Some Notes on Integration Preconditioning with Pseudospectral Integration Matrices

Michael Daniel V. Samson*
daniel.samson@digipen.edu
DigiPen Institute of Technology Singapore
Singapore, Singapore

ABSTRACT

This details the comparison between the integration preconditioning in [4, 8, 9] and the preconditioning enabled by pseudospectral integration matrices (PSIMs) [10, Sec. 3]. The preceding text established well-conditioned collocation methods for second-order boundary value problems (BVPs), to develop PSIMs as the core of efficient and stable well-conditioned collocation schemes: as in [1], evaluating Birkhoff interpolation basis polynomials $\{B_k\}$, which result from particular Birkhoff interpolation problems that incorporate boundary data from the differential equation, at the collocation points gives PSIMs; when using spectral collocation points, these values can be derived in an efficient and stable manner. PSIMs are then used in the collocation scheme corresponding to this basis (herein referred to as BCOL) for these differential equations.

KEYWORDS

interpolation, differential equation, preconditioner, Birkhoff interpolation

1 PSEUDOSPECTRAL DIFFERENTIATION MATRIX

The pseudospectral differentiation matrix (PSDM) is an essential building block for collocation methods. Let $\{x_j\}_{j=0}^N$ (with $x_0 = -1$ and $x_N = 1$), and let $\{L_j\}_{j=0}^N$ be the *Lagrange interpolation basis polynomials* such that $L_j \in \mathbb{P}_N$ and $L_j(x_i) = \delta_{ij}$, for $0 \le i, j \le N$, where δ_{ij} is 1 if $i = j$ and 0 otherwise. Recall that

$$L_j(x) = \frac{q(x)}{(x - x_j)q'(x_j)}, \quad \text{where} \quad q(x) = c\prod_{j=0}^{N}(x - x_j), c \neq 0. \quad (1)$$

We have

$$p(x) = \sum_{j=0}^{N} p(x_j)L_j(x), \quad \forall p \in \mathbb{P}_N. \quad (2)$$

Denoting $d_{ij}^{(k)} := L_j^{(k)}(x_i)$, we introduce the matrices

$$\boldsymbol{D}^{(k)} = [d_{ij}^{(k)}]_{0\le i,j\le N}, \qquad \boldsymbol{D}_{\text{in}}^{(k)} = [d_{ij}^{(k)}]_{0<i,j<N}, \quad k \ge 1.$$

Note that $\boldsymbol{D}_{\text{in}}^{(k)}$ is obtained by deleting the last and first rows and columns of $\boldsymbol{D}^{(k)}$, so it is associated with interior points. In particular, we denote $\boldsymbol{D} := \boldsymbol{D}^{(1)}$, and $\boldsymbol{D}_{\text{in}} := \boldsymbol{D}_{\text{in}}^{(1)}$. The matrix $\boldsymbol{D}^{(k)}$ is usually referred to as the kth order PSDM. We highlight the following property (see e.g., [11, Thm. 3.10]):

$$\boldsymbol{D}^{(k)} = \overbrace{\boldsymbol{D}\boldsymbol{D}\cdots\boldsymbol{D}}^{k \text{ copies}} = \boldsymbol{D}^k, \quad k \ge 1, \quad (3)$$

so the higher-order PSDM is a product of the first-order PSDM.

Set

$$\vec{p}^{(k)} := (p^{(k)}(x_0), \dots, p^{(k)}(x_N))^t, \quad \vec{p} := \vec{p}^{(0)}.$$

By (2) and (3), the pseudospectral differentiation process is performed via

$$\boldsymbol{D}^{(k)}\vec{p} = \boldsymbol{D}^k\vec{p} = \vec{p}^{(k)}, \quad k \ge 1. \quad (4)$$

Remark 1. *Differentiation via* (4) *suffers from significant round-off errors for large N, due to the involvement of ill-conditioned operations* (cf. [14]).

The matrix $\boldsymbol{D}^{(k)}$ is singular ($\boldsymbol{D}^{(k)}\vec{1}^t = \vec{0}^t$, where $\vec{1} = (1, 1, \dots, 1)$, so the rows of $\boldsymbol{D}^{(k)}$ are linearly dependent), while $\boldsymbol{D}_{\text{in}}^{(k)}$ is nonsingular. In addition, the condition numbers of $\boldsymbol{D}_{\text{in}}^{(k)}$ and $\boldsymbol{D}^{(k)} - \boldsymbol{I}_{N+1}$, where $\boldsymbol{I}_m$ is the $m \times m$ identity matrix, behave like $O(N^{2k})$.

2 AUGMENTED AND TRUNCATED SOLVERS

The following are numerical solvers for the Helmholtz problem

$$\text{Find } u: \qquad -u'' + ku = f \text{ on } I, \qquad \mathcal{B}_\pm u = u(\pm 1) = u_\pm, \quad (5)$$

using PSDM, where the data provided is $\vec{f} = (f(x_1), \dots, f(x_{N-1}))^t$ and $u_\pm$, and the solution is $\vec{u} = (u_N(x_1), \dots, u_N(x_{N-1}))^t$.

Solver 2.1. *Given $u_\pm$, $\vec{f}$ and $f(\pm 1)$, solve for $\vec{u}$:*

$$(-\mathbf{D}^{(2)} + k\mathbf{I}_{N+1})\begin{bmatrix} u_- \\ \vec{u} \\ u_+ \end{bmatrix} = \begin{bmatrix} f(-1) \\ \vec{f} \\ f(1) \end{bmatrix}.$$

Spectral solvers only asymptotically approach f on the boundaries [11, Sec. 4.3], so the values $f(\pm 1)$ are not required and are usually not available; *satisfactory* schemes require only the boundary conditions $u_\pm$ and $\vec{f}$.

Remark 2. *Solver 2.1 is not satisfactory.*

Preconditioning a solver, such as Solver 2.1, is to premultiply the system so that the leading differential operator $\boldsymbol{D}^{(2)}$ approximates $\boldsymbol{I}_{N+1}$.

Solver 2.2 provides the so-called τ-method for solving (5): the first and last equations of the system in Solver 2.1, $-u_N''(\pm 1) + \gamma u_N(\pm 1) = f(\pm 1)$, are replaced by equations for the boundary conditions, $-\mathcal{B}_\pm u_N = -u_N(\pm 1) = -u_\pm$.

Solver 2.2. *Given $u_\pm$ and $\vec{f}$, solve for $\vec{u}$:*

$$\left(-\widetilde{\boldsymbol{D}}^{(2)} + k\begin{bmatrix} 0 & \vec{0} & 0 \\ \vec{0}^t & \boldsymbol{I}_{N-1} & \vec{0}^t \\ 0 & \vec{0} & 0 \end{bmatrix}\right)\begin{bmatrix} u_- \\ \vec{u} \\ u_+ \end{bmatrix} = \begin{bmatrix} -u_- \\ \vec{f} \\ -u_+ \end{bmatrix},$$

*While working at Nanyang Technological University where $\widetilde{\boldsymbol{D}}^{(2)}$ is given by

$$\widetilde{\boldsymbol{D}}^{(2)} \text{ is } \boldsymbol{D}^{(2)} \text{ with the first row replaced by } \vec{e}_0 = (1, \vec{0}) \text{ and the last row replaced by } \vec{e}_N = (\vec{0}, 1). \tag{6}$$

Remark 3. *Solver 2.2 is satisfactory and can be used to handle other boundary conditions $\mathcal{B}_\pm$ by modifying the first and last rows of $\widetilde{\boldsymbol{D}}^{(2)}$ (see, e.g. [11, Ch 4.3]).*

Both Solver 2.1 and Solver 2.2 can be *truncated to interior systems*, i.e., the first and last equations of the system are removed.

Solver 2.3. *Given $u_\pm$ and $\vec{f}$, solve for $\vec{u}$:*

$$(-\boldsymbol{D}_{\text{in}}^{(2)} + k\boldsymbol{I}_{N-1})\vec{u} = \vec{f} + u_- \vec{d}_0^{(2)} + u_+ \vec{d}_N^{(2)},$$

where $\vec{d}_j = (L_j'(x_1), \dots, L_j'(x_{N-1}))^t$.

Remark 4. *Truncated solvers are always satisfactory.*

Remark 5. *Solvers 2.1–2.2 are* augmented *systems of Solver 2.3—that is, they have the same solutions, even though Solvers 2.1–2.2 use more equations.*

3 PSEUDOSPECTRAL INTEGRATION MATRIX AS PRECONDITIONER

Consider the Birkhoff interpolation problem on $-1 = x_0 < x_1 < \cdots < x_N = 1$:

$$\begin{cases} \text{Find } p \in \mathbb{P}_N \text{ such that for any } u \in C^2(I), \\ p(-1) = u(-1); p''(x_j) = u''(x_j),\ 0 < j < N; p(1) = u(1). \end{cases} \tag{7}$$

The Birkhoff interpolation polynomial p of u can be uniquely determined by

$$p(x) = u(-1)B_0(x) + \sum_{j=1}^{N-1} u''(x_j)B_j(x) + u(1)B_N(x),\ x \in I, \tag{8}$$

where $I = [-1, 1]$, if one can find $\{B_j\}_{j=0}^N \subseteq \mathbb{P}_N$, such that

$$\begin{aligned} &B_0(-1) = 1, \quad B_0(1) = 0, \quad B_0''(x_i) = 0, \quad 0 < i < N; \\ &B_j(-1) = 0, \quad B_j(1) = 0, \quad B_j''(x_i) = \delta_{ij}, \quad 0 < i, j < N; \\ &B_N(-1) = 0, \quad B_N(1) = 1, \quad B_N''(x_i) = 0, \quad 0 < i < N. \end{aligned} \tag{9}$$

We call $\{B_j\}_{j=0}^N$ the *Birkhoff interpolation basis polynomials* of (7), which are the counterpart of the Lagrange basis polynomials $\{L_j\}_{j=0}^N$ (1).

Let $b_{ij}^{(k)} := B_j^{(k)}(x_i)$, and define the matrices

$$\boldsymbol{B}^{(k)} = [b_{ij}^{(k)}]_{0 \le i,j \le N}, \qquad \boldsymbol{B}_{\text{in}}^{(k)} = [b_{ij}^{(k)}]_{0<i,j<N},\ k \ge 0.$$

In particular, denote $b_{ij} := B_j(x_i)$, $\boldsymbol{B} = \boldsymbol{B}^{(0)}$ and $\boldsymbol{B}_{\text{in}} = \boldsymbol{B}_{\text{in}}^{(0)}$.

We have the following analogue of (3), and this approach leads to the exact inverse of second-order PSDM associated with the interior interpolation points. The last assertion is indispensable for optimally preconditioning the collocation systems.

Theorem 1. *There hold*

$$\boldsymbol{B}^{(k)} = \boldsymbol{D}^{(k)}\boldsymbol{B} = \boldsymbol{D}^k\boldsymbol{B} = \boldsymbol{D}\boldsymbol{B}^{(k-1)}, \quad k \ge 1,$$

and

$$\boldsymbol{D}_{\text{in}}^{(2)}\boldsymbol{B}_{\text{in}} = \boldsymbol{I}_{N-1}, \qquad \widetilde{\boldsymbol{D}}^{(2)}\boldsymbol{B} = \boldsymbol{I}_{N+1}, \tag{10}$$

where $\boldsymbol{I}_M$ is an $M \times M$ identity matrix, and the matrix $\widetilde{\boldsymbol{D}}^{(2)}$ is defined in (6).

In view of Thm. 1, we call $\boldsymbol{B}$ and $\boldsymbol{B}^{(1)}$ the second-order and first-order *pseudospectral integration matrices* (PSIMs), respectively. It is useful to mention the works [5, 12] discussed the inverse of the pseudospectral differential matrix (PSDM) from a different perspective.

Remark 6. *We note that*

$$\boldsymbol{B} = \begin{bmatrix} 1 & \vec{0} & 0 \\ \dfrac{\vec{1}^t - \vec{x}^t}{2} & \boldsymbol{B}_{\text{in}} & \dfrac{\vec{1}^t + \vec{x}^t}{2} \\ 0 & \vec{0} & 1 \end{bmatrix}, \qquad \boldsymbol{B}^{(1)} = \begin{bmatrix} -\frac{1}{2} & \bar{b}_0^{(1)} & \frac{1}{2} \\ -\dfrac{\vec{1}^t}{2} & \boldsymbol{B}_{\text{in}}^{(1)} & \dfrac{\vec{1}^t}{2} \\ -\frac{1}{2} & \bar{b}_N^{(1)} & \frac{1}{2} \end{bmatrix},$$

where

$$\vec{1} = (1, \dots, 1), \quad \vec{x} = (x_1, \dots, x_{N-1}),$$
$$\bar{b}_k^{(1)} = (B_1'(x_k), \dots B_{N-1}'(x_k)), \quad \boldsymbol{B}_{\text{in}}^{(1)} = (\bar{b}_1^{(1)}, \dots, \bar{b}_{N-1}^{(1)})^t.$$

For the rest of the Solvers, we employ the use of PSIM as a preconditioner to the previous Solvers (also, in (11), as the matrix for the Birkhoff interpolation basis polynomials).

Remark 7. *Different from [4, 8, 9], we* invert *the highest differentiation matrix of Solver 2.2, $\boldsymbol{D}_{\text{in}}^{(2)}$ (i.e., unknowns at interior points, for Dirichlet boundary) or $\widetilde{\boldsymbol{D}}^{(2)}$ (for all non-Neumann boundary conditions), using* (10)*, rather than $\boldsymbol{D}^{(2)}$. Moreover, the boundary conditions are imposed exactly (see [10, Sec. 3.5], Sec. 7 for general mixed boundary conditions), rather than using the penalty method [9] or auxiliary equations [4].*

Consequently, our approach leads to optimal IPs (due to invertibility of $\boldsymbol{D}_{\text{in}}^{(2)}$ and $\widetilde{\boldsymbol{D}}^{(2)}$) and well-conditioned preconditioned systems (due to eigenanalysis [10, Prop. 3.5, Rem. 3.9]).

The Birkhoff interpolation matrix $\boldsymbol{B}$ can be used as a preconditioner of Solver 2.1, as in [9]:

$$\begin{aligned} (-\boldsymbol{I}_{N+1} + k\boldsymbol{B}) \begin{bmatrix} u_- \\ \vec{u} \\ u_+ \end{bmatrix} &= \boldsymbol{B} \begin{bmatrix} f(-1) \\ \vec{f} \\ f(1) \end{bmatrix} - c_B^+ \vec{1}^t - c_B^- \vec{x}^t \\ &= \boldsymbol{B} \begin{bmatrix} f(-1) - (u_- - u_N''(-1)) \\ \vec{f} \\ f(1) - (u_+ - u_N''(1)) \end{bmatrix}, \end{aligned}$$

where

$$c_B^\pm = \frac{[u_+ \pm u_-] - [u_N''(1) \pm u_N''(-1)]}{2},$$
$$\vec{x} = (x_0 = -1, \dots, x_N = 1).$$

Noting that the first and the last equations are, by (5),

$$u_N''(\pm 1) = u''(\pm 1) = ku_\pm - f(\pm 1) \Longrightarrow (k-1)u_\pm = f(\pm 1) - u_\pm + u_N''(\pm 1),$$

Solver 3.4. *Given $\vec{f}$ and $u_\pm$, solve for $\vec{u}$:*

$$(-\boldsymbol{I}_{N+1} + k\boldsymbol{B}) \begin{bmatrix} u_- \\ \vec{u} \\ u_+ \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} (k-1)u_- \\ \vec{f} \\ (k-1)u_+ \end{bmatrix}.$$

Solver 3.4 can be truncated:

Solver 3.5. *Given $\vec{f}$ and $u_\pm$, solve for $\vec{u}$:*

$$(-\boldsymbol{I}_{N-1} + k\boldsymbol{B}_{\text{in}})\vec{u} = \boldsymbol{B}_{\text{in}}\vec{f} - u_+\vec{b}_N - u_-\vec{b}_0.$$

Remark 8. *If we start with the truncated system in Birkhoff coefficients*

$$(-\boldsymbol{I}_{N-1} + k\boldsymbol{B}_{\text{in}})\vec{v} = \vec{f} - ku_+\vec{b}_N - ku_-\vec{b}_0, \tag{11}$$

such that $\vec{u}$ is derived from $\vec{v}$, preconditioning (11) *by $\boldsymbol{B}_{\text{in}}$ results in Solver 3.5.*

Remark 9. *Using $\boldsymbol{B}_{\text{in}}$ to precondition Solver 2.3 also gives Solver 3.5, as [10, eq. (3.25)] is preconditioned to [10, eq. (3.26)], since $\vec{b}_0 + \boldsymbol{B}_{\text{in}}\vec{d}_0^{(2)} = \vec{b}_N + \boldsymbol{B}_{\text{in}}\vec{d}_N^{(2)} = \vec{0}^t$.*

The system Solver 2.2 has a leading differential term $\widetilde{\boldsymbol{D}}^{(2)}$ that is nonsingular, and premultiplying Solver 2.2 by $\boldsymbol{B} = (\widetilde{\boldsymbol{D}}^{(2)})^{-1}$ gives:

Solver 3.6. *Given $\vec{f}$ and $u_\pm$, solve for $\vec{u}$:*

$$\left(-\boldsymbol{I}_{N+1} + k\begin{bmatrix} 0 & \vec{0} & 0 \\ \vec{0}^t & \boldsymbol{B}_{\text{in}} & \vec{0}' \\ 0 & \vec{0} & 0 \end{bmatrix}\right)\begin{bmatrix} u_- \\ \vec{u} \\ u_+ \end{bmatrix} = \boldsymbol{B}\begin{bmatrix} -u_- \\ \vec{f} \\ -u_+ \end{bmatrix}.$$

Remark 10. *Truncating Solver 3.6 gives Solver 3.5. Hence, Solver 3.4 and Solver 3.6 are augmentated systems of [10, eq. (3.25)].*

Remark 11. *Solvers 2.2 and 3.6 (thus, Solver 3.5) arise from τ-method for* any non-Neumann boundary condition. *Solver 2.2 cannot be truncated for non-Dirichlet boundary conditions, as the system then solves for $u(\pm 1)$, whereas the use of PSIM allows the truncation from Solver 3.6 to Solver 3.5 by moving boundary data to the right-hand side, even for non-Dirichlet boundary conditions—if values for $u(\pm 1)$ are required, they can be attained by solving* (11) *for coefficients used in [10, eq. (3.2)]. Note that* (11) *is the same linear system as that of Solver 3.5, with different right-hand side data.*

To summarize, when solving (5): using the Lagrange basis polynomials, two augmented systems (Solvers 2.1–2.2) can be truncated to the LCOL scheme Solver 2.3. By preconditioning the augmented systems by $\boldsymbol{B}$, two augmented systems (Solver 3.4 and Solver 3.6) result, and can be truncated to the preconditioned LCOL scheme Solver 3.5. Preconditiong the LCOL scheme and the BCOL scheme (11) by $\boldsymbol{B}_{\text{in}}$ result in the preconditioned LCOL scheme. All the Solvers generate the same solution.

4 LEGENDRE-GAUSS-LOBATTO COLLOCATION

To properly compare the integration preconditioners, we contextualize the problem using Legendre-Gauss-Lobatto (LGL) collocation. We collect below some properties of Legendre polynomials (see e.g., [6, 11, 13]) to be used throughout this text.

Let $P_k(x)$ be the Legendre polynomial of degree k. Legendre polynomials are mutually orthogonal:

$$\int_{-1}^{1} P_k(x)P_j(x)\,\mathrm{d}x = \gamma_k\delta_{kj} \tag{12}$$

with $\gamma_k = 1/(2k+1)$. There hold

$$\begin{aligned} P_k(x) &= \frac{1}{2k+1}(P'_{k+1}(x) - P'_{k-1}(x)), \quad k \geq 1, \\ P_k(\pm 1) &= (\pm 1)^k, \quad P'_k(\pm 1) = \frac{1}{2}(\pm 1)^{k-1}k(k+1). \end{aligned} \tag{13}$$

Also,

$$(1-x^2)P'_k(x) = \frac{k(k+1)}{2k+1}[P_{k-1}(x) - P_{k+1}(x)].$$

LGL points are zeros of $(1-x^2)P'_N(x)$, and the corresponding quadrature weights are

$$\omega_j = \frac{2}{N(N+1)}\frac{1}{P_N^2(x_j)}, \quad 0 \leq j \leq N. \tag{14}$$

Then the LGL quadrature has the exactness

$$\int_{-1}^{1}\phi(x)\,\mathrm{d}x = \sum_{j=0}^{N}\phi(x_j)\omega_j, \forall\phi \in \mathbb{P}_{2N-1}; \sum_{j=0}^{N}[P_N(x_j)]^2\omega_j = \frac{2}{N}. \tag{15}$$

5 PSEUDOSPECTRAL INTEGRATION MATRIX FOR LGL

Let $u''(x) \equiv 1$ in (8): by the uniqueness of the interpolant (7)

$$B_0(x) = \frac{1-x}{2}, \qquad B_N(x) = \frac{1+x}{2}, \qquad \sum_{k=1}^{N-1} B_i(x) = \frac{x^2-1}{2}.$$

Let $\{x_j\}_{j=0}^N$ be LGL points. For convenience, we introduce integral operators that will differ slightly from those in [10, Sec. 3], which are based on

$$\partial_x^{-1}u(x) = \int_{-1}^{x} u(t)\,\mathrm{d}t; \qquad \partial_x^{-m}u(x) = \partial_x^{-1}\left(\partial_x^{1-m}u(x)\right), \quad m \geq 2.$$

By (13),

$$\partial_x^{-1}P_k(x) = \frac{1}{2k+1}[P_{k+1}(x) - P_{k-1}(x)], \quad k > 0; \tag{16}$$

$$\begin{aligned} \partial_x^{-2}P_k(x) &= \frac{1}{(2k+1)(2k+3)}[P_{k+2}(x) - P_k(x)], k = 0, 1. \\ \partial_x^{-2}P_k(x) &= \frac{P_{k+2}(x)}{(2k+1)(2k+3)} - \frac{2P_k(x)}{(2k-1)(2k+3)} + \frac{P_{k-2}(x)}{(2k-1)(2k+1)}, \end{aligned} \tag{17}$$

for $k > 1$. Note that, for each defined integral operator, $\partial_x^{-m}P_k(\pm 1) = 0$ while retaining $\partial_x^m[\partial_x^{-m}P_k] = P_k$.

Proposition 1 (Birkhoff interpolation at LGL points). *Let $\{x_j, \omega_j\}_{j=0}^N$ be the LGL points and weights given in* (14). *Then the Birkhoff interpolation basis polynomials* (9) *can be computed by*

$$\begin{aligned} B_j(x) &= \sum_{k=0}^{N-2}\frac{\omega_j}{\gamma_k}[P_k(x_j) - P_{N-(N+k) \bmod 2}(x_j)]\partial_x^{-2}P_k(x) \\ &= (\beta_{1j} - \beta_{0j})\frac{1+x}{2} + \sum_{k=0}^{N-2}\beta_{kj}\frac{\partial_x^{-2}P_k(x)}{\gamma_k}, \end{aligned} \tag{18}$$

where $\gamma_k = 2/(2k+1)$, $\partial_x^{-2}P_k(x)$ is given in (17), *and*

$$\beta_{kj} = \left(P_k(x_j) - \frac{1-(-1)^{N+k}}{2}P_{N-1}(x_j) - \frac{1+(-1)^{N+k}}{2}P_N(x_j)\right)\omega_j.$$

From (16)–(17),

$$\partial_x^{-2} P_k(x) = \begin{cases} \dfrac{1}{(2k+1)}[\partial_x^{-1}P_{k+1}(x) - \partial_x^{-1}P_{k-1}(x)], & k > 1; \\ \dfrac{1}{(2k+1)}\partial_x^{-1}P_{k+1}(x), & k = 0, 1. \end{cases}$$

Thus,

$$B_j(x) = -(1-x^2)(1-x_j^2)\omega_j \sum_{k=1}^{N-1} \frac{P_k'(x_j)P_k'(x)}{\gamma_k k^2(k+1)^2},$$

which implies $B_j(x_i)\omega_i = B_i(x_j)\omega_j$, $0 < i, j < N$.

Remark 12. *Considering that* $\{x_j\}_{j=1}^{N-1}$ *are roots of* $P_{N-1}^{1,1}(x)$*, the Jacobi (Gegenbauer) polynomial (see, e.g. [11, Ch. 3]) associated with the weight* $\omega^{1,1} = (1-x^2)$ *and Jacobi-Gauss quadrature weights* $\{\omega_j^{1,1}\}_{j=1}^{N-1}$*, we have*

$$B_j''(x) = \sum_{k=0}^{N-2} \frac{\omega_j^{1,1}}{\gamma_k^{1,1}} P_k^{1,1}(x_j)P_k^{1,1}(x),$$

$$B_j(x) = (x^2-1)\sum_{k=0}^{N-2} \frac{\omega_j^{1,1}}{\gamma_k^{1,1}} \frac{P_k^{1,1}(x_j)P_k^{1,1}(x)}{(k+1)(k+2)},$$

where $\gamma_k^{1,1} = 8(k+1)/(2k+3)(k+2)$.

We briefly examine the essential idea of constructing integration preconditioners in [4, 9] (inspired by [2, 3]) for the Legendre case. By (12) and (15),

$$L_j(x) = \sum_{k=0}^{N} \frac{\omega_j}{\tilde{\gamma}_k} P_k(x_j)P_k(x), \quad 0 \le j \le N, \tag{19}$$

where $\tilde{\gamma}_k = 2/(2k+1)$ for $0 \le k < N$, and $\tilde{\gamma}_N = 2/N$. This follows from letting

$$L_j(x) = \sum_{k=0}^{N} \alpha_{jk}P_k(x), \quad \text{where} \quad \alpha_{jk} = \frac{1}{\tilde{\gamma}_k}\int_{-1}^{1} L_j(x)P_k(x)\,\mathrm{d}x.$$

Then

$$L_j''(x) = \sum_{k=2}^{N} \frac{\omega_j}{\tilde{\gamma}_k} P_k(x_j)P_k''(x). \tag{20}$$

The key observation in [4, 9] (also see [2, 3]) is that *the pseudo-spectral differentiation process actually involves the ill-conditioned transform:*

$$\mathrm{span}\{P_k'' : 2 \le k \le N\} := Q_2^N \longmapsto Q_0^{N-2} := \mathrm{span}\{P_k : 0 \le k \le N-2\}. \tag{21}$$

Indeed, we have (see [11, (3.176c)]):

$$P_k''(x) = \sum_{\substack{0\le l\le k-2 \\ k+l \text{ even}}} (l+1/2)(k(k+1) - l(l+1))P_l(x),$$

so the transform matrix is dense and the coefficients grow like k^2.

However, *the inverse transform* $Q_0^{N-2} \mapsto Q_2^N$ *is sparse and stable,* thanks to the "compact" formula, derived from (13):

$$P_k(x) = \alpha_k P_{k-2}''(x) + \beta_k P_k''(x) + \alpha_{k+1}P_{k+2}''(x), \quad k \ge 2, \tag{22}$$

where the coefficients are

$$\alpha_k = \frac{1}{(2k-1)(2k+1)}, \qquad \beta_k = -\frac{2}{(2k-1)(2k+3)},$$

which decay like k^{-2}.

6 DECOMPOSITION OF PRECONDITIONERS

We introduce the following matrices, as in [4]:

$$\boldsymbol{T} = [t_{kj} := \omega_j P_k(x_j)/\gamma_k]_{0\le k,j\le N}, \quad \boldsymbol{P} = [p_{ik} := P_k(x_i)]_{0\le i,k\le N},$$
$$\widehat{\boldsymbol{T}} = [t_{ki}]_{0\le i\le N}^{0\le k\le N-2}, \quad \widehat{\boldsymbol{P}} = [p_{ik}]_{0\le k\le N-2}^{0\le i\le N},$$
$$\widetilde{\boldsymbol{T}} = [t_{k+2,i}]_{0\le i\le N}^{0\le k\le N-2}, \quad \widetilde{\boldsymbol{P}} = [p_{i,k+2}]_{0\le k\le N-2}^{0\le i\le N}.$$

Then there holds

$$\boldsymbol{T}\boldsymbol{P} = \boldsymbol{I}_{N+1}, \quad \widehat{\boldsymbol{T}}\widehat{\boldsymbol{P}} = \boldsymbol{I}_{N-1} = \widetilde{\boldsymbol{T}}\widetilde{\boldsymbol{P}} \quad \text{but} \quad \widehat{\boldsymbol{P}}\widehat{\boldsymbol{T}} \ne \boldsymbol{I}_{N+1} \ne \widetilde{\boldsymbol{P}}\widetilde{\boldsymbol{T}},$$

which follows from (15) and (19). We remark that, with a re-normalization of $\{P_k\}$, we have $\boldsymbol{P} = \boldsymbol{T}^t$. By (20), we obtain

$$\boldsymbol{D}^{(2)} = \widetilde{\boldsymbol{P}}^{(2)}\widetilde{\boldsymbol{T}} \quad \text{where} \quad \widetilde{\boldsymbol{P}}^{(2)} = [p_{ik}^{(2)} = P_{k+2}''(x_i)]_{0\le k\le N-2}^{0\le i\le N}.$$

On the other hand, (22) leads to $\widehat{\boldsymbol{P}} = \widetilde{\boldsymbol{P}}^{(2)}\boldsymbol{A}$, where $\boldsymbol{A}$ is a sparse matrix formed by the coefficients $\{\alpha_k, \beta_k\}$.

Similar to the notion in (21), the ill-conditioned differentiation process is actually observed from the identity $\widetilde{\boldsymbol{D}}^{(2)}\boldsymbol{P} = \bar{\boldsymbol{P}}^{(2)}$ (to be defined shortly). Indeed, we obtain from (7)–(8) that, for $0 \le i, k \le N$,

$$p_{ik} := P_k(x_i) = P_k(x_0)B_0(x_i) + \sum_{j=1}^{N-1} P_k''(x_j)B_j(x_i) + P_k(x_N)B_N(x_i)$$
$$= \sum_{j=0}^{N} b_{ij}^{(0)}\bar{p}_{jk}^{(2)},$$

which implies $\boldsymbol{P} = \boldsymbol{B}\bar{\boldsymbol{P}}^{(2)}$ with $\bar{\boldsymbol{P}}^{(2)} = [\bar{p}_{jk}^{(2)}]_{0\le j,k\le N}$. As $\widetilde{\boldsymbol{D}}^{(2)}\boldsymbol{B} = \boldsymbol{I}_{N+1}$ (cf. Thm. 1), we have $\widetilde{\boldsymbol{D}}^{(2)}\boldsymbol{P} = \bar{\boldsymbol{P}}^{(2)}$ (*which performs second-order differentiation at the interior points, but preserves the function values at the endpoints*).

Remark 13. *Another way of looking at* $\bar{\boldsymbol{P}}^{(2)} = \widetilde{\boldsymbol{D}}^{(2)}\boldsymbol{P}$*: let* $\{\phi_i(x)\}_{i=0}^{N} \subset \mathbb{P}_N$ *be defined as*

$$\phi_0(x) = L_N(x) + L_0(x); \qquad \phi_1(x) = L_N(x) - L_0(x);$$
$$\text{for } 0 < j \le N/2, \quad \phi_{2j+c}(x) = P_{2j+c}''(x) + [1 - P_{2j+c}''(1)]\phi_c(x),$$

where $c = 0, 1$*. Then* $\bar{\boldsymbol{P}}^{(2)} = [\phi_j(x_i)]_{0\le i,j\le N}$.

On the other hand, we infer from (18) and the three-term recurrence formula of Legendre polynomials (see e.g., [13]) that we can formulate the computation of $\boldsymbol{B}$ as the transform $\boldsymbol{B} = \boldsymbol{P}\boldsymbol{M}$, where the linear transform matrix $\boldsymbol{M}$ is formed by linear combinations of $\{\beta_{kj}, \hat{\beta}_{kj}\}$ in Thm. 1. It follows from $\boldsymbol{P} = \boldsymbol{B}\bar{\boldsymbol{P}}^{(2)}$ and $\boldsymbol{B} = \boldsymbol{P}\boldsymbol{M}$ that $\boldsymbol{M} = (\bar{\boldsymbol{P}}^{(2)})^{-1}$. Its implication is twofold: (i) the computation of $\boldsymbol{B}$ is a stable and well-conditioned integration process; and (ii) $\boldsymbol{P}\boldsymbol{M}$ is an optimal integration preconditioner for second-order equations with Dirichlet boundary conditions (e.g., Solver 2.2).

Let $\vec{x} = (x_1, \ldots, x_{N-1})$. The literature provides three distinct techniques used to solve (5):

- Greengard [8] makes use of an intermediate function $\sigma(x) = u''(x)$, such that $\vec{u} = \boldsymbol{P}\widehat{\boldsymbol{A}}\vec{\sigma} + c_{\mathrm{G}}^1\vec{x}^t + c_{\mathrm{G}}^0\vec{1}^t$, where $\widehat{\boldsymbol{A}}$ is an $(N+1)\times(N+1)$ matrix with $\boldsymbol{A}$ in its lower-left, and zeros elsewhere, and σ can be interpolated ($\vec{\sigma}$) to an Nth-degree polynomial, i.e. σ is not the second derivative of $\mathrm{I}_N u$.

- Hesthaven [9] preconditions Solver 2.1 by premultiplying by $\boldsymbol{P\widetilde{A}T}$, where $\widetilde{\boldsymbol{A}}$ differs from $\widehat{\boldsymbol{A}}$ only in that the first two rows are zero except for the last two columns, such that $\widetilde{\boldsymbol{A}}$, and thus $\boldsymbol{P\widetilde{A}T}$, are nonsingular.
- Elbarbary [4] preconditions Solver 2.1 by premultiplying by $\boldsymbol{A\widehat{T}}$. The final system is then made square by adding equations that correspond to the boundary conditions, so that the left-hand side matrix is $\boldsymbol{T}$. The approach to the Helmholtz problem is detailed in [4, pp. 1196–7].

Remark 14. *Each of the methods outlined also have a corresponding approach to solving the first-order ordinary differential equation*

$$u' + \gamma u = f \text{ on } I, \qquad u(-1) = u_-,$$

although both Hesthaven's and Elbarbary's approaches, as given, prefer the right boundary condition $u(1) = u_+$.

The application of these three techniques can be cast in preconditioner form:

$$(-\boldsymbol{I}_{N+1} + k\boldsymbol{B}_\star)\begin{bmatrix} u_- \\ \vec{u} \\ u_+ \end{bmatrix} = \boldsymbol{B}_\star \begin{bmatrix} f(-1) \\ \vec{f} \\ f(1) \end{bmatrix} - c_\star^0 \vec{1}^t - c_\star^1 \vec{x}^t, \tag{23}$$

where

$$\boldsymbol{B}_\star = \boldsymbol{P}\boldsymbol{A}_\star \boldsymbol{T}$$

is what will be referred to as the preconditioner of the scheme (in the effect of preconditioning Solver 2.1 by $\boldsymbol{B}_\star$—not consistent with some of the documented preconditioners, i.e., Elbarbary's preconditioner is $\boldsymbol{A\widehat{T}}$, $\boldsymbol{A}_\mathrm{E}\boldsymbol{T}$ without the first two rows [4]), where the $\boldsymbol{A}_\star$ (the corresponding *spectral* integration preconditioners, in the vein of [2, 3]) are as follows: $\boldsymbol{A}_\mathrm{T}$ is the antiderivative matrix in spectral coefficients, thus

$$\boldsymbol{A}_\mathrm{G} = \begin{bmatrix} \vec{0} & 1 & 0 \\ \vec{a}_\mathrm{T}^{(2)} & 0 & 1 \\ \boldsymbol{A} & \vec{0}^t & \vec{0}^t \end{bmatrix},$$

$$\boldsymbol{A}_\mathrm{H} = \begin{bmatrix} \vec{0} & a_{00} & a_{01} \\ \vec{0} & a_{10} & a_{11} \\ \boldsymbol{A} & \vec{a}_{N-1}^{(2)} & \vec{a}_N^{(2)} \end{bmatrix},$$

$$\boldsymbol{A}_\mathrm{E} = \begin{bmatrix} \vec{0} & 0 & 0 \\ \vec{0} & 0 & 0 \\ \boldsymbol{A} & \vec{0}^t & \vec{0}^t \end{bmatrix}.$$

Remark 15. *Hesthaven's approach only differs from Elbarbary's on the preconditioner.*

Based on (22), [4, 8, 9] attempted to precondition the collocation system by the "inverse" of $\boldsymbol{D}^{(2)}$. However, since $\boldsymbol{D}^{(2)}$ is singular, there exist multiple ways to manipulate the involved singular matrices. The boundary conditions were imposed by the penalty method (cf. [7]) in [9], and using auxiliary equations in [4].

Remark 16. *Note that the condition number of the preconditioned system for e.g., the operator $\frac{\mathrm{d}^2}{\mathrm{d}x^2} - k$ with Dirichlet boundary conditions, behaves like $O(\sqrt{N})$.*

Rearranging the columns of (23) can construct a solvable system [4].

Solver 6.7. *Given $\vec{f}$ and $f(\pm 1)$, solve for $\vec{u}$, $c_\star^0$ and $c_\star^1$:*

$$\begin{bmatrix} \vec{1}^t & \vec{m}_1 & \cdots & \vec{m}_{N-1} & \vec{x}^t \end{bmatrix} \begin{bmatrix} c_\star^0 \\ \vec{u} \\ c_\star^1 \end{bmatrix} = \boldsymbol{B}_\star \begin{bmatrix} f(-1) \\ \vec{f} \\ f(1) \end{bmatrix} - u_- \vec{m}_0 - u_+ \vec{m}_N,$$

where $-\boldsymbol{I}_{N+1} + k\boldsymbol{B}_\star = (\vec{m}_0, \ldots, \vec{m}_N)$.

Remark 17. *The computed values of $c_\star^0$ and $c_\star^1$ in Solver 6.7 can be discarded. Elbarbary's approach is not satisfactory.*

In comparison,

$$\boldsymbol{A}_\mathrm{B} = \boldsymbol{TBP} = \boldsymbol{T}[f_j(x_i)]_{0 \le i,j \le N},$$

where, for $0 \le j \le N-2$, $\partial_x^{-2} P_k$ are as in (17) and

$$f_j(x) = \partial_x^{-2} P_j(x) + c_j^+ + c_j^- x;$$
$$c_j^\pm = \frac{1 \pm (-1)^j - \partial_x^{-2} P_j(1) \mp \partial_x^{-2} P_j(-1)}{2}.$$

and, for $j \in \{N-1, N\}$, $f_j(x) = \sum_{k=1}^{N-1} P_j(x_k) B_k(x) + [(-1)^j(1-x) + 1 + x]/2$. Thus, if $\vec{c}^\pm = (c_0^\pm, \ldots, c_{N-2}^\pm)$,

$$\boldsymbol{A}_\mathrm{B} = \boldsymbol{T}\begin{bmatrix} \partial_x^{-2}P_0(-1) & \cdots & \partial_x^{-2}P_{N-2}(-1) & (-1)^{N-1} & (-1)^N \\ \partial_x^{-2}P_0(\vec{x}^t) & \cdots & \partial_x^{-2}P_{N-2}(\vec{x}^t) & f_{N-1}(\vec{x}^t) & f_N(\vec{x}^t) \\ \partial_x^{-2}P_0(1) & \cdots & \partial_x^{-2}P_{N-2}(1) & 1 & 1 \end{bmatrix}$$
$$+ \boldsymbol{T}[\vec{1}^t\ \vec{x}^t] \begin{bmatrix} \vec{c}^+ & 0 & 0 \\ \vec{c}^- & 0 & 0 \end{bmatrix} = \begin{bmatrix} \vec{c}^+ & f_{00} & f_{01} \\ \vec{c}^- & f_{10} & f_{11} \\ \boldsymbol{A} & \vec{f}_0 & \vec{f}_1 \end{bmatrix},$$

where

$$f_{ij} = \int_{-1}^1 \frac{P_i(x)}{\gamma_i} \frac{(-1)^{N-1+j}(1-x) + (1+x)}{2} \omega \,\mathrm{d}x$$
$$+ \sum_{l=1}^{N-1} P_{N-1+j}(x_l) \int_{-1}^1 \frac{P_i(x)}{\gamma_i} B_l(x) \omega \,\mathrm{d}x.$$

The first integral is 1 if $i = 0, 1$ and $N + j + i$ is odd, and 0 otherwise.

Alternatively,

$$\boldsymbol{M} = (\bar{\boldsymbol{P}}^{(2)})^{-1} = \boldsymbol{A}_\mathrm{B}\boldsymbol{T} = \boldsymbol{WA}(\widehat{\boldsymbol{T}} - \boldsymbol{C}) + \boldsymbol{V} = \boldsymbol{WAHT} + \boldsymbol{V},$$

where $\boldsymbol{V} = [v_{ij}]_{0 \le i,j \le N}$, $\boldsymbol{C} = [c_{ij}]_{0 \le j \le N}^{0 \le i \le N-2}$,

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{I}_{N-1} \end{bmatrix}, \qquad \boldsymbol{R} = [r_{ij}]_{0 \le j \le N-2}^{0 \le i \le 1}, \qquad r_{ij} = -\frac{1 + (-1)^{i+j}}{2};$$

$$\boldsymbol{H} = [\boldsymbol{I}_{N-1}\ \boldsymbol{S}], \qquad \boldsymbol{S} = [s_{ij}]_{N-1 \le j \le N}^{0 \le i \le N-2}, \qquad s_{ij} = -\frac{\gamma_j}{\gamma_i}\frac{1 + (-1)^{i+j}}{2};$$

$$v_{ij} = \begin{cases} \frac{1}{2} & \text{if } (i,j) \in \{(0,0), (0,N), (1,N)\}, \\ -\frac{1}{2} & \text{if } (i,j) = (1,0), \\ 0 & \text{otherwise}; \end{cases}$$

$$c_{ij} = \frac{\omega_j}{\gamma_i} P_k(x_j), \qquad k = \begin{cases} N & \text{if } N + i \text{ is even}, \\ N-1 & \text{if } N + i \text{ is odd}; \end{cases}$$

hence, if $\boldsymbol{P} = [\vec{1}^t\ \vec{x}^t\ \widetilde{\boldsymbol{P}}]$, then $\boldsymbol{PW} = [\vec{1}^t\ \vec{x}^t]\boldsymbol{R} + \widetilde{\boldsymbol{P}}$ and

$$\boldsymbol{B} = \boldsymbol{B}_\mathrm{E} + [\vec{1}^t\ \vec{x}^t]\boldsymbol{RA\widehat{T}} - \boldsymbol{PWAC} + \boldsymbol{PV} = \boldsymbol{P}(\boldsymbol{WAH} + \boldsymbol{G})\boldsymbol{T},$$

where apparently $\boldsymbol{G} = [g_{ij}]$ where, for $0 \le i \le N$, $g_{ij} = [1 + (-1)^{i+j}]/2$ for $j = 0, 1$ and $g_{ij} = 0$ when $j > 1$.

The upshot of this decomposition is that $\boldsymbol{WAH} + \boldsymbol{G}$ represents a spectral integration preconditioner: precisely, since all of the spectral integration preconditioners contain $\boldsymbol{A}$, the first two rows of $\boldsymbol{WAH} + \boldsymbol{G}$ provide the spectral-τ conditions corresponding to the PSIM.

7 PSEUDOSPECTRAL INTEGRATION MATRIX FOR MIXED BOUNDARY CONDITIONS

Finally, we briefly outline an alternative formulation for the pseudo-spectral integration matrices (PSIMs) on mixed boundary conditions (cf. [10, Sec. 3.5]) based on PSIMs on Dirichlet boundary conditions (cf. [10, Sec. 3.1], Rem. 6).

Consider the mixed boundary conditions:

$$\begin{aligned} \mathcal{B}_-[u] &:= a_- u(-1) + b_- u'(-1) = c_-, \\ \mathcal{B}_+[u] &:= a_+ u(1) + b_+ u'(1) = c_+, \end{aligned} \tag{24}$$

where $a_\pm, b_\pm$ and $u_\pm$ are given constants. We first assume that

$$d := 2a_+a_- - a_+b_- + a_-b_+ \neq 0, \tag{25}$$

which excludes Neumann boundary conditions (i.e., $a_- = a_+ = 0$).

We associate (24) with the Birkhoff-type interpolation:

$$\begin{cases} \text{Find } p \in \mathbb{P}_N \text{ such that} \\ \mathcal{B}_-[p] = c_-, \quad p''(x_j) = c_j, 0 < j < N, \quad \mathcal{B}_+[p] = c_+, \end{cases} \tag{26}$$

where $\{c_\pm, c_j\}$ are given. We look for the interpolation basis polynomials, denoted by $\{\tilde{B}_j\}_{j=0}^N$, satisfying

$$\begin{aligned} &\mathcal{B}_-[\tilde{B}_0] = 1, \quad \tilde{B}_0''(x_i) = 0, \quad 0 < i < N, \quad \mathcal{B}_+[\tilde{B}_0] = 0; \\ &\mathcal{B}_-[\tilde{B}_j] = 0, \quad \tilde{B}_j''(x_i) = \delta_{ij}, \quad 0 < i < N, \quad \mathcal{B}_+[\tilde{B}_j] = 0, \quad 0 < j < N; \\ &\mathcal{B}_-[\tilde{B}_N] = 0, \quad \tilde{B}_N''(x_i) = 0, \quad 0 < i < N, \quad \mathcal{B}_+[\tilde{B}_N] = 1. \end{aligned} \tag{27}$$

The proof of existence of these basis functions is similar to that in [10, Thm. 3.1]. Let $\widetilde{\boldsymbol{B}} = [\tilde{B}_j(x_i)]$ and $\widetilde{\boldsymbol{B}}^{(k)} = [\tilde{B}_j^{(k)}(x_i)]$.

Remark 18. *Analogous to Thm. 1, for the corresponding $\widetilde{\boldsymbol{D}}^{(2)}$ τ-method matrix, i.e.*

$\widetilde{\boldsymbol{D}}^{(2)}$ *is* $\boldsymbol{D}^{(2)}$

with the first row replaced by the first row of $a_-\boldsymbol{I}_{N+1} + b_-\boldsymbol{D}$

and the last row replaced by the last row of $a_+\boldsymbol{I}_{N+1} + b_+\boldsymbol{D}$.

Then, $\widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{D}}^{(2)} = \boldsymbol{I}_{N+1}$. However, the truncated analogue (cf. $\boldsymbol{B}_{\text{in}}\boldsymbol{D}_{\text{in}}^{(2)} = \boldsymbol{I}_{N-1}$) only holds for Dirichlet conditions, i.e. $a_\pm = 1$, $b_\pm = 0$.

Using the observation in Rem. 18, we can determine the matrices

$$\widetilde{\boldsymbol{B}} = \begin{bmatrix} \frac{2a_+ + b_+}{d} & \widehat{b}_0 & -\frac{b_-}{d} \\ \dfrac{a_+(\vec{1}^t - \vec{x}^t) + b_+}{d} & \widetilde{\boldsymbol{B}}_{\text{in}} & \dfrac{a_-(\vec{1}^t + \vec{x}^t) - b_-}{d} \\ \frac{b_+}{d} & \widehat{b}_N & \frac{2a_- - b_-}{d} \end{bmatrix},$$

$$\widetilde{\boldsymbol{B}}^{(1)} = \begin{bmatrix} -\frac{a_+}{d} & \widehat{b}_0^{(1)} & \frac{a_-}{d} \\ -\dfrac{a_+\vec{1}^t}{d} & \widetilde{\boldsymbol{B}}_{\text{in}}^{(1)} & \dfrac{a_-\vec{1}^t}{d} \\ -\frac{a_+}{d} & \widehat{b}_N^{(1)} & \frac{a_-}{d} \end{bmatrix}$$

for general boundary conditions $\{a_\pm, b_\pm\}$, at least one $a_\pm \neq 0$, from $\boldsymbol{B}$ and $\boldsymbol{B}^{(1)}$.

It is easy to see that the first and last columns derivable from the corresponding columns of the analogous matrices, and the rest are derived as follows:

$$\begin{aligned} \widehat{b}_0 &= -\frac{b_-(2a_+ + b_+)}{d}\bar{b}_0^{(1)} + \frac{b_+b_-}{d}\bar{b}_N^{(1)}, \\ \widehat{b}_0^{(1)} &= \frac{a_-(2a_+ + b_+)}{d}\bar{b}_0^{(1)} - \frac{b_+a_-}{d}\bar{b}_N^{(1)}, \\ \widehat{b}_N &= -\frac{b_+b_-}{d}\bar{b}_0^{(1)} - \frac{b_+(2a_- - b_-)}{d}\bar{b}_N^{(1)}, \\ \widehat{b}_N^{(1)} &= \frac{a_+b_-}{d}\bar{b}_0^{(1)} + \frac{a_+(2a_- - b_-)}{d}\bar{b}_N^{(1)}, \\ \widetilde{\boldsymbol{B}}_{\text{in}} &= \boldsymbol{B}_{\text{in}} - \frac{1}{d}[\vec{1}^t([b_-(a_+ + b_+)]\bar{b}_0^{(1)} + [b_+(a_- - b_-)]\bar{b}_N^{(1)})] \\ &\quad - \frac{1}{d}[\vec{x}^t(a_+b_-\bar{b}_0^{(1)} - a_-b_+\bar{b}_N^{(1)})], \\ \widetilde{\boldsymbol{B}}_{\text{in}}^{(1)} &= \boldsymbol{B}_{\text{in}}^{(1)} + \frac{1}{d}\vec{1}^t(a_+b_-\bar{b}_0^{(1)} - a_-b_+\bar{b}_N^{(1)}). \end{aligned}$$

8 SOME RESULTS

As these techniques are alternative formulations, results are replicated from the author's previous publications, such as [10].

8.1 Collocation schemes

Consider the BVP

$$u''(x) + r(x)u'(x) + s(x)u(x) = f(x), \quad x \in I; \quad u(\pm 1) = u_\pm, \tag{28}$$

where the given functions $r, s, f \in C(I)$. Let $\{x_j\}_{j=0}^N$ be the set of Gauss-Lobatto points as in (7). Then the collocation scheme for (28) is to find $u_N \in \mathbb{P}_N$ such that

$$\begin{aligned} &u_N''(x_i) + r(x_i)u_N'(x_i) + s(x_i)u_N(x_i) = f(x_i), \quad 0 < i < N; \\ &u_N(\pm 1) = u_\pm. \end{aligned} \tag{29}$$

As the Birkhoff interpolation polynomial of u_N is itself, we have from (8) that

$$u_N(x) = u_-B_0(x) + u_+B_N(x) + \sum_{j=1}^{N-1} u_N''(x_j)B_j(x). \tag{30}$$

Then the matrix form of (29) reads

$$\left(-\boldsymbol{I}_{N-1} + \boldsymbol{\Lambda}_r\boldsymbol{B}_{\text{in}}^{(1)} + \boldsymbol{\Lambda}_s\boldsymbol{B}_{\text{in}}\right)\vec{v} = \vec{f} - u_-\vec{v}_- - u_+\vec{v}_+, \tag{31}$$

where $\boldsymbol{I}_{N-1}$ is the $(N-1)\times(N-1)$ identity matrix, and

$$\boldsymbol{\Lambda}_r = \operatorname{diag}(r(x_1), \ldots, r(x_{N-1})), \quad \boldsymbol{\Lambda}_s = \operatorname{diag}(s(x_1), \ldots, s(x_{N-1})),$$

$$\vec{v} = (u_N''(x_1), \ldots, u_N''(x_{N-1}))^t, \quad \vec{f} = (f(x_1), \ldots, f(x_{N-1}))^t,$$

$$\vec{v}_\pm = \left(\pm\frac{r(x_1)}{2} + s(x_1)\frac{1 \pm x_1}{2}, \ldots, \pm\frac{r(x_{N-1})}{2} + s(x_{N-1})\frac{1 \pm x_{N-1}}{2}\right)^t.$$

It is seen that, under the basis $\{B_j\}$, the matrix of the highest derivative is identity, and it also allows for exact imposition of boundary conditions.

In summary, we take the following steps to solve (29):

- Pre-compute $\boldsymbol{B}$ and $\boldsymbol{B}^{(1)}$ via the formulas in Rem. 6.
- Find $\vec{v}$ by solving the system (31).

- Recover $\vec{u} = (u_N(x_1), \dots, u_N(x_{N-1}))^t$ from (30):

$$\vec{u} = \boldsymbol{B}_{\text{in}} \vec{v} + u_- \vec{b}_0 + u_+ \vec{b}_N,$$

where $\vec{b}_j = (B_j(x_1), \dots, B_j(x_{N-1}))^t$ for $j = 0, N$.

Remark 19. *The unknowns under the new basis in* (30)–(31) *are the approximations to* $\{u''(x_j)\}$. *This situation is reminiscent of the spectral integration method* [8], *which is built upon the orthogonal polynomial expansion of* $u''(x)$. *Thus, the approach can be regarded as the collocation counterpart of the modal approach in* [8].

For comparison, we look at the usual collocation scheme (29) under the Lagrange basis. Write the solution of (29) as

$$u_N(x) = u_- L_0(x) + u_+ L_N(x) + \sum_{j=1}^{N-1} u_N(x_j) L_j(x),$$

and insert it into (29), leading to the usual collocation system for (29):

$$\left(-\boldsymbol{D}_{\text{in}}^{(2)} + \boldsymbol{\Lambda}_r \boldsymbol{D}_{\text{in}}^{(1)} + \boldsymbol{\Lambda}_s\right) \vec{u} = \vec{f} + \vec{u}_B, \tag{32}$$

where $\vec{f}$ and $\vec{u}$ are as above, and $\vec{u}_B$ is the vector $\{u_-(d_{i0}^{(2)} - r(x_i) d_{i0}^{(1)}) + u_+(d_{iN}^{(2)} - r(x_i) d_{iN}^{(1)})\}_{i=1}^{N-1}$. It is known that the condition number of the coefficient matrix in (32) grows like $O(N^4)$.

The matrix $\boldsymbol{B}_{\text{in}}$ can be used to precondition the ill-conditioned system (32), leading to

$$\left(-\boldsymbol{I}_{N-1} + \boldsymbol{B}_{\text{in}} \boldsymbol{\Lambda}_r \boldsymbol{D}_{\text{in}}^{(1)} + \boldsymbol{B}_{\text{in}} \boldsymbol{\Lambda}_s\right) \boldsymbol{u} = \boldsymbol{B}_{\text{in}} (\vec{f} + \vec{u}_B). \tag{33}$$

Remark 20. *The right-hand side of* (33) *can be expanded:*

$$\boldsymbol{B}_{\text{in}}(\vec{f} + \vec{u}_B) = \boldsymbol{B}_{\text{in}} \vec{f} - u_-(\vec{b}_0 + \boldsymbol{B}_{\text{in}} \boldsymbol{\Lambda}_r \vec{d}_0^{(1)}) - u_+(\vec{b}_N + \boldsymbol{B}_{\text{in}} \boldsymbol{\Lambda}_r \vec{d}_N^{(1)}),$$

where $\vec{d}_j^{(k)} = (d_{1j}^{(k)}, d_{2j}^{(k)} \dots, d_{N-1,j}^{(k)})^t$ *for* $j = 0, N$ *and* $k = 1$. *This improves the accuracy of the resulting computation.*

To illustrate, we compare the condition numbers of the linear systems between the Lagrange collocation (LCOL) scheme (32), Birkhoff collocation (BCOL) scheme (31), preconditioned LCOL (P-LCOL) scheme (33), and the preconditioned scheme from [4] (which improved that in [9]) (PLCOL), respectively. We also look at the number of iterations for solving the systems via BiCGSTAB in Matlab, and compare their convergence behavior.

We first consider the example

$$\begin{aligned} &u''(x) - (1 + \sin x) u'(x) + e^x u(x) = f(x), \quad x \in (-1, 1); \\ &u(\pm 1) = u_\pm, \end{aligned} \tag{34}$$

with the exact solution $u(x) = e^{(x^2-1)/2}$. Observe from Table 1 that the condition numbers of two new approaches are independent of N, and do not induce round-off errors. As already mentioned, the condition number of PLCOL in [4] grows like $O(\sqrt{N})$, and that of LCOL behaves like $O(N^4)$.

In Fig. 1, we depict the distribution of the eigenvalues (in magnitude) of the coefficient matrices of BCOL, PLCOL and P-LCOL with $N = 1024$. Observe that almost all of them are concentrated around 1.

We next consider (34) with $f \in C^1(\bar{I})$ and the exact solution $u \in C^3(\bar{I})$, given by

$$u(x) = \begin{cases} \cosh(x+1) - x^2/2 - x, & -1 \le x < 0, \\ \cosh(x+1) - \cosh(x) - x + 1, & 0 \le x \le 1. \end{cases}$$

Table 1: Comparison of condition numbers, accuracy and iterations for (34).

N	LCOL (32)			PLCOL [4]		
	Cond.#	Error	iters	Cond.#	Error	iters
64	3.97e+05	3.82e-14	286	80.1	1.44e-15	14
128	6.23e+06	4.42e-13	1251	156	2.66e-15	13
256	9.91e+07	3.95e-13	6988	308	2.33e-15	13
512	1.58e+09	1.02e-11	9457	612	3.77e-15	13
N	BCOL (31)			P-LCOL (33)		
	Cond.#	Error	iters	Cond.#	Error	iters
64	6.36	5.55e-16	10	2.86	1.67e-15	8
128	6.46	1.11e-15	10	2.86	2.44e-15	8
256	6.51	1.11e-15	11	2.86	2.55e-15	8
512	6.54	1.89e-15	11	2.86	4.77e-15	8

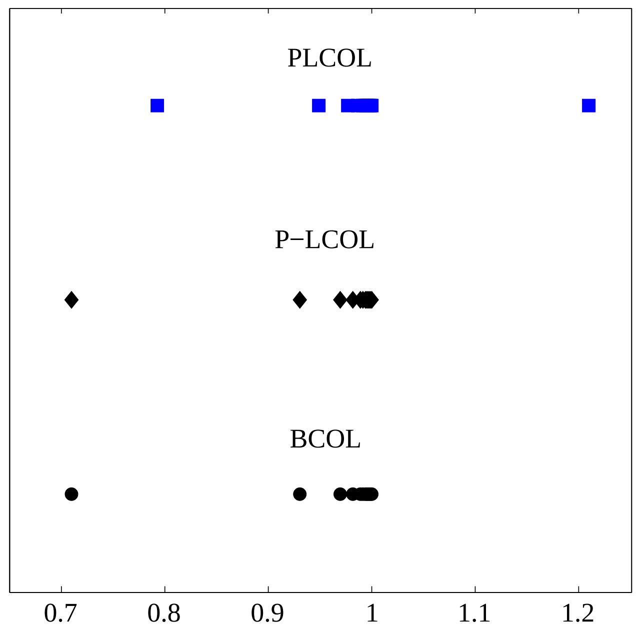

Figure 1: Distribution of magnitude of eigenvalues for the coefficient matrices of collocation schemes with $N = 1024$.

Note that u has Sobolev-regularity in $H^{4-\varepsilon}(I)$ with $\varepsilon > 0$. In Fig. 2, we graph the maximum point-wise errors for BCOL, LCOL and PLCOL, where the slope of the lines is approximately -4. We see that the BCOL and PLCOL are free of round-off errors even for thousands of points, though the PLCOL system (in [4]) has a mildly-growing condition number.

As noted in the original, the condition number of $\boldsymbol{A} = \boldsymbol{I}_{N-1} - k\boldsymbol{B}_{\text{in}}$ is independent of N.

8.2 Mixed boundary conditions

Consider the second-order BVP (28), equipped with mixed boundary conditions (24)–(25).

We associate (24) with the Birkhoff-type interpolation (26), indicating the use of interpolation basis polynomials satisfying (27).

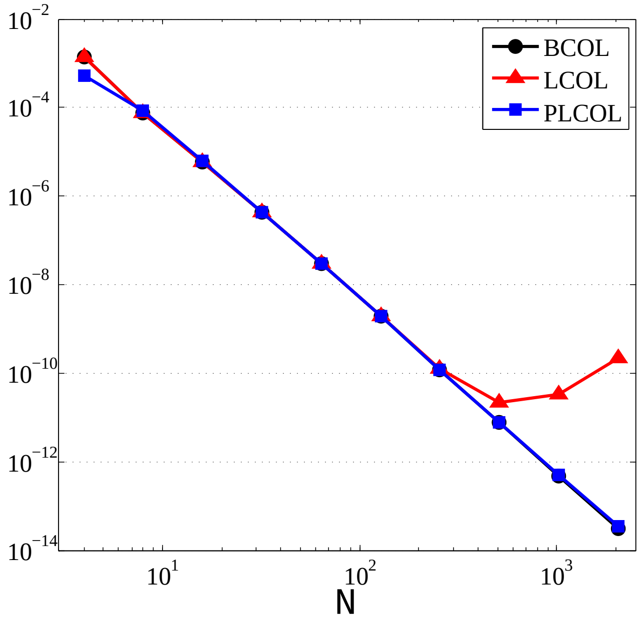

Figure 2: Comparison of maximum pointwise errors.

Thus, for any $u \in C^2(I)$, its interpolation polynomial is given by

$$p(x) = \big(\mathcal{B}_-[u]\big)B_0(x)+\sum_{j=1}^{N-1} u''(x_j)B_j(x)+\big(\mathcal{B}_+[u]\big)B_N(x), \quad x \in I. \tag{35}$$

Armed with the new basis, we can impose mixed boundary conditions exactly, and the linear system resulting from the corresponding collocation scheme is well-conditioned. Here, we test the method on the second-order equation in (28) but with the mixed boundary conditions: $u(\pm 1) \pm u'(\pm 1) = u_\pm$. In Table 2, we list the condition numbers of the usual collocation method (LCOL, where the boundary conditions are treated by the τ-method), and the Birkhoff collocation method (BCOL, as in (31)).

Table 2: Comparison of condition numbers.

N	$r \equiv 0$ and $s \equiv 1$		$r \equiv s \equiv -1$	
	BCOL	LCOL	BCOL	LCOL
32	2.45	6.66e+04	2.61	7.87e+04
64	2.45	1.41e+06	2.63	1.68e+06
128	2.45	3.09e+07	2.64	3.70e+07
256	2.45	6.88e+08	2.64	8.26e+08
512	2.44	1.54e+10	2.65	1.86e+10
1024	2.44	3.48e+11	2.65	4.19e+11

REFERENCES

[1] F. A. Costabile and E. Longo. 2010. A Birkhoff interpolation problem and application. *Calcolo* 47, 1 (2010), 49–63. https://doi.org/10.1007/s10092-009-0014-9

[2] E. A. Coutsias, T. Hagstrom, J. S. Hesthaven, and D. Torres. 1996. Integration preconditioners for differential operators in spectral τ-methods. In *Proceedings of the Third International Conference on Spectral and High Order Methods, Houston, TX.* 21–38.

[3] E. A. Coutsias, T. Hagstrom, and D. Torres. 1996. An efficient spectral method for ordinary differential equations with rational function coefficients. *Math. Comp.* 65, 214 (1996), 611–635. https://doi.org/10.1090/S0025-5718-96-00704-1

[4] M. E. Elbarbary. 2006. Integration preconditioning matrix for ultraspherical pseudospectral operators. *SIAM J. Sci. Comput.* 28, 3 (2006), 1186–1201. https://doi.org/10.1137/050630982

[5] D. Funaro. 1988. Computing the inverse of the Chebyshev collocation derivative. *SIAM J. Sci. Stat. Comput.* 9, 6 (1988), 1050–1057.

[6] D. Funaro. 1992. *Polynomial Approximation of Differential Equations.* Lecture Notes in Physics New Series m: Monographs, Vol. 8. Springer-Verlag.

[7] D. Funaro and D. Gottlieb. 1988. A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations. *Math. Comp.* 51, 184 (1988), 599–613. https://doi.org/10.2307/2008765

[8] L. Greengard. 1991. Spectral integration and two-point boundary value problems. *SIAM J. Numer. Anal.* 28, 4 (1991), 1071–1080. https://doi.org/10.1137/0728057

[9] J. S. Hesthaven. 1998. Integration preconditioning of pseudospectral operators. I. Basic linear operators. *SIAM J. Numer. Anal.* 35, 4 (1998), 1571–1593. https://doi.org/10.1137/S0036142997319182

[10] M. D. Samson L.-L. Wang and X. Zhao. 2014. A well-conditioned collocation method using a pseudospectral integration matrix. *SIAM J. Sci. Comput.* 36, 3 (2014), A907–A929.

[11] J. Shen, T. Tang, and L. L. Wang. 2011. *Spectral Methods: Algorithms, Analysis and Applications.* Series in Computational Mathematics, Vol. 41. Springer-Verlag, Berlin, Heidelberg.

[12] D. M. Sloan. 2004. On the norms of inverses of pseudospectral differentiation matrices. *SIAM J. Numer. Anal.* 42, 1 (2004), 30–48.

[13] G. Szegö. 1975. *Orthogonal Polynomials* (fourth ed.). AMS Colloquium Publications.

[14] J. A. C. Weideman and S. C. Reddy. 2000. A MATLAB differentiation matrix suite. *ACM TOMS* 26, 4 (2000), 465–519.