

A Matlab Spectral Integration Suite (SISMatlab)

Gokul Hariharan,¹ Satish Kumar,¹ and Mihailo R. Jovanović²

¹*Department of Chemical Engineering and Materials Science,
University of Minnesota, Minneapolis, MN 55455, USA*

²*Ming Hsieh Department of Electrical and Computer Engineering,
University of Southern California, Los Angeles, CA 90089, USA*

I. THE SPECTRAL INTEGRATION METHOD: INTRODUCTION VIA AN EXAMPLE

We develop a customized Spectral Integration Suite in Matlab (SISMatlab) that is based on the method described in [2, 4]. In order to facilitate application to differential equations of order n , our implementation introduces minor modifications that we summarize next.

I.1. Basic features of SISMatlab

Let us consider a second-order linear differential equation,

$$\left(D^2 + \frac{1}{y^2 + 1} D - \epsilon^2 I\right) \phi(y) = d(y), \quad (1a)$$

with Dirichlet, Neumann, or Robin boundary conditions,

$$\phi(\pm 1) = \pm 1, \quad (1b)$$

$$[D\phi(\cdot)](\pm 1) = \pm 2, \quad (1c)$$

$$4[D\phi(\cdot)](\pm 1) + 3\phi(\pm 1) = \pm 3. \quad (1d)$$

Here, ϕ is the field of interest, d is an input, $\epsilon \in \mathbb{R}$ is a given constant, $y \in [-1, 1]$, and $D := d/dy$.

In the spectral integration method, the highest derivative in a differential equation is expressed in a basis of Chebyshev polynomials. Specifically, for Eq. (1a), we have

$$D^2 \phi(y) = \sum_{i=0}^{\infty} \phi_i^{(2)} T_i(y) =: \mathbf{t}_y^T \Phi^{(2)}, \quad (2)$$

where \sum' denotes a summation with the first term halved, $\Phi^{(2)} := [\phi_0^{(2)} \ \phi_1^{(2)} \ \phi_2^{(2)} \ \cdots]^T$ is the infinite vector of spectral coefficients $\phi_i^{(2)}$, and \mathbf{t}_y is the vector of Chebyshev polynomials of the first kind $T_i(y)$,

$$\mathbf{t}_y^T := [\tfrac{1}{2}T_0(y) \ T_1(y) \ T_2(y) \ \cdots]. \quad (3)$$

Integration of Eq. (2) in conjunction with the recurrence relations for integration of Chebyshev polynomials are used to determine spectral coefficients corresponding to lower derivatives of ϕ . For example, indefinite integration of Eq. (2) yields

$$D \phi(y) = \sum_{i=0}^{\infty} \phi_i^{(1)} T_i(y) + c_1 =: \mathbf{t}_y^T \Phi^{(1)} + c_1, \quad (4)$$

where c_1 is an integration constant and

$$\phi_i^{(1)} = \begin{cases} \frac{1}{2} \phi_1^{(2)}, & i = 0, \\ \frac{1}{2i} (\phi_{i-1}^{(2)} - \phi_{i+1}^{(2)}), & i \geq 1. \end{cases} \quad (5)$$

These expressions for $\phi_i^{(1)}$ are derived in Appendix A.

Similarly, indefinite integration of $D\phi$ allows us to express ϕ as

$$\phi(y) = \sum_{i=0}^{\infty} \phi_i^{(0)} T_i(y) + \tilde{c}_0 + c_1 y =: \mathbf{t}_y^T \mathbf{\Phi}^{(0)} + \tilde{c}_0 + c_1 y,$$

where \tilde{c}_0 and c_1 are integration constants and

$$\phi_i^{(0)} = \begin{cases} \frac{1}{2} \phi_1^{(1)}, & i = 0, \\ \frac{1}{2i} (\phi_{i-1}^{(1)} - \phi_{i+1}^{(1)}), & i \geq 1. \end{cases}$$

Equation (5) provides a recursive relation that is used to determine spectral coefficients of lower derivatives from the spectral coefficients of the highest derivative of the variable ϕ ,

$$\mathbf{\Phi}^{(1)} = \mathbf{Q} \mathbf{\Phi}^{(2)}, \quad \mathbf{\Phi}^{(0)} = \mathbf{Q}^2 \mathbf{\Phi}^{(2)},$$

where $\mathbf{Q}^2 := \mathbf{Q} \mathbf{Q}$, and

$$\mathbf{Q} := \begin{bmatrix} 0 & \frac{1}{2} & 0 & \dots \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & \dots \\ 0 & \frac{1}{4} & 0 & -\frac{1}{4} & 0 & \dots \\ 0 & 0 & \frac{1}{6} & 0 & -\frac{1}{6} & 0 & \dots \\ \vdots & \vdots & & \ddots & \ddots & \ddots \end{bmatrix}. \quad (6)$$

Since $T_0(y) = 1$ and $T_1(y) = y$, we let $c_0 := 2\tilde{c}_0$ and represent integration constants in the basis expansion of ϕ and $D\phi$ in terms of Chebyshev polynomials,

$$\phi(y) = \mathbf{t}_y^T \mathbf{Q}^2 \mathbf{\Phi}^{(2)} + \underbrace{\begin{bmatrix} \frac{1}{2} T_0(y) & T_1(y) \end{bmatrix}}_{\mathbf{K}^0} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}, \quad (7)$$

$$D\phi(y) = \mathbf{t}_y^T \mathbf{Q} \mathbf{\Phi}^{(2)} + \underbrace{\begin{bmatrix} \frac{1}{2} T_0(y) & T_1(y) \end{bmatrix}}_{\mathbf{K}^1} \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}. \quad (8)$$

By introducing the vector of integration constants $\mathbf{c}^{(2)} := [c_0 \ c_1]^T$, we can represent ϕ , $D\phi$, and $D^2\phi$ as

$$\phi(y) = \mathbf{t}_y^T (\mathbf{Q}^2 \mathbf{\Phi}^{(2)} + \mathbf{R}_2 \mathbf{c}^{(2)}) = \mathbf{t}_y^T \underbrace{\begin{bmatrix} \mathbf{Q}^2 & \mathbf{R}_2 \end{bmatrix}}_{\mathbf{J}_2} \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}, \quad (9)$$

$$D\phi(y) = \mathbf{t}_y^T (\mathbf{Q}^1 \mathbf{\Phi}^{(2)} + \mathbf{R}_1 \mathbf{c}^{(2)}) = \mathbf{t}_y^T \underbrace{\begin{bmatrix} \mathbf{Q}^1 & \mathbf{R}_1 \end{bmatrix}}_{\mathbf{J}_1} \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}, \quad (10)$$

$$D^2\phi(y) = \mathbf{t}_y^T (\mathbf{Q}^0 \mathbf{\Phi}^{(2)} + \mathbf{R}_0 \mathbf{c}^{(2)}) = \mathbf{t}_y^T \underbrace{\begin{bmatrix} \mathbf{Q}^0 & \mathbf{R}_0 \end{bmatrix}}_{\mathbf{J}_0} \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}, \quad (11)$$

where $\mathbf{Q}^0 = \mathbf{I}$ is an identity operator,

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{K}^{2-i} \\ \mathbf{0} \end{bmatrix}, \quad i = 0, 1, 2,$$

are matrices with an infinite number of rows and two columns, and $\mathbf{K}^2 = \mathbf{K} \mathbf{K}$ is a 2×2 zero matrix.

Finally, we utilize the expression for the product of two Chebyshev series [2, 6] to account for the nonconstant coefficient $a(y) := 1/(y^2 + 1)$ in Eq. (1a). For a function $a(y)$ in the basis of Chebyshev polynomials,

$$a(y) = \sum_{i=0}^{\infty} a_i T_i(y), \quad (12a)$$

the multiplication operator is given by

$$\mathbf{M}_a = \frac{1}{2} \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_1 & a_0 & a_1 & a_2 & \ddots \\ a_2 & a_1 & a_0 & a_1 & \ddots \\ a_3 & a_2 & a_1 & a_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ a_1 & a_2 & a_3 & a_4 & \cdots \\ a_2 & a_3 & a_4 & a_5 & \ddots \\ a_3 & a_4 & a_5 & a_6 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \quad (12b)$$

Thus, in the basis of Chebyshev polynomials, we can express the differential equation (1a) as,

$$\mathbf{t}_y^T (\mathbf{J}_0 + \mathbf{M}_a \mathbf{J}_1 - \epsilon^2 \mathbf{J}_2) \begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \mathbf{t}_y^T \mathbf{d}, \quad (13)$$

where \mathbf{d} is the vector of spectral coefficients associated with the input $d(y)$ in Eq. (1a). Furthermore, we can use Eq. (9) to write the Dirichlet boundary conditions in Eq. (1b) as

$$\mathbf{t}_{\pm 1}^T \mathbf{J}_2 \begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \pm 1, \quad (14a)$$

equation (10) to express the Neumann boundary conditions in Eq. (1c) as

$$\mathbf{t}_{\pm 1}^T \mathbf{J}_1 \begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \pm 2, \quad (14b)$$

and Eqs. (9) and (10) to represent the Robin boundary conditions in Eq. (1d) as

$$\mathbf{t}_{\pm 1}^T (4 \mathbf{J}_1 + 3 \mathbf{J}_2) \begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \pm 3. \quad (14c)$$

Combining Eqs. (13) and (14a), we can represent Eq. (1a) with Dirichlet boundary conditions (1b) in the basis of Chebyshev polynomials as

$$\underbrace{\begin{bmatrix} \mathbf{J}_0 + \mathbf{M}_a \mathbf{J}_1 - \epsilon^2 \mathbf{J}_2 \\ \mathbf{t}_{+1}^T \mathbf{J}_2 \\ \mathbf{t}_{-1}^T \mathbf{J}_2 \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}}_{\mathbf{v}} = \underbrace{\begin{bmatrix} \mathbf{d} \\ +1 \\ -1 \end{bmatrix}}_{\mathbf{f}} \Rightarrow \mathbf{F} \mathbf{v} = \mathbf{f}. \quad (15)$$

Similarly, the problem with Neumann and Robin boundary conditions can be solved by replacing the last two rows in Eq. (15) with Eqs. (14b) and (14c), respectively.

We use the projection operator (see [6, Section 2.4])

$$\mathbf{P} = [\mathbf{I}_{N+1} \quad \mathbf{0}], \quad (16)$$

where \mathbf{I}_{N+1} is an identity matrix of dimension $N+1$, and \mathbf{P} is a matrix with an infinite number of columns, to obtain a finite-dimensional approximation of Eq. (15) by truncating the infinite vector of spectral coefficients $\Phi^{(2)}$ to a vector $\hat{\Phi}^{(2)}$ with $N+1$ components,

$$\hat{\mathbf{F}} \hat{\mathbf{v}} = \hat{\mathbf{f}}, \quad (17)$$

where

$$\hat{\mathbf{F}} = \mathbf{S} \mathbf{F} \mathbf{S}^T, \quad \hat{\mathbf{v}} = \mathbf{S} \mathbf{v}, \quad \hat{\mathbf{f}} = \mathbf{S} \mathbf{f}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}. \quad (18)$$

Based on Eq. (9), approximate solution $\hat{\phi}(y)$ to Eq. (1a) is obtained by integrating the solution $\hat{\mathbf{v}}$ to Eq. (17) twice,

$$\hat{\phi}(y) = \hat{\mathbf{t}}_y^T \hat{\mathbf{J}}_2 \hat{\mathbf{F}}^{-1} \hat{\mathbf{f}}, \quad (19)$$

where $\hat{\mathbf{t}}_y = \mathbf{P} \mathbf{t}_y$ and $\hat{\mathbf{J}}_2 = \mathbf{P} \mathbf{J}_2 \mathbf{S}^T$.

I.2. Eigenvalue and frequency response analysis

Herein, we illustrate how the spectral integration method can be used to conduct modal and nonmodal analysis of the reaction-diffusion equation,

$$\phi_t(y, t) = \phi_{yy}(y, t) - \epsilon^2 \phi(y, t) + d(y, t), \quad (20a)$$

with homogeneous Neumann boundary conditions,

$$[\partial_y \phi(\cdot, t)](\pm 1) = 0, \quad (20b)$$

where t is time, $y \in [-1, 1]$ is a spatial variable, and $\epsilon \in \mathbb{R}$.

Eigenvalue decomposition. Let us consider the eigenvalue decomposition of the operator in system (20),

$$(\mathbf{D}^2 - \epsilon^2 \mathbf{I}) \phi(y) = \lambda \phi(y), \quad (21a)$$

$$\mathbf{D} \phi(\pm 1) = 0, \quad (21b)$$

where λ is an eigenvalue and ϕ is an eigenfunction. Using the relations in Eqs. (9)-(11), differential equation (21a) and boundary conditions (21b) can be expressed in the basis of Chebyshev polynomials as,

$$\underbrace{(\mathbf{J}_0 - \epsilon^2 \mathbf{J}_2)}_{\mathbf{F}} \underbrace{\begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}}_{\mathbf{v}} = \lambda \mathbf{J}_2 \underbrace{\begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}}_{\mathbf{v}}, \quad (22a)$$

$$\underbrace{\begin{bmatrix} \mathbf{t}_{-1} \mathbf{J}_1 \\ \mathbf{t}_{+1} \mathbf{J}_1 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}}_{\mathbf{v}} = 0, \quad (22b)$$

or, equivalently,

$$\mathbf{F} \mathbf{v} = \lambda \mathbf{J}_2 \mathbf{v}, \quad (23a)$$

$$\mathbf{M} \mathbf{v} = 0, \quad (23b)$$

Thus, only functions \mathbf{v} that belong to the null-space of the operator \mathbf{M} in Eq. (23b) are permissible solutions to Eq. (23a). A finite-dimensional representation of system (23) is given by

$$\hat{\mathbf{F}} \hat{\mathbf{v}} = \lambda \hat{\mathbf{J}}_2 \hat{\mathbf{v}}, \quad (24a)$$

$$\hat{\mathbf{M}} \hat{\mathbf{v}} = 0, \quad (24b)$$

where,

$$\hat{\mathbf{F}} = \mathbf{P} \mathbf{F} \mathbf{S}^T, \quad \hat{\mathbf{M}} = \mathbf{M} \mathbf{S}^T, \quad \hat{\mathbf{J}}_2 = \mathbf{P} \mathbf{J}_2 \mathbf{S}^T, \quad \hat{\mathbf{v}} = \mathbf{S} \mathbf{v},$$

with \mathbf{P} and \mathbf{S} defined in Eqs. (16) and (18), respectively. SVD of the fat full-row-rank matrix $\hat{\mathbf{M}}$ in (24b)

can be used to parameterize its null-space [5],

$$\hat{\mathbf{M}} \hat{\mathbf{v}} = \hat{\mathbf{U}} \hat{\Sigma} \hat{\mathbf{V}}^\dagger \hat{\mathbf{v}} = \hat{\mathbf{U}} \begin{bmatrix} \hat{\Sigma}_1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_1^\dagger \\ \hat{\mathbf{V}}_2^\dagger \end{bmatrix} \hat{\mathbf{v}} = 0. \quad (25)$$

Hence, $\hat{\mathbf{v}} := \hat{\mathbf{V}}_2 \hat{\mathbf{u}}$ parametrizes the null-space [8] of the matrix $\hat{\mathbf{M}}$ in Eq. (24b). Substituting this expression for $\hat{\mathbf{v}}$ to Eq. (24a) yields the finite-dimensional generalized eigenvalue problem,

$$(\hat{\mathbf{F}} \hat{\mathbf{V}}_2) \hat{\mathbf{u}} = \lambda (\hat{\mathbf{J}}_2 \hat{\mathbf{V}}_2) \hat{\mathbf{u}}, \quad (26)$$

which can be used to compute the eigenpair $(\lambda, \hat{\mathbf{u}})$. The eigenfunctions in Eq. (21) can be recovered by using the null-space parametrization in conjunction with the integration operator (see Eq. (9)),

$$\hat{\phi}(y) = \hat{\mathbf{t}}_y^T \hat{\mathbf{J}}_2 \hat{\mathbf{V}}_2 \hat{\mathbf{u}}.$$

Frequency response analysis. The temporal Fourier transform can be used to represent the frequency response operator associated with system (20) as a two-point boundary value problem (TPBVP),

$$\begin{aligned} [\mathcal{A}(\omega) \phi(\cdot)](y) &= [\mathcal{B}(\omega) d(\cdot)](y), \\ \xi(y) &= [\mathcal{C}(\omega) \phi(\cdot)](y), \\ [\mathcal{L}_a \phi(\cdot)](a) &= [\mathcal{L}_b \phi(\cdot)](b) = 0, \end{aligned} \quad (27)$$

where $\mathcal{A}(\omega) = (i\omega + \epsilon^2)I - D^2$, $\mathcal{B} = \mathcal{C} = I$, and $\mathcal{L}_{\pm 1} = D$. A feedback interconnection of the frequency response operator with its adjoint can be used to compute singular values (see our paper and [1])

$$\begin{bmatrix} 0 & \mathcal{B}\mathcal{B}^\dagger \\ \mathcal{C}^\dagger \mathcal{C} & 0 \end{bmatrix} \begin{bmatrix} \phi(y) \\ \psi(y) \end{bmatrix} = \gamma \begin{bmatrix} \mathcal{A} & 0 \\ 0 & \mathcal{A}^\dagger \end{bmatrix} \begin{bmatrix} \phi(y) \\ \psi(y) \end{bmatrix}, \quad (28a)$$

$$\begin{bmatrix} [\mathcal{L}_{\pm 1} \phi(\cdot)](\pm 1) \\ [\mathcal{L}_{\pm 1} \psi(\cdot)](\pm 1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (28b)$$

where $(\cdot)^\dagger$ denotes the adjoint of the operator (\cdot) and ψ is the auxiliary variable associated with the adjoint operator. The resulting eigenvalues determine the singular values in pairs of opposite signs, i.e., $\gamma = \pm \sigma$. Using definition of the operators \mathcal{A} , \mathcal{B} , \mathcal{C} , as well as their adjoints, we can express Eq. (28a) as

$$\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \phi(y) \\ \psi(y) \end{bmatrix} = \lambda \begin{bmatrix} (i\omega + \epsilon^2)I - D^2 & 0 \\ 0 & (-i\omega + \epsilon^2)I - D^2 \end{bmatrix} \begin{bmatrix} \phi(y) \\ \psi(y) \end{bmatrix}, \quad (28c)$$

and employ Eqs. (9)-(11) to obtained a discretized approximation of Eq. (28),

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{J}_2 \\ \mathbf{J}_2 & \mathbf{0} \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(\phi)} \\ \Psi^{(2)} \\ \mathbf{c}^{(\psi)} \end{bmatrix}}_{\mathbf{v}} = \gamma \underbrace{\begin{bmatrix} (i\omega + \epsilon^2)\mathbf{J}_2 - \mathbf{J}_0 & \mathbf{0} \\ \mathbf{0} & (-i\omega + \epsilon^2)\mathbf{J}_2 - \mathbf{J}_0 \end{bmatrix}}_{\mathbf{E}} \underbrace{\begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(\phi)} \\ \Psi^{(2)} \\ \mathbf{c}^{(\psi)} \end{bmatrix}}_{\mathbf{v}}, \quad (29a)$$

$$\underbrace{\begin{bmatrix} \mathbf{t}_{+1}^T \mathbf{J}_1 & \mathbf{0} \\ \mathbf{t}_{-1}^T \mathbf{J}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{t}_{+1}^T \mathbf{J}_1 \\ \mathbf{0} & \mathbf{t}_{-1}^T \mathbf{J}_1 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \Phi^{(2)} \\ \mathbf{c}^{(\phi)} \\ \Psi^{(2)} \\ \mathbf{c}^{(\psi)} \end{bmatrix} = \mathbf{0}. \quad (29b)$$

Finally, the projection operator (16) yields a finite dimensional approximation of system (29),

$$\hat{\mathbf{F}} \hat{\mathbf{v}} = \gamma \hat{\mathbf{E}} \hat{\mathbf{v}}, \quad (30a)$$

$$\hat{\mathbf{M}} \hat{\mathbf{v}} = \mathbf{0}, \quad (30b)$$

with

$$\hat{\mathbf{F}} = \mathbf{P}_2 \mathbf{F} \mathbf{S}_2^T, \quad \hat{\mathbf{M}} = \mathbf{M} \mathbf{S}_2^T, \quad \hat{\mathbf{E}} = \mathbf{P}_2 \mathbf{E} \mathbf{S}_2^T, \quad \hat{\mathbf{v}} = \mathbf{S}_2 \mathbf{v}, \quad \mathbf{P}_2 = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix},$$

where the expressions for \mathbf{P} and \mathbf{S} are given in Eqs. (16) and (18), respectively. The same procedure as in the modal analysis can be used to parametrize the null-space of the matrix $\hat{\mathbf{M}}$ in Eq. (30b). Namely, SVD of $\hat{\mathbf{M}}$ in Eq. (25) reduces Eq. (30) to the generalized eigenvalue problem,

$$(\hat{\mathbf{F}} \hat{\mathbf{V}}_2) \hat{\mathbf{u}} = \gamma (\hat{\mathbf{E}} \hat{\mathbf{V}}_2) \hat{\mathbf{u}}, \quad (31)$$

and yielding,

$$\begin{bmatrix} \hat{\phi}(y) \\ \hat{\psi}(y) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{t}}_y^T \hat{\mathbf{J}}_2 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{t}}_y^T \hat{\mathbf{J}}_2 \end{bmatrix} \hat{\mathbf{V}}_2 \mathbf{u}. \quad (32)$$

II. ARBITRARY ORDER LINEAR DIFFERENTIAL EQUATION WITH NON-CONSTANT COEFFICIENTS

In the previous section we considered second-order differential equations, and discussed how spectral integration is used to solve for a forcing, eigenvalues, and frequency responses. In this section, we illustrate this process for an n th order linear differential equation with non-constant coefficients.

Consider a general representation of an n th order linear differential equation with non-constant coefficients,

$$\sum_{k=0}^n a^{(k)}(y) D^k \phi(y) = d(y), \quad (33a)$$

$$\sum_{k=0}^{n-1} b^{(k, \mathbf{p})} D^k \phi(\mathbf{p}) = \mathbf{q}, \quad (33b)$$

where $a^{(k)}$ are the non-constant coefficients, and d is an input, $b^{(k, \mathbf{p})}$ are constant coefficients associated with boundary constraints (a general case of mixed boundary conditions), at a vector of evaluation points, \mathbf{p} , and corresponding values at the boundaries, \mathbf{q} .

II.1. Differential equation

In the same manner as the second derivative in Eq. (2) for the reaction-diffusion equation Eq. (14a), the highest derivative of the variable $\phi(y)$ in Eq. (33a) is expressed in a basis of Chebyshev polynomials as

$$D^n \phi(y) = \sum_{i=0}^{\infty} \phi_i^{(n)} T_i(y) =: \mathbf{t}_y^T \Phi^{(n)}, \quad (34)$$

where, $\Phi^{(n)} = [\phi_0^{(n)} \quad \phi_1^{(n)} \quad \dots \quad \phi_{\infty}^{(n)}]^T$. The lower derivatives are expressed as,

$$D^i \phi(y) = \mathbf{t}_y^T \left(\mathbf{Q}^{n-i} \Phi^{(n)} + \mathbf{R}_{n-i} \mathbf{c} \right) = \underbrace{\begin{bmatrix} \mathbf{Q}^{n-i} & \mathbf{R}_{n-i} \end{bmatrix}}_{\mathbf{J}_{n-i}} \begin{bmatrix} \Phi^{(n)} \\ \mathbf{c} \end{bmatrix}, \quad (35)$$

where, \mathbf{Q} is defined in Eq. (6), $\mathbf{c} = [c_0 \ c_1 \ \cdots \ c_{n-1}]^T$ are the n constants of integration that result from integrating the highest derivative Eq. (34), and

$$\mathbf{R}_i := \begin{bmatrix} \mathbf{K}^{n-i} \\ \mathbf{0} \end{bmatrix},$$

are matrices with n columns and infinite number of rows, where [4, Eq. 10]

$$\mathbf{K} = \begin{bmatrix} 0 & 2 & 0 & 6 & 0 & 10 & \cdots \\ 0 & 0 & 4 & 0 & 8 & 0 & \ddots \\ 0 & 0 & 0 & 6 & 0 & 10 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \quad (36)$$

is a matrix of dimension $n \times n$.

II.2. Infinite-dimensional representation

The differential equation in Eq. (33a) is expressed in a Chebyshev basis using Eqs. (35) and (12) as

$$\mathbf{t}_y^T \sum_{k=0}^n \mathbf{M}_{a^{(k)}} \mathbf{J}_{n-k} \begin{bmatrix} \Phi^{(n)} \\ \mathbf{c} \end{bmatrix} = \mathbf{t}_y^T \mathbf{d}, \quad (37)$$

and the boundary conditions Eq. (33b) as

$$\mathbf{t}_p^T \sum_{k=0}^{n-1} b^{(k,p)} \mathbf{J}_{n-k} \begin{bmatrix} \Phi^{(n)} \\ \mathbf{c} \end{bmatrix} = \mathbf{q}. \quad (38)$$

Thus the infinite-dimensional representation based on equating the terms of the same basis for the differential equation Eq. (33a) using Eq. (37) and appending boundary conditions in Eq. (38) is given by,

$$\underbrace{\begin{bmatrix} \sum_{k=0}^n \mathbf{M}_{a^{(k)}} \mathbf{J}_{n-k} \\ \mathbf{t}_p^T \sum_{k=0}^{n-1} b^{(k,p)} \mathbf{J}_{n-k} \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \Phi^{(n)} \\ \mathbf{c} \end{bmatrix}}_{\mathbf{v}} = \underbrace{\begin{bmatrix} \mathbf{d} \\ \mathbf{q} \end{bmatrix}}_{\mathbf{f}}. \quad (39)$$

II.3. Finite-dimensional approximation

The infinite-dimensional representation in Eq. (39)

$$\mathbf{F} \mathbf{v} = \mathbf{f}, \quad (40)$$

is reduced to finite-dimensions,

$$\hat{\mathbf{F}} \hat{\mathbf{v}} = \hat{\mathbf{f}}, \quad (41)$$

using the projection operator Eq. (16) where,

$$\hat{\mathbf{F}} = \mathbf{S} \mathbf{F} \mathbf{S}^T, \quad \hat{\mathbf{v}} = \mathbf{S} \mathbf{v}, \quad \hat{\mathbf{f}} = \mathbf{S} \mathbf{f}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix}. \quad (42)$$

The expression for $\hat{\mathbf{v}}$ is obtained by solving Eq. (41), the solution for $\phi(y)$ in Eq. (33) is approximated using Eq. (35) with $i = 0$ as,

$$\phi(y) \approx \hat{\mathbf{t}}_y^T \hat{\mathbf{J}}_n \hat{\mathbf{v}},$$

where,

$$\hat{\mathbf{t}}_y = \mathbf{P} \mathbf{t}_y, \quad \hat{\mathbf{J}}_n = \mathbf{P} \mathbf{J}_n \mathbf{S}^T.$$

and \mathbf{S} and \mathbf{P} are defined in Eqs. (42) and (16) respectively.

III. THE SPECTRAL INTEGRATION SUITE

The spectral integration suite is a set of handy routines to derive finite-dimensional approximations to linear operators and block-matrix operators using the spectral integration method [2, 4]. In this section, we describe the functions that make up this suite.

Note: Our implementation needs that N (where $N + 1$ is the number of basis functions) is an odd number.

- **sety(N)**: Sets points in physical space, $y_i = \cos(\pi(i + 0.5)/(N + 1))$ over $N + 1$ points. These points are such that when we take a discrete cosine transform, we have an array that represents spectral coefficients of a Chebyshev basis [7, Eq. 12.4.16-17].

```
N = 63;
y = sety(N);
f = 1 - y.^2;
plot(y,f);
```

- **phys2cheb.m**: Takes points in physical space (we refer to this as phys-space in this text and our codes) and converts them to an array of spectral coefficients in the basis of Chebyshev polynomials of the first kind (we refer to this as cheb-space in this text and our codes) using [7, Eq. 12.4.16-17].

```
N = 63;
y = sety(N);
f_phys = 1 - y.^2; % [0.5*T_0 T_1 T_2]^T [1 0 -0.5] = 0.5 - 0.5 (2 y^2 -1) = 1 - y^2
f_cheb = phys2cheb(f);
disp(f_cheb);
```

- **cheb2phys.m**: Takes an array of spectral coefficients in the basis of Chebyshev polynomials of the first kind, and converts them to points in phys-space using [7, Eq. 12.4.16-17].
- **Matgen(n,N)**: Generates $[\mathbf{Q}, \mathbf{K}, \mathbf{J}] = \text{Matgen}(n, N)$, where \mathbf{Q} contains matrices corresponding to \mathbf{Q} in Eq. (6), \mathbf{K} corresponding to \mathbf{K} in Eq. (36), and the matrices in \mathbf{J} for \mathbf{J} in (35).

```
N = 9; % Number of basis functions
m = 2; % Order of differential equation
[Q,K,J] = Matgen(m,N);
Q{1} % Q^0: identity
Q{2} % Q^1: see eq. for Q.
Q{3} % Q^2.

K{1} % K^1 see eq. for K
K{2} % K^0

J{1} % J_0 see eq. for J
J{2} % J_1
J{3} % J_2
```

- **MultMat(f)**: Produces the finite-dimensional representation of the matrix needed to account for non-constant coefficients, i.e., \mathbf{M}_a in Eq. (12).

- **Discretize(n,N,L)**: Produces the finite-dimensional representation of linear operators or block-matrix operators using **Matgen** and **MultMat**, taking inputs as the highest differential order of the variable, n , N , and the linear operator L (linear operators are specified using cells in Matlab, e.g., the operator $aD^2 + bD + c$ is represented by a 3×1 cell with values $L\{1\} = a$, $L\{2\} = b$, and $L\{3\} = c$). (See the basic-example code in the examples directory to learn how to use **Matgen** and **MultMat** directly).

Block-matrix operators are again cells of linear operators. For example,

```
% Dy operator: 1 Dy + 0
L11 = cell(2,1), L11{1} = 1; L11{2} = 0;

% Dyy + 2 Dy operator:
L12 = cell(3,1); L12{1} = 1; L12{2} = 2; L12{3} = 0;

% 2 Dyy + 3 Dy + 1 operator:
L21 = cell(3,1); L21{1} = 2; L21{2} = 3; L21{3} = 1;

% Identity operator: 1
L22 = cell(1,1), L22{1} = 1;

% Make block-matrix operator:
L = cell(2,2);
L{1,1} = L11; L{1,2} = L12;
L{2,1} = L21; L{2,2} = L22;
```

- **AdjointFormal** Takes a linear operator or a block-matrix operator and returns the formal adjoint by integrating by parts.

```
Lad = AdjointFormal(L);
```

- **MultOps** Gives the composition of two linear (block-matrix) operators of compatible dimensions.

```
L_composition = MultOps(L11,L12);
```

- **BcMat(n,N,eval,L)**: Generates a matrix of boundary evaluations given the highest order of the linear differential equation, n , N , the evaluation points, **eval**, and the linear operator to be applied at that point (Dirichlet, Neumann or mixed).

In addition to these primary functions, we provide the following auxiliary functions that are useful in certain applications:

- **ChebMat2CellMat** Takes a matrix of size $mN \times n$, and returns a cell of arrays of size $m \times n$, each element in the cell is a vector representing a function in y .
- **keepConverged** Takes in eigenvalues, eigenvectors, and N , and returns those eigenvalues and vectors that have converged to machine precision.
- **integ** Integrates a function in phys-space.
- **ChebEval** Evaluates a function in cheb-space at points in the domain.

In summary, these are sufficient for most problems to compute eigenvalues of or solve for inputs to linear differential equations or block-matrix operators. Readers are encouraged to try out the code snippets (these are compiled into a Matlab live-script, **Illustrations_m** in the examples directory) to see how the codes relate to the documentation presented here. Furthermore, the examples directory has several applications that use this suite and may serve as templates to your own problems. The examples whose file-names end with “_details ” provide implementation details.

User-level functions

The routines mentioned in the previous section are useful to solve a broader class of problems, however, for users who would like to use a simple interface, we provide the following user-level functions.

- `sisSolves(m,N,A,bc,f)` This routine solves a TPBVP, $\mathcal{A}\phi(y) = f(y)$ by taking inputs as `m` – the differential order of the system, `N` – the number of Chebyshev polynomials, `A` – the linear (block-matrix) operator, `bc` – the boundary conditions, and `f` – the input in phys-space. By default, this routine uses sparse matrix solvers, and the optional argument, 'Full', i.e., `sisSolves(m,N,A,bc,f,'Full')` can be passed to use the regular matrix solver.

The boundary conditions `bc` are specified via the boundary conditions class `BCs` with inputs as the number of boundary conditions and number of variables. For example, the robin boundary conditions,

$$\begin{aligned} 2[D\phi(\cdot)](-1) + 3\phi(-1) &= 4, \\ 5[D\phi(\cdot)](+1) + 6\phi(+1) &= 7, \end{aligned}$$

are represented as

```
bc = BCs(2,1); % Two boundary conditions, on one variable

OpLeft = cell(2,1); OpLeft{1} = 2; OpLeft{2} = 3; % for 2 Dy + 3 I
OpRight = cell(2,1); OpRight{1} = 5; OpRight{2} = 6; % for 5 Dy + 6 I

bc.Operator = {OpLeft;OpRight};
bc.Points = [-1;1];
bc.Values = [4;7];
```

- `sisEigs(m,N,F,E,bc)` Use this function to solve differential eigenvalue problems $\mathcal{F}\phi(y) = \lambda\mathcal{E}\phi(y)$. This function can be called with the following arguments:

```
[V,lambda] = sisEigs(m,N,F,E,bc)
[V,lambda] = sisEigs(m,N,F,E,bc,K)
[V,lambda] = sisEigs(m,N,F,E,bc,K,SIGMA)
```

where, the outputs `V` and `lambda` are the eigenvectors and eigenvalues, and the inputs are: `m` – a vector specifying the differential order of each variable in the block-matrix operator, `N` – the number of Chebyshev polynomials, `(F,E)` – the linear (block-matrix) operators in generalized eigenvalue problem with $(\mathcal{F},\mathcal{E})$, and `bc` – boundary conditions using the class `BCs()`.

By default, this function computes the 6 eigenvalues of largest real parts (and corresponding eigenvectors). The optional arguments modify this behaviour mimicking `eigs()` in Matlab: `K` – specify number of eigenvalues to be computed, `SIGMA` – if `SIGMA` is a scalar, the eigenvalues found are the ones closest to `SIGMA`, and for `SIGMA` being a character array, use 'LR' and 'SR' for the eigenvalues of largest and smallest real part, and 'LM' and 'SM' for largest and smallest magnitude. Lastly, setting `SIGMA` to 'Full' will force the solver to use Matlab's `eig()` instead of `eigs()`; in this special case, `K` is not used and all eigenvalues from the finite-dimensional approximation are the output.

- `sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj)` Use this function to solve for singular values of frequency responses of TPBVPs in the representation in Eq. (27). This function can be called with the following arguments:

```
[V,gamma] = sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj);
[V,gamma] = sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj,K);
[V,gamma] = sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj,K,SIGMA);
```

where, the outputs `V` and `gamma` are the singular vectors and singular values, and the inputs are: `m` – a vector specifying the differential order of each variable in the regular operator, `mad` – a vector specifying the differential order of each variable in the adjoint operator, `N` – the number of Chebyshev polynomials, `(A,B,C)` – the linear (block-matrix) operators \mathcal{A} , \mathcal{B} and \mathcal{C} in the TPBVP representation in Eq. (27), `(bcReg,bcAdj)` – boundary conditions on the regular and adjoint variables specified using the class `BCs()`.

Note: We currently do not provide a routine to compute adjoint boundary conditions, you need to explicitly provide it as an input. However, we understand that this can be cumbersome as adjoint boundary conditions are derived by integrating by parts. A Mathematica package 'AdjointFinder'

posted on our website can help you derive analytical expressions for adjoint boundary conditions (and operators) by automatic integration by parts.

The optional arguments (K,SIGMA) work in the same manner as `sisEigs`.

Appendix A: Recurrence relations

Consider the expression for the highest derivative of a second order differential equation in a Chebyshev basis,

$$D^2 u(y) = u_0^{(2)} \frac{1}{2} T_0(y) + u_1^{(2)} T_1(y) + u_2^{(2)} T_2(y) + u_3^{(2)} T_3(y) + \dots \quad (\text{A1})$$

Relation of Chebyshev polynomials with derivatives is given by [3, Equation 3.25]

$$T_0(y) = T_1'(y), \quad (\text{A2a})$$

$$T_1(y) = \frac{1}{4} T_2'(y), \quad (\text{A2b})$$

$$T_n(y) = \frac{1}{2} \left(\frac{T_{n+1}'(y)}{n+1} - \frac{T_{n-1}'(y)}{n-1} \right), \quad n > 1. \quad (\text{A2c})$$

Substituting Eq. (A2) in Eq. (A1) and making an indefinite integration on the resultant expression yields,

$$\begin{aligned} D v(y) &= \frac{u_0^{(2)}}{2} y + \frac{u_1^{(2)}}{2} y^2 + \frac{u_2^{(2)}}{2} \left(\frac{T_3(y)}{3} - \frac{T_1(y)}{1} \right) + \frac{u_3^{(2)}}{2} \left(\frac{T_4(y)}{4} - \frac{T_2(y)}{2} \right) \\ &+ \frac{u_4^{(2)}}{2} \left(\frac{T_5(y)}{5} - \frac{T_3(y)}{3} \right) + \dots + c_0, \end{aligned} \quad (\text{A3})$$

where c_0 is the effective integration constant. As $y^2 = (T_0(y) + T_2(y))/2$, Eq. (A3) takes the form:

$$D v(y) = \frac{u_0^{(2)}}{2} y + \frac{u_1^{(2)}}{2} \left(\frac{T_0(y) + T_2(y)}{2} \right) + \frac{u_2^{(2)}}{2} \left(\frac{T_3(y)}{3} - \frac{T_1(y)}{1} \right) \quad (\text{A4})$$

$$\begin{aligned} &+ \frac{u_3^{(2)}}{2} \left(\frac{T_4(y)}{4} - \frac{T_2(y)}{2} \right) + \frac{u_4^{(2)}}{2} \left(\frac{T_5(y)}{5} - \frac{T_3(y)}{3} \right) + \dots + c_0, \\ &= T_1(y) \underbrace{\left(\frac{u_0^{(2)}}{2} - \frac{u_2^{(2)}}{2} \right)}_{u_1^{(1)}} + T_2(y) \underbrace{\left(\frac{u_1^{(2)}}{4} - \frac{u_3^{(2)}}{4} \right)}_{u_2^{(1)}} + T_3(y) \underbrace{\left(\frac{u_2^{(2)}}{6} - \frac{u_4^{(2)}}{6} \right)}_{u_3^{(1)}} + \dots + \underbrace{\frac{u_1^{(2)}}{4}}_{u_0^{(1)}/2} + c_0. \end{aligned} \quad (\text{A5})$$

Hence we have from Eq. (A5),

$$D v(y) = u_0^{(1)} \frac{1}{2} T_0(y) + u_1^{(1)} T_1(y) + u_2^{(1)} T_2(y) + u_3^{(1)} T_3(y) + \dots + c_0, \quad (\text{A6})$$

where $u_0^{(1)} = u_1^{(2)}/2$ and the remaining coefficients for $u_i^{(1)}$ from the recursive relation in Eq. (5).

-
- [1] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the \mathcal{H}_∞ norm of a transfer matrix and related problems. *Math. Control Signals Syst.*, 2(3):207–219, 1989.
 - [2] K. Du. On well-conditioned spectral collocation and spectral methods by the integral reformulation. *SIAM J. Sci. Comp.*, 38:A3247–A3263, 2016.
 - [3] A. Gil, J. Segura, and N. M. Temme. Chebyshev expansions. In *Numerical Methods for Special Functions*, chapter 3, pages 51–86. Society for Industrial and Applied Mathematics, 2007.
 - [4] L. Greengard. Spectral integration and two-point boundary value problems. *SIAM J. Numer. Anal.*, 28:1071–1080, 1991.
 - [5] M. R. Jovanović and B. Bamieh. A formula for frequency responses of distributed systems with one spatial variable. *Syst. Control Lett.*, 55(1):27–37, January 2006.
 - [6] S. Olver and A. Townsend. A fast and well-conditioned spectral method. *SIAM Rev.*, 55:462–489, 2013.
 - [7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.
 - [8] G. Strang. The fundamental theorem of linear algebra. *Amer. Math. Monthly*, 100(9):848–855, 1993.