

LEAST-SQUARES SPECTRAL METHODS FOR ODE EIGENVALUE PROBLEMS*

BEHNAM HASHEMI[†] AND YUJI NAKATSUKASA[‡]

Abstract. We develop spectral methods for ODEs and operator eigenvalue problems that are based on a least-squares formulation of the problem. The key tool is a method for rectangular generalized eigenvalue problems, which we extend to quasimatrices and objects combining quasimatrices and matrices. The strength of the approach is its flexibility that lies in the quasimatrix formulation allowing the basis functions to be chosen arbitrarily, a good choice (e.g., those obtained by solving nearby problems) leading to rapid convergence, and often giving high accuracy. We also show how our algorithm can easily be modified to solve problems with eigenvalue-dependent boundary conditions, and discuss reformulations as an integral equation, which often improves the accuracy.

Key words. operator eigenvalue problems, least-squares method, rectangular matrix pencils, spectral methods, quasimatrix

MSC codes. 47A75, 65N35, 65F25, 65F15

DOI. 10.1137/21M1445934

1. Introduction.

Consider a differential eigenvalue problem

$$(1.1) \quad \mathcal{L}u = \lambda u,$$

where \mathcal{L} is a linear differential operator of order d , mapping a univariate function $u : [-1, 1] \rightarrow \mathbb{R}$ to another function $Lu : [-1, 1] \rightarrow \mathbb{R}$, and $b_i(u, u', \dots, u^{(d)}, \lambda) = 0$ for $i = 1, \dots, d$ are homogeneous boundary conditions, which are expressed as a linear combination of the inputs. An example is $u'(0) = \lambda(\frac{3}{2}u(0) + u'(0))$, and $u'(\frac{\pi}{2}) = 0$ for $-u'' = \lambda u$ on $[0, \pi/2]$.¹ The framework to be introduced in this paper may also work for more general problems under well-posedness assumptions. We assume that the spectrum of (1.1) is discrete and that all eigenvalues have finite algebraic multiplicity. The goal is to compute the (selected) eigenvalues λ and their corresponding eigenfunctions u .

Operator eigenvalue problems have a long history. Classical problems include the Schrödinger equation in quantum mechanics [27], the Sturm–Liouville equation [19], and the Orr–Sommerfeld equation in fluid dynamics [39]. A number of further examples can be found in [52, 53], which are illustrated in [52] using Chebfun [18].

In this paper we develop an algorithm for solving operator eigenvalue problems (1.1). The focus is on spectral methods, which are one of the main computational frameworks for solving ODEs of the form $\mathcal{L}u = f$ or (1.1). Broadly speaking, the guiding principle for these methods is as follows.

*Submitted to the journal’s Methods and Algorithms for Scientific Computing section September 13, 2021; accepted for publication (in revised form) August 3, 2022; published electronically October 11, 2022.

<https://doi.org/10.1137/21M1445934>

Funding: The work of the first author was supported by Iran National Science Foundation (INSF) grant 98012590 and by a grant from IPM (1400650036).

[†]Mathematics Program, Department of Mathematics, Statistics and Physics, College of Arts and Sciences, Qatar University, 2713, Doha, Qatar, and School of Mathematics, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395-5746, Tehran, Iran (hoseynhashemi@gmail.com).

[‡]Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK (nakatsukasa@maths.ox.ac.uk).

¹This particular problem is studied in [41, p. 265], which shows that it has three eigenfunctions that have no roots on $(0, \pi/2)$.

1. Form a set of basis functions $\{u_1, \dots, u_n\}$, where each $u_i : [-1, 1] \rightarrow \mathbb{R}$.
2. Find an approximate solution $u = \sum_{i=1}^n c_i u_i \in \mathbf{U} := \text{span}\{u_1, \dots, u_n\}$.

Spectral methods differ with respect to how the basis functions are chosen and how the approximate solution $u \in \mathbf{U}$ is determined. A typical choice of u_i is the Chebyshev polynomial T_{i-1} , or a Fourier (trigonometric) polynomial [10]. The rationale is that if the solution is smooth, then it can be approximated by such polynomials efficiently; for example, one gets exponential convergence for analytic solutions [50]. Methods with this property are called spectrally accurate. We list some variants of spectral methods below. We describe the methods both for operator equations (ODEs) $\mathcal{L}u = f$ and eigenvalue problems (1.1).

- Spectral Galerkin methods: Choose u_i to satisfy the boundary conditions. Then choose c_i by enforcing that the residual is orthogonal to $\text{span}(\mathbf{V})$ (usually $\mathbf{V} = \mathbf{U}^2$). For ODEs this becomes $\langle v, \mathcal{L}u - f \rangle = 0$ for all $v \in \mathbf{V}$. For eigenvalue problems, the condition is $\langle v, \mathcal{L}u - \lambda u \rangle = 0$. Here and below, the inner product is often the L_2 -inner product $\langle u, v \rangle = \int_{-1}^1 u(x)v(x) dx$.
- tau method: Choose u_i as Chebyshev polynomials³ and apply boundary conditions as side constraints (boundary bordering). Then choose c_i via imposing that the residual is a linear combination of high-degree Chebyshev polynomials. For ODEs this means $\mathcal{L}u - f = \sum_{i \geq n-d} \tau_i T_i$ and for eigenvalue problems it is equivalent to $\mathcal{L}u - \lambda u = \sum_{i \geq n-d} \tau_i T_i$.
- Spectral collocation methods [48]: Choose u_i to be smooth functions, usually Chebyshev or Fourier polynomials. Choose $\mathbf{c} = [c_1, \dots, c_n]^T$ (the solution u is usually represented via the values at x_i ; mathematically this is equivalent to choosing \mathbf{c}) by forcing the equation to hold ($\mathcal{L}u - f = 0$ or $\mathcal{L}u - \lambda u = 0$) at these points $(x_i)_{i=0}^n$, called collocation points. Impose boundary conditions by replacing rows of the equation. Chebfun's default ODE solver is based on a collocation approach [1, 17, 52].

Spectral methods are used in Chebfun for solving both operator equations and operator eigenvalue problems to high accuracy. Chebfun exploits block structure information about the corresponding operators at the continuous level. Such information includes the index, nullity, and deficiency of the operator, whose connection to the amount of rectangularity of the operator is made precise in [1]. It is this rectangularity that clarifies the right size of the rectangular discretization matrix [17] to be used in numerical computation.

In this paper we consider a framework in which, conceptually, we would like to

$$\underset{u \in \mathbf{U}}{\text{minimize}} \left\| \begin{bmatrix} \text{residual in operator equation with } u \\ \text{residual in boundary conditions with } u \end{bmatrix} \right\|_2.$$

Here the norm $\|\cdot\|_2$ is applied to an object of the form $\begin{bmatrix} u \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \text{function} \\ \text{vector} \end{bmatrix}$. We define this by⁴ $\|\begin{bmatrix} u \\ \mathbf{b} \end{bmatrix}\|_2 = \sqrt{\|u\|_{L_2}^2 + \|\mathbf{b}\|_2^2} = \sqrt{\int_{-1}^1 |u(x)|^2 dx + \|\mathbf{b}\|_2^2}$. Specifically, we focus on (1.1) and develop spectral methods which proceed as follows. Start with (or keep building) a set of basis functions $\{\phi_1, \dots, \phi_n\}$, and consider finding the function in $\mathbf{U} := \text{span}\{\phi_1, \dots, \phi_n\}$ such that the residual is minimized, that is,

²We follow a common abuse of notation that allows a matrix to represent its span too. Here, \mathbf{U} or \mathbf{V} may denote a quasimatrix or the space spanned by its columns; see comments near the end of this section.

³Legendre polynomials have also been used in the literature [11], although to a lesser extent.

⁴Different weightings can be considered, i.e., $\|\begin{bmatrix} u \\ \mathbf{b} \end{bmatrix}\|_2 = \sqrt{\alpha \|u\|_{L_2}^2 + \beta \|\mathbf{b}\|_2^2}$ for some $\alpha, \beta > 0$. We will revisit this in section 6.1.

$$(1.2) \quad \underset{\substack{u \in \mathbf{U}, \lambda \\ \|u\|_2=1}}{\text{minimize}} \left\| \begin{bmatrix} \mathcal{L}u - \lambda u \\ b_1(u, u', \dots, u^{(d)}, \lambda) \\ \vdots \\ b_d(u, u', \dots, u^{(d)}, \lambda) \end{bmatrix} \right\|_2.$$

Note that the boundary conditions are part of the objective function; thus they are enforced approximately. This lets us start with any set of basis functions \mathbf{U} , unlike the traditional Galerkin method where the basis functions are usually chosen a priori to satisfy the boundary conditions [24, 31], making the method inapplicable when the boundary conditions depend on λ . If it is important to impose the boundary conditions strictly, we can apply a large weight to the last d components of the objective function. We also describe imposing them exactly in section 6.1.

In order to solve (1.2) we formulate it as an “infinite+finite” dimensional, rectangular eigenvalue problem. We then solve it using a generalization of the algorithm by Ito and Murota [33], based on a minimal-perturbation (backward error) approach and originally designed for discrete rectangular eigenvalue problems [9].

Our proposed algorithm falls in the category of spectral methods. It has the following properties:

- No conditions are imposed on the basis functions u_1, \dots, u_n , giving the flexibility to choose them freely.
- The least-squares formulation directly minimizes the residual. Hence, convergence of the ODE residual in the $\|\cdot\|_2$ norm is straightforward to understand: the residual converges precisely at the speed with which the basis functions are able to reduce the residual. However, convergence analysis in terms of the forward error in u is not trivial, as it depends on the condition number of the problem. For eigenvalue problems, even the convergence of the residual is not trivial, as our algorithm is not guaranteed to find solutions with smallest residuals individually; instead it attempts to minimize the norm of the total residuals.

Our method has the property that it allows u_i to be chosen arbitrarily, yet it has a mixture of a Galerkin (in the sense that the “joint” residual is orthogonal to a well-chosen subspace) and a collocation flavor (where the collocation points are chosen to be the entire domain; of course the operator equation cannot be imposed exactly everywhere, so instead we find the least-squares fit).

1.1. Contributions. Below we list some of the advantages of our least-squares approach.

Flexibility: nonstandard basis functions. We believe the most significant advantage of our approach is the flexibility in terms of the choice of the basis functions. In classical approaches—either collocation or Galerkin—the basis functions are chosen a priori, usually as degree-graded polynomials or Fourier series. Such expansion is often very effective, and convergence is exponential provided that the solution is analytic in the domain. However, when the solution is not smooth, these methods converge slowly. By contrast, in our approach the basis functions are allowed to be arbitrary. In particular, when prior knowledge about the solution is available, such as the location of a singularity or solutions for nearby problems, then by incorporating tailor-made functions into the basis functions we can dramatically accelerate the convergence, as we illustrate through numerical examples.

Flexibility: λ -dependent boundary conditions. Our formulation also naturally allows for nonstandard boundary conditions, in particular, those that depend affinely on

the eigenvalue λ , that is, the boundary conditions are of the form $b_i(u, u', \dots, u^{(d)}, \lambda) = \lambda(\sum_{i=1}^d c_i u^{(d)}) + (\sum_{i=1}^d \tilde{c}_i u^{(d)}) = 0$ where $c_i, \tilde{c}_i \in \mathbb{C}$. An example is $u^{(4)} = \lambda u''$ with boundary conditions $u(0) = u'(0) = 0, u''(1) = 0, u^{(3)}(1) - \lambda \gamma u'(1) = 0$ where $\gamma \in [0, 1]$. This problem arises in the study of critical loads for divergence of a clamped-free elastic bar; see [37] and the references therein.

While it is not impossible to solve eigenvalue problems with λ -dependent boundary conditions using classical methods, we shall show that it is straightforward to deal with such problems using our framework.

Flexibility: generalized eigenvalue problems. Our approach also makes it straightforward to treat *generalized* operator eigenvalue problems of the form

$$(1.3) \quad \mathcal{L}_A u = \lambda \mathcal{L}_B u,$$

where \mathcal{L}_A and \mathcal{L}_B are linear differential operators of orders d and p , respectively. Again the boundary conditions may be λ -dependent.

Miscellaneous. Our least-squares approach also allows for variants that can often be formulated in a straightforward fashion. As an example, we discuss a reformulation of the differential equation as an integral equation, which is known (as shown in Greengard's papers, e.g., [26]) to often reduce the condition number of the discretized problem and therefore provide more accurate solutions than standard approaches for differential equations. This idea was revisited by Driscoll using Chebfun [16]. We observe that the same holds also for our algorithm.

Disadvantages. Having described the advantages of our least-squares approach, it is important to also highlight the drawbacks of our approach. Most notably, it relies heavily on one's ability to work with and manipulate quasimatrices. Such operations are conveniently offered by Chebfun (and indeed the essence of our algorithms can be implemented in a few lines of MATLAB code), but otherwise (on other platforms) the algorithm can be nontrivial to implement (to emulate the method approximately, one can take many sample points in the domain and perform a least-squares fit [30]). In addition, for "easy" problems with smooth solutions, classical algorithms work exceptionally well, with exponential convergence and fast solution of the discretized linear algebra problem often available. Our approach will not be more efficient in such settings (e.g., when one takes U to be the same polynomial basis). Finally, (approximately) minimizing the residual does not mean the error is minimized. It is possible for the error to be larger than other methods.

This paper is organized as follows. In section 2 we describe our algorithm LSode for ODEs. In section 3 we explain our formulation for ODE eigenvalue problems. Section 4 describes the SVD of objects containing quasimatrices and matrices, which is needed for our eigenvalue algorithm. We then introduce our algorithm LSeig in section 5. In section 6 we discuss issues related to boundary conditions, namely imposing them exactly, and nonstandard conditions involving λ . Section 7 treats reformulations into an integral equation and its adaptation to LSeig. We present numerical experiments in section 8 to illustrate the performance of LSeig.

Notation. We denote by u', u'' the first and second derivatives of a function u , and more generally $u^{(d)}$ denotes the d th derivative. We use lowercase boldface letters for vectors, simple capital letters for matrices, and calligraphic letters for operators.

We denote quasimatrices by sans serif capital letters, e.g., \mathbf{A} . A (column) quasimatrix \mathbf{A} is an $\infty \times n$ "matrix" whose columns a_j are functions rather than discrete vectors. This means that the first index of a rectangular matrix becomes continuous while the second one remains discrete. Thus, \mathbf{A} represents a linear transformation

from \mathbb{C}^n to $L^2[a, b]$. We mainly treat problems on $[-1, 1]$ for concreteness, but the extension to $[a, b]$ is straightforward. The term quasimatrix was coined by Stewart [44, p. 33]. Quasimatrices are discussed by de Boor [15] and Trefethen and Bau [51, pp. 52–54]. They became part of computational practice with the introduction of Chebfun in 2004 [2, 3].

In what follows, $\mathbf{U} = [u_1 \dots, u_n] \in \mathbb{R}^{\infty \times n}$ denotes a quasimatrix [18] whose columns are the basis functions, whose choice is made a priori and often nonstandard and problem-dependent (one can take u_i to be polynomials but the strength of our approach is exhibited with other choices; we shall discuss several specific choices for specific problems).

We will make extensive use of decompositions of quasimatrices, in particular the QR factorization and SVD. For details on these see [47, 49] and [18, Chap. 6]. Almost all operations with matrices have counterparts for quasimatrices. For example, for a function u and a quasimatrix $\mathbf{Q} = [q_1, \dots, q_n] \in \mathbb{C}^{\infty \times n}$, we let $\mathbf{Q}^T f$ denote the $n \times 1$ vector whose i th element is $\langle q_i, f \rangle$. Moreover, a nonstandard (but straightforward) object that we need to deal with extensively is a “quasimatrix-matrix” of the form $\begin{bmatrix} \mathbf{U} \\ B \end{bmatrix}$, where $\mathbf{U} \in \mathbb{C}^{\infty \times n}$ is a quasimatrix and B is a $d \times n$ matrix. We say that such an object is of size⁵ “ $(\infty + d) \times n$.” These are an instance of block operators introduced in [1]. We use standard operations with such objects, including multiplication by a vector $\mathbf{v} \in \mathbb{C}^n$ as defied by the formula $\begin{bmatrix} \mathbf{U} \\ B \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{U} \mathbf{v} \\ B \mathbf{v} \end{bmatrix}$, which yields a “function-vector” object.

2. Least-squares formulation for ODE $\mathcal{L}u = f$. Our emphasis is on operator eigenvalue problems, but for a gentle introduction we first discuss solving differential equations $\mathcal{L}u = f$ with boundary conditions $b_i(u, u', \dots, u^{(d)}) = f_{b_i}$ for $i = 1, \dots, d$, where $f_{b_i} \in \mathbb{R}$ and b_i is a linear function of its arguments.

The main idea is to find $u \in U$ such that

$$(2.1) \quad \underset{u \in U}{\text{minimize}} \left\| \begin{bmatrix} \mathcal{L}u - f \\ b_1(u, \dots, u^{(d)}) - f_{b_1} \\ \vdots \\ b_d(u, \dots, u^{(d)}) - f_{b_d} \end{bmatrix} \right\|_2,$$

that is, the norm is minimized of the function-vector consisting of the residual of the operator and boundary conditions.

Basis functions satisfying the boundary conditions. For simplicity first consider the case where the basis functions u_j satisfy the boundary conditions $b_i(u_j, u'_j, \dots) = 0$. Then the bottom d elements are 0 and writing the solution $u = \sum_{i=1}^n c_i u_i$, (2.1) reduces to

$$\underset{u \in U}{\text{minimize}} \|\mathcal{L}u - f\|_2 = \underset{\mathbf{c} \in \mathbb{C}^n}{\text{minimize}} \|(\mathcal{L}\mathbf{U})\mathbf{c} - f\|_2.$$

This is a least-squares problem involving a quasimatrix. As with a matrix least-squares problem, it can be solved via the quasimatrix QR factorization⁶ $\mathcal{L}\mathbf{U} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q} \in \mathbb{C}^{\infty \times n}$ is orthonormal $\mathbf{Q}^T \mathbf{Q} = I_n$ (the (i, j) entry of $\mathbf{Q}^T \mathbf{Q}$ is $\langle q_i, q_j \rangle$) and $\mathbf{R} \in \mathbb{C}^{n \times n}$ is upper-triangular; once $\mathbf{Q}\mathbf{R}$ is obtained, the least-squares solution is given as $\mathbf{c} = \mathbf{R}^{-1} \mathbf{Q}^T f$. The formulation is mathematically (but not numerically) equivalent to Galerkin with the choice $\mathbf{V} := \mathcal{L}\mathbf{U}$.

⁵Perhaps it is more accurate to say they are of size $([-1, 1] + d) \times n$. We use $(\infty + d) \times n$ for brevity.

⁶This is a “thin” QR factorization. For quasimatrices it serves little purpose to think of the “full” QR, as the functions live in an infinite-dimensional space.

General basis functions. More generally, when the basis functions u_j do not satisfy the boundary conditions, we rewrite (2.1) as finding the minimizer for

$$(2.2) \quad \min_{\mathbf{c} \in \mathbb{C}^n} \left\| \begin{array}{c} (\mathcal{L}\mathbf{U})\mathbf{c} - f \\ b_1(\mathbf{U}\mathbf{c}, (\mathbf{U}\mathbf{c})', \dots, (\mathbf{U}\mathbf{c})^{(d)}) - f_{b_1} \\ \vdots \\ b_d(\mathbf{U}\mathbf{c}, (\mathbf{U}\mathbf{c})', \dots, (\mathbf{U}\mathbf{c})^{(d)}) - f_{b_d} \end{array} \right\|_2 = \min_{\mathbf{c} \in \mathbb{C}^n} \left\| \begin{bmatrix} \mathcal{L}\mathbf{U} \\ B \end{bmatrix} \mathbf{c} - \begin{bmatrix} f \\ \mathbf{f}_b \end{bmatrix} \right\|_2,$$

where $B \in \mathbb{C}^{d \times n}$ and $\mathbf{b} \in \mathbb{C}^d$.

The problem (2.2) is still solvable, once one has the QR factorization for the quasimatrix-matrix $\begin{bmatrix} \mathcal{L}\mathbf{U} \\ B \end{bmatrix}$, which can be obtained as follows. We first compute the quasimatrix QR factorization $\mathcal{L}\mathbf{U} = \mathbf{Q}_L R_L$ [49] and the standard matrix QR factorization $B = Q_B R_B$, so that

$$(2.3) \quad \begin{bmatrix} \mathcal{L}\mathbf{U} \\ B \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_L R_L \\ Q_B R_B \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_L & 0 \\ 0 & Q_B \end{bmatrix} \begin{bmatrix} R_L \\ R_B \end{bmatrix}.$$

Note that $\tilde{\mathbf{Q}} := \begin{bmatrix} \mathbf{Q}_L & 0 \\ 0 & Q_B \end{bmatrix} \in \mathbb{C}^{(\infty+d) \times 2n}$ is an orthonormal quasimatrix-matrix satisfying $\tilde{\mathbf{Q}}^T \tilde{\mathbf{Q}} = I_{2n}$. (For the quasimatrix-matrix $\tilde{\mathbf{Q}}$ we use the same sans serif font as quasimatrices.) One can complete the QR factorization via another thin QR $\begin{bmatrix} R_L \\ R_B \end{bmatrix} = \tilde{Q} R$, giving

$$(2.4) \quad \begin{bmatrix} \mathcal{L}\mathbf{U} \\ B \end{bmatrix} = \left(\begin{bmatrix} \mathbf{Q}_L & 0 \\ 0 & Q_B \end{bmatrix} \tilde{Q} \right) R = QR.$$

We summarize the algorithm below.

Algorithm 2.1 LSode: Least-squares method for $\mathcal{L}u = f$, and boundary conditions $b_i(\mathbf{U}\mathbf{c}, (\mathbf{U}\mathbf{c})', \dots, (\mathbf{U}\mathbf{c})^{(d)}) = f_{b_i}$ for $i = 1, \dots, d$. $\mathbf{U} := [u_1, \dots, u_n]$ are the user-defined basis functions.

- 1: Form the quasimatrix $\mathcal{L}\mathbf{U}$.
- 2: Represent boundary conditions as $B\mathbf{c} = \mathbf{f}_b$, where $\mathbf{c} = [c_1, \dots, c_n]^T, u = \sum c_i u_i$.
- 3: Compute QR factorization of the $(\infty + d) \times n$ quasimatrix-matrix (as in (2.4))

$$(2.5) \quad \begin{bmatrix} \mathcal{L}\mathbf{U} \\ B \end{bmatrix} = QR.$$

- 4: Obtain solution $u = \mathbf{U}\mathbf{c}$ where $\mathbf{c} = R^{-1}(Q^* \begin{bmatrix} f \\ \mathbf{f}_b \end{bmatrix})$.
-

The features of this least-squares formulation include the following:

1. The least-squares residual represents exactly the residual in the problem, including the boundary condition.
2. If desired, the boundary conditions can be imposed exactly (see section 6.1).
3. It has the flavor of spectral collocation methods wherein the collocation points are taken to vary continuously. It also has the Galerkin flavor, as the solution can be characterized equivalently as follows: the augmented residual $\begin{bmatrix} \mathcal{L}u - f \\ B\mathbf{c} - \mathbf{f}_b \end{bmatrix}$ is orthogonal to \mathbf{Q} .

These features carry over to the eigenvalue algorithm we develop in the next section.

Let us note, in passing, that there is an attractive class of finite element methods, called least-squares finite element methods (LSFEM), for the numerical solution of

PDEs. These methods minimize the residual in the differential equation together with the residual in the boundary conditions. These methods have been motivated by the desire to recover, in general settings, the advantageous features of Rayleigh–Ritz methods; see [8] and the references therein. LSode is in some sense a spectral analogue of LSFEM for ODEs.

2.1. Numerical illustration. To illustrate the advantages offered by the flexibility of basis functions in LSode, here we consider building the basis functions u_1, \dots, u_n by solving a nearby ODE (which might have been solved or is easier to solve).

Namely, consider the main problem of solving the ODE $\mathcal{L}u = f$, where $f(x) = \exp(x)$ and

$$(2.6) \quad \mathcal{L}u = 0.1u'' + |x|u' + u, \quad u(\pm 1) = 0.$$

We first solve an “auxiliary” ODE $\widehat{\mathcal{L}}u = f$ where $\widehat{\mathcal{L}}$ is related but not equal to \mathcal{L} . Here, we take

$$(2.7) \quad \widehat{\mathcal{L}}u = 0.1u'' + |x|u', \quad u(\pm 1) = 0,$$

and use its solutions as the basis functions u_i for solving the slightly different ODE $\mathcal{L}u = f$. Specifically, we used $\widehat{\mathcal{L}}$ (or its inverse) to form a Krylov subspace (informally, this is the subspace $\text{span}(\widehat{\mathcal{L}}^{-1}f, \widehat{\mathcal{L}}^{-2}f, \dots, \widehat{\mathcal{L}}^{-n}f)$, orthogonalized by the Arnoldi process; see [23] for more discussion on Krylov subspaces for differential operators). We then let $U = [u_1, \dots, u_n]$ be the orthonormal basis and invoke LSode. We compare its performance with the standard choice of u_i , global polynomials $u_i(x) = T_{i-1}(x)$. The results are shown in Figure 1.

Figure 1 demonstrates that by using specially crafted basis functions, one can obtain much faster convergence than standard bases. The flexibility offered by the least-squares framework allows us to easily explore nonstandard basis functions.

Note that since by definition LSode minimizes the residual (together with the boundary conditions), no standard collocation or coefficient method based on global polynomials can do better than the LSode with polynomials shown in the figure.

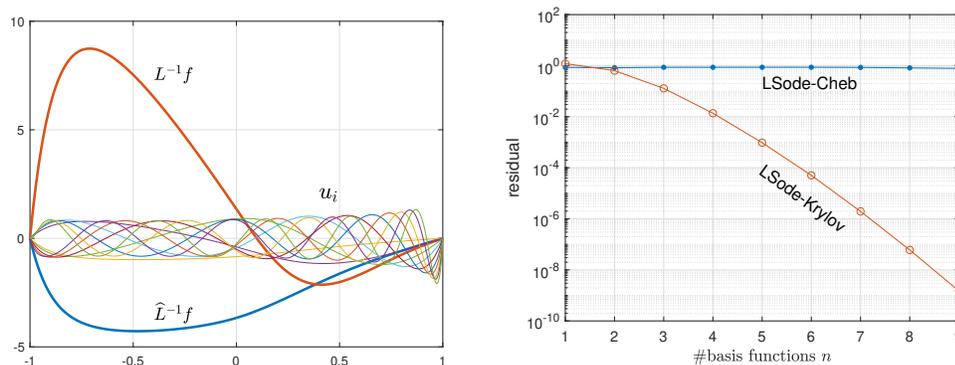


FIG. 1. Illustration of choice of basis functions for solving (2.6) with LSode: Chebyshev polynomials versus Krylov basis functions obtained by solving nearby ODEs (2.7). Left: solutions $\mathcal{L}^{-1}f, \widehat{\mathcal{L}}^{-1}f$ together with the basis functions u_i . Note that $\mathcal{L}^{-1}f$ and $\widehat{\mathcal{L}}^{-1}f$ are themselves not very similar, yet a Krylov basis w.r.t. $\widehat{\mathcal{L}}^{-1}$ offers a powerful subspace in which to find $\mathcal{L}^{-1}f$.

Here we considered problems with the $|x|$ term in (2.7), (2.6) so that they are not easily solvable by a classical approach. The idea of using solutions for nearby problems appears to work well in a variety of settings (a detailed study is left for future work). Clearly, once \mathbf{U} is obtained, one can consider solving a host of ODEs similar to (2.7) using the same \mathbf{U} ; this is much more efficient than solving each problem separately.

3. Least-squares formulation for $\mathcal{L}u = \lambda u$. Now we turn to the main topic of operator eigenvalue problems $\mathcal{L}u = \lambda u$, with boundary conditions $b_i(u, u', \dots, u^{(d)}) = 0$ for $i = 1, \dots, d$.

Again the core idea is to find $u \in U$ such that the operator and boundary condition residuals are minimized simultaneously:

$$(3.1) \quad \underset{u \in U}{\text{minimize}} \left\| \begin{array}{c} \mathcal{L}u - \lambda u \\ b_1(u, \dots, u^{(d)}, \lambda) \\ \vdots \\ b_d(u, \dots, u^{(d)}, \lambda) \end{array} \right\|_2.$$

Basis functions satisfying the boundary conditions. If b_i does not depend on λ and each of the basis functions u_j satisfy the boundary conditions, that is, $b_i(u_j, u'_j, \dots) = 0$ for $i = 1, \dots, d$, then (3.1) reduces to (writing as before the solution $u = \sum_{i=1}^n c_i u_i$)

$$\min_{u \in U} \|\mathcal{L}u - \lambda u\|_2 = \min_{\mathbf{c} \in \mathbb{C}^n} \|(\mathcal{L}\mathbf{U})\mathbf{c} - \lambda \mathbf{U}\mathbf{c}\|_2.$$

This is a least-squares eigenvalue problem involving quasimatrices.

At first sight this is a difficult problem: supposing there is a solution with zero residual $(\mathcal{L}\mathbf{U})\mathbf{c} = \lambda \mathbf{U}\mathbf{c}$, it is an $\infty \times n$ generalized eigenvalue problem. A matrix analogue would be an $m \times n$ ($m > n$) *rectangular* eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$ for $A, B \in \mathbb{C}^{m \times n}$ [56]. Such problems are not classically treated in numerical linear algebra, and they usually do not have solutions satisfying the equation exactly. Nonetheless, several practical methods for rectangular eigenvalue problems have been proposed, including [14, 34, 35, 36].

In this paper we focus on the algorithm by Ito and Murota [33], which is easy to implement and finds solutions that are optimal in a certain sense (it finds the smallest perturbations of A, B such that the equation has n solutions). The Ito–Murota algorithm relies on an SVD to reduce the problem to a square $n \times n$ generalized eigenvalue problem. We review the algorithm in section 5.1.

Fortunately, this algorithm can be generalized readily to quasimatrices, and then in turn to quasimatrix-matrix objects. The crucial tool is the SVD of such objects, which we discuss in section 4.

General basis functions. When a boundary condition b_i depends on λ or the basis functions u_j do not satisfy the boundary conditions $b_i(u_j, u'_j, \dots) \neq 0$, the goal is to find the minimizer for

$$(3.2) \quad \min_{\mathbf{c} \in \mathbb{C}^n} \left\| \begin{array}{c} (\mathcal{L}\mathbf{U})\mathbf{c} - \lambda \mathbf{U}\mathbf{c} \\ b_1(\mathbf{U}\mathbf{c}, (\mathbf{U}\mathbf{c})', \dots, (\mathbf{U}\mathbf{c})^{(d)}) \\ \vdots \\ b_d(\mathbf{U}\mathbf{c}, (\mathbf{U}\mathbf{c})', \dots, (\mathbf{U}\mathbf{c})^{(d)}) \end{array} \right\|_2 = \min_{\mathbf{c} \in \mathbb{C}^n} \left\| \begin{bmatrix} \mathcal{L}\mathbf{U} \\ B \end{bmatrix} \mathbf{c} - \lambda \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix} \mathbf{c} \right\|_2,$$

where $B \in \mathbb{C}^{d \times (n+1)}$. This is a rectangular eigenvalue problem for quasimatrix-matrix objects. We show that this can also be solved using the Ito–Murota method, via the SVD for quasimatrix-matrix objects. We now turn to computing such an SVD.

4. SVD of objects involving quasimatrices and matrices. In this section we explain how to compute the SVD of different objects required in our least-squares framework for operator eigenvalue problems. We first review the simplest case of computing the SVD of a quasimatrix and then proceed with more involved objects.

Let A be an $\infty \times n$ quasimatrix. Following [49], the SVD of A can be computed in two steps. We first compute the QR decomposition $A = Q_A R_A$, where Q_A is an $\infty \times n$ quasimatrix with orthogonal columns and R_A is an $n \times n$ upper-triangular matrix. We then proceed with computing the SVD of

$$(4.1) \quad R_A = U_R \Sigma V^*$$

resulting in the SVD

$$A = \underbrace{(Q_A U_R)}_U \Sigma V^*,$$

where U is an $\infty \times n$ quasimatrix with orthogonal columns, Σ is $n \times n$ diagonal with singular values arranged in nonincreasing order, and V is an $n \times n$ unitary matrix.

We will also need the SVD of the horizontal concatenation of two quasimatrices. Let $[A \ B]$ be an $\infty \times 2n$ quasimatrix. Analogously to the previous case, we first compute the QR decomposition $[A \ B] = QR$ where Q is an $\infty \times 2n$ quasimatrix with orthogonal columns and then compute the SVD of the $2n \times 2n$ upper-triangular matrix $R = U_R \Sigma V^*$. Partitioning we have

$$U_R = \begin{bmatrix} \underbrace{U_{R1}}_{2n \times n} & \underbrace{U_{R2}}_{2n \times n} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{1:n} & \\ & \Sigma_{n+1:2n} \end{bmatrix}, \quad V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}.$$

Defining $U_1 = QU_{R1}$ and $U_2 = QU_{R2}$, note that these $\infty \times n$ quasimatrices have orthogonal columns. Thus we obtain the SVD

$$(4.2) \quad [A \ B] = U \Sigma V^* = \underbrace{[U_1 \ U_2]}_U \begin{bmatrix} \Sigma_{1:n} & \\ & \Sigma_{n+1:2n} \end{bmatrix} \begin{bmatrix} V_{11}^* & V_{21}^* \\ V_{12}^* & V_{22}^* \end{bmatrix}.$$

Next, we discuss the computation of the SVD of a quasimatrix-matrix object which is the vertical concatenation of an $\infty \times n$ quasimatrix A and a $d \times n$ matrix C . Starting from two QR factorizations we have

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} Q_A R_A \\ Q_C R_C \end{bmatrix} = \begin{bmatrix} Q_A & 0 \\ 0 & Q_C \end{bmatrix} \begin{bmatrix} R_A \\ R_C \end{bmatrix}.$$

We then compute the thin SVD of the $(n+d) \times n$ matrix

$$(4.3) \quad \begin{bmatrix} R_A \\ R_C \end{bmatrix} = U_R \Sigma V^*,$$

resulting in the SVD

$$(4.4) \quad \begin{bmatrix} A \\ C \end{bmatrix} = \underbrace{\begin{bmatrix} Q_A & 0 \\ 0 & Q_C \end{bmatrix} U_R}_U \Sigma V^*.$$

Here, U is an $(\infty+d) \times n$ quasimatrix-matrix. It can be easily checked that $U^*U = I_n$ and $V^*V = I_n$.

Finally, consider objects of the form $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and B are $\infty \times n$ quasimatrices and C, D are discrete matrices of the size $d \times n$. We first compute the two QR decompositions

$$\underbrace{\begin{bmatrix} A & B \end{bmatrix}}_{\infty \times 2n} = \underbrace{Q}_{\infty \times 2n} \underbrace{R}_{2n \times 2n}, \quad \underbrace{\begin{bmatrix} C & D \end{bmatrix}}_{d \times 2n} = \underbrace{\tilde{Q}}_{d \times d} \underbrace{\tilde{R}}_{d \times 2n},$$

and then compute the matrix thin SVD

$$\underbrace{\begin{bmatrix} R \\ \tilde{R} \end{bmatrix}}_{(2n+d) \times 2n} = \underbrace{U_R}_{(2n+d) \times 2n} \underbrace{\Sigma}_{2n \times 2n} \underbrace{V^*}_{2n \times 2n}.$$

Thus we obtain the following SVD:

$$(4.5) \quad \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} Q & 0_{\infty \times d} \\ 0_{d \times 2n} & \tilde{Q} \end{bmatrix} \begin{bmatrix} R \\ \tilde{R} \end{bmatrix} = \underbrace{\left(\begin{bmatrix} Q & 0 \\ 0 & \tilde{Q} \end{bmatrix} U_R \right)}_{(\infty+d) \times 2n} \Sigma V^*.$$

As we can see, these computations crucially rely on the QR factorization of quasimatrices, which can be obtained in different ways; a numerically stable method is to use a continuous analogue of Householder reflections [49]. Computing the SVD of these objects then becomes relatively straightforward and efficient.

5. Least-squares rectangular eigenvalue solver. Having discussed how to compute the SVD of quasimatrix-matrix objects, we are now in a position to introduce our algorithm LSeig for ODE eigenvalue problems, based on solving (3.2). As discussed above, the key is to solve a quasimatrix rectangular eigenvalue problem. We start by reviewing the Ito–Murota algorithm for the discrete case.

5.1. Discrete case: Ito–Murota algorithm. Consider an $m \times n$ ($m > n$) rectangular eigenvalue problem

$$A\mathbf{x} = \lambda B\mathbf{x}.$$

As noted previously, there is usually no exact solution. Furthermore, the number of local minima for $h(\lambda) = \sigma_{\min}(A - \lambda B)$ is unknown. To tackle these difficulties, Ito and Murota adopt a perturbation approach via total least-squares: minimize $\|[\Delta A \ \Delta B]\|_F$, which can be seen as the backward error, such that n exact eigenpairs (λ_i, x_i) exist for the perturbed problem,

$$(5.1) \quad (A + \Delta A)X = (B + \Delta B)X \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix},$$

and the vectors x_i are linearly independent. Here, only A, B are given and the rest are unknowns. Our goal is to find X and λ_i 's, with the minimal $\|[\Delta A \ \Delta B]\|_F$. They show that this can be accomplished as follows: Take the SVD

$$\begin{bmatrix} A & B \end{bmatrix} = U \Sigma V^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_{1:n} & \\ & \Sigma_{n+1:2n} \end{bmatrix} \begin{bmatrix} V_{11}^* & V_{21}^* \\ V_{12}^* & V_{22}^* \end{bmatrix},$$

then solve the square generalized eigenvalue problem $V_{11}^* X = V_{21}^* X \Lambda$ by the standard QZ algorithm. With these X and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, we have (5.1) with $\|[\Delta A \ \Delta B]\|_F = \sqrt{\sum_{i=1}^n (\sigma_{n+i}([A \ B]))^2}$. As explained by Ito and Murota [33], their algorithm is equivalent to solving $(U_1^* A)X = (U_1^* B)X \Lambda$, that is, it is a projection onto the subspace spanned by the n leading left singular vectors of the two matrices $[A, B]$, and hence the residuals $A\mathbf{x} - \lambda B\mathbf{x}$ of the outputs are orthogonal to U_1 .

5.2. Continuous case. We rewrite $\mathbf{A} = \begin{bmatrix} \mathcal{L}\mathbf{U} \\ \mathbf{B} \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix}$ in (3.2) and consider the quasimatrix-matrix rectangular generalized eigenvalue problem

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}, \quad \mathbf{A}, \mathbf{B} \in \mathbb{C}^{(\infty+d) \times n}.$$

As in Ito–Murota, we attempt to solve for the minimal $\|[\Delta\mathbf{A} \ \Delta\mathbf{B}]\|_F$ such that

$$(5.2) \quad (\mathbf{A} + \Delta\mathbf{A})\mathbf{X} = (\mathbf{B} + \Delta\mathbf{B})\mathbf{X} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}.$$

The direct analogue of Ito–Murota is to use the SVD (for $[\mathbf{A} \ \mathbf{B}] = \begin{bmatrix} \text{quasimatrix} & \text{quasimatrix} \\ \text{matrix} & \text{matrix} \end{bmatrix}$, as described in (4.5))

$$[\mathbf{A} \ \mathbf{B}] = \mathbf{U}\Sigma\mathbf{V}^* = [\mathbf{U}_1 \ \mathbf{U}_2] \begin{bmatrix} \Sigma_{1:n} & \\ & \Sigma_{n+1:2n} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{11}^* & \mathbf{V}_{21}^* \\ \mathbf{V}_{12}^* & \mathbf{V}_{22}^* \end{bmatrix},$$

where \mathbf{U}_1 is a quasimatrix-matrix of size $(\infty + d) \times n$, and then reduce the problem to the fully discrete, $n \times n$ generalized eigenvalue problem

$$(\mathbf{U}_1^*\mathbf{A})\mathbf{X} = (\mathbf{U}_1^*\mathbf{B})\mathbf{X}\Lambda.$$

In the supplementary materials (supplement.pdf [local/web 629KB]) we show that this actually minimizes $\|\Delta\mathbf{A} \ \Delta\mathbf{B}\|_F$.

This completes the algorithm for solving (1.1). An extension to the generalized problem (1.3) is straightforward: all we need is to allow for the second operator \mathcal{L}_B to act on the basis.

Finally, as mentioned in the introduction, our approach can be adapted easily to boundary conditions that depend affinely on λ . To do this we represent the boundary conditions as follows: find $B_A, B_B \in \mathbb{C}^{d \times n}$ such that $b_i(\mathbf{U}\mathbf{c}, (\mathbf{U}\mathbf{c})', \dots, (\mathbf{U}\mathbf{c})^{(d)}, \lambda) = 0$ is equivalent to $(B_A - \lambda B_B)[c_1, \dots, c_n]^T = 0$, where $u = \sum c_i u_i$. Putting these together, for the general problem (1.3) with λ -dependent boundary conditions we solve instead of (3.2)

$$\underset{\mathbf{c} \in \mathbb{C}^n}{\text{minimize}} \left\| \begin{bmatrix} \mathcal{L}_A \mathbf{U} \\ B_A \end{bmatrix} \mathbf{c} - \lambda \begin{bmatrix} \mathcal{L}_B \mathbf{U} \\ B_B \end{bmatrix} \mathbf{c} \right\|_2.$$

We now summarize the algorithm.

Remarks.

- The eigenvalue problem (5.4) gives n eigenpairs, and hence LSeig outputs n solutions $(\lambda_i, \mathbf{U}\mathbf{c}_i)$ for $i = 1, \dots, n$. Among these, we are often interested in a specific (e.g., largest) eigenpair. In any case, since usually only some of the computed eigenpairs are accurate, we recommend examining the residuals $\|\mathcal{L}_A \mathbf{U}\mathbf{c}_i - \lambda_i \mathcal{L}_B \mathbf{U}\mathbf{c}_i\|_2$ and accepting those that are small enough; see (8.1).
- No condition is imposed on the basis functions in \mathbf{U} ; they can be chosen arbitrarily. Hence, standard choices such as Chebyshev or Fourier (for periodic problems) are applicable, but in addition, other basis functions are equally applicable. We explore nonstandard choices in our forthcoming experiments.
- As with LSode, LSeig has a flavor of both collocation and Galerkin: the residual $\begin{bmatrix} \mathcal{L}_A u - \lambda \mathcal{L}_B u \\ (B_A - \lambda B_B)\mathbf{c} \end{bmatrix}$ is orthogonal to \mathbf{U}_1 .
- The dominant computational cost is in computing the SVD and solving the generalized eigenvalue problem. Computing the SVD in Chebfun requires roughly $O(kn^2)$ operations, where k is the maximum degree of u_i , and the generalized eigenproblem requires $O(n^3)$ operations, with a total cost of $O((n+k)n^2)$.

Algorithm 5.1 LSeig: Least-squares method for $\mathcal{L}_A u = \lambda \mathcal{L}_B u$, and boundary conditions $b_i(\mathbf{Uc}, (\mathbf{Uc})', \dots, (\mathbf{Uc})^{(d)}, \lambda) = 0$. $\mathbf{U} := [u_1, \dots, u_n]$ are the user-defined basis functions.

- 1: Form quasimatrices $\mathbf{A} = \mathcal{L}_A \mathbf{U}, \mathbf{B} = \mathcal{L}_B \mathbf{U}$.
- 2: Represent boundary conditions as $(B_A - \lambda B_B)\mathbf{c} = 0$, where $u = \mathbf{Uc}$.
- 3: Compute SVD of $(\infty + d) \times 2n$ quasimatrix-matrix (as in (4.5))

$$(5.3) \quad \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ B_A & B_B \end{bmatrix} = [\mathbf{U}_1 \ \mathbf{U}_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} V.$$

- 4: Solve the $n \times n$ discrete eigenvalue problem

$$(5.4) \quad \left(\mathbf{U}_1^* \begin{bmatrix} \mathbf{A} \\ B_A \end{bmatrix} \right) X = \left(\mathbf{U}_1^* \begin{bmatrix} \mathbf{B} \\ B_B \end{bmatrix} \right) X \Lambda.$$

- 5: Output $(\lambda_i, \mathbf{Uc}_i)$ for $i = 1, \dots, n$ (or its subset) as eigenpairs, where $X = [\mathbf{c}_1, \dots, \mathbf{c}_n] \in \mathbb{C}^{n \times n}$ and $\Lambda = \text{diag}(\lambda_i)$.
-

- We can obtain straightforward variants by introducing weighting to specific rows of (5.3) to give more weight to, e.g., the boundary conditions (if enforcing them tightly is important), and scaling the columns (corresponding to diagonal scaling, as employed in the MATLAB `eig` command).

6. Boundary conditions. Here we discuss various aspects of boundary conditions. We discuss a variant of our algorithm where the boundary conditions are imposed exactly (instead of being minimized together with the residual). We then discuss boundary conditions that arise specifically in eigenvalue problems, namely those that depend on λ .

6.1. Imposing boundary conditions exactly. We can modify Algorithm 5.1 to impose boundary conditions exactly by restricting the projection space to contain the parts representing the boundary conditions. Conceptually, this is done by introducing an infinite weight in the final d rows of (5.3). This forces the leading d left singular vectors to span $\begin{bmatrix} 0_{\infty \times d} \\ I_d \end{bmatrix}$. We then choose the remaining $(n - d)$ -dimensional part by the SVD of the remaining part, that is, the quasimatrix $[\mathbf{A} \ \mathbf{B}]$.

Namely, steps 3 and 4 in Algorithm 5.1 become the following:

- Compute the SVD of the $\infty \times 2n$ quasimatrix

$$[\mathbf{A} \ \mathbf{B}] = [\mathbf{U}_1 \ \mathbf{U}_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} V, \quad \mathbf{U}_1 \in \mathbb{C}^{\infty \times (n-d)}.$$

- Set $\mathbf{U} := \begin{bmatrix} \mathbf{U}_1 & 0_{\infty \times d} \\ 0_{d \times (n-d)} & I_d \end{bmatrix}$ and solve the $n \times n$ square eigenvalue problem

$$\left(\mathbf{U}^* \begin{bmatrix} \mathbf{A} \\ B_A \end{bmatrix} \right) X = \left(\mathbf{U}^* \begin{bmatrix} \mathbf{B} \\ B_B \end{bmatrix} \right) X \Lambda.$$

We refer to this algorithm as LSeig-bc.

For ODEs, we can similarly modify Algorithm 2.1 as follows: take the leading $n - d$ left singular functions of $\mathcal{L}\mathbf{U}$, call it \mathbf{U}_1 , define $\mathbf{Q} = \begin{bmatrix} \mathbf{U}_1 & 0 \\ 0 & I_d \end{bmatrix}$, and find c by imposing $\mathbf{Q}^*([\mathcal{L}_B^U \mathbf{c} - \begin{bmatrix} f \\ \mathbf{f}_b \end{bmatrix}]) = 0$.

6.2. λ -dependent boundary conditions. ODE eigenvalue problems involving boundary conditions that depend on the unknown spectral parameter λ are an interesting class of nonstandard eigenvalue problems. Such problems arise, in particular, in elasticity and hydrodynamics [21, 37]. One example is the Orr–Sommerfeld equation for a liquid film flowing over an inclined plane, with a surface tension gradient which involves boundary conditions that depend linearly on λ ; see [22, 25]. Other problems with λ -dependent boundary conditions can be found in [28, 29, 37].

In addition, the development of a Sturm theory (ordering of eigenvalues, studying oscillation of eigenfunctions, etc.) for eigenvalue problems of second-order ODEs with λ -dependent boundary conditions has been studied in [32, 5, 12].

Among the earliest references on nonclassical boundary conditions is Birkhoff [6, 7]. Tamarkin [45, 46] considered problems where the differential equation and the boundary conditions depend polynomially on λ . For more historical notes on λ -dependent boundary conditions, see [55].

From the algorithmic point of view it is known that imposing boundary conditions that depend on λ is challenging for spectral collocation methods and the tau method is considered the method of choice; see [21, p. 112]. As seen above, it is straightforward for our least-squares framework to deal with boundary conditions that depend affinely⁷ on λ . Other techniques include the regularized sampling method (RSM) with which we compare the performance of our framework in Example 5.

7. Integral equation reformulation. The differentiation operator is unbounded, while integration is compact. Reformulating a differential equation as integral equations can thus sometimes improve the conditioning of the discretized equation [26], [16].

As previously mentioned, one of the strengths of our framework is its flexibility. We can readily adapt our methods to integral equations. In Example 4 we illustrate this for a differential eigenvalue problem of order four, where higher accuracy is obtained this way. For the sake of definiteness, here let us consider the advection-diffusion eigenvalue problem on the interval $[a, b]$

$$u'' + u' = \lambda u, \quad u(a) = u(b) = 0.$$

In the integral reformulation we take $v(x) := u''(x)$ as the unknown. Then, $u'(x) = \alpha + \int_a^x v(\eta) d\eta$, $u(x) = \alpha x + \beta + \int_a^x \int_a^t v(\eta) d\eta dt$ and the problem becomes

$$v(x) + \alpha + \int_a^x v(\eta) d\eta = \lambda \left(\alpha x + \beta + \int_a^x \int_a^t v(\eta) d\eta dt \right)$$

with boundary conditions

$$(7.1) \quad \alpha a + \beta = 0, \quad \alpha b + \beta + \int_a^b \int_a^t v(\eta) d\eta dt = 0.$$

We can formulate these in a least-squares eigenvalue problem. To do so we write $v(x) = \sum_{i=0}^{n-1} c_i T_i(x)$, and on the left-hand side we take the basis to be $[T_0, \dots, T_{n-1}, 1, 0]$ (note that this quasimatrix is rank-deficient). The right-hand side has basis $[T_0, \dots, T_{n-1}, x, 1]$ and the eigenvector is $[c_0, c_1, \dots, c_{n-1}, \alpha, \beta]$. We used monomials to represent constant terms resulting from each integration but, as in Example 4, the Chebyshev basis could also be used. To summarize, LSeig takes the two input quasimatrices

⁷It is, however, not straightforward with boundary conditions that depend *polynomially* on λ . We leave this for future work.

$A = [\mathcal{I}_A Q, 1, 0]$ and $B = [\mathcal{I}_B Q, x, 1]$ where $\mathcal{I}_A v(x) := v(x) + \int_a^x v(\eta) d\eta$ and $\mathcal{I}_B v(x) := \int_a^x \int_a^t v(\eta) d\eta dt$, together with the boundary conditions (7.1). The rest is the same as in Algorithm 5.1. We illustrate this idea for the Orr–Sommerfeld problem in Example 4.

8. Experiments. This section reports the performance of our least-squares framework for various problems, focusing on the flexibility of our approach and the accuracy of the computed eigenpairs. All numerical experiments were done using Chebfun on a standard laptop. Inputs to our LSeig and LSeig-bc algorithms include quasi-matrices A and B together with matrices representing boundary conditions. Another input is the tolerance `tol` on the following relative residual ensuring that we only output eigenpairs satisfying

$$(8.1) \quad \frac{\|AV - BV\Lambda\|_2}{\|AV\|_2} < \text{tol}.$$

We report `tol` in each experiment. When we report residuals with LSeig we report the augmented residual that includes the residual of the boundary conditions. In most of our experiments, LSeig-bc tended to give slightly more accurate results.

Example 1. Tailor-made basis functions. The flexibility of our approach is advantageous especially when the solutions are complicated (e.g., nonsmooth), so that classical approaches based on global polynomials are inefficient. When the solution is nonsmooth, for example, oscillatory or has singularities (e.g., of absolute-value type), it is much desirable to incorporate tailor-made basis functions.

To illustrate this, we consider the Airy eigenvalue problem

$$(8.2) \quad \mathcal{L}u := \epsilon u'' + xu = \lambda u, \quad u(\pm 1) = 0.$$

Here we target the smallest eigenpair. When $|\epsilon| \ll 1$, the eigenfunctions are highly oscillatory (see the right panel of Figure 2), and representing them accurately with polynomials would require a very high degree, making the computation inefficient.

To overcome this difficulty, one can consider using subspace methods that successively build a subspace that is designed to be rich in the desired eigenspace. Examples for matrix eigenvalue problems include Krylov methods and their variants

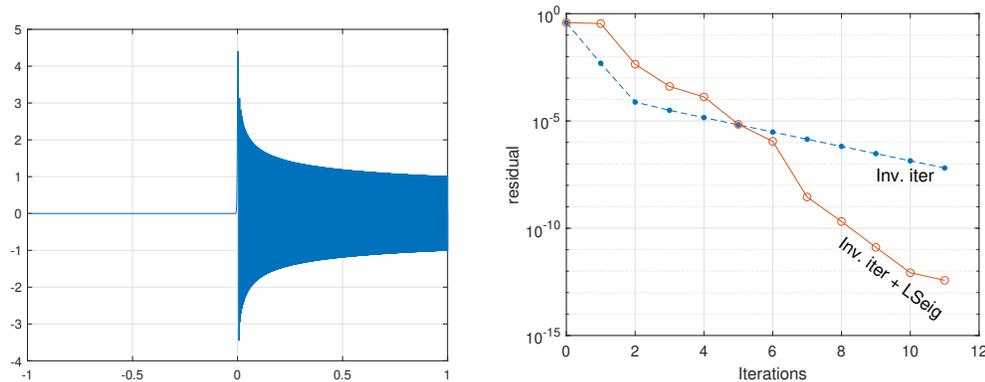


FIG. 2. Solving the Airy eigenvalue problem (8.2) with $\epsilon = 10^{-8}$, using (i) inverse iteration (*Inv. iter*) and (ii) LSeig using the subspace obtained by inverse iteration. Right: Solution \hat{u} . Left: Residual history $\|\mathcal{L}\hat{u} - \lambda\hat{u}\|_2$.

(shift-and-invert, rational) the Rayleigh quotient iteration, Jacobi–Davidson [43], and the generalized Davidson method, each differing in how the subspace is computed.

Here we illustrate a subspace built via the inverse iteration: given an approximate eigenfunction \tilde{u} , we solve the ODE

$$(8.3) \quad \mathcal{L}u = \tilde{u}$$

for u , update $u := \frac{\tilde{u}}{\|\tilde{u}\|_2}$, and repeat until convergence. To solve the ODE (8.3) we used the Olver–Townsend algorithm [38], which is also available in Chebfun.⁸ We compare inverse iteration against LSeig using the subspace generated by the iterates \tilde{u} , that is, $\text{span}(\mathbf{U}) = \text{span}(u_0, \mathcal{L}^{-1}u_0, \mathcal{L}^{-2}u_0, \dots, \mathcal{L}^{-(n-1)}u_0)$, the Krylov subspace with respect to the inverse \mathcal{L}^{-1} as in section 2.1. Here u_0 is the initial approximation, which we take to be $u_0(x) = (x-1)(x+1)$; unlike in section 2.1, we start the subspace from u_0 as it satisfies the boundary conditions.

The results for $\epsilon = 10^{-8}$ are shown in Figure 2. We see that (i) both inverse iteration and LSeig together with the inverse iteration Krylov subspace are effective algorithms for the problem (both are dramatically better than global polynomials, which would need $n \gg 10^3$), and (ii) LSeig outperforms inverse iteration by finding “the best” solution in the subspace; this difference is analogous to that between the standard power method and the Lanczos method for symmetric eigenvalue problems.

We note that for $\epsilon < 10^{-6}$, the solution becomes too oscillatory for Chebfun’s command `eigs` based on a collocation method to converge. When $\epsilon < 10^{-9}$ we further observed that inverse iteration fails to converge, as the ODE (8.3) becomes increasingly difficult to solve. Using the same subspace, LSeig nonetheless obtained solutions with residual smaller than 10^{-10} . LSeig-bc (imposing boundary conditions exactly) led to similar results. Here and in the next example the `tol` value in LSeig was selected to 1 so that all quantities are initially output, and we chose the eigenpair with the smallest residual.

Example 2. Problems with singularities. We next consider Example 20 in [52, App. B], a one-dimensional Schrödinger equation

$$(8.4) \quad \mathcal{L}u = -h^2u'' + V(x)u = \lambda u, \quad V(x) = |x|,$$

with $h = 0.1$ and boundary conditions $u(\pm 3) = 0$ (the domain is $[-3, 3]$). Here the potential $V(x)$ has a singularity at $x = 0$, and this forces the eigenfunctions to also have a (weaker but genuine) singularity: they are twice differentiable but not more.

The nonsmoothness of u presents difficulties for global polynomial-based methods, since as is well known, polynomials cannot approximate such functions efficiently (the convergence is $O(1/n)$ or slower, where n is the degree).

To remedy this, one can employ locally supported, piecewise polynomial basis functions as follows:

$$(8.5) \quad u_{2k-1}(x) = \begin{cases} T_k(2(x + \frac{1}{2})), & x < 0, \\ 0, & x \geq 0, \end{cases} \quad u_{2k}(x) = \begin{cases} 0, & x < 0, \\ T_k(2(x - \frac{1}{2})), & x \geq 0. \end{cases}$$

We use LSeig with these basis functions, imposing continuity of the solution and its first and second derivatives at the splitting point $x = 0$; this can be done by adding three more boundary conditions in the least-squares formulation. We compare the algorithm with LSeig using the standard global polynomials $u_i = T_i(x)$. The results are shown in Figure 3, which illustrates the advantages of using local piecewise polynomials when the location and nature of the singularity is known.

⁸Using the flag `options.discretization = 'coeffs'`.

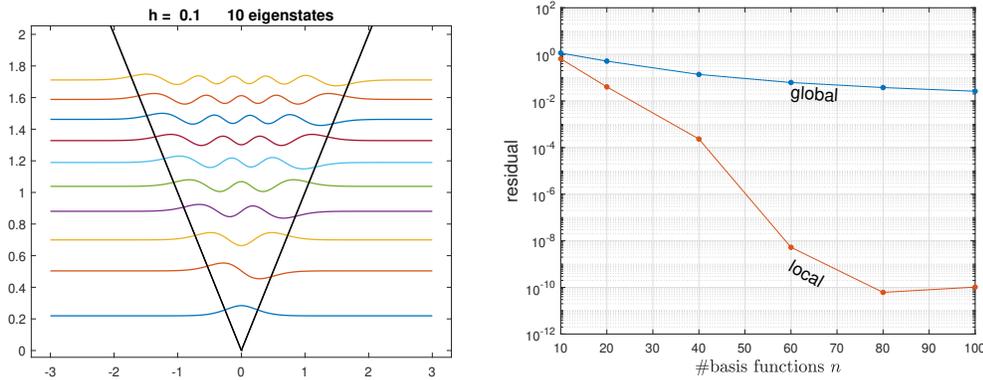


FIG. 3. Eigenfunctions (left; reprint of [52, App. B, Ex. 20]) and convergence to smallest eigenfunction for (8.4) using LSeig, with global and local (8.5) polynomial basis functions.

It is worth noting that Chebfun’s algorithm can also deal with such problems efficiently, by detecting singularities and working with piecewise polynomials [40].

Example 3. Generalized eigenproblems. Consider the Sturm–Liouville problem

$$(e^{3x}u'(x))' + 2e^{3x}u(x) + \lambda e^{3x}u(x) = 0, \quad u(0) = u(1) = 0.$$

Starting from 100 Chebyshev polynomials and $\text{tol} = 10^{-10}$, the LSeig method with exact boundary conditions obtains 41 eigenvalues in 0.14 seconds. The Chebfun `eigs` command takes 0.74 seconds when asked to compute the same number of eigenvalues.⁹ Relative residuals (8.1) are depicted in Figure 4. We also report the average number of correct digits d by looking at the logarithm of the geometric mean of the relative residuals. Note that the eigenfunctions of this Sturm–Liouville equation should be orthogonal with respect to the weight e^{3x} ; see [54, Chap. 13], for instance. The departures from orthonormality in the computed (normalized) eigenfunctions are similar: 2.1×10^{-8} for LSeig-bc and 1.8×10^{-8} for Chebfun `eigs`.

Example 4. Basis recombination and integral reformulation. Let us consider the famous Orr–Sommerfeld equation

$$(8.6) \quad \frac{1}{R}(u'''' - 2u'' + u) - 2iu - i(1 - x^2)(u'' - u) = \lambda(u'' - u),$$

with boundary conditions $u(\pm 1) = u'(\pm 1) = 0$, which arises in determining conditions for hydrodynamic stability. See [42], for instance. Here, R denotes the Reynolds number, which corresponds roughly to velocity divided by viscosity. We take $R = 5772$, which is about the critical value at which an eigenvalue first crosses into the right half-plane, corresponding to an unstable flow.

We write (8.6) as $\mathcal{L}_A u = \lambda \mathcal{L}_B u$ and apply three methods within our least-squares framework. First of all, we apply LSeig-bc directly to (8.6). The second method applies LSeig with basis recombination using the following basis functions satisfying the basis functions:

⁹We are not claiming LSeig is a faster algorithm—Chebfun’s method finds an appropriate degree adaptively, whereas in LSeig the degree is an input. The point here is that the speed of LSeig is comparable.

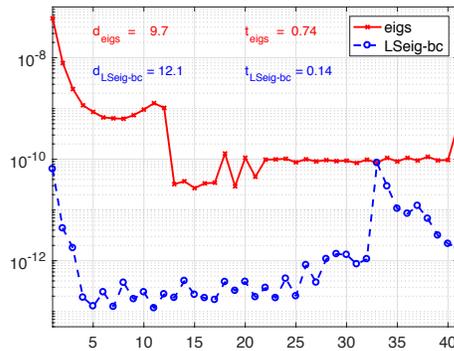


FIG. 4. Relative residuals (8.1) in 41 computed eigenvalues of the Sturm–Liouville problem of Example 3.

$$(1 + x)^2(1 - x)^2T_i(x), \quad i = 0, 1, \dots, n - 1.$$

We also consider an integral reformulation of (8.6) solving for $v := u''''$, with eigenvector $[c_0, c_1 \dots, c_{n-1}, a_0, a_1, a_2, a_3]$, where

$$(8.7) \quad u(x) = \int_{-1}^x \int_{-1}^w \int_{-1}^s \int_{-1}^t v(\eta) \, d\eta \, dw \, ds \, dt + p(x),$$

$v(x) = \sum_{i=0}^{n-1} a_i T_i(x)$, and $p(x)$, which is a polynomial of degree three resulting from integrations, is represented with $\sum_{i=0}^3 a_i T_i(x)$. In addition,

$$u''(x) = \int_{-1}^x \int_{-1}^w v(\eta) \, d\eta \, dw + p''(x).$$

Thus, we can write $v = Qc$ where $Q := [T_0, \dots, T_{n-1}]$ and expand u in terms of the columns of the basis quasimatrix $[Q, T_0, T_1, T_2, T_3]$. We rewrite the Orr–Sommerfeld equation as $\mathcal{I}_A v + \mathcal{L}_A p = \lambda(\mathcal{I}_B v + \mathcal{L}_B p)$, where

$$\begin{aligned} \mathcal{I}_A v(x) &:= \frac{1}{R} v(x) - \left(\frac{2}{R} + i(1 - x^2) \right) \int_{-1}^x \int_{-1}^w v(\eta) \, d\eta \, dw \\ &\quad + \left(\frac{1}{R} - 2i + i(1 - x^2) \right) \int_{-1}^x \int_{-1}^w \int_{-1}^s \int_{-1}^t v(\eta) \, d\eta \, dw \, ds \, dt, \\ \mathcal{I}_B v(x) &:= \int_{-1}^x \int_{-1}^w v(\eta) \, d\eta \, dw - \int_{-1}^x \int_{-1}^w \int_{-1}^s \int_{-1}^t v(\eta) \, d\eta \, dw \, ds \, dt, \end{aligned}$$

$\mathcal{L}_A p(x) := \sum_{i=0}^3 a_i q_i(x)$, and $\mathcal{L}_B p(x) := \sum_{i=0}^3 a_i \tilde{q}_i(x)$. Therefore, we apply our LSeig-bc algorithm to the quasimatrices $A := [\mathcal{I}_A Q, q_0, q_1, q_2, q_3]$ and $B := [\mathcal{I}_B Q, \tilde{q}_0, \tilde{q}_1, \tilde{q}_2, \tilde{q}_3]$ where $q_i = \mathcal{L}_A T_i$ and $\tilde{q}_i = \mathcal{L}_B T_i$ for $i = 0, 1, 2, 3$. Also, LSeig takes two matrices B_A and B_B of size $d \times (n + d)$ where each row represents one of the $d = 4$ boundary conditions. This means that $B_B = 0$ as boundary conditions do not depend on λ . In addition, having (8.7) in mind, for instance, the row of B_A corresponding with the boundary condition $u(+1) = 0$ contains factors multiplied by a_i and c_i in the following equation:

$$\sum_{i=0}^{n-1} a_i \int_{-1}^1 \int_{-1}^w \int_{-1}^s \int_{-1}^t T_i(\eta) \, d\eta \, dw \, ds \, dt + \sum_{i=0}^3 c_i T_i(+1) = 0.$$

Notice the upper bound $x = 1$ of the outermost integral.

In this example, we use 100 Chebyshev polynomials and set $\text{tol} = 10^{-2}$. Figure 5 (left) depicts eigenvalues computed with our least-squares method applied to the integral reformulation in which boundary conditions are imposed exactly. The right panel depicts the residuals corresponding with eigenvalues computed with our algorithms applied to the integral reformulation and with basis recombination. We see that LSeig-bc applied to the integral reformulation computes accurate results for a larger number of eigenvalues compared with basis recombination. See also Table 1.

There are a number of techniques that can often improve the performance of LSeig. For example, *balancing* (diagonal similarity or diagonal equivalence transformation) is a standard technique for improving the conditioning of eigenvalue problems. As a simple version of this, we have observed that the residuals reported in the second and third columns of Table 1 improve by a factor of 10^{-2} to 10^{-3} if we divide A by $\nu := \frac{\|A\|_F}{\|B\|_F} \approx 100$ and then recover λ 's by multiplying computed eigenvalues by ν .

Our next example is an eigenvalue problem with eigenvalue-dependent boundary conditions.

Example 5. λ -dependent boundary conditions. We consider the problem $-u''(x) = \lambda u(x)$ where $x \in [0, 1]$ with boundary conditions $-u(0) = (\lambda + d)u'(0)$ and $u(1) = \lambda u'(1)$ in which $d = -4\pi^2$. This is a problem taken from [5, 13]. The first three eigenvalues of this problem have been computed in [13] by the RSM, which is based on the Shannon sampling theory. Using 100 Chebyshev polynomials and $\text{tol} = 10^{-9}$, our method computes 42 real eigenvalues in 0.14 seconds. Table 2 reports our results with

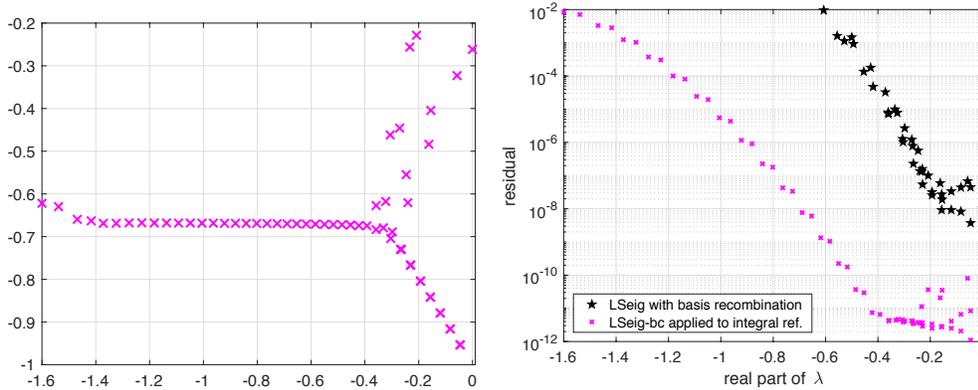


FIG. 5. Performance of methods for the Orr-Sommerfeld equation in Example 4. Left: 60 eigenvalues computed with integral reformulation. The rightmost eigenvalue is $-7.8191 \times 10^{-5} - 0.26157i$. Right: Relative residuals in (8.6) for 60 eigenvalues obtained with integral reformulation and 39 eigenvalues obtained with basis recombination applied to the original differential problem.

TABLE 1
Residuals $\|\mathcal{L}_A u - \lambda \mathcal{L}_B u\|_2 / \|\mathcal{L}_A u\|_2$ for the six eigenvalues of largest real part in Example 4.

Chebfun eigs	LSeig-bc	LSeig (BR)	LSeig-bc (int)
3.2×10^{-5}	5.9×10^{-5}	1.0×10^{-8}	2.0×10^{-10}
7.0×10^{-7}	1.9×10^{-4}	4.5×10^{-8}	8.4×10^{-12}
3.4×10^{-6}	1.6×10^{-4}	3.7×10^{-9}	1.1×10^{-12}
1.2×10^{-5}	1.1×10^{-4}	6.8×10^{-8}	8.0×10^{-11}
2.8×10^{-6}	3.7×10^{-4}	4.4×10^{-8}	6.6×10^{-12}
9.2×10^{-6}	3.1×10^{-4}	8.2×10^{-9}	2.1×10^{-12}

TABLE 2
Three smallest real eigenvalues of the problem in Example 5.

Exact	RSM [13]	LSeig
9.730886578213082033	<u>9.730887696302056807</u>	<u>9.730886578221018</u>
88.76331625258976337	<u>88.76323738197181406</u>	<u>88.76331625258112</u>
157.88411043863472059	<u>157.88422274978466468</u>	<u>157.8841104386164</u>

TABLE 3
Summary and comparison of spectral methods.

	Galerkin	Collocation	LSeig
$u = \sum c_i \phi_i$ represent operator	c_i	$u(x_i)$	c_i
boundary conditions	residual orth. impose on ϕ_i	force at x_i replace rows	least-squares least-squares

TABLE 4
Comparison between spectral methods, method of fundamental solutions, and LSeig.

	Galerkin	Fundamental solutions	LSeig
operator	choose c_i	✓	choose c_i
boundary conditions	✓	choose c_i	choose c_i

those available in [13], which sets “precision” to 10^{-10} . In each instance, underlined digits are those which match the exact eigenvalues. Note that RSM is applicable even if boundary conditions are nonlinear in terms of λ .

9. Discussion. Let us close with a high-level discussion of the different approaches of various methods. In the introduction we discussed the differences among spectral methods (collocation, Galerkin) and LSeig, which we summarize in Table 3. Recall that LSeig has a flavor of both spectral collocation and Galerkin methods.

Another intriguing class of methods is the method of fundamental solutions [20], revisited in (among others) [4]. These methods proceed in the “opposite” manner to spectral Galerkin/coefficient methods: whereas spectral Galerkin/coefficient methods use basis functions satisfying the boundary conditions and fit the operator, the method of fundamental solutions uses basis functions that satisfy the operator exactly, and then fits the boundary conditions, usually in the least-squares sense. In this sense LSeig can be regarded as a mixture of spectral coefficient methods and the method of fundamental solutions. We summarize the discussion in Table 4.

We believe the least-squares framework of LSeig and LSode would be an attractive alternative when nonstandard and tailor-made basis functions are available or can be computed.

Acknowledgments. The authors are grateful to Nick Trefethen for his encouragement, insightful discussions, and valuable comments on a draft. They also thank Davoud Mirzaei for pointing out the connection to LSFEM and the referees for providing several insightful comments which improved the presentation of this work.

REFERENCES

- [1] J. L. AURENTZ AND L. N. TREFETHEN, *Block operators and spectral discretizations*, SIAM Rev., 59 (2017), pp. 423–446, <https://doi.org/10.1137/16M1065975>.
- [2] Z. BATTLES, *Numerical Linear Algebra for Continuous Functions*, Ph.D. thesis, University of Oxford, 2005.

- [3] Z. BATTLES AND L. N. TREFETHEN, *An extension of MATLAB to continuous functions and operators*, SIAM J. Sci. Comput., 25 (2004), pp. 1743–1770, <https://doi.org/10.1137/S1064827503430126>.
- [4] T. BETCKE AND L. N. TREFETHEN, *Reviving the method of particular solutions*, SIAM Rev., 47 (2005), pp. 469–491, <https://doi.org/10.1137/S0036144503437336>.
- [5] P. A. BINDING AND P. J. BROWNE, *Oscillation theory for indefinite Sturm-Liouville problems with eigenparameter-dependent boundary conditions*, Proc. Roy. Soc. Edinburgh Sect. A, 127 (1997), pp. 1123–1136, <https://doi.org/10.1017/S0308210500026974>.
- [6] G. D. BIRKHOFF, *Boundary value and expansion problems of ordinary linear differential equations*, Trans. Amer. Math. Soc., 9 (1908), pp. 373–395, <https://doi.org/10.2307/1988661>.
- [7] G. D. BIRKHOFF, *On the asymptotic character of the solutions of certain linear differential equations containing a parameter*, Trans. Amer. Math. Soc., 9 (1908), pp. 219–231, <https://doi.org/10.2307/1988652>.
- [8] P. B. BOCHEV AND M. D. GUNZBURGER, *Least-Squares Finite Element Methods*, Springer, New York, 2009, <https://doi.org/10.1007/b13382>.
- [9] G. BOUTRY, M. ELAD, G. H. GOLUB, AND P. MILANFAR, *The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 582–601, <http://dx.doi.org/10.1137/S0895479803428795>.
- [10] J. P. BOYD, *Chebyshev and Fourier Spectral Methods*, Dover, New York, 2001.
- [11] C. CANUTO, M. Y. HUSSAINI, A. QUARTERONI, AND T. A. ZANG, *Spectral Methods—Fundamentals in Single Domains*, Springer, New York, 2006.
- [12] C.-H. CHAN, *Some eigenvalue problems for vectorial Sturm-Liouville equations with eigenparameter dependent boundary conditions*, Trans. Amer. Math. Soc., 364 (2012), pp. 119–136, <https://doi.org/10.1090/S0002-9947-2011-05269-4>.
- [13] B. CHANANE, *Computation of the eigenvalues of Sturm-Liouville problems with parameter dependent boundary conditions using the regularized sampling method*, Math. Comp., 74 (2005), pp. 1793–1801, <https://doi.org/10.1090/S0025-5718-05-01717-5>.
- [14] S. DAS AND A. NEUMAIER, *Solving overdetermined eigenvalue problems*, SIAM J. Sci. Comput., 35 (2013), pp. A541–A560, <https://doi.org/10.1137/110828514>.
- [15] C. DE BOOR, *An alternative approach to (the teaching of) rank, basis, and dimension*, Linear Algebra Appl., 146 (1991), pp. 221–229, [https://doi.org/10.1016/0024-3795\(91\)90026-S](https://doi.org/10.1016/0024-3795(91)90026-S).
- [16] T. A. DRISCOLL, *Automatic spectral collocation for integral, integro-differential, and integrally reformulated differential equations*, J. Comput. Phys., 229 (2010), pp. 5980–5998, <https://doi.org/10.1016/j.jcp.2010.04.029>.
- [17] T. A. DRISCOLL AND N. HALE, *Rectangular spectral collocation*, IMA J. Numer. Anal., 36 (2016), pp. 108–132, <https://doi.org/10.1093/imanum/dru062>.
- [18] T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
- [19] G. B. FOLLAND, *Fourier Analysis and Its Applications*, AMS, Providence, RI, 1992.
- [20] L. FOX, P. HENRICI, AND C. MOLER, *Approximations and bounds for eigenvalues of elliptic operators*, SIAM J. Numer. Anal., 4 (1967), pp. 89–102, <https://doi.org/10.1137/0704008>.
- [21] C.-I. GHEORGHIU, *Spectral Methods for Non-Standard Eigenvalue Problems. Fluid and Structural Mechanics and Beyond*, Springer, Cham, 2014.
- [22] C.-I. GHEORGHIU AND I. S. POP, *A modified Chebyshev-tau method for a hydrodynamic stability problem*, in Approximation and Optimization, Vol. II, 1997, pp. 119–126, <https://doi.org/10.1006/abio.1997.2380>.
- [23] M. A. GILLES AND A. TOWNSEND, *Continuous analogues of Krylov subspace methods for differential operators*, SIAM J. Numer. Anal., 57 (2019), pp. 899–924.
- [24] D. GOTTLIEB AND S. A. ORSZAG, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia, PA, 1977.
- [25] L. GREENBERG AND M. MARLETTA, *Numerical solution of non-self-adjoint Sturm-Liouville problems and related systems*, SIAM J. Numer. Anal., 38 (2001), pp. 1800–1845, <https://doi.org/10.1137/S0036142999358743>.
- [26] L. GREENGARD, *Spectral integration and two-point boundary value problems*, SIAM J. Numer. Anal., 28 (1991), pp. 1071–1080, <https://doi.org/10.1137/0728057>.
- [27] D. J. GRIFFITHS AND D. F. SCHROETER, *Introduction to Quantum Mechanics*, Cambridge University Press, Cambridge, UK, 2018.
- [28] N. J. GULIYEV, *Inverse Square Singularities and Eigenparameter Dependent Boundary Conditions are Two Sides of the Same Coin*, preprint, arXiv:2001.00061, 2019.
- [29] N. J. GULIYEV, *A Riesz basis criterion for Schrödinger operators with boundary conditions dependent on the eigenvalue parameter*, Anal. Math. Phys., 10 (2020), 2, <https://doi.org/10.1007/s13324-019-00348-0>.

- [30] B. HASHEMI, Y. NAKATSUKASA, AND L. N. TREFETHEN, *Rectangular Eigenvalue Problems*, preprint, arXiv:2112.13698, 2021.
- [31] J. S. HESTHAVEN, S. GOTTLIEB, AND D. GOTTLIEB, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, Cambridge, UK, 2007, <https://doi.org/10.1017/CBO9780511618352>.
- [32] H. HOCHSTADT, *On inverse problems associated with second-order differential operators*, Acta Math., 119 (1967), pp. 173–192, <https://doi.org/10.1007/BF02392082>.
- [33] S. ITO AND K. MUROTA, *An algorithm for the generalized eigenvalue problem for nonsquare matrix pencils by minimal perturbation approach*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 409–419, <https://doi.org/10.1137/14099231X>.
- [34] J.-F. LI, W. LI, X.-F. DUAN, AND M. XIAO, *Newton's method for the parameterized generalized eigenvalue problem with nonsquare matrix pencils*, Adv. Comput. Math., 47 (2021), 29, <https://doi.org/10.1007/s10444-021-09855-w>.
- [35] J.-F. LI, W. LI, S.-W. VONG, Q.-L. LUO, AND M. XIAO, *A Riemannian optimization approach for solving the generalized eigenvalue problem for nonsquare matrix pencils*, J. Sci. Comput., 82 (2020), 67, <https://doi.org/10.1007/s10915-020-01173-5>.
- [36] J.-F. LI, K. WANG, Y.-Y. LIU, X.-F. DUAN, AND X.-L. ZHOU, *A trust-region method for the parameterized generalized eigenvalue problem with nonsquare matrix pencils*, Numer. Linear Algebra Appl., 28 (2021), e2363, <https://doi.org/10.1002/nla.2363>.
- [37] M. MARLETTA, A. SHKALIKOV, AND C. TRETTER, *Pencils of differential operators containing the eigenvalue parameter in the boundary conditions*, Proc. Roy. Soc. Edinburgh Sect. A Math., 133 (2003), pp. 893–917.
- [38] S. OLVER AND A. TOWNSEND, *A fast and well-conditioned spectral method*, SIAM Rev., 55 (2013), pp. 462–489, <https://doi.org/10.1137/120865458>.
- [39] S. A. ORSZAG, *Accurate solution of the Orr–Sommerfeld stability equation*, J. Fluid Mech., 50 (1971), pp. 689–703.
- [40] R. PACHÓN, R. B. PLATTE, AND L. N. TREFETHEN, *Piecewise-smooth chebfun*s, IMA J. Numer. Anal., 30 (2010), pp. 898–916, <http://dx.doi.org/10.1093/imanum/drp008>.
- [41] J. D. PRYCE, *Numerical Solution of Sturm–Liouville Problems*, Monographs on Numerical Analysis, Oxford University Press, New York, 1993.
- [42] P. J. SCHMID AND D. S. HENNINGSON, *Stability and Transition in Shear Flows*, Springer, New York, 2001.
- [43] G. L. G. SLEIJPEN AND H. A. VANDERVORST, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
- [44] G. W. STEWART, *Afternotes Goes to Graduate School*, SIAM, Philadelphia, PA, 1998, <https://doi.org/10.1137/1.9781611971422>.
- [45] J. TAMARKIN, *Some General Problems of the Theory of Ordinary Differential Equations and Expansion of an Arbitrary Function in Series of Fundamental Functions*, Ph.D. thesis, Petrograd, 1917 (in Russian).
- [46] J. TAMARKIN, *Some general problems of the theory of ordinary linear differential equations and expansion of an arbitrary function in series of fundamental functions*, Math. Z., 27 (1928), pp. 1–54, <https://doi.org/10.1007/BF01171084>.
- [47] A. TOWNSEND AND L. N. TREFETHEN, *Continuous analogues of matrix factorizations*, Proc. R. Soc. A., 471 (2015), 20140585, <https://doi.org/10.1098/rspa.2014.0585>.
- [48] L. N. TREFETHEN, *Spectral Methods in MATLAB*, SIAM, Philadelphia, PA, 2000, <https://doi.org/10.1137/1.9780898719598>.
- [49] L. N. TREFETHEN, *Householder triangularization of a quasimatrix*, IMA J. Numer. Anal., 30 (2010), pp. 887–897, <https://doi.org/10.1093/imanum/drp018>.
- [50] L. N. TREFETHEN, *Approximation Theory and Approximation Practice, Extended Edition*, SIAM, Philadelphia, PA, 2019.
- [51] L. N. TREFETHEN AND D. BAU III, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997, <https://doi.org/10.1137/1.9780898719574>.
- [52] L. N. TREFETHEN, A. BIRKISSON, AND T. A. DRISCOLL, *Exploring ODEs*, SIAM, Philadelphia, PA, 2017.
- [53] L. N. TREFETHEN AND M. EMBREE, *Spectra and Pseudospectra*, Princeton University Press, Princeton, NJ, 2005.
- [54] W. F. TRENCH, *Elementary Differential Equations with Boundary Value Problems*, Trinity University, 2021, <https://math.libretexts.org/@go/page/9475>.
- [55] C. TRETTER, *Boundary eigenvalue problems for differential equations $N\eta = \lambda P\eta$ with λ -polynomial boundary conditions*, J. Differential Equations, 170 (2001), pp. 408–471.
- [56] T. G. WRIGHT AND L. N. TREFETHEN, *Pseudospectra of rectangular matrices*, IMA J. Numer. Anal., 22 (2002), pp. 501–519, <http://dx.doi.org/10.1093/imanum/22.4.501>.