TAIWANESE JOURNAL OF MATHEMATICS
Vol. 25, No. 1, pp. 125–154, February 2021
DOI: 10.11650/tjm/200802

From Stepwise Integrations and Low-rank Updates to a Pseudospectral Solution Operator Matrix for the Helmholtz Operator $\frac{d}{dx}a(x)\frac{d}{dx} + c(x)$

Yung-Ta Li, Ping-Hsuan Tsai and Chun-Hao Teng*

Abstract. In this study we propose a construction framework utilizing stepwise integrations and the Sherman-Morrison-Woodbury formula to seek pseudospectral integration preconditioning matrices for differential operators. We illustrate this framework through formulating an inverse matrix for the Helmholtz differential operator of the form $\frac{d}{dx}a(x)\frac{d}{dx} + c(x)$. Numerical experiments were conducted to examine the performance of the derived operator. The results show that the inverse matrix is an effective solution operator to numerically solve general second order differential equations.

1. Introduction

Spectral and pseudospectral methods [21, 22], due to their exponential convergence property of approximating smooth functions, have been applied to accurately solve differential equations for decades. However, to enjoy the advantage of the methods, great care must be exercised to effectively resolve those ill-conditioned systems of equations, resulting from the pseudospectral differentiation matrices. Generally speaking, the condition number of a pseudospectral differentiation matrix is scaled with $\mathcal{O}(N^{2m})$, where m is the order of the approximated differential operator, and N refers to the grid resolution of a mesh. As a result, one often encounters inaccurate computations due to this ill-conditioned issue, especially when high-order differential equations are solved on dense grids.

To overcome this obstacle, many methods have been proposed to construct operating matrices to effectively invert numerical differentiations. In [4–6, 23–25, 27–29], operating matrices are derived based on recombined basis functions satisfying boundary conditions. In [10, 20], operating matrices are formulated based on orthogonal polynomials with boundary conditions enforced through different techniques, for instance, boundary bordering [20] or auxiliary equations deduced from boundary conditions [10]. Based on an integration approach, operating or preconditioning matrices have been developed [15] for basic differentiation matrix operators with boundary conditions enforced weakly in a

Received April 16, 2020; Accepted August 12, 2020.
Communicated by Suh-Yuh Yang.
2010 *Mathematics Subject Classification.* 74S25, 65N12, 65F05.
Key words and phrases. spectral/pseudospectral methods, integration preconditioning, low-rank, Sherman-Morrison-Woodbury.
*Corresponding author.

penalty manner [8, 11, 12, 16, 17]. Arbitrary-grids-based integration preconditioning matrices have been constructed [13]. Beyond the commonly used Lagrange's interpolation formulation, pseudospectral integration matrix operators have been developed based on a suitable Birkhoff interpolation [30]. Indeed, applying these preconditioning operators significantly reduces the condition numbers of the targeted differentiation matrices, and numerical equations are solved effectively.

In the aforementioned studies, most of the developed operating matrices are primarily for pure differential operators, and the derivations rely upon recurrence relationships between orthogonal polynomials and their derivatives. A nice feature shared by these matrices is that they are banded matrices in spectral spaces. Thus, once discretized equations are transformed into spectral spaces, problems can be solved efficiently by exploring the sparsity of those banded matrices. However, for mixed differential operators, variable coefficients are sandwiched by differential operators. The resulting operating matrices constructed based on those recurrence relationships for mixed differential operators are generally full matrices, and the advantage of solving problems in spectral spaces is lost.

Numerical differentiation involving variable coefficients also encounters a similar situation, if variable coefficients and differential operators are integrated into a single matrix through using the orthogonal polynomial recurrence identities. However, as proposed in the seminal study [21, 22], a simple and widely used technique for mixed differentiation is through a differentiation-multiplication-differentiation procedure. This long-established technique has been brought to our attention, and a question is raised. Is it possible to seek inverse matrices for mixed differential operators through reversing the stepwise procedure? The study [19] gives an affirmative answer for the radial direction Laplace differential operators in polar coordinates.

The goal of this study is to realize an integration-division-integration framework to construct integration preconditioning matrices for differential operators. Our construction framework is based on two natural properties of nodal-based pseudospectral formulations: (1) a general pseudospectral differentiation operator, whether it is pure or mixed, is composed of multiple matrices with each one being either the first order differentiation matrix or a simple diagonal one, and (2) boundary operators are by themselves low-rank matrices. These two properties enlighten us to consider combining stepwise integrations and low-rank updates to construct inverse operators in a factored form for differentiation matrices. We demonstrate this approach through deriving an inverse matrix for the Helmholtz differential operator, $\frac{d}{dx}a(x)\frac{d}{dx} + c(x)$, based on the Legendre pseudospectral approximation method with boundary conditions enforced weakly through a penalty approach.

The construction procedures are summarized below. In [15], two boundary condition penalized (BCP) first order differentiation matrices were proposed and their inverses were

developed. We adopt these inverse operators as building blocks. Employing these inverse operators, we express the BCP pseudospectral Laplace operator, resulting from discretizing the differential operator $\frac{d}{dx}a(x)\frac{d}{dx}$, into a factored form, a diagonal matrix perturbed by a low-rank one, inserted in between the two basic BCP differentiation matrices. Since the inverse matrices of the BCP basic differentiation operators are available, seeking the inverse of the BCP pseudospectral Laplace differentiation operator boils down to finding the inverse of the sandwiched matrix. We complete this step by applying the Sherman-Morrison-Woodbury (SMW) formula. As a result, the inverse of the BCP pseudospectral Laplace operator is the matrix product of these three inverse matrices. Once the inverse of the BCP pseudospectral Laplace operator is obtained, we then recursively apply the SMW formula to construct the inverse matrix of the BCP pseudospectral Helmholtz operator.

The derived inverse pseudospectral Helmholtz matrix serves as a solution operator for advection-diffusion-reaction problems. We conduct various numerical experiments to examine the performance of the inverse operator. The convergence results show that the derived operator is effective, even for nonlinear problems.

The potential applications of the proposed framework advertise the significance of the framework itself. In general, numerical differential equations are systems of equations composed of pure/mixed differentiation matrices and low-rank numerical boundary operators. By exploring these properties as we will show in this study for the Helmholtz operator, one may adopt the present divide-and-conquer approach to construct efficient preconditioning matrices for other numerical differential equations. We will address relevant issues at the end of this study.

The rest of the paper is organized as follows. In the next section we demonstrate the proposed framework through seeking an inverse pseudospectral matrix for the targeted mixed differential operator. Section 3 is devoted to the validations of the derived operator. Conclusions are given in the last section.

2. Formulation

2.1. Basic concepts of the Legendre pseudospectral method and the SMW formula

Let $\mathsf{I} = [-1, 1]$ be the pseudospectral reference domain, $N \geq 0$ be an integer, and $P_N(x)$ defined on I be the N-th degree Legendre polynomial satisfying the differential equation

$$\big((1-x^2)P_N'(x)\big)' = -N(N+1)P_N(x), \quad x \in \mathsf{I},$$

where $'$ denotes the differentiation with respect to the function argument. The roots of the polynomial $(1-x^2)P_N'(x)$, ordered ascendantly as $-1 = x_0 < x_1 < \cdots < x_{N-1} < x_N = 1$, are known as the Gauss-Lobatto-Legendre (GLL) grid points. Denoted by $l_j(x)$, the

Lagrange basis polynomials based on the GLL points are given as

$$l_j(x) = -\frac{(1-x^2)P'_N(x)}{N(N+1)(x-x_j)P_N(x_j)}, \quad j = 0, 1, \ldots, N$$

satisfying the property $l_j(x_i) = \delta_{ij}$ with δ_{ij} being the Kronecker delta function.

Consider a function $u(x)$ defined on I. Employing the Lagrange basis polynomials, $l_j(x)$, we can construct an N-th degree polynomial to approximate $u(x)$ and its derivative $u'(x)$ as follows:

$$u(x) \approx \mathcal{I}_N[u(x)] = u_N(x) = \sum_{j=0}^{N} l_j(x)u(x_j), \quad u' \approx u'_N(x) = \sum_{j=0}^{N} l'_j(x)u(x_j),$$

where $\mathcal{I}_N$ is called the interpolation operator. Introducing the differentiation matrix $\boldsymbol{D}$ with entries $d_{ij} = l'_j(x_i)$ for $0 \le i, j \le N$, we can compute the values of $u'_N(x)$ at the GLL nodes by a matrix-vector multiplication approach:

$$\boldsymbol{u}'_N = \boldsymbol{D}\boldsymbol{u}_N, \quad \boldsymbol{u}_N = [u(x_0), \ldots, u(x_N)]^T, \quad \boldsymbol{u}'_N = \big[u'_N(x_0), \ldots, u'_N(x_N)\big]^T,$$

where T denotes the vector transpose.

Notice that the numerical differentiation becomes an exact differentiation, provided that $u(x)$ is a polynomial of degree at most N. As a consequence, the numerical differentiation of the n-th degree Legendre polynomial, $P_n(x)$, is exact for $n \le N$, i.e.,

$$\boldsymbol{p}'_n = \boldsymbol{D}\boldsymbol{p}_n, \quad \boldsymbol{p}_n = [P_n(x_0), \ldots, P_n(x_N)]^T, \quad \boldsymbol{p}'_n = \big[P'_n(x_0), \ldots, P'_n(x_N)\big]^T.$$

Among these derivative grid vectors, $\boldsymbol{p}'_0$, $\boldsymbol{p}'_1$ and $\boldsymbol{p}'_N$ are important to the present study. For the sake of convenience, we introduce the following vector expressions

$$\boldsymbol{e}_- = [1, 0, \ldots, 0]^T, \quad \boldsymbol{e}_+ = [0, \ldots, 0, 1]^T, \quad \boldsymbol{1} = [1, 1, \ldots, 1]^T, \quad \boldsymbol{0} = [0, 0, \ldots, 0]^T.$$

Then, the expressions of $\boldsymbol{p}'_0$, $\boldsymbol{p}'_1$ and $\boldsymbol{p}'_N$ are simply

$$\boldsymbol{p}'_0 = \boldsymbol{0}, \quad \boldsymbol{p}'_1 = \boldsymbol{p}_0 = \boldsymbol{1}, \quad \boldsymbol{p}'_N = \boldsymbol{D}\boldsymbol{p}_N = \frac{N(N+1)}{2}\big(\boldsymbol{e}_+ - (-1)^N\boldsymbol{e}_-\big),$$

where the first and the second expressions are due to the fact that $P_0(x) = 1$ and $P_1(x) = x$, and the last one results from the fact that $P'_N(x_i)$ vanishes at each interior GLL grid point and $P'_N(\pm 1) = (\pm 1)^{N+1}N(N+1)/2$.

Associated with the GLL nodes is the quadrature integration rule:

$$\int_{-1}^{1} u(x)\,dx = \sum_{i=0}^{N} u(x_i)\omega_i, \quad \omega_i = \frac{2}{N(N+1)(P_N(x_i))^2},$$

where ω_i for $i = 0, 1, \ldots, N$ are the quadrature weights, and the exactness of the integration rule is held true provided that $u(x)$ is a polynomial of degree at most $2N-1$. Employing the integration rule, we define the diagonal mass matrix $\boldsymbol{M}$ and the stiffness matrix $\boldsymbol{S}$ with their entries m_{ij} and s_{ij}, respectively, given by

$$m_{ij} = \sum_{k=0}^{N} l_i(x_k) l_j(x_k) \omega_k = \omega_i \delta_{ij}, \qquad 0 \le i, j \le N,$$
$$s_{ij} = \sum_{k=0}^{N} l_i(x_k) l_j'(x_k) \omega_k = \omega_i l_j'(x_i), \qquad 0 \le i, j \le N.$$

Three important properties concerning the matrices $\boldsymbol{M}$, $\boldsymbol{S}$, and $\boldsymbol{D}$ are shown in [3]. We directly quote the results here: (1) the mass matrix $\boldsymbol{M}$ is positive definite and diagonal, and thus invertible, (2) the stiffness matrix $\boldsymbol{S}$ is almost skew-symmetric,

$$\begin{aligned} \boldsymbol{S} + \boldsymbol{S}^T &= \boldsymbol{I}_+ - \boldsymbol{I}_-, \\ \boldsymbol{I}_+ &= \boldsymbol{e}_+ \boldsymbol{e}_+^T = \mathrm{diag}(0, \ldots, 0, 1), \\ \boldsymbol{I}_- &= \boldsymbol{e}_- \boldsymbol{e}_-^T = \mathrm{diag}(1, 0, \ldots, 0), \end{aligned} \tag{2.1}$$

and (3) the matrices, $\boldsymbol{M}$, $\boldsymbol{S}$, and $\boldsymbol{D}$, are related as

$$\boldsymbol{D} = \boldsymbol{M}^{-1} \boldsymbol{S}. \tag{2.2}$$

We now review concepts related to integration preconditioning. In [15] the invertible pseudospectral first order differentiation matrix has the form

$$\boldsymbol{D}_+ = \boldsymbol{D} - \eta \boldsymbol{I}_+, \quad \eta = \frac{N(N+1)}{4} = \frac{1}{2\omega_0} = \frac{1}{2\omega_N},$$

where η is the penalty parameter. In the present study we adopt the formulation of $\boldsymbol{D}_+$ and define the $\boldsymbol{D}_-$ and $\boldsymbol{D}_+$ operators, called the left-ended and right-ended BCP differentiation matrix, respectively, with their expressions given as

$$\boldsymbol{D}_\pm = \boldsymbol{D} \mp \eta \boldsymbol{I}_\pm, \quad \eta = \frac{N(N+1)}{2} = \frac{1}{\omega_0} = \frac{1}{\omega_N}.$$

With this slight change in the value of the penalty parameter, we follow the approach shown in [19] and find that the inverse matrices of $\boldsymbol{D}_+$ and $\boldsymbol{D}_-$ have the following factored form

$$\boldsymbol{D}_\pm^{-1} = \boldsymbol{P} \boldsymbol{J}_\pm^{-1} \boldsymbol{P}^T \boldsymbol{M}, \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_0 & \boldsymbol{p}_1 & \cdots & \boldsymbol{p}_N \end{bmatrix},$$

$$\boldsymbol{J}_{\pm}^{-1} = \frac{1}{2}\begin{bmatrix} \mp 1 & -1 & 0 & \cdots & \cdots & 0 & 0 \\ 1 & 0 & -1 & \ddots & & & 0 \\ 0 & 1 & 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 & -1 & 0 \\ \vdots & & & \ddots & 1 & 0 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & \mp 1 \end{bmatrix}.$$

Let $\boldsymbol{f}$ be the grid vector with components $f(x_i)$ for $i = 0, 1, \ldots, N$. Then $\boldsymbol{D}_{-}^{-1}\boldsymbol{f}$ is an approximation solution to the differential equation $u'(x) = f(x)$ subject to the left boundary condition $u(-1) = 0$. Likewise, $\boldsymbol{D}_{+}^{-1}\boldsymbol{f}$ returns an approximation solution to the same differential equation subject to the right boundary condition $u(+1) = 0$. Thus, we call $\boldsymbol{D}_{\pm}^{-1}$ the first order numerical solution operators.

Due to the simple matrix structure of $\boldsymbol{J}_{\pm}^{-1}$, it is worth introducing identities associated with $\boldsymbol{D}_{\pm}^{-1}$ for later use. Notice that

$$\boldsymbol{P}^T\boldsymbol{e}_{\pm} = [P_0(\pm 1), P_1(\pm 1), \ldots, P_N(\pm 1)]^T = \left[(\pm 1)^0, (\pm 1)^1, \ldots, (\pm 1)^N\right]^T.$$

Then we obtain the following identities

$$\begin{aligned} \eta\boldsymbol{D}_{\pm}^{-1}\boldsymbol{e}_{\pm} &= \mp\boldsymbol{p}_0, & \eta\boldsymbol{D}_{\mp}^{-1}\boldsymbol{e}_{\pm} &= (\pm 1)^{N-1}\boldsymbol{p}_N, \\ \boldsymbol{e}_{\pm}^T\boldsymbol{D}_{\pm}^{-1} &= -(\pm 1)^{N-1}\boldsymbol{p}_N^T\boldsymbol{M}, & \boldsymbol{e}_{\pm}^T\boldsymbol{D}_{\mp}^{-1} &= \pm\boldsymbol{p}_0^T\boldsymbol{M} \end{aligned} \tag{2.3}$$

by direct computations.

We have summarized the main concepts related to the Legendre pseudospectral method for the present study. For further details of the pseudospectral methods we refer the readers to [1, 2, 18, 26].

In this study, we also need the SMW formula to conduct low rank updates. For the purpose of self-containedness, we summarize some relevant concepts.

The SMW formula relates the inverse of a matrix perturbed by a low rank matrix with the inverse of the original matrix. Assume that a matrix $\boldsymbol{A}$ of order n-by-n is invertible. Now consider the sum of $\boldsymbol{A}$ and a low rank matrix of the form $\boldsymbol{UCV}$, where $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{C}$ are of conformable sizes. Then the inverse of $\boldsymbol{A}$ after being perturbed by the low rank matrix $\boldsymbol{UCV}$ can be computed using the following formula

$$(\boldsymbol{A} + \boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\big(\boldsymbol{C}^{-1} - \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\big)^{-1}\boldsymbol{V}\boldsymbol{A}^{-1} \tag{2.4}$$

provided that $(\boldsymbol{C}^{-1} - \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})$ is invertible. Thus, when $\boldsymbol{A}^{-1}$ is available, the formula provides a convenient approach to finding the inverse of the $\boldsymbol{A}$ matrix after being perturbed

by the low rank matrix, especially if the dimension of the matrix $\boldsymbol{C}$ is much smaller than the dimension of $\boldsymbol{A}$. A well known case is the following. If both $\boldsymbol{U} = \boldsymbol{u}$ and $\boldsymbol{V}^T = \boldsymbol{v}$ are column vectors, and $\boldsymbol{C}$ is a nonzero scalar, say 1 without losing generality, then (2.4) becomes the Sherman-Morrison formula:

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 - \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

For further details of the SMW formula, we refer the readers to [14] and the references therein.

2.2. BCP pseudospectral discretization of the $\frac{d}{dx}a(x)\frac{d}{dx} + c(x)$ operator

Let $u(x)$ be a function defined on the interval I satisfying the problem:

$$\begin{aligned} &\frac{d}{dx}\left(a(x)\frac{du(x)}{dx}\right) + c(x)u(x) = f(x), \qquad && x \in \mathsf{I},\ a(x) > 0, \\ &\alpha_\pm u(\pm 1) \pm \beta_\pm u'(\pm 1) = g_\pm, && \alpha_\pm, \beta_\pm \geq 0,\ \alpha_\pm^2 + \beta_\pm^2 \neq 0, \end{aligned} \tag{2.5}$$

where $a(x)$, $c(x)$ and $f(x)$ are given smooth functions, the parameters α_+, α_-, β_+ and β_- are non-negative reals, and g_+ and g_- are real numbers.

To solve the problem we introduce the GLL nodes on I and denote $a_j = a(x_j)$ and $f_j = f(x_j)$ as the pointwise function values at these points. To approximate $u(x)$, we collocate grid function values v_j for $j = 0, 1, \ldots, N$ and seek a numerical solution $v(x)$ of the form $v(x) = \sum_{j=0}^{N} l_j(x)v_j$ to satisfy the collocation equations

$$\begin{aligned} f_i = {} & \left(\mathcal{I}_N[av']\right)'\Big|_i + c(x_i)v(x_i) \\ & - \left(\tau_+\eta^2 a_N\delta_{Ni} - \chi_+\frac{a_N l_i'(+1)}{\omega_i}\right)(\alpha_+ v_N + \beta_+ v_N' - g_+) \\ & - \left(\tau_-\eta^2 a_0\delta_{0i} + \chi_-\frac{a_0 l_i'(-1)}{\omega_i}\right)(\alpha_- v_0 - \beta_- v_0' - g_-), \quad i = 0, 1, \ldots, N, \end{aligned} \tag{2.6}$$

where $\tau_\pm$ and $\chi_\pm$ are called penalty parameters related by

$$1 - \tau_-\beta_-\eta = \chi_-\alpha_-, \quad 1 - \tau_+\beta_+\eta = \chi_+\alpha_+. \tag{2.7}$$

Depending on the enforced boundary conditions, the values of the parameters τ_- and τ_+ are given as follows:

- non-Neumann condition enforced at $x = \pm 1$:

$$\tau_\pm > \frac{1}{\alpha_\pm + \beta_\pm\eta}, \quad \chi_\pm = \frac{1 - \tau_\pm\beta_\pm\eta}{\alpha_\pm}.$$

- Neumann condition enforced at $x = \pm 1$:

$$\tau_\pm = \frac{1}{\eta}, \quad \chi_\pm = 0.$$

Notice that for the Neumann boundary condition case, (2.7) is full-filled for any value of $\chi_\pm$. Thus, for simplicity we set $\chi_\pm = 0$.

To show the solvability of (2.6), we introduce the following vector and matrix notations

$$\begin{aligned} \boldsymbol{v} &= [v_0, v_1, \ldots, v_N]^T, & \boldsymbol{f} &= [f_0, f_0, \ldots, f_N]^T, \\ \boldsymbol{A} &= \mathrm{diag}(a_0, a_1, \ldots, a_N), & \boldsymbol{C} &= \mathrm{diag}(c_0, c_1, \ldots, c_N). \end{aligned}$$

In these matrix and vector notations, the scheme (2.6) takes the following form

$$\begin{aligned} (\boldsymbol{L} + \boldsymbol{C})\boldsymbol{v} = \boldsymbol{f} &- \boldsymbol{M}^{-1}\big(\tau_+ \boldsymbol{M}^{-1} - \chi_+ \boldsymbol{D}^T\big)\boldsymbol{A}\boldsymbol{e}_+ g_+ \\ &- \boldsymbol{M}^{-1}\big(\tau_- \boldsymbol{M}^{-1} + \chi_- \boldsymbol{D}^T\big)\boldsymbol{A}\boldsymbol{e}_- g_- \end{aligned} \tag{2.8}$$

with

$$\begin{aligned} \boldsymbol{L} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D} &- \boldsymbol{M}^{-1}\big(\tau_+ \boldsymbol{M}^{-1} - \chi_+ \boldsymbol{D}^T\big)\boldsymbol{A}\boldsymbol{e}_+\big(\alpha_+ \boldsymbol{e}_+^T + \beta_+ \boldsymbol{e}_+^T \boldsymbol{D}\big) \\ &- \boldsymbol{M}^{-1}\big(\tau_- \boldsymbol{M}^{-1} + \chi_- \boldsymbol{D}^T\big)\boldsymbol{A}\boldsymbol{e}_-\big(\alpha_- \boldsymbol{e}_-^T - \beta_- \boldsymbol{e}_-^T \boldsymbol{D}\big). \end{aligned} \tag{2.9}$$

The matrix $\boldsymbol{L}$ is called a BCP pseudospectral Laplace differentiation matrix, which is a discrete analogy of the Laplace differential operator $\frac{d}{dx}a(x)\frac{d}{dx}$ with the boundary conditions taken into account. The resultant matrix of $\boldsymbol{L} + \boldsymbol{C}$ is a BCP pseudospectral matrix representation of the Helmholtz operator $\frac{d}{dx}a(x)\frac{d}{dx} + c(x)$.

To seek the inverse matrix of targeted operator $\boldsymbol{L}+\boldsymbol{C}$ we first seek the inverse operator of $\boldsymbol{L}$. We start by showing that the operator $\boldsymbol{L}$ is invertible. Multiplying matrix $-\boldsymbol{M}$ to (2.9) from the left, and employing (2.1) and (2.2) as well as (2.7), we have

$$-\boldsymbol{M}\boldsymbol{L} = \boldsymbol{S}^T \boldsymbol{A}\boldsymbol{M}^{-1}\boldsymbol{S} - \boldsymbol{R}_+ + \boldsymbol{R}_-,$$

where

$$\boldsymbol{R}_\pm = \chi_\pm \alpha_\pm \big(\boldsymbol{I}_\pm \boldsymbol{A}\boldsymbol{D}_\pm + (\boldsymbol{I}_\pm \boldsymbol{A}\boldsymbol{D}_\pm)^T\big) \mp \tau_\pm \alpha_\pm \eta \boldsymbol{A}\boldsymbol{I}_\pm \mp \beta_\pm \chi_\pm \boldsymbol{D}^T(\boldsymbol{A}\boldsymbol{I}_\pm)\boldsymbol{D}. \tag{2.10}$$

Since the matrices, $\boldsymbol{S}^T\boldsymbol{A}\boldsymbol{M}^{-1}\boldsymbol{S}$, $\boldsymbol{R}_-$ and $\boldsymbol{R}_+$, are all symmetric, the matrix $-\boldsymbol{M}\boldsymbol{L}$ is also symmetric, indicating that $-\boldsymbol{M}\boldsymbol{L}$ is diagonalizable and all the eigenvalues of $-\boldsymbol{M}\boldsymbol{L}$ are real.

We now show that these eigenvalues are positive by considering the eigenvalue-eigenvector problem:

$$\lambda \boldsymbol{u} = -\boldsymbol{M}\boldsymbol{L}\boldsymbol{u} = \big(\boldsymbol{S}^T \boldsymbol{A}\boldsymbol{M}^{-1}\boldsymbol{S} - \boldsymbol{R}_+ + \boldsymbol{R}_-\big)\boldsymbol{u}, \quad \boldsymbol{u} = [u_0, u_1, \ldots, u_N]^T, \tag{2.11}$$

where λ is an eigenvalue of $-\boldsymbol{ML}$ and $\boldsymbol{u}$ is the associated eigenvector of unit length, that is, $\boldsymbol{u}^T\boldsymbol{u} = \sum_{i=0}^{N} u_i^2 = 1$.

Then multiplying $\boldsymbol{u}^T$ to (2.11) from the left and invoking the relationship $\boldsymbol{MD} = \boldsymbol{S}$ (see (2.2)) and the expressions of $\boldsymbol{R}_+$ and $\boldsymbol{R}_-$ in (2.10), we have

$$\lambda = \sum_{i=1}^{N-1} (u_i')^2 a_i \omega_i + a_N \boldsymbol{r}_+^T \boldsymbol{W}_+ \boldsymbol{r}_+ + a_0 \boldsymbol{r}_-^T \boldsymbol{W}_- \boldsymbol{r}_- \geq 0, \tag{2.12}$$

where $u_i' = \sum_{j=0}^{N} l_j'(x_i) u_j$, $\boldsymbol{r}_+ = \left[u_N, -u_N'\right]^T$, and $\boldsymbol{r}_- = \left[u_0, u_0'\right]^T$, and the matrices $\boldsymbol{W}_+$ and $\boldsymbol{W}_-$ are

$$\boldsymbol{W}_\pm = \begin{bmatrix} \tau_\pm \alpha_\pm \eta & \frac{1}{2}(1 - \beta_\pm \tau_\pm \eta + \chi_\pm \alpha_\pm) \\ \frac{1}{2}(1 - \beta_\pm \tau_\pm \eta + \chi_\pm \alpha_\pm) & \eta^{-1} - \beta_\pm \chi_\pm \end{bmatrix}.$$

Notice that $\boldsymbol{W}_-$ or $\boldsymbol{W}_+$ is semi-positive definite if the enforced boundary condition at the associated end point is of the Neumann type, and is positive definite otherwise. However, if both boundary conditions are of the Neumann type, then the Poisson problem does not have a unique solution. Hence, we exclude the pure Neumann case in this part of the analysis and conclude that either $\boldsymbol{W}_+$ or $\boldsymbol{W}_-$ is positive definite. We will discuss the pure Neumann case for the Helmholtz problem later. Thus, shown in (2.12) the non-negativity of λ follows from the facts that (1) the term $\sum_{i=1}^{N-1} a_i (u_i')^2 \omega_i$ is nonnegative, and (2) both $\boldsymbol{W}_+$ and $\boldsymbol{W}_-$ are semi-positive definite.

We now claim that λ is strictly positive. For an eigenvector $\boldsymbol{u}$, the values of u_i' for $1 \leq i \leq N-1$ are either (1) non-zero for some i, or (2) all zeros. For the former case, from (2.12) we have

$$\lambda \geq \sum_{i=1}^{N-1} a_i (u_i')^2 \omega_i > 0.$$

For the latter one ($u_i' = 0$ for $i = 1, \ldots, N-1$), we identify that u' is a polynomial of degree at most $N-1$, and vanishes at all the interior nodes, implying that $u(x) = \sum_{j=0}^{N} l_j(x) u_j$, the continuous representation of the eigenvector $\boldsymbol{u}$, has the form

$$u(x) = c_1 P_N(x) + c_2, \quad c_1^2 + c_2^2 \neq 0,$$

where c_1 and c_2 are constants. Consequently, $\boldsymbol{r}_+$ and $\boldsymbol{r}_-$ are non-zero vectors. Then from (2.12) we have

$$\lambda = a_0 \boldsymbol{r}_-^T \boldsymbol{W}_- \boldsymbol{r}_- + a_N \boldsymbol{r}_+^T \boldsymbol{W}_+ \boldsymbol{r}_+ > 0$$

since either $\boldsymbol{W}_-$ or $\boldsymbol{W}_+$ is positive definite.

We have shown that the matrix $-\boldsymbol{ML}$ is symmetric positive definite, and thus, invertible. As a result, the existence of $\boldsymbol{L}^{-1}$ is established through the relationship that $\boldsymbol{L}^{-1} = (\boldsymbol{ML})^{-1}\boldsymbol{M}$.

2.3. Inverse matrix of the BCP pseudospectral Laplace operator

We now focus on explicitly formulating $\boldsymbol{L}^{-1}$ based on matrix factorizations. Applying the operators $\boldsymbol{D}_{\pm}$, we rewrite $\boldsymbol{L}$ defined in (2.9) as

$$\boldsymbol{L} = \boldsymbol{D}_{-}\boldsymbol{A}\boldsymbol{D}_{+} + \boldsymbol{Q},$$

where

$$\begin{aligned}\boldsymbol{Q} = {} & (1-\chi_{+}(\alpha_{+}+\beta_{+}\eta))\eta a_N \boldsymbol{D}_{-}\boldsymbol{I}_{+} + (\chi_{+}-\tau_{+})(\alpha_{+}+\beta_{+}\eta)\eta^2 a_N \boldsymbol{I}_{+} \\ & + (\chi_{+}-\tau_{+})\beta_{+}\eta^2 a_N \boldsymbol{I}_{+}\boldsymbol{D}_{+} - \chi_{+}\beta_{+}\eta a_N \boldsymbol{D}_{-}\boldsymbol{I}_{+}\boldsymbol{D}_{+} - \tau_{-}\alpha_{-}\eta^2 a_0 \boldsymbol{I}_{-} \\ & - (1-\tau_{-}\beta_{-}\eta)\eta a_0 \boldsymbol{I}_{-}\boldsymbol{D}_{+} + \chi_{-}\alpha_{-}\eta a_0 \boldsymbol{D}_{-}\boldsymbol{I}_{-} - \chi_{-}\beta_{-}\eta a_0 \boldsymbol{D}_{-}\boldsymbol{I}_{-}\boldsymbol{D}_{+}.\end{aligned} \tag{2.13}$$

Since $\boldsymbol{D}_{-}$ and $\boldsymbol{D}_{+}$ are invertible, we express $\boldsymbol{L}$ in a factored form as

$$\boldsymbol{L} = \boldsymbol{D}_{-}\big(\boldsymbol{A} + \boldsymbol{D}_{-}^{-1}\boldsymbol{Q}\boldsymbol{D}_{+}^{-1}\big)\boldsymbol{D}_{+},$$

which is suggested by the equation itself to formulate $\boldsymbol{L}^{-1}$ as

$$\boldsymbol{L}^{-1} = \boldsymbol{D}_{+}^{-1}\big(\boldsymbol{A} + \boldsymbol{D}_{-}^{-1}\boldsymbol{Q}\boldsymbol{D}_{+}^{-1}\big)^{-1}\boldsymbol{D}_{-}^{-1}.$$

We now proceed to find the desired operator $\big(\boldsymbol{A} + \boldsymbol{D}_{-}^{-1}\boldsymbol{Q}\boldsymbol{D}_{+}^{-1}\big)^{-1}$. To illustrate the construction of $\boldsymbol{L}^{-1}$, we consider the problem subject to Dirichlet boundary conditions $(\alpha_{\pm}, \beta_{\pm}) = (1, 0)$, for the sake of clarity.

For the Dirichlet problem, the corresponding penalty parameters are $\chi_{\pm} = 1$ and we set $\tau_{-} = \tau_{+} = \tau > 1$. Then the $\boldsymbol{Q}$ matrix is simplified as

$$\boldsymbol{Q} = (1-\tau)\eta^2 a_N \boldsymbol{I}_{+} - \tau\eta^2 a_0 \boldsymbol{I}_{-} - \eta a_0 \boldsymbol{I}_{-}\boldsymbol{D}_{+} + \eta a_0 \boldsymbol{D}_{-}\boldsymbol{I}_{-}.$$

We factor $\boldsymbol{Q}$ as follows:

$$\boldsymbol{Q} = \widehat{\boldsymbol{\Phi}}\widehat{\boldsymbol{\Psi}}^{T}, \quad \widehat{\boldsymbol{\Phi}} = \begin{bmatrix}\boldsymbol{D}_{-}\boldsymbol{e}_{-} & \eta\boldsymbol{e}_{-} & \eta\boldsymbol{e}_{+}\end{bmatrix}, \quad \widehat{\boldsymbol{\Psi}}^{T} = \begin{bmatrix} a_0\eta\boldsymbol{e}_{-}^{T} \\ -a_0\tau\eta\boldsymbol{e}_{-}^{T} - a_0\boldsymbol{e}_{-}^{T}\boldsymbol{D}_{+} \\ a_N(1-\tau)\eta\boldsymbol{e}_{+}^{T}\end{bmatrix}.$$

Notice that $\boldsymbol{Q}$ is a low-rank matrix and $\boldsymbol{A}$ is a diagonal one. Then, invoking the SMW formula we have

$$\big(\boldsymbol{A} + \boldsymbol{D}_{-}^{-1}\boldsymbol{Q}\boldsymbol{D}_{+}^{-1}\big)^{-1} = \big(\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{\Phi}\boldsymbol{K}^{-1}\boldsymbol{\Psi}^{T}\big)\boldsymbol{A}^{-1}, \tag{2.14}$$

where

$$\boldsymbol{\Phi} = \boldsymbol{D}_{-}^{-1}\widehat{\boldsymbol{\Phi}}, \quad \boldsymbol{\Psi}^{T} = \widehat{\boldsymbol{\Psi}}^{T}\boldsymbol{D}_{+}^{-1}, \quad \boldsymbol{K} = \boldsymbol{I}_3 + \boldsymbol{\Psi}^{T}\boldsymbol{A}^{-1}\boldsymbol{\Phi} \tag{2.15}$$

with $\boldsymbol{I}_3$ being the identity matrix of order three. As shown in (2.14), inverting the matrix $\boldsymbol{A}+\boldsymbol{D}_-^{-1}\boldsymbol{Q}\boldsymbol{D}_+^{-1}$ hinges upon finding $\boldsymbol{\Phi}\boldsymbol{K}^{-1}\boldsymbol{\Psi}^T$, which is treated next.

Employing the $\boldsymbol{D}_-^{-1}$ and $\boldsymbol{D}_+^{-1}$ operators and invoking the expressions shown in (2.3), we compute $\boldsymbol{D}_-^{-1}\widehat{\boldsymbol{\Phi}}$ and $\widehat{\boldsymbol{\Psi}}^T\boldsymbol{D}_+^{-1}$ and have

$$\boldsymbol{\Phi}=\boldsymbol{D}_-^{-1}\widehat{\boldsymbol{\Phi}}=\begin{bmatrix}\boldsymbol{e}_- & \boldsymbol{p}_0 & \boldsymbol{p}_N\end{bmatrix}, \tag{2.16}$$

and

$$\boldsymbol{\Psi}^T=\widehat{\boldsymbol{\Psi}}^T\boldsymbol{D}_+^{-1}=\begin{bmatrix}-\eta a_0\boldsymbol{p}_0^T\\ \tau_-\eta a_0\boldsymbol{p}_0^T-\eta a_0\boldsymbol{e}_-^T\\ -(1-\tau)\eta a_N\boldsymbol{p}_N^T\end{bmatrix}\boldsymbol{M}=\boldsymbol{\Lambda}\boldsymbol{\Phi}^T\boldsymbol{M},$$

where $\boldsymbol{\Lambda}$ and its inverse are given as

$$\boldsymbol{\Lambda}=\eta\begin{bmatrix}0 & -a_0 & 0\\ -a_0 & \tau a_0 & 0\\ 0 & 0 & (\tau-1)a_N\end{bmatrix},\quad \boldsymbol{\Lambda}^{-1}=\frac{1}{\eta}\begin{bmatrix}-\frac{\tau}{a_0} & -a_0^{-1} & 0\\ -a_0^{-1} & 0 & 0\\ 0 & 0 & \frac{1}{(\tau-1)a_N}\end{bmatrix}.$$

Employing the expression of $\boldsymbol{\Psi}^T$ we have

$$\boldsymbol{\Phi}\boldsymbol{K}^{-1}\boldsymbol{\Psi}^T=\boldsymbol{\Phi}\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T\boldsymbol{M}.$$

Thus, to find the matrix $\left(\boldsymbol{A}+\boldsymbol{D}_-^{-1}\boldsymbol{Q}\boldsymbol{D}_+^{-1}\right)^{-1}$ we are led to find the matrix $\boldsymbol{K}^{-1}\boldsymbol{\Lambda}$. From the expression of $\boldsymbol{K}$ in (2.15) we have the inverse of $\boldsymbol{K}^{-1}\boldsymbol{\Lambda}$ as

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{K}=\boldsymbol{\Lambda}^{-1}+\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi}.$$

Employing (2.16) we compute $\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi}$ and obtain

$$\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi}=\frac{1}{\eta}\begin{bmatrix}a_0^{-1} & a_0^{-1} & (-1)^N a_0^{-1}\\ a_0^{-1} & \kappa_2 & \kappa_1\\ (-1)^N a_0^{-1} & \kappa_1 & \kappa_0\end{bmatrix},\quad \kappa_\nu=\sum_{i=0}^{N}\frac{1}{a_i(P_N(\xi))^\nu}.$$

Then the matrix $\boldsymbol{\Lambda}^{-1}\boldsymbol{K}$ is given as

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{K}=\frac{1}{\eta}\begin{bmatrix}(1-\tau)\frac{1}{a_0} & 0 & \frac{(-1)^N}{a_0}\\ 0 & \kappa_2 & \kappa_1\\ \frac{(-1)^N}{a_0} & \kappa_1 & \kappa_0-\frac{1}{(1-\tau)a_N}\end{bmatrix}, \tag{2.17}$$

and its inverse is found as

$$\boldsymbol{K}^{-1}\boldsymbol{\Lambda}=\frac{\eta}{T}\begin{bmatrix}t_{11} & t_{12} & t_{13}\\ t_{12} & t_{22} & t_{23}\\ t_{13} & t_{23} & t_{33}\end{bmatrix},$$

where T is given as

$$T = \frac{(1-\tau)^2}{a_0 a_N}(\kappa_1^2 - \kappa_2\kappa_0) + \frac{(1-\tau)\kappa_2}{a_0 a_N}\left(\frac{1}{a_0} + \frac{1}{a_N}\right),$$

and t_{ij} for $i = 1, 2, 3$ and $i \le j \le 3$ are given as

$$\begin{aligned}
t_{11} &= \frac{(1-\tau)}{a_N}\big(\kappa_1^2 - \kappa_0\kappa_2\big) + \frac{\kappa_2}{a_N^2}, & t_{12} &= 0,\\
t_{13} &= \frac{(-1)^N(1-\tau)\kappa_2}{a_0 a_N}, & t_{22} &= \frac{(1-\tau)}{a_0 a_N}\left(\frac{a_0 + a_N}{a_0 a_N} - (1-\tau)\kappa_0\right),\\
t_{23} &= \frac{(1-\tau)^2\kappa_1}{a_0 a_N}, & t_{33} &= -\frac{(1-\tau)^2\kappa_2}{a_0 a_N}.
\end{aligned}$$

The expression of the operator $\big(\boldsymbol{A} + \boldsymbol{D}_-^{-1}\boldsymbol{Q}\boldsymbol{D}_+^{-1}\big)^{-1}$ is given as

$$\big(\boldsymbol{A} + \boldsymbol{D}_-^{-1}\boldsymbol{Q}\boldsymbol{D}_+^{-1}\big)^{-1} = \big(\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{\Phi}\big(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\big)\boldsymbol{\Phi}^T\boldsymbol{M}\big)\boldsymbol{A}^{-1},$$

and thus, $\boldsymbol{L}^{-1}$ is given as

$$\boldsymbol{L}^{-1} = \boldsymbol{D}_+^{-1}\big(\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{\Phi}\big(\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\big)\boldsymbol{\Phi}^T\boldsymbol{M}\big)\boldsymbol{A}^{-1}\boldsymbol{D}_-^{-1}.$$

We have shown the construction of $\boldsymbol{L}^{-1}$ for problems subject to the Dirichlet boundary conditions. For problems subject to general boundary constraints, the corresponding vectors $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}^T$, and the matrix $\boldsymbol{K}^{-1}\boldsymbol{\Lambda}$ to formulate $\boldsymbol{L}^{-1}$ are given in Appendix A.

Before proceeding further we address an issue related to examining the invertibility of the $\boldsymbol{L}$ operator in the present inverse matrix construction framework. The $\boldsymbol{L}$ operator is specifically constructed such that $-\boldsymbol{L}$ becomes symmetric positive definite under the mass matrix weighted discrete 2-norm, and thus invertible. In the construction of $\boldsymbol{L}^{-1}$ one can also investigate the invertibility of the problem through examining the invertibility of the 3-by-3 matrix $\boldsymbol{\Lambda}^{-1}\boldsymbol{K}$ given in (2.17), which does not require knowing properties of the operator $\boldsymbol{L}$. Thus, for other pseudospectral differential operators which are not confined to the regular discrete 2-norm as the present $\boldsymbol{L}$ operator is, one may still establish the invertibility of those operators by examining the corresponding compatible matrices.

2.4. Inverse matrix of the BCP pseudospectral Helmholtz operator

We now use the derived $\boldsymbol{L}^{-1}$ matrix to construct the $(\boldsymbol{L}+\boldsymbol{C})^{-1}$ operator by applying the SMW formula and by exploiting the diagonal matrix structure of $\boldsymbol{C}$.

Let $\boldsymbol{e}_i$, for $i = 0, 1, \ldots, N$, be the i-th column vectors of the identity matrix of order $N+1$. To find the inverse of $\boldsymbol{L}+\boldsymbol{C}$ we express $\boldsymbol{L}+\boldsymbol{C}$ as

$$\boldsymbol{H} = \boldsymbol{L} + \boldsymbol{C} = \boldsymbol{L}_k + \sum_{i=k}^{N} c_i \boldsymbol{e}_i \boldsymbol{e}_i^T, \quad \boldsymbol{L}_k = \boldsymbol{L} + \sum_{i=0}^{k-1} c_i \boldsymbol{e}_i \boldsymbol{e}_i^T.$$

Then $\boldsymbol{L}_0 = \boldsymbol{L}$ and $\boldsymbol{L}_N = \boldsymbol{H}$, and furthermore, $\boldsymbol{L}_{k+1}$ and $\boldsymbol{L}_k$ are related as

$$\boldsymbol{L}_{k+1} = \boldsymbol{L}_k + c_k \boldsymbol{e}_k \boldsymbol{e}_k^T.$$

Assume that $\boldsymbol{L}_k^{-1}$ exists. Then $\boldsymbol{L}_{k+1}^{-1}$ is given as

$$\boldsymbol{L}_{k+1}^{-1} = \boldsymbol{L}_k^{-1} - \big(\boldsymbol{L}_k^{-1}\boldsymbol{e}_k\big)\big(c_k^{-1} + \boldsymbol{e}_k^T \boldsymbol{L}_k^{-1}\boldsymbol{e}_k\big)^{-1}\big(\boldsymbol{e}_k^T \boldsymbol{L}_k^{-1}\big) \tag{2.18}$$

provided that the value $c_k^{-1} + \boldsymbol{e}_k^T \boldsymbol{L}_k^{-1}\boldsymbol{e}_k$ is nonzero. Since $\boldsymbol{L}_0^{-1} = \boldsymbol{L}^{-1}$ is available we can recursively apply (2.18) $N+1$ times to construct $\boldsymbol{H}^{-1}$.

For $c(x) < 0$, $\boldsymbol{H}^{-1}$ definitely exists, because $\boldsymbol{L} + \boldsymbol{C}$ is a negative definite operator weighted by the mass matrix $\boldsymbol{M}$. However, for a general $c(x)$ we are unable to analytically draw out any result regarding the inverse of $(\boldsymbol{L} + \boldsymbol{C})$. In this situation, we may try constructing $\boldsymbol{H}^{-1}$ by the updating procedure (2.18). If the construction is successfully executed, then one may try to use the resultant $\boldsymbol{H}^{-1}$ operator to compute numerical solutions. If the updating procedure fails to construct the $\boldsymbol{H}^{-1}$ operator, for example, if at any k step the value $c_k^{-1} + \boldsymbol{e}_k^T \boldsymbol{L}_k^{-1}\boldsymbol{e}_k$ vanishes, it does not mean that the $\boldsymbol{H}$ operator is not invertible. For this situation, one would have to use other approaches to seek the $\boldsymbol{H}^{-1}$ operator, if it exists.

We now discuss the construction of the Helmholtz operator when the boundary conditions at the end points are both of the Neumann type. When we previously discussed the construction of the $\boldsymbol{L}$ matrix we excluded the pure Neumann case, because the solution to the corresponding Poisson problem is not unique. However, the solution to the Helmholtz equation subject to the Neumann boundary condition at both end points is unique. In what follows we derive the corresponding $\boldsymbol{H}^{-1}$ operator.

The Helmholtz operator with pure Neumann boundary conditions can be expressed as

$$\boldsymbol{H} = \boldsymbol{L} + \boldsymbol{C} = \boldsymbol{L} - \varepsilon \boldsymbol{I}_+ + \boldsymbol{C} + \varepsilon \boldsymbol{I}_+,$$

where, deduced from (2.9) with $(\alpha_\pm, \beta) = (0, 1)$, the matrix $\boldsymbol{L}$ given as

$$\boldsymbol{L} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D} - \eta^{-1}\boldsymbol{M}^{-2}\boldsymbol{A}\boldsymbol{e}_+\boldsymbol{e}_+^T\boldsymbol{D} + \eta^{-1}\boldsymbol{M}^{-2}\boldsymbol{A}\boldsymbol{e}_-\boldsymbol{e}_-^T\boldsymbol{D},$$

is the pure Neumann type BCP-pseudosectral Laplace operator, and ε is an introduced parameter with its value to be determined such that the resultant matrix $\boldsymbol{L} - \varepsilon\boldsymbol{I}_+$ is invertible.

To seek the inverse of the matrix operator $\boldsymbol{L} - \varepsilon\boldsymbol{I}_+$ we rewrite the matrix as

$$\boldsymbol{L} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D} - \eta^{-1}\boldsymbol{M}^{-2}\boldsymbol{A}\boldsymbol{e}_+\boldsymbol{e}_+^T\boldsymbol{D} + \eta^{-1}\boldsymbol{M}^{-2}\boldsymbol{A}\boldsymbol{e}_-\boldsymbol{e}_-^T\boldsymbol{D} - \varepsilon\boldsymbol{I}_+ = \boldsymbol{D}_-\boldsymbol{A}\boldsymbol{D}_+ + \boldsymbol{Q}$$

with $\boldsymbol{Q}$ factored as

$$\boldsymbol{Q} = \widehat{\boldsymbol{\Phi}}\widehat{\boldsymbol{\Psi}}^T, \quad \widehat{\boldsymbol{\Phi}} = \Big[\eta\boldsymbol{e}_+ \quad \boldsymbol{D}_-\boldsymbol{e}_+\Big], \quad \widehat{\boldsymbol{\Psi}}^T = \begin{bmatrix} -\big(\frac{\varepsilon}{\eta} + a_N\eta\big)\boldsymbol{e}_+^T - a_N\boldsymbol{e}_+^T\boldsymbol{D}_+ \\ \eta a_N \boldsymbol{e}_+ \end{bmatrix}.$$

Computing $\boldsymbol{D}_-^{-1}\widehat{\boldsymbol{\Phi}}$ and $\widehat{\boldsymbol{\Psi}}^T\boldsymbol{D}_+^{-1}$, we obtain

$$\boldsymbol{\Phi} = \boldsymbol{D}_-^{-1}\widehat{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{p}_N & \boldsymbol{e}_+ \end{bmatrix}, \quad \boldsymbol{\Psi}^T = \widehat{\boldsymbol{\Psi}}^T\boldsymbol{D}_+^{-1} = \begin{bmatrix} \left(a_N\eta + \frac{\varepsilon}{\eta}\right)\boldsymbol{p}_N^T - a_N\eta\boldsymbol{e}_+^T \\ -\eta a_N \boldsymbol{P}_N^T \end{bmatrix} \boldsymbol{M},$$

and $\boldsymbol{\Psi}^T$ and $\boldsymbol{\Phi}$ are related by

$$\boldsymbol{\Psi}^T = \boldsymbol{\Lambda}\boldsymbol{\Phi}^T\boldsymbol{M}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \eta a_N + \frac{\varepsilon}{\eta} & -\eta a_N \\ -\eta a_N & 0 \end{bmatrix}.$$

We then compute $\boldsymbol{\Lambda}^{-1}$ and $\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi}$ and have

$$\boldsymbol{\Lambda}^{-1} = \frac{-1}{\eta a_N}\begin{bmatrix} 0 & 1 \\ 1 & 1 + \frac{\varepsilon}{\eta^2 a_N} \end{bmatrix}, \quad \boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi} = \frac{1}{\eta a_N}\begin{bmatrix} \kappa_0 a_N & 1 \\ 1 & 1 \end{bmatrix}.$$

Then we have

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{K} = \boldsymbol{\Lambda}^{-1} + \boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi} = \frac{1}{\eta a_N}\begin{bmatrix} \kappa_0 a_N & 0 \\ 0 & -\frac{\varepsilon}{\eta^2 a_N} \end{bmatrix},$$

leading to

$$\boldsymbol{K}^{-1}\boldsymbol{\Lambda} = \operatorname{diag}\left(\frac{\eta}{\kappa_0}, -\frac{\eta^3 a_N^2}{\varepsilon}\right).$$

The inverse of $\boldsymbol{L} - \varepsilon\boldsymbol{I}_+$ is given as

$$(\boldsymbol{L} - \varepsilon\boldsymbol{I}_+)^{-1} = \boldsymbol{D}_+^{-1}\big(\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{\Phi}\boldsymbol{K}^{-1}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T\boldsymbol{M}\big)\boldsymbol{A}^{-1}\boldsymbol{D}_-^{-1}.$$

We have the formula to construct $(\boldsymbol{L} - \varepsilon\boldsymbol{I}_+)^{-1}$, and we can then apply (2.18) to derive the matrix $\boldsymbol{H}^{-1}$.

2.5. Inhomogeneous boundary conditions

The $\boldsymbol{H}^{-1}$ operator can now be used to compute numerical solutions to the second order problems subject to homogeneous boundary conditions, that is, $g_- = g_+ = 0$, by performing $\boldsymbol{v} = \boldsymbol{v}_h = \boldsymbol{H}^{-1}\boldsymbol{f}$. If inhomogeneous boundary conditions are imposed, then the numerical solution $\boldsymbol{v}$ to the second order problem is the sum of the homogeneous solution $\boldsymbol{v}_h$ and the particular solution $\boldsymbol{v}_p$ satisfying the problem

$$\begin{aligned} \boldsymbol{H}\boldsymbol{v}_p = &-a_0\big(\tau_-\eta^2\boldsymbol{e}_- + \chi_-\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_-\big)g_- \\ &- a_N\big(\tau_+\eta^2\boldsymbol{e}_+ - \chi_+\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_+\big)g_+. \end{aligned} \tag{2.19}$$

To seek the particular solution $\boldsymbol{v}_p$ we assume that $\boldsymbol{v}_p$ is of the form

$$\boldsymbol{v}_p = \boldsymbol{H}^{-1}(-z_1\boldsymbol{D}\boldsymbol{a} - z_1\boldsymbol{C}\boldsymbol{p}_1 - z_0\boldsymbol{C}\boldsymbol{p}_0) + z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0,$$

where z_0 and z_1 are constants to be determined. Performing $\boldsymbol{H}\boldsymbol{v}_p$ we obtain

$$\begin{aligned}\boldsymbol{H}\boldsymbol{v}_p &= \boldsymbol{H}\boldsymbol{H}^{-1}(-z_1\boldsymbol{D}\boldsymbol{a} - z_1\boldsymbol{C}\boldsymbol{p}_1 - z_0\boldsymbol{C}\boldsymbol{p}_0) + \boldsymbol{L}(z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0) + \boldsymbol{C}(z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0)\\ &= -z_1\boldsymbol{D}\boldsymbol{a} - z_1\boldsymbol{C}\boldsymbol{p}_1 - z_0\boldsymbol{C}\boldsymbol{p}_0 + \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}(z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0) + z_1\boldsymbol{C}\boldsymbol{p}_1 + z_0\boldsymbol{C}\boldsymbol{p}_0\\ &\quad - a_N\left(\frac{\tau_+}{\omega_N^2}\boldsymbol{e}_+ - \chi_+\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_+\right)\big(\alpha_+\boldsymbol{e}_+^T + \beta_+\boldsymbol{e}_+^T\boldsymbol{D}\big)(z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0)\\ &\quad - a_0\left(\frac{\tau_-}{\omega_0^2}\boldsymbol{e}_- + \chi_-\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_-\right)\big(\alpha_-\boldsymbol{e}_-^T - \beta_-\boldsymbol{e}_-^T\boldsymbol{D}\big)(z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0).\end{aligned}$$

Applying the following relationships

$$\begin{gathered}\boldsymbol{D}\boldsymbol{p}_1 = \boldsymbol{p}_0, \quad \boldsymbol{D}\boldsymbol{p}_0 = \boldsymbol{0}, \quad \boldsymbol{A}\boldsymbol{D}\boldsymbol{p}_1 = \boldsymbol{A}\boldsymbol{p}_0 = \boldsymbol{a},\\ \big(\alpha_\pm\boldsymbol{e}_\pm^T \pm \beta_\pm\boldsymbol{e}_\pm^T\boldsymbol{D}\big)(z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0) = \alpha_\pm(\pm z_1 + z_0) \pm \beta_\pm z_1,\end{gathered}$$

we arrive at

$$\begin{aligned}\boldsymbol{H}\boldsymbol{v}_p &= -a_N\big(\tau_+\eta^2\boldsymbol{e}_+ - \chi_+\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_+\big)(\alpha_+(z_1 + z_0) + \beta_+ z_1)\\ &\quad - a_0\big(\tau_-\eta^2\boldsymbol{e}_- + \chi_-\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_-\big)(\alpha_-(-z_1 + z_0) - \beta_- z_1).\end{aligned}$$

By matching the coefficients on the right-hand side of (2.19), we have

$$(\alpha_+ + \beta_+)z_1 + \alpha_+ z_0 = g_+, \quad -(\alpha_- + \beta_-)z_1 + \alpha_- z_0 = g_-.$$

Solving the equations we obtain

$$z_0 = \frac{(\alpha_- + \beta_-)g_+ + (\alpha_+ + \beta_+)g_-}{(\alpha_- + \beta_-)\alpha_+ + (\alpha_+ + \beta_+)\alpha_-}, \quad z_1 = \frac{\alpha_- g_+ - \alpha_+ g_-}{(\alpha_- + \beta_-)\alpha_+ + (\alpha_+ + \beta_+)\alpha_-},$$

and $\boldsymbol{v}_p$ is determined.

If the inhomogeneous boundary conditions at the two end points are both of the Neumann type, then we have the inhomogeneous problem as follows:

$$\boldsymbol{H}\boldsymbol{v}_p = -a_0(\eta\boldsymbol{e}_-)g_- - a_N(\eta\boldsymbol{e}_+)g_+,$$

implying that

$$\boldsymbol{v}_p = -\eta a_0 g_-\boldsymbol{H}^{-1}\boldsymbol{e}_- - \eta a_N g_+\boldsymbol{H}^{-1}\boldsymbol{e}_+.$$

Notice that the vectors $\boldsymbol{H}^{-1}\boldsymbol{e}_\pm$ are available once $\boldsymbol{H}^{-1}$ is constructed, since they are the first and the last column vectors of $\boldsymbol{H}^{-1}$.

Finally, to numerically solve (2.8), we propose the computing steps after initializing the required variables.

Step 1:

$$\boldsymbol{f}^* = \boldsymbol{f} - \begin{cases} z_1(\boldsymbol{D}\boldsymbol{a} + \boldsymbol{C}\boldsymbol{p}_1) + z_0\boldsymbol{C}\boldsymbol{p}_0 & \text{general case},\\ \boldsymbol{0} & \text{pure Neumann case}.\end{cases} \tag{2.20}$$

Step 2:

$$\boldsymbol{v} = \boldsymbol{H}^{-1}\boldsymbol{f}^* + \begin{cases} z_1\boldsymbol{p}_1 + z_0\boldsymbol{p}_0 & \text{general case,} \\ -\eta a_0 g_- \boldsymbol{H}^{-1}\boldsymbol{e}_- - \eta a_N g_+ \boldsymbol{H}^{-1}\boldsymbol{e}_+ & \text{pure Neumann case.} \end{cases} \tag{2.21}$$

2.6. Solving second order differential equations

The constructed $\boldsymbol{H}^{-1}$ operator can be used as a numerical solution operator for the second order boundary value problem of the form

$$\begin{gathered} a^*(x)u''(x) + b^*(x)u'(x) + c^*(x)u(x) = f^*(x),\ x \in \mathsf{I}, \quad a^*(x) \neq 0, \\ \alpha_\pm u(\pm 1) \pm \beta_\pm u'(\pm) = g_\pm. \end{gathered} \tag{2.22}$$

Notice that (2.22) can be transformed into the form

$$(\theta(x)u'(x))' + \theta(x)\frac{c^*(x)}{a^*(x)}u(x) = \theta(x)\frac{f^*(x)}{a^*(x)}, \quad \theta(x) = \exp\left(\int \frac{b^*(x)}{a^*(x)}\,dx\right), \tag{2.23}$$

by introducing the integrating factor $\theta(x)$ playing the role of $a(x)$ in (2.5). If θ is not available analytically, we approximate θ numerically as

$$\boldsymbol{\theta} = [\theta_0, \ldots, \theta_N] = \left[e^{\mu_0}, \ldots, e^{\mu_N}\right]^T, \quad [\mu_0, \ldots, \mu_N]^T = \boldsymbol{D}_-^{-1}\left[\frac{b_0^*}{a_0^*}, \ldots, \frac{b_N^*}{a_N^*}\right]^T.$$

Employing $\boldsymbol{L}$ with the diagonal elements of $\boldsymbol{A}$ by θ_i and denoting

$$\boldsymbol{C} = \operatorname{diag}\left(\frac{\theta_0 c_0^*}{a_0^*}, \frac{\theta_1 c_1^*}{a_1^*}, \ldots, \frac{\theta_N c_N^*}{a_N^*}\right), \quad \boldsymbol{f} = \left[\theta_0\frac{f_0^*}{a_0^*}, \theta_1\frac{f_1^*}{a_1^*}, \ldots, \theta_N\frac{f_N^*}{a_N^*}\right]^T,$$

we can apply (2.20) and (2.21) to solve (2.23).

3. Numerical validations and discussions

In this section, we present results obtained by the proposed method. In each numerical experiment, an exact solution $u(x)$, a coefficient function $a(x)$, and a set of parameters $(\alpha_\pm, \beta_\pm)$, are specified. With these pieces of information in hand we compute the corresponding $f(x)$ and $g_\pm$. To examine the performance of the derived inverse operator when solving a problem, we measure the discrete l_2 error, defined as $e_2(N) = \left(\sum_{i=0}^N |u(x_i) - v(x_i)|^2\omega_i\right)^{1/2}$, and the maximum pointwise error, defined as $e_\infty(N) = \max_{i=0,\ldots,N}|u(x_i) - v(x_i)|$, where $v(x)$ is a numerical solution obtained by the scheme on a grid mesh characterized by N which is the degree of the approximation polynomial $v(x)$.

Our first numerical experiment is solving the boundary value problem

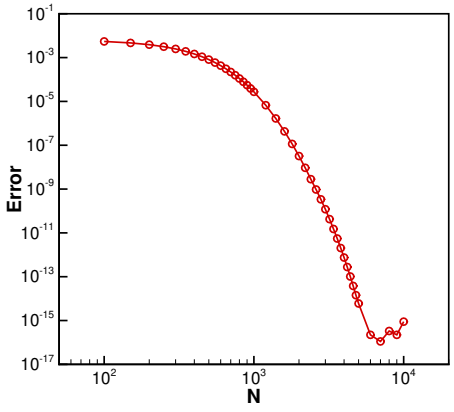
$$u''(x) + \frac{1}{ax^2+1}u'(x) = f(x),\ x \in (-1,1), \quad u(\pm 1) = g_{\pm} \tag{3.1}$$

with an exact solution $u(x)$ given as

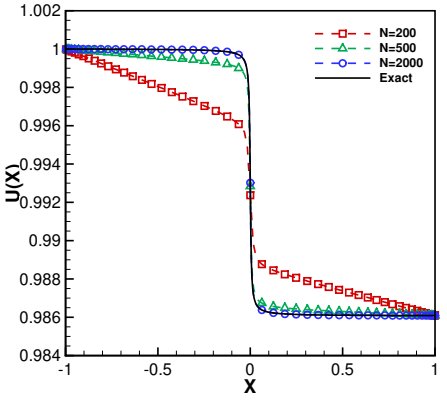
$$u(x) = \exp\left(-\frac{\tan^{-1}(\sqrt{a}x) + \tan^{-1}(\sqrt{a})}{\sqrt{a}}\right), \quad a = 5 \times 10^4.$$

The function $f(x)$ and the values of g_- and g_+ are computed by $u(x)$.

A convergence study for this problem and computed solution profiles are presented in Figure 3.1. We observe that the error decays exponentially as the grid resolution increases. Moreover, the accuracy of the numerical solutions remains at a machine accuracy level during further grid refinements. As pointed out in [20], the function $u(x)$ forms a thin boundary layer near $x = 0$, and a dense grid is needed to resolve $u(x)$ accurately. Similarly, we need to use a dense grid mesh to resolve u to a machine accuracy level, as shown in Figure 3.1(a) which agrees with the result in [20].



(a) Errors of numerical solutions computed by $\boldsymbol{L}^{-1}\boldsymbol{f}$.



(b) Computed solution profiles for different values of N.

Figure 3.1: Convergence results of (3.1).

The next numerical experiment is solving the problem

$$u''(x) - b^2 u(x) = -e^{ax},\ x \in (-1,1), \quad u(\pm 1) = 0 \tag{3.2}$$

for the parameter sets $(a,b) = (1,0)$ and $(a,b) = (2,1)$ with an exact smooth solution

$$u(x) = \frac{e^{ax} + e^{-bx}\sinh(a-b)\,\mathrm{csch}(2b) - e^{bx}\sinh(a+b)\,\mathrm{csch}(2b)}{b^2 - a^2}.$$

Notice that for the case $(a,b) = (1,0)$ the corresponding exact solution is obtained through taking the limit of the general expression as $b \to 0$, leading to $u(x) = -e^x$.

	Present		JG1	JG2
N	$e_2(N)$	$e_\infty(N)$	$e_\infty(N)$	$e_\infty(N)$
4	1.42E-04	1.34E-04	2.45E-03	7.96E-04
8	5.81E-10	5.34E-10	3.93E-08	1.57E-08
16	8.66E-16	8.88E-16	7.64E-16	7.13E-16
32	2.74E-16	4.44E-16	-	-
64	6.22E-16	8.88E-16	-	-
128	1.45E-15	1.55E-15	-	-
256	3.05E-16	8.88E-16	-	-
512	6.91E-16	1.11E-15	-	-
1024	8.11E-16	1.33E-15	-	-

Table 3.1: Convergence results for the Poisson problem formulated by (3.2) with $a = 1$ and $b = 0$. Reference results, labeled JG1 and JG2, are collected from the Jacobi polynomial based spectral-Galerkin methods [7]. (see Table 1 in [7] with the parameters $\alpha = \beta = 0$).

	Present		JG1	JG2
N	$e_2(N)$	$e_\infty(N)$	$e_\infty(N)$	$e_\infty(N)$
4	3.05E-03	3.43E-03	2.63E-02	8.45E-03
8	2.05E-07	1.95E-07	7.14E-06	2.81E-06
16	4.06E-16	4.44E-16	6.41E-15	2.93E-15
32	3.83E-16	6.10E-16	-	-
64	6.27E-16	9.43E-16	-	-
128	7.86E-16	1.11E-15	-	-
256	3.66E-16	8.32E-16	-	-
512	4.65E-16	9.99E-16	-	-
1024	6.73E-16	1.55E-15	-	-

Table 3.2: Convergence results for the Helmholtz problem formulated by (3.2) with $a = 2$ and $b = 1$. Reference results, labeled JG1 and JG2, are collected from the Jacobi polynomial based spectral-Galerkin methods [7]. (see Table 1 in [7] with the parameters $\alpha = \beta = 0$).

Tables 3.1 and 3.2 present convergence studies of our method, with results from the Jacobi polynomial based spectral-Galerkin method [7] for comparison. For the grid resolutions $N = 4$ and $N = 8$, our method is clearly better than the Jacobi polynomial based spectral-Galerkin method [7], for both parameter cases $(a, b) = (1, 0)$ and $(a, b) = (2, 1)$. As the grid resolution increases to $N = 16$, all the numerical solutions have arrived at the machine error accuracy level. As the grid solution N further increases, the accuracy of each numerical solution computed by the present method remains at the machine error level.

The third numerical experiment is solving the problem

$$u''(x) + \cos(3x)u'(x) + \sin(3x)u(x) = f(x),\ x \in (-1, 1), \quad u(\pm 1) = \sin(\pm 6) \tag{3.3}$$

with an exact solution given as $u(x) = \sin(6x)$. This problem was solved by the arbitrary grid based integration preconditioning matrix method [13]. A convergence study is presented in Table 3.3, with reference results provided in [13] for comparison.

	Present		GMH-PS	H-PS	PS
N	$e_2(N)$	$e_\infty(N)$	$e_\infty(N)$	$e_\infty(N)$	$e_\infty(N)$
8	1.19E-01	1.34E-01	1.70E+01	1.10E-01	2.10E+01
16	6.46E-05	7.87E-05	3.80E-02	1.60E-06	1.50E-01
32	9.70E-12	9.67E-12	5.40E-11	1.60E-05	1.00E-05
64	2.86E-15	4.38E-15	8.90E-12	No Conv.	No Conv.
128	2.65E-15	4.44E-15	-	-	-
256	2.35E-15	5.21E-15	-	-	-
512	7.10E-15	1.04E-14	-	-	-
1024	6.08E-15	1.26E-14	-	-	-

Table 3.3: Convergence results for the Helmholtz problem formulated by (3.3). Reference results, labeled GMH-PS, H-PS, and PS, were obtained by the arbitrary grid based pseudospectral integration precondition matrix [13], the pseudospectral integration preconditioning matrix method [15], and the traditional pseudospectral method, respectively. All these reference results were reported in [13, Table 5]. In [13] the problem was solved by iterative methods, and those computations which failed to converge were labeled as No Conv..

It is clearly seen from the computed errors that the present method is better, except for the case $N = 16$ by the method [15]. Furthermore, the accuracy of the numerical

solutions obtained by the present method is preserved and not ruined by round off errors, even using a dense grid mesh.

The fourth numerical experiment is solving the Helmholtz problem

$$\varepsilon u''(x) - xu'(x) - u(x) = 0,\ x \in (-1,1), \quad u(\pm 1) = 0 \tag{3.4}$$

for $\varepsilon = 0.1$. This problem was solved by the ultraspherical polynomial based integration preconditioning matrix method [10], with an exact solution given as $u(x) = e^{\frac{x^2-1}{2\varepsilon}}$ which forms a boundary layer of width $\mathcal{O}(1/\varepsilon)$ in the vicinity of each end point.

Table 3.4 presents a convergence study of our method, with reference results provided in [10] for comparison. As shown in the results, for all the grid resolutions used in the computations, our method is as good as the ultraspherical polynomial based integration preconditioning matrix method [10], and both integration type methods are better than the traditional pseudospectral method.

	Present		PLP	LP
N	$e_2(N)$	$e_\infty(N)$	$e_\infty(N)$	$e_\infty(N)$
8	7.59E-03	7.63E-03	-	-
16	8.42E-06	7.81E-06	-	-
32	6.35E-14	5.76E-14	-	-
64	8.03E-16	1.99E-15	9.99E-16	1.59E-13
128	4.71E-16	9.99E-16	1.22E-15	9.82E-14
256	4.43E-16	1.66E-15	1.67E-15	6.61E-13
512	8.39E-16	2.77E-15	2.22E-15	1.11E-11
1024	7.23E-16	2.66E-15	3.71E-15	2.72E-11

Table 3.4: Convergence results for the Helmholtz problem formulated by (3.4) with $\varepsilon = 0.1$. Reference results, labeled PLP and LP, are collected from the ultraspherical polynomial based integration preconditioning matrix methods [10, Table 3].

Our next numerical experiment is solving the boundary value problem

$$u''(x) - (1 + \sin x)u'(x) + e^x u(x) = f(x),\ x \in \mathsf{I}, \quad u(\pm 1) = 1 \tag{3.5}$$

with an exact solution given as $u(x) = e^{(x^2-1)/2}$. The corresponding f is computed by the given u. We use (2.20) and (2.21) to solve the discretized problem.

A convergence study is presented in Table 3.5. Similar to the previous examples, the numerical error vanishes exponentially as the grid resolution increases. Furthermore, the

measured errors are as small as those reported in [30], indicating that the present method is effective and comparable to the approach [30], even during further grid refinements.

	Present		BCOL	P-LCOL
N	$e_2(N)$	$e_\infty(N)$	$e_\infty(N)$	$e_\infty(N)$
4	1.84e-03	1.75e-03	-	-
8	5.49e-07	7.93e-07	-	-
16	1.94e-13	1.98e-13	-	-
32	8.70e-16	8.88e-16	-	-
64	8.10e-16	1.11e-15	5.55e-16	1.67e-15
128	1.02e-15	1.22e-15	1.11e-15	2.44e-15
256	1.79e-15	1.88e-15	1.11e-15	2.55e-15
512	6.95e-15	8.10e-15	1.89e-15	4.77e-15
1024	8.37e-15	6.43e-15	3.44e-15	1.15e-14

Table 3.5: Convergence results for (3.5). Reference results are from [30] by the Birkhoff collocation (BCOL) scheme and the preconditioned Lagrange collocation (P-LCOL) scheme.

In theory, the accuracy of a numerical solution is affected by the truncation errors and the round off errors. In these experiments the results show that the present approach is better than the other methods in most cases, when round off errors are much smaller than the truncation errors of the numerical solutions. For computations based on dense grid meshes and when round off errors and the truncation errors of the numerical solutions are of the same order of magnitude, our method is as good as the other methods.

Encouraged by the performances of our method in the previous experiments for solving linear differential equations, we further test the performance of the present method for solving a nonlinear differential equation.

The following experiment is solving the problem described by the charge conserving Poisson-Boltzmann equation of the form

$$u''(\xi) = \frac{\sinh(u(\xi))}{d^{-1}\int_{-d}^{d} e^{u(\xi)}\, d\xi}, \quad \xi \in \left(-\frac{d}{2}, \frac{d}{2}\right), \quad u\left(\pm\frac{d}{2}\right) = \pm V.$$

Introducing the linear coordinate transformation, $\xi(x) = xd/2$, and denoting the transformed variable $u(\xi(x))$ by $v(x)$, we have the transformed problem as follows:

$$\frac{d^2v(x)}{dx^2} = \frac{d^3}{8}\frac{\sinh(v(x))}{\int_{-1}^{1} e^{v(x)}\, dx}, \quad x \in (-1, 1), \quad v(\pm 1) = \pm V. \tag{3.6}$$

Let $\boldsymbol{v}$ be the grid vector of $v(x)$ and $\boldsymbol{f}$ be the grid vector defined as

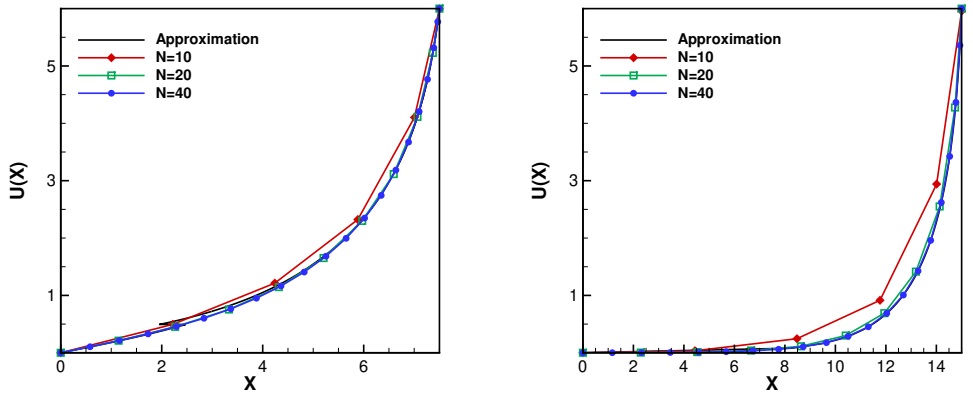
$$\boldsymbol{f} = \begin{bmatrix} f_0 & f_1 & \cdots & f_N \end{bmatrix}, \quad f_i = \frac{\sinh(v(x_i))}{\sum_{k=0}^{N} e^{v(x_i)}\omega_i}.$$

We solve (3.6) by the following scheme

$$\boldsymbol{v} = \boldsymbol{L}^{-1}\left[\frac{d^3}{8}\boldsymbol{f} - \left(\tau_-\eta^2\boldsymbol{e}_- + \chi_-\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_-\right)V_- - \left(\tau_+\eta^2\boldsymbol{e}_+ - \chi_+\boldsymbol{M}^{-1}\boldsymbol{D}^T\boldsymbol{e}_+\right)V_+\right]$$

by a nonlinear equation solver.

Figure 3.2 presents computed solutions for various values of N. For certain particular parameter ranges a good approximation of the Dirichlet problem has been derived [9], and we use the derived formula to compute an approximation solution for comparison. In Figure 3.2 it is shown that the difference between the numerical solution and the approximation solution becomes smaller as the grid resolution N increases.



(a) (b)

Figure 3.2: Numerical solutions of (3.6) with problem parameters given by $V = 6$ and $d = 15$ (a) and $d = 30$ (b). Approximation solutions are computed by the formula given in [9].

Note that the plotted approximation solution has a zigzag for $2 \le x \le 2.5$ in Figure 3.2(a). It is caused by the approximation formula provided in [9] (Eq. (3.15b) in the reference), which is for approximating $x(u)$ (the inverse of u) instead of $u(x)$. The formula was derived by singular perturbation theory, with range splitting and asymptotic matching. Thus, depending on the value of u, different approximation formula were applied. As a result, the plotted curve $x(u)$ (black line) has a jump discontinuity at the matching point in u (approximately at $u = 0.5$), forming a zigzag shape in the plot in this case. For further information about the property of the approximation solution $x(u)$ we refer the readers to [9].

We close this section by demonstrating the performance of the derived $\boldsymbol{H}^{-1}$ operator when solving a time-dependent partial differential equation. Consider $u(x,t)$ satisfying the model problem

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), & x \in (-1,1),\ t > 0, \\ u(x,0) = u_0(x), & x \in [-1,1], \\ \pm \frac{\partial u(\pm 1,t)}{\partial x} = g_{\pm}(t), & t \geq 0 \end{cases} \tag{3.7}$$

with an exact solution given as

$$u(x,t) = e^{-\pi^2 t}\sin(\pi x) + e^{-4\left(x-\frac{\pi}{12}\right)^2}\cos(6\pi x).$$

The functions $f(x,t)$, $u_0(x)$, and $g_{\pm}(t)$ are evaluated by the given exact solution.

We discretize the problem by the present pseudospectral method in space and the Crank-Nicholson difference method in time, and have the scheme as

$$\begin{aligned} \frac{\boldsymbol{v}^{n+1}-\boldsymbol{v}^n}{\Delta t} &= \boldsymbol{L}\left(\frac{\boldsymbol{v}^{n+1}+\boldsymbol{v}^n}{2}\right) + \frac{\boldsymbol{f}^{n+1}+\boldsymbol{f}^n}{2} + \eta \boldsymbol{e}_- \frac{g_-^{n+1}+g_-^n}{2} + \eta \boldsymbol{e}_+ \frac{g_+^{n+1}+g_+^n}{2}, \\ \boldsymbol{v}^0 &= [u_0(x_0), u_0(x_1), \ldots, u_0(x_N)]^T, \end{aligned} \tag{3.8}$$

where Δt is the time step, the superscript n denotes the integer time level, $\boldsymbol{v}^n = [v_0^n, \ldots, v_N^n]^T$ is the solution grid vector at discrete time $t_n = n\Delta t$ with v_i^n being the value approximating $u(x_j, t_n)$, $\boldsymbol{f}^n = [f_0^n, \ldots, f_N^n]^T$ is the grid vector of f at time level t_n, and $\boldsymbol{L}$ is the BCP pseudospectral Laplace operator defined in (2.9).

The fully discrete scheme (3.8) is stable independent of the size of the time step. To show this fact it is sufficient to consider the homogeneous version of the scheme. Multiplying $(\boldsymbol{v}^{n+1}+\boldsymbol{v}^n)\boldsymbol{M}$ to the scheme from the left and invoking that $\boldsymbol{M}$ is symmetric positive definite and $\boldsymbol{ML}$ is semi-negative definite, we have

$$(\boldsymbol{v}^{n+1})^T\boldsymbol{M}\boldsymbol{v}^{n+1} = (\boldsymbol{v}^n)^T\boldsymbol{M}\boldsymbol{v}^n + \frac{\Delta t}{2}\big(\boldsymbol{v}^{n+1}+\boldsymbol{v}^n\big)^T\boldsymbol{M}\boldsymbol{L}\big(\boldsymbol{v}^{n+1}+\boldsymbol{v}^n\big) \leq (\boldsymbol{v}^n)^T\boldsymbol{M}\boldsymbol{v}^n.$$

As a result, the discrete energy norm of the numerical solution at time level t_n is bounded as follows:

$$(\boldsymbol{v}^n)^T\boldsymbol{M}\boldsymbol{v}^n \leq (\boldsymbol{v}^{n-1})^T\boldsymbol{M}\boldsymbol{v}^{n-1} \leq \cdots \leq (\boldsymbol{v}^0)^T\boldsymbol{M}\boldsymbol{v}^0.$$

Thus, for a fixed terminal time $T = n\Delta t$ the numerical solution $\boldsymbol{v}^n$ is bounded by the initial data and is independent of the time step, during grid refinements.

For the sake of computation accuracy, we still need to use a proper time step size to advance the solution in time, although the scheme is unconditional stable. For this numerical experiment we use $\Delta t = 1/N$ for computations. As we will see soon, this choice of time step gives satisfactory results.

To compute $\boldsymbol{v}^{n+1}$ we rewrite the system of equations as

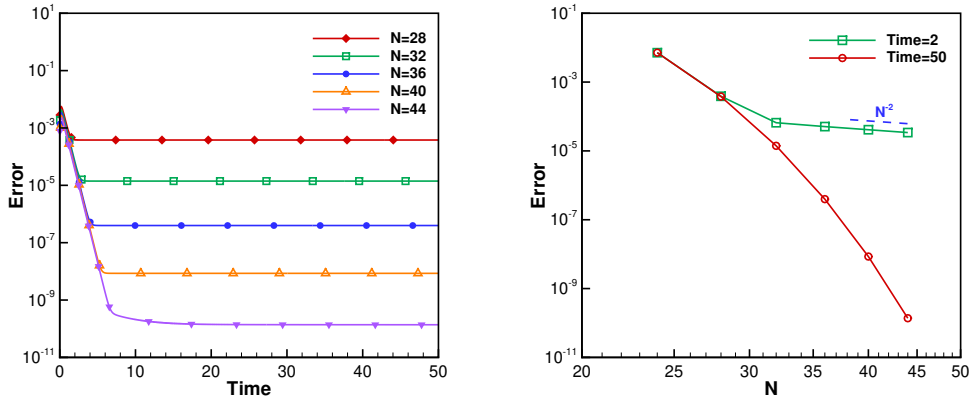
$$\begin{aligned}\boldsymbol{v}^{n+1} = &-\boldsymbol{v}^n - \boldsymbol{H}^{-1}\left(\frac{4}{\Delta t}\boldsymbol{v}^n + \boldsymbol{f}^{n+1} + \boldsymbol{f}^n\right) \\ &- \boldsymbol{H}^{-1}\boldsymbol{e}_-\eta\big(g_-^{n+1} + g_-^n\big) - \boldsymbol{H}^{-1}\boldsymbol{e}_+\eta\big(g_+^{n+1} + g_+^n\big),\end{aligned} \tag{3.9}$$

where $\boldsymbol{H}^{-1}$ is given as

$$\boldsymbol{H}^{-1} = \left(\boldsymbol{L} - \frac{2}{\Delta t}\boldsymbol{I}\right)^{-1},$$

and is formulated by the method described in Section 2.4. To examine the performance of the scheme, we measure the approximation error defined by $e(N, t_n) = \big(\sum_{i=0}^{N} |u(x_i, t_n) - v_i^n|^2 \omega_i\big)^{1/2}$.

Figure 3.3(a) presents the error history curves resulting from computing numerical solutions by different grid resolutions. For a given grid resolution N the corresponding curve indicates that the computed solution gradually evolves to the steady state solution within a certain level of accuracy, where the higher the grid resolution, the better the solution accuracy. Figure 3.3(b) shows the convergence rate of the numerical solution at times $t = 2$ and $t = 50$. At $t = 2$ the solution profile has not yet arrived at the steady state, and hence, the convergence rate is of second order due to the Crank-Nicholson difference in time. On the other hand at $t = 50$ the solution profile has reached the steady state, and thus, as the grid resolution increases, the error decays exponentially due to the pseudospectral discretization in space.



(a) (b)

Figure 3.3: (a) Error history curves of the model heat problem (3.7) solved by (3.9) with different grid resolutions. (b) Measured errors at time $t = 2$ and $t = 50$ for different values of N. We take time step as $\Delta = 1/N$ in each computation.

4. Conclusions and future plans

We have presented a framework based on stepwise integrations and low-rank updates to seek an inverse pseudospectral matrix for the Helmholtz differential operator. The resultant inverse matrix was used to solve general second order differential equations, and the computation results clearly illustrated the performance of the operators as expected. Before closing, we would like to address potential applications of the present framework for seeking integration preconditioning matrices.

The present framework for seeking inverse matrices is applied in nodal spaces. Thus, for the basic mixed differential operator, $\frac{d}{dx}a(x)\frac{d}{dx}$, the linkage between differential operators and sandwiched variable coefficients is easily decoupled, and the inverse matrix for the mixed differential operator is formed by a matrix product of the inverses of the decoupled ones. Consequently, existing inverse matrices, developed by other orthogonal polynomial methods for the first order differential operator, can be adopted into the present framework as basic elements to formulate different orthogonal polynomials based operating matrices for the mixed operator as well. Furthermore, using existing inverse spectral/pseudospectral matrices as building blocks in the present divide-and-conquer approach, we can further seek integration preconditioning matrices for other high-order pure, mixed, or even nonlinear differential operators.

We may further apply the present framework to devise preconditioning techniques for multidomain schemes based on pseudospectral, discontinuous Galerkin and summation-by-parts finite difference methods. Similar to the model single domain scheme present in this study, multidomain schemes are composed of element-wise differentiation operators which are commonly products of basic differentiation matrices and element-wise low-rank boundary operators which relate field values between adjacent elements. By exploring these properties within these schemes as shown in this study, we strongly believe that efficient and accurate numerical procedures may be devised. We hope to report these works in the near future.

A. General $\boldsymbol{L}^{-1}$

Following a parallel procedure as shown in Section 2.3, we now proceed to finding the matrix $\boldsymbol{K}^{-1}\boldsymbol{\Lambda}$ to formulate the $\boldsymbol{L}^{-1}$ matrix for the general case. We factor $\boldsymbol{Q}$ described in (2.13) as

$$\boldsymbol{Q} = \widehat{\boldsymbol{\Phi}}\widehat{\boldsymbol{\Psi}}^T,$$

where $\widehat{\boldsymbol{\Phi}}$ and $\widehat{\boldsymbol{\Psi}}^T$ are, respectively, given as

$$\widehat{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{D}_-\boldsymbol{e}_- & \eta\boldsymbol{e}_- & \eta\boldsymbol{e}_+ & \boldsymbol{D}_-\boldsymbol{e}_+ \end{bmatrix},$$

and

$$\widehat{\boldsymbol{\Psi}}^T = \begin{bmatrix} \eta a_0 \chi_-(\alpha_- \boldsymbol{e}_-^T - \beta_- \boldsymbol{e}_-^T \boldsymbol{D}_+) \\ -\tau_-\alpha_-\eta a_0 \boldsymbol{e}_-^T - (1-\tau_-\beta_-\eta) a_0 \boldsymbol{e}_-^T \boldsymbol{D}_+ \\ (\chi_+ - \tau_+)(\alpha_+ + \beta_+\eta)\eta a_N \boldsymbol{e}_+^T + (\chi_+ - \tau_+)\beta_+\eta a_N \boldsymbol{e}_+^T \boldsymbol{D}_+ \\ (1 - \chi_+(\alpha_+ + \beta_+\eta))\eta a_N \boldsymbol{e}_+^T - \chi_+\beta_+\eta a_N \boldsymbol{e}_+^T \boldsymbol{D}_+ \end{bmatrix}.$$

Employing the inverse operators, $\boldsymbol{D}_-^{-1}$ and $\boldsymbol{D}_+$, we compute $\boldsymbol{D}_-^{-1}\widehat{\boldsymbol{\Phi}}$ and $\widehat{\boldsymbol{\Psi}}^T \boldsymbol{D}_+^{-1}$ and have

$$\boldsymbol{\Phi} = \boldsymbol{D}_-^{-1}\widehat{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{e}_- & \boldsymbol{p}_0 & \boldsymbol{p}_n & \boldsymbol{e}_+ \end{bmatrix},$$

and

$$\boldsymbol{\Psi}^T = \widehat{\boldsymbol{\Psi}}^T \boldsymbol{D}_+^{-1} = \begin{bmatrix} -\chi_-\alpha_-\eta a_0 \boldsymbol{p}_0^T - \chi_-\beta_-\eta^2 a_0 \boldsymbol{e}_-^T \\ +\tau_-\alpha_-\eta a_0 \boldsymbol{p}_0^T - (1-\tau_-\beta_-\eta)\eta a_0 \boldsymbol{e}_-^T \\ -(\chi_+ - \tau_+)(\alpha_+ + \beta_+\eta)\eta a_N \boldsymbol{p}_N^T + (\chi_+ - \tau_+)\beta_+\eta^2 a_N \boldsymbol{e}_+^T \\ -(1 - \chi_+(\alpha_+ + \beta_+\eta))\eta a_N \boldsymbol{p}_N^T - \chi_+\beta_+\eta^2 a_N \boldsymbol{e}_+ \end{bmatrix} \boldsymbol{M} = \boldsymbol{\Lambda}\boldsymbol{\Phi}^T\boldsymbol{M},$$

where with the use of the relationships in (2.7) the matrix $\boldsymbol{\Lambda}$ is given as

$$\boldsymbol{\Lambda} = \eta \begin{bmatrix} -\chi_-\beta_-\eta a_0 & -\chi_-\alpha_- a_0 & 0 & 0 \\ -\chi_-\alpha_- a_0 & \tau_-\alpha_- a_0 & 0 & 0 \\ 0 & 0 & -(\chi_+ - \tau_+)(\alpha_+ + \beta_+\eta)a_N & (\chi_+ - \tau_+)\beta_+\eta a_N \\ 0 & 0 & -(\chi_+ - \tau_+)\beta_+\eta a_N & -\chi_+\beta_+\eta a_N \end{bmatrix}.$$

To find $\boldsymbol{K}^{-1}\boldsymbol{\Lambda}$ we compute $\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}^{-1}$, and the results are

$$\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{A}^{-1}\boldsymbol{\Phi} = \frac{1}{\eta}\begin{bmatrix} \frac{1}{a_0} & \frac{1}{a_0} & \frac{(-1)^N}{a_0} & 0 \\ \frac{1}{a_0} & \kappa_2 & \kappa_1 & \frac{1}{a_N} \\ \frac{(-1)^N}{a_0} & \kappa_1 & \kappa_0 & \frac{1}{a_N} \\ 0 & \frac{1}{a_N} & \frac{1}{a_N} & \frac{1}{a_N} \end{bmatrix}, \quad \kappa_\nu = \sum_{i=0}^{N} \frac{1}{a_i (P_N(x_i))^\nu},$$

and

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{\eta}\begin{bmatrix} -\frac{\tau_-}{\chi_- a_0} & -\frac{1}{a_0} & 0 & 0 \\ -\frac{1}{a_0} & \frac{\beta_-\eta}{\alpha_- a_0} & 0 & 0 \\ 0 & 0 & -\frac{\chi_+}{(\chi_+ - \tau_+)a_N} & -\frac{1}{a_N} \\ 0 & 0 & -\frac{1}{a_N} & -\frac{1}{a_N}\left(\frac{\alpha_+}{\beta_+\eta} + 1\right) \end{bmatrix}.$$

Thus, we have $\boldsymbol{\Lambda}^{-1}\boldsymbol{K}$ as

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{K} = \boldsymbol{\Lambda}^{-1} + \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{A}^{-1} \boldsymbol{\Phi} = \frac{1}{\eta} \begin{bmatrix} \left(1 - \frac{\tau_-}{\chi_-}\right) \frac{1}{a_0} & 0 & \frac{(-1)^N}{a_0} & 0 \\ 0 & \kappa_2 + \frac{\beta_- \eta}{\alpha_- a_0} & \kappa_1 & \frac{1}{a_N} \\ \frac{(-1)^N}{a_0} & \kappa_1 & \kappa_0 - \frac{\chi_+}{(\chi_+ - \tau_+) a_N} & 0 \\ 0 & \frac{1}{a_N} & 0 & -\frac{\alpha_+}{\beta_+ \eta a_N} \end{bmatrix}.$$

Then the inverse of $\boldsymbol{\Lambda}^{-1}\boldsymbol{K}$ is found as

$$\boldsymbol{K}^{-1}\boldsymbol{\Lambda} = \frac{\eta}{T} \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{12} & t_{22} & t_{23} & t_{24} \\ t_{13} & t_{23} & t_{33} & t_{34} \\ t_{14} & t_{24} & t_{34} & t_{44} \end{bmatrix},$$

where $\boldsymbol{T}$ is given as

$$\begin{aligned} T = {} & \frac{(\chi_- - \tau_-)(\chi_+ - \tau_+)}{a_0 a_N} \left((\kappa_1^2 - \kappa_0 \kappa_2) \alpha_- \alpha_+ - \frac{\alpha_+ \beta_- \eta \kappa_0}{a_0} \right) \\ & + \frac{1}{a_0 a_N} \left(\kappa_2 \alpha_- \alpha_+ + \frac{\alpha_+ \beta_- \eta}{a_0} \right) \left(\frac{(\chi_- - \tau_-)\chi_+}{a_N} + \frac{(\chi_+ - \tau_+)\chi_-}{a_0} \right) \\ & - \frac{\alpha_- \beta_+ \eta}{a_0 a_N^2} \left((\chi_- - \tau_-)(\chi_+ - \tau_+)\kappa_0 - \frac{(\chi_- - \tau_-)\chi_+}{a_N} - \frac{(\chi_+ - \tau_+)\chi_-}{a_0} \right), \end{aligned}$$

and t_{ij} for $1 \le i \le 4$ and $i \le j \le 4$ are found as follows:

$$\begin{aligned} t_{11} = {} & \frac{\chi_-}{a_N}(\chi_+ - \tau_+)(\kappa_1^2 - \kappa_0 \kappa_2)\alpha_- \alpha_+ + \frac{\chi_- \chi_+}{a_N^2}(\kappa_2 \alpha_- \alpha_+) \\ & - \frac{\chi_-}{a_N} \left(\frac{\alpha_+ \beta_- \eta}{a_0} + \frac{\alpha_- \beta_+ \eta}{a_N} \right) \left(\kappa_0 (\chi_+ - \tau_+) - \frac{\chi_+}{a_N} \right), \\ t_{12} = {} & \frac{(-1)^N \chi_- \alpha_- \beta_+ \eta \kappa_1}{a_0} \left(\kappa_0 (\chi_+ - \tau_+) - \frac{\chi_+}{a_N} \right), \\ t_{13} = {} & \frac{(-1)^N \chi_- (\chi_+ - \tau_+)}{a_0 a_N} \left(\frac{\alpha_- \beta_+ \eta}{a_N} + \frac{\alpha_+ \beta_- \eta}{a_0} + \kappa_2 \alpha_- \alpha_+ \right), \\ t_{14} = {} & \frac{(-1)^{N+1} \chi_- (\chi_+ - \tau_+) \kappa_1 \alpha_- \beta_+ \eta}{a_0 a_N}, \\ t_{22} = {} & \left(\frac{\chi_+ (\chi_- - \tau_-)}{a_N} + \frac{\chi_- (\chi_+ - \tau_+)}{a_0} - (\chi_- - \tau_-)(\chi_+ - \tau_+)\kappa_0 \right) \frac{\alpha_- \alpha_+}{a_0 a_N}, \\ t_{23} = {} & \frac{(\chi_+ - \tau_+)(\chi_- - \tau_-)}{a_0 a_N} \kappa_1 \alpha_- \alpha_+, \\ t_{24} = {} & \frac{\alpha_- \beta_+ \eta}{a_0 a_N} \left(\frac{\chi_- (\chi_+ - \tau_+)}{a_0} + \frac{\chi_+ (\chi_- - \tau_-)}{a_N} - \kappa_0 (\chi_- - \tau_-)(\chi_+ - \tau_+) \right), \\ t_{33} = {} & -\frac{(\chi_- - \tau_-)(\chi_+ - \tau_+)}{a_0 a_N} \left(\frac{\alpha_+ \beta_- \eta}{a_0} + \frac{\alpha_- \beta_+ \eta}{a_N} + \kappa_2 \alpha_- \alpha_+ \right), \end{aligned}$$

$$t_{34} = (\chi_+ - \tau_+)(\chi_- - \tau_-)\frac{\alpha_-\beta_+\eta\kappa_1}{a_0 a_N},$$
$$t_{44} = -\frac{\alpha_-\beta_+\eta}{a_0}\left((\kappa_1^2 - \kappa_0\kappa_2)(\chi_- - \tau_-)(\chi_+ - \tau_+) + \kappa_2\left(\frac{\chi_+(\chi_- - \tau_-)}{a_N} + \frac{\chi_-(\chi_+ - \tau_+)}{a_0}\right)\right)$$
$$+ \frac{\beta_-\beta_+\eta^2}{a_0^2}\left(\kappa_0(\chi_- - \tau_-)(\chi_+ - \tau_+) - \frac{\chi_+(\chi_- - \tau_-)}{a_N} - \frac{\chi_-(\chi_+ - \tau_+)}{a_0}\right).$$

Acknowledgments

We thank the anonymous reviewers for their many insightful comments and suggestions. Li was supported in part by the Ministry of Science and Technology of Taiwan under grant 106-2115-M-030-001. Teng was supported in part by the Ministry of Science and Technology of Taiwan under grant 105-2115-M-005-007.

References

[1] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral Methods: Fundamentals in Single Domains*, Scientific Computation, Springer, Berlin, 2006.

[2] ———, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*, Scientific Computation, Springer, Berlin, 2007.

[3] M. H. Carpenter and D. Gottlieb, *Spectral methods on arbitrary grids*, J. Comput. Phys. **129** (1996), no. 1, 74–86.

[4] E. H. Doha, *On the construction of recurrence relations for the expansion and connection coefficients in series of Jacobi polynomials*, J. Phys. A **37** (2004), no. 3, 657–675.

[5] E. H. Doha and W. M. Abd-Elhameed, *Efficient spectral-Galerkin algorithms for direct solution of second-order equations using ultraspherical polynomials*, SIAM J. Sci. Comput. **24** (2002), no. 2, 548–571.

[6] ———, *Efficient spectral ultraspherical-dual-Petrov-Galerkin algorithms for the direct solution of* $(2n+1)$*th-order linear differential equations*, Math. Comput. Simulation **79** (2009), no. 11, 3221–3242.

[7] E. H. Doha and A. H. Bhrawy, *Efficient spectral-Galerkin algorithms for direct solution for second-order differential equations using Jacobi polynomials*, Numer. Algorithms **42** (2006), no. 2, 137–164.

[8] W. S. Don and D. Gottlieb, *The Chebyshev-Legendre method: Implementing Legendre methods on Chebyshev points*, SIAM J. Numer. Anal. **31** (1994), no. 6, 1519–1534.

[9] D. Elad and N. Gavish, *Finite domain effects in steady state solutions of Poisson-Nernst-Planck equations*, SIAM J. Appl. Math. **79** (2019), no. 3, 1030–1050.

[10] E. M. E. Elbarbary, *Integration preconditioning matrix for ultraspherical pseudospectral operators*, SIAM J. Sci. Comput. **28** (2006), no. 3, 1186–1201.

[11] D. Funaro and D. Gottlieb, *A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations*, Math. Comp. **51** (1988), no. 184, 599–613.

[12] ———, *Convergence results for pseudospectral approximations of hyperbolic systems by a penalty-type boundary treatment*, Math. Comp. **57** (1991), no. 196, 585–596.

[13] F. Ghoreishi and S. M. Hosseini, *A preconditioned implementation of pseudospectral methods on arbitrary grids*, Appl. Math. Comput. **148** (2004), no. 1, 15–34.

[14] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev. **31** (1989), no. 2, 221–239.

[15] J. S. Hesthaven, *Integration preconditioning of pseudospectral operators I: Basic linear operators*, SIAM J. Numer. Anal. **35** (1998), no. 4, 1571–1593.

[16] ———, *Spectral penalty methods*, Appl. Numer. Math. **33** (2000), no. 1-4, 23–41.

[17] J. S. Hesthaven and D. Gottlieb, *A stable penalty method for the compressible Navier-Stokes equations I: Open boundary contitions*, SIAM J. Sci. Comput. **17** (1996), no. 3, 579–612.

[18] J. S. Hesthaven, S. Gottlieb and D. Gottlieb, *Spectral Methods for Time-dependent Problems*, Cambridge Monographs on Applied and Computational Mathematics **21**, Cambridge University Press, Cambridge, 2007.

[19] Y.-T. Li, P.-Y. Lin and C.-H. Teng, *Inverse matrices for pseudospectral differentiation operators in polar coordinates by stepwise integrations and low-rank updates*, Appl. Numer. Math. **150** (2020), 519–535.

[20] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, SIAM Rev. **55** (2013), no. 3, 462–489.

[21] S. A. Orszag, *Numerical simulation of incompressible flows within simple boundaries: Accuracy*, J. Fluid Mech. **49** (1971), 75–112.

[22] ———, *Numerical simulation of incompressible flows within simple boundaries I: Galerkin (spectral) representations*, Studies in Appl. Math. **50** (1971), no. 4, 293–327.

[23] J. Shen, *Efficient spectral-Galerkin method I: Direct solvers of second- and fourth-order equations using Legendre polynomials*, SIAM J. Sci. Comput. **15** (1994), no. 6, 1489–1505.

[24] ———, *Efficient spectral-Galerkin method II: Direct solvers of second- and fourth-order equations using Chebyshev polynomials*, SIAM J. Sci. Comput. **16** (1995), no. 1, 74–87.

[25] ———, *A new dual-Petrov-Galerkin method for third and higher odd-order differential equations: Application to the KdV equation*, SIAM J. Numer. Anal. **41** (2003), no. 5, 1595–1619.

[26] J. Shen, T. Tang and L.-L. Wang, *Spectral Methods: Algorithms, analysis and applications*, Springer, Heidelberg, 2011.

[27] J. Shen and L.-L. Wang, *Legendre and Chebyshev dual-Petrov-Galerkin methods for hyperbolic equations*, Comput. Methods Appl. Mech. Engrg. **196** (2007), no. 37-40, 3785–3797.

[28] ———, *Some recent advances on spectral methods for unbounded domains*, Commun. Comput. Phys. **5** (2009), no. 2-4, 195–241.

[29] J. Shen, Y. Wang and J. Xia, *Fast structured direct spectral methods for differential equations with variable coefficients I: The one-dimensional case*, SIAM J. Sci. Comput. **38** (2016), no. 1, A28–A54.

[30] L.-L. Wang, M. D. Samson and X. Zhao, *A well-conditioned collocation method using a pseudospectral integration matrix*, SIAM J. Sci. Comput. **36** (2014), no. 3, A907–A929.

Yung-Ta Li
Department of Mathematics, Fu Jen Catholic University, New Taipei City 242, Taiwan
E-mail address: ytli@math.fju.edu.tw

Ping-Hsuan Tsai
Department of Computer Science, University of Illinois at Urbana-Campaign, Urbana, IL 61801, USA
E-mail address: pht2@illinois.edu

Chun-Hao Teng
Department of Applied Mathematics, National Chung Hsing University, Taichung 402, Taiwan
E-mail address: tengch@email.nchu.edu.tw