# Validated and numerically efficient Chebyshev spectral methods for linear ordinary differential equations

Florent Bréhard, Nicolas Brisebarre, Mioara Joldes

▶ **To cite this version:**

Florent Bréhard, Nicolas Brisebarre, Mioara Joldes. Validated and numerically efficient Chebyshev spectral methods for linear ordinary differential equations. ACM Transactions on Mathematical Software, In press. hal-01526272v2

**HAL Id: hal-01526272**
**https://hal.science/hal-01526272v2**

Submitted on 28 Jul 2017 (v2), last revised 5 Jun 2018 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Validated and numerically efficient Chebyshev spectral methods for linear ordinary differential equations

Florent Bréhard* Nicolas Brisebarre† Mioara Joldeş‡

**Abstract**

In this work we develop a validated numerics method for the solution of linear ordinary differential equations (LODEs). A wide range of algorithms (i.e., Runge-Kutta, collocation, spectral methods) exist for numerically computing approximations of the solutions. Most of these come with proofs of asymptotic convergence, but usually, provided error bounds are non-constructive. However, in some domains like critical systems and computer-aided mathematical proofs, one needs validated effective error bounds. We focus on both the theoretical and practical complexity analysis of a so-called *a posteriori* quasi-Newton validation method, which mainly relies on a fixed-point argument of a contracting map. Specifically, given a polynomial approximation, obtained by some numerical algorithm and expressed in Chebyshev basis, our algorithm efficiently computes an accurate and rigorous error bound. For this, we study theoretical properties like compactness, convergence, invertibility of associated linear integral operators and their truncations in a suitable coefficient space of Chebyshev series. Then, we analyze the almost-banded matrix structure of these operators, which allows for very efficient numerical algorithms for both numerical solutions of LODEs and rigorous computation of the approximation error. Finally, several representative examples show the advantages of our algorithms as well as their theoretical and practical limits.

## 1 Introduction

Solutions of Linear Ordinary Differential Equations (LODEs) are ubiquitous in modeling and solving common problems. Examples include elementary and special functions evaluation, manipulation or plotting, numerical integration, or locally solving nonlinear problems using linearizations.

While many numerical methods have been developed over time [24], in some areas like safety-critical systems or computer-assisted proofs [49], numerical approximations are not sufficiently reliable and one is interested not only in computing *approximations*, but also *enclosures* of the approximation errors [43].

*LIP, ENS Lyon, 46 Allée d'Italie, 69364 Lyon Cedex 07, France and LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France (florent.brehard@ens-lyon.fr).

†CNRS, LIP, ENS Lyon, 46 Allée d'Italie, 69364 Lyon Cedex 07, France (nicolas.brisebarre@ens-lyon.fr).

‡CNRS, LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France (joldes@laas.fr)

The width of such an enclosure gives an effective quality measurement of the computation, and can be used to adaptively improve accuracy at run-time. Most often, machine approximations rely on polynomials [34], since they are compact to store and efficient to evaluate and manipulate via basic arithmetic operations implemented in hardware on current processors. For widely used functions, such polynomial approximations used to be tabulated in handbooks [1]. Nowadays, computer algebra systems also provide symbolic solutions when possible, but usually they are handled through numeric routines. However, when bounds for the approximation errors are available, they are not guaranteed to be accurate and are sometimes unreliable.

Our contribution is an efficient algorithm for computing *rigorous polynomial approximations* (RPAs) for LODEs, that is to say a polynomial approximation to the solution of the LODE together with a rigorous error bound. More specifically, we deal with the following problem:

**Problem 1.1.** *Let $r$ be a positive integer, $\alpha_0, \alpha_1, \ldots, \alpha_{r-1}$ and $\gamma$ continuous functions over $[-1,1]$. Consider the LODE*

$$f^{(r)}(t) + \alpha_{r-1}(t) f^{(r-1)}(t) + \cdots + \alpha_1(t) f'(t) + \alpha_0(t) f(t) = \gamma(t),\ t \in [-1,1], \quad (1)$$

*together with conditions uniquely characterizing the solution:*

*a) For an* initial value problem *(IVP), consider:*

$$\mathbf{\Lambda} \cdot f := (f(t_0), f'(t_0), \ldots, f^{(r-1)}(t_0)) = (v_0, v_1, \ldots, v_{r-1}) \quad (1a)$$

*for given $t_0 \in [-1,1]$ and $(v_0, v_1, \ldots, v_{r-1}) \in \mathbb{R}^r$.*

*b) For a generalized* boundary value problem *(BVP), conditions are given by $r$ linearly independent linear functionals $\lambda_i : \mathcal{C}^0 \to \mathbb{R}$:*

$$\mathbf{\Lambda} \cdot f := (\lambda_0(f), \ldots, \lambda_{r-1}(f)) = (\ell_0, \ldots, \ell_{r-1}) \quad (1b)$$

*for given $(\ell_0, \ldots, \ell_{r-1}) \in \mathbb{R}^r$.*

*Given an approximation degree $p \in \mathbb{N}$, find the coefficients of a polynomial $\varphi(t) = \sum_{n=0}^{p} c_n T_n(t)$ written in Chebyshev basis $(T_n)$, together with a tight and rigorous error bound $\eta$ such that $\|f - \varphi\|_\infty := \sup_{t\in[-1,1]} |f(t) - \varphi(t)| \leqslant \eta$, where $\|\cdot\|_\infty$ denotes the supremum norm over $[-1,1]$.*

## 1.1 Previous works

Within the scopes of RPAs, all sources of inaccuracies such as *rounding errors* and *method approximation errors* must be taken into account without compromising efficiency, in order to render the whole process rigorous from a mathematical point of view. One important tool used is interval analysis [32, 33, 36, 43, 49]. In general, replacing all numerical computations with interval ones does not yield a tight enclosure of rounding or truncation errors: issues like overestimation or wrapping effect [37] are often present. So, important care has to be put in the adaptation of numerical algorithms to rigorous computing, which often accounts for finding suitable symbolic-numeric objects that maintain both the efficiency and the reliability of computations.

Previous works computed such RPAs using either Taylor series [30, 35, 37], Chebyshev interpolants and truncated Chebyshev series [8, 25], minimax polynomials [10] (best approximation polynomials with respect to supremum norm).

Broadly speaking, the idea of working with polynomial approximations instead of functions is analogous to using floating-point arithmetic instead of real numbers. A first work in this sense, was the *ultra-arithmetic* [15, 16, 27], where various *generalized Fourier series*, including Chebyshev series, play the role of floating-point numbers. A purely numeric approach to this idea is Trefethen's Chebfun [46, 13, 47]. Other methods for computing numerical Chebyshev series include for instance spectral or pseudospectral methods such as Galerkin, tau or collocation methods [21], Clenshaw's algorithm [11] or Olver and Townsend's fast algorithm for LODEs [38].

However, ultra-arithmetic also comprises a function space counterpart of interval arithmetic, based on truncated series with interval coefficients and rigorous truncation error bounds. The main appeal of this approach is the ability to rigorously solve functional equations using enclosure methods [33, 27, 30, 37]. The great majority of these so-called *higher order enclosure methods* were developed based on Taylor approximations (called Taylor Models, Taylor Forms or Taylor Differential Algebra) [30, 12, 37], due to their simplicity. However, their well-known shortcoming is their limited local convergence properties. Chebyshev or Fourier series are superior [37, 6, 47] for real function approximation and their use in validated computing was recently revived [8, 4, 29].

Concerning validation methods, two classes can be distinguished: firstly, *self-validating* methods produce an approximation together with a rigorous error bound at each step. This is typically done for basic tasks like arithmetic operations, i.e. addition, multiplication, etc., on RPAs. Secondly, when working with function spaces, for instance, *a posteriori* validation methods are used. They consist in computing a validated approximation error bound, given a numerical approximation which was independently computed by some numerical algorithm. Most of these methods rely on fixed-point theorems combined with a quasi-Newton approach [52, 29, 50, 23]. These methods focus on *ad-hoc* solutions for specific problems with little emphasis on a generic algorithmic approach, as well as its complexity study.

In [4], one of the authors of this article proposed an *a posteriori* validation method, based on convergent Neumann series of linear operators in the Banach space of continuous functions $(\mathcal{C}^0, \|\cdot\|_\infty)$, for efficient RPAs solutions of linear ordinary differential equations (LODE) with polynomial coefficients, also called *D-finite functions* [45]. This class of functions includes many elementary (e.g., exp, sin, cos) and special functions (e.g., Airy, Bessel, erf) commonly used in mathematical physics. D-finite functions have an efficient symbolic-numeric algorithmic treatment [53, 44], which allowed for the study of the *complexity* of validated enclosure methods, from a computer algebra point of view. In this article, we extend this complexity study to the framework of quasi-Newton validation methods for Problem 1.1.

## 1.2 General setting for quasi-Newton validation

Consider the equation $\mathbf{F} \cdot x = 0$ where $\mathbf{F}$ is an operator acting on a Banach space $(E, \|\cdot\|)$. A numerical method provides an approximation $\widetilde{x}$ of some exact solution $x$. One is interested in rigorously bounding the approximation error

between $x$ and $\widetilde{x}$. For that, a classical idea is to reformulate the problem as a fixed-point equation $\mathbf{T} \cdot x = x$ with $\mathbf{T} : E \to E$ an operator whose fixed points correspond to the zeros of $\mathbf{F}$. The distance between a given approximation and a fixed point of $\mathbf{T}$ is bounded based on the following theorem [5, Thm 2.1]:

**Theorem 1.** *Let $(E, \|\cdot\|)$ be a Banach space, $\mathbf{T} : E \to E$ a continuous operator and $\widetilde{x} \in E$ an approximate solution of the fixed-point equation $\mathbf{T} \cdot x = x$. If there is a radius $r > 0$ such that*

- $\mathbf{T} \cdot \overline{B}(\widetilde{x}, r) := \{\mathbf{T} \cdot x \mid \|x - \widetilde{x}\| \leqslant r\} \subseteq \overline{B}(\widetilde{x}, r) := \{x \mid \|x - \widetilde{x}\| \leqslant r\}$, *and*
- $\mathbf{T}$ *is contracting over $\overline{B}(\widetilde{x}, r)$: there exists a constant $\mu \in (0, 1)$ such that for all $x_1, x_2 \in \overline{B}(\widetilde{x}, r)$, $\|\mathbf{T} \cdot x_1 - \mathbf{T} \cdot x_2\| \leqslant \mu \|x_1 - x_2\|$,*

*then $\mathbf{T}$ admits a unique fixed point $x^*$ in $\overline{B}(\widetilde{x}, r)$ and we have the following enclosure of the approximation error:*

$$\frac{\|\mathbf{T} \cdot \widetilde{x} - \widetilde{x}\|}{1 + \mu} \leqslant \|x^* - \widetilde{x}\| \leqslant \frac{\|\mathbf{T} \cdot \widetilde{x} - \widetilde{x}\|}{1 - \mu}.$$

One special class of such operators $\mathbf{T}$ are the Newton-like operators acting on Banach spaces (see [40, Chap.4] and references therein). Suppose that $\mathbf{F}$ is of class $\mathcal{C}^2$ over $E$, and suppose that $\mathbf{A} = (\mathrm{d}\mathbf{F}|_{x=\widetilde{x}})^{-1}$ exists. Then the fixed points of:

$$\mathbf{T} = \mathbf{I} - \mathbf{A} \cdot \mathbf{F} : E \to E \tag{2}$$

are exactly the zeros of $\mathbf{F}$ and $\mathbf{T}$ has a null derivative at $\widetilde{x}$, so that it is locally contracting around $\widetilde{x}$. Hence, if for a well-chosen $r > 0$, the hypotheses of Theorem 1 are respected, one obtains an upper bound for the approximation error $\|x^* - \widetilde{x}\|$. In general however, we cannot exactly compute $(\mathrm{d}\mathbf{F}|_{x=\widetilde{x}})^{-1}$ and $\mathbf{A}$ is only an approximation. Still, this may be sufficient to get a contracting operator $\mathbf{T}$ around $\widetilde{x}$.

**Remark 1.1.** *Since this article exclusively deals with linear problems, the validation operators $\mathbf{T}$ that we consider are always affine, so that they are contracting if and only if their linear part $\mathcal{D}\mathbf{T}$ has operator norm $\|\mathcal{D}\mathbf{T}\| = \mu < 1$. In particular, for an affine operator $\mathbf{T}$, being locally or globally contracting are equivalent. Therefore, the ball $\overline{B}(\widetilde{x}, r)$ can be replaced by the whole space $E$ in Theorem 1 and the first condition becomes trivially true.*

The general abstract formulation above provides the road-map for our approach, which is mainly focused on both its theoretical and practical complexity analysis, which are modeled as follows.

## 1.3 Computation and complexity model

Our numerical algorithms rely on *floating-point* arithmetics, either in standard double precision, or in arbitrary precision when needed. In the later case, `GNU-MPFR` library [17] is used. For *validated* computations, we make use of *interval arithmetics* via the `MPFI` library [41].

Complexity results are given in the uniform complexity model: all basic arithmetic operations (addition, subtraction, multiplication, division and square root), either in floating-point or interval arithmetics, induce a unit cost of time.

In particular, we do not investigate the incidence of the precision parameter on the global time complexity.

Concerning arithmetic operations on functions, when safe enclosure is needed, we use classes of RPAs called Chebyshev models, specifically defined in Section 2.3.

## 1.4 Overview of our approach and main results

We develop an efficient algorithm for solving Problem 1.1 when the coefficients $\alpha_j$ and the right hand side $\gamma$ are represented by Chebyshev models, which can be done to an arbitrary accuracy under mild regularity assumptions, such as Lipschitz continuity.

The first contribution is the effective construction of a Banach space denoted $\Upsilon^1$ (which plays the role of $E$), together with a suitable norm, and the operators $\mathbf{F}$, $\mathbf{T}$ and $\mathbf{A}$, cf. Equation (2), when dealing with Chebyshev series solutions of linear differential equations. Theoretical properties of the chosen Banach space $\Upsilon^1$, analogous to the Wiener algebra, are given in Section 2. Then, we give in Section 3 a classical integral reformulation of (1). This has the advantage of directly producing a compact operator, yielding appropriate fast convergence results of the solution of truncated linear systems to the exact one (see Theorem 4).

Moreover, in Section 3 we prove an important property from an algorithmic point of view: this compact operator has an *almost-banded matrix representation* when Equation (1) has polynomial coefficients. This leads to the formulation of the following subproblem, where for the sake of simplicity, we focus on the case of an IVP. Note also that approximations over other real or complex segments (in Chebyshev basis adapted to the segment) are reduced to approximations on $[-1,1]$ by means of an affine change of variables.

**Problem 1.2.** *Let $a_0, a_1, \ldots, a_{r-1}, g \in \mathbb{R}[t]$. Consider the LODE*

$$f^{(r)}(t) + a_{r-1}(t)f^{(r-1)}(t) + \cdots + a_1(t)f'(t) + a_0(t)f(t) = g(t),\ t \in [-1,1], \quad (3)$$

*over $[-1,1]$ together with initial conditions at $t_0 = -1$:*

$$f(t_0) = v_0, \quad f'(t_0) = v_1, \quad \ldots \quad , f^{(r-1)}(t_0) = v_{r-1}.$$

*Given $p \in \mathbb{N}$, find the coefficients of $\varphi(t) = \sum_{n=0}^{p} c_n T_n(t)$ and a tight and rigorous error bound $\eta$ such that $\|f - \varphi\|_\infty := \sup_{t\in[-1,1]} |f(t) - \varphi(t)| \leqslant \eta$.*

**Remark 1.2.** *Note that in this problem we focus on the case $t_0 = -1$ for technical reasons explained in Section 5.1, but our results remain valid for any $t_0 \in [-1,1]$.*

This problem is solved with the following steps:

*Step 2.1.* An approximate solution is necessary. This can be provided by the user that is, computed by some numerical algorithm of choice (such as that of [38] or [4]). For completeness of our implementation, we propose a linear (with respect to the approximation degree) time approximation algorithm, which combines the classical integral reformulation mentioned above and the algorithm for almost-banded linear systems from [38], recalled in Section 4.

Then, we develop a new variant of this algorithm, which is efficient (in many practical cases) for obtaining the approximate inverse operator $A$ in Equation (2) and which is required by the next step.

*Step 2.2.* A new algorithm based on Theorem 1 is proposed, which provides the rigorous approximation error bound in Section 5.

In particular, for a *fixed* given LODE, our validation algorithm runs in linear time, in terms of basic arithmetic operations, with respect to the degree $p$ of the approximation to be validated.

Then, we generalize this method in Section 6 in two directions:

- when the coefficients $\alpha_j$ are not polynomials anymore, but functions in $\mathcal{Y}^1$ represented by Chebyshev models,
- and when the conditions are generalized boundary conditions (1b).

This allows us to construct Chebyshev models for a quite large class of functions, starting from $\mathcal{H}_0 = \mathbb{R}[t]$ and defining $\mathcal{H}_{i+1}$ as the solutions of Problem 1.1 where all the $\alpha_j(t)$ and $\gamma(t)$ are in $\mathcal{H}_i$, or some closure of it under other operations like inversion, square root, etc. In fact, if the $\alpha_j(t)$ and $\gamma(t)$ are rigorously approximated by Chebyshev models, then the generalized method gives us a Chebyshev model for the solution. Thus, a chain of recursive calls to the method can be used to approximate any function of $\mathcal{H} = \bigcup_i \mathcal{H}_i$.

Finally, we illustrate our approach with four different examples in Section 7 and conclude in Section 8. A C code implementation is available at `http://homepages.laas.fr/fbrehard`, which includes the core library algorithms as well as the presented examples.

## 2 Function approximation by Chebyshev series

Taylor expansions are among the best established polynomial approximations. For instance, a function $f$, supposed to be analytic at 0, can be approximated by its $n$-th order truncated Taylor series $f(0)+f'(0)t+f''(t)t^2/2+\cdots+f^{(n)}(0)t^n/n!$. This is in some sense the best "infinitesimal" polynomial approximation of $f$ of degree $n$ around 0. Despite its simplicity, Taylor expansion has several drawbacks when uniformly approximating a function $f$ over a given compact interval. The domain of convergence of Taylor series of $f$ at $x_0$ is a complex disc centered at $x_0$ which avoids all the singularities of $f$. Thus when $f$ is not smooth enough on the disc surrounding the considered interval, convergence cannot be ensured and one needs to suitably split the interval and provide a Taylor series for each subsegment. Moreover, even when convergent, the $n$-th order truncated Taylor series of $f$ is usually not the best uniform polynomial approximation of degree $n$ over the segment under consideration. From this point of view, Chebyshev series approximations prove to be a better choice (see also Theorems 2 and 3 below) and excellent accounts for that are given in [6, 9, 18, 31, 39, 47, 42]. In this section, we recall some facts useful in the sequel.

## 2.1 Chebyshev polynomials and Chebyshev series

The Chebyshev family of polynomials is defined using the following three-term recurrence relation:

$$\begin{aligned} T_0(X) = 1, \quad T_1(X) = X, \\ T_{n+2}(X) = 2XT_{n+1}(X) - T_n(X), \quad n \geqslant 0, \end{aligned}$$

which gives a basis for $\mathbb{R}[X]$. Equivalently, $T_n$ is defined to be the only polynomial satisfying $T_n(\cos(\theta)) = \cos(n\theta)$ for all $\theta \in \mathbb{R}$. In particular, one gets that $|T_n(t)| \leqslant 1$ for all $t \in [-1, 1]$. To obtain more symmetric formulas, one can define $T_{-n} := T_n$ for all $n \geqslant 0$, which is consistent with the trigonometric definition of $T_n$.

Similarly to the monomial basis, we have simple formulas for multiplication and (indefinite) integration:

$$\begin{aligned} T_nT_m &= \frac{1}{2}(T_{n+m} + T_{n-m}), \quad n, m \in \mathbb{Z}, \\ \int T_n &= \frac{1}{2}\left(\frac{T_{n+1}}{n+1} - \frac{T_{n-1}}{n-1}\right), \quad n \in \mathbb{Z}, \end{aligned} \tag{4}$$

where $T_{n+1}/(n+1)$, resp. $T_{n-1}/(n-1)$, is 0 by convention when the denominator vanishes (that is, when $n = 1$, resp. $n = -1$). However, contrary to the monomial basis, derivation in the Chebyshev basis does not have a compact expression:

$$T_n' = n \sum_{\substack{|i|<|n| \\ i \neq n \bmod 2}} T_i = \begin{cases} n(T_{-n+1} + \cdots + T_{-1} + T_1 + \cdots + T_{n-1}), & n \text{ even}, \\ n(T_{n-1} + \cdots + T_0 + \cdots + T_{n-1}), & n \text{ odd}. \end{cases} \tag{5}$$

Another important property is that Chebyshev polynomials form a family $(T_n)_{n\geqslant 0}$ of orthogonal polynomials with respect to the following inner product, defined on $L^2$, the space of real-valued measurable functions over $[-1, 1]$ for which $\int_{-1}^{1} f(t)^2(1-t^2)^{-1/2}\,\mathrm{d}t < +\infty$:

$$\langle f, g\rangle := \int_{-1}^{1} \frac{f(t)g(t)}{\sqrt{1-t^2}}\,\mathrm{d}t = \int_0^{\pi} f(\cos\theta)g(\cos\theta)\,\mathrm{d}\theta \in \mathbb{R}, \qquad f, g \in L^2.$$

One has: $\langle T_0, T_0\rangle = \pi$, $\langle T_n, T_n\rangle = \frac{\pi}{2}$, for $n > 0$, and $\langle T_n, T_m\rangle = 0$, for $n \neq m$.

Whence, the $n$-th order Chebyshev coefficient of $f \in L^2$ is defined by:

$$[f]_n := \frac{1}{\pi}\langle f, T_n\rangle = \frac{1}{\pi}\int_0^{\pi} f(\cos\theta)\cos(n\theta)\,\mathrm{d}\theta, \quad n \in \mathbb{Z}. \tag{6}$$

Note that $[f]_{-n} = [f]_n$ for all $n \in \mathbb{Z}$ and the symmetric $n$-th order truncated Chebyshev series of $f$ is defined by:

$$\mathrm{I\!I}_n \cdot f := \sum_{|i|\leqslant n} [f]_i T_i = [f]_{-n}T_{-n} + \cdots + [f]_0T_0 + \cdots + [f]_nT_n, \quad n \geqslant 0.$$

**Remark 2.1.** *Note that we chose a so-called two-sided symmetric expression for the Chebyshev series, but this is exactly the orthogonal projection of $f$ onto the linear subspace spanned by $T_0, T_1, \ldots, T_n$, to which we shall refer as the so-called one-sided expression for Chebyshev series.*

Therefore, $\pi_n \cdot f$ is the best polynomial approximation of $f$ of degree $n$ for the norm $\|\cdot\|_2$ induced by the inner product.

Beside convergence of $\pi_n \cdot f$ to $f$ in $L^2$ [9, Chap. 4], one also has the following result of uniform and absolute convergence [47, Thm. 3.1]:

**Theorem 2.** *If $f$ is Lipschitz continuous on $[-1,1]$, it has a unique representation as a Chebyshev series,*

$$f(x) = \sum_{k=-\infty}^{\infty} [f]_k T_k(x), \text{ with } [f]_{-k} = [f]_k \text{ for all } k \in \mathbb{Z},$$

*which is absolutely and uniformly convergent.*

This theorem shows the effectiveness of approximating by truncated Chebyshev series even when functions have low regularity. Moreover, the smoother $f$ is, the faster its approximants converge. From [47, Thm 7.2], one has that if the $\nu$th derivative of $f$ is of bounded variation $V$, then for a truncation order $n$, the speed of convergence is in $O(Vn^{-\nu})$. According to [47, Thm 8.2] for analytic functions, if $\rho > 0$ and $f$ is analytic in the neighborhood of the set bounded by the Bernstein $\rho$-ellipse $\mathcal{E}_\rho = \{z = (\rho e^{i\theta} + \rho^{-1} e^{-i\theta})/2 \in \mathbb{C} \mid \theta \in [0, 2\pi]\}$ of foci $-1$ and 1, the convergence is in $O(M\rho^{-n})$, where $M$ upper bounds $|f|$ on $\mathcal{E}_\rho$. In particular, for entire functions ($\rho = \infty$), the convergence is faster than any geometric sequence [6].

Moreover, truncated Chebyshev series are near-best approximations with respect to the uniform norm on the space $\mathcal{C}^0 = \mathcal{C}^0([-1,1])$ of continuous functions over $[-1,1]$ [47, Thm. 16.1]:

**Theorem 3.** *Let $n \in \mathbb{N}, n \geqslant 1$ and $f \in \mathcal{C}^0$, let $p_n^*$ denote the polynomial of degree at most $n$ that minimizes $||f - p||_\infty$, Then*

$$||f - \pi_n \cdot f||_\infty \leqslant \left(4 + \frac{4}{\pi^2} \log(n+1)\right) \|f - p_n^*\|_\infty.$$

It turns out that for computing rigorous upper bounds on $||f - \pi_n \cdot f||_\infty$, we need to define a more convenient intermediate norm, which upper bounds the uniform norm, and thus set our approach in a corresponding Banach space defined in what follows.

## 2.2 The Banach space $(\Psi^1, \|\cdot\|_{\Psi^1})$

For a function $f \in \mathcal{C}^0$, we define the quantity:

$$\|f\|_{\Psi^1} := \sum_{n \in \mathbb{Z}} |[f]_n| \in [0, +\infty].$$

Let $\Psi^1$ denote the subset of $\mathcal{C}^0$ containing all the functions $f$ with $\|f\|_{\Psi^1} < +\infty$. These functions exactly coincide with their Chebyshev series in the following sense:

**Lemma 1.** *If $f \in \Psi^1$, then $\pi_n \cdot f$ converges absolutely and uniformly to $f$.*

*Proof.* Since, for all $i \in \mathbb{Z}$, $\|[f]_i T_i\|_\infty \leqslant |[f]_i|$ and $\sum_{i \in \mathbb{Z}} |[f]_i| = \|f\|_{\Psi^1} < \infty$ by definition of $f \in \Psi^1$, $\pi_n \cdot f = \sum_{|i| \leqslant n} [f]_i T_i$ converges absolutely and uniformly (and therefore also in $L^2$) to a continuous function, which is necessarily $f$ from Fejer's theorem [26, §I.3.1]. □

Note that $\Psi^1$ is analogous to the Wiener algebra $A(\mathbb{T})$ of absolutely convergent Fourier series [26, §I.6]: for $f \in \Psi^1$, we have $\|f\|_{\Psi^1} = \|f(\cos)\|_{A(\mathbb{T})}$. More precisely we have:

**Lemma 2.** *$(\Psi^1, \|\cdot\|_{\Psi^1})$ is a Banach algebra, which means that it is a Banach space satisfying*

$$\|fg\|_{\Psi^1} \leqslant \|f\|_{\Psi^1}\|g\|_{\Psi^1} \quad \textit{for all } f, g \in \Psi^1. \tag{7}$$

*Proof.* It is identical to the proofs from [26, §I.6]. □

It follows from Lemma 1 and Theorem 2 that $\Psi^1$ is included in $\mathcal{C}^0$ and contains the set of Lipschitz functions over $[-1, 1]$. Actually, the inclusions are strict, see [54, §VIII.1] and [54, §VI.3] respectively.

Moreover, the uniform and $\Psi^1$ norms can be partially ordered:

$$\|g\|_\infty \leqslant \sum_{n\in\mathbb{Z}} \|[g]_n T_n\|_\infty \leqslant \sum_{n\in\mathbb{Z}} |[g]_n| = \|g\|_{\Psi^1} \quad \text{for all } g \in \Psi^1.$$

Conversely, we have from (6):

$$|[g]_n| \leqslant \|g\|_\infty \quad \text{for all } g \in \Psi^1 \text{ and } n \in \mathbb{Z}.$$

However, since $g$ has in general an infinite number of non-zero coefficients, this fact cannot be used directly to bound $\|g\|_{\Psi^1}$ by the uniform norm of $g$.

We now consider the action of a bounded linear operator $\mathbf{F} : \Psi^1 \to \Psi^1$. By definition, its operator norm is $\|\mathbf{F}\|_{\Psi^1} := \sup_{\|f\|_{\Psi^1}\leqslant 1} \|\mathbf{F} \cdot f\|_{\Psi^1}$. Such operators include multiplication by $g \in \Psi^1$ or integration (indefinite or from a specific point).

**Proposition 1.** *Let $f = \sum_{n\in\mathbb{Z}} a_n T_n \in \Psi^1$. For indefinite integration operator $\int$ and respectively definite integration $\int_{t_0}^t$ from specific $t_0 \in [-1, 1]$, with $t \in [-1, 1]$, one has:*

$$\int f = \sum_{n\neq 0} \frac{a_{n-1} - a_{n+1}}{2n} T_n,$$

$$\int_{t_0}^t f \mathrm{d}t = \sum_{n\neq 0} \frac{a_{n-1} - a_{n+1}}{2n} (T_n(t) - T_n(t_0)),$$

$$\| \int \|_{\Psi^1} = 1,$$

$$\| \int_{t_0}^t \|_{\Psi^1} \leqslant 2. \tag{8}$$

*Proof.* The first two equations follow from (4). Whence, we deduce $\| \int f\|_{\Psi^1} \leqslant \|f\|_{\Psi^1}$, and so $\| \int \|_{\Psi^1} \leqslant 1$. Equality is attained for $\int T_0 = (T_{-1} + T_1)/2 = T_1$. For definite integration the operator bound is tight for $t_0 = -1$ since $\| \int_{-1}^t T_0 \mathrm{d}t\|_{\Psi^1} = \|T_1 + T_0\|_{\Psi^1} = 2$, but not for $t_0 = 0$, where $\| \int_0^t \|_{\Psi^1} = 1$. □

In general, computing the $\Psi^1$-norm of an operator $\mathbf{F}$ reduces to evaluating $\mathbf{F}$ at all the polynomials $T_i$ for $i \in \mathbb{Z}$:

**Lemma 3.** *For a bounded linear operator* $\mathbf{F} : \mathrm{Ч}^1 \to \mathrm{Ч}^1$*, its* $\mathrm{Ч}^1$*-operator norm is given by:*

$$\|\mathbf{F}\|_{\mathrm{Ч}^1} = \sup_{i \geqslant 0} \|\mathbf{F} \cdot T_i\|_{\mathrm{Ч}^1}.$$

*Proof.* Take $f = \sum_{n\in\mathbb{Z}} a_n T_n \in \mathrm{Ч}^1$. We have:

$$\begin{aligned}\|\mathbf{F} \cdot f\|_{\mathrm{Ч}^1} = \|\sum_{n\in\mathbb{Z}} a_n \mathbf{F} \cdot T_n\|_{\mathrm{Ч}^1} &\leqslant \sum_{n\in\mathbb{Z}} |a_n| \|\mathbf{F} \cdot T_n\|_{\mathrm{Ч}^1} \\ &\leqslant \left(\sum_{n\in\mathbb{Z}} |a_n|\right) \sup_{i\in\mathbb{Z}} \|\mathbf{F} \cdot T_i\|_{\mathrm{Ч}^1} = \|f\|_{\mathrm{Ч}^1} \sup_{i\geqslant 0} \|\mathbf{F} \cdot T_i\|_{\mathrm{Ч}^1}\end{aligned}$$

which shows that $\|\mathbf{F}\|_{\mathrm{Ч}^1} \leqslant \sup_{i\geqslant 0} \|\mathbf{F} \cdot T_i\|_{\mathrm{Ч}^1}$. The converse inequality is clearly true since the family of the $T_i$ is a subset of $\{f \in \mathrm{Ч}^1 \mid \|f\|_{\mathrm{Ч}^1} \leqslant 1\}$. □

### 2.2.1 Matrix representation

It is sometimes convenient to study a bounded linear operator $\mathbf{F} : \mathrm{Ч}^1 \to \mathrm{Ч}^1$ using its *matrix representation* [20], which is usually constructed based on the action of $\mathbf{F}$ on a basis of $\mathrm{Ч}^1$. In our case, the one-sided Chebyshev family $(T_n)_{n\in\mathbb{N}}$ is a Schauder basis for the space $\mathrm{Ч}^1$.

**Proposition 2.** *A bounded linear operator* $\mathbf{F} : \mathrm{Ч}^1 \to \mathrm{Ч}^1$ *can be uniquely described by the matrix* $(F_{ij})_{i,j\in\mathbb{N}}$*, where* $F_{ij}$ *is the* $i$*-th coefficient of the one-sided (see also Remark 2.1) Chebyshev series of* $\mathbf{F} \cdot T_j$*. Specifically, using* (6)*,* $F_{ij} = 2[F \cdot T_j]_i$ *for* $i > 0$ *and* $F_{0j} = [F \cdot T_j]_0$*. Moreover,*

$$\|\mathbf{F}\|_{\mathrm{Ч}^1} = \sup_{j\in\mathbb{N}} \sum_{i\in\mathbb{N}} |F_{ij}| := ||F||_1. \tag{9}$$

*Proof.* In [20], the one-to-one correspondence between bounded linear operators and infinite matrices is given. Equation (9) follows from Lemma 3, remarking that for fixed $j$, $\sum_{i\in\mathbb{N}} |F_{ij}| = \|\mathbf{F} \cdot T_j\|_{\mathrm{Ч}^1}$. □

It is important to remark that for assessing the action of $\mathbf{F}$, it is sometimes more convenient to use *two-sided infinite matrices*, which do not necessarily correspond to symmetric Chebyshev series. For completeness, this is formally defined in the sequel.

**Definition 1.** *Let a bounded linear operator* $\mathbf{F} : \mathrm{Ч}^1 \to \mathrm{Ч}^1$*. A matrix* $\hat{F} = (\hat{F}_{ij})_{i,j\in\mathbb{Z}}$ *is a representation of the operator* $\mathbf{F}$ *if for all* $j \in \mathbb{Z}$*,* $\mathbf{F} \cdot T_j = \sum_{i\in\mathbb{Z}} \hat{F}_{ij} T_i$*.*

In general, this representation is not unique but the following necessary condition holds for $(\hat{F}_{ij})_{i,j\in\mathbb{Z}}$:

$$[\mathbf{F} \cdot T_j]_i = \frac{\hat{F}_{ij} + \hat{F}_{(-i)j}}{2} = [\mathbf{F} \cdot T_{-j}]_i = \frac{\hat{F}_{i(-j)} + \hat{F}_{(-i)(-j)}}{2}, \tag{10}$$

which implies unicity when the series $\sum_{i\in\mathbb{Z}} \hat{F}_{ij} T_i$ is symmetric.

However, relaxing the symmetry requirement makes it possible to obtain numerically interesting sparse matrix forms as described in Section 3.2.1 for instance. Note that for computing $\|\mathbf{F}\|_{\mathrm{Ч}^1}$, one readily has:

$$\|\mathbf{F}\|_{\mathrm{Ч}^1} = \sup_{j\geqslant 0} \|\mathbf{F} \cdot T_j\|_{\mathrm{Ч}^1} = \sup_{j\geqslant 0} \left( |F_{0j}| + \sum_{i>0} |\hat{F}_{ij} + \hat{F}_{-ij}| \right). \tag{11}$$

## 2.3 Definition of Chebyshev models and elementary operations

Now we define Chebyshev models for the $\|\cdot\|_{\mathrm{Ч}^1}$ norm in order to provide a rigorous tool for computations in function spaces. This is slightly different from [8], where the uniform norm is considered.

**Definition 2.** *A Chebyshev model* $\boldsymbol{f}$ *of degree* $n$*, for a function* $f$ *in* $\mathrm{Ч}^1$*, is a pair* $(\boldsymbol{P}, \varepsilon)$ *where* $\boldsymbol{P} = \boldsymbol{c}_0 T_0 + \boldsymbol{c}_1 T_1 + \cdots + \boldsymbol{c}_n T_n$ *is a polynomial given in the Chebyshev basis with* interval *coefficients and* $\varepsilon \geqslant 0$ *is a floating-point error bound, such that*

$$\exists P = c_0 T_0 + \cdots + c_n T_n, \quad (\forall i \in [\![0, n]\!],\, c_i \in \boldsymbol{c}_i) \text{ and } \|f - P\|_{\mathrm{Ч}^1} \leqslant \varepsilon.$$

Similarly to [8], basic operations like addition, subtraction and multiplication by a scalar are easily defined for Chebyshev models. The corresponding operation is performed on the underlying polynomials and the error bound is trivially deduced. In the case of addition and subtraction for instance, the error bound of the result is equal to the rounded-up sum of the error bounds of the two operands. The complexity in the uniform model is linear with respect to the degrees of the operands.

For multiplication, let $\boldsymbol{f} = (\boldsymbol{P}, \varepsilon)$ and $\boldsymbol{g} = (\boldsymbol{Q}, \eta)$ two Chebyshev models. Then $\boldsymbol{f} \cdot \boldsymbol{g} = (\boldsymbol{R}, \delta)$ with $\boldsymbol{R} = \boldsymbol{P} \cdot \boldsymbol{Q}$ and $\delta = \|\boldsymbol{Q}\|_{\mathrm{Ч}^1}\eta + \|\boldsymbol{P}\|_{\mathrm{Ч}^1}\varepsilon + \varepsilon\eta$, using Inequality (7). Complexity is mainly determined by the multiplication of two polynomials expressed in Chebyshev basis with interval coefficients. Although numerical fast multiplication algorithms exist for polynomials in Chebyshev basis [3, 19], their interval arithmetics translations fail as of today to produce accurate results when the degrees become large. This is why we keep on with the traditional quadratic time multiplication algorithm.

Using Formula (8), we can easily define the integration of a Chebyshev model $\boldsymbol{f} = (\boldsymbol{P}, \varepsilon)$ from $t_0 \in [-1, 1]$ by $\int_{t_0}^{t} \boldsymbol{f} = (\boldsymbol{R}, \delta)$ with $\boldsymbol{R}(t) = \int_{t_0}^{t} \boldsymbol{P}(s)\mathrm{d}s$ and $\delta = 2\varepsilon$.

Concerning multiplication and integration, one notices that the degree of the resulting Chebyshev model $\boldsymbol{h} = (\boldsymbol{R}, \delta)$ model exceeds the one of the operand(s). As a consequence, if we want to fix a maximal degree $n$ for the Chebyshev models, then the polynomial part $\boldsymbol{R}$ of the resulting Chebyshev model $\boldsymbol{h}$ must be truncated at degree $n$ and the error corresponding to the discarded part must be added to the total error bound. One gets $\overline{\boldsymbol{h}} = (\mathrm{п}_n \cdot \boldsymbol{R}, \delta + \|(\mathbf{I} - \mathrm{п}_n) \cdot \boldsymbol{R}\|_{\mathrm{Ч}^1})$, whose degree does not exceed $n$.

Other operations like division or square root cannot be defined in such an algebraic way. The method we implemented is to first compute a polynomial approximation in Chebyshev basis and then obtain a rigorous error bound by means of a fixed-point method.

Note also that solving Problem 1.1 can be seen as obtaining a Chebyshev model solution of a LODE: if its coefficients are polynomials or functions in $\mathrm{Ч}^1$ rigorously approximated by Chebyshev models, then the procedure returns a Chebyshev model for its exact mathematical solution.

# 3 Integral operator and its truncations

From Formula (5), we observe that the action of the derivation operator on the Chebyshev coefficients is represented by a dense upper triangular matrix. It implies that a direct translation of the differential equation (1) into a linear problem produces a dense infinite-dimensional system of linear equations. Moreover it is ill-conditioned in the general case [22]. Hence, numerical algorithms to solve (1) using this method are neither efficient nor accurate. From the validation point of view, since the derivation is not an endomorphism of $\mathcal{T}^1$ (some functions in $\mathcal{T}^1$ are *not* even differentiable), designing a topological fixed-point method directly from Equation (1) seems rather tedious.

One way to circumvent these limitations consists in transforming the differential equation (1) into an integral one. The indefinite integration operator has far better properties: first, it is an endomorphism of $\mathcal{T}^1$. Second, it has a sparse matrix representation in $\mathcal{T}^1$, cf. (4), and its conditioning is significantly better than that of the differential one [22]. Thus, one can expect more efficient and accurate numerical algorithms in this case. The following standard, but crucial proposition (see [28] or [51, Chap. 2] for a proof) establishes this transformation, which was already used in purely numerical works for LODEs (for example in [11]) as well as for validation purposes [4].

**Proposition 3.** *Let $f$ be a function of class $\mathcal{C}^r$ over $[-1,1]$. Then $f$ is a solution of the linear IVP problem* (1a) *if and only if $\varphi = f^{(r)} \in \mathcal{C}^0$ is solution of the Volterra integral equation:*

$$\varphi + \mathbf{K}\cdot\varphi = \psi \quad \textit{with} \quad (\mathbf{K}\cdot\varphi)(t) = \int_{t_0}^{t} k(t,s)\varphi(s)\mathrm{d}s,\ t\in[-1,1], \tag{12}$$

*where:*

- *the kernel $k(t,s)$ is a bivariate continuous function given by:*

$$k(t,s) = \sum_{j=0}^{r-1} \alpha_j(t)\frac{(t-s)^{r-1-j}}{(r-1-j)!},\ (t,s)\in[-1,1]^2,$$

- *the right hand side $\psi$ is given by:*

$$\psi(t) = \gamma(t) - \sum_{j=0}^{r-1}\alpha_j(t)\sum_{k=0}^{r-1-j} v_{j+k}\frac{(t-t_0)^k}{k!},\ t\in[-1,1].$$

*By a slight abuse of terminology, we shall call $r$ the order of the integral operator* $\mathbf{K}$.

**Remark 3.1.** *Proposition 3 can be applied to the polynomial case of Problem 1.2 by replacing $\alpha_j$ and $\gamma$ with polynomials $a_j$ and $g$. It produces an equivalent integral equation with a bivariate* polynomial *kernel $k(t,s)$ and a* polynomial *right hand side $\psi(t)$. This will be of first importance in Section 3.2 where we deal with the polynomial case.*

**Remark 3.2.** *Henceforth, as noted in equation* (12)*, the new unknown function is $\varphi = f^{(r)}$. Although a similar integral formulation for the unknown $f$ is possible, this choice allows for the validation in Section 5 of numerical solutions for*

*both $f$ and its derivatives $f^{(i)}$, $i = 1, \dots, r$, which is often required in validated dynamics cf. Example 7.4.*

Solving Equation (12) with numerical algorithms on computers relies most of the time [20, 21] on a reduction of this infinite-dimensional problem to a finite-dimensional one. In fact, usually, one approximately computes several coefficients of the Chebyshev expansion of the exact solution. This is often done based on approximations of the inverse operator. The question of which functional space the solution $\varphi$ belongs to is of major importance both for the numerical approximation and the computation of the validated uniform error bound. In what follows we first recall the classical action of $\mathbf{K}$ on $(\mathcal{C}^0([-1,1]), \|\cdot\|_\infty)$, with a focus on Picard iteration. Then, in Section 3.2, we prove analogous properties in the $(\text{Ч}^1, \|\cdot\|_{\text{Ч}^1})$ space, based on operator iterations and truncations. This Banach space proves to be the natural framework to deal with Chebyshev coefficients without losing the link with the norm $\|\cdot\|_\infty$ (since $\|\cdot\|_\infty \leqslant \|\cdot\|_{\text{Ч}^1}$).

## 3.1 Inverse of $\mathbf{I} + \mathbf{K}$ in $(\mathcal{C}^0([-1,1]), \|\cdot\|_\infty)$

It is classical that in this Banach space the operators $\mathbf{K}$ and $\mathbf{I}+\mathbf{K}$ are bounded linear endomorphisms. For $n \in \mathbb{N}$, the operator $\mathbf{K}^n$ is a bounded linear operator with operator norm

$$\|\mathbf{K}^n\|_\infty \leqslant \frac{(2C)^n}{n!}, \text{ where } C := \sup_{-1\leqslant s,t\leqslant 1} |k(t,s)| < \infty. \tag{13}$$

Picard iteration [28, 40] is a standard way to prove the invertibility of $\mathbf{I}+\mathbf{K}$ in $(\mathcal{C}^0([-1,1]), \|\cdot\|_\infty)$ and give an explicit form for $(\mathbf{I}+\mathbf{K})^{-1}$, by its Neumann series:

$$(\mathbf{I}+\mathbf{K})^{-1} = \sum_{n\in\mathbb{N}} (-1)^n \mathbf{K}^n = \mathbf{I} - \mathbf{K} + \mathbf{K}^2 + \cdots + (-1)^n \mathbf{K}^n + \cdots .$$

This yields an explicit approximation process for the solution of (12):

$$\varphi_0 = \psi, \qquad \varphi_{n+1} = \psi - \mathbf{K}\cdot\varphi_n = \left(\sum_{k=0}^{n} (-1)^k \mathbf{K}^k\right)\cdot\psi,\ n \in \mathbb{N}.$$

Iterating the integral operator $\mathbf{K}$ can also be used for validation purposes, as presented for example in [4]. However, in our quasi-Newton validation context, the Banach space $(\mathcal{C}^0, \|\cdot\|_\infty)$ seems difficult to work with when considering multiplication, integration and truncation of Chebyshev series as operations on the coefficients.

## 3.2 Inverse of $\mathbf{I} + \mathbf{K}$ in $(\text{Ч}^1, \|\cdot\|_{\text{Ч}^1})$

In this section, we provide a concrete description of the action of the integral operator $\mathbf{K}$ on the Chebyshev coefficients of a function.

**Remark 3.3.** *Henceforth and until the end of Section 3, we exclusively consider the polynomial case given by* (3)*. The results presented below could to some extent be generalized to the non-polynomial case (where all functions belong to* $\text{Ч}^1$*), but this would require a more complicated two-variable approximation theory without being essential to the validation procedure of the general problem 1a, presented in Section 6.*

Under this assumption, the kernel $k(t,s)$ is polynomial and hence we can decompose it in the Chebyshev basis according to the variable $s$:

$$k(t,s) = \sum_{j=0}^{r-1} b_j(t) T_j(s), \tag{14}$$

with $b_0, \ldots, b_{r-1}$ polynomials written in the Chebyshev basis. Such an elementary procedure is described in Algorithm 1. To implement it in a rigorous framework, one can use interval arithmetics or even rational arithmetics when the coefficients of the $a_j(t)$ are rationals.

**Algorithm 1** Computation of the $b_j(t)$ defining the kernel $k(t,s)$

**Input:** The order $r$ and the polynomials $a_j(t)$ $(j = 0, \ldots, r-1)$ written in the Chebyshev basis.

**Output:** The polynomials $b_j(t)$ $(j = 0, \ldots, r-1)$ defining the kernel $k(t,s)$ as in (14).

▷ *Expand* $(t-s)^k = \sum_{\ell=0}^{k} \xi_{k\ell}(t) T_\ell(s)$ *for* $k \in [\![0, r-1]\!]$.

$\xi_{00}(t) \leftarrow 1$
**for** $k = 1$ **to** $r-1$ **do**
  **for** $\ell = 0$ **to** $k$ **do** $\xi_{k\ell}(t) = 0$ **end for**
  **for** $\ell = 0$ **to** $k-1$ **do**
    $\xi_{k\ell}(t) \leftarrow \xi_{k\ell}(t) + t\xi_{k-1,\ell}(t)$
    $\xi_{k,\ell+1}(t) \leftarrow \xi_{k,\ell+1}(t) - \xi_{k-1,\ell}(t)/2$
    $\xi_{k,|\ell-1|}(t) \leftarrow \xi_{k,|\ell-1|}(t) - \xi_{k-1,\ell}(t)/2$
  **end for**
**end for**

▷ *Compute the* $b_j(t)$.

**for** $j = 0$ **to** $r-1$ **do**
  $b_j(t) \leftarrow 0$
  **for** $k = 0$ **to** $r-1$ **do**
    $b_j(t) \leftarrow b_j(t) + a_k(t)\xi_{r-1-k,j}(t)/(r-1-k)!$
  **end for**
**end for**

If $\varphi \in \mathrm{Y}^1$, then $\mathbf{K} \cdot \varphi$ is in $\mathrm{Y}^1$ since

$$\|\mathbf{K} \cdot \varphi\|_{\mathrm{Y}^1} = \|\sum_{j=0}^{r-1} b_j(t) \int_{t_0}^{t} T_j \varphi \mathrm{d}s\|_{\mathrm{Y}^1} \leqslant 2B\|\varphi\|_{\mathrm{Y}^1},$$

$$\text{with } B = \sum_{j=0}^{r-1} \|b_j\|_{\mathrm{Y}^1} \geqslant C.$$

where we used (7), (8) and $C$ was defined in Equation (13). This shows that $\mathbf{K}$, and hence $\mathbf{I} + \mathbf{K}$, are bounded linear endomorphisms of $\mathrm{Y}^1$. However, we do not have for the moment any information about the invertibility of $\mathbf{I} + \mathbf{K}$ in $\mathrm{Y}^1$. So far, its injectivity in $\mathrm{Y}^1$ is established, because this operator was an isomorphism (hence injective) over the superspace $\mathcal{C}^0([-1,1]))$.

### 3.2.1 Matrix representation of $\mathbf{I}+\mathbf{K}$ in $\mathrm{Y}^1$

According to Definition 1, let us establish a convenient two-sided matrix representation of $\mathbf{K}$. For that, we consider the polynomials $b_j$ of degree $d_j$, in the symmetric Chebyshev basis:

$$b_j = \sum_{-d_j \leqslant k \leqslant d_j} b_{j,k} T_k,\ b_{j,k} \in \mathbb{R},\ 0 \leqslant j < r.$$

For $i, j \in \mathbb{Z}$, we have $T_j T_i = (T_{i+j} + T_{i-j})/2$. Now, for $t \in [-1, 1]$, we have

$$\int_{t_0}^{t} T_j(s) T_i(s) \mathrm{d}s = \gamma_{iji}(t) - \gamma_{iji}(t_0),$$

with

$$\begin{aligned}\gamma_{ijk}(t) = &-\frac{1}{4(i-j-1)} T_{k-j-1}(t) + \frac{1}{4(i-j+1)} T_{k-j+1}(t) \\ &- \frac{1}{4(i+j-1)} T_{k+j-1}(t) + \frac{1}{4(i+j+1)} T_{k+j+1}(t), \quad (15)\end{aligned}$$

where, following the convention in Section 2, the terms for which the denominator vanishes are 0.

In particular, for $t_0 = -1$ and using $T_k(-1) = (-1)^k$ one obtains:

$$\gamma_{ijk}(-1) = -(-1)^{k+j} \left( \frac{j+1}{2(i^2 - (j+1)^2)} + \frac{j-1}{2(i^2 - (j-1)^2)} \right). \quad (16)$$

Let $j \in [\![0, r-1]\!]$, multiplying by $b_j$, we get, for $t \in [-1, 1]$,

$$b_j(t) \int_{t_0}^{t} T_j(s) T_i(s) \mathrm{d}s = -\gamma_{iji}(t_0) \sum_{-d_j \leqslant k \leqslant d_j} b_{j,k} T_k(t) + \sum_{-d_j \leqslant k \leqslant d_j} b_{j,k} \gamma_{ij(i+k)}(t). \quad (17)$$

This expression shows that there exists a two-sided matrix representation, say $(\hat{B}_{j,ki})_{k,i \in \mathbb{Z}}$, of the operator $\varphi \to b_j(t) \int_{t_0}^{t} T_j(s) \varphi(s) \mathrm{d}s$ which is sparse and has a so-called almost-banded structure. More precisely, it is made of a central horizontal band of non-zero coefficients $\hat{B}_{j,ki}$, with $-d_j \leqslant k \leqslant d_j$, $i \in \mathbb{Z}$, which we call initial coefficients together with a diagonal band of non-zero coefficients $\hat{B}_{j,ki}$, with $i \in \mathbb{Z}$ and $i - j - 1 - d_j \leqslant k \leqslant i + j + 1 + d_j$, which we call diagonal coefficients. A graphic view of this structure is shown in Figure 1.

**Remark 3.4.** *Note that this matrix representation does not ensure symmetry of the series $\sum_{k \in \mathbb{Z}} \hat{B}_{j,ki} T_k$ for any $i \in \mathbb{Z}$. As explained in Section 2.2.1, this relaxation allows for a structure which is interesting for numerical solving. The action of the operator in terms of symmetric Chebyshev series, as well as its norm, can be easily recovered with Formulas* (10) *and* (11).

The following definition formally establishes the notion of almost-banded matrix, in the finite as well as in the (one or two-sided) infinite case. It is robust in the sense that if a two-sided infinite matrix representing an endomorphism of $\mathrm{Y}^1$ is $(h, d)$-almost-banded, then so is its unique one-sided representation, and so are all its finite-dimensional truncations (defined in Subsection 3.3).

**Definition 3.** *Let $\mathcal{I}$ be a set of indices (typically $\mathbb{N}$, $\mathbb{Z}$ or $[\![0, n-1]\!]$ for some $n > 0$), and $h$, $d$ two nonnegative integers.*

1. *For $i \in \mathcal{I}$, $v \in \mathbb{R}^{\mathcal{I}}$ is said to be $(h, d)$-almost-banded around index $i$ if for all $j \in \mathcal{I}$, $v_j = 0$ whenever $|j| > h$ and $|i - j| > d$.*

2. *The order $n$ square matrix $A = (a_{ij})_{i,j\in\mathcal{I}} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ is said to be $(h, d)$-almost-banded if for all $j \in \mathcal{I}$, the $j$-th column $v^{(j)} = (a_{ij})_{i\in\mathcal{I}} \in \mathbb{R}^{\mathcal{I}}$ of $A$ is almost-banded around index $j$.*

It turns out that a two-sided matrix representation of $\mathbf{K}$ has an almost-banded structure: to obtain $\mathbf{K} \cdot T_i$, it suffices to sum all the contributions from Equation (17) for $0 \leqslant j < r$. Hence $\mathbf{K} \cdot T_i$ is $(h, d)$-almost-banded around index $i$, which shows that the integral operator $\mathbf{K}$ has an $(h, d)$-almost-banded matrix representation, where:

$$h = \max_{0\leqslant j<r} d_j, \qquad d = \max_{0\leqslant j<r} j + 1 + d_j.$$

The width of the horizontal band is $2h + 1$ centered around 0 and that of the diagonal band is $2d + 1$, as shown in Figure 1. With a slight terminology abuse, such operators are directly called *almost-banded operators* in what follows.

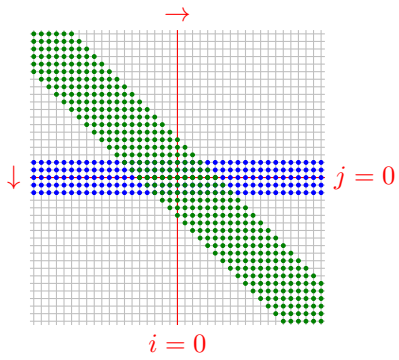



Figure 1: Almost-banded structure of operator $\mathbf{K}$.

### 3.2.2 Iterations of $\mathbf{K}$ in $\mathcal{H}^1$ and almost-banded approximations of $(\mathbf{I} + \mathbf{K})^{-1}$

We recalled in Subsection 3.1 the convergence of the Neumann series $\mathbf{I} - \mathbf{K} + \mathbf{K}^2 - \dots$ to $(\mathbf{I} + \mathbf{K})^{-1}$ in $(\mathcal{C}^0, \|\cdot\|_\infty)$. The following lemma establishes an analogous result in $\mathcal{H}^1$:

**Lemma 4.** *The operator $\mathbf{I} + \mathbf{K}$ is invertible in $\mathcal{H}^1$ and its inverse is given by the Neumann series $\sum_{i\geqslant 0}(-\mathbf{K})^i$ which converges in $o(\varepsilon^n)$ for all $\varepsilon > 0$. More precisely:*

- *$\sum_{i=0}^{n}(-1)^i\mathbf{K}^i$ is a sequence of $(dn, dn)$-almost-banded operators;*
- *$\|\sum_{i>n}(-1)^i\mathbf{K}^i\|_{\mathcal{H}^1} \leqslant \sum_{i>n}(6di + 1)\frac{(2C)^i}{i!}$ ($C$ defined in Equation (13)).*

*Proof.* In $\mathrm{Ч}^1$, since $\mathbf{K}$ is $(h,d)$-almost-banded, with $h < d$, a straightforward induction shows that $\mathbf{K}^n$ is $(h_n, d_n)$-almost-banded, with $d_n = nd$ and $h_n < d_n$.

Fix an index $j \in \mathbb{Z}$. Then the *symmetric* Chebyshev series of $\mathbf{K}^n \cdot T_j$ has at most $(2d_n+1)+(2h_n+1)+(2d_n+1) \leqslant 6nd+1$ non-zero coefficients. Moreover, for each index $k$ corresponding to a non-zero coefficient, we have, from (2.2) and (13), $|[\mathbf{K}^n \cdot T_j]_k| \leqslant \|\mathbf{K}^n \cdot T_j\|_\infty \leqslant (2C)^n/n!$. Hence, we get:

$$\|\mathbf{K}^n \cdot T_j\|_{\mathrm{Ч}^1} \leqslant (6dn+1)\frac{(2C)^n}{n!},$$

from which we conclude using Lemma 3. □

This shows that obtaining an approximate solution of (12) via iterations of $\mathbf{K}$, is possible both in $(\mathcal{C}^0([-1,1]), \|\cdot\|_\infty)$ and in $(\mathrm{Ч}^1, \|\cdot\|_{\mathrm{Ч}^1})$. However, the action of $\mathbf{K}$ and its iterates involves handling an infinite dimensional space. In the sequel, we prove that suitable *truncations* of $\mathbf{K}$ allow for obtaining approximate solutions in finite dimensional subspaces of $\mathrm{Ч}^1$ and these solutions converge in $o(\varepsilon^n)$ for all $\varepsilon > 0$ to the exact solution of (12).

### 3.3 Approximate solutions via truncations $\mathbf{K}^{[n]}$ of $\mathbf{K}$

The $n$-th truncation (also called the $n$-th section in [20]) of the integral operator $\mathbf{K}$ is defined as follows:

$$\mathbf{K}^{[n]} = \mathrm{п}_n \cdot \mathbf{K} \cdot \mathrm{п}_n. \tag{18}$$

The truncation method (also called projection method in [20]) to solve Equation (12) consists in replacing $\mathbf{K}$ by $\mathbf{K}^{[n]}$ and solving the finite-dimensional linear problem:

$$\varphi + \mathbf{K}^{[n]} \cdot \varphi = \psi. \tag{19}$$

Note that a representation matrix $M$ of $\mathbf{K}^{[n]}$ can be trivially obtained by extracting the square matrix $(\hat{K}_{ij})_{-n\leqslant i,j\leqslant n}$ from the infinite representation matrix $(\hat{K}_{ij})_{i,j\in\mathbb{Z}}$ of $\mathbf{K}$. Obviously, $M$ has an $(h,d)$-almost-banded structure. This implies that solving Equation (19) reduces to solving a linear system of equations with a specific almost-banded structure. We revisit in Section 4 efficient algorithms for solving such systems.

Moreover, we prove the following important fast convergence result:

**Theorem 4.** *Let $\varphi^* := (\mathbf{I}+\mathbf{K})^{-1}\cdot\psi$ the exact solution of integral equation* (12) *and $\widetilde{\varphi}_n := (\mathbf{I}+\mathbf{K}^{[n]})^{-1}\cdot\psi$ the solution of the truncated system* (19)*. We have:*

$$\|\varphi^* - \widetilde{\varphi}_n\|_{\mathrm{Ч}^1} = o(\varepsilon^n) \quad \textit{for all } \varepsilon > 0.$$

In [4, Thm. 4.4] and [38, Thm. 4.5], similar convergence rates were proven in the different context of the uniform norm and for rather different approximations schemes: either the considered operator is different (the differential operator is handled in [38]) or the employed tools are more involved (*main asymptotic existence theorem* for linear recurrences is needed in [4]). The proof of Theorem 4 requires important theoretical properties concerning the truncated operator $\mathbf{K}^{[n]}$ in relation with $\mathbf{K}$ in the space $\mathrm{Ч}^1$, which are given in the next two additional lemmas. They are also of first importance for the validation method developed in Section 5.

Firstly, let us prove that $\mathbf{K}^{[n]}$ is a good approximation of $\mathbf{K}$ in the $\mathrm{Ч}^1$ sense.

**Lemma 5.** *Let* $\mathbf{K}$ *be the integral operator in* (12)*, of order* $r$ *and polynomial coefficients* $b_j$*. Let* $(h, d)$ *be the parameters of its almost-banded structure and* $n \geqslant r + d$ *be the truncation order, then:*

*(i)* $\mathbf{K}^{[n]} \cdot T_i = \mathbf{K} \cdot T_i$ *for all* $i$ *such that* $|i| \leqslant n - d$*.*

*(ii)* $\mathbf{K}^{[n]} \to \mathbf{K}$ *in* $\Upsilon^1$ *as* $n \to \infty$*. More precisely:*

$$\|\mathbf{K} - \mathbf{K}^{[n]}\|_{\Upsilon^1} \leqslant B \max\left(\frac{1}{n+1-r-d}, \frac{2}{n-r}\right) \quad \text{with } B = \sum_{j=0}^{r-1} \|b_j\|_{\Upsilon^1},$$

*which implies a convergence speed of* $O(1/n)$ *as* $n \to \infty$*.*

*Proof.* For *(i)*, if $|i| \leqslant n-d$, then $\mathbf{K} \cdot T_i$ is of degree at most $\max(h, n-d+d) = n$ because $n \geqslant d \geqslant h$. Hence:

$$\mathbf{K}^{[n]} \cdot T_i := \pi_n \cdot \mathbf{K} \cdot \pi_n \cdot T_i = \pi_n \cdot \mathbf{K} \cdot T_i = \mathbf{K} \cdot T_i.$$

For *(ii)*, note first that from Lemma 3 and *(i)*, one has:

$$\|\mathbf{K} - \mathbf{K}^{[n]}\|_{\Upsilon^1} = \sup_{i \geqslant 0} \|\mathbf{K} \cdot T_i - \mathbf{K}^{[n]} \cdot T_i\|_{\Upsilon^1} = \sup_{i > n-d} \|\mathbf{K} \cdot T_i - \mathbf{K}^{[n]} \cdot T_i\|_{\Upsilon^1}.$$

Now, for $\varphi \in \Upsilon^1$ one has the following decomposition:

$$(\mathbf{K} - \mathbf{K}^{[n]})\varphi = \mathbf{K} \cdot \varphi - \pi_n \cdot \mathbf{K} \cdot \pi_n \cdot \varphi = \mathbf{K} \cdot (\mathbf{I} - \pi_n) \cdot \varphi + (\mathbf{I} - \pi_n) \cdot \mathbf{K} \cdot \pi_n \cdot \varphi.$$

Whence, one can evaluate $\mathbf{K} - \mathbf{K}^{[n]}$ on all remaining $T_i$'s for $i > n - d$:

- If $n - d < i \leqslant n$, then $(\mathbf{K} - \mathbf{K}^{[n]}) \cdot T_i = (\mathbf{I} - \pi_n) \cdot \mathbf{K} \cdot T_i$. Note that, since $n \geqslant h$, only the diagonal coefficients of $\mathbf{K} \cdot T_i$ may bring a nonzero contribution. Moreover, we have $|i \pm j \pm 1| \geqslant n + 1 - d - r$. From that we deduce an upper bound of the approximation error:

$$\|(\mathbf{I} - \pi_n) \cdot \mathbf{K} \cdot T_i\|_{\Upsilon^1} \leqslant \frac{B}{n+1-r-d}.$$

- If $i > n$, then $(\mathbf{K} - \mathbf{K}^{[n]}) \cdot T_i = \mathbf{K} \cdot T_i$. We have that $|i \pm j \pm 1| \geqslant n - r$ for $0 \leqslant j < r$. Hence:

$$\|\mathbf{K} \cdot T_i\|_{\Upsilon^1} \leqslant \frac{2B}{n-r}$$

We conclude by taking the maximum of these two bounds. □

The convergence of $\mathbf{K}^{[n]}$ to $\mathbf{K}$ also implies that $\mathbf{I} + \mathbf{K}^{[n]}$ is invertible for $n$ large enough:

**Lemma 6.** *For* $n$ *large enough, we have:*

*(i) the endomorphism* $\mathbf{I} + \mathbf{K}^{[n]}$ *is invertible.*

*(ii)* $(\mathbf{I} + \mathbf{K}^{[n]})^{-1}$ *converges to* $(\mathbf{I} + \mathbf{K})^{-1}$*, with:*

$$\begin{aligned}\|(\mathbf{I} + \mathbf{K}^{[n]})^{-1} - (\mathbf{I} + \mathbf{K})^{-1}\|_{\Upsilon^1} &\leqslant \frac{\|(\mathbf{I} + \mathbf{K})^{-1}\|_{\Upsilon^1}^2}{1 - \|(\mathbf{I} + \mathbf{K})^{-1} \cdot (\mathbf{K} - \mathbf{K}^{[n]})\|_{\Upsilon^1}} \|\mathbf{K} - \mathbf{K}^{[n]}\|_{\Upsilon^1} \\ &= O\left(\frac{1}{n}\right) \quad \text{as } n \to \infty\end{aligned}$$

*(iii)* $(\mathbf{I}+\mathbf{K}^{[n]})^{-1} = \sum_{i\geqslant 0}(-\mathbf{K}^{[n]})^i$.

*Proof.* For $(i)$ and $(ii)$, using the bound in $O(1/n)$ for $\|\mathbf{K}-\mathbf{K}^{[n]}\|_{\mathcal{Y}^1}$ obtained in Lemma 5, the invertibility of $\mathbf{I}+\mathbf{K}^{[n]}$ as well as the announced explicit upper bound for $\|(\mathbf{I}+\mathbf{K}^{[n]})^{-1}-(\mathbf{I}+\mathbf{K})^{-1}\|_{\mathcal{Y}^1}$ directly follow from [20, Chap. 2, Cor. 8.2].

For $(iii)$, since by Lemma 4 the Neumann series of $\mathbf{K}$ absolutely converges, there is a $p>0$ such that $\|\mathbf{K}^p\|_{\mathcal{Y}^1}<1$. Since $\mathbf{K}^{[n]}\to\mathbf{K}$ as $n\to\infty$, there is an $n$ such that $\|(\mathbf{K}^{[n]})^p\|_{\mathcal{Y}^1}<1$. Therefore, the Neumann series of $(\mathbf{K}^{[n]})^p$ is absolutely convergent, and the following factorization establishes the absolute convergence of the Neumann series of $\mathbf{K}^{[n]}$:

$$\sum_{i\geqslant 0}(-\mathbf{K}^{[n]})^i = \left(\sum_{i<p}(-\mathbf{K}^{[n]})^i\right)\cdot\left(\sum_{i\geqslant 0}(-\mathbf{K}^{[n]})^{pi}\right)$$

□

Note that from the previous lemma, one readily obtains that $\widetilde{\varphi}_n := (\mathbf{I}+\mathbf{K}^{[n]})^{-1}\cdot\psi$ converges to the exact solution $\varphi := (\mathbf{I}+\mathbf{K})^{-1}\cdot\psi$ in $O(1/n)$. However, we can now prove the far better convergence result of the main Theorem 4.

*Proof of Theorem 4.* Take $n>d$ large enough so that $\mathbf{I}+\mathbf{K}^{[n]}$ is invertible by Lemma 6. Let $\overline{\varphi}_n = (\sum_{i\leqslant\lfloor n/2d\rfloor}(-1)^i\mathbf{K}^i)\cdot\psi$ denote the approximate solution obtained by computing the Neumann series of $\mathbf{K}$ at order $\lfloor n/2d\rfloor$. Since this series is an $(d\lfloor n/2d\rfloor, d\lfloor n/2d\rfloor)$-almost-banded operator, we get that $\overline{\varphi}_n$ is a polynomial of degree at most $\deg(\psi)+d\lfloor n/2d\rfloor \leqslant \deg(\psi)+n/2$. Hence, for $n$ large enough, the degree of $\overline{\varphi}_n$ does not exceed $n-d$, so that we have the key equality $\mathbf{K}^{[n]}\cdot\overline{\varphi}_n = \mathbf{K}\cdot\overline{\varphi}_n$, according to Lemma 5 *(i)*. From that we deduce:

$$\varphi^* - \widetilde{\varphi}_n = \left(\mathbf{I}-\left(\mathbf{I}+\mathbf{K}^{[n]}\right)^{-1}(\mathbf{I}+\mathbf{K})\right)\cdot(\varphi^*-\overline{\varphi}_n)\,.$$

From Lemma 6 *(ii)* and Lemma 4, we finally get:

$$\|\varphi^*-\widetilde{\varphi}_n\|_{\mathcal{Y}^1} = O\left(\frac{(2C)^{\lfloor n/2d\rfloor}}{\lfloor n/2d\rfloor!}\right),$$

which is an $o(\varepsilon^n)$ for all $\varepsilon>0$. □

For completeness, we note the following alternative proof of Lemma 4. The convergence of the finite-dimensional truncations $\mathbf{K}^{[n]}$ to $\mathbf{K}$ in $\mathcal{Y}^1$ implies that $\mathbf{K}$ is a compact endomorphism of the Banach space $\mathcal{Y}^1$. The Fredholm alternative [7] says in that case that $\mathbf{I}+\mathbf{K}:\mathcal{Y}^1\to\mathcal{Y}^1$ is injective if and only if it is surjective. But, as mentioned at the beginning of Subsection 3.2, we already have the injectivity of this operator. Hence, we conclude that $\mathbf{I}+\mathbf{K}$ is bijective, and moreover that it is a bicontinuous isomorphism of $\mathcal{Y}^1$ (using the Banach continuous inverse theorem).

We discuss in the next section algorithms concerning almost-banded matrices, since this structure is essential both for efficient algorithmic computation of $\widetilde{\varphi}$ and its *a posteriori* validation step.

# 4 Algorithms involving almost-banded matrices

Let $A$ and $B$ be two order $n$ square matrices, respectively $(h_A, d_A)$ and $(h_B, d_B)$-almost-banded. In Table 1 we recall several elementary operations which are straightforward, the result is an almost-banded matrix, and their complexity is in $O(n)$ provided that the almost-banded parameters are supposed constant with respect to $n$.

Note that the product $A \cdot B$ is computable in $O(n(h_A + d_A)(h_B + d_B))$ operations by applying the evaluation on a sparse vector defined at line 5 in the table on all the columns of $B$. In the dense case, when $h_A + d_A \approx n$ and $h_B + d_B \approx n$, this corresponds to the naive $O(n^3)$ algorithm, hence a fast multiplication algorithm becomes more appropriate.

| Operation | Result's a.-b. structure | Complexity |
|---|---|---|
| $\lambda A$, with $\lambda \in \mathbb{R}$ | $(h_A, d_A)$ | $O(n(h_A + d_A))$ |
| $A + B$ or $A - B$ | $(\max(h_A, h_B), \max(d_A, d_B))$ | $O(n(\max(h_A, h_B) + \max(d_A, d_B)))$ |
| $A \cdot v$ dense $v \in \mathbb{R}^n$ | dense | $O(n(h_A + d_A))$ |
| $A \cdot v$ $(h_v, d_v)$ a.-b.$v \in \mathbb{R}^n$ | $(\max(h_v + d_A, h_A), d_v + d_A)$ | $O((h_A + d_A)(h_v + d_v))$ |
| $A \cdot B$ | $(\max(h_B + d_A, h_A), d_B + d_A)$ | $O(n(h_A + d_A)(h_B + d_B))$ |
| $\|A\|_1$ | - | $O(n(h_A + d_A))$ |

Table 1: Elementary operations on almost-banded (a.-b.) matrices or vectors: $A$ and $B$ are order $n$ square matrices, respectively $(h_A, d_A)$ and $(h_B, d_B)$-almost-banded, and $v \in \mathbb{R}^n$ is either dense or almost-banded (a.-b.) around some index $i \in [\![0, n-1]\!]$.

We now turn to efficient algorithms for solving almost-banded linear systems as well as matrix inversion. In Subsection 4.1, we recall Olver and Townsend's algorithm for solving order $n$ almost-banded linear systems in linear complexity with respect to $n$. This directly leads to a quadratic algorithm for inverting an almost-banded matrix. To achieve linear complexity for inversion, we give in Subsection 4.2 a modified version of this algorithm.

## 4.1 A reminder on Olver and Townsend's algorithm for almost-banded linear systems

Let $M$ denote an $(h, d)$-almost-banded order $n$ square matrix with $h \leqslant d$, and $y \in \mathbb{R}^n$. The goal is to solve the almost-banded linear system $M \cdot x = y$ for unknown $x \in \mathbb{R}^n$. The procedure is split into two parts. First, a QR decomposition $Q \cdot M = R$ is computed, with $Q$ orthogonal and $R$ upper triangular. Then, the equivalent system $R \cdot x = Q \cdot y$ is solved by back-substitution. The key challenge is to maintain a linear complexity with respect to $n$ in both steps.

### 4.1.1 First step: QR decomposition

This is computed in Algorithm 2 using Givens rotations' method which eliminates line after line the coefficients of $M$ under the diagonal to finally obtain $R$,

as shown in Figure 2.

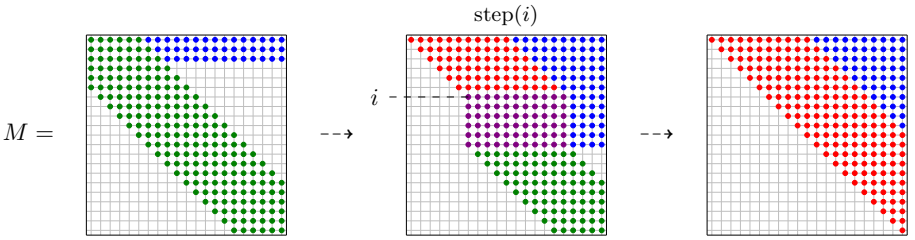



Figure 2: Step 1 of Olver and Townsend's algorithm

More precisely, at step $i$, for each $j \in [\![i+1, \min(i+d, n-1)]\!]$, we apply a well-chosen rotation $\begin{pmatrix} c_{ij} & -s_{ij} \\ s_{ij} & c_{ij} \end{pmatrix}$ on lines $i$ and $j$ in order to get $R_{ji} = 0$. Note that at the end of each step $i$, $R_{ii} \neq 0$ if and only if the matrix $M$ is invertible.

The direct application of this process would cause the progressive filling-in of the rows, which would give a dense upper triangular matrix $R$. In fact, this phenomenon can be controlled by noticing that for each $i < n - 2d - 1$, the "end of the row" $i$ of $R$, $\begin{pmatrix} R_{i,i+2d+1} & \dots & R_{i,n-1} \end{pmatrix}$, is a linear combination of the corresponding dense part of $M$: $\begin{pmatrix} M_{\ell,i+2d+1} & \dots & M_{\ell,n-1} \end{pmatrix}$ for $\ell \in [\![0, h]\!]$. Hence, it suffices to manipulate instead the coefficients $\lambda_{i\ell}$ of the linear combination:

$$\begin{pmatrix} R_{i,i+2d+1} & \dots & R_{i,n-1} \end{pmatrix} = \sum_{\ell=0}^{h} \lambda_{i\ell} \begin{pmatrix} M_{\ell,i+2d+1} & \dots & M_{\ell,n-1} \end{pmatrix}. \tag{20}$$

Based on this observation, Algorithm 2 returns the QR decomposition $Q \cdot M = R$ under the following representation:

- $Q$ is completely determined by $c_{ij}, s_{ij}$:

$$Q = \prod_{i=0}^{n-1} \prod_{j=i+1}^{\min(i+d,n-1)} Q^{(ij)},$$

  where the $Q^{(ij)}$ are rotation matrices defined by:

$$(Q^{(ij)})_{k\ell} = \begin{cases} 1 & \text{if } k = \ell \text{ and } k \neq i, j, \\ c_{ij} & \text{if } k = \ell = i \text{ or } k = \ell = j, \\ s_{ij} & \text{if } k = j \text{ and } \ell = i, \\ -s_{ij} & \text{if } k = i \text{ and } \ell = j, \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

- $R$ is upper triangular and represented by its $2d+1$ upper diagonals (entries $R_{ij}$ for $i \in [\![0, n-1]\!]$ and $j \in [\![i, \min(i+2d, n-1)]\!]$ are given explicitly) together with the coefficients $\lambda_{i\ell}$ ($i \in [\![0, n-1]\!]$ and $\ell \in [\![0, h]\!]$) defining the rest of $R$ as in (20).

**Algorithm 2** Step 1 of Olver and Townsend's algorithm

**Input:** An $(h, d)$-almost-banded order $n$ matrix $M$ with $h \leqslant d$.

**Output:** A QR factorization $Q \cdot M = R$: $Q$ defined by $c_{ij}, s_{ij}$ as in (21); $R$ defined by $R_{ij}$ ($i \in [\![0, n-1]\!], j \in [\![i, \min(i+2d, n-1)]\!]$) and $\lambda_{i\ell}$ as in (20).

$R \leftarrow M$
**for** $i = 0$ **to** $n-1$ **do for** $j = 0$ **to** $h$ **do** $\lambda_{ij} \leftarrow 0$ **end for end for**
**for** $i = 0$ **to** $h$ **do** $\lambda_{ii} \leftarrow 1$ **end for**
**for** $i = 0$ **to** $n-1$ **do**
  **for** $j = i+1$ **to** $\min(i+d, n-1)$ **do**
    **if** $R_{ji} = 0$ **then**
      $c_{ij} \leftarrow 1$ **and** $s_{ij} \leftarrow 0$
    **else**
      $r \leftarrow \sqrt{R_{ii}^2 + R_{ij}^2}$
      $c_{ij} \leftarrow R_{ii}/r$ **and** $s_{ij} \leftarrow -R_{ji}/r$
      **for** $k = i$ **to** $\min(i+2d, n-1)$ **do** $\begin{pmatrix} R_{ik} \\ R_{jk} \end{pmatrix} \leftarrow \begin{pmatrix} c_{ij} & -s_{ij} \\ s_{ij} & c_{ij} \end{pmatrix} \cdot \begin{pmatrix} R_{ik} \\ R_{jk} \end{pmatrix}$
      **for** $\ell = 0$ **to** $h$ **do** $\begin{pmatrix} \lambda_{i\ell} \\ \lambda_{j\ell} \end{pmatrix} \leftarrow \begin{pmatrix} c_{ij} & -s_{ij} \\ s_{ij} & c_{ij} \end{pmatrix} \cdot \begin{pmatrix} \lambda_{i\ell} \\ \lambda_{j\ell} \end{pmatrix}$
    **end if**
  **end for**
**end for**

Formally, one has:

**Proposition 4.** *Algorithm 2 applied on an $(h, d)$-almost-banded matrix of order $n$ with $h \leqslant d$ is correct and runs in $O(nd^2)$ operations.*

*Proof.* Given in [38]. □

#### 4.1.2 Second step: back-substitution

Once step 1 is performed and returns $Q \cdot M = R$, we first apply the rotations $Q^{(ij)}$ on the right hand side $y \in \mathbb{R}^n$ to obtain $Q \cdot y$ in $O(nd)$ operations. Now we have to solve $R \cdot x = Q \cdot y := y_Q$.

If $R$ is regarded as a dense upper triangular matrix, the classical back-substitution algorithm requires $O(n^2)$ operations. However, based on the sparse representation of $R$, the back-substitution in Algorithm 3 is more efficient. Its main idea to compute the solution $x_i$ (for $i$ going backwards from $n-1$ to 0) is to use Equation (20) for expressing $R_{ij}$ as soon as $i < n-2d-1$, $j > i$:

$$x_i = \left( y_{Qi} - \sum_{j=i+1}^{n-1} R_{ij} x_j \right) / R_{ii} = \left( y_{Qi} - \sum_{j=i+1}^{i+2d} R_{ij} x_j - \sum_{\ell=0}^{h} \lambda_{i\ell} z_{i\ell} \right) / R_{ii},$$

where

$$z_{i\ell} = \begin{pmatrix} M_{\ell,i+2d+1} & \dots & M_{\ell,n-1} \end{pmatrix} \cdot \begin{pmatrix} x_{i+2d+1} & \dots & x_{n-1} \end{pmatrix}^T = \sum_{j=i+2d+1}^{n-1} M_{\ell j} x_j.$$

Then, once $z_{i\ell}$ is computed, $z_{i-1,\ell}$ is updated in constant time:

$$z_{i-1,\ell} = M_{\ell,i+2d} x_{i+2d} + z_{i\ell}.$$

This leads to the following proposition:

**Proposition 5.** *Algorithm 3 is correct and requires $O(nd)$ operations.*

*Proof.* Given in [38]. □

**Algorithm 3** Step 2 of Olver and Townsend's algorithm

**Input:** An invertible $(h,d)$-almost-banded order $n$ matrix $M$ with $h \leqslant d$, its QR decomposition produced by Algorithm 2 and a vector $y \in \mathbb{R}^n$.
**Output:** The solution vector $x$ of $M \cdot x = y$.

▷ *Compute $Q \cdot y$*
1: **for** $i = 0$ **to** $n-1$ **do**
2: **for** $j = i+1$ **to** $\min(i+d, n-1)$ **do**
3: $\begin{pmatrix} y_i \\ y_j \end{pmatrix} \leftarrow \begin{pmatrix} c_{ij} & -s_{ij} \\ s_{ij} & c_{ij} \end{pmatrix} \cdot \begin{pmatrix} y_i \\ y_j \end{pmatrix}$
4: **end for**
5: **end for**
▷ *Back-substitution*
6: **for** $\ell = 0$ **to** $h$ **do** $z_\ell \leftarrow 0$ **end for**
7: **for** $i = n-1$ **down to** $0$ **do**
▷▷ *Update $z_\ell$*
8: **if** $i + 2d + 1 < n$ **then**
9: **for** $\ell = 0$ **to** $h$ **do** $z_\ell \leftarrow z_\ell + M_{\ell,i+2d+1} x_{i+2d+1}$ **end for**
10: **end if**
▷▷ *Compute $x_i$*
11: $x_i \leftarrow \left( y_i - \sum_{j=i+1}^{\min(i+2d,n-1)} R_{ij} x_j - \sum_{\ell=0}^{h} \lambda_{i\ell} z_\ell \right) / R_{ii}$
12: **end for**

## 4.2 An algorithm for almost-banded approximation of inverse of almost-banded matrix

Based on Olver and Townsend's algorithm, the inverse of an $(h,d)$-almost-banded order $n$ matrix $M$ (with $h \leqslant d$) can be computed in quadratic time $O(n^2 d)$. First, step 1 is performed in $O(nd^2)$ operations (Proposition 4) to obtain a QR decomposition $Q \cdot M = R$. Then, each column $v^{(i)}$ of index $i \in [\![0, n-1]\!]$ of $M^{-1}$ is computed by solving $M \cdot v^{(i)} = e^{(i)}$, where $e^{(i)}$ denotes the $i$-th vector of the canonical basis of $\mathbb{R}^n$. This is achieved by using $n$ times step 2, resulting in a total of $O(n^2 d)$ operations.

Unfortunately, this algorithm has quadratic running time and returns a dense inverse matrix representation. In some cases however, such as the validation process developed in Section 5, a *sparse approximation* of $M^{-1}$ is sufficient. As proved in Lemma 6 (*iii*), the inverse of $M = \mathbf{I} + \mathbf{K}^{[n]}$ is approximable by almost-banded matrices. This leads to adapting the full inversion procedure described above to compute only coefficients on diagonal and horizontal bands.

Let $A \simeq M^{-1}$ be the required approximate inverse with an almost-banded structure given by the parameters $(h', d')$ (we do not require $h' \leqslant d'$). Firstly,

one computes the QR decomposition $Q \cdot M = R$ in $O(nd^2)$ operations. Then, Step 2 of Olver and Townsend's algorithm is modified, as detailed in Algorithm 4. For each $i \in [\![0, n-1]\!]$, the $i$-th column $v^{(i)}$ of $A$ is computed as an approximate solution of $R \cdot x = Q \cdot e^{(i)}$, in the form of an $(h', d')$-almost-banded vector around index $i$:

1. $Q \cdot e^{(j)} \in \mathbb{R}^n$ is computed only partially, between entries $i - d$ and $i + d'$. Note that in general $Q \cdot e^{(j)}$ has zero entries between indices 0 and $i-d-1$, and is dense from $i - d$ to $n - 1$.

2. The back-substitution only computes entries of the solution from indices $i + d'$ to $i - d'$, and from $h'$ to 0. Since the other entries are implicitly set to 0, these computed coefficients are only *approximations* of the entries at the same position in the exact solution. But considering that the neglected entries were small enough, this approximation is expected to be convenient.

**Algorithm 4** Almost-banded approximate column inversion

**Input:** An $(h, d)$-almost-banded order $n$ matrix $M$ with the QR decomposition $Q \cdot M = R$ produced by Algorithm 2 and parameters $h'$, $d'$, $i$ with $h' \in [\![h, n-1]\!]$, $d' \in [\![d, n-1]\!]$ and $i \in [\![0, n-1]\!]$.

**Output:** $(h', d')$-almost-banded vector $x$ around index $i$ such that $M \cdot x \approx e^{(i)}$.

$\mathfrak{D} \leftarrow [\![i - d', i + d']\!] \cap [\![0, n-1]\!]$ **and** $\mathfrak{H} \leftarrow [\![0, h']\!] - \mathfrak{D}$

▷ *Compute diagonal coefficients of* $Q \cdot y$

**for** $j$ **in** $\mathfrak{D} \cup [\![i + d' + 1, i + d' + d]\!] - \{i\}$ **do** $y_j \leftarrow 0$ **end for**

$y_i \leftarrow 1$

**for** $j$ **in** $\mathfrak{D}$ going upwards **do**

  **for** $k$ **in** $[\![j + 1, j + d]\!] \cap [\![0, n-1]\!]$ going upwards **do**

    $\begin{pmatrix} y_j \\ y_k \end{pmatrix} \leftarrow \begin{pmatrix} c_{jk} & -s_{jk} \\ s_{jk} & c_{jk} \end{pmatrix} \cdot \begin{pmatrix} y_j \\ y_k \end{pmatrix}$

  **end for**

**end for**

▷ *Partial back-substitution*

**for** $\ell \in [\![0, h]\!]$ **do** $z_\ell \leftarrow 0$ **end for**

$j_z \leftarrow n - 1$

**for** $j$ **in** $\mathfrak{D} \cup \mathfrak{H}$ going downwards **do**

  ▷▷ *Update* $z_\ell$

  **if** $j + 2d < j_z$ **then**

    **for** $\ell \in [\![0, h]\!]$ **do** $z_\ell \leftarrow z_\ell + \sum_{k \in [\![j+2d+1, j_z]\!] \cap (\mathfrak{D} \cup \mathfrak{H})} M_{\ell k} x_k$ **end for**

    $j_z \leftarrow j + 2d$

  **end if**

  ▷▷ *Compute* $x_j$

  **if** $j \in \mathfrak{D}$ **then** $c \leftarrow y_j$ **else** $c \leftarrow 0$

  $x_j \leftarrow \left( c - \sum_{k \in [\![j+1, j+2d]\!] \cap (\mathfrak{D} \cup \mathfrak{H})} R_{jk} x_k - \sum_{\ell=0}^{h} \lambda_{j\ell} z_\ell \right) / R_{jj}$

**end for**

**Algorithm 5** Almost-banded approximate inverse

**Input:** An $(h, d)$-almost-banded order $n$ matrix $M$ with the QR decomposition $Q \cdot M = R$ produced by Algorithm 2 and parameters $h'$, $d'$ with $h' \in [\![h, n-1]\!]$ and $d' \in [\![d, n-1]\!]$.

**Output:** $(h', d')$-almost-banded matrix $A$ with $A \approx M^{-1}$.

```
for i = 0 to n − 1 do
  for j = 0 to n − 1 do V[j] ← 0 end for
  V[i] ← 1
  Compute W ≈ M^{-1} · V using Algorithm 4
  Set i-th column of A to W
end for
```

We provide a complexity analysis of Algorithm 4, but nothing is stated concerning the accuracy of the obtained approximation. This procedure should really be seen as a heuristic in general.

**Proposition 6.** *Algorithm 4 involves $O((h + d)(h' + d'))$ operations.*

*Proof.* The first step (computing the diagonal coefficients of $Q \cdot y$) clearly requires $O(dd')$ arithmetic operations. Now consider the second step (the partial back-substitution) and enter the main loop at line 11, where index $j$ lives in a set of size $O(h' + d')$. First, we need to update the values $z_\ell$. At first sight, each $z_\ell$ seems to involve a sum of $O(h' + d')$ terms. But in fact, the total amortized cost related to line 13 is $O((h' + d')h)$, since at the end of the algorithm, each $z_\ell$ is equal to $\sum_{k \in [\![2d+1, n-1]\!] \cap (\mathfrak{D} \cup \mathfrak{H})} M_{\ell k} x_k$, which is a sum of $O(h' + d')$ terms. As a matter of fact, $j_z \leqslant j + 2d + 1$ most of the time, except when $h' < i - d'$ and the current index $j$ falls from $i - d'$ to $h'$. After that, the computation of $x_j$ involves two sums with a total of $O(h + d)$ terms. We therefore obtain the claimed complexity. □

**Corollary 1.** *Algorithm 5 produces an $(h', d')$-almost-banded approximation of the inverse of an $(h, d)$-almost-banded order $n$ matrix $M$ in $O(n(h+d)(h'+d'))$ operations.*

We now turn to the *a posteriori* validation step.

# 5 A quasi-Newton validation method

Given an approximate solution $\widetilde{\varphi}$ of the integral equation (12), we propose an *a posteriori* validation method which computes a rigorous upper bound for the approximation error $\|\varphi^* - \widetilde{\varphi}\|_{\mathrm{Y}^1}$, where $\varphi^*$ denotes the exact solution of (12). This is based on the general quasi-Newton framework explained in Section 1.2. In this case, $\mathbf{F} \cdot \varphi := \varphi + \mathbf{K} \cdot \varphi - \psi$ is affine, with linear part $\mathbf{I} + \mathbf{K}$. The quasi-Newton method requires an approximate inverse operator $\mathbf{A} \approx (\mathbf{I} + \mathbf{K})^{-1}$ such that $\|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I} + \mathbf{K})\|_{\mathrm{Y}^1} < 1$. Of course, computing an exact inverse would solve the problem but is out of reach. Instead of that, from Lemma 6, we know that for $n$ large enough, $(\mathbf{I} + \mathbf{K}^{[n]})^{-1}$ exists and is a good approximation of $(\mathbf{I} + \mathbf{K})^{-1}$. Since $(\mathbf{I} + \mathbf{K}^{[n]})^{-1}$ is defined by an $(n + 1)$-order square matrix (its restriction over $\pi_n \cdot \mathrm{Y}^1$) extended over the whole space $\mathrm{Y}^1$ by the identity, we

define the operator $\mathbf{A}$ over $\mathcal{Y}^1$ as an $(n+1)$-order square matrix $A$ approximating $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ over $\pi_n \cdot \mathcal{Y}^1$, extended by the identity over the whole space:

$$\mathbf{A} \cdot \varphi = A \cdot \pi_n \cdot \varphi + (\mathbf{I} - \pi_n) \cdot \varphi.$$

The first technical issue is to numerically compute (or represent) both very accurately and efficiently such a matrix $A$. Specifically, we aim both for a linear complexity computation with respect to $n$ and for minimizing $\|I_{n+1} - A \cdot M\|_1$, where $M$ is an order $n+1$ matrix representation for $\mathbf{I}+\mathbf{K}^{[n]}$. Among several possibilities to achieve these two requirements, we found none optimal for both. Therefore, we propose two solutions:

**S1.** As seen in Section 4, Olver and Tonwsend's Algorithm 3 can be used to numerically compute $M^{-1}$. The main advantage is that the approximation error $\|I_{n+1} - A \cdot M\|_1$ is really close to 0 using standard precision in the underlying computations. Drawback is the quadratic complexity in $O(n^2 d)$.

**S2.** Our new heuristic approach is based on Lemma 6 (*iii*) which states that $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ is well approximated by almost-banded matrices. So it is natural to look for a matrix $A$ with a $(h', d')$-almost-banded structure. Given $h'$ and $d'$, Algorithm 4, detailed in Section 4, produces an $(h', d')$-almost-banded approximation $A$ of $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ in $O(n(h'+d')(h+d))$ arithmetic operations (Proposition 1). If the parameters $(h', d')$ of the almost-banded structure of $A$ can be chosen small enough compared to $n$, this alternative method should be substituted to the standard one.

Deciding which of these two methods should be used in practice is non-trivial: while the second one is more appealing due to the resulting sparsity of $A$, unfortunately nothing is said about the order of magnitude of $n$ such that the conclusion of Lemma 6 (*iii*) is valid, nor about the precise speed of convergence of the Neumann series of $M$, which would give a good intuition for the values of $h'$ and $d'$ to choose. In what follows, the complexity analysis is thus provided for both cases: a sparse vs. a dense structure of the matrix $A$. This will allow us to discuss in detail the choice of these parameters in Section 5.2.1.

Next, one has to provide a rigorous Lipschitz constant $\mu$ (required by Theorem 1) for the Newton-like operator. We have:

$$\|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I}+\mathbf{K})\|_{\mathcal{Y}^1} \leqslant \|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I}+\mathbf{K}^{[n]})\|_{\mathcal{Y}^1} + \|\mathbf{A} \cdot (\mathbf{K} - \mathbf{K}^{[n]})\|_{\mathcal{Y}^1}, \qquad (22)$$

which can be interpreted as:

- $\|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I}+\mathbf{K}^{[n]})\|_{\mathcal{Y}^1}$ is the approximation error because $A$ was (maybe) not the exact representation matrix of $\mathbf{I}+\mathbf{K}^{[n]}$.
- $\|\mathbf{A} \cdot (\mathbf{K} - \mathbf{K}^{[n]})\|_{\mathcal{Y}^1}$ is the truncation error because $\mathbf{K}^{[n]}$ is not exactly $\mathbf{K}$.

Section 5.1 focuses on the truncation error, which is tightly bounded by some rather technical inequalities, summarized in Algorithm 6. The more straightforward computation of the approximation error is directly included in Algorithm 7.

Once we have obtained a quasi-Newton operator $\mathbf{T}$ with a certified Lipschitz constant $\mu < 1$, the validation of a candidate solution $\widetilde{\varphi}$ is summarized in Subsection 5.2, together with its complexity analysis.

## 5.1 Bounding the truncation error

The truncation error is computed by providing an upper bound for $\sup_{i\geqslant 0} B(i)$ where $B(i) := \|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\cdot T_i\|_{\curlywedge^1}$. The indices $i$ are divided into four groups:
– For $i \in [\![0, n-d]\!]$, $\mathbf{K}^{[n]}\cdot T_i = \mathbf{K}\cdot T_i$ (Lemma 5) and hence $B(i) = 0$.
– For $i \in [\![n-d+1, n]\!]$, $\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\cdot T_i = (\mathbf{I}-\pi_n)\cdot\mathbf{K}\cdot T_i$ are explicitly computed.
– For $i \in [\![n+1, n+d]\!]$, $B(i) = \|\mathbf{A}\cdot\mathbf{K}\cdot T_i\|_{\curlywedge^1}$ and some of the diagonal coefficients of $\mathbf{K}\cdot T_i$ are of index less than $n$ and are therefore non-trivially affected by $\mathbf{A}$. We choose to explicitly compute all these $\mathbf{A}\cdot\mathbf{K}\cdot T_i$.
– For $i > n+d$, $(\mathbf{K}-\mathbf{K}^{[n]})\cdot T_i = \mathbf{K}\cdot T_i$ and the diagonal coefficients of $\mathbf{K}\cdot T_i$ are all located at indices strictly greater than $n$. We have $B(i) = B_I(i) + B_D(i)$ with:

- $B_D(i) = \|(\mathbf{I}-\pi_n)\cdot\mathbf{K}\cdot T_i\|_{\curlywedge^1}$ due to diagonal coefficients, which decrease in $O(1/i)$ from Equation (15).
- $B_I(i) = \|A\cdot\pi_n\cdot\mathbf{K}\cdot T_i\|_{\curlywedge^1}$ due to initial coefficients multiplied by $A$, which decrease in $O(1/i^2)$ from Equation (16).

The main difficulty is to bound $B(i)$ for $i > n+d$, since we deal with an infinite number of indices $i$. For that, a natural idea is to use the explicit expression (17), replace $i$ by the interval $[n+d+1, +\infty)$ and evaluate $\mathbf{A}\cdot\mathbf{K}\cdot T_i$ in interval arithmetics. Since these evaluations often lead to overestimations, one needs to choose a large value for $n$, such that the convergence in $O(1/n)$ is sufficiently small to compensate. Usually, the chosen $n$ is far larger than the one needed for $\mathbf{T}$ to be contracting.

A better solution consists in computing $\mathbf{A}\cdot\mathbf{K}\cdot T_{i_0}$ where $i_0 > n+d$ and bounding the difference between $B(i)$ and $B(i_0)$ for all the remaining indices $i \geqslant i_0$.

**Lemma 7.** *Let $i \geqslant i_0 > n+d$. Then*

*1) For the diagonal coefficients, we have*

$$B_D(i) \leqslant B_D(i_0) + \frac{r\sum_{j=0}^{r-1}\|b_j\|_{\curlywedge^1}}{(i_0-r)^2}.$$

*2) For the initial coefficients, we have*

$$B_I(i) \leqslant B_I(i_0) + \frac{r^3\sum_{j=0}^{r-1}\|\mathbf{A}\cdot b_j\|_{\curlywedge^1}}{(i_0^2-r^2)^2}.$$

*Proof.* For 1), from (17) we know that the diagonal coefficients of $\mathbf{K}\cdot T_i$, and respectively $\mathbf{K}\cdot T_{i_0}$, are those of the polynomials $\sum_{0\leqslant j<r}\sum_{-d_j\leqslant k\leqslant d_j} b_{jk}\gamma_{ij(i+k)}$, and respectively $\sum_{0\leqslant j<r}\sum_{-d_j\leqslant k\leqslant d_j} b_{jk}\gamma_{i_0j(i_0+k)}$. All these coefficients are of positive index, so that we can shift them by $i-i_0$ positions to the right by replacing $\gamma_{i_0j(i_0+k)}$ with $\gamma_{i_0j(i+k)}$ without changing the norm. This ruse allows

us to compare polynomials of equal degree i.e., $\gamma_{ij(i+k)}$ and $\gamma_{i_0j(i+k)}$:

$$|iB_D(i) - i_0B_D(i_0)| = \left| i \left\| \sum_{j=0}^{r-1} \sum_{k=-d_j}^{d_j} b_{jk}\gamma_{ij(i+k)} \right\|_{\mathrm{U}^1} - i_0 \left\| \sum_{j=0}^{r-1} \sum_{k=-d_j}^{d_j} b_{jk}\gamma_{i_0j(i+k)} \right\|_{\mathrm{U}^1} \right|$$
$$\leqslant \sum_{j=0}^{r-1} \sum_{k=-d_j}^{d_j} |b_{jk}| \| i\gamma_{ij(i+k)} - i_0\gamma_{i_0j(i+k)} \|_{\mathrm{U}^1}.$$

Using the fact that for all $x$ such that $|x| < i_0 \leqslant i$,

$$\left| \frac{i}{i+x} - \frac{i_0}{i_0+x} \right| \leqslant \frac{i_0}{(i_0-|x|)^2}|x|,$$

we get that for any $\ell$, $\|i\gamma_{ij\ell} - i_0\gamma_{i_0j\ell}\|_{\mathrm{U}^1} \leqslant ri_0/(i_0-r)^2$. We conclude by noticing that

$$B_D(i) \leqslant \frac{i}{i_0}B_D(i) \leqslant B_D(i_0) + \frac{1}{i_0}|iB_D(i) - i_0B_D(i_0)| \leqslant B_D(i_0) + \frac{r\sum_{j=0}^{r-1}\|b_j\|_{\mathrm{U}^1}}{(i_0-r)^2}.$$

For 2) we have that

$$|i^2B_I(i) - i_0^2B_I(i_0)| = \left| i^2 \left\| \mathbf{A} \cdot \sum_{j=0}^{r-1} \gamma_{iji}(-1)b_j \right\|_{\mathrm{U}^1} - i_0^2 \left\| \mathbf{A} \cdot \sum_{j=0}^{r-1} \gamma_{i_0ji}(-1)b_j \right\|_{\mathrm{U}^1} \right|$$
$$\leqslant \sum_{j=0}^{r-1} \|\mathbf{A} \cdot b_j\|_{\mathrm{U}^1} |i^2\gamma_{iji}(-1) - i_0^2\gamma_{i_0ji}(-1)|.$$

We conclude using (16) and a similar inequality:

$$\left| \frac{i^2}{i^2-x^2} - \frac{i_0^2}{i_0^2-x^2} \right| \leqslant \frac{i_0^2}{(i_0^2-x^2)^2}x^2.$$

□

In practice, this method yields more accurate bounds when the parameters of the problem become somehow large. This is due to the fact that the part potentially affected by overestimations is divided by greater power of $i_0$ ($i_0^2$ and $i_0^4$) than in the previously mentioned method ($i_0$ and $i_0^2$).

Note that the bounds announced by Lemma 7 can be sharpened if we don't replace $|j \pm 1|$ with $r$. The obtained formulas are essentially not more difficult to implement, but we omit these details for the sake of clarity.

**Proposition 7.** *Algorithm 6 is correct and requires $O((h'+d')(h+d)d)$ operations when $A$ is $(h',d')$-almost-banded, or $O(n(h+d)d)$ operations when $A$ is dense.*

*Proof.* The correctness is straightforward, using Lemma 7. To reach the claimed complexity, the polynomials $\mathbf{K} \cdot T_i$ involved in the algorithm must be sparsely computed as an $(h,d)$-almost-banded vector around index $i$, using $O(rh)$ arithmetic operations. Clearly, step 1 for $\delta_{trunc}^{(1)}$ (lines 1-5) costs $O(drh)$ operations. For each $i$ in step 2 for $\delta_{trunc}^{(2)}$ (lines 6-10), computing $\mathbf{K} \cdot T_i$ costs

**Algorithm 6** Bounding the truncation error

**Input:** A polynomial integral operator $\mathbf{K}$ (given by its order $r$ and the $b_j(t)$), a truncation order $n$ and an approximate inverse $A$ of $\mathbf{I}+\mathbf{K}^{[n]}$.

**Output:** An upper bound $\delta_{trunc}$ for the truncation error $\|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\|_{\curlywedge^1}$.

▷ ***All operations are to be performed in interval arithmetics***

▷ *Compute* $\delta^{(1)}_{trunc} \geqslant \sup_{i\in[\![n-d+1,n]\!]} B(i)$

$\delta^{(1)}_{trunc} \leftarrow 0$
**for** $i = n-d+1$ **to** $n$ **do**
  $P \leftarrow (\mathbf{I}-\pi_n)\cdot\mathbf{K}\cdot T_i$
  **if** $\|P\|_{\curlywedge^1} > \delta^{(1)}_{trunc}$ **then** $\delta^{(1)}_{trunc} \leftarrow \|P\|_{\curlywedge^1}$
**end for**

▷ *Compute* $\delta^{(2)}_{trunc} \geqslant \sup_{i\in[\![n+1,n+d]\!]} B(i)$

$\delta^{(2)}_{trunc} \leftarrow 0$
**for** $i = n+1$ **to** $n+d$ **do**
  $P \leftarrow \mathbf{A}\cdot\mathbf{K}\cdot T_i$
  **if** $\|P\|_{\curlywedge^1} > \delta^{(2)}_{trunc}$ **then** $\delta^{(2)}_{trunc} \leftarrow \|P\|_{\curlywedge^1}$
**end for**

▷ *Compute* $\delta^{(3)}_{trunc} \geqslant \sup_{i\geqslant n+d+1} B_D(i)$

$i_0 \leftarrow n+d+1$ **and** $B \leftarrow \sum_{j=0}^{r-1}\|b_j\|_{\curlywedge^1}$
$P \leftarrow (\mathbf{I}-\pi_n)\cdot\mathbf{K}\cdot T_{i_0}$ **and** $\delta^{(3)}_{trunc} \leftarrow \|P\|_{\curlywedge^1}$
$\delta^{(3)}_{trunc} \leftarrow \delta^{(3)}_{trunc} + \frac{rB}{(i_0-r)^2}$

▷ *Compute* $\delta^{(4)}_{trunc} \geqslant \sum_{i\geqslant n+d+1} B_I(i)$

$B \leftarrow \sum_{j=0}^{r-1}\|A\cdot b_j\|_{\curlywedge^1}$
$P \leftarrow A\cdot\pi_n\cdot\mathbf{K}\cdot T_{i_0}$ **and** $\delta^{(4)}_{trunc} \leftarrow \|P\|_{\curlywedge^1}$
$\delta^{(4)}_{trunc} \leftarrow \delta^{(4)}_{trunc} + \frac{r^3B}{(i_0^2-r^2)^2}$

$\delta_{trunc} \leftarrow \max(\delta^{(1)}_{trunc}, \delta^{(2)}_{trunc}, \delta^{(3)}_{trunc}+\delta^{(4)}_{trunc})$
**return** $\delta_{trunc}$

$O(rh)$ operations to obtain an $(h,d)$-almost-banded vector, and applying $\mathbf{A}$ costs $O((h'+d')(h+d))$ or $O(n(h+d))$ operations, depending on whether $A$ is $(h',d')$-almost-banded or dense (see Table 1 in Section 4). Hence we get $O((h'+d')(h+d)d)$ or $O(n(h+d)d)$ operations. After that, step 3 for $\delta^{(3)}_{trunc}$ (lines 11-13) costs $O(rh)$ operations both to compute $(\mathbf{I}-\Pi_n)\cdot\mathbf{K}\cdot T_{i_0}$ and $\sum_{j=0}^{r-1}\|b_j\|_{\mathcal{C}^1}$. Finally, at step 4 (lines 14-16), computing $\sum_{j=0}^{r-1}\|A\cdot b_j\|_{\mathcal{C}^1}$ costs $O((h'+d')rh)$ or $O(nrh)$ operations, and computing $A\cdot\Pi_n\cdot\mathbf{K}\cdot T_{i_0}$ costs $O(rh+(h'+d')(h+d))$ or $O(rh+n(h+d))$ operations. We see that in both cases ($A$ $(h',d')$-almost-banded or dense), the most expensive step is the second one, which gives the respective expected total complexities. □

## 5.2 Complete validation method and complexity

We now have all the ingredients for the complete validation process: Algorithm 7 obtains a contracting Newton-like operator $\mathbf{T}$ and Algorithm 8 validates a candidate solution $\widetilde{\varphi}$.

For Algorithm 7, the parameters $h$, $d$ and the $\|b_j\|_{\mathcal{C}^1}$ directly come from LODE (3), while the other input parameters $n$, $h'$ and $d'$ must either be known by the user or obtained from a decision procedure. For that, first, Proposition 8 analyses the complexity of Algorithm 7 (and Algorithm 8) when $n$, $h'$ and $d'$ are given. Then, a more detailed study of the magnitude of these parameters and an intuition on how to choose them is proposed.

### 5.2.1 Complexity in function of the chosen parameters

**Proposition 8.** *Let $\mathbf{K}$ be the integral operator associated to the polynomial LODE (3), $n$ be the truncation order chosen for the quasi-Newton method, $M=\mathbf{I}+\mathbf{K}^{[n]}$ and $(h,d)$ the parameters of its almost-banded structure, $A$ the approximation of $M^{-1}$ used for $\mathbf{T}$, either dense or $(h',d')$-almost-banded. We have the following complexity results:*

*(i) The complexity of producing the Newton-like operator $\mathbf{T}$ and validating its $\mathcal{C}^1$-norm using Algorithm 7 is:*

$$O(n(h+d)(h'+d')) \quad (\text{or } O(n^2(h+d)) \text{ when } A \text{ is dense}).$$

*(ii) Having this validated Newton-like operator, an approximate solution $\widetilde{\varphi}$ of (3) (with $p=\deg\widetilde{\varphi}$ and $q=\deg\psi$) is validated using Algorithm 8 in:*

$$O(prh+q+(h'+d')\min(n,\max(p+d,q))) \\ (\text{or } O(prh+q+n\min(n,\max(p+d,q))) \text{ when } A \text{ is dense}).$$

*Proof.* For $(i)$, we consider the different steps to obtain $\mathbf{T}$ and bound its $\mathcal{C}^1$-norm:

- Computing $M=\mathbf{I}+\mathbf{K}^{[n]}$ (line 1) costs $O(nrh)$ operations, using the defining formula (14) of $\mathbf{K}$, and $O(nd^2)$ operations are needed for the QR decomposition (line 2) according to Proposition 4.

**Algorithm 7** Creating and validating a Newton-like operator $\mathbf{T}$

**Input:** A polynomial integral operator $\mathbf{K}$ (given by its order $r$ and the $b_j(t)$), a truncation order $n$ and optional parameters $h'$ and $d'$.

**Output:** An approximate inverse $A$ of $\mathbf{I}+\mathbf{K}^{[n]}$ ($(h',d')$-almost-banded if $h'$ and $d'$ were specified, dense otherwise) and a certified Lipschitz constant $\mu$.

$M \leftarrow \mathbf{I} + \mathbf{K}^{[n]}$, computed as an $(h,d)$-almost-banded matrix.
Compute $(Q,R)$ for $M$ using Algorithm 2($M$)

▷ *Compute the approximate inverse $A$ of $M$*
**if** $h'$ and $d'$ are specified with $h \leqslant h' < n$ and $d \leqslant d' < n$ **then**
  ▷▷ *$A$ is $(h',d')$-almost-banded*
  Compute $A$ $(h',d')$-almost-banded using Algorithm 5
**else**
  ▷▷ *$A$ is dense*
  **for** $i = 0$ **to** $n-1$ **do**
    **for** $j = 0$ **to** $n-1$ **do** $V[j] \leftarrow 0$ **and** $V[i] \leftarrow 1$
    Numerically compute $W = M^{-1} \cdot V$ using Algorithm 3
    Set $i$-th column of $A$ to $W$ **end for**
  **end for**
**end if**

▷ *Compute the approximation error $\delta_{approx} \geqslant \|\mathbf{I} - \mathbf{A}\cdot(\mathbf{I}+\mathbf{K}^{[n]})\|_{\mathfrak{U}^1}$*
$\delta_{approx} \leftarrow 0$
**for** $i = 0$ **to** $n-1$ **do**
  Set $V$ to the $i$-th column of $M$, as an $(h,d)$-almost-banded vector
  Compute $W \leftarrow A \cdot V$ **and** $W[i] \leftarrow W[i] - 1$ with interval arithmetics
  **if** $\|W\|_1 > \delta_{approx}$ **then** $\delta_{approx} \leftarrow \|W\|_1$
**end for**
▷ *Compute the truncation error $\delta_{trunc} \geqslant \|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\|_{\mathfrak{U}^1}$*
Compute $\delta_{trunc}$ using Algorithm 6

$\mu \leftarrow \delta_{approx} + \delta_{trunc}$
**if** $\mu < 1$ **then**
  **return** $\mu$
**else**
  **print** "Fail, $\mu > 1$"
**end if**

**Algorithm 8** Validating a candidate solution of an integral equation

**Input:** A polynomial integral operator $\mathbf{K}$ (given by its order $r$ and the $b_j(t)$), a polynomial right hand side $\psi$, a truncation order $n$, $(A, \mu)$ obtained from Algorithm 7 with $\mu < 1$, and a candidate solution $\widetilde{\varphi}$.

**Output:** An error bound $\varepsilon$ such that $\|\widetilde{\varphi} - \varphi^*\|_{\mathrm{Q}^1} \leqslant \varepsilon$.

▷ ***All operations are to be performed in interval arithmetics***

1: $P \leftarrow \widetilde{\varphi} + \mathbf{K} \cdot \widetilde{\varphi} - \psi$
2: **for** $i = 0$ **to** $n$ **do** $V[i] \leftarrow [P]_i$
3: $W \leftarrow A \cdot V$
4: **for** $i = 0$ **to** $n$ **do** $[P]_i \leftarrow W[i]$
5: $\varepsilon \leftarrow \|P\|_{\mathrm{Q}^1}/(1-\mu)$
6: **return** $\varepsilon$

- Computing $A$ (lines 3-11) needs $O(n(h+d)(h'+d'))$ operations in the almost-banded case (Corollary 1), or $O(n^2(h+d))$ in the dense case.
- Using Table 1, line 15 costs $O((h'+d')(h+d))$ operations when $A$ is $(h', d')$-almost-banded, or $O(n(h+d))$ when it is dense. Hence the computation of the approximation error is performed in $O(n(h'+d')(h+d))$ (almost-banded case) or $O(n^2(h+d))$ (dense case) operations.
- The truncation error (line 18) costs $O((h'+d')(h+d)d)$ operations when $A$ is $(h', d')$-almost-banded, or $O(n(h+d)d)$ in the dense case, following Proposition 7.

Hence, the total complexity is in $O(n(h+d)(h'+d'))$ when $A$ is $(h', d')$-almost-banded, or $O(n^2(h+d))$ when $A$ is dense.

For $(ii)$, computing $P$ (of degree $\max(p+d, q)$) at line 1 costs $O(prh + q)$ operations. Multiplying by $A$ its $n+1$ first coefficients (line 3) requires $O((h'+d')\min(n, \max(p+d, q))$ operations (if $A$ $(h', d')$-almost-banded) or $O(n\min(n, \max(p+d, q)))$ operations (if $A$ dense). Note that at line 2, copying the $n+1$ first coefficients of $P$ costs $\min(\max(p+d, q), n)$ (neglect the null coefficients), and in the almost-banded case when $\max(p+d, q) < n$, lines 4 costs $(\max(p+d, q) + h' + d')$ operations (again, neglect the final null coefficients). This yields the claimed total complexity. □

### 5.2.2 Choosing and estimating parameters $n$, $h'$ and $d'$

The complexity claimed by Proposition 8 depends on the parameters $n$, $h'$ and $d'$. Hence, the performance of the validation method is directly linked to the minimal values we can choose for these parameters.

In practice, one initializes $n = 2d$ (to avoid troubles with too small values of $n$) and then estimates (from below) the norm $\|(\mathbf{I} + \mathbf{K}^{[n]})^{-1} \cdot (\mathbf{K} - \mathbf{K}^{[n]})\|_{\mathrm{Q}^1}$ by numerically applying this operator on $T_{n+1}$. This heuristic is similar to estimating the truncation error of a Chebyshev series by its first neglected term [9, §4.4, Thm. 6][1]. Specifically, for intermediate or large values of $n$, one has for $i \leqslant n$ that $\|(\mathbf{K} - \mathbf{K}^{[n]}) \cdot T_i\|_{\mathrm{Q}^1} \leqslant \|\mathbf{K} \cdot T_i\|_{\mathrm{Q}^1}$, and for $i \geqslant n+1$, one has a decrease of $\|\mathbf{K} \cdot T_i\|_{\mathrm{Q}^1}$ in $O(1/i)$. Recall that for $i \geqslant n$, $\mathbf{K}^{[n]} \cdot T_i = 0$, from (18). So,

[1] [6, §2.12] presents, as a rule-of-thumb, the estimate of the truncation error by the last term retained.

$\max_{i\geqslant 0} \|(\mathbf{I}+\mathbf{K}^{[n]})^{-1}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\cdot T_i\|_{\mathrm{Y}^1}$ is heuristically achieved for $i=n+1$. Concretely, computing $\|(\mathbf{I}+\mathbf{K}^{[n]})^{-1}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\cdot T_{n+1}\|_{\mathrm{Y}^1}$ reduces to numerically solving the corresponding almost-banded system with input parameters $M$ and $\mathrm{n}_n\cdot K\cdot T_{n+1}$ using Algorithms 2 and 3.

If this estimate from below of the norm of $\mathbf{T}$ is greater than 1, we double the value of $n$ until the estimated norm falls below 1. Then we initialize $h'=h$ and $d'=d$, compute an $(h',d')$ almost-banded approximation of $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ using Algorithm 5 and double their values each time the approximation error exceeds 0.25. After that, Algorithm 6 produces a certified upper bound for the truncation error. If it exceeds 0.25, then again we double the value of $n$ and restart the validation process.

In what follows, we give theoretical estimates for the order of magnitude of the above mentioned parameters. First a bound for $n$ is

$$n=O(dB^2\exp(2B)),\quad \text{where } B=\sum_{j=0}^{r-1}\|b_j\|_{\mathrm{Y}^1}.$$

This can be proved since $n$ must be chosen large enough so that the sum of the approximation and truncation errors falls below 1. For this, a sufficient condition is $\|(\mathbf{I}+\mathbf{K})^{-1}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\|_{\mathrm{Y}^1}<1$, using the proof of Lemma 6 $(i),(ii)$ and [20, Chap. 2, Cor. 8.2]. The estimate follows since $\|\mathbf{K}-\mathbf{K}^{[n]}\|_{\mathrm{Y}^1}=O(B/n)$, from Lemma 5 and

$$\|(\mathbf{I}+\mathbf{K})^{-1}\|_{\mathrm{Y}^1}\leqslant\sum_{i\geqslant 0}(6di+1)\frac{2C}{i!}\leqslant(12dB+1)\exp(2B),$$

using Lemma 4 and the fact that $C$ (defined in (13)) is upper bounded by $B$.

Now, for the almost-banded parameters $h',d'$, we provide a practical estimate of

$$h',d'=O(dB).$$

This is based on the observation that for sufficiently large $n$, we can expect the $\ell$-th iterated operator $(\mathbf{K}^{[n]})^\ell$ to behave approximately like $\mathbf{K}^\ell$. Since $\|\mathbf{K}^\ell\|_{\mathrm{Y}^1}\leqslant(6d\ell+1)(2B)^\ell/\ell!$ (proof of Lemma 4), this quantity falls below 1 as soon as $\ell\approx 2B\exp(1)$. Then $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}=\sum_{i\geqslant 0}(-\mathbf{K}^{[n]})^i$, and $\mathbf{A}=\sum_{i=0}^{\ell-1}$ is $(d(\ell-1),d(\ell-1))$-almost-banded (again in proof of Lemma 4). We therefore obtain an approximation error $\|\mathbf{I}-\mathbf{A}(\mathbf{I}+\mathbf{K}^{[n]})\|_{\mathrm{Y}^1}=\|(\mathbf{K}^{[n]})^\ell\|_{\mathrm{Y}^1}<1$.

To conclude, although it provides a rigorous complexity estimate, the bound concerning $n$ is usually very pessimistic. This is because the above mentioned practical approach of doubling $n$ ends up with far smaller values in most cases. It often happens that $\|(\mathbf{I}+\mathbf{K})^{-1}\|_{\mathrm{Y}^1}$ does not follow an exponential growth when $B=\sum_{j=0}^{r-1}\|b_j\|_{\mathrm{Y}^1}$ becomes large. For instance, when $k(t,s)$ is nonnegative, then the Neumann series $\sum_{i\geqslant 0}\mathbf{K}^i$ (equal to $\mathbf{I}+\mathbf{K}$) alternates signs and the $\mathrm{Y}^1$-norm of $(\mathbf{I}+\mathbf{K})^{-1}$ is far smaller than the term-by-term exponential bound. Several examples in Section 7 illustrate this phenomenon. In the difficult cases involving an exponential growth of $\|(\mathbf{I}+\mathbf{K})^{-1}\|_{\mathrm{Y}^1}$, the examples in Section 7 also show how the almost-banded approach helps to keep the computation tractable up to some extent.

# 6 Extensions to non-polynomial LODEs

In this section, we show how to address the general case stated in Problem 1.1. In Subsection 6.1 we extend the previously described method to the non-polynomial case with Cauchy boundary conditions. Then we discuss the case of other boundary conditions in Subsection 6.2.

## 6.1 Extension to non-polynomial IVP

We consider the IVP problem (1a) where the coefficients $\alpha_j$, $j = 0, \dots, r-1$, and the right hand side $\gamma$ belong to $\mathrm{Ч}^1$ and are rigorously approximated by Chebyshev models $\boldsymbol{\alpha}_j = (a_j, \varepsilon_j)$ and $\boldsymbol{\gamma} = (g, \tau)$ obtained as in Section 2.3. Using Proposition 3, we get an integral operator $\mathbf{K}$ with a kernel $k(t,s)$ which is polynomial in the variable $s$ only:

$$k(t,s) = \sum_{j=0}^{r-1} \beta_j(t) T_j(s),$$

where the $\beta_j$ are non-polynomial functions in $\mathrm{Ч}^1$.

To obtain Chebyshev models $\boldsymbol{\beta}_j = (\widetilde{b}_j, \eta_j)$ for $\beta_j$ it suffices to run Algorithm 1 where one replaces the polynomials $a_j$ by Chebyshev models $\boldsymbol{\alpha}_j$ and overloads corresponding arithmetic operations. Then, the polynomials $\widetilde{b}_j$ define a polynomial kernel $k_P(t,s)$ as in equation (14) and respectively the polynomial integral operator $\mathbf{K}_P$, such that:

$$\|\mathbf{K} - \mathbf{K}_P\|_{\mathrm{Ч}^1} \leqslant 2 \sum_{j=0}^{r-1} \eta_j. \tag{23}$$

Moreover, since Algorithm 1 only performs linear operations on the Chebyshev models $\boldsymbol{\alpha}_j$ to produce the $\boldsymbol{\beta}_j$, the quantity $\sum_{0 \leqslant j < r} \eta_j$ is upper bounded by $C \sum_{0 \leqslant j < r} \varepsilon_j$ for some constant $C$ depending only on $r$. This justifies the fact that $\mathbf{K}$ is well approximated by $\mathbf{K}_P$ when the coefficients $\alpha_j$ are well approximated by the $a_j$.

Let us prove that the truncated operators $\mathbf{K}^{[n]} := \mathrm{п}_n \cdot \mathbf{K} \cdot \mathrm{п}_n$ still converge to $\mathbf{K}$ and that $\mathbf{I} + \mathbf{K}$ is an isomorphism of $\mathrm{Ч}^1$:

**Lemma 8.** *Let $\mathbf{K}$ be the integral operator obtained from Proposition 3. We have*

*1) $\mathbf{K}$ is a bounded linear operator of $\mathrm{Ч}^1$ with:*

$$\|\mathbf{K}\|_{\mathrm{Ч}^1} \leqslant 2 \sum_{j=0}^{r-1} \|\beta_j\|_{\mathrm{Ч}^1}.$$

*2) $\mathbf{K}^{[n]} \to \mathbf{K}$ for the $\|\cdot\|_{\mathrm{Ч}^1}$-operator norm as $n \to \infty$. Hence $\mathbf{K}$ is compact.*

*3) $\mathbf{I} + \mathbf{K}$ is a bicontinuous isomorphism of $\mathrm{Ч}^1$.*

*Proof.* 1) Let $\varphi \in \Psi^1$. From (7) and (8), we have

$$\|\mathbf{K}\cdot\varphi\|_{\Psi^1} \leqslant \sum_{j=0}^{r-1} \|\beta_j\|_{\Psi^1}(2\|T_j\|_{\Psi^1}\|\varphi\|_{\Psi^1}) = \left(2\sum_{j=0}^{r-1}\|\beta_j\|_{\Psi^1}\right)\|\varphi\|_{\Psi^1}.$$

This shows that $\mathbf{K}\cdot\varphi \in \Psi^1$ and that $\mathbf{K}$ is bounded as endomorphism of $(\Psi^1, \|\cdot\|_{\Psi^1})$ with the bound claimed above.

2) Let $\varepsilon > 0$. If the $\varepsilon_j$ (and consequently the $\eta_j$) are small enough, then $\|\mathbf{K}-\mathbf{K}_P\|_{\Psi^1} \leqslant \varepsilon/3$ by (23). We know from Lemma 5, since $\mathbf{K}_P$ is polynomial, that for $n$ large enough, $\|\mathbf{K}_P - \mathbf{K}_P^{[n]}\|_{\Psi^1} \leqslant \varepsilon/3$. We finally get:

$$\begin{aligned}\|\mathbf{K}-\mathbf{K}^{[n]}\|_{\Psi^1} &\leqslant \|\mathbf{K}-\mathbf{K}_P\|_{\Psi^1} + \|\mathbf{K}_P-\mathbf{K}_P^{[n]}\|_{\Psi^1} + \|\mathbf{K}_P^{[n]}-\mathbf{K}^{[n]}\|_{\Psi^1}\\ &\leqslant \|\mathbf{K}-\mathbf{K}_P\|_{\Psi^1} + \|\mathbf{K}_P-\mathbf{K}_P^{[n]}\|_{\Psi^1} + \|\mathbf{K}_P-\mathbf{K}\|_{\Psi^1}\\ &\leqslant \frac{\varepsilon}{3}+\frac{\varepsilon}{3}+\frac{\varepsilon}{3} = \varepsilon.\end{aligned}$$

3) The proof works exactly as in the polynomial case: we know that $\mathbf{K}$ is compact by 2) and that $\mathbf{I}+\mathbf{K}$ is injective because it is injective over the superspace $\mathcal{C}^0$, cf. Section 3.1, and we conclude thanks to the Fredholm alternative. □

Using this result, we can again form the Newton-like operator $\mathbf{T}$ as in Section 5, with $\mathbf{A} \approx (\mathbf{I}+\mathbf{K}_P^{[n]})^{-1}$ for some large enough value of $n$.

The operator norm of the linear part of $\mathbf{T}$ can now be decomposed into three parts:

$$\|\mathbf{I}-\mathbf{A}\cdot(\mathbf{I}+\mathbf{K})\|_{\Psi^1} \leqslant \|\mathbf{I}-\mathbf{A}\cdot(\mathbf{I}+\mathbf{K}_P^{[n]})\|_{\Psi^1} + \|\mathbf{A}\cdot(\mathbf{K}_P-\mathbf{K}_P^{[n]})\|_{\Psi^1} + \|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}_P)\|_{\Psi^1}.$$

The first two parts are exactly the ones of (22) (where the polynomial integral operator $\mathbf{K}$ is now called $\mathbf{K}_P$) and can be rigorously upper bounded using the same techniques. The last part can be upper bounded thanks to (23):

$$\|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}_P)\|_{\Psi^1} \leqslant 2\|\mathbf{A}\|_{\Psi^1}\sum_{j=0}^{r-1}\eta_j. \tag{24}$$

It is interesting to notice that the order of magnitude of $n$ is largely determined by the second part (as in the polynomial case), whereas the third part forces the $\eta_j$ (and hence the $\varepsilon_j$) to be small, which mainly depends on the degree of the approximating polynomials $a_j(t)$ for $\alpha_j(t)$.

Finally, let $\widetilde{\varphi}$ be the numerical approximation for the solution of the IVP problem (1a), given as a polynomial in the Chebyshev basis. One upper bounds $\|\mathbf{T}\cdot\widetilde{\varphi}-\widetilde{\varphi}\|_{\Psi^1} = \|\mathbf{A}\cdot(\widetilde{\varphi}+\mathbf{K}\cdot\widetilde{\varphi}-\psi)\|_{\Psi^1} \leqslant \|\mathbf{A}\cdot z\|_{\Psi^1}+\tau\|\mathbf{A}\|_{\Psi^1}$, where $\boldsymbol{\zeta} = (z,\tau)$ is a Chebyshev model of $\widetilde{\varphi}+\mathbf{K}\cdot\widetilde{\varphi}-\psi$ obtained by arithmetic operations described in Section 2.3.

**Proposition 9.** *The results of Proposition 8 remain valid for the IVP validation in the non-polynomial case* (1a).

*Proof.* For computing a rigorous Lipschitz constant for $\mathbf{T}$, the additional term $\|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}_P)\|_{\Psi^1}$ is bounded by (24). Clearly, this additional cost is dominated by the complexity obtained in Proposition 8 (*i*) for the polynomial case.

Then, validating a candidate solution $\widetilde{\varphi}$ has the same cost as in the polynomial case (Proposition 8 (*ii*)), since all polynomial operations are essentially replaced by their Chebyshev model extensions. □

In conclusion, we observe that our validation method is easily adapted to the general case where the coefficients $\alpha_j$ are non-polynomial functions rigorously approximated by polynomials $a_j$. However, contrary to the polynomial case where the involved degrees are usually low, the degrees of the approximants $a_j$ can be rather large, resulting in a dense linear problem and poorer time efficiency. And yet, in practice, the method remains efficient on problems with reasonable coefficient magnitude and time interval under consideration, which will be exemplified in Section 7.

## 6.2 The case of other boundary conditions

Consider now the general boundary conditions operator $\mathbf{\Lambda} : \Upsilon^1 \to \mathbb{R}^r$ of Problem (1b). In [52] an *ad-hoc* integral reformulation is proposed to treat a specific case of such boundary conditions, while other works like [14] propose a generic reformulation method. Our method consists in reducing a general BVP validation problem to $r+1$ IVP validation problems. This is easily observed, since the initial values $v_j = f^{(j)}(t_0)$ appearing in the integral reformulation of Proposition 3 are now unknown. At first sight, this may seem rather naive and time-consuming. However, the most difficult part which consists in obtaining a contracting Newton-like operator is performed only once, thus considerably reducing the total computation time.

Suppose we have a candidate polynomial approximation $\widetilde{f}$ of the solution of BVP problem (1b), given in Chebyshev basis. Our method consists in rigorously computing a very accurate approximation $\overline{f}$ and then comparing it with $\widetilde{f}$.

1. The first step is to provide $\mathbf{A}$ and compute an upper bound $\mu$ for $\|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I} + \mathbf{K})\|_{\Upsilon^1}$. This depends neither on the initial conditions nor on the right hand side $\gamma(t)$.

2. Then, for each $i \in [\![0, r-1]\!]$, we compute (with Algorithms 2 and 3 for the underlying linear algebra) and validate with Algorithm 8 an approximation $\widetilde{f}_i$ for the solution $f_i^*$ of the homogeneous LODE associated to (1) (that is, with right hand side $g = 0$) with initial conditions:

$$v_j = f_i^{(j)}(t_0) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \qquad 0 \leqslant j < r.$$

   Similarly, we approximate and validate the solution $f_r^*$ of Equation (1) with right hand side $g$ and null initial conditions ($f_r^{(j)} = 0$ for $0 \leqslant j < r$). Since the validation kernel has been produced at the previous step, the numerical solving procedure (Algorithms 2 and 3) as well as the validation (Algorithm 8) are linear in the degree of the approximant. Thus, we obtain Chebyshev models for $f_i^*$, and for their derivatives $f_i^{*(j)}$, $0 \leqslant j \leqslant r$.

3. The original equation with boundary conditions $\mathbf{\Lambda} \cdot f = (\lambda_0(f), \ldots, \lambda_{r-1}(f)) = (v_0, \ldots, v_{r-1})$ admits a unique solution $f^*$ if and only if there exist $c_0, c_1, \ldots, c_{r-1}$ uniquely determined such that

$$f^* = c_0 f_0^* + c_1 f_1^* + \cdots + c_{r-1} f_{r-1}^* + f_r^*$$

and

$$\begin{aligned}
\lambda_0(f_0^*)c_0 + \lambda_0(f_1^*)c_1 + \cdots + \lambda_0(f_{r-1}^*)c_{r-1} &= -\lambda_0(f_r^*),\\
\lambda_1(f_0^*)c_0 + \lambda_1(f_1^*)c_1 + \cdots + \lambda_1(f_{r-1}^*)c_{r-1} &= -\lambda_1(f_r^*),\\
&\vdots\\
\lambda_{r-1}(f_0^*)c_0 + \lambda_{r-1}(f_1^*)c_1 + \cdots + \lambda_{r-1}(f_{r-1}^*)c_{r-1} &= -\lambda_{r-1}(f_r^*).
\end{aligned}$$

If the quantities $\lambda_j(f_i^*)$ can be rigorously and accurately computed using the Chebyshev models of the $f_i^{*(j)}$ obtained at the previous step, then one can solve this linear system in interval arithmetics [43].

4. Using the (interval) coefficients $c_0, \ldots, c_{r-1}$ and the Chebyshev models $\boldsymbol{f_0}, \ldots, \boldsymbol{f_{r-1}}, \boldsymbol{f_r}$, we get that

$$\boldsymbol{f} = (\overline{f}, \varepsilon) := c_0\boldsymbol{f_0} + \cdots + c_{r-1}\boldsymbol{f_{r-1}} + \boldsymbol{f_r}$$

is a Chebyshev model for the exact solution $f^*$. Now, it suffices to compute $\eta = \|\widetilde{f} - \overline{f}\|_{\mathrm{L}^1}$ (which is straightforward since both $\widetilde{f}$ and $\overline{f}$ are polynomials in Chebyshev basis) and we deduce that the exact error $\|\widetilde{f} - f^*\|_{\mathrm{L}^1}$ belongs to the interval $[\max(\eta - \varepsilon, 0), \eta + \varepsilon]$. Note that the intermediate approximant $\overline{f}$ has to be sharp enough (that is, the approximation degree has to be chosen large enough), such that $\varepsilon \ll \eta$ which gives a sharp enclosure of the error.

# 7 Experimental results

Four examples illustrate our validation method and investigate its limitations, two of which are already treated in [38] from the numerical point of view. First, Airy differential equation exemplifies the polynomial IVP case. Second, the non-polynomial IVP case is illustrated by the mechanical study of the undamped pendulum with variable length. Third, a non-polynomial BVP problem is exemplified by a boundary layer problem. Finally, we apply our method to a practical space mission problem, namely, the trajectory validation in linearized Keplerian dynamics. Based on the validation method presented in this article, more detailed applications to space mission problems are exposed in [2].

**Remark 7.1.** *As explained in Section 5.2, the magnitude of the validation parameters $n$, $h'$ and $d'$ required by Algorithm 7 mainly determines the time complexity of the method. In the examples analyzed in this section, we particularly investigate their evolution in function of the parameters of the problems. Usually, they are automatically determined as proposed in Section 5.2.2 (doubling them until the operator $\mathbf{T}$ is proved to be contracting).*

## 7.1 Airy equation

The Airy function of the first kind is a special function defined by $\mathrm{Ai}(x) = 1/\pi \int_0^\infty \cos(s^3/3 + xs)\mathrm{d}s$ and solution of the Airy differential equation:

$$y''(x) - xy(x) = 0, \tag{25}$$

with the initial conditions at 0:

$$\mathrm{Ai}(0) = \frac{1}{3^{2/3}\Gamma(2/3)}, \qquad \mathrm{Ai}'(0) = -\frac{1}{3^{1/3}\Gamma(1/3)}.$$

Airy functions, Ai and Bi, depicted in Figure 3, form together the standard basis of the solutions space of (25) (see [1], Chap. 10 *Bessel Functions of Fractional Order*).

In what follows, we apply the validation method on intervals of the form $[-a, 0]$ or $[0, a]$ (for $a > 0$), and investigate its behavior in these two different cases.

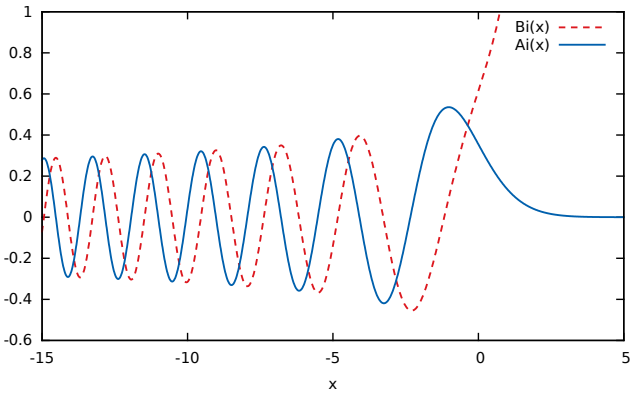



Figure 3: Airy functions of the first and second kinds

### 7.1.1 Validation over the negative axis

We rigorously approximate Ai over $[-a, 0]$ for some $a > 0$, or equivalently $u(t) = \mathrm{Ai}(-(1+t)a/2)$ over $[-1, 1]$. This appears for instance in quantum mechanics when considering a particle in a one-dimensional uniform electric field. The function $u$ is solution of the following IVP problem:

$$u''(t) + \frac{a^3}{8}(1+t)u(t) = 0,$$
$$u(-1) = \frac{1}{3^{2/3}\Gamma(2/3)} \quad \text{and} \quad u'(-1) = \frac{a/2}{3^{1/3}\Gamma(1/3)}.$$

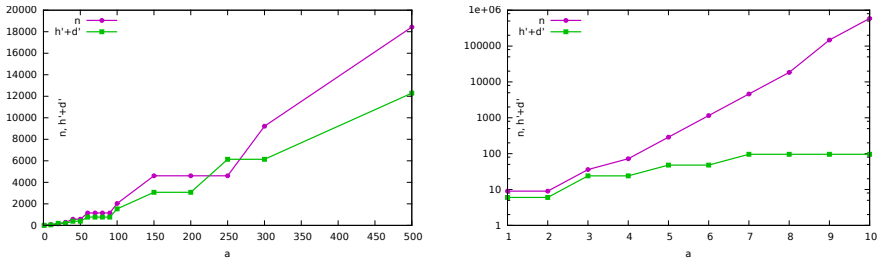
After the integral transform, we get:

$$k(t,s) = \frac{a^3}{8}(1+t)(t-s), \quad \psi(t) = -\frac{a^3}{8}((1+t)u(-1) + (1+t)^2u'(-1)),$$
$$b_0(t) = \frac{a^3}{16}(T_0(t) + 2T_1(t) + T_2(t)), \quad b_1(t) = -\frac{a^3}{8}(T_0(t) + T_1(t)),$$
$$h = 2, \quad d = 3.$$

Figure 4(a) illustrates the growth of the parameters $n$, $h'$ and $d'$ chosen for Algorithm 7 in order to obtain a contracting Newton-like operator $\mathbf{T}$ when

$a$ varies. We observe that $n$ grows considerably slower than the pessimistic exponential bound claimed in Section 5.2.2. In counterpart, the quantity $h'+d'$ is of the order of magnitude of $n$, which means that the almost-banded approach computes a dense $A$ and could therefore be replaced by a direct numerical computation of $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$.

As an example, let $\widetilde{f}$ be a degree 48 approximation of Ai over $[-10,0]$ that reaches machine precision, obtained using the integral reformulation and Algorithms 2 and 3 for linear algebra. We want to validate this candidate approximation. The problem is rescaled over $[-1,1]$ as above, with $a=10$. Call $\widetilde{\varphi}=\widetilde{f}''$, and $\varphi^*$ the exact mathematical solution of the integral equation associated to our problem. With $n=72$, $h'=24$, $d'=24$ (automatically obtained as recalled in Remark 7.1), Algorithm 7 produces a contracting operator $\mathbf{T}$ with $\mu=0.128$. After that, we run Algorithm 8: we evaluate $\|\widetilde{\varphi}-\mathbf{T}\cdot\widetilde{\varphi}\|_{\mathrm{Y}^1}=\|\mathbf{A}\cdot(\widetilde{\varphi}+\mathbf{K}\cdot\widetilde{\varphi}-\psi)\|_{\mathrm{Y}^1}$ and obtain $3.87\cdot10^{-18}$. So we finally get $\|\widetilde{\varphi}-\varphi^*\|_{\mathrm{Y}^1}\leqslant 3.87\cdot10^{-18}/(1-0.128)=4.43\cdot10^{-18}$ and $\widetilde{f}=u(-1)+(1+t)u'(-1)+\int_{-1}^{t}\int_{-1}^{s}\widetilde{\varphi}(\tau)\mathrm{d}\tau\mathrm{d}s$ (of degree 50) approximates $u(t)=\mathrm{Ai}(-(1+t)a/2)$ within a $\mathrm{Y}^1$-error equal to $1.78\cdot10^{-17}$, which is already beyond machine precision.




(a) over $[-a,0]$ (b) over $[0,a]$

Figure 4: Parameters $n$ and $h'+d'$ chosen during the validation of the Airy function, in function of $a$

### 7.1.2 Validation over the positive axis

We similarly pose $u(t)=\mathrm{Ai}((1+t)a/2)$ to study Ai over the segment $[0,a]$ (with $a>0$) on the real positive axis. The differential equation and integral transform are similar to the case above, except for the signs:

$$u''(t)-\frac{a^3}{8}(1+t)u(t)=0,$$

$$u(-1)=\frac{1}{3^{2/3}\Gamma(2/3)}\quad\text{and}\quad u'(-1)=-\frac{a/2}{3^{1/3}\Gamma(1/3)},$$

$$k(t,s)=-\frac{a^3}{8}(1+t)(t-s),\quad \psi(t)=\frac{a^3}{8}((1+t)u(-1)+(1+t)^2u'(-1)),$$

$$b_0(t)=-\frac{a^3}{16}(T_0(t)+2T_1(t)+T_2(t)),\quad b_1(t)=\frac{a^3}{8}(T_0(t)+T_1(t)),$$

$$h=2,\quad d=3.$$

Though, we observe a very different behavior for the parameters in Figure 4(b). The truncation order $n$ grows exponentially and seems in accordance with the pessimistic bound (5.2.2). On the opposite, $h' + d'$ remains significantly smaller than $n$, justifying the use of the almost-banded approach.

To understand the difference between the positive and negative cases, let us analyze the behavior of Ai and Bi, see Figure 3, which is directly linked to the norm of $(\mathbf{I} + \mathbf{K})^{-1}$. Over the negative axis, both Ai and Bi have bounded oscillations. Thus, the intuition is that $\|(\mathbf{I} + \mathbf{K})^{-1}\|_{\mathcal{H}^1}$ does not grow so fast when the interval $[-a, 0]$ becomes large, and stays far below the exponential bound $B \exp(2B)$ despite its necessary growth due to these oscillations. On the contrary, Bi grows exponentially fast over the positive axis (we have the asymptotic formula $\mathrm{Bi}(x) \sim \frac{\exp(\frac{2}{3}x^{3/2})}{\sqrt{\pi}x^{1/4}}$ when $x \to +\infty$ [1, 10.4.63]). This clearly implies that $\|(\mathbf{I} + \mathbf{K})^{-1}\|_{\mathcal{H}^1}$ must grow exponentially with $a$. For the bound based on the Neumann series of the operator, one can reason by analogy with the scalar case, using for example the exponential series $\exp(x) = \sum_{i \geqslant 0} x^i/i!$. When $x$ is a large negative number, the series is alternating, hence, its evaluation $\exp(x)$ is far smaller than the bound $\exp(|x|)$ computed by taking each term of the series in absolute value.

## 7.2 Undamped pendulum with variable length

Consider the motion of an undamped pendulum with variable length $\ell(t)$, which is modeled by the equation:

$$\theta''(t) + 2\frac{\ell'(t)}{\ell(t)}\theta'(t) + \frac{g}{\ell(t)}\sin\theta(t) = 0, \tag{26}$$

where $\theta(t)$ is the angle at time $t$ between the pendulum and its equilibrium position, and $g = 9.81$ the gravitational acceleration. On the time interval $[-1, 1]$ and for a constant variation of the length $\ell(t) = \ell_0(1+\zeta t)$ (with $|\zeta| < 1$), we analyze the evolution of $\theta(t)$ in a small neighborhood of 0 such that $\sin\theta$ can be linearized into $\theta$. Equation (26) becomes:

$$\theta''(t) + \frac{2\zeta}{1+\zeta t}\theta'(t) + \frac{g}{\ell_0(1+\zeta t)}\theta(t) = 0, \qquad \theta(-1) = \theta_0 \ll 1 \text{ and } \theta'(-1) = 0.$$

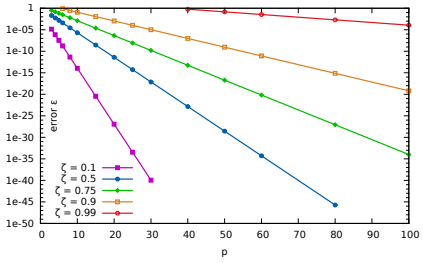
The coefficients of this equation are not polynomials. Hence, we first provide a Chebyshev model for $\xi(t) = 1/(1+\zeta t)$ with $|\zeta| < 1$. If $\zeta \in \mathbb{Q}$, we can use the algorithm for certified Chebyshev expansion of rational functions presented in [4, Algorithm 5.6], which relies on the Bronstein-Salvy algorithm. Otherwise, our solution consists in a generic fixed-point validation method for quotient of Chebyshev models. Figure 5(a) summarizes the obtained error bound $\varepsilon$ (for the $\mathcal{H}^1$-norm) in function of the approximation degree $p$ for different values of $\zeta$, with $\ell_0 = 1$ fixed.

Next, we create and bound the contracting Newton-like operator $\mathbf{T}$. Figure 5(b) shows the corresponding values of $p$ (degree of the approximant of $t \mapsto 1/(1+\zeta t)$) and $n$ (truncation order for the integral operator) we use for Algorithm 7, and the values of $h' + d'$, expressing the advantage of taking an almost-banded $A$ instead of a dense one.

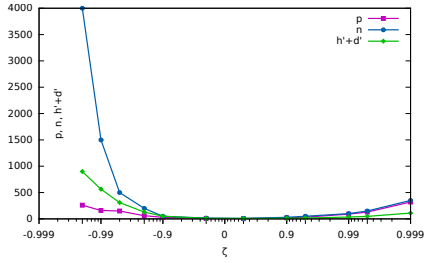
We first observe that $n$ grows when $|\zeta|$ gets close to 1, which is due to the growth of the $\mathcal{H}^1$-norm of $t \mapsto 1/(1+\zeta t)$. However, the situation is very different

depending on the sign of $\zeta$. When $\zeta$ gets close to $-1$, $n$ grows exponentially fast. The quantity $h' + d'$ grows slowlier, so that the almost-banded approach helps a little. As for the Airy function, this exponential behavior is due to the large negative coefficient in front of $y'$ in equation (26). This difficult case corresponds to a decrease in the rope's length, resulting in increasing oscillations of the pendulum (see Figure 5(d)). On the contrary, the case $\zeta \to 1$ is easier to treat, since it corresponds to an increase of the rope's length, producing damped oscillations of the pendulum (see Figure 5(c)).

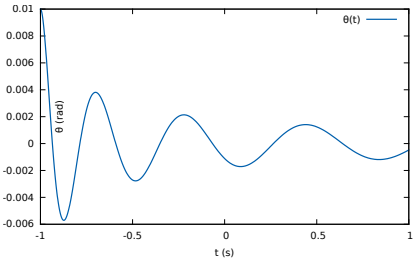
The two numerical solutions plotted on Figures 5(c) and 5(d) were certified using Algorithm 8. For the damped case ($\ell_0 = 0.1$ and $\zeta = 0.9$), we obtained a Chebyshev model of degree 50 with a $\mathrm{Y}^1$-error equal to $1.40 \cdot 10^{-4}$. The diverging case ($\ell_0 = 0.1$ and $\zeta = -0.9$) used a Chebyshev model of degree 65 with an error of $1.15 \cdot 10^{-4}$.



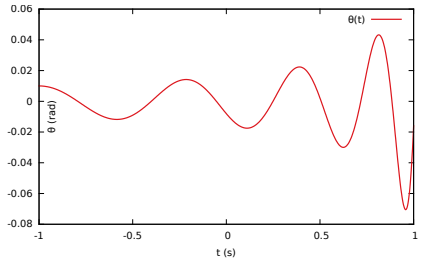

(a) Approximation error $\varepsilon$ of the coefficient $t \mapsto 1/(1+\zeta t)$ in function of approximation degree $d$




(b) Parameters $p$, $n$ and $h' + d'$ chosen during the validation, for $\ell_0 = 1$ fixed and in function of $\zeta$




(c) Damped oscillations of the pendulum obtained for $\ell_0 = 0.1$ and $\zeta = 0.9$




(d) Diverging oscillations of the pendulum obtained for $\ell_0 = 0.1$ and $\zeta = -0.9$

Figure 5: Validation process for the parametric pendulum

## 7.3 Boundary layer problem

We take from [38] the example of the boundary layer problem, modeled by the following BVP problem, with $\varepsilon > 0$:

$$u''(x) - \frac{3x}{\varepsilon}\left(\cos x - \frac{8}{10}\right)u'(x) + \left(\cos x - \frac{8}{10}\right)u(x) = 0, \quad x \in [-1, 1], \quad u(-1) = 1, \quad u(1) = 1. \tag{27}$$

The numerical solution of this BVP is plotted in Figure 6(a) for three different values of $\varepsilon$. Figure 6(b) shows the basis $(u_1, u_2)$ of the solution space of LODE (27) associated to the canonical initial conditions $\{u_1(-1) = 1, u_1'(-1) = 0\}$ and $\{u_2(-1) = 0, u_2'(-1) = 1\}$, for $\varepsilon = 0.00001$. Thus, the exact solution $u$ of the BVP is given by:

$$u(x) = u_1(x) + \lambda u_2(x), \quad \text{with } \lambda = \frac{1 - u_1(1)}{u_2(1)}. \tag{28}$$

Since $u_1(1)$ and $u_2(1)$ tend to be very large when $\varepsilon$ gets close to zero, obtaining $u$ from $u_1$ and $u_2$ is an ill-conditioned problem. With $\varepsilon = 0.00001$, the obtained approximation using the binary64 (double) format is completely inaccurate (see Figure 6(c)). Note that a better solution (regarding the conditioning) is to directly compute the BVP solution with Algorithms 2 and 3 as in [38]. In any case, validating a candidate solution is useful to detect such numerical troubles.

The first task consists in rigorously approximating the cosine function over $[-1, 1]$. This can be done by a recursive call to our validation method on the differential equation $\xi'' + \xi = 0$ with $\xi(-1) = \cos(-1)$ and $\xi'(-1) = -\sin(-1)$. For this application, a degree 10 Chebyshev model for cos is sufficient.

Then, we run Algorithm 7 to get a contracting Newton-like operator. Figure 6(d) illustrates the growth of the validation parameters in function of $\varepsilon$. When $\varepsilon > 0$ gets small, the coefficient in front of $y'$ takes large negative values, yielding an exponential growth of $\|(\mathbf{I}+\mathbf{K})^{-1}\|_{\mathcal{Y}^1}$ and hence of the minimal truncation order $n$ we can choose. Since $h' + d'$ remains small compared to $n$, we get here a typical example where the exponential bound prevents us from validating a solution of LODE (27) with very small $\varepsilon$, but where however the choice of an almost-banded $A$ allows us to treat intermediate cases: $\varepsilon \in [0.005, 0.01]$.

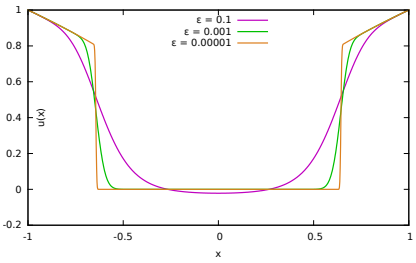
Next, we compute high-degree Chebyshev models $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ for the basis $(u_1, u_2)$. This requires Algorithms 2 and 3 to obtain a numerical approximation, and using the previously obtained Newton-like operator $\mathbf{T}$ to certify them with Algorithm 8. Hence, this step has a linear complexity with respect to the approximation degree we use. Computing the value of $\lambda$ in Equation (28) in interval arithmetics gives a Chebyshev model $\boldsymbol{u}$ for the exact solution $u$ using $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. Finally, the error associated to the candidate numerical approximate solution $\widetilde{u}$ is obtained by adding the certified error of $\boldsymbol{u}$ with the $\mathcal{Y}^1$-distance between $\widetilde{u}$ and the polynomial of $\boldsymbol{u}$.

As an example, for $\varepsilon = 0.01$, the minimal degree for which we found an approximation of the solution of BVP (27) within a certified error of $2^{-53}$ (corresponding to standard double precision) is 72.

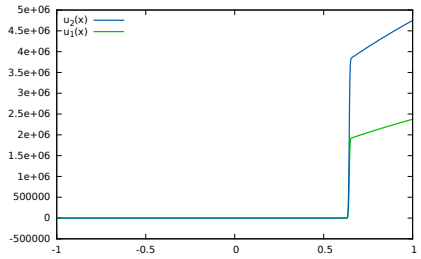
## 7.4 Spacecraft trajectories using linearized equations for Keplerian motion

We consider the case of Tschauner-Hempel equations, which model the linearized relative motion of an active spacecraft around a passive target (such as the International Space Station for instance) in elliptic orbit around the Earth, provided that their relative distance is small with respect to their distance to the Earth. These equations are very used in robust rendezvous space missions [48], where the accuracy of their computed solutions is at stake.

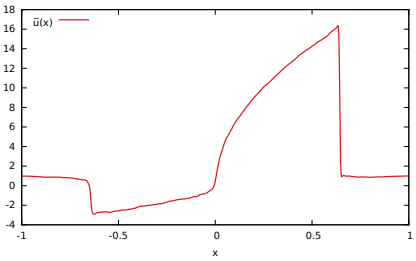
Call $e \in [0, 1)$ the eccentricity of the fixed orbit of the target, and let $\nu$ be the true anomaly (an angular parameter that defines the position of a body moving



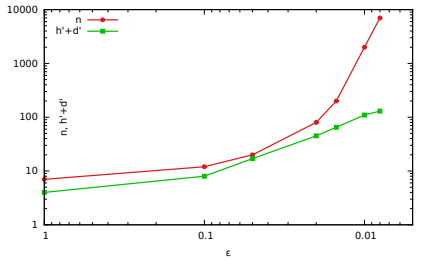

(a) Numerical solution of BVP Problem (27) for different values of $\varepsilon$




(b) Numerically computed basis $(u_1, u_2)$ of the solution space for $\varepsilon = 0.00001$




(c) Inaccurate numerical solution $\widetilde{u}$ obtained with $\varepsilon = 0.00001$




(d) Parameters $n$ and $h' + d'$ chosen during the validation, in function of $\varepsilon$

Figure 6: Validation process for the boundary layer problem

along a Keplerian orbit) associated to the target, which is the independent variable in our problem. The in-plane motion of the spacecraft relatively to the target (that is, the component of the motion inside the plane supported by the elliptic orbit of the chaser) is defined using two position variables $x(\nu)$ and $z(\nu)$, satisfying the following linearized system over the interval $[\nu_0, \nu_f]$:

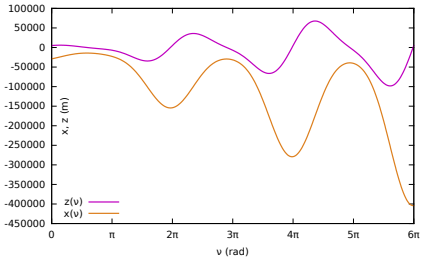
$$z''(\nu) + \left(4 - \frac{3}{1 + e\cos\nu}\right) z(\nu) = c,$$

$$x(\nu) = x(\nu_0) + (x'(\nu_0) - 2z(\nu_0))(\nu - \nu_0) + 2\int_{\nu_0}^{\nu} z(s)\mathrm{d}s,$$

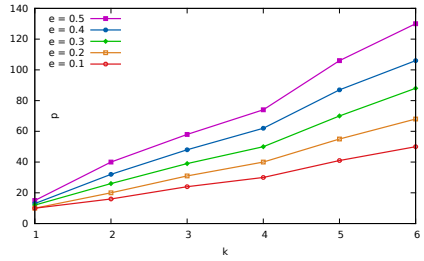
$$c = 4z(\nu_0) - 2x'(\nu_0) \quad \text{and} \quad \nu \in [\nu_0, \nu_f].$$

As an example, fix the eccentricity $e = 0.5$, the interval $[\nu_0, \nu_f] = [0, 6\pi]$ (corresponding to 3 periods) and the initial conditions $(x(\nu_0), z(\nu_0), x'(\nu_0), z'(\nu_0)) = (-3 \cdot 10^4 \text{ m}, 5 \cdot 10^3 \text{ m}, 9 \cdot 10^3 \text{ m} \cdot \text{rad}^{-1}, 4 \cdot 10^3 \text{ m} \cdot \text{rad}^{-1})$. The corresponding functions $x(\nu)$ and $z(\nu)$ are plotted in Figure 7(a). Figure 7(c) represents an approximation of degree $n = 18$ of $z''(\nu)$ (radial acceleration), together with the rigorous error bound obtained by our method. The dashed curve corresponds to the exact solution, which as expected lies inside the tube defined by our rigorous approximation. One notices that we obtain a quite tight error bound, even for the $\|\cdot\|_\infty$ norm.

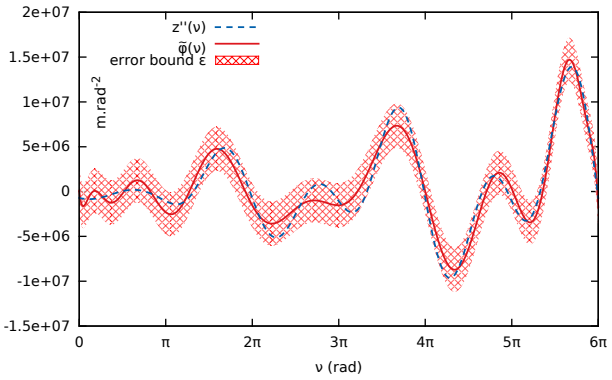
Figure 7(b) gives the minimal degree $p$ corresponding to an approximation of $z$ for which Algorithm 8 is able to certify an error below one meter, in function of the period length and the eccentricity of the target reference orbit.




(a) Exact representation of $x(\nu)$ and $z(\nu)$ for $\nu \in [0, 6\pi]$ and $e = 0.5$




(b) Approximation degree $p$ needed to rigorously approximate $z$ on $[0, k\pi]$ within an error of 1m, in function of $k$ and the eccentricity $e$




(c) Rigorous approximation $\widetilde{\varphi}$ of degree $n = 18$ of $z''$, over $[0, 6\pi]$ and for $e = 0.5$

Figure 7: Different validation results related to the spacecraft rendezvous problem

# 8 Conclusion and future directions

In this article, we proposed a generic efficient algorithm for computing rigorous polynomial approximations for LODEs. We focused on both its theoretical and practical complexity analysis. For this, firstly, we studied theoretical properties like compactness, convergence, invertibility of associated linear integral operators and their truncations over $\mathcal{T}^1$, the coefficient space of Chebyshev series. Then, we focused on the almost-banded matrix structure of these operators, which allowed for very efficient numerical algorithms for both the numerical solutions of LODEs and the rigorous computation of the approximation error. More specifically, the proposed a posteriori validation algorithm is based on a quasi-Newton method, which benefits from the almost-banded structure of intervening operators. Finally, several representative examples showed the advantages of our algorithms as well as their theoretical and practical limits.

Several extensions of this work are possible:

- One of the easiest generalizations, which is work in progress, is the multi-dimensional case, where we have a system of linear ordinary differential equations. In fact, extending the $\mathcal{T}^1$ space to the multi-dimensional case, where functions are of type $[-1, 1] \to \mathbb{R}^p$, is sufficient to that purpose: we still get an almost-banded integral operator in the coefficient space.

- Another work in progress is to rewrite the Picard iterations based validation method presented in [4] as a quasi-Newton validation technique. Then, using our current almost-banded operator based algorithms, we will be able to generalize the method in [4] to the non-homogeneous and non-polynomial LODE case with a better complexity bound, by allowing a more involved analysis of the iterated kernels.

- We also consider the generalization to other classes of orthogonal polynomials, such as Legendre polynomials, or Hermite and Laguerre polynomials over unbounded intervals. In fact, orthogonal polynomials always satisfy a three-term-recurrence, so that the multiplication and integration formulas remain similar, which should produce similar almost-banded integral operators.

- The propagation of uncertain initial conditions via LODEs may also be explored based on our current techniques.

- The existing C implementation will be made available as open source code. Moreover, we also intend to provide a `Coq` implementation to guarantee both the theoretical correctness of that method and the soundness of its current C implementation.

- More involved generalizations are non-linear ODEs and (linear) PDEs. In both cases however, we have to rely on a multivariate approximation theory with orthogonal polynomials (such theories exist but are not unique and depend on the domain of approximation) and the theory for such differential equations are far less structured than the easy linear univariate case. In particular, the time complexity of such extensions may be huge compared to the present case.

## Acknowledgments

This work was partially funded by the ANR FastRelax project ANR-14-CE25-0018-01. We thank Denis Arzelier and Marc Mezzarobba for many useful discussions and/or comments regarding this work.

## References

[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. Courier Corporation, 1964.

[2] P. R. Arantes Gilz, F. Bréhard, and G. Clément. Validated semi-analytical transition matrices for linearized spacecraft dynamics via Chebyshev series approximations. `https://hal.laas.fr/hal-01540170`, June 2017. Preprint.

[3] G. Baszenski and M. Tasche. Fast polynomial multiplication and convolutions related to the discrete cosine transform. *Linear Algebra Appl.*, 252:1–25, 1997.

[4] A. Benoit, M. Joldes, and M. Mezzarobba. Rigorous uniform approximation of D-finite functions using Chebyshev expansions. *Mathematics of Computation*, 2016.

[5] V. Berinde. *Iterative approximation of fixed points*, volume 1912 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007.

[6] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, 2001.

[7] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[8] N. Brisebarre and M. Joldeş. Chebyshev interpolation polynomial-based tools for rigorous computing. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 147–154. ACM, 2010.

[9] E. W. Cheney. *Introduction to approximation theory*. AMS Chelsea Publishing, Providence, RI, 1998. Reprint of the second (1982) edition.

[10] S. Chevillard, J. Harrison, M. Joldeş, and C. Lauter. Efficient and accurate computation of upper bounds of approximation errors. *Theoretical Computer Science*, 412(16):1523–1543, 2011.

[11] C. W. Clenshaw. The numerical solution of linear differential equations in chebyshev series. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 53, pages 134–149. Cambridge Univ Press, 1957.

[12] P. Di Lizia. *Robust Space Trajectory and Space System Design using Differential Algebra*. Ph.D. thesis, Politecnico di Milano, Milano, Italy, 2008.

[13] T. A. Driscoll, F. Bornemann, and L. N. Trefethen. The chebop system for automatic solution of differential equations. *BIT Numerical Mathematics*, 48(4):701–723, 2008.

[14] K. Du. On well-conditioned spectral collocation and spectral methods by the integral reformulation. *SIAM J. Sci. Comput.*, 38(5):A3247–A3263, 2016.

[15] C. Epstein, W. Miranker, and T. Rivlin. Ultra-arithmetic I: function data types. *Mathematics and Computers in Simulation*, 24(1):1–18, 1982.

[16] C. Epstein, W. Miranker, and T. Rivlin. Ultra-arithmetic II: intervals of polynomials. *Mathematics and Computers in Simulation*, 24(1):19–29, 1982.

[17] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding. *ACM Transactions on Mathematical Software*, 33(2), 2007. Available at `http://www.mpfr.org/`.

[18] L. Fox and I. B. Parker. *Chebyshev polynomials in numerical analysis*. Oxford University Press, London-New York-Toronto, Ont., 1968.

[19] P. Giorgi. On polynomial multiplication in Chebyshev basis. *IEEE Trans. Comput.*, 61(6):780–789, 2012.

[20] I. Gohberg, S. Goldberg, and M. A. Kaashoek. *Basic classes of linear operators*. Birkhäuser Verlag, Basel, 2003.

[21] D. Gottlieb and S. A. Orszag. *Numerical Analysis of Spectral Methods: Theory and Applications*, volume 26. Siam, 1977.

[22] L. Greengard. Spectral integration and two-point boundary value problems. *SIAM Journal on Numerical Analysis*, 28(4):1071–1080, 1991.

[23] A. Hungria, J.-P. Lessard, and J. D. Mireles James. Rigorous numerics for analytic solutions of differential equations: the radii polynomial approach. *Mathematics of Computation*, 85(299):1427–1459, 2016.

[24] A. Iserles. *A first course in the numerical analysis of differential equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, second edition, 2009.

[25] M. Joldeş. *Rigorous Polynomial Approximations and Applications*. PhD thesis, École normale supérieure de Lyon – Université de Lyon, Lyon, France, 2011.

[26] Y. Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004.

[27] E. Kaucher and W. Miranker. *Self-validating numerics for function space problems*. Acad. Press, 1984.

[28] T. Lalescu. *Introduction à la théorie des équations intégrales (Introduction to the Theory of Integral Equations)*. Librairie Scientifique A. Hermann, 1911.

[29] J.-P. Lessard and C. Reinhardt. Rigorous numerics for nonlinear differential equations using Chebyshev series. *SIAM Journal on Numerical Analysis*, 52(1):1–22, 2014.

[30] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4(4):379–456, 2003.

[31] J. C. Mason and D. C. Handscomb. *Chebyshev polynomials*. CRC Press, 2002.

[32] R. E. Moore. *Interval Analysis*. Prentice-Hall, 1966.

[33] R. E. Moore and F. Bierbaum. *Methods and applications of interval analysis*, volume 2. SIAM, 1979.

[34] J.-M. Muller. *Elementary Functions, Algorithms and Implementation*. Birkhäuser, Boston, 3rd edition, 2016.

[35] M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45(1):236–262, 2007.

[36] A. Neumaier. *Interval methods for systems of equations*. Cambridge University Press, Cambridge, UK, 1990.

[37] A. Neumaier. Taylor forms – Use and limits. *Reliable Computing*, 9(1):43–79, 2003.

[38] S. Olver and A. Townsend. A fast and well-conditioned spectral method. *SIAM Review*, 55(3):462–489, 2013.

[39] M. J. D. Powell. *Approximation theory and methods*. Cambridge University Press, 1981.

[40] L. B. Rall. *Computational solution of nonlinear operator equations*. Wiley New York, 1969.

[41] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliable Computing*, 11:1–16, 2005. Available at `http://mpfi.gforge.inria.fr/`.

[42] T. J. Rivlin. *The Chebyshev Polynomials*. Wiley, 1974.

[43] S. M. Rump. Verification methods: rigorous results using floating-point arithmetic. *Acta Numer.*, 19:287–449, 2010.

[44] B. Salvy. D-finiteness: Algorithms and applications. In M. Kauers, editor, *ISSAC 2005: Proceedings of the 18th International Symposium on Symbolic and Algebraic Computation, Beijing, China, July 24-27, 2005*, pages 2–3. ACM Press, 2005. Abstract for an invited talk.

[45] R. P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1(2):175–188, 1980.

[46] L. N. Trefethen. Computing Numerically with Functions Instead of Numbers. *Mathematics in Computer Science*, 1(1):9–19, 2007.

[47] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.

[48] J. Tschauner and P. Hempel. Optimale Beschleunigungsprogramme fur das Rendezvous-Manover. *Acta Astronautica*, 10(5-6):296–307, 1964.

[49] W. Tucker. *Validated numerics: a short introduction to rigorous computations*. Princeton University Press, 2011.

[50] J. B. van den Berg and J.-P. Lessard. Rigorous numerics in dynamics. *Notices of the AMS*, 62(9), 2015.

[51] A.-M. Wazwaz. *Linear and nonlinear integral equations: methods and applications*. Springer Science & Business Media, 2011.

[52] N. Yamamoto. A numerical verification method for solutions of boundary value problems with local uniqueness by Banach's fixed-point theorem. *SIAM Journal on Numerical Analysis*, 35(5):2004–2013, 1998.

[53] D. Zeilberger. A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, 32(3):321–368, 1990.

[54] A. Zygmund. *Trigonometric series. Vol. I, II.* Cambridge Mathematical Library. Cambridge University Press, Cambridge, third edition, 2002.