, S. and Zanna, A. (2000), 'Lie-group methods', *Acta Numerica* **9**, 215–365.

Ixaru, L. G. (2000), 'CP methods for the Schrödinger equation', *Journal of Computational and Applied Mathematics* **125**(1-2), 347–357.

Ixaru, L. G., De Meyer, H. and Berghe, G. V. (1997), 'CP methods for the Schrödinger equation revisited', *Journal of Computational and Applied Mathematics* **88**(2), 289–314.

Ixaru, L. G., De Meyer, H. and Berghe, G. V. (1999), 'SLCPM12 — A program for solving regular Sturm–Liouville problems', *Computer Physics Communications* **118**(2-3), 259–277.

Ledoux, V. and Daele, M. V. (2010), 'Solution of Sturm–Liouville problems using modified Neumann schemes', *SIAM Journal on Scientific Computing* **32**(2), 563–584.

Ledoux, V., Daele, M. V. and Berghe, G. V. (2004), 'CP methods of higher order for Sturm–Liouville and Schrödinger equations', *Computer Physics Communications* **162**(3), 151–165.

Ledoux, V., Daele, M. V. and Berghe, G. V. (2005), 'MATSLISE: A MATLAB package for the numerical solution of Sturm–Liouville and Schrödinger equations', *ACM Transactions on Mathematical Software* **31**(4), 532–554.

Ledoux, V., Daele, M. V. and Berghe, G. V. (2010), 'Efficient numerical solution of the 1D Schrödinger eigenvalue problem using Magnus integrators', *IMA Journal of Numerical Analysis* **30**, 751–776.

Marletta, M. and Pryce, J. D. (1992), 'Automatic solution of Sturm–Liouville problems using the Pruess method', *Journal of Computational and Applied Mathematics* **39**(1), 57–78.

Munthe–Kaas, H. and Owren, B. (1999), 'Computations in a free lie algebra', *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **357**(1754), 957–981.

Paine, J. and de Hoog, F. (1980), 'Uniform estimation of the eigenvalues of Sturm–Liouville problems', *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics* **21**(3), 365–383.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007), *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3 edn, Cambridge University Press, New York, NY, USA.

Pruess, S. (1973), 'Estimating the eigenvalues of Sturm–Liouville problems by approximating the differential equation', *SIAM Journal on Numerical Analysis* **10**(1), 55–68.

Pruess, S. and Fulton, C. T. (1993), 'Mathematical software for Sturm–Liouville problems', *ACM Transactions on Mathematical Software* **19**(3), 360–376.

Pryce, J. D. (1993), *Numerical Solution of Sturm–Liouville Problems*, Oxford University Press.

Ramos, A. G. C. P. (2015*a*), Uniform and high-order discretization schemes for Sturm–Liouville problems via Fer streamers, Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, United Kingdom. Submitted.

Ramos, A. G. C. P. (2015*b*), Uniform and high-order practical implementation of Sturm–Liouville problems via Fer streamers, Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, United Kingdom. Submitted.

Ramos, A. G. C. P. (2015*c*), 'Uniform and high-order MATLAB software for Sturm–Liouville problems via Fer streamers'. Provided as supplementary material to (Ramos, 2015*b*). Available at: (http://www.damtp.cam.ac.uk/user/agcpr2/documents/SL_via_Fer_streamers.zip).

Ramos, A. G. C. P. and Iserles, A. (2015), 'Numerical solution of Sturm–Liouville problems via Fer streamers', *Numerische Mathematik* **131**(3), 541–565.

Zettl, A. (2005), *Sturm–Liouville Theory*, American Mathematical Society.