

Orthogonal Polynomials and Polynomial Approximations

3.1. Orthogonal polynomials

The orthogonal polynomials play the most important role in spectral methods, so it is useful to understand some general properties of the orthogonal polynomials.

Given an interval (a, b) and a weight function $\omega(x)$ which is positive on (a, b) and $w \in L^1(a, b)$, we define the weighted Sobolev space $L_\omega^2(a, b)$ by

$$L_\omega^2(a, b) = \left\{ f : \int_a^b f^2(x)\omega(x)dx < +\infty \right\}. \quad (3.1.1)$$

It is obvious that $(\cdot, \cdot)_\omega$ defined by

$$(f, g)_\omega := \int_a^b f(x)g(x)\omega(x)dx$$

is an inner product on $L_\omega^2(a, b)$. Hence, $\|f\|_{L_\omega^2} = (f, f)_\omega^{\frac{1}{2}}$. Hereafter, the subscript ω will be omitted from the notation when $\omega(x) \equiv 1$.

Two functions f and g are said to be *orthogonal* in $L_\omega^2(a, b)$ if

$$(f, g)_\omega = 0.$$

A sequence of polynomials $\{p_n\}_{n=0}^\infty$ with $\deg(p_n) = n$ is said to be *orthogonal* in $L_\omega^2(a, b)$ if

$$(p_i, p_j)_\omega = 0 \quad \text{for } i \neq j. \quad (3.1.2)$$

Since the orthogonality is not altered by a multiplicative constant, we may normalize the polynomial p_n so that the coefficient of x^n is one, i.e.

$$p_n(x) = x^n + a_{n-1}^{(n)}x^{n-1} + \cdots + a_0^{(n)}.$$

Such a polynomial is said to be *monic*.

3.1.1. Existence and uniqueness. Our immediate goal is to establish the existence of a sequence of orthogonal polynomials. Although we could, in principle, determine the coefficients $a_j^{(n)}$ of p_n in the natural basis by using the orthogonality conditions (3.1.2), it is computationally advantageous to express p_n in terms of lower-order orthogonal polynomials.

Let us denote

$$P_n := \text{span} \{1, x, x^2, \dots, x^n\}. \quad (3.1.3)$$

Then, if $\{p_k\}_{k=0}^{\infty}$ is a sequence of polynomials such that p_k is exactly of degree k , it is obvious by dimension argument that

$$P_n := \text{span}\{p_0, p_1, \dots, p_n\}. \quad (3.1.4)$$

A direct consequence of this result is the following:

LEMMA 3.1. *If the sequence of polynomials $\{p_k\}_{k=0}^{\infty}$ is orthogonal, then the polynomial p_{n+1} is orthogonal to any polynomial q of degree n or less.*

PROOF. This can be established by writing

$$q = b_n p_n + b_{n-1} p_{n-1} + \dots + b_0 p_0$$

and observing

$$(p_{n+1}, q)_{\omega} = b_n (p_{n+1}, p_n)_{\omega} + b_{n-1} (p_{n+1}, p_{n-1})_{\omega} + \dots + b_0 (p_{n+1}, p_0)_{\omega} = 0,$$

the last equality following from the orthogonality of the polynomials. \square

THEOREM 3.1. *For any given positive weight function $\omega(x) \in L^1(a, b)$, there exists a unique set of monic orthogonal polynomials $\{p_n\}$. More precisely, $\{p_n\}$ can be constructed as follows:*

$$p_0 = 1, \quad p_1 = x - \alpha_1 \quad \text{with} \quad \alpha_1 = \int_a^b \omega(x)x dx / \int_a^b \omega(x) dx,$$

and

$$p_{n+1} = (x - \alpha_{n+1})p_n - \beta_{n+1}p_{n-1}, \quad n \geq 1, \quad (3.1.5)$$

where

$$\alpha_{n+1} = \int_a^b x \omega p_n^2 dx / \int_a^b \omega p_n^2 dx$$

and

$$\beta_{n+1} = \int_a^b x \omega p_n p_{n-1} dx / \int_a^b \omega p_{n-1}^2 dx.$$

PROOF. We shall first establish the existence of monic orthogonal polynomials. We begin by computing the first two. Since p_0 is monic and of degree zero,

$$p_0(x) \equiv 1.$$

Since p_1 is monic and of degree one, it must have the form

$$p_1(x) = x - \alpha_1.$$

To determine α_1 , we use orthogonality:

$$0 = (p_1, p_0)_{\omega} = \int_a^b \omega(x)x dx - \alpha_1 \int_a^b \omega(x) dx.$$

Since the weight function is positive in the interval of integration, it follows that

$$\alpha_1 = \int_a^b \omega(x)x dx / \int_a^b \omega(x) dx.$$

In general we will seek p_{n+1} in the form

$$p_{n+1} = xp_n - \alpha_{n+1}p_n - \beta_{n+1}p_{n-1} - \gamma_{n+1}p_{n-2} - \cdots.$$

As in the construction of p_1 , we will use orthogonality to determine the coefficients above. To determine α_{n+1} , write

$$0 = (p_{n+1}, p_n)_\omega = (xp_n, p_n)_\omega - \alpha_{n+1}(p_n, p_n)_\omega - \beta_{n+1}(p_{n-1}, p_n)_\omega - \cdots.$$

By orthogonality, we have

$$\int_a^b x\omega p_n^2 dx - \alpha_{n+1} \int_a^b \omega p_n^2 dx = 0,$$

which gives

$$\alpha_{n+1} = \int_a^b x\omega p_n^2 dx / \int_a^b \omega p_n^2 dx.$$

For β_{n+1} , using the fact that $(p_{n+1}, p_{n-1})_\omega = 0$ gives

$$\beta_{n+1} = \int_a^b x\omega p_n p_{n-1} dx / \int_a^b \omega p_{n-1}^2 dx.$$

Next, we have

$$\gamma_{n+1} = \int_a^b \omega p_n x p_{n-2} dx / \int_a^b \omega p_{n-2}^2 dx = 0$$

since xp_{n-2} is of degree $n-1$, and thus orthogonal to p_n . Likewise, the coefficients of p_{n-3}, p_{n-4}, \dots are all zero.

The above procedure leads to a sequence of monic orthogonal polynomials. We now show that such a sequence is unique.

Assume that $\{q_n\}_{n=0}^\infty$ is another sequence of orthogonal (monic) polynomials, i.e. the coefficient of x^n in q_n is 1, and

$$\int_a^b \omega q_i(x) q_j(x) dx = 0, \quad i \neq j.$$

Since p_n , given by (3.1.5), is also monic, we obtain that

$$\deg(p_{n+1} - q_{n+1}) \leq n.$$

Now using Lemma 3.1 gives

$$(p_{n+1}, p_{n+1} - q_{n+1})_\omega = 0, \quad (q_{n+1}, p_{n+1} - q_{n+1})_\omega = 0.$$

The above results yield $(p_{n+1} - q_{n+1}, p_{n+1} - q_{n+1})_\omega = 0$. This implies that $p_{n+1} - q_{n+1} \equiv 0$ for all $n \geq 0$. \square

The above theorem reveals a remarkable property of the orthogonal polynomials, namely, all orthogonal polynomials can be constructed with a three-term recurrence relation (3.1.5).

3.1.2. Zeros of orthogonal polynomials. The zeros of the orthogonal polynomials play an important role in the implementations of the spectral methods. The main result concerning the zeros of orthogonal polynomials is the following:

THEOREM 3.2. *The zeros of p_{n+1} are all real, simple, and lie in the interval (a, b) .*

PROOF. Since $(1, p_{n+1})_\omega = 0$, there exists at least one zero of p_{n+1} in (a, b) . Let x_0, x_1, \dots, x_k in (a, b) be the zeros of odd multiplicity of p_{n+1} ; i.e., x_0, x_1, \dots, x_k are the points where p_{n+1} changes sign. If $k = n$, we are through, since $\{x_i\}_{i=0}^n$ are the $n + 1$ simple zeros of p_{n+1} . If $k < n$, we consider the polynomial

$$q(x) = (x - x_0)(x - x_1) \cdots (x - x_k).$$

Since $\deg(q) = k + 1 < n + 1$, by orthogonality

$$(p_{n+1}, q)_\omega = 0.$$

On the other hand $p_{n+1}(x)q(x)$ cannot change sign on (a, b) since each sign change in $p_{n+1}(x)$ is canceled by a corresponding sign change in $q(x)$. It follows that

$$(p_{n+1}, q)_\omega \neq 0,$$

which is a contradiction. □

??? Give a procedure to compute zeroes (from an eigenvalue problem)
???

3.1.3. Gauss type quadratures. We wish to create quadrature formulae of the type

$$\int_a^b f(x)\omega(x)dx \approx \sum_{j=0}^N f(x_j)\omega_j.$$

If the choice of nodes x_0, x_1, \dots, x_N are made *a priori*, then in general the above formula can only be exact for polynomials of degree $\leq N$. More precisely, setting

$$\omega_j = \int_a^b h_j(x)\omega(x)dx, \quad \text{with } h_j(x) = \frac{\prod_{i \neq j}(x - x_i)}{\prod_{i \neq j}(x_j - x_i)} \quad (3.1.6)$$

being the Lagrange polynomial associated with the nodes $\{x_j\}_{j=0}^N$, we have

$$\int_a^b p(x)\omega(x)dx = \sum_{j=0}^N p(x_j)\omega_j, \quad \forall p \in P_N. \quad (3.1.7)$$

However, if we are free to choose nodes $\{x_k\}_{k=0}^N$, we can expect the quadrature formulae of the above form to be exact for polynomials of degree $\leq 2N + 1$.

We assume in this subsection that $\{p_n\}_{n=0}^\infty$ is a sequence of orthogonal polynomials, associated with a weight function ω in the interval (a, b) .

THEOREM 3.3. (Gauss quadrature) *Let x_0, x_1, \dots, x_N be the roots of p_{N+1} and we define ω_j ($j = 0, 1, \dots, N$) by (3.1.6). Then $\omega_j > 0$ for $j = 0, 1, \dots, N$ and*

$$\int_a^b p(x)\omega(x)dx = \sum_{j=0}^N p(x_j)\omega_j, \quad \text{for all } p \in P_{2N+1}. \quad (3.1.8)$$

???? Add formulae for ω_j ???

PROOF. For any $p \in P_N$, we can write $p(x) = \sum_{j=0}^N p(x_j)h_j(x)$. Hence,

$$\int_a^b p(x)\omega(x)dx = \sum_{j=0}^N p(x_j) \int_a^b h_j(x)\omega(x)dx = \sum_{j=0}^N p(x_j)\omega_j.$$

Now for any $p \in P_{2N+1}$, we can write $p = rp_{N+1} + s$ where $r, s \in P_N$. Since $P_{N+1}(x_j) = 0$, we have $p(x_j) = s(x_j)$ for $j = 0, 1, \dots, N$. Since P_{N+1} is orthogonal to r , and $s \in P_N$, we find

$$\begin{aligned} \int_a^b p(x)\omega(x)dx &= \int_a^b s(x)\omega(x)dx \\ &= \sum_{j=0}^N s(x_j)\omega_j = \sum_{j=0}^N p(x_j)\omega_j. \end{aligned}$$

It remains to prove that $\omega_j > 0$ for $j = 0, 1, \dots, N$. To this end, we take $p(x) = h_k^2(x)$ in the above relation to find

$$0 < \int_a^b h_k^2(x)\omega(x)dx = \sum_{j=0}^N h_k^2(x_j)\omega_j = \omega_k.$$

□

The Gauss quadrature is optimal in the sense that it is not possible to find $\{x_j, \omega_j\}_{j=0}^N$ such that (3.1.8) holds for all $p \in P_{2N+2}$. However, it is difficult to enforce any boundary condition since the end points a and b are not among the Gauss nodes. Therefore, we need generalized Gauss quadratures which are suitable for enforcing boundary conditions. More precisely, to enforce the Dirichlet boundary condition at both end points, we need the so called Gauss-Lobatto quadrature below, and to enforce the boundary condition at one end point, we should use the Gauss-Radau quadrature. Other generalized Gauss quadratures enforcing derivative boundary conditions can also be constructed similarly, see [9, 4].

Assuming we would like to include the left endpoint a in the quadrature. We choose $\alpha_N = -p_{N+1}(a)/p_N(a)$ and set

$$q_N(x) = \frac{P_{N+1}(x) + \alpha_N p_N(x)}{x - a}.$$

It is obvious that $q_N \in P_N$, and for any $r_{N-1} \in P_{N-1}$, we have

$$\begin{aligned} \int_a^b q_N(x)r_{N-1}(x)\omega(x)(x-a)dx = \\ \int_a^b (p_{N+1}(x) + \alpha_N p_N(x))r_{N-1}(x)\omega(x)dx = 0. \end{aligned} \quad (3.1.9)$$

Hence, $\{q_n\}$ is a set of orthogonal polynomials with weight $\omega(x)(x-a)$.

THEOREM 3.4. (Gauss-Radau quadrature) *Let x_0, x_1, \dots, x_N be the roots of $(x-a)q_N$ and ω_j ($j = 0, 1, \dots, N$) defined by (3.1.7). Then $\omega_j > 0$ for $j = 0, 1, \dots, N$ and*

$$\int_a^b p(x)\omega(x)dx = \sum_{j=0}^N p(x_j)\omega_j, \quad \text{for all } p \in P_{2N}. \quad (3.1.10)$$

PROOF. The proof is similar to that of Theorem 3.3. Obviously,

$$\int_a^b p(x)\omega(x)dx = \sum_{j=0}^N p(x_j) \int_a^b h_j(x)\omega(x)dx = \sum_{j=0}^N p(x_j)\omega_j, \quad \forall p \in P_N. \quad (3.1.11)$$

Now for any $p \in P_{2N}$, we write $p = r(x-a)q_N + s$ with $r \in P_{N-1}$, $s \in P_N$, and $p(x_j) = s(x_j)$ for $j = 0, 1, \dots, N$. Therefore, thanks to (3.1.9), we have

$$\begin{aligned} \int_a^b p(x)\omega(x)dx &= \int_a^b s(x)\omega(x)dx \\ &= \sum_{j=0}^N s(x_j)\omega_j = \sum_{j=0}^N p(x_j)\omega_j. \end{aligned}$$

Again, by taking $p(x) = h_k^2(x)$ in the above relation, we conclude that $w_k > 0$ for $k = 0, 1, \dots, N$. \square

A second Gauss-Radau quadrature can be constructed if we want to include the right endpoint b instead of the leftend point a .

We now consider the Gauss-Lobatto quadrature whose nodes include the two endpoints. We choose α_N and β_N such that

$$p_{N+1}(x) + \alpha_N p_N(x) + \beta_N p_{N-1}(x) = 0, \quad x = a, b,$$

and set

$$z_{N-1}(x) = \frac{p_{N+1}(x) + \alpha_N p_N(x) + \beta_N p_{N-1}(x)}{(x-a)(b-x)}.$$

Hence, $z_{N-1} \in P_{N-1}$ and for any $r_{N-2} \in P_{N-2}$, we have

$$\begin{aligned} \int_a^b z_{N-1}(x)r_{N-2}(x)\omega(x)(x-a)(b-x)dx = \\ \int_a^b (p_{N+1}(x) + \alpha_N p_N(x) + \beta_N p_{N-1}(x))r_{N-2}(x)\omega(x)dx = 0. \end{aligned} \quad (3.1.12)$$

Hence, $\{z_n\}$ is a set of orthogonal polynomials with weight $\omega(x)(x-a)(b-x)$. Using exactly the same procedure as in the proof of Theorem (3.4), we can prove the following:

THEOREM 3.5. (Gauss-Lobatto quadrature) *Let $\{x_i\}_{i=0}^N$ be the $N+1$ roots of $w_{N-1}(x)(x-a)(b-x)$, and ω_j ($j = 0, 1, \dots, N$) defined by (3.1.7). Then, $\omega_j = \int_a^b h_j(x)\omega(x)dx > 0$ for $j = 0, 1, \dots, N$ and*

$$\int_a^b p(x)\omega(x)dx = \sum_{j=0}^N p(x_j)\omega_j, \quad \text{for all } p \in P_{2N-1}. \quad (3.1.13)$$

3.1.4. Discrete inner products, interpolation polynomials and discrete transforms. Let $\omega(x) > 0$ be a weight function, and $\{x_j, \omega_j\}_{j=0}^N$ be a set of quadrature points (e.g. Gauss, Gauss-Radau or Gauss-Lobatto points) and associated weights. We define

$$\langle u, v \rangle_{N, \omega} := \sum_{j=0}^N u(x_j)v(x_j)\omega_j \text{ for } u, v \text{ continuous on } [a, b]. \quad (3.1.14)$$

Then, $\langle \cdot, \cdot \rangle_{N, \omega}$ is a discrete inner product in P_N , and $\| \cdot \|_{N, \omega}$ defined by

$$\|u\|_{N, \omega} = \langle u, u \rangle_{N, \omega}^{\frac{1}{2}}$$

is a norm in P_N . In particular, the Gauss, Gauss-Radau and Gauss-Lobatto quadrature formulae imply that

$$\langle u, v \rangle_{N, \omega} = (u, v)_{\omega} \text{ for } uv \in P_{2N+\delta}, \quad (3.1.15)$$

where $\delta = 1, 0$ and -1 respectively for Gauss, Gauss-Radau and Gauss-Lobatto quadrature.

Let u be a continuous function on $[-1, 1]$. The interpolation polynomial, associated with $\{x_i\}_{i=0}^N$, $I_N u$ is defined as a polynomial of degree less than or equal to N such that

$$I_N u(x_i) = u(x_i), \quad i = 0, 1, \dots, N.$$

Hence, we may write

$$I_N u = \sum_{l=0}^N \tilde{u}_l p_l(x). \quad (3.1.16)$$

Obviously, we have

$$u(x_j) = I_N u(x_j) = \sum_{l=0}^N \tilde{u}_l p_l(x_j). \quad (3.1.17)$$

Thus, $\{\tilde{u}_k\}$ are called the discrete coefficients of u and are determined by

LEMMA 3.2.

$$\tilde{u}_k = \frac{1}{\langle p_k, p_k \rangle_{N, \omega}} \sum_{j=0}^N u(x_j) p_k(x_j) \omega_j, \quad k = 0, 1, \dots, N.$$

PROOF. Thanks to (3.1.15), we derive the desired result by taking the discrete inner product of (3.1.17) with p_k . \square

3.2. Jacobi polynomials

From now on, we shall restrict our attention to a special class of orthogonal polynomials — the so called Jacobi polynomials — which are denoted by $J_n^{\alpha,\beta}(x)$ and generated from (3.1.5) with

$$\omega(x) = (1-x)^\alpha(1+x)^\beta \text{ for } \alpha, \beta > -1, (a, b) = (-1, 1),$$

and normalized by

$$J_n^{\alpha,\beta}(1) = \frac{\Gamma(n+\alpha+1)}{n!\Gamma(\alpha+1)}, \quad (3.2.1)$$

where $\Gamma(x)$ is the usual Gamma function.

In fact, we shall mainly be concerned with two special cases of Jacobi polynomials, namely the Legendre polynomials which corresponds to $\alpha = \beta = 0$ and the Chebyshev polynomials which corresponds to $\alpha = \beta = -\frac{1}{2}$. Any generic treatments of Jacobi polynomials apply in particular to both the Legendre and Chebyshev polynomials.

???Add three-term recurrence relation here !

By definition, the Jacobi polynomials which satisfy the orthogonality condition

$$\int_{-1}^1 J_n^{\alpha,\beta}(x) J_m^{\alpha,\beta}(x) (1-x)^\alpha (1+x)^\beta dx = 0 \text{ for } n \neq m. \quad (3.2.2)$$

A property of fundamental importance is the following:

THEOREM 3.6. *The Jacobi polynomials satisfy the following singular Sturm-Liouville problem:*

$$(1-x)^{-\alpha}(1+x)^{-\beta} \frac{d}{dx} \left\{ (1-x)^{\alpha+1}(1+x)^{\beta+1} \frac{d}{dx} J_n^{\alpha,\beta}(x) \right\} + n(n+1+\alpha+\beta) J_n^{\alpha,\beta}(x) = 0.$$

PROOF. We denote $\omega(x) = (1-x)^\alpha(1+x)^\beta$. By integration by parts twice, we find that for any $\phi \in P_{n-1}$,

$$\begin{aligned} & \int_{-1}^1 \frac{d}{dx} \left\{ (1-x)^{\alpha+1}(1+x)^{\beta+1} \frac{dJ_n^{\alpha,\beta}}{dx} \right\} \phi dx \\ &= - \int_{-1}^1 \omega(1-x^2) \frac{dJ_n^{\alpha,\beta}}{dx} \frac{d\phi}{dx} dx \\ &= \int_{-1}^1 J_n^{\alpha,\beta} \left\{ [-(\alpha+1)(1+x) + (\beta+1)(1-x)] \frac{d\phi}{dx} + (1-x^2) \frac{d^2\phi}{dx^2} \right\} \omega dx = 0. \end{aligned}$$

The last equality follows from the fact that

$$\int_{-1}^1 J_n^{\alpha,\beta} \psi \omega(x) dx = 0, \quad \forall \psi \in P_{n-1}.$$

An immediate consequence of the above relation is that there exists λ such that

$$-\frac{d}{dx} \left\{ (1-x)^{\alpha+1} (1+x)^{\beta+1} \frac{d}{dx} J_n^{\alpha,\beta}(x) \right\} = \lambda J_n^{\alpha,\beta}(x) \omega(x).$$

To determine λ , we take the coefficients of the leading term $x^{n+\alpha+\beta}$ in the above relation. Assuming $J_n^{\alpha,\beta}(x) = k_n x^n + \{\text{lower order terms}\}$, we get

$$k_n n(n+1+\alpha+\beta) = k_n \lambda$$

which implies that $\lambda = n(n+1+\alpha+\beta)$. \square

One derives immediately from Theorem 3.6 and (3.2.2) the following result:

COROLLARY 3.1.

$$\int_{-1}^1 (1-x)^{\alpha+1} (1+x)^{\beta+1} \frac{dJ_n^{\alpha,\beta}}{dx} \frac{dJ_m^{\alpha,\beta}}{dx} dx = 0 \quad \text{for } n \neq m. \quad (3.2.3)$$

The above relation indicates that $\frac{d}{dx} J_n^{\alpha,\beta}$ forms a sequence of orthogonal polynomials with weight $\omega(x) = (1-x)^{\alpha+1} (1+x)^{\beta+1}$. Hence, by the uniqueness, we find that $\frac{d}{dx} J_n^{\alpha,\beta}$ is proportional to $J_{n-1}^{\alpha+1,\beta+1}$.

THEOREM 3.7. (*Rodrigues' formula*)

$$(1-x)^\alpha (1+x)^\beta J_n^{\alpha,\beta}(x) = \frac{(-1)^n}{2^n n!} \frac{d^n}{dx^n} [(1-x)^{n+\alpha} (1+x)^{n+\beta}]. \quad (3.2.4)$$

PROOF. For any $\phi \in P_{N-1}$, we have

$$\begin{aligned} \int_{-1}^1 \frac{d^n}{dx^n} \left((1-x)^{n+\alpha} (1+x)^{n+\beta} \right) \phi dx \\ = (-1)^n \int_{-1}^1 \left((1-x)^{n+\alpha} (1+x)^{n+\beta} \right) \frac{d^n \phi}{dx^n} dx = 0. \end{aligned}$$

Hence, there is a constant α_n such that

$$\frac{d^n}{dx^n} \left((1-x)^{n+\alpha} (1+x)^{n+\beta} \right) = \alpha_n (1-x)^\alpha (1+x)^\beta J_n^{\alpha,\beta}(x). \quad (3.2.5)$$

On the other hand, it is easy to see that

$$\begin{aligned} \alpha_n &= \frac{1}{J_n^{\alpha,\beta}(1)} \left\{ \frac{1}{(1-x)^\alpha (1+x)^\beta} \frac{d^n}{dx^n} \left((1-x)^{n+\alpha} (1+x)^{n+\beta} \right) \right\} \Big|_{x=1} \\ &= (-1)^n n! 2^n. \end{aligned}$$

The proof is complete. \square

REMARK 3.2.1. When $\alpha = \beta > -1$, the corresponding Jacobi polynomials are called Gegenbauer polynomials or ultraspherical polynomials. In this case, one derives from the Rodrigues' formula that $J_n^{\alpha,\alpha}$ is an odd function for n odd and an even function for n even.

??? Add a Corollary for the formula k_n ???

3.3. Legendre polynomials

The Legendre polynomials, denoted by $L_k(x)$, are the orthogonal polynomials with $\omega(x) = 1$ and $(a, b) = (-1, 1)$. The three-term recurrence relation for the Legendre polynomials reads

$$\begin{aligned} L_0(x) &= 1, & L_1(x) &= x, \\ (n+1)L_{n+1}(x) &= (2n+1)xL_n(x) - nL_{n-1}(x), & n &\geq 1. \end{aligned} \quad (3.3.1)$$

On the other hand, the Legendre polynomials are Jacobi polynomials with $\alpha = \beta = 0$. Hence, it satisfies the following singular Sturm-Liouville problem

$$\left((1-x^2)L'_n(x)\right)' + n(n+1)L_n(x) = 0, \quad x \in (-1, 1), \quad (3.3.2)$$

and

$$\int_{-1}^1 L_k(x)L_j(x)dx = \frac{2}{2k+1}\delta_{kj}. \quad (3.3.3)$$

We infer from the above two relations that

$$\int_{-1}^1 L'_k(x)L'_j(x)(1-x^2)dx = \frac{2k(k+1)}{2k+1}\delta_{kj}, \quad (3.3.4)$$

i.e. the polynomial sequence $\{L'_k(x)\}$ are mutually orthogonal with respect to the weight $\omega(x) = 1 - x^2$.

An important property of the Legendre polynomials is the following:

LEMMA 3.3.

$$\int_{-1}^x L_n(\xi)d\xi = \frac{1}{2n+1}(L_{n+1}(x) - L_{n-1}(x)), \quad n \geq 1. \quad (3.3.5)$$

PROOF. Let $S_{n+1}(x) = \int_{-1}^x L_n(t)dt$. Hence, $S_{n+1} \in P_{n+1}$ and $S_{n+1}(\pm 1) = 0$. Therefore, for any $m < n - 1$,

$$\begin{aligned} \int_{-1}^1 S_{n+1}L_m dx &= \int_{-1}^1 S_{n+1}S'_{m+1} dx \\ &= - \int_{-1}^1 S'_{n+1}S_{m+1} dx = \int_{-1}^1 L_n S_{m+1} dx = 0. \end{aligned} \quad (3.3.6)$$

Hence, we can write

$$S_{n+1} = a_{n-1}L_{n-1} + a_n L_n + a_{n+1}L_{n+1}.$$

By parity argument, $a_n = 0$. On the other hand, by writing $L_n = k_n x^n + k_{n-1}x^{n-1} + \dots$, we find from the definition of S_{n+1} that $\frac{k_n}{n+1} = a_{n+1}k_{n+1}$. We then derive from the formula of k_n (??) that $a_{n+1} = \frac{1}{2n+1}$. Finally, we derive from $S_{n+1}(-1) = 0$ that $a_{n-1} = -a_{n+1} = -\frac{1}{2n+1}$. \square

We derive from above a recursive relation for computing the derivatives of the Legendre polynomials:

$$L_n(x) = \frac{1}{2n+1}(L'_{n+1}(x) - L'_{n-1}(x)), \quad n \geq 1. \quad (3.3.7)$$

We can derive from the above formula that

$$L'_n(x) = \sum_{\substack{k=0 \\ k+n \text{ odd}}}^{n-1} (2k+1)L_k(x), \quad (3.3.8a)$$

$$L''_n(x) = \sum_{\substack{k=0 \\ k+n \text{ even}}}^{n-2} \left(k + \frac{1}{2}\right) (n(n+1) - k(k+1)) L_k(x). \quad (3.3.8b)$$

We also derive from (??) and (3.3.2) that

$$L_n(\pm 1) = (\pm 1)^n, \quad (3.3.9a)$$

$$L'_n(\pm 1) = \frac{1}{2}(\pm 1)^{n-1}n(n+1), \quad (3.3.9b)$$

$$L''_n(\pm 1) = (\pm 1)^n(n-1)n(n+1)(n+2)/8. \quad (3.3.9c)$$

For the Legendre series, The Gauss quadrature points and weights can be derived from Theorems 3.3 and 3.5 (cf. [9]).

LEMMA 3.4. *For the Legendre-Gauss quadrature: $\{x_j\}_{j=0}^N$ are the roots of $L_{N+1}(x)$, and*

$$\omega_j = \frac{2}{(1-x_j^2)[L'_{N+1}(x_j)]^2}, \quad 0 \leq j \leq N.$$

For the Legendre-Gauss-Lobatto quadrature: $\{x_j\}_{j=0}^N$ are the roots of $(1-x^2)L'_N(x)$, and

$$\omega_j = \frac{2}{N(N+1)[L_N(x_j)]^2}, \quad 0 \leq j \leq N.$$

PROOF. To be added ?? (p. 49, Furano) □

3.3.1. Zeros of Legendre polynomials. It is seen from the last subsection that the quadrature points for the Legendre polynomials are related to the zeros of the Legendre polynomials. Below we will discuss how to find the zero points of $L_N^{(m)}(x)$ numerically, where $m < N$ is the order of derivative.

We start from left boundary -1 and try to find the small interval of width H which contains the first zero point z_1 . The idea for locating the interval is similar to that used by the *bisection method*. In the resulting (small) interval, we use *Newton's method* to find the first zero point. The Newton's method for finding the root of $f(x)$ is

$$x_{k+1} = x_k - f(x_k)/f'(x_k). \quad (3.3.10)$$

After finding the first zero point, we use the point $z_1 + H$ as the starting point and repeat the previous procedure to get the second zero point z_2 . This will give us all the zero points of $L_N^{(m)}(x)$. The parameter H , which is related to the smallest gap of the zero points, will be chosen as N^{-2} .

The following pseudocode is to find the zeros of $L_N^{(m)}(x)$.

```

CODE LGauss.1
%This code is to find the roots of  $L_N^{(m)}(x)$ .
Input N,  $\varepsilon$ , m ( $\varepsilon$  gives the accuracy of the roots)
H=N-2
a=-1
for k=1 to N-m do
%search a small interval containing a zero point of  $L_N^{(m)}(x)$ 
  b=a+H
  while  $L_N^{(m)}(a) \cdot L_N^{(m)}(b) > 0$ 
    a=b; b=a+H
  endwhile
%the Using Newton's method in (a,b) to find the root
  root=(a+b)/2; right=b
  while  $|\text{root}-\text{right}| \geq \varepsilon$ 
    right=root
    root=root- $L_N^{(m)}(\text{root})/L_N^{(m+1)}(\text{root})$ 
  endwhile
  z(k)=root
  a=root+H
endfor
Output z(1), z(2), ..., z(N-m)

```

(This code needs to be changed to a more efficient one !!!)

In the above pseudocode, the parameter ε is used to control the accuracy of the roots. Also, we need to use the recurrence formulae (3.3.1) and (3.3.7) to compute $L_n^{(m)}(x)$ which are used in the above code.

```

CODE LGauss.2
%This code is to evaluate  $L_n^{(m)}(x)$ .
Input n, m, x
For j=0 to m do
  if j=0 then
    s(0,j)=1; s(1,j)=x
    s(k+1,j)=((2k+1)*x*s(k,j)-k*s(k-1,j))/(k+1)  1≤k≤n-1
  else
    s(0,j)=0
    if j=1 then
      s(1,j)=1
    else
      s(1,j)=0
    endif
    s(k+1,j)=(2k+1)*s(k,j-1)+s(k-1,j)  1≤k≤n-1
  endif
endFor

```

$$L_n^{(m)}(x) = \mathbf{s}(n, m)$$

As an example, by setting $N = 7, m = 0$ and $\varepsilon = 10^{-8}$ in `LGauss.1` we obtain the roots for $L_7(x)$:

z_1	-0.94910791	z_5	0.40584515
z_2	-0.74153119	z_6	0.74153119
z_3	-0.40584515	z_7	0.94910791
z_4	0.00000000		

By setting $N = 6, m = 1$ and $\varepsilon = 10^{-8}$ in `LGauss.1` we obtain the roots for $L_6'(x)$. Together with $Z_1 = -1$ and $Z_7 = 1$ they form the Legendre-Gauss-Lobatto points:

Z_1	-1.00000000	Z_5	0.46884879
Z_2	-0.83022390	Z_6	0.83022390
Z_3	-0.46884879	Z_7	1.00000000
Z_4	0.00000000		

3.3.2. Interpolation and discrete Legendre transform. Here, we give some detailed results on the interpolation operator I_N based on the Legendre-Gauss-Lobatto points $\{x_i\}_{i=0}^N$. For any function u which is continuous on $[-1, 1]$, we have

$$u(x_j) = I_N u(x_j) = \sum_{k=0}^N \tilde{u}_k L_k(x_j). \quad (3.3.11)$$

In this case, the lemma 3.2 reads:

LEMMA 3.5.

$$\tilde{u}_k = \gamma_k \sum_{j=0}^N u(x_j) \frac{L_k(x_j)}{L_N^2(x_j)}, \quad k = 0, 1, \dots, N, \quad (3.3.12)$$

where $\gamma_k = \frac{2k+1}{N(N+1)}$ for $k = 0, 1, \dots, N-1$ and $\gamma_N = \frac{1}{N+1}$.

PROOF. Taking the discrete inner product (i.e. (3.1.14) with $\omega = 1$) of (3.3.11) with L_n , we get

$$\tilde{u}_n \langle L_n, L_n \rangle_N = \sum_{j=0}^N u(x_j) L_n(x_j) \frac{2}{N(N+1)[L_N(x_j)]^2}.$$

Then, the desired result follows from

$$\langle L_n, L_n \rangle_N = (L_n, L_n) = \frac{2}{2n+1}, \quad 0 \leq n \leq N-1,$$

and

$$\langle L_N, L_N \rangle_N = \sum_{j=0}^N L_N^2(x_j) \frac{2}{N(N+1)} \frac{1}{[L_N(x_j)]^2} = \frac{2}{N}. \quad (3.3.13)$$

□

Next, we show that the discrete norm based on the Legendre-Gauss-Lobatto quadrature is equivalent to the usual L^2 norm in P_N .

LEMMA 3.6. *Let $\|\cdot\|_N$ be the discrete norm relative to the Legendre-Gauss-Lobatto quadrature. Then*

$$\|u\|_{L^2} \leq \|u\|_N \leq \sqrt{3}\|u\|_{L^2}, \text{ for all } u \in P_N.$$

PROOF. Let $u = \sum_{k=0}^N \tilde{u}_k L_k$, we have

$$\|u\|_{L^2}^2 = \sum_{k=0}^N \tilde{u}_k^2 \frac{2}{2k+1}.$$

On the other hand,

$$\|u\|_N^2 = \sum_{k=0}^{N-1} \tilde{u}_k^2 \frac{2}{2k+1} + \tilde{u}_N^2 \langle L_N, L_N \rangle_N.$$

We then conclude from (3.3.13) and the fact that $\frac{2}{2N+1} \leq \frac{2}{N} \leq 3\frac{2}{2N+1}$. \square

REMARK 3.3.1. Assuming that $\{L_k(x_j)\}_{k,j=0,1,\dots,N}$ have been precomputed (see the code LGAUSS.2), then, the discrete Legendre transforms (3.3.11) and (3.3.12) can be carried out using a standard matrix-vector multiplication routine in about N^2 flops. The cost of the discrete Legendre transforms can be cut by half by using the identity $L_k(x_j) = (-1)^k L_k(x_{N-j})$ which is a consequence of the odd-even parity of the Legendre polynomials.

Although the discrete Legendre transforms do not have optimal computational complexity, but they can be accelerated significantly by using a high performance/parallel matrix multiplication routine.

A similar interpolation operator and discrete Legendre transform can also be defined on the Legendre-Gauss points.

In implementating the spectral methods, one often needs to evaluate derivatives or form derivative matrices. The derivatives can be evaluated either in the frequency space or in the physical space. We emphasize that in both cases, the derivatives are computed *exactly* rather than *approximately* in the finite difference or finite element methods.

3.3.3. Differentiation in frequency space. Given $u = \sum_{k=0}^N \tilde{u}_k L_k \in P_N$, we can write

$$u' = \sum_{k=1}^N \tilde{u}_k L'_k = \sum_{k=0}^N \tilde{u}_k^{(1)} L_k \text{ with } \tilde{u}_N^{(1)} = 0.$$

Thanks to (3.3.7), we find

$$\begin{aligned} u' &= \sum_{k=0}^N \tilde{u}_k^{(1)} L_k = \tilde{u}_0^{(1)} + \sum_{k=1}^{N-1} \tilde{u}_k^{(1)} \frac{1}{2k+1} (L'_{k+1} - L'_{k-1}) \\ &= \frac{\tilde{u}_{N-1}^{(1)}}{2N-1} L'_N + \sum_{k=1}^{N-2} \left\{ \frac{\tilde{u}_{k-1}^{(1)}}{2k-1} - \frac{\tilde{u}_{k+1}^{(1)}}{2k+3} \right\} L'_k. \end{aligned}$$

Comparing the coefficients of L'_k , we find that the coefficients $\{\tilde{u}_k^{(1)}\}$ of u' are determined by the recursive relation:

$$\begin{aligned} \tilde{u}_N^{(1)} &= 0, \quad \tilde{u}_{N-1}^{(1)} = (2N-1)\tilde{u}_N, \\ \tilde{u}_{k-1}^{(1)} &= \left(\tilde{u}_k + \frac{\tilde{u}_{k+1}^{(1)}}{2k+3} \right) (2k-1), \quad k = N-1, N-2, \dots, 1. \end{aligned} \quad (3.3.14)$$

Higher derivatives can be obtained by repeatedly applying the above formula.

3.3.4. Derivative matrices in physical space. Given $u \in P_N$ and its values at a set of collocation points $\{x_j\}_{j=0}^N$. Let $\{h_j(x)\}_{j=0}^N$ be the Lagrange polynomials relative to $\{x_j\}_{j=0}^N$. Then,

$$u^{(m)}(x) = \sum_{j=0}^N u^{(m-1)}(x_j) h'_j(x), \quad m \geq 1. \quad (3.3.15)$$

Setting $d_{kj} = h'_j(x_k)$, and $D = (d_{kj})_{k,j=0,1,\dots,N}$, and

$$\bar{u}^{(m)} = (u^{(m)}(x_0), u^{(m)}(x_1), \dots, u^{(m)}(x_N))^T, \quad m = 0, 1, 2, \dots,$$

we can rewrite (3.3.15) as

$$u^{(m)}(x_k) = \sum_{j=0}^N d_{kj} u^{(m-1)}(x_j) \quad \text{or} \quad \bar{u}^{(m)} = D \bar{u}^{(m-1)} \quad (3.3.16)$$

which implies that

$$\bar{u}^{(m)} = D^m \bar{u}^{(0)}. \quad (3.3.17)$$

Hence, the derivatives of u in the physical space are totally determined by the matrix D .

The above discussion is valid for any set of collocation points. In the implementation of spectral methods, one is interested mostly in using the Gauss-Lobatto points or Gauss points. Below, we provide an explicit expression for the derivative matrix in case $\{x_j\}_{j=0}^N$ are the Legendre-Gauss-Lobatto points.

LEMMA 3.7. *Let $\{x_j\}_{j=0}^N$ be the zeros of $(1-x^2)L'_N(x)$. Then,*

$$h_j(x) = -\frac{1}{N(N+1)L_N(x_j)} \frac{(1-x^2)L'_N(x)}{x-x_j}, \quad j = 0, 1, \dots, N, \quad (3.3.18)$$

and

$$d_{kj} = \begin{cases} \frac{L_N(x_k)}{L_N(x_j)} \frac{1}{x_k - x_j} & k \neq j \\ \frac{N(N+1)}{4} & k = j = 0 \\ -\frac{N(N+1)}{4} & k = j = N \\ 0 & 1 \leq k = j \leq N-1 \end{cases}. \quad (3.3.19)$$

PROOF. Since $\{x_j\}$ are zeros of $(1-x^2)L'_N(x)$, (3.3.18) is a easy consequence of (3.3.2). We then derive from (3.3.18) that for all $x \neq x_j$,

$$h'_j(x) = \frac{L_N(x)}{L_N(x_j)(x-x_j)} + \frac{1}{N(N+1)L_N(x_j)} \frac{(1-x^2)L'_N(x)}{(x-x_j)^2}, \quad (3.3.20)$$

which implies the statement (3.3.19) for $k \neq j$.

For $k = j$, applying the L'Hospital's rule to (3.3.20) (twice to the second-term), we get

$$h'_j(x_j) = \frac{L'_N(x_j)}{L_N(x_j)} - \frac{L'_N(x_j)}{2L_N(x_j)} = \frac{L'_N(x_j)}{2L_N(x_j)}.$$

We conclude from (3.3.9) and the fact that $L'_N(x_j) = 0$ for $1 \leq j \leq N-1$. \square

REMARK 3.3.2. The derivative matrix D is a full matrix, so $O(N^2)$ flops are needed to compute $\{u'(x_j)\}_{j=0}^N$ from $\{u(x_j)\}_{j=0}^N$ by using the derivative matrix.

Since $u^{(N+1)}(x) \equiv 0$ for any $u \in P_N$, we have $D^{N+1}\bar{u}^{(0)} = 0$ for any $\bar{u}^{(0)} \in \mathbf{R}^{N+1}$. Hence, the only eigenvalue of D is zero which has a multiplicity of order $N+1$.

3.4. Chebyshev polynomials

3.4.1. Chebyshev polynomials. The Chebyshev polynomials, denoted by $T_k(x)$, are the sequence of orthogonal polynomials with $\omega(x) = (1-x^2)^{-\frac{1}{2}}$ and $(a, b) = (-1, 1)$. The three-term recurrence for the Chebyshev polynomials is:

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x), & n &\geq 1. \end{aligned} \quad (3.4.1)$$

On the other hand, the Chebyshev polynomials are the Jacobi polynomials with $\alpha = \beta = -\frac{1}{2}$. Hence, it satisfies the following singular Sturm-Liouville problem

$$\sqrt{1-x^2} \left(\sqrt{1-x^2} T'_k(x) \right)' + k^2 T_k(x) = 0, \quad x \in (-1, 1), \quad (3.4.2)$$

and the orthogonality relation

$$\int_{-1}^1 T_k(x)T_j(x) \frac{1}{\sqrt{1-x^2}} dx = \frac{c_k \pi}{2} \delta_{kj}, \quad (3.4.3)$$

where $c_0 = 2$ and $c_k = 1$ for $k \geq 1$.

We infer from the above two equations that

$$\int_{-1}^1 T'_k(x)T'_j(x) \sqrt{1-x^2} dx = \frac{k^2 c_k \pi}{2} \delta_{kj}, \quad (3.4.4)$$

i.e., the polynomials $\{T'_k(x)\}$ are mutually orthogonal with respect to the weight function $w(x) = \sqrt{1-x^2}$. From the fact that $\cos(n \cos^{-1} x)$ is a polynomial of degree n and the trigonometric relation

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2 \cos \theta \cos(n\theta),$$

We find that $\cos(n \cos^{-1} x)$ satisfies also the three term recurrence relation (3.4.1). Hence,

$$T_n(x) = \cos(n \cos^{-1} x), \quad n = 0, 1, \dots. \quad (3.4.5)$$

This explicit representation allows us to derive easily many useful properties of the Chebyshev polynomials. In fact, let $\theta = \cos^{-1} x$, it follows from (3.4.5) that

$$2T_n(x) = \frac{1}{n+1} T'_{n+1}(x) - \frac{1}{n-1} T'_{n-1}(x), \quad n \geq 2, \quad (3.4.6)$$

It can be easily shown by using (3.4.5) that

$$|T_n(x)| \leq 1, \quad |T'_n(x)| \leq n^2, \quad (3.4.7a)$$

$$2T_m(x)T_n(x) = T_{m+n}(x) + T_{m-n}(x), \quad m \geq n \geq 0. \quad (3.4.7b)$$

One can also derive from (3.4.2) that

$$T'_n(\pm 1) = (\pm 1)^{n-1} n^2, \quad (3.4.8a)$$

$$T''_N(\pm 1) = \frac{1}{3} (\pm 1)^N N^2 (N^2 - 1). \quad (3.4.8b)$$

Moreover, we can derive from (3.4.6) that

$$T'_n(x) = 2n \sum_{\substack{k=0 \\ k+n \text{ odd}}}^{n-1} \frac{1}{c_k} T_k(x), \quad (3.4.9a)$$

$$T''_n(x) = \sum_{\substack{k=0 \\ k+n \text{ even}}}^{n-2} \frac{1}{c_k} n(n^2 - k^2) T_k(x). \quad (3.4.9b)$$

For the Chebyshev series, one can determine from Theorems 3.3 and 3.5 the quadrature points and weights (cf. [19]).

LEMMA 3.8. *For Chebyshev-Gauss quadrature:*

$$x_j = \cos \frac{(2j+1)\pi}{2N+2}, \quad \omega_j = \frac{\pi}{N+1}, \quad 0 \leq j \leq N.$$

For Chebyshev-Gauss-Lobatto quadrature:

$$x_j = \cos \frac{\pi j}{N}, \quad \omega_j = \frac{\pi}{\tilde{c}_j N}, \quad 0 \leq j \leq N.$$

where $\tilde{c}_0 = \tilde{c}_N = 2$ and $\tilde{c}_j = 1$ for $j = 1, 2, \dots, N-1$.

Note that for the simplicity of the notation, the Chebyshev-Gauss and Chebyshev-Gauss-Lobatto points are arranged in descending order, namely,

$$x_N < x_{N-1} < \dots < x_1 < x_0.$$

We shall keep this convention in the Chebyshev case unless otherwise specified.

3.4.2. Interpolation and discrete Chebyshev transform. As in the Legendre case, we present here some detailed results on the interpolation operator I_N based on the Chebyshev-Gauss-Lobatto points $\{x_i\}_{i=0}^N$. For any function u which is continuous on $[-1, 1]$, we have

$$u(x_j) = I_N u(x_j) = \sum_{k=0}^N \tilde{u}_k T_k(x_j). \quad (3.4.10)$$

In this case, the lemma 3.2 reads:

LEMMA 3.9.

$$\tilde{u}_k = \frac{2}{\tilde{c}_k N} \sum_{j=0}^N \frac{1}{\tilde{c}_j} u(x_j) \cos \frac{kj\pi}{N}. \quad (3.4.11)$$

PROOF. Taking the discrete inner product (i.e. (3.1.14) with $\omega = (1 - x^2)^{-\frac{1}{2}}$) of (3.4.10) with T_n , we get

$$\tilde{u}_n \langle T_n, T_n \rangle_{N, \omega} = \sum_{j=0}^N u(x_j) \cos \frac{nj\pi}{N} \frac{\pi}{\tilde{c}_j N}.$$

The desired result follows from the above and

$$\langle T_n, T_n \rangle_{N, \omega} = (T_n, T_n)_\omega = \frac{\tilde{c}_n \pi}{2}, \quad 0 \leq n \leq N-1,$$

and

$$\langle T_N, T_N \rangle_{N, \omega} = \sum_{j=0}^N \cos^2 j\pi \frac{\pi}{\tilde{c}_j N} = \pi. \quad (3.4.12)$$

□

As in the Legendre case, the discrete norm based on the Chebyshev-Gauss-Lobatto quadrature is equivalent to the usual L_ω^2 norm in P_N . In fact, we have

LEMMA 3.10. *Let $\|\cdot\|_{N,\omega}$ be the discret norm relative to the Chebyshev-Gauss-Lobatto quadrature. Then*

$$\|u\|_{L_\omega^2} \leq \|u\|_{N,\omega} \leq \sqrt{2}\|u\|_{L_\omega^2}, \text{ for all } u \in P_N.$$

PROOF. For $u = \sum_{k=0}^N \tilde{u}_k T_k$, we have

$$\|u\|_{L_\omega^2}^2 = \sum_{k=0}^N \tilde{u}_k^2 \frac{c_k \pi}{2}.$$

On the other hand,

$$\|u\|_{N,\omega}^2 = \sum_{k=0}^{N-1} \tilde{u}_k^2 \frac{c_k \pi}{2} + \tilde{u}_N^2 \langle T_N, T_N \rangle_{N,\omega}.$$

We conclude from the above and (3.4.12). \square

The most important practical feature of the Chebyshev series is that the discrete Chebyshev transforms (3.4.10) and (3.4.11) can be performed in $O(N \log_2 N)$ operations thanks to the Fast Fourier Transform (FFT). In fact, the discrete Chebyshev transforms (3.4.10) and (3.4.11) can be computed most easily by calling the subroutine **cost.f** in the **FFTPACK** (available from www.netlib.org), they can also be computed by using other FFT programs in various platforms. For readers' convenience, a Fortran subroutine using **cost.f** is listed below:

```

subroutine chtrfm1(n,x,cosn,iflag)
*****
*** Purpose: 1-D discrete Chebyshev transform
*** Input: cosn: output from calling costi(n+1,cosn)
***          x: Chebyshev coefficients if iflag=1
***             values at CGL points   if iflag=0
*** Output:  x: values at CGL points   if iflag=1
***             Chebyshev coefficients if iflag=0
*****
dimension x(0:n),cosn(3*n+15)
if (iflag.eq.1) then
  val=.5
  do i=1,n-1
    x(i)=val*x(i)
  enddo
  call cost(n+1,x,cosn)
else
  call cost(n+1,x,cosn)
  val=1./n
  do i=1,n-1
    x(i)=x(i)*val
  enddo

```

```

x(0)=x(0)*val*.5
x(n)=x(n)*val*.5
endif
return
end

```

3.4.3. Differentiation in frequency space. Given $u = \sum_{k=0}^N \tilde{u}_k T_k \in P_N$, we derive from (3.4.6) that

$$\begin{aligned}
u' &= \sum_{k=1}^N \tilde{u}_k T'_k = \sum_{k=0}^N \tilde{u}_k^{(1)} T_k \\
&= \tilde{u}_0^{(1)} + \tilde{u}_1^{(1)} x + \sum_{k=2}^{N-1} \tilde{u}_k^{(1)} \left(\frac{T'_{k+1}}{2(k+1)} - \frac{T'_{k-1}}{2(k-1)} \right) \\
&= \frac{\tilde{u}_{N-1}^{(1)}}{2N} T'_N + \sum_{k=1}^{N-2} \frac{1}{2k} (c_{k-1} \tilde{u}_{k-1}^{(1)} - \tilde{u}_{k+1}^{(1)}) T'_k,
\end{aligned} \tag{3.4.13}$$

where $c_0 = 2$ and $c_k = 1$ for $k \geq 1$. Comparing the coefficients of T'_k , we find that the Chebyshev coefficients $\{\tilde{u}_k^{(1)}\}$ of u' are determined by the recursive relation:

$$\begin{aligned}
\tilde{u}_N^{(1)} &= 0, \quad \tilde{u}_{N-1}^{(1)} = 2N\tilde{u}_N, \\
\tilde{u}_{k-1}^{(1)} &= (2k\tilde{u}_k + \tilde{u}_{k+1}^{(1)})/c_{k-1}, \quad k = N-1, N-2, \dots, 1.
\end{aligned} \tag{3.4.14}$$

Higher derivatives can be obtained by repeatedly applying the above formula.

3.4.4. Derivative matrices in physical space. To compute the derivative matrix in the physical space, we can use the same notations as in the Legendre case except that now we choose $\{x_j = \cos \frac{j\pi}{N}\}$ to be the Chebyshev-Gauss-Lobatto points.

LEMMA 3.11. *The Lagrange polynomials associated to the Chebyshev-Gauss-Lobatto points are*

$$h_j(x) = \frac{(-1)^j (x^2 - 1) T'_N(x)}{\tilde{c}_j N^2 (x - x_j)}, \quad 0 \leq j \leq N. \tag{3.4.15}$$

The derivative matrix $(d_{kj} = h'_j(x_k))$ is given by

$$\begin{aligned}
d_{kj} &= \frac{\tilde{c}_k}{\tilde{c}_j} \frac{(-1)^{k+j}}{x_k - x_j}, \quad j \neq k, \\
d_{kk} &= -\frac{x_k}{2(1 - x_k^2)}, \quad k = 1, 2, \dots, N-1, \\
d_{00} &= -d_{NN} = (2N^2 + 1)/6,
\end{aligned} \tag{3.4.16}$$

where $\tilde{c}_k = 1$ for $1 \leq k \leq N-1$ and $\tilde{c}_0 = \tilde{c}_N = 2$.

PROOF. Let us denote $\omega_N(x) = (x^2 - 1)T'_N(x)$. Then, the Lagrange polynomials associated with $\{x_j\}_{j=0}^N$ can be written as

$$h_j(x) = \frac{\omega_N(x)}{\omega'_N(x_j)(x - x_j)}, \quad 0 \leq j \leq N.$$

Consequently,

$$h'_j(x) = \frac{1}{\omega'_N(x_j)} \frac{(x - x_j)\omega'_N(x) - \omega_N(x)}{(x - x_j)^2}. \quad (3.4.17)$$

We derive from (3.4.2) that

$$\begin{aligned} \omega'_N(x) &= -(\sqrt{1-x^2}\sqrt{1-x^2}T'_N(x))' \\ &= xT'_N(x) - \sqrt{1-x^2}(\sqrt{1-x^2}T'_N(x))' \\ &= xT'_N(x) + N^2T_N(x). \end{aligned} \quad (3.4.18)$$

We then derive from (3.4.8a) that $\omega'_N(x_j) = (-1)^j \tilde{c}_j N^2$. This above result, together with (3.4.17) and (3.4.8a), lead to

$$h'_j(x_k) = \frac{\tilde{c}_k (-1)^{k+j}}{\tilde{c}_j x_k - x_j}, \quad 0 \leq k \neq j \leq N.$$

For $k = j$, we apply the L'Hospital's rule on (3.4.17) to get

$$h'_j(x_j) = \frac{w''_N(x_j)}{2\omega'_N(x_j)}. \quad (3.4.19)$$

Thanks to (3.4.18), we have

$$\omega''_N(x_j) = (N^2 + 1)T'_N(x_j) + xT''_N(x_j), \quad j = 0, 1, \dots, N. \quad (3.4.20)$$

We derive from (3.4.2) that

$$T''_N(x_j) = (-1)^{j+1} N^2 \frac{1}{1-x_j^2}, \quad 1 \leq j \leq N-1. \quad (3.4.21)$$

The desired results follow from the above and (3.4.8b). \square

REMARK 3.4.1. The remark 3.3.2 applies also to the Chebyshev case. However, in the Chebyshev case, a more efficient alternative for computing derivatives is to proceed in the frequency space as described earlier in this section:

- (1) Compute the discrete Chebyshev coefficients $\{\tilde{u}_k\}$ of u from $u(x_j) = \sum_{k=0}^N \tilde{u}_k T_k(x_j)$;
- (2) Compute the discrete Chebyshev coefficients $\{\tilde{u}_k^{(1)}\}$ of u' using (3.4.14);
- (3) Compute $u'(x_j)$ from $u'(x_j) = \sum_{k=0}^N \tilde{u}_k^{(1)} T_k(x_j)$.

The cost of this approach is only $O(N \log N)$ thanks to the Fast Fourier Transform.

A pseudocode for computing D using Lemma 3.11 is given below.

```

CODE ChyDM.1
Input N
%collocation points, and  $\tilde{c}_k$ 
x(j)=cos( $\pi j/N$ )   $0 \leq j \leq N$ 
 $\tilde{c}(0)=2$ ,  $\tilde{c}(N)=2$ 
 $\tilde{c}(j)=1$    $1 \leq j \leq N-1$ 
% first order differentiation matrix
for k=0 to N do
  for j=0 to N-k do
    if k=0 and j=0
      D1(k,j)=( $2N^2+1$ )/6
    elseif k=N and j=N
      D1(k,j)=-D1(0,0)
    elseif k=j
      D1(k,j)=-x(k)/(2*(1-x(k)^2))
    else
      D1(k,j)= $\tilde{c}(k)*(-1)^{j+k}/(\tilde{c}(j)*(x(k)-x(j)))$ 
    endif
  endfor
endfor
for k=1 to N do
  for j=N-k+1 to N do
    D1(k,j)=-D1(N-k,N-j)
  endfor
endfor

```

It has been observed that for large N the direct implementation of the above formulas suffers from cancellation, causing errors in the elements of the matrix D . Thus, it is advisable to replace the first two formulas using trigonometric identities by the formulas

$$d_{kj} = \frac{\tilde{c}_k (-1)^{k+j}}{\tilde{c}_j} \left(\sin \frac{(j+k)\pi}{2N} \sin \frac{(j-k)\pi}{2N} \right)^{-1}, \quad k \neq j, \quad (3.4.22)$$

$$d_{kk} = -\frac{x_k}{2 \sin^2(k\pi/N)}, \quad k \neq 0, N.$$

Finally, to avoid computing the $\sin x$ for $|x| > \frac{\pi}{2}$, we take advantage of the symmetry property

$$d_{N-k, N-j} = -d_{k,j}. \quad (3.4.23)$$

Thus, an accurate method to compute D is using formulas (3.4.22) to find the *upper left triangle* of D (i.e., compute d_{kj} with $k+j \leq N$), and then use the relation (3.4.23) and (3.4.22) for the other elements.

A pseudocode for computing D using (3.4.23) and (3.4.22) is given below.

```

CODE ChyDM.2

```

```

Input N
%collocation points, and  $\tilde{c}_k$ 
x(j)=cos( $\pi j/N$ )    0 $\leq j \leq N$ 
 $\tilde{c}(0)=2, \tilde{c}(N)=2$ 
 $\tilde{c}(j)=1$     1 $\leq j \leq N-1$ 
% first order differentiation matrix
for k=0 to N do
  for j=0 to N-k do
    if k=0 and j=0
      D1(k,j)=(2N2+1)/6
    elseif k=N and j=N
      D1(k,j)=-D1(0,0)
    elseif k=j
      D1(k,j)=-x(k)/(2sin2(k $\pi/N$ ))
    else
      D1(k,j)= $\tilde{c}(k)*(-1)^{j+k}/(2*\tilde{c}(j)*\sin((j+k)\pi/2N)*\sin((j-k)\pi/2N))$ 
    endif
  endfor
endfor
for k=1 to N do
  for j=N-k+1 to N do
    D1(k,j)=-D1(N-k,N-j)
  endfor
endfor

```

3.5. Laguerre and Hermite polynomials

To be added !

3.6. Error estimates of the polynomial projections

3.6.1. Orthogonal projection error in $L_\omega^2(I)$. We derive first a fundamental result on the projection error of the Jacobi series.

We shall restrict our discussion to $-1 < \alpha, \beta < 1$. For $\omega(x) = (1-x)^\alpha(1+x)^\beta$, we define an operator A associated to the Jacobi polynomials by

$$A\phi := -(1-x)^{-\alpha}(1+x)^{-\beta} \frac{d}{dx} \left\{ (1-x)^{\alpha+1}(1+x)^{\beta+1} \frac{d}{dx} \phi \right\}. \quad (3.6.1)$$

Thanks to Theorem 3.6, we have

$$AJ_n^{\alpha,\beta} = n(n+1+\alpha+\beta)J_n^{\alpha,\beta}. \quad (3.6.2)$$

Furthermore, we derive easily by integration by parts that A is self adjoint with respect to the inner product $(\cdot, \cdot)_\omega$, i.e.,

$$(A\phi, \psi)_\omega = (\phi, A\psi)_\omega \quad \text{for all } \phi, \psi \in D(A) = \{u : u, Au \in L_\omega^2(I)\}. \quad (3.6.3)$$

On the other hand, we derive by the definition (3.6.1) that

$$A\phi = [(\alpha + 1)(x + 1) + (\beta + 1)(1 - x)]\phi' + (1 - x^2)\phi''. \quad (3.6.4)$$

Therefore, for $-1 < \alpha, \beta < 1$, we have

$$\|A\phi\|_{L_\omega^2} \lesssim \|\phi\|_{H_\omega^2} \quad \text{for all } \phi \in H_\omega^2(I). \quad (3.6.5)$$

DEFINITION 3.6.1. The orthogonal projector $\pi_{N,\omega} : L_\omega^2(I) \rightarrow P_N$ is defined by

$$(u - \pi_{N,\omega}u, v)_\omega = 0 \quad \text{for } v \in P_N. \quad (3.6.6)$$

Since any $u \in L_\omega^2(I)$ can be expanded as a Jacobi series

$$u = \sum_{j=0}^{+\infty} \tilde{u}_j J_j^{\alpha,\beta} \quad \text{with } \tilde{u}_n = \frac{(u, J_n^{\alpha,\beta})_\omega}{\|J_n^{\alpha,\beta}\|_{L_\omega^2}^2}, \quad (3.6.7)$$

we have by definition $\pi_{N,\omega}u = \sum_{n=0}^N \tilde{u}_n J_n^{\alpha,\beta}$ and

$$\|u - \pi_{N,\omega}u\|_{L_\omega^2}^2 = \sum_{n=N+1}^{+\infty} \tilde{u}_n^2 \|J_n^{\alpha,\beta}\|_{L_\omega^2}^2 = \sum_{n=N+1}^{+\infty} \frac{(u, J_n^{\alpha,\beta})_\omega^2}{\|J_n^{\alpha,\beta}\|_{L_\omega^2}^2}. \quad (3.6.8)$$

THEOREM 3.8.

$$\|u - \pi_{N,\omega}u\|_{L_\omega^2} \lesssim N^{-m} \|u\|_{H_\omega^m} \quad \text{for all } u \in H_\omega^m(I).$$

PROOF. Let us denote $\lambda_n^{\alpha,\beta} = n(n + 1 + \alpha + \beta)$. We consider first the case $m = 2r$.

Applying repeatedly (3.6.2) and (3.6.3), we obtain

$$\begin{aligned} (u, J_n^{\alpha,\beta})_\omega &= \frac{1}{\lambda_n^{\alpha,\beta}} (u, AJ_n^{\alpha,\beta})_\omega = \frac{1}{\lambda_n^{\alpha,\beta}} (Au, J_n^{\alpha,\beta})_\omega \\ &= \frac{1}{(\lambda_n^{\alpha,\beta})^r} (A^r u, J_n^{\alpha,\beta})_\omega. \end{aligned}$$

Hence, we derive from (3.6.8) that

$$\begin{aligned} \|u - \pi_{N,\omega}u\|_{L_\omega^2}^2 &= \sum_{n=N+1}^{+\infty} \frac{(u, J_n^{\alpha,\beta})_\omega^2}{\|J_n^{\alpha,\beta}\|_{L_\omega^2}^2} \\ &= \sum_{n=N+1}^{+\infty} \frac{1}{(\lambda_n^{\alpha,\beta})^{4r}} \frac{(A^r u, J_n^{\alpha,\beta})_\omega^2}{\|J_n^{\alpha,\beta}\|_{L_\omega^2}^2} \\ &\lesssim \frac{1}{N^{4r}} \sum_{n=N+1}^{+\infty} \frac{(A^r u, J_n^{\alpha,\beta})_\omega^2}{\|J_n^{\alpha,\beta}\|_{L_\omega^2}^2} \\ &\leq \frac{1}{N^{4r}} \sum_{n=0}^{+\infty} \frac{(A^r u, J_n^{\alpha,\beta})_\omega^2}{\|J_n^{\alpha,\beta}\|_{L_\omega^2}^2} = \frac{1}{N^{2m}} \|A^r u\|_{L_\omega^2}^2. \end{aligned}$$

We then derive from (3.6.5) that

$$\|u - \pi_{N,\omega}u\|_{L_\omega^2} \lesssim N^{-m} \|u\|_{H_\omega^m} \quad \text{for all } u \in H_\omega^m(I).$$

We now consider the case $m = 2r + 1$. Exactly as above, we have

$$(u, J_n^{\alpha, \beta})_\omega = \frac{1}{(\lambda_n^{\alpha, \beta})^r} (A^r u, J_n^{\alpha, \beta})_\omega.$$

We use (3.6.1) once more and integrate by parts to get

$$(u, J_n^{\alpha, \beta})_\omega = \frac{1}{(\lambda_n^{\alpha, \beta})^{r+1}} \int_{-1}^1 (A^r u)' (J_n^{\alpha, \beta})' \hat{\omega} dx,$$

where we have denoted $\hat{\omega} = (1-x)^{1+\alpha}(1+x)^{1+\beta}$. Therefore,

$$\begin{aligned} \|u - \pi_{N, \omega} u\|_{L_\omega^2}^2 &= \sum_{n=N+1}^{+\infty} \frac{1}{(\lambda_n^{\alpha, \beta})^{2r+2} \|J_n^{\alpha, \beta}\|_{L_\omega^2}^2} \left(\int_{-1}^1 (A^r u)' (J_n^{\alpha, \beta})' \hat{\omega} dx \right)^2 \\ &\lesssim \frac{1}{N^{4r+2}} \sum_{n=N+1}^{+\infty} \frac{1}{\lambda_n^{\alpha, \beta} \|J_n^{\alpha, \beta}\|_{L_\omega^2}^2} \left(\int_{-1}^1 (A^r u)' (J_n^{\alpha, \beta})' \hat{\omega} dx \right)^2. \end{aligned} \quad (3.6.9)$$

We recall that $\{(J_n^{\alpha, \beta})'\}$ form a set of orthogonal polynomials in L_ω^2 , and

$$\|(J_n^{\alpha, \beta})'\|_{L_\omega^2}^2 = \lambda_n^{\alpha, \beta} \|J_n^{\alpha, \beta}\|_{L_\omega^2}^2. \quad (3.6.10)$$

Thus, for any $v = \sum_{n=0}^{+\infty} \tilde{v}_n J_n^{\alpha, \beta} \in L_\omega^2$, we have the expansion

$$v' = \sum_{n=1}^{+\infty} \tilde{v}_n (J_n^{\alpha, \beta})' \quad \text{with} \quad \tilde{v}_n = \frac{1}{\|(J_n^{\alpha, \beta})'\|_{L_\omega^2}^2} \int_{-1}^1 v' (J_n^{\alpha, \beta})' \hat{\omega} dx,$$

and thanks to (3.6.10),

$$\begin{aligned} \int_{-1}^1 (v')^2 \hat{\omega} dx &= \sum_{n=1}^{+\infty} \tilde{v}_n^2 \|(J_n^{\alpha, \beta})'\|_{L_\omega^2}^2 \\ &= \sum_{n=1}^{+\infty} \frac{1}{\|(J_n^{\alpha, \beta})'\|_{L_\omega^2}^2} \left(\int_{-1}^1 v' (J_n^{\alpha, \beta})' \hat{\omega} dx \right)^2 \\ &\geq \sum_{n=N+1}^{+\infty} \frac{1}{\lambda_n^{\alpha, \beta} \|J_n^{\alpha, \beta}\|_{L_\omega^2}^2} \left(\int_{-1}^1 v' (J_n^{\alpha, \beta})' \hat{\omega} dx \right)^2 \end{aligned}$$

Now, we take $v = A^r u$ in the above relation, thanks to (3.6.9) and the fact that $\hat{\omega} = \omega(1-x^2) \leq \omega$, we get

$$\begin{aligned} \|u - \pi_{N, \omega} u\|_{L_\omega^2}^2 &\lesssim \frac{1}{N^{4r+2}} \int_{-1}^1 ((A^r u)')^2 \hat{\omega} dx \leq \frac{1}{N^{4r+2}} \int_{-1}^1 ((A^r u)')^2 \omega dx \\ &= \frac{1}{N^{2m}} |A^r u|_{H_\omega^1}^2 \lesssim \frac{1}{N^{2m}} \|u\|_{H_\omega^m}^2. \end{aligned}$$

□

3.6.2. Orthogonal projection error in $H_{0,\omega}^1(I)$. Now, we consider an orthogonal projector in $H_{0,\omega}^1$. We denote

$$P_N^0 = \{u \in P_N : u(\pm 1) = 0\}.$$

DEFINITION 3.6.2. The orthogonal projector $\pi_{N,\omega}^{1,0}$ from $H_{0,\omega}^1(I)$ to P_N^0 is defined by

$$((u - \pi_{N,\omega}^{1,0}u)', v')_\omega = 0 \text{ for } v \in P_N^0. \quad (3.6.11)$$

THEOREM 3.9.

$$|u - \pi_{N,\omega}^{1,0}u|_{H_\omega^1} \lesssim N^{1-m} \|u\|_{H_\omega^m} \text{ for all } u \in H_\omega^m(I) \cap H_{0,\omega}^1(I).$$

PROOF. For any $u \in H_\omega^m(I) \cap H_{0,\omega}^1(I)$, we set

$$u_N = \int_{-1}^x \left\{ \pi_{N-1,\omega} u' - \frac{1}{2} \int_{-1}^1 \pi_{N-1,\omega} u' d\eta \right\} d\xi. \quad (3.6.12)$$

Therefore,

$$u_N \in P_N^0 \text{ and } u'_N = \pi_{N-1,\omega} u' - \frac{1}{2} \int_{-1}^1 \pi_{N-1,\omega} u' d\eta.$$

Hence,

$$\|u' - u'_N\|_{L_\omega^2} \leq \|u' - \pi_{N-1,\omega} u'\|_{L_\omega^2} + \left| \frac{1}{2} \int_{-1}^1 \pi_{N-1,\omega} u' d\eta \right|. \quad (3.6.13)$$

On the other hand, since $u(\pm 1) = 0$, we derive by Cauchy-Schwarz inequality that

$$\begin{aligned} \left| \int_{-1}^1 \pi_{N-1,\omega} u' dx \right| &= \left| \int_{-1}^1 (\pi_{N-1,\omega} u' - u') dx \right| \\ &\leq \left(\int_{-1}^1 w^{-1} dx \right)^{\frac{1}{2}} \|\pi_{N-1,\omega} u' - u'\|_{L_\omega^2} \\ &\lesssim \|\pi_{N-1,\omega} u' - u'\|_{L_\omega^2} \text{ (for } \alpha, \beta < 1). \end{aligned} \quad (3.6.14)$$

We then conclude from (3.6.13)–(3.6.14) and Theorem 3.8 that

$$\begin{aligned} |u - \pi_{N,\omega}^{1,0}u|_{H_\omega^1} &= \inf_{\phi_N \in P_N^0} |u - \phi_N|_{H_\omega^1} \leq \|u - u_N\|_{L_\omega^2} \\ &\lesssim \|u' - \pi_{N-1,\omega} u'\|_{L_\omega^2} \lesssim (N-1)^{1-m} \|u'\|_{H_\omega^{m-1}} \\ &\lesssim N^{1-m} \|u\|_{H_\omega^m}. \end{aligned}$$

□

REMARK 3.6.1. One can also prove that

$$\|u - \pi_{N,\omega}^{1,0}u\|_{L_\omega^2} \lesssim N^{-m} \|u\|_{H_\omega^m} \text{ for all } u \in H_\omega^m(I) \cap H_{0,\omega}^1(I).$$

However, the proof of the second result is very technical so we refer to [3] (Thm. 4.2) for details.

Another frequently used orthogonal projector in $H_{0,\omega}^1(I)$ is considered in Section 4.5.2.

3.6.3. Interpolation error. We present below an optimal error estimate for the interpolation polynomials based on the Gauss-Lobatto points. We refer to Section 21 in [5] for the proof.

THEOREM 3.10. *Let $\{x_j\}_{j=0}^N$ be the roots of $(1-x^2)J_N^{\alpha,\alpha}(x)$ with $-1 < \alpha < 1$ and $\omega = (1-x^2)^\alpha$. Let $I_N : C[-1,1] \rightarrow P_N$ be the interpolation operator with respect to $\{x_j\}_{j=0}^N$. Then,*

$$\|u - I_N u\|_{H_\omega^1} \lesssim N^{1-m} \|u\|_{H_\omega^m}, \quad \text{for all } u \in H_\omega^m(I), \quad m \geq 1;$$

$$\|u - I_N u\|_{L_\omega^2} \lesssim N^{-m} \|u\|_{H_\omega^m}, \quad \text{for all } u \in H_\omega^m(I), \quad m \geq 1.$$

PROOF. to be added !!!

□

The above results indicate that error estimates for the interpolation polynomial based on the Gauss-Lobatto points are optimal in both the H_ω^1 and L_ω^2 norms. One should note that an interpolation polynomial based on uniformly spaced points is usually a very poor approximation unless the function is periodic in the concerned interval.

3.6.4. Inverse inequalities.

Spectral Methods for Two-Point Boundary Value Problems

We consider in this chapter spectral algorithms for solving the two points boundary value problem:

$$-\varepsilon U'' + p(x)U' + q(x)U = F, \quad x \in I = (-1, 1), \quad (4.0.1)$$

with the general boundary condition

$$a_-U(-1) + b_-U'(-1) = c_-, \quad a_+U(1) + b_+U'(1) = c_+, \quad (4.0.2)$$

which includes in particular the Dirichlet ($a_{\pm} = 1$ and $b_{\pm} = 0$), the Neumann ($a_{\pm} = 0$ and $b_{\pm} = 1$) boundary conditions, and the mixed boundary conditions ($a_- = b_+ = 0$ or $a_+ = b_- = 0$). Whenever possible, we will give a uniform treatment for all these boundary conditions. Without loss of generality, we assume $a_{\pm} \geq 0$. We will also assume

$$\begin{aligned} \text{(i)} \quad & a_-^2 + b_-^2 \neq 0, \quad \text{and} \quad a_-b_- \leq 0; \quad a_+^2 + b_+^2 \neq 0, \quad a_+b_+ \geq 0; \\ \text{(ii)} \quad & q(x) - \frac{1}{2}p'(x) \geq 0, \quad x \in (I); \\ \text{(iii)} \quad & p(1) > 0 \text{ if } b_+ \neq 0, \quad p(-1) < 0 \text{ if } b_- \neq 0. \end{aligned} \quad (4.0.3)$$

The above conditions are necessary for the well-posedness of (4.0.1)–(4.0.2).

In the first section, we present several *Galerkin* methods which are based on variational formulations for (4.0.1-4.0.2) using *continuous* inner products. In the second section, we present the *collocation methods in the strong form* which looks for approximate solutions to satisfy (4.0.2) and (4.0.1) *exactly* at a set of collocation points. In the third section, we consider the *collocation methods in the weak form* which are based on variational formulations for (4.0.1-4.0.2) using *discrete* inner products. In, Section 4, we present some preconditioned iterative methods for solving the linear systems arising from the spectral approximations of two-point boundary value problems. In section 5, we provide an error analysis for two model cases.

4.1. Galerkin methods

Let us first reduce the problem (4.0.1-4.0.2) to a problem with homogeneous boundary conditions:

- Case 1. $a_{\pm} = 0$ and $b_{\pm} \neq 0$:

We set $\tilde{u} = \beta x^2 + \gamma x$, where β and γ are uniquely determined by

asking \tilde{u} satisfies (4.0.2), namely

$$\begin{aligned} -2b_-\beta + b_-\gamma &= c_-, \\ 2b_+\beta + b_+\gamma &= c_+. \end{aligned} \quad (4.1.1)$$

- Case 2. $a_-^2 + a_+^2 \neq 0$:

We set $\tilde{u} = \beta x + \gamma$, where β and γ can again be uniquely determined by asking \tilde{u} satisfies (4.0.2). Indeed, we have

$$\begin{aligned} (-a_- + b_-)\beta + a_-\gamma &= c_-, \\ (a_+ + b_+)\beta + a_+\gamma &= c_+, \end{aligned} \quad (4.1.2)$$

whose determinant is

$$\text{DET} = -a_-a_+ + b_-a_+ - a_-a_+ - b_+a_-.$$

Thus, (4.0.3) implies that $b_- \leq 0$ and $b_+ \geq 0$ which imply that $\text{DET} < 0$.

Now, we set

$$u = U - \tilde{u}, \quad f = F - (-\varepsilon \tilde{u}'' + p(x)\tilde{u}' + q(x)\tilde{u}).$$

Then u satisfies the following equation

$$-\varepsilon u'' + p(x)u' + q(x)u = f, \text{ in } I = (-1, 1), \quad (4.1.3)$$

with the homogeneous boundary conditions

$$a_-u(-1) + b_-u'(-1) = 0, \quad a_+u(1) + b_+u'(1) = 0. \quad (4.1.4)$$

In this section, we will only consider a special case of (4.1.3), namely,

$$-u'' + \alpha u = f, \text{ in } I = (-1, 1), \quad (4.1.5)$$

where $\alpha > 0$ is a given constant. The general case (4.0.1)–(4.0.2) will be treated later by using a collocation approach.

Let us denote

$$H_\star^1(I) = \{u \in H^1(I) : u(\pm 1) = 0 \text{ if } b_\pm = 0\}, \quad (4.1.6)$$

and

$$h_- = \begin{cases} 0 & \text{if } a_-b_- = 0 \\ \frac{a_-}{b_-} & \text{if } a_-b_- \neq 0 \end{cases}, \quad h_+ = \begin{cases} 0 & \text{if } a_+b_+ = 0 \\ \frac{a_+}{b_+} & \text{if } a_+b_+ \neq 0 \end{cases}. \quad (4.1.7)$$

A standard variation formulation for (4.1.5)–(4.1.4) is:

Find $u \in H_\star^1(I)$ such that

$$\begin{aligned} a(u, v) &:= (u', v') + h_+u(1)v(1) - h_-u(-1)v(-1) \\ &+ \alpha(u, v) = (f, v) \text{ for all } v \in H_\star^1(I). \end{aligned} \quad (4.1.8)$$

It is easy to see that the bilinear form $a(\cdot, \cdot)$ defined above is continuous and coercive in $H_\star^1(I) \times H_\star^1(I)$ under the condition (4.0.3). One derive immediately by Lax-Milgram lemma that the problem (4.1.8) admits a unique solution. Note that only the Dirichlet boundary condition(s) is enforced *exactly* in $H_\star^1(I)$, all other boundary conditions are treated *naturally*.

4.1.1. Weighted Galerkin formulation. We consider the approximation of (4.1.5)–(4.1.4) by using a weighted Galerkin method in the polynomial space $\tilde{X}_N = H_{x,\omega}^1(I) \cap P_N$. A straightforward extension of (4.1.8) using the weighted inner product leads to the following formulation: Find $u_N \in \tilde{X}_N$ such that

$$\begin{aligned} & (u'_N, \omega^{-1}(v_N \omega)')_\omega + \omega(1)h_+ u_N(1)v_N(1) - \omega(-1)h_- u_N(-1)v_N(-1) \\ & + \alpha(u_N, v_N)_\omega = (f, v_N)_\omega \quad \text{for all } v \in \tilde{X}_N. \end{aligned} \quad (4.1.9)$$

However, there are several problems associated with this formulation. First, the above formulation does not make sense if $\lim_{x \rightarrow \pm 1} \omega(x)$ does not exist, except in the case of Dirichlet boundary conditions. Hence it can not be used for Jacoby weight with $\alpha < 0$ or $\beta < 0$, including in particular the Chebyshev weight (cf. [7] and [10] p. 194-196 for some special weighted variational formulations for (4.1.5)–(4.1.4)). Second, as it will become clear later in this section, even in the case $\omega(x) \equiv 1$, this formulation will not lead to sparse or special linear system that can be efficiently inverted. The cure is to use a new weighted variation formulation in which the general boundary condition (4.1.4) is enforced *exactly* rather than *approximately* in (4.1.9).

Let us denote

$$X_N = \{v \in P_N : a_\pm v(\pm 1) + b_\pm v'(\pm 1) = 0\}. \quad (4.1.10)$$

the new weighted Galerkin method for (4.1.5)–(4.1.4) is to look for $u_N \in X_N$ such that

$$-(u''_N, v_N)_\omega + \alpha(u_N, v_N)_\omega = (f_N, v_N)_\omega \quad \forall v_N \in X_N, \quad (4.1.11)$$

where f_N is an appropriate polynomial approximation of f . The main difference with (4.1.9) is that the Robin boundary condition is enforced *exactly* here. We will see below that by choosing appropriate basis functions for X_N , we will be able to reduce (4.1.11) to a linear system with sparse or special matrix that can be solved efficiently.

Given a set of basis functions $\{\phi_k\}_{k=0,1,\dots,N-2}$ for X_N , we denote

$$\begin{aligned} f_k &= \int_I f_N \phi_k \omega dx, \quad \bar{f} = (f_0, f_1, \dots, f_{N-2})^T; \\ u_N(x) &= \sum_{n=0}^{N-2} \hat{u}_n \phi_n(x), \quad \bar{u} = (\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-2})^T, \\ s_{kj} &= - \int_I \phi_j'' \phi_k \omega dx, \quad m_{kj} = \int_I \phi_j \phi_k \omega dx, \end{aligned} \quad (4.1.12)$$

and

$$S = (s_{kj})_{0 \leq k, j \leq N-2}, \quad M = (m_{kj})_{0 \leq k, j \leq N-2}.$$

By setting $u_N(x) = \sum_{n=0}^{N-2} \hat{u}_n \phi_n(x)$ and $v_N(x) = \phi_j(x)$ ($j = 0, 1, \dots, N-2$) in (4.1.11), we find that the equation (4.1.11) is equivalent to the following

linear system:

$$(S + \alpha M)\bar{u} = \bar{f}. \quad (4.1.13)$$

Below, we will determine S and M in two special cases, namely $\omega(x) \equiv 1$ and $\omega(x) = (1 - x^2)^{-\frac{1}{2}}$.

4.1.2. Legendre-Galerkin method. We set $\omega(x) \equiv 1$ and $f_N = I_N f$ which is the Legendre interpolation polynomial of f relative to the Legendre-Gauss-Lobatto points. Then (4.1.11) becomes

$$-\int_I u_N'' v_N dx + \alpha \int_I u_N v_N dx = \int_I I_N f v_N dx, \quad \forall v_N \in X_N, \quad (4.1.14)$$

which we refer as Legendre-Galerkin method for (4.1.3-4.1.4).

The actual linear system for (4.1.14) will depend on the basis functions of X_N . Just as in the finite-element methods, neighboring points are used to form basis functions so as to minimize their interactions in the physical space, neighboring orthogonal polynomials should be used to form basis functions in a spectral-Galerkin method so as to minimize their interactions in the frequency space. Therefore, we look for basis functions in the following form:

$$\phi_k(x) = L_k(x) + a_k L_{k+1}(x) + b_k L_{k+2}(x). \quad (4.1.15)$$

LEMMA 4.1. *For all $k \geq 0$, there exist unique $\{a_k, b_k\}$ such that $\phi_k(x) = L_k(x) + a_k L_{k+1}(x) + b_k L_{k+2}(x)$ verifies the boundary condition (4.1.4).*

PROOF. Since $L_k(\pm 1) = (\pm 1)^k$ and $L_k'(\pm 1) = \frac{1}{2}(\pm 1)^{k-1}k(k+1)$, the boundary condition (4.1.4) leads to the following system for $\{a_k, b_k\}$:

$$\begin{aligned} (a_+ + \frac{b_+}{2}(k+1)(k+2))a_k + (a_+ + \frac{b_+}{2}(k+2)(k+3))b_k \\ = -a_+ - \frac{b_+}{2}k(k+1), \\ -(a_- - \frac{b_-}{2}(k+1)(k+2))a_k + (a_- - \frac{b_-}{2}(k+2)(k+3))b_k \\ = -a_- + \frac{b_-}{2}k(k+1). \end{aligned} \quad (4.1.16)$$

The determinant of the above system is

$$\begin{aligned} \text{DET}_k = & 2a_+a_- + a_-b_+(k+2)^2 - a_+b_-(k+2)^2 \\ & - \frac{1}{2}b_-b_+(k+1)(k+2)^2(k+3). \end{aligned}$$

We then derive from (4.0.3) that the four terms (including the signs before them) of DET_k are all positive for any k . Hence, $\{a_k, b_k\}$ can be uniquely

determined from (4.1.16), namely

$$\begin{aligned} a_k &= - \left\{ \left(a_+ + \frac{b_+}{2}(k+2)(k+3) \right) \left(-a_- + \frac{b_-}{2}k(k+1) \right) \right. \\ &\quad \left. - \left(a_- - \frac{b_-}{2}(k+2)(k+3) \right) \left(-a_+ - \frac{b_+}{2}k(k+1) \right) \right\} / \text{DET}_k, \\ b_k &= \left\{ \left(a_+ + \frac{b_+}{2}(k+1)(k+2) \right) \left(-a_- + \frac{b_-}{2}k(k+1) \right) \right. \\ &\quad \left. + \left(a_- - \frac{b_-}{2}(k+1)(k+2) \right) \left(-a_+ - \frac{b_+}{2}k(k+1) \right) \right\} / \text{DET}_k. \end{aligned} \quad (4.1.17)$$

□

Note that in particular,

- if $a_{\pm} = 1$ and $b_{\pm} = 0$ (Dirichlet boundary conditions), we have $a_k = 0$ and $b_k = -1$;
- if $a_{\pm} = 0$ and $b_{\pm} = 1$ (Neumann boundary conditions), we have $a_k = 0$ and $b_k = -k(k+1)/((k+2)(k+3))$.

It is obvious that $\{\phi_k(x)\}$ are linearly independent. Therefore, by dimension argument we have

$$X_N = \text{span}\{\phi_k(x) : k = 0, 1, 2, \dots, N-2\}.$$

REMARK 4.1.1. In the very special case

$$-u_{xx} = f, \quad u_x(\pm 1) = 0,$$

with the condition $\int_{-1}^1 f dx = 0$, since the solution is only determined up to a constant, we should use

$$X_N = \text{span}\{\phi_k(x) : k = 1, 2, \dots, N-2\}.$$

This remark applies also to the Chebyshev-Galerkin method presented below.

LEMMA 4.2. *The stiffness matrix S is a diagonal matrix with*

$$s_{kk} = -(4k+6)b_k, \quad k = 0, 1, 2, \dots \quad (4.1.18)$$

The mass matrix M is symmetric penta-diagonal whose nonzero elements are

$$m_{jk} = m_{kj} = \begin{cases} \frac{2}{2k+1} + a_k^2 \frac{2}{2k+3} + b_k^2 \frac{2}{2k+5}, & j = k, \\ a_k \frac{2}{2k+3} + a_{k+1} b_k \frac{2}{2k+5}, & j = k+1, \\ b_k \frac{2}{2k+5}, & j = k+2. \end{cases} \quad (4.1.19)$$

PROOF. By integration by parts and taking into account the boundary condition (4.1.4), we find that

$$\begin{aligned} s_{jk} &= - \int_I \phi_k''(x) \phi_j(x) dx \\ &= \int_I \phi_k'(x) \phi_j'(x) dx + \frac{a_+}{b_+} \phi_k(1) \phi_j(1) - \frac{a_-}{b_-} \phi_k(-1) \phi_j(-1) \quad (4.1.20) \\ &= - \int_I \phi_k(x) \phi_j''(x) dx = s_{kj}, \end{aligned}$$

where $\frac{a_+}{b_+}$ (resp. $\frac{a_-}{b_-}$) should be replaced by zero when $b_+ = 0$ (resp. $b_- = 0$). It is then obvious from (4.1.20) and the definition of $\{\phi_k(x)\}$ that S is a diagonal matrix. Thanks to (3.3.8b) and (3.3.3), we find

$$\begin{aligned} s_{kk} &= -b_k \int_I L_{k+2}''(x) L_k(x) dx \\ &= -b_k \left(k + \frac{1}{2}\right) (4k + 6) \int_I L_k^2 dx = -b_k (4k + 6). \end{aligned}$$

The nonzero entries for M can be easily obtained using (3.3.3). \square

REMARK 4.1.2. An immediate consequence is that $\{\phi_k\}_{k=0}^{N-2}$ form an orthogonal basis in X_N with respect to the inner product $-(u_N'', v_N)$. Furthermore, an *orthonormal* basis of X_N is $\{\tilde{\phi}_k := \frac{1}{-b_k(4k+6)} \phi_k\}_{k=0}^{N-2}$.

In summary: given the values of f at LGL points $\{x_i\}_{0 \leq i \leq N}$, we determine the values of u_N (solution of (4.1.11)) at these LGL points as follows:

- (1) (pre-computation) Compute LGL points, $\{a_k, b_k\}$ and nonzero elements of S and M ;
- (2) Evaluate the Legendre coefficients of $I_N f(x)$ from $\{f(x_i)\}_{i=0}^N$ (backward Legendre transform) and evaluate \bar{f} ;
- (3) Solve \bar{u} from (4.1.13);
- (4) Evaluate $u_N(x_j) = \sum_{i=0}^{N-2} \hat{u}_i \phi_i(x_j)$, $j = 0, 1, \dots, N$ ("modified" forward Legendre transform).

A pseudocode outlining the above solution procedure is provided below:

CODE SG.1

Input N , collocation points x_k and $f(x_k)$ for $k = 0, 1, \dots, N$

Compute a_k, b_k, s_{kk}, m_{kj}

%Backward Legendre transform

for k=0 to N-1 do

$$g_k = \frac{2k+1}{N(N+1)} \sum_{j=0}^N f(x_j) \frac{L_k(x_j)}{L_N(x_j)^2}$$

end

$$g_N = \frac{1}{N+1} \sum_{j=0}^N f(x_j) \frac{1}{L_N(x_j)}$$

%Evaluate \bar{f} from $f_k = (\sum_{j=0}^N g_j L_j(x), \phi_k(x))$

for k=0 to N-2 do

$$f_k = g_k / (k + \frac{1}{2}) + a_k g_{k+1} / (k + \frac{3}{2}) + b_k g_{k+2} / (k + \frac{5}{2})$$

```

end
Solve  $(\alpha S + M)u = f$ 
%Evaluate  $g_k$  from  $\sum_{j=0}^{N-2} \hat{u}_j \phi_j(x) = \sum_{j=0}^N g_j L_j(x)$ 
 $g_0 = \hat{u}_0$ 
 $g_1 = \hat{u}_1 + a_0 \hat{u}_0$ 
for k=2 to N-2 do
     $g_k = \hat{u}_k + a_{k-1} \hat{u}_{k-1} + b_{k-2} \hat{u}_{k-2}$ 
end
 $g_{N-1} = a_{N-2} \hat{u}_{N-2} + b_{N-3} \hat{u}_{N-3}$ 
 $g_N = b_{N-2} \hat{u}_{N-2}$ 
%forward Legendre transform
for k=0 to N do
     $\hat{u}_k = \sum_{j=0}^N g_j L_j(x_k)$ 
end
Output  $\hat{u}_0, \hat{u}_1, \dots, \hat{u}_N$ 

```

Although the solution of the linear system (4.1.13) can be done in $O(N)$ flops, but the two discrete Legendre transforms in the above procedure cost about $2N^2$ flops. To reduce the cost of the discrete transforms between physical and spectral spaces, a natural choice is to use Chebyshev polynomials so that the discrete Chebyshev transforms can be accelerated by using FFT.

4.1.3. Chebyshev-Galerkin method. We set $\omega(x) = (1-x^2)^{-\frac{1}{2}}$ and $f_N = I_N f$ which is the Chebyshev interpolation polynomial of f relative to the Chebyshev-Gauss-Lobatto points. Then (4.1.11) becomes

$$\begin{aligned}
 & - \int_I u_N'' v_N \omega dx + \alpha \int_I u_N v_N \omega(x) dx \\
 & = \int_I I_N f v_N \omega(x) dx, \quad \forall v_N \in X_N,
 \end{aligned} \tag{4.1.21}$$

which we refer as Chebyshev-Galerkin method for (4.1.3-4.1.4).

As before, we would like to seek the basis functions of X_N in the form

$$\phi_k(x) = T_k(x) + a_k T_{k+1}(x) + b_k T_{k+2}(x). \tag{4.1.22}$$

LEMMA 4.3. *For all $k \geq 0$, there exist unique $\{a_k, b_k\}$ such that*

$$\phi_k(x) = T_k(x) + a_k T_{k+1}(x) + b_k T_{k+2}(x)$$

satisfies the boundary condition (4.1.4).

PROOF. Since $T_k(\pm 1) = (\pm 1)^k$ and $T_k'(\pm 1) = (\pm 1)^{k-1} k^2$, we find from (4.1.4) that $\{a_k, b_k\}$ must satisfy the system

$$\begin{aligned}
 (a_+ + b_+(k+1)^2)a_k + (a_+ + b_+(k+2)^2)b_k &= -a_+ - b_+ k^2, \\
 -(a_- - b_-(k+1)^2)a_k + (a_- - b_-(k+2)^2)b_k &= -a_- + b_- k^2,
 \end{aligned} \tag{4.1.23}$$

whose determinant is

$$\text{DET}_k = 2a_+a_- + (k+1)^2(k+2)^2(a_-b_+ - a_+b_- - 2b_-b_+).$$

As in the Legendre case, the condition (4.0.3) implies that $\text{DET}_k \neq 0$. Hence, $\{a_k, b_k\}$ are uniquely determined by

$$\begin{aligned} a_k &= - \left\{ (a_+ + b_+(k+2)^2)(-a_- + b_-k^2) \right. \\ &\quad \left. - (a_- - b_-(k+2)^2)(-a_+ - b_+k^2) \right\} / \text{DET}_k, \\ b_k &= \left\{ (a_+ + b_+(k+1)^2)(-a_- + b_-k^2) \right. \\ &\quad \left. + (a_- - b_-(k+1)^2)(-a_+ - b_+k^2) \right\} / \text{DET}_k. \end{aligned} \quad (4.1.24)$$

□

Therefore, we have by dimension argument that

$$X_N = \text{span}\{\phi_k(x) : k = 0, 1, 2, \dots, N-2\}.$$

One easily derives from (3.4.3) that the mass matrix M is a symmetric positive definite penta-diagonal matrix whose nonzero elements are

$$m_{jk} = m_{kj} = \begin{cases} c_k \frac{\pi}{2} (1 + a_k^2 + b_k^2) & j = k, \\ \frac{\pi}{2} (a_k + a_{k+1} b_k) & j = k + 1, \\ \frac{\pi}{2} b_k & j = k + 2, \end{cases} \quad (4.1.25)$$

where $c_0 = 2$ and $c_k = 1$ for $k \geq 1$. However, the computation of s_{kj} is much more involved. Below, we shall derive explicit expression of s_{kj} for two special cases.

LEMMA 4.4. *For the case $a_{\pm} = 1$ and $b_{\pm} = 0$ (Dirichlet boundary conditions), we have $a_k = 0$, $b_k = -1$ and*

$$s_{kj} = \begin{cases} 2\pi(k+1)(k+2), & j = k \\ 4\pi(k+1), & j = k+2, k+4, k+6, \dots \\ 0, & j < k \text{ or } j+k \text{ odd} \end{cases} \quad (4.1.26)$$

For the case $a_{\pm} = 0$ and $b_{\pm} = 1$ (Neumann boundary conditions), we have $a_k = 0$, $b_k = -\frac{k^2}{(k+2)^2}$ and

$$s_{kj} = \begin{cases} 2\pi(k+1)k^2/(k+2), & j = k \\ 4\pi j^2(k+1)/(k+2)^2, & j = k+2, k+4, k+6, \dots \\ 0, & j < k \text{ or } j+k \text{ odd} \end{cases} \quad (4.1.27)$$

PROOF. One observes immediately that

$$s_{kj} = - \int_I \phi_j'' \phi_k \omega dx = 0 \text{ for } j < k.$$

Hence, S is an upper triangular matrix. By the odd-even parity of the Chebyshev polynomials, we have also $s_{kj} = 0$ for $j+k$ odd.

Thanks to (3.4.9b), we have

$$\begin{aligned} T''_{k+2}(x) &= \frac{1}{c_k}(k+2)((k+2)^2 - k^2)T_k(x) \\ &\quad + \frac{1}{c_{k-2}}(k+2)((k+2)^2 - (k-2)^2)T_{k-2}(x) + \cdots \end{aligned} \quad (4.1.28)$$

We consider first the case $a_{\pm} = 1$ and $b_{\pm} = 0$. From (4.1.17), we find $\phi_k(x) = T_k(x) - T_{k+2}(x)$. It follows immediately from (4.1.28) and (3.4.3) that

$$\begin{aligned} -(\phi''_k(x), \phi_k(x))_{\omega} &= (T''_{k+2}(x), T_k(x))_{\omega} \\ &= (k+2)((k+2)^2 - k^2)(T_k(x), T_k(x))_{\omega} \\ &= 2\pi(k+1)(k+2). \end{aligned}$$

Setting $\phi''_j(x) = \sum_{n=0}^j d_n T_n(x)$, by a simple computation using (4.1.28), we derive

$$d_n = \begin{cases} \frac{1}{c_j} 4(j+1)(j+2), & n = j \\ \frac{1}{c_n} \{(j+2)^3 - j^3 - 2n^2\}, & n < j \end{cases}.$$

Hence for $j = k+2, k+4, \dots$, we find

$$\begin{aligned} -(\phi''_j(x), \phi_k(x))_{\omega} &= d_k(T_k(x), T_k(x))_{\omega} - d_{k+2}(T_{k+2}(x), T_{k+2}(x))_{\omega} \\ &= 4\pi(k+1). \end{aligned}$$

The case $a_{\pm} = 0$ and $b_{\pm} = 1$ can be treated similarly as above. \square

The Chebyshev-Galerkin method for (4.1.5-4.1.4) involves the following steps:

- (1) (pre-computation) Compute $\{a_k, b_k\}$ and nonzero elements of S and M ;
- (2) Evaluate the Chebyshev coefficients of $I_N f(x)$ from $\{f(x_i)\}_{i=0}^N$ (backward Chebyshev transform) and evaluate \tilde{f} ;
- (3) Solve \bar{u} from (4.1.13);
- (4) Evaluate $u_N(x_j) = \sum_{i=0}^{N-2} \hat{u}_i \phi_i(x_j)$, $j = 0, 1, \dots, N$ (forward Chebyshev transform).

Note that the forward and backward Chebyshev transforms can be performed by using the Fast Fourier Transform (FFT) in $O(N \log_2 N)$ operations. However, the cost of Step 3 depends on the boundary conditions (4.1.4). For the special but important cases described in the above Lemma, the special structures of S would allow us to solve the system (4.1.13) in $O(N)$ operations. More precisely, in (4.1.26) and (4.1.27), the nonzero elements of S take the form $s_{kj} = a(j) * b(k)$, hence, a special Gaussian elimination procedure for (4.1.13) (cf. [20]) would only require $O(N)$ flops instead of $O(N^3)$ flops for a general full matrix.

**** ??? add a code for the special Gauss elimination here ??? *****

Therefore, thanks to the fast Fourier transforms which can be used for the discrete Chebyshev transforms, the computational complexity of

Chebyshev-Galerkin method for the above cases is $O(N \log N)$ which is quas-optimal (i.e., optimal up to a logarithmic term).

The following pseudo-code outlines the solution procedure for (4.1.3) by the Chebyshev-Galerkin method:

```

CODE SG.2
Input N
%Set up collocation points  $x_k$  and  $\tilde{c}_k$ 
for j=0 to N do
    x(j)=cos( $\pi j/N$ ),  $\tilde{c}(j)=1$ 
end
 $\tilde{c}(0)=2$ ,  $\tilde{c}(N)=2$ 
Input  $f(x_k)$ 
Compute  $a_k$ ,  $b_k$ ,  $s_{kj}$ ,  $m_{kj}$ 
%Backward Chebyshev transform
for k=0 to N do
     $g_k = \frac{2}{c_k N} \sum_{j=0}^N \frac{1}{c_j} f(x_j) \cos\left(\frac{kj\pi}{N}\right)$ 
end
%Evaluate  $\bar{f}$  from  $f_k = (\sum_{j=0}^N g_j T_j(x), \phi_k(x))$ 
 $f_0 = \frac{\pi}{2}(2g_0 + a_0 g_1 + b_0 g_2)$ 
for k=1 to N-2 do
     $f_k = \frac{\pi}{2}(g_k + a_k g_{k+1} + b_k g_{k+2})$ 
end
Solve  $(\alpha S + M)\hat{u} = \bar{f}$ 
%Evaluate  $g_k$  from  $\sum_{j=0}^{N-2} \hat{u}_j \phi_j(x) = \sum_{j=0}^N g_j T_j(x)$ 
 $g_0 = \hat{u}_0$ 
 $g_1 = \hat{u}_1 + a_0 \hat{u}_0$ 
for k=2 to N-2 do
     $g_k = \hat{u}_k + a_{k-1} \hat{u}_{k-1} + b_{k-2} \hat{u}_{k-2}$ 
end
 $g_{N-1} = a_{N-2} \hat{u}_{N-2} + b_{N-3} \hat{u}_{N-3}$ 
 $g_N = b_{N-2} \hat{u}_{N-2}$ 
%forward Chebyshev transform
for k=0 to N do
     $\hat{u}_k = \sum_{j=0}^N g_j \cos\left(\frac{kj\pi}{N}\right)$ 
end
Output  $\hat{u}_0, \hat{u}_1, \dots, \hat{u}_N$ 

```

4.1.4. Chebyshev-Legendre Galerkin method. The main advantage of using Chebyshev polynomials is that the discrete Chebyshev transforms can be performed in $O(N \log_2 N)$ operations by using FFT. However, the Chebyshev-Galerkin method leads to non-symmetric and full stiffness matrices. On the other hand, the Legendre-Galerkin method leads to symmetric sparse matrices, but the discrete Legendre transforms are expensive ($O(N^2)$ operations). In order to take advantage and overcome disadvantage of both the Legendre and Chebyshev polynomials, one may use the so called

Chebyshev-Legendre Galerkin method:

$$\alpha \int_I u_N v_N dx + \int_I u'_N v'_N dx = \int_I I_N^c f v_N dx, \quad (4.1.29)$$

where I_N^c denotes the interpolation operator relative to the Chebyshev-Gauss-Lobatto points. So the only difference with (4.1.14) is that the Chebyshev interpolation operator I_N^c is used here instead of the Legendre interpolation operator in (4.1.14). Thus, as in the Legendre-Galerkin case, (4.1.29) leads to the linear system (4.1.13) with \bar{u} , S and M defined in (4.1.12) and (4.1.18)–(4.1.19), but \bar{f} defined by

$$f_k = \int_I I_N^c f \phi_k dx, \quad \bar{f} = (f_0, f_1, \dots, f_{N-2})^T. \quad (4.1.30)$$

Hence, the solution procedure of (4.1.29) is essentially the same as that of (4.1.14) except that *Chebyshev-Legendre transforms* (between the value of a function at the *CGL* points and the coefficients of its *Legendre* expansion) are needed instead of the *Legendre transforms*. More precisely, given the values of f at the CGL points $\{x_i = \cos(\frac{i\pi}{N})\}_{0 \leq i \leq N}$, we determine the values of u_N (solution of (4.1.11)) at the CGL points as follows:

- (1) (pre-computation) compute $\{a_k, b_k\}$ and nonzero elements of S and M ;
- (2) Evaluate the Legendre coefficients of $I_N^c f(x)$ from $\{f(x_i)\}_{i=0}^N$ (backward Chebyshev-Legendre transform);
- (3) Evaluate \bar{f} from (4.1.30) and solve \bar{u} from (4.1.13);
- (4) Evaluate $u_N(x_j) = \sum_{i=0}^{N-2} \hat{u}_i \phi_i(x_j)$, $j = 0, 1, \dots, N$ (“modified” forward Chebyshev-Legendre transform).

The backward and forward Chebyshev-Legendre transforms can be efficiently implemented. Indeed, each Chebyshev-Legendre transform can be split into two steps:

- (1) The transform between its values at Chebyshev-Gauss-Lobatto points and the coefficients of its Chebyshev expansion. This can be done by using FFT in $O(N \log_2 N)$ operations.
- (2) The transform between the coefficients of the Chebyshev expansion and of the Legendre expansion. Alpert and Rokhlin [2] have developed an $O(N)$ algorithm for this transform given a prescribed precision.

Therefore, the total computational cost for (4.1.29) is of the order $O(N \log_2 N)$.

The algorithm in [2] is based on the fast multipole method [12]. Hence, it is most attractive for very large N . For moderate N , the algorithm described below appears to be more competitive.

Let us write

$$p(x) = \sum_{i=0}^N f_i T_i(x) = \sum_{i=0}^N g_i L_i(x),$$

and

$$\mathbf{f} = (f_0, f_1, \dots, f_N)^T, \quad \mathbf{g} = (g_0, g_1, \dots, g_N)^T.$$

What we need is to transform between \mathbf{f} and \mathbf{g} . The relation between \mathbf{f} and \mathbf{g} can be easily obtained by computing $(p, T_j)_\omega$ and (p, L_j) . In fact, let us denote

$$a_{ij} = \frac{2}{c_i \pi} (T_i, L_j)_\omega, \quad b_{ij} = (i + \frac{1}{2})(L_i, T_j),$$

where $c_0 = 2$ and $c_i = 1$ for $i \geq 1$, and

$$A = (a_{ij})_{i,j=0,1,\dots,N}, \quad B = (b_{ij})_{i,j=0,1,\dots,N}.$$

Then we have

$$\mathbf{f} = A\mathbf{g}, \quad \mathbf{g} = B\mathbf{f}, \quad AB = BA = I. \quad (4.1.31)$$

By the orthogonality and parity of the Chebyshev and Legendre polynomials, we observe immediately that

$$a_{ij} = b_{ij} = 0, \quad \text{for } i > j \text{ or } i + j \text{ odd.}$$

Hence, both A and B only have about $\frac{1}{4}N^2$ nonzero elements, and the cost of each transform between \mathbf{f} and \mathbf{g} is about $\frac{1}{2}N^2$ operations. Consequently, the cost of each Chebyshev-Legendre transform is about $(\frac{5}{2}N \log_2 N + 4N) + \frac{1}{2}N^2$ operations as opposed to $2N^2$ operations for the Legendre transform. In pure operational counts, the cost of the two transforms is about the same at $N = 8$, and the Chebyshev-Legendre transform costs about one third of the Legendre transform at $N = 128$ (see [21] for computational comparisons of the three methods).

In summary, the one-dimensional Chebyshev-Legendre transform can be done in about

$$(\frac{5}{2}N \log_2 N + 4N) + \min(\frac{1}{2}N^2, CN) \sim O(N \log_2 N)$$

operations, where C is a large constant in Alpert and Rokhlin's algorithm [2]. Since multi-dimensional transforms in the tensor product form are performed through a sequence of one-dimensional transforms, the d -dimensional Chebyshev-Legendre transform can be done in $O(N^d \log_2 N)$ operations and it has the same speedup as in the 1-D case, when compared with the d -dimensional Legendre transform.

The nonzero elements of A and B can be easily determined by the recurrence relations:

$$\begin{aligned} T_{i+1}(x) &= 2xT_i(x) - T_{i-1}(x), \quad i \geq 1, \\ L_{i+1}(x) &= \frac{2i+1}{i+1}xL_i(x) - \frac{i}{i+1}L_{i-1}(x), \quad i \geq 1. \end{aligned}$$

Indeed, let $\tilde{a}_{ij} = (T_i, L_j)_\omega$, then for $j \geq i \geq 1$,

$$\begin{aligned} \tilde{a}_{ij+1} &= (T_i, L_{j+1})_\omega \\ &= (T_i, \frac{2j+1}{j+1}xL_j - \frac{j}{j+1}L_{j-1}) \\ &= \frac{2j+1}{j+1}(xT_i, L_j)_\omega - \frac{j}{j+1}\tilde{a}_{ij-1} \\ &= \frac{2j+1}{2j+2}(T_{i+1} + T_{i-1}, L_j)_\omega - \frac{j}{j+1}\tilde{a}_{ij-1} \\ &= \frac{2j+1}{2j+2}(\tilde{a}_{i+1j} + \tilde{a}_{i-1j}) - \frac{j}{j+1}\tilde{a}_{ij-1}. \end{aligned}$$

Similarly, let $\tilde{b}_{ij} = (L_i, T_j)$, we have for $j \geq i \geq 1$,

$$\tilde{b}_{ij+1} = \frac{2i+2}{2i+1}\tilde{b}_{i+1j} + \frac{2i}{2i+1}\tilde{b}_{i-1j} - \tilde{b}_{ij-1}.$$

Thus, each nonzero element of A and B can be obtained by just a few operations. Furthermore, the Chebyshev-Legendre transform (4.1.31) is extremely easy to implement, while the algorithm in [2] requires considerable programming effort.

REMARK 4.1.3. Note that only for equations with constant or polynomial (and rational polynomials in some special cases) coefficients, one can expect the matrices resulting from a Galerkin method to be sparse or with special structures. In the more general cases such as (4.1.3), the Galerkin matrices are usually full so a direct application of the Galerkin methods is not advisable. However, for many practical situations, the Galerkin system for a suitable constant coefficient problems provides an optimal preconditioner for solving problems with variable coefficients, see Section 4.4 for further details.

4.2. Collocation methods

The collocation method, or more specifically the *collocation method in the strong form*, is fundamentally different from the Galerkin method. Unlike the Galerkin method, the collocation method is not based on a variational formulation (although in some special cases to be specified below, the collocation method can be reformulated as a variational method). Instead, it looks for an approximate solution which enforce the boundary conditions **exactly**, and for which the equation (4.1.3) is verified at a set of interior collocation points. On the other hand, the *collocation method in the weak form* (See the next section) is based on a suitable variational formulation in which the general boundary conditions are treated *naturally* and are only satisfied asymptotically, and the approximate solution verifies (4.1.3) at a set of interior collocation points.

We describe below the collocation method for the two-point boundary value problem (4.1.3) with the general boundary condition (4.0.2). For the

sake of clarity, we consider first the Dirichlet boundary condition followed by the treatment of general boundary condition (4.0.2). Note also that the non homogeneous boundary conditions can be treated directly in a collocation method so there is no need to “homogenize” the boundary conditions as we did in the previous section for the Galerkin method.

4.2.1. Dirichlet boundary conditions. We consider (4.1.3) with the Dirichlet boundary conditions $u(\pm 1) = c_{\pm}$. Given a set of collocation points $\{x_j\}_{j=0}^N$ in descending order with $x_0 = 1$ and $x_N = -1$, we look for $u_N \in P_N$ such that

$$\begin{aligned} -\varepsilon u_N''(x_j) + p(x_j)u_N'(x_j) + q(x_j)u_N(x_j) \\ = f(x_j), \quad 1 \leq j \leq N-1, \end{aligned} \quad (4.2.1)$$

$$u_N(-1) = c_-, \quad u_N(1) = c_+.$$

Let $\{h_j(x)\}$ be the Lagrange interpolation polynomials associated with $\{x_j\}$ and $d_{kj} = h_j'(x_k)$, $D = (d_{kj})_{k,j=0,1,\dots,N}$. Taking $u^{(m)}(x) = h_i''(x)$ in (3.3.16), we find

$$(D^2)_{ki} := h_i''(x_k) = \sum_{j=0}^N d_{kj}h_j'(x_k) = \sum_{j=0}^N d_{kj}d_{ji}. \quad (4.2.2)$$

Let $\{w_j = u_N(x_j)\}_{j=0}^N$ be the unknowns to be determined, we can write

$$u_N(x_k) = \sum_{j=0}^N w_j h_j(x_k) = \delta_{kj} w_j,$$

$$\begin{aligned} u_N'(x_k) &= \sum_{j=0}^N w_j h_j'(x_k) = \sum_{j=0}^N d_{kj} w_j \\ &= \sum_{j=1}^{N-1} d_{kj} w_j + d_{k0} w_0 + d_{kN} w_N, \end{aligned}$$

$$\begin{aligned} u_N''(x_k) &= \sum_{j=0}^N w_j h_j''(x_k) = \sum_{j=0}^N (D^2)_{kj} w_j \\ &= \sum_{j=1}^{N-1} (D^2)_{kj} w_j + (D^2)_{k0} w_0 + (D^2)_{kN} w_N. \end{aligned}$$

Substitute the above in (4.2.1), we can rewrite (4.2.1) as

$$\begin{aligned} \sum_{j=1}^{N-1} [-\varepsilon (D^2)_{ij} + p(x_i)d_{ij} + q(x_i)\delta_{ij}] w_j = f(x_i) \\ - [-\varepsilon (D^2)_{i0} + p(x_i)d_{i0}] c_+ - [-\varepsilon (D^2)_{iN} + p(x_i)d_{iN}] c_-. \end{aligned} \quad (4.2.3)$$

Let us denote

$$\begin{aligned}
a_{ij} &= -\varepsilon (D^2)_{ij} + p(x_i)d_{ij} + q(x_i)\delta_{ij}, \quad A = (a_{ij})_{N-1, N-1}, \\
b_i &= f(x_i) - [-\varepsilon (D^2)_{i0} + p(x_i)d_{i0}] c_+ \\
&\quad - [-\varepsilon (D^2)_{iN} + p(x_i)d_{iN}] c_-, \\
\bar{w} &= (w_1, w_2, \dots, w_{N-1})^T, \quad \bar{b} = (b_1, b_2, \dots, b_{N-1})^T.
\end{aligned} \tag{4.2.4}$$

Then, we can rewrite (4.2.3) as

$$A\bar{w} = \bar{b}. \tag{4.2.5}$$

REMARK 4.2.1. Notice that the above formulation is valid for any set of collocation points. However, the choice of collocation points is important for the convergence and efficiency of the collocation method. For two-point boundary value problems, only Gauss-Lobatto points should be used.

A Galerkin reformulation

It is interesting to note that the collocation method (4.2.1) can be reformulated into a suitable variational formulation. Indeed, we have

LEMMA 4.5. *Let $\omega = (1+x)^\alpha(1-x)^\beta$ be the Jacoby weight, $\{x_j\}_{j=0}^N$ be the Jacoby-Gauss-Lobatto points and $\langle \cdot, \cdot \rangle_{N, \omega}$ be the discrete inner product associated to the Jacoby-Gauss-Lobatto quadrature. Then, (4.2.1) with $c_\pm = 0$ is equivalent to:*

Find $u_N \in P_N^0$ such that

$$\begin{aligned}
\varepsilon \langle u'_N, \omega^{-1}(v_N \omega)' \rangle_{N, \omega} + \langle p(x)u'_N, v_N \rangle_{N, \omega} \\
+ \alpha \langle u_N, v_N \rangle_{N, \omega} = \langle f, v \rangle_{N, \omega} \quad \text{for all } v_N \in P_N^0.
\end{aligned} \tag{4.2.6}$$

PROOF. By direct computation, we find

$$\omega^{-1}(v_N \omega)' = \omega^{-1}(v'_N \omega + v \omega') = v'_N + [\alpha(1+x) - \beta(1-x)] \frac{v_N}{1-x^2}.$$

Since $v_N(\pm 1) = 0$ and $v_N \in P_N$, we derive that $\omega^{-1}(v_N \omega)' \in P_{N-1}$. Therefore, thanks to (3.1.15), we find that

$$\begin{aligned}
\langle u'_N, \omega^{-1}(v_N \omega)' \rangle_{N, \omega} &= (u'_N, \omega^{-1}(v_N \omega)')_\omega \\
&= -(u''_N, v_N)_\omega = -\langle u''_N, v_N \rangle_{N, \omega}.
\end{aligned} \tag{4.2.7}$$

Now, take the discrete inner product of (4.2.1) with $h_k(x)$ for $k = 1, 2, \dots, N-1$. Thanks to (4.2.7) and the fact that

$$P_N^0 = \text{span}\{h_1(x), h_2(x), \dots, h_{N-1}(x)\},$$

we find that the solution u_N of (4.2.1) verifies (4.2.6). \square

This lemma indicates that for (4.1.3) with the Dirichlet boundary condition, the Jacoby-collocation method, including the Legendre- and Chebyshev-collocation methods, can be reformulated as a *Galerkin method with numerical integration*. An obvious advantage of this reformulation is that error

estimates for the collocation method can be derived in the same way as in the Galerkin method.

A pseudo-code for (4.2.1) using the Chebyshev collocation points is given below.

```

CODE PSBVP.1
Input N, ε, p(x), q(x), f(x), c-, c+
%collocation points
for j=0 to N do
    x(j)=cos(πj/N)
end
%compute first-order differentiation matrix
call CODE ChyDM.1 in Section 3.4 to get D
%compute second-order differentiation matrix
D2=D*D
% compute the stiffness matrix A
for i=1 to N-1 do
    for j=1 to N-1 do
        if i=j
            A(i,j)=-ε*D2(i,j)+p(x(i))*D(i,j)+q(x(i))
        else
            A(i,j)=-ε*D2(i,j)+p(x(i))*D(i,j)
        end
    end
end
% compute the right side vector b
ss1=-ε*D2(i,0)+p(x(i))*D(i,0);
ss2=-ε*D2(i,N)+p(x(i))*D(i,N);
b(i)=f(i)-ss1*c+-ss2*c-
end
% solve the linear system to get the unknown vector
u=A-1b
Output u(1), u(2), ..., u(N-1)

```

4.2.2. General boundary conditions. We now consider the general boundary conditions (4.0.2). The collocation method for (4.1.3)–(4.0.2) is: Find $u_N \in P_N$ such that

$$\begin{aligned}
 -\varepsilon u_N''(x_j) + p(x_j)u_N'(x_j) \\
 + q(x_j)u_N(x_j) = f(x_j), \quad 1 \leq j \leq N-1,
 \end{aligned} \tag{4.2.8}$$

$$a_- u_N(-1) + b_- u_N'(-1) = c_-, \quad a_+ u_N(1) + b_+ u_N'(1) = c_+. \tag{4.2.9}$$

The first step is to determine w_0 and w_N in terms of $\{w_j\}_{j=1}^{N-1}$. It follows from (4.0.2) that

$$a_- w_N + b_- \sum_{j=0}^N d_{Nj} w_j = c_-, \quad a_+ w_0 + b_+ \sum_{j=0}^N d_{0j} w_j = c_+,$$

which leads to

$$\begin{aligned} b_- d_{N0} w_0 + (a_- + b_- d_{NN}) w_N &= c_- - b_- \sum_{j=1}^{N-1} d_{Nj} w_j, \\ (a_+ + b_+ d_{00}) w_0 + b_+ d_{0N} w_N &= c_+ - b_+ \sum_{j=1}^{N-1} d_{0j} w_j. \end{aligned}$$

Solving the above equations give

$$w_0 = \tilde{c}_+ - \sum_{j=1}^{N-1} \tilde{\alpha}_{0j} w_j, \quad w_N = \tilde{c}_- - \sum_{j=1}^{N-1} \tilde{\alpha}_{Nj} w_j, \quad (4.2.10)$$

where the parameters \tilde{c}_+ , $\tilde{\alpha}_{0j}$, \tilde{c}_- , $\tilde{\alpha}_{Nj}$ are defined by

$$\begin{aligned} \tilde{c}_+ &= (\tilde{d}c_- - \tilde{b}c_+)/DET, \\ \tilde{\alpha}_{0j} &= (\tilde{d}b_- d_{Nj} - \tilde{b}b_+ d_{0j})/DET, \\ \tilde{c}_- &= (\tilde{a}c_+ - \tilde{c}c_-)/DET, \\ \tilde{\alpha}_{Nj} &= (\tilde{a}b_+ d_{0j} - \tilde{c}b_- d_{Nj})/DET, \end{aligned}$$

with

$$\begin{aligned} \tilde{a} &= b_- d_{N0}, \quad \tilde{b} = a_- + b_- d_{NN}, \\ \tilde{c} &= a_+ + b_+ d_{00}, \quad \tilde{d} = b_+ d_{0N}, \\ DET &= \tilde{a}\tilde{d} - \tilde{b}\tilde{c}. \end{aligned}$$

Note that in both the Legendre and Chebyshev cases, we have $d_{N0} = -d_{0N}$, $d_{00} > 0$ and $d_{NN} < 0$ (see Sections 3.3 and 3.4). With the assumption $a_{\pm} \geq 0$, (4.0.3) implies that $DET < 0$.

Therefore, the general boundary conditions (4.2.9) can be replaced by (4.2.10) which is of Dirichlet type. Hence, the previous discussion about (4.2.1) enables us to rewrite the collocation system (4.2.8) as

$$A\bar{w} = \bar{b}, \quad (4.2.11)$$

where $A = (a_{ij})$ is a $(N-1) \times (N-1)$ matrix and $\bar{b} = (b_j)$ is a $(N-1)$ -dimensional vector with

$$\begin{aligned} a_{ij} &= -\varepsilon (D^2)_{ij} + p(x_i) d_{ij} + q(x_i) \delta_{ij} \\ &\quad - [-\varepsilon (D^2)_{i0} + p(x_i) d_{i0}] \tilde{\alpha}_{0j} - [-\varepsilon (D^2)_{iN} + p(x_i) d_{iN}] \tilde{\alpha}_{Nj}, \\ b_i &= f(x_i) - [-\varepsilon (D^2)_{i0} + p(x_i) d_{i0}] \tilde{c}_+ - [-\varepsilon (D^2)_{iN} + p(x_i) d_{iN}] \tilde{c}_-. \end{aligned}$$

A Petrov-Galerkin reformulation

Unlike the Dirichlet case, The collocation method (4.2.8)–(4.2.9) can not be formulated as a Galerkin method. However, it can be reformulated into

a Petrov-Galerkin method for which the trial functions and test functions are taken from different spaces. Indeed, we have

LEMMA 4.6. *Let $\omega = (1+x)^\alpha(1-x)^\beta$ be the Jacoby weight, $\{x_j\}_{j=0}^N$ be the Jacoby-Gauss-Lobatto points and $\langle \cdot, \cdot \rangle_{N,\omega}$ be the discrete inner product associated to the Jacoby-Gauss-Lobatto quadrature. Then, (4.2.8)–(4.2.9) with $c_\pm = 0$ is equivalent the following Petrov-Galerkin method:*

Find $u_N \in X_N$ such that

$$\begin{aligned} -\varepsilon \langle u'_N, \omega^{-1}(v_N \omega)' \rangle_{N,\omega} + \langle p(x)u'_N, v_N \rangle_{N,\omega} \\ + \langle q(x)u_N, v_N \rangle_{N,\omega} = \langle f, v \rangle_{N,\omega} \quad \text{for all } v_N \in P_N^0, \end{aligned} \quad (4.2.12)$$

where X_N is defined in (4.1.10).

PROOF. By definition, the solution u_N of (4.2.8)–(4.2.9) is in X_N . Taking the discrete inner product of (4.2.8) with $h_k(x)$ for $k = 1, 2, \dots, N-1$, we find that the solution u_N of (4.2.8)–(4.2.10) verifies (4.2.12) thanks to (4.2.7) and the fact that

$$P_N^0 = \text{span}\{h_1(x), h_2(x), \dots, h_{N-1}(x)\}.$$

□

This reformulation will allow us to obtain error estimates for the collocation method (4.2.8)–(4.2.9) using the standard techniques developed for Petrov-Galerkin methods.

A pseudo-code for (4.2.8)–(4.2.9) using the Chebyshev collocation points is given below.

```

CODE PSBVP.3
Input N, ε, p(x), q(x), f(x), a±, b±, c±
%collocation points
for j=0 to N do
    x(j)=cos(πj/N)
end
%first-order differentiation matrix
call CODE ChyDM.1 in Section 3.4 to get D
%compute second-order differentiation matrix
D2=D*D
% calculate some constants
ta=b-*D(N,0);  tb=a_-+b_*D(N,N)
tc=a_+ +b_+*D(0,0);  td=b_+*D(0,N)
te=ta*td-tc*tb
~c_+=(td*c_- -tb*c_+)/te
~c_+=(ta*c_+ -tc*c_-)/te
% compute the stiffness matrix A
for i=1 to N-1 do
    ss1=-ε *D2(i,0)+p(x(i))*D(i,0)
    ss2=-ε *D2(i,N)+p(x(i))*D(i,N)

```

```

for j=1 to N-1 do
  ss3=(td*b_-*D(N,j)-tb*b_+*D(0,j))/te
  ss4=(ta*b_+*D(0,j)-tc*b_-*D(N,j))/te
  ss5=ss1*ss3+ss2*ss4
  if i=j
    A(i,j)=-ε*D2(i,j)+p(x(i))*D(i,j)+q(x(i))-ss5
  else
    A(i,j)=-ε*D2(i,j)+p(x(i))*D(i,j)-ss5
  end
end
% compute the right side vector b
b(i)=f(i)-ss1*c_+ -ss2*c_-
end
% solve the linear system to get the unknown vector
u=A-1b
Output u(1), u(2), ..., u(N-1)

```

4.2.3. Numerical experiments. In this subsection we will consider two numerical examples. The numerical results will be obtained by using CODE PSBVP.1 and CODE PSBVP.3, respectively.

EXAMPLE 4.2.1. Consider the following problem

$$\begin{aligned}
 -u'' - xu'(x) + u(x) &= f(x), \quad x \in (-1, 1), \\
 u(\pm 1) &= e^{\pm 5} + \sin(1),
 \end{aligned}$$

where

$$f(x) = (4x^2 + 1) \sin(x^2) - (24 + 5x)e^{5x} - (2 + 2x^2) \cos(x^2)$$

so that the exact solution for Example 4.2.1 is

$$u(x) = e^{5x} + \sin(x^2).$$

We solve this problem by using different values of N and compute the maximum error which is defined by

$$\text{Maximum Error} = \max_{1 \leq j \leq N-1} |w_j - u(x_j)|.$$

It is the maximum error at the interior collocation points. Here is the output.

N	Maximum error	N	Maximum error
5	2.828e+00	13	6.236e-06
6	8.628e-01	14	9.160e-07
7	1.974e-01	15	1.280e-07
8	3.464e-02	16	1.689e-08
9	7.119e-03	17	2.135e-09
10	1.356e-03	18	2.549e-10
11	2.415e-04	19	2.893e-11
12	3.990e-05	20	3.496e-12

Exponential convergence rate can be observed from the above table. For comparison, we also solve Example 4.2.1 using the finite-difference method. We use the central differences for the derivatives:

$$u'' \approx \frac{w_{j+1} - 2w_j + w_{j-1}}{h^2},$$

$$u' \approx \frac{w_{j+1} - w_{j-1}}{2h}, \quad h = \frac{2}{N}.$$

As usual the mesh points are given by $x_j = -1 + jh$. The maximum errors given by the finite-difference method are listed below.

N	Maximum error
16	3.100e+00
32	7.898e-01
64	1.984e-01
128	4.968e-02
256	1.242e-02
512	3.106e-03

As expected that the convergence rate for the central difference method is 2. The error obtained by the finite differences with $N = 512$ is almost the same as that obtained by the spectral method with $N = 10$.

The following example deals with BVPs with the general boundary conditions. We follow `CODE PSBVP.3` and use `MATLAB` to get the following results.

EXAMPLE 4.2.2. Consider the same problem as above, except with different boundary conditions:

$$u(-1) - u'(-1) = -4e^{-5} + \sin(1) + 2 \cos(1),$$

$$u(1) + u'(1) = 6e^5 + \sin(1) + 2 \cos(1).$$

The exact solution is also $u(x) = e^{5x} + \sin(x^2)$.

The numerical results are given below.

N	Maximum error	N	Maximum error
5	3.269e+01	13	3.254e-04
6	9.696e+00	14	4.903e-05
7	2.959e+00	15	8.823e-06
8	7.292e-01	16	1.164e-06
9	1.941e-01	17	1.884e-07
10	3.996e-02	18	2.204e-08
11	9.219e-03	19	3.225e-09
12	1.609e-03	20	3.432e-10

It is observed that the convergence rate for problems with general boundary conditions is slower than that for problems with Dirichlet boundary conditions. This can be heristically explained by the fact that unlike in the Dirichlet case, the collocation method (4.2.1) with Neumann or mixed

boundary conditions (4.0.2) can not be reformulated as a Galerkin method, but instead it corresponds to a Petrov-Galerkin method which does not seem to lead to optimal error estimates. In the next section, we will present a weak form of the collocation method for (4.1.3)–(4.1.4) which can be reformulated as a Galerkin method.

REMARK 4.2.2. The main advantage of a collocation method is its simplicity and flexibility in implementation. In fact, problems with variable coefficients and general boundary conditions are treated the same way and result in same type of linear systems as problems with constant coefficients and simple boundary conditions. However, It is well known that the *collocation* matrices, even for the simplest differential equation, are full and ill conditioned, so it is in general not advisable to invert the derivative matrices using a direct method for N large. Instead, an iterative method using an appropriate preconditioner should be used, see Section 4.4 for more details. In particular, the above pseudo-codes are only suitable for small and moderate N .

4.3. Collocation methods in the weak form

The *Galerkin* method is based on a variational formulation which usually preserves essential properties of the continuous problem such as coercivity, continuity and symmetry of the bilinear form, and it usually leads to optimal error estimates. However, the Galerkin method presented in Section 4.1 is not feasible for problems with general variable coefficients. On the other hand, the *collocation method* is flexible and easy to implement, but it can not be always reformulated into a suitable variational formulation which is usually necessary for obtaining optimal error estimates (see Section 4.5 below). A third approach is to use a so called *collocation method in the weak form*, or sometimes called *Galerkin method with numerical integration*, which is based on a variational formulation with discrete inner product. We have noted in the last section that the collocation method for (4.1.3) with Dirichlet boundary conditions can be reformulated as a *Galerkin method with numerical integration* so here we will consider (4.1.3) with the general boundary conditions (4.1.4).

Since the weighted variational formulation (4.1.9) does not make sense in the Chebyshev case with (4.1.4), we shall only deal with the Legendre case. The *Legendre-collocation method in the weak form* for (4.1.3)–(4.1.4) is:

Find $u_N \in \tilde{X}_N = P_N \cap H_*^1(I)$ such that

$$b_N(u_N, v_N) = \langle f, v \rangle_N \quad \text{for all } v_N \in \tilde{X}_N, \quad (4.3.1)$$

where

$$\begin{aligned} b_N(u_N, v_N) := & \varepsilon \langle u'_N, v'_N \rangle_N + \varepsilon h_+ u_N(1) v_N(1) - \varepsilon h_- u_N(-1) v_N(-1) \\ & + \langle p(x) u'_N, v_N \rangle_N + \langle q(x) u_N, v_N \rangle_N. \end{aligned}$$

To fix the idea, we assume $b_{\pm} \neq 0$. Let us denote

$$\begin{aligned} u_N(x) &= \sum_{k=0}^N u_N(x_k) h_k(x), \quad \bar{w} = (u_N(x_0), u_N(x_1), \dots, u_N(x_N))^T, \\ a_{kj} &= b_N(h_j, h_k), \quad A = (a_{kj})_{k,j=0,1,\dots,N}, \\ \bar{f} &= (f(x_0), f(x_1), \dots, f(x_N))^T, \quad W = \text{diag}(\omega_0, \omega_1, \dots, \omega_N), \end{aligned}$$

where $\{\omega_k\}_{k=0}^N$ are the weights in the Legendre-Gauss-Lobatto quadrature. Then, (4.3.1) is equivalent to the linear system

$$A\bar{w} = W\bar{f}. \quad (4.3.2)$$

The entries a_{kj} can be determined as follows. Using (3.1.15) and integration by parts, we have

$$\begin{aligned} \langle h'_j, h'_k \rangle_N &= (h'_j, h'_k) = -(h''_j, h_k) + h'_j h_k|_{\pm 1} \\ &= -(D^2)_{kj} \omega_k + d_{0j} \delta_{0k} - d_{Nj} \delta_{Nk}. \end{aligned} \quad (4.3.3)$$

Consequently,

$$\begin{aligned} a_{kj} &= [-\varepsilon (D^2)_{kj} + p(x_k) d_{kj} + q(x_k) \delta_{kj}] \omega_k \\ &\quad + \varepsilon (d_{0j} + h_+ \delta_{0j}) \delta_{0k} - \varepsilon (d_{Nj} + h_- \delta_{Nj}) \delta_{Nk}. \end{aligned} \quad (4.3.4)$$

Note that the matrix A here is of order $(N+1) \times (N+1)$ instead of order $(N-1) \times (N-1)$ as in the collocation case (4.2.11).

We can also reinterpret (4.3.1) in a collocation form. To this end, we observe that

$$\langle u'_N, h'_k \rangle_N = -u''_N(x_k) \omega_k + u'_N(1) \delta_{0k} - u'_N(-1) \delta_{Nk}.$$

Then, taking $v_N = h_j(x)$ in (4.3.1) for $j = 0, 1, \dots, N$, since $\omega_0 = \omega_N = \frac{2}{N(N+1)}$, we find

$$\begin{aligned} -\varepsilon u''_N(x_j) + p(x_j) u'_N(x_j) \\ + q(x_j) u_N(x_j) &= f(x_j), \quad 1 \leq j \leq N-1, \\ a_- u_N(-1) + b_- u'_N(-1) &= \frac{b_-}{\varepsilon} \frac{2}{N(N+1)} \\ &\quad [f(-1) - (-\varepsilon u''_N(-1) + p(-1) u'_N(-1) + q(-1) u_N(-1))], \\ a_+ u_N(1) + b_+ u'_N(1) &= \frac{b_+}{\varepsilon} \frac{2}{N(N+1)} \\ &\quad [f(1) - (-\varepsilon u''_N(1) + p(1) u'_N(1) + q(1) u_N(1))]. \end{aligned} \quad (4.3.5)$$

We see that the solution of (4.3.1) satisfies (4.1.3) exactly at the interior collocation points $\{x_j\}_{j=1}^{N-1}$, but the boundary condition (4.1.4) is only satisfied approximately with an error proportional to the residue of the equation (4.1.3), with u replaced by the approximate solution u_N , at the boundary. Thus, (4.3.1) does not correspond exactly to a collocation method, so it is

referred as a collocation method in the weak form. It is clear that in the Dirichlet case (i.e. $b_{\pm} = 0$), the collocation method in the weak form (4.3.5) is *equivalent* to the collocation method.

4.4. Preconditioned iterative method

As we have seen in the previous two sections, there is no suitable direct spectral solver for equations with general variable coefficients. Hence, an appropriate iterative method should be used. Since the bilinear form associated to the equation (4.1.3)-(4.1.4) is not symmetric nor necessarily positive definite, it is in general not advisable to apply an iterative method directly, unless the equation is diffusion dominant, i.e., ε is sufficiently large with respect to $p(x)$. Note that the equation (4.1.3)-(4.1.4) can be transformed into an equivalent equation whose bilinear form becomes positive definite. Indeed, multiplying the function

$$a(x) = \exp\left(-\frac{1}{\varepsilon} \int p(x) dx\right)$$

to (4.1.3) and since $-\varepsilon a'(x) = a(x)p(x)$, we find that (4.1.3) is equivalent to

$$-(a(x)u'(x))' + b(x)u = g(x), \quad (4.4.1)$$

where $b(x) = a(x)q(x)/\varepsilon$ and $g(x) = a(x)f(x)/\varepsilon$. We assume hereafter that there are three constants c_1 , c_2 and c_3 such that

$$0 < c_1 \leq a(x) \leq c_2, \quad 0 \leq b(x) \leq c_3. \quad (4.4.2)$$

Let us denote

$$\begin{aligned} b(u, v) = & \int_{-1}^1 a(x)u'v' dx + a(1)h_+u(1)v(1) - a(-1)h_-u(-1)v(-1) \\ & + \int_{-1}^1 b(x)uv dx, \quad u, v \in H_{\star}^1(I), \end{aligned} \quad (4.4.3)$$

where h_{\pm} and $H_{\star}^1(I)$ are defined in (4.1.7) and (4.1.6). The variational form associated with (4.4.1)-(4.1.4) is:

Find $u \in H_{\star}^1(I)$ such that

$$b(u, v) = (g, v), \quad \text{for all } v \in H_{\star}^1(I). \quad (4.4.4)$$

Hence, under the conditions (4.0.3) and (4.4.2), we find that $b(u, v)$ is continuous and coercive so that the problem (4.4.4) admits a unique solution. Hence, instead of (4.1.3)-(4.1.4), we shall consider below its equivalent form (4.4.1)-(4.1.4) whose associated bilinear form is symmetric and positive definite.

4.4.1. Preconditioning in frequency space. Let p_k be the Legendre or Chebyshev polynomials of degree k , X_N be defined in (4.1.10), and $\{\phi_k = p_k + a_k p_{k+1} + b_k p_{k+2}\}_{k=0}^{N-2}$ be the basis functions of X_N constructed in Section 4.1. Let I_N be the interpolation operator based on the Legendre or Chebyshev Gauss-Lobatto points $\{x_j\}_{j=0}^N$ and $\langle \cdot, \cdot \rangle_{N,\omega}$ be the associated discrete inner product. We consider the following *Galerkin method with numerical integration* (4.4.1)–(4.4.4):

Find $u_N = \sum_{k=0}^{N-2} \hat{u}_k \phi_k \in X_N$, such that

$$b_{N,\omega}(u_N, \phi_j) = \langle g, \phi_j \rangle_{N,\omega}, \quad j = 0, 1, \dots, N-2, \quad (4.4.5)$$

where

$$b_{N,\omega}(u_N, v_N) := -\langle [I_N(au'_N)]', v_N \rangle_{N,\omega} + \langle bu_N, v_N \rangle_{N,\omega}. \quad (4.4.6)$$

Let us denote

$$\begin{aligned} b_{ij} &= b_{N,\omega}(\phi_j, \phi_i), \quad B = B_N = (b_{ij})_{i,j=0,1,\dots,N-2}, \\ g_i &= \langle g, \phi_i \rangle_{N,\omega}, \quad \bar{g} = (g_0, g_1, \dots, g_{N-2})^T, \\ \bar{u} &= (\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-2})^T. \end{aligned} \quad (4.4.7)$$

Then, (4.4.5) is equivalent to the following linear system:

$$B\bar{u} = \bar{g}. \quad (4.4.8)$$

Furthermore, for $u_N = \sum_{k=0}^{N-2} \hat{u}_k \phi_k \in X_N$ and $v_N = \sum_{k=0}^{N-2} \hat{v}_k \phi_k \in X_N$, we have

$$\langle B\bar{u}, \bar{v} \rangle_{l^2} = b_{N,\omega}(u_N, v_N). \quad (4.4.9)$$

It is easy to see that in general B is a full matrix. So we shall resort to an iterative method for solving (4.4.8).

Notice that the above scheme is neither a usual Galerkin nor collocation method, it is rather a combination of the two methods. We will see below that the bilinear form $b_{N,\omega}(\cdot, \cdot)$ is so designed to preserve, whenever possible, the coercivity and symmetry of the original problem, and to allow efficient evaluation of the matrix-vector product $B\bar{u}$.

We now describe how to evaluate

$$(B\bar{u})_j = -\langle [I_N(au'_N)]', \phi_j \rangle_{N,\omega} + \langle bu_N, \phi_j \rangle_{N,\omega} \quad \text{for } j = 0, 1, \dots, N-2$$

in some detail. Given $u_N = \sum_{k=0}^{N-2} \hat{u}_k \phi_k$, we compute “ $-\langle [I_N(au'_N)]', \phi_j \rangle_{N,\omega}$ ” as follows:

- (1) Using (3.3.14) or (3.4.14) to determine $\tilde{u}_k^{(1)}$ from

$$u'_N(x) = \sum_{k=0}^{N-2} \hat{u}_k \phi'_k(x) = \sum_{k=0}^N \tilde{u}_k^{(1)} p_k(x);$$

- (2) (Forward discrete transform) Compute

$$u'_N(x_j) = \sum_{k=0}^N \tilde{u}_k^{(1)} p_k(x_j), \quad j = 0, 1, \dots, N;$$

(3) (Backward discrete transform) Determine $\{\tilde{w}_k\}$ from

$$I_N(au'_N)(x_j) = \sum_{k=0}^N \tilde{w}_k p_k(x_j), \quad j = 0, 1, \dots, N;$$

(4) Using (3.3.14) or (3.4.14) to determine $\{\tilde{w}_k^{(1)}\}$ from

$$[I_N(au'_N)]'(x) = \sum_{k=0}^N \tilde{w}_k p'_k(x) = \sum_{k=0}^N \tilde{w}_k^{(1)} p_k(x);$$

(5) For $j = 0, 1, \dots, N-2$, compute

$$-\langle [I_N(au'_N)]', \phi_j \rangle_\omega = -\sum_{k=0}^N \tilde{w}_k^{(1)} \langle p_k, \phi_j \rangle_{N,\omega}.$$

Note that the main cost in the above procedure is the two discrete transforms in Steps 2 and 3. The cost for Steps 1, 4 and 5 are all $O(N)$ flops. The term $\langle bu_N, \phi_j \rangle_{N,\omega}$ can also be computed similarly as follows:

(1) Compute

$$u_N(x_j) = \sum_{k=0}^N \hat{u}_k \phi_k(x_j), \quad j = 0, 1, \dots, N,$$

(2) Determine $\{\tilde{w}_k\}$ from

$$I_N(bu_N)(x_j) = \sum_{k=0}^N \tilde{w}_k p_k(x_j), \quad j = 0, 1, \dots, N;$$

(3) compute

$$-\langle bu_N, \phi_j \rangle_{N,\omega}, \quad j = 0, 1, \dots, N-2.$$

Hence, if $b(x)$ is not a constant, two additional discrete transforms are needed. In summary, the total cost for evaluate $B\bar{u}$ is dominated by four (only two if $b(x)$ is a constant) discrete transforms, and is $O(N^2)$ (resp. $O(N \log N)$) flops in the Legendre (resp. Chebyshev) case.

Legendre case

Thanks to (3.1.15), we have for $u_N, v_N \in X_N$,

$$\begin{aligned} -\langle [I_N(au'_N)]', v_N \rangle_N &= \langle au'_N, v'_N \rangle_N + a(1)h_+ u_N(1)v_N(1) \\ &\quad - a(-1)h_- u_N(-1)v_N(-1), \end{aligned} \quad (4.4.10)$$

where h_\pm is defined in (4.1.7). Hence,

$$b_N(u_N, v_N) = b_N(v_N, u_N), \quad \text{for all } u_N, v_N \in X_N.$$

Consequently, B is symmetric.

Under the conditions (4.0.3) and (4.4.2), we have

$$\begin{aligned} b_N(u_N, u_N) &= \langle au'_N, u'_N \rangle_N + a(1)h_+ u_N^2(1) - a(-1)h_- u_N^2(-1) \\ &\geq c_1 \langle u'_N, u'_N \rangle_N = c_1(u'_N, u'_N). \end{aligned}$$

On the other hand, using Lemma 3.6, the Poincaré inequality

$$\|u\|_{L^2} \leq c_0|u|_{H^1}, \quad \text{for all } u \in H_\star^1(I), \quad (4.4.11)$$

and the Sobolev inequality (cf. [1])

$$\|u\|_{L^\infty(I)} \lesssim \|u\|_{L^2(I)}^{\frac{1}{2}} \|u\|_{H^1(I)}^{\frac{1}{2}}, \quad (4.4.12)$$

we find that

$$b_N(u_N, u_N) \leq c_4(u'_N, u'_N).$$

Hence, let $s_{ij} = (\phi'_j, \phi'_i)$ and $S = (s_{ij})_{i,j=0}^{N-2}$. We have

$$0 < c_1 \leq \frac{\langle B\bar{u}, \bar{u} \rangle_{l^2}}{\langle S\bar{u}, \bar{u} \rangle_{l^2}} = \frac{b_N(u_N, u_N)}{(u'_N, u'_N)} \leq c_4. \quad (4.4.13)$$

Since $S^{-1}B$ is symmetric with respect to the inner product $\langle \bar{u}, \bar{v} \rangle_S := \langle S\bar{u}, \bar{v} \rangle_{l^2}$, (4.4.13) implies immediately

$$\text{cond}(S^{-1}B) \leq \frac{c_4}{c_1}. \quad (4.4.14)$$

In other words, S^{-1} is an optimal preconditioner for B , and the convergence rate of the conjugate gradient method applied to the preconditioned system

$$S^{-1}B\bar{u} = S^{-1}\bar{g} \quad (4.4.15)$$

will be independent of N . We recall from Section 4.1 that S is a diagonal matrix so the cost of applying S^{-1} is negligible. Hence, the main cost in each iteration is the evaluation of $B\bar{u}$ for \bar{u} given.

REMARK 4.4.1. In the case of Dirichlet boundary condition, we have $\phi_k(x) = L_k(x) - L_{k+2}(x)$ which, together with (3.3.5), implies that $\phi'_k(x) = -(2k+3)L_{k+1}(x)$. Therefore, from $u = \sum_{k=0}^{N-2} \hat{u}_k \phi_k(x)$, we can obtain the derivative $u' = -\sum_{k=0}^{N-2} (2k+3)\hat{u}_k L_{k+1}(x)$ in the frequency space without using (3.3.14).

REMARK 4.4.2. If we use the normalized basis function

$$\tilde{\phi}_k := \sqrt{-b_k(4k+6)}^{-1} \phi_k \quad \text{with } (\tilde{\phi}'_j, \tilde{\phi}'_i) = \delta_{ij},$$

the condition number of the corresponding matrix B with $b_{ij} = b(\tilde{\phi}_j, \tilde{\phi}_i)$ is uniformly bounded. Hence, we can apply the conjugate gradient method directly to this matrix without preconditioning.

REMARK 4.4.3. If c_3 in (4.4.2) is large, the condition number in (4.4.14) will be large even though independent of N . In this case, one may replace the bilinear form (u'_N, v'_N) by $\hat{a}(u'_N, v'_N) + \hat{b}(u, u)$ with

$$\hat{a} = \frac{1}{2} \left(\max_{x \in [-1,1]} a(x) + \min_{x \in [-1,1]} a(x) \right), \quad \hat{b} = \frac{1}{2} \left(\max_{x \in [-1,1]} b(x) + \min_{x \in [-1,1]} b(x) \right).$$

The matrix corresponding to this new bilinear form is $\hat{a}S + \hat{b}M$ which is positive definite and penta-diagonal (cf. Section 4.1).

Chebyshev case

In the Chebyshev case, an appropriate preconditioner for the inner product $b_{N,\omega}(u_N, v_N)$ in $X_N \times X_N$ is $(u'_N, \omega^{-1}(v_N \omega)')_\omega$ for which the associated linear system can be solved in $O(N)$ flops as shown in Section 4.1. Unfortunately, we do not have an estimate similar to (4.4.14) since no coercivity result for $b_{N,\omega}(u_N, v_N)$ is available to the authors' knowledge. However, ample numerical results indicate that the convergence rate of a conjugate gradient type method for non-symmetric systems such as Conjugate Gradient Square (CGS) or BI-CGSTAB is similar to that in the Legendre case.

The advantage of using the Chebyshev polynomials is of course that the evaluation of $B\bar{u}$ can be accelerated by FFT.

The preconditioning in the frequency space will be less effective if the coefficients $a(x)$ and $b(x)$ have large variations, since the variation of the coefficients is not taken into account in the construction of the preconditioner. One approach is to

4.4.2. Preconditioning in physical space. In this case, it is better to construct preconditioners using finite difference or finite element approximations for (4.4.1)–(4.1.4).

4.4.2.1. *Finite difference preconditioning for the collocation method.* The collocation method in the strong form for (4.4.1)–(4.1.4) is:

Find $u_N \in P_N$ such that

$$\begin{aligned} - (au'_N)'(x_j) + q(x_j)u_N(x_j) &= f(x_j), \quad 1 \leq j \leq N-1, \\ a_- u_N(-1) + b_- u'_N(-1) &= 0, \quad a_+ u_N(1) + b_+ u'_N(1) = 0. \end{aligned} \quad (4.4.16)$$

As in Section 4.2.2, (4.4.16) can be rewritten as a $(N-1) \times (N-1)$ linear system

$$A\bar{w} = \bar{f} \quad (4.4.17)$$

where the unknowns are $\{w_j = u_N(x_j)\}_{j=1}^{N-1}$, and

$$\bar{w} = (w_1, w_2, \dots, w_{N-1})^T, \quad \bar{f} = (f(x_1), f(x_2), \dots, f(x_{N-1}))^T. \quad (4.4.18)$$

The entries of A are given in Section 4.2.2.

As suggested by Orszag [17], we can build a preconditioner for A by using a finite difference approximation to (4.4.1)–(4.1.4). Let us denote

$$h_k = x_{k-1} - x_k, \quad \tilde{h}_k = \frac{1}{2}(x_{k-1} - x_{k+1}), \quad a_{k+\frac{1}{2}} = a((x_{k+1} + x_k)/2). \quad (4.4.19)$$

Then, the second-order finite difference scheme for (4.4.1)–(4.1.4) with first-order one-side difference at the boundaries reads:

$$\begin{aligned} - \frac{a_{i-\frac{1}{2}}}{\tilde{h}_i h_i} w_{i-1} + \left(\frac{a_{i-\frac{1}{2}}}{\tilde{h}_i h_i} + \frac{a_{i+\frac{1}{2}}}{\tilde{h}_i h_{i+1}} \right) w_i - \frac{a_{i+\frac{1}{2}}}{\tilde{h}_i h_{i+1}} w_{i+1} \\ + h(x_i) u_i = f(x_i), \quad 1 \leq i \leq N-1, \\ a_- w_N + b_- \frac{w_{N-1} - w_N}{h_N} = 0, \quad a_+ w_0 + b_+ \frac{w_0 - w_1}{h_1} = 0. \end{aligned} \quad (4.4.20)$$

We can rewrite (4.4.20) in the following matrix form:

$$A_{fd}\bar{w} = \bar{f}. \quad (4.4.21)$$

where A_{fd} is a non-symmetric tridiagonal matrix.

It has been shown (cf. [17, 13, 14]) that in the Dirichlet case, A_{fd}^{-1} is an optimal preconditioner for A , but $\text{cond}(A_{fd}^{-1}A)$ deteriorates with other boundary conditions.

REMARK 4.4.4. The above discussion is valid for both the Legendre and Chebyshev collocation method.

4.4.2.2. *Finite element preconditioning for the collocation method.* A more robust preconditioner can be constructed by using a finite element approximation. Since the finite element method is always based on a variational formulation, it can only be used for the preconditioning of the collocation methods which can be cast into a variational formulation. Namely, the collocation method for the Dirichlet boundary conditions or the collocation method in the weak form for the general boundary conditions.

We consider first the treatment of the general boundary conditions. Let us denote

$$X_h = \{u \in H_*^1(I) : u|_{[x_{i+1}, x_i]} \in P_1, i = 0, 1, \dots, N-1\}. \quad (4.4.22)$$

Then, based on the variational formulation (4.4.4), the piecewise linear finite element approximation to (4.4.1)–(4.1.4) is:

Find $u_h \in X_h$ such that for all $v_h \in X_h$,

$$b_h(u_h, v_h) = \langle f, v_h \rangle_h, \quad (4.4.23)$$

where

$$\begin{aligned} b_h(u_h, v_h) := & \langle au'_h, v'_h \rangle_h + a(1)h_+u_h(1)v_h(1) \\ & - a(-1)h_-u_h(-1)v_h(-1) + \langle bu_h, v_h \rangle_h, \end{aligned}$$

and $\langle \cdot, \cdot \rangle_h$ is an appropriate discrete inner product associated with the piecewise linear finite element approximation.

To fix the idea, we assume $b_{\pm} \neq 0$. Let us denote for $k = 1, 2, \dots, N-1$,

$$\hat{h}_k(x) = \begin{cases} \frac{x-x_{k+1}}{x_k-x_{k+1}}, & x \in [x_{k+1}, x_k] \\ \frac{x_{k-1}-x}{x_{k-1}-x_k}, & x \in [x_k, x_{k-1}] \\ 0, & \text{otherwise} \end{cases}, \quad (4.4.24)$$

and

$$\hat{h}_0(x) = \begin{cases} \frac{x-x_1}{x_0-x_1}, & x \in [x_1, x_0] \\ 0, & \text{otherwise} \end{cases}, \quad \hat{h}_N(x) = \begin{cases} \frac{x_{N-1}-x}{x_{N-1}-x_N}, & x \in [x_N, x_{N-1}] \\ 0, & \text{otherwise} \end{cases}. \quad (4.4.25)$$

Then,

$$X_h = \text{span}\{\hat{h}_0, \hat{h}_1, \dots, \hat{h}_N\}. \quad (4.4.26)$$

We further denote

$$\begin{aligned} u_h(x) &= \sum_{k=0}^N u_h(x_k) \hat{h}_k(x), \quad \bar{w} = (u_h(x_0), u_h(x_1), \dots, u_h(x_N))^T \\ b_{kj} &= b_h(\hat{h}_j, \hat{h}_k), \quad B_{fe} = (b_{kj})_{k,j=0,1,\dots,N}, \\ m_{kj} &= \langle \hat{h}_j, \hat{h}_k \rangle_h, \quad M_{fe} = (m_{kj})_{k,j=0,1,\dots,N}, \\ \bar{f} &= (f(x_0), f(x_1), \dots, f(x_N))^T. \end{aligned} \quad (4.4.27)$$

Then, (4.4.23) is equivalent to the following linear system

$$B_{fe} \bar{w} = M_{fe} \bar{f} \quad \text{or} \quad M^{-1} B_{fe} \bar{w} = \bar{f}. \quad (4.4.28)$$

On the other hand, as in Section 4.3, we can formulate the linear system associated with the Legendre-collocation method in the weak form for (4.4.1)–(4.1.4):

$$A \bar{w} = W \bar{f} \quad \text{or} \quad W^{-1} A \bar{w} = \bar{f}. \quad (4.4.29)$$

Since both (4.4.29) and (4.4.28) provide approximate solutions to (4.4.1)–(4.1.4), it is expected that $M^{-1} B_{fe}$ is a good preconditioner for $W^{-1} A$.

4.5. Error estimates

In this section, we present error analysis for two typical cases. One is the Legendre-Galerkin method for (4.1.5) with general boundary conditions (4.1.4). The other is the Chebyshev-collocation method for (4.1.3) with Dirichlet boundary conditions. The error analysis for other cases may be derived in a similar manner. We refer to the books [6, 5] for more details.

The following technical lemma is needed for the treatment of the general boundary conditions. Note that the special case of $b_{\pm} = 0$ in the following lemma is well known (see [6]).

LEMMA 4.7. *Let*

$$X = \{u \in H_{\omega}^2(I) : a_{\pm} u(\pm 1) + b_{\pm} u_x(\pm 1) = 0\}, \quad X_N = X \cap P_N.$$

We have

$$\inf_{\phi_N \in X_N} \|\phi - \phi_N\|_{H_{\omega}^{\nu}} \lesssim N^{\nu-m} \|\phi\|_{H_{\omega}^m}, \quad \forall \phi \in X \cap H_{\omega}^m(I), \quad m \geq 2, \quad \nu = 0, 1, 2.$$

PROOF. Let $H(1; x)$ be the Hermite polynomial (of degree 3) such that

$$H(1; 1) = 1, \quad H'(1; 1) = 0, \quad H(1; -1) = H'(1; -1) = 0,$$

and $\hat{H}(1; x)$ such that

$$\hat{H}(1; 1) = 0, \quad \hat{H}'(1; 1) = 1, \quad \hat{H}(1; -1) = \hat{H}'(1; -1) = 0.$$

Similarly, we define $H(-1; x)$ and $\hat{H}(-1; x)$. Then for $\phi \in X \cap H_{\omega}^m(I)$ with $m \geq 2$, we set

$$\begin{aligned} \tilde{\phi}(x) &= \phi(x) - \phi(1)H(1; x) - \phi(-1)H(-1; x) \\ &\quad - \phi'(1)\hat{H}(1; x) - \phi'(-1)\hat{H}(-1; x). \end{aligned} \quad (4.5.1)$$

By construction, we have

$$\tilde{\phi}(\pm 1) = \tilde{\phi}'(\pm 1) = 0 \quad \text{and consequently} \quad \tilde{\phi} \in H_{0,\omega}^2(I).$$

Furthermore, we derive from (4.5.1) that

$$\begin{aligned} \|\tilde{\phi}\|_{H_\omega^m} &\lesssim \|\phi\|_{H_\omega^m} + |\phi(1)| + |\phi(-1)| + |\phi'(1)| + |\phi'(-1)| \\ &\lesssim \|\phi\|_{H_\omega^m} \quad \forall m \geq 2. \end{aligned} \quad (4.5.2)$$

Let $\pi_{N,\omega}^{2,0}$ be the orthogonal projector from $H_0^2(I)$ onto $H_0^2(I) \cap P_N$ defined by

$$((u - \pi_{N,\omega}^{2,0}u)'', v'')_\omega = 0, \quad \text{for all } v \in H_{0,\omega}^2(I). \quad (4.5.3)$$

It is proven (see [3]) that for $\nu = 0, 1, 2$, we have

$$\|\tilde{\phi} - \pi_{N,\omega}^{2,0}\tilde{\phi}\|_{H_\omega^\nu} \lesssim N^{\nu-m} \|\tilde{\phi}\|_{H_\omega^m}, \quad \forall \tilde{\phi} \in H_{0,\omega}^2(I) \cap H_\omega^m(I), \quad m \geq 2. \quad (4.5.4)$$

We now define an operator $B_N : H_\omega^2(I) \rightarrow P_N$ by

$$\begin{aligned} B_N\phi &= \pi_{N,\omega}^{2,0}\tilde{\phi} + \phi(1)H(1; x) + \phi(-1)H(-1; x) \\ &\quad + \phi'(1)\hat{H}(1; x) + \phi'(-1)\hat{H}(-1; x). \end{aligned} \quad (4.5.5)$$

Then by (4.5.1), we have

$$\begin{aligned} \phi - B_N\phi &= \tilde{\phi} - \pi_{N,\omega}^{2,0}\tilde{\phi}, \\ (B_N\phi)(\pm 1) &= \phi(\pm 1), \quad (B_N\phi)'(\pm 1) = \phi'(\pm 1). \end{aligned}$$

Hence, $B_N\phi \in X_N$. Therefore, thanks to (4.5.2) and (4.5.4), we have

$$\begin{aligned} \inf_{\phi_N \in X_N} \|\phi - \phi_N\|_{H_\omega^\nu} &\leq \|\phi - B_N\phi\|_{H_\omega^\nu} = \|\tilde{\phi} - \pi_{N,\omega}^{2,0}\tilde{\phi}\|_{H_\omega^\nu} \\ &\lesssim N^{\nu-m} \|\tilde{\phi}\|_{H_\omega^m} \lesssim N^{\nu-m} \|\phi\|_{H_\omega^m}, \quad \forall m \geq 2. \end{aligned} \quad (4.5.6)$$

□

4.5.1. Legendre-Galerkin method with general boundary conditions. We consider the Legendre-Galerkin approximation of (4.1.5) with the Robin boundary condition (4.1.4).

THEOREM 4.1. *Let u and u_N be respectively the solutions of (4.1.5)–(4.1.4) and (4.1.14). Then,*

$$\|u - u_N\|_{H^1} \lesssim N^{1-m} \|u\|_{H^m} + N^{-k} \|f\|_{H^k}, \quad m \geq 2, \quad k \geq 1.$$

PROOF. We derive from (4.1.5) and (4.1.14) that

$$\alpha(u - u_N, v_N) - ((u - u_N)'', v_N) = (f - I_N f, v_N), \quad \text{for all } v_N \in X_N.$$

Since for all $v, w \in H^2(I) \cap H_\star^1(I)$,

$$-(v'', w) = (v', w') + h_+ v(1)w(1) - h_- v(-1)w(-1), \quad (4.5.7)$$

where $H_\star^1(I)$ and h_\pm are defined in (4.1.6) and (4.1.7). Thanks to (4.0.3) and the assumption $a_\pm \geq 0$, we find that

$$\begin{aligned} - (v'', w) &\lesssim \|v\|_{H^1} \|w\|_{H^1}, \quad \text{for all } v, w \in H^2(I) \cap H_\star^1(I), \\ - (v'', v) &\geq (v', v'), \quad \text{for all } v \in H^2(I) \cap H_\star^1(I). \end{aligned} \quad (4.5.8)$$

Hence,

$$\begin{aligned} \min(\alpha, 1) \|u - u_N\|_{H^1}^2 &= \alpha(u - u_N, u - u_N) + ((u - u_N)', (u - u_N)') \\ &\lesssim \alpha(u - u_N, u - u_N) - ((u - u_N)'', u - u_N) \\ &= \alpha(u - u_N, u - B_N u) - ((u - u_N)'', u - B_N u) \\ &\quad + (f - I_N f, B_N u - u_N) \\ &\lesssim \alpha \|u - u_N\|_{L^2} \|u - B_N u\|_{L^2} + \|u - B_N u\|_{H^1} \|u - u_N\|_{H^1} \\ &\quad + \|f - I_N f\|_{L^2} (\|u - B_N u\|_{L^2} + \|u - u_N\|_{L^2}). \end{aligned}$$

Therefore, thanks to (4.5.6) and Theorem 3.10, we have

$$\begin{aligned} \|u - u_N\|_{H^1} &\lesssim \|u - B_N u\|_{H^1} + \|f - I_N f\|_{L^2} \\ &\lesssim N^{1-m} \|u\|_{H^m} + N^{-k} \|f\|_{H^k}. \end{aligned}$$

The proof of Theorem 4.1 is then complete. \square

4.5.2. Chebyshev-collocation method with Dirichlet boundary conditions. An essential element in the analysis of the Chebyshev method for the second-order equations with Dirichlet boundary conditions is to show the bilinear form

$$a_\omega(u, v) := (u_x, \omega^{-1}(v\omega)_x)_\omega = \int_{-1}^1 u_x (v\omega)_x dx \quad (4.5.9)$$

is continuous and coercive in $H_{0,\omega}^1(I) \times H_{0,\omega}^1(I)$. To this end, we need to establish first the following inequality of Hardy type:

LEMMA 4.8.

$$\int_{-1}^1 u^2 (1+x^2) \omega^5 dx \leq \int_{-1}^1 u_x^2 \omega dx, \quad \text{for all } u \in H_{0,\omega}^1(I), \quad (4.5.10)$$

where $\omega = (1-x^2)^{-\frac{1}{2}}$.

PROOF. For any $u \in H_{0,\omega}^1(I)$, we find by integration by parts that

$$\begin{aligned} 2 \int_{-1}^1 u_x u x \omega^3 dx &= \int_{-1}^1 (u^2)_x x \omega^3 dx = - \int_{-1}^1 u^2 (x\omega^3)_x dx \\ &= - \int_{-1}^1 u^2 (1+2x^2) \omega^5 dx. \end{aligned} \quad (4.5.11)$$

Hence,

$$\begin{aligned} 0 &\leq \int_{-1}^1 (u_x + ux\omega^2)^2 \omega dx = \int_{-1}^1 u_x^2 \omega dx + \int_{-1}^1 u^2 x^2 \omega^5 dx \\ &\quad + 2 \int_{-1}^1 u_x u x \omega^3 dx = \int_{-1}^1 u_x^2 \omega dx - \int_{-1}^1 u^2 (1+x^2) \omega^5 dx. \end{aligned}$$

□

LEMMA 4.9.

$$\begin{aligned} a_\omega(u, v) &\leq 2|u_x|_{L_\omega^2} |v_x|_{L_\omega^2}, \quad \text{for all } u, v \in H_{0,\omega}^1(I), \\ a_\omega(u, u) &\geq \frac{1}{4}|u_x|_{L_\omega^2}^2, \quad \text{for all } u \in H_{0,\omega}^1(I). \end{aligned}$$

PROOF. Using the Cauchy-Schwarz inequality, the identity $\omega_x = x\omega^3$ and (4.5.10), we have

$$\begin{aligned} a_\omega(u, v) &= \int_{-1}^1 u_x (v_x + vx\omega^2) \omega dx \\ &\leq |u_x|_{L_\omega^2} |v_x|_{L_\omega^2} + |u_x|_{L_\omega^2} \left(\int_{-1}^1 v^2 x^2 \omega^5 dx \right)^{\frac{1}{2}} \\ &\leq 2|u_x|_{L_\omega^2} |v_x|_{L_\omega^2}, \quad \text{for all } u, v \in H_{0,\omega}^1(I). \end{aligned}$$

On the other hand, thanks to (4.5.11) and (4.5.10), we find

$$\begin{aligned} a_\omega(u, u) &= \int_{-1}^1 u_x^2 \omega dx + \int_{-1}^1 uu_x x \omega^3 dx \\ &= |u_x|_{L_\omega^2}^2 - \frac{1}{2} \int_{-1}^1 u^2 (1+2x^2) \omega^5 dx \\ &\geq |u_x|_{L_\omega^2}^2 - \frac{3}{4} \int_{-1}^1 u^2 (1+x^2) \omega^5 dx \\ &\geq \frac{1}{4} |u_x|_{L_\omega^2}^2, \quad \text{for all } u \in H_{0,\omega}^1(I). \end{aligned}$$

□

Thanks to the above Lemma, we can define a new orthogonal projector from $H_{0,\omega}^1$ to P_N^0 based on the bilinear form $a_\omega(\cdot, \cdot)$.

DEFINITION 4.5.1. $\tilde{\pi}_{N,\omega}^{1,0} : H_{0,\omega}^1 \rightarrow P_N^0$ is defined by

$$a_\omega(u - \tilde{\pi}_{N,\omega}^{1,0} u, v) = \int_{-1}^1 (u - \tilde{\pi}_{N,\omega}^{1,0} u)' (v\omega)' dx = 0, \quad \text{for } v \in P_N^0. \quad (4.5.12)$$

LEMMA 4.10.

$$\|u - \tilde{\pi}_{N,\omega}^{1,0} u\|_{H_\omega^\nu} \lesssim N^{\nu-m} \|u\|_{H_\omega^m} \quad \text{for all } u \in H_\omega^m(I) \cap H_{0,\omega}^1(I), \quad \nu = 0, 1. \quad (4.5.13)$$

PROOF. We recall that a different orthogonal projector $\pi_{N,\omega}^{1,0}$ was defined in (3.6.11). Using (4.5.10), the definition (4.5.12) and Theorem 3.9, we find

$$\begin{aligned} \|u - \tilde{\pi}_{N,\omega}^{1,0} u\|_{H_\omega^1}^2 &\lesssim |u - \tilde{\pi}_{N,\omega}^{1,0} u|_{H_\omega^1}^2 \leq a_\omega(u - \tilde{\pi}_{N,\omega}^{1,0} u, u - \tilde{\pi}_{N,\omega}^{1,0} u) \\ &= a_\omega(u - \tilde{\pi}_{N,\omega}^{1,0} u, u - \pi_{N,\omega}^{1,0} u) \\ &\leq 2|u - \tilde{\pi}_{N,\omega}^{1,0} u|_{H_\omega^1} |u - \pi_{N,\omega}^{1,0} u|_{H_\omega^1}. \end{aligned}$$

We then derive (4.5.13) with $\nu = 1$ from above and Theorem 3.9. To prove the result with $\nu = 0$, we use a standard duality argument. We consider the dual problem

$$-\phi_{xx} = u - \tilde{\pi}_{N,\omega}^{1,0} u, \quad \phi(\pm 1) = 0. \quad (4.5.14)$$

Its variational formulation is: find $\phi \in H_{0,\omega}^1(I)$ such that

$$(\phi', (\psi\omega)') = (u - \tilde{\pi}_{N,\omega}^{1,0} u, \psi\omega) \quad \forall \psi \in H_{0,\omega}^1(I). \quad (4.5.15)$$

Thanks to Lemma 4.9, there exists a unique solution $\phi \in H_{0,\omega}^1(I)$ for the above problem, and furthermore, we derive from (4.5.14) that $\phi \in H_\omega^2(I)$ and

$$\|\phi\|_{H_\omega^2} \lesssim \|u - \tilde{\pi}_{N,\omega}^{1,0} u\|_{L_\omega^2}. \quad (4.5.16)$$

Now we take $\psi = u - \tilde{\pi}_{N,\omega}^{1,0} u$ in (4.5.15), thanks to Lemma 4.9, (4.5.16) and (4.5.13) with $\nu = 1$,

$$\begin{aligned} (u - \tilde{\pi}_{N,\omega}^{1,0} u, u - \tilde{\pi}_{N,\omega}^{1,0} u)_\omega &= \int_{-1}^1 \phi_x ((u - \tilde{\pi}_{N,\omega}^{1,0} u)\omega)_x dx \\ &= \int_{-1}^1 (\phi - \tilde{\pi}_{N,\omega}^{1,0} \phi)_x ((u - \tilde{\pi}_{N,\omega}^{1,0} u)\omega)_x dx \\ &\leq 2|\phi - \tilde{\pi}_{N,\omega}^{1,0} \phi|_{H_\omega^1} |u - \tilde{\pi}_{N,\omega}^{1,0} u|_{H_\omega^1} \\ &\lesssim N^{-1} \|\phi\|_{H_\omega^2} |u - \tilde{\pi}_{N,\omega}^{1,0} u|_{H_\omega^1} \\ &\lesssim N^{-1} \|u - \tilde{\pi}_{N,\omega}^{1,0} u\|_{L_\omega^2} |u - \tilde{\pi}_{N,\omega}^{1,0} u|_{H_\omega^1}. \end{aligned}$$

We conclude from the above and (4.5.13) with $\nu = 1$. \square

We are now in the position to establish an error estimate of the Chebyshev-collocation method for

$$\alpha u - u_{xx} = f, \quad x \in (-1, 1); \quad u(\pm 1) = 0. \quad (4.5.17)$$

THEOREM 4.2. *Let u_N be the solution of the Chebyshev-collocation approximation to (4.5.17). Then,*

$$|u - u_N|_{H_\omega^1} \lesssim N^{1-m} \|u\|_{H_\omega^m} + N^{-k} \|f\|_{H_\omega^k}, \quad m \geq 2, \quad k \geq 1.$$

PROOF. Thanks to (4.2.6) and (4.2.7), the Chebyshev-collocation method for (4.5.17) can be written as:

Find $u_N \in P_N^0$ such that

$$\alpha \langle u_N, v_N \rangle_{N,\omega} + a_\omega(u_N, v_N) = \langle f, v_N \rangle_{N,\omega}, \quad \text{for all } v_N \in P_N^0.$$

On the other hand, we derive from (4.5.17) and (4.5.12) that

$$\alpha(u, v_N)_\omega + a_\omega(\tilde{\pi}_{N,\omega}^{1,0}u, v_N) = (f, v_N)_\omega, \quad \text{for all } v_N \in P_N^0.$$

Taking the difference of the above two relations and using the definition (3.6.6) and Theorem 3.8, we find that for all $v_N \in P_N^0$,

$$\begin{aligned} & \alpha\langle \tilde{\pi}_{N,\omega}^{1,0}u - u_N, v_N \rangle_{N,\omega} + a_\omega(\tilde{\pi}_{N,\omega}^{1,0}u - u_N, v_N) \\ &= (f, v_N)_\omega - \langle I_N f, v \rangle_{N,\omega} + \alpha\langle \tilde{\pi}_{N,\omega}^{1,0}u, v_N \rangle_{N,\omega} - \alpha(u, v_N)_\omega \\ &= (f - \pi_{N-1,\omega}f, v_N)_\omega - \langle I_N f - \pi_{N-1,\omega}f, v \rangle_{N,\omega} \\ & \quad + \alpha\langle \tilde{\pi}_{N,\omega}^{1,0}u - \pi_{N-1,\omega}u, v_N \rangle_{N,\omega} - \alpha(u - \pi_{N-1,\omega}u, v_N)_\omega. \end{aligned}$$

Take $v_N = \tilde{\pi}_{N,\omega}^{1,0}u - u_N$ in the above relation, we find by using Cauchy-Schwarz inequality and Lemma 3.10 that

$$\begin{aligned} & \alpha\|\tilde{\pi}_{N,\omega}^{1,0}u - u_N\|_{L_\omega^2}^2 + |\tilde{\pi}_{N,\omega}^{1,0}u - u_N|_{H_\omega^1}^2 \lesssim \|f - \pi_{N-1,\omega}f\|_{L_\omega^2}^2 \\ & \quad + \|f - I_N f\|_{L_\omega^2}^2 + \alpha\|u - \tilde{\pi}_{N,\omega}^{1,0}u\|_{L_\omega^2}^2 + \|u - \pi_{N-1,\omega}u\|_{L_\omega^2}^2. \end{aligned}$$

We then derive from the above and Theorems 3.8, 3.10 and Lemma 4.5.12 that

$$\begin{aligned} \|u - u_N\|_{H_\omega^1} &\leq \|u - \tilde{\pi}_{N,\omega}^{1,0}u\|_{H_\omega^1} + \|\tilde{\pi}_{N,\omega}^{1,0}u - u_N\|_{H_\omega^1} \\ &\lesssim N^{1-m}\|u\|_{H_\omega^m} + N^{-k}\|f\|_{H_\omega^k}. \end{aligned}$$

□

Bibliography

- [1] R.A. Adams. *Soblov Spaces*. Acadmic Press, New York, 1975.
- [2] B. K. Alpert and V. Rokhlin. A fast algorithm for the evaluation of Legendre expansions. *SIAM J. Sci. Stat. Comput.*, 12:158–179, 1991.
- [3] C. Bernardi and Y. Maday. Properties of some weighted sobolev spaces, and applications to spectral approximations. *SIAM J. Numer. Anal.*, 26:769–829, 1989.
- [4] C. Bernardi and Y. Maday. *Approximations Spectrales de Problèmes aux Limites Elliptiques*. Springer-Verlag, Paris, 1992.
- [5] C. Bernardi and Y. Maday. Spectral method. In P. G. Ciarlet and L. L. Lions, editors, *Handbook of Numerical Analysis, V. 5 (Part 2)*. North-Holland, 1997.
- [6] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1987.
- [7] C. Canuto and A. Quarteroni. Variational methds in the theoretical analysis of spectral approximations. In R.G. Voigt, D. Gottlieb, and M.Y. Hussaini, editors, *Spectral Methods for Partial Differential Equations*, pages 55–78. SIAM, 1994.
- [8] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.
- [9] Philip J. Davis and Philip Rabinowitz. *Methods of numerical integration*. Computer Science and Applied Mathematics. Academic Press Inc., Orlando, FL, second edition, 1984.
- [10] D. Funaro. *Polynomial Approxiamtions of Differential Equations*. Springer-verlag, 1992.
- [11] D. Gottlieb and S. A. Orszag. *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM-CBMS, Philadelphia, 1977.
- [12] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.
- [13] P. Haldenwang, G. Labrosse, S. Abboudi, and M. Deville. Chebyshev 3-d spectral and 2-d pseudospectral solvers for the helmholtz equation. *J. Comput. Phys.*, 55:115–128, 1984.
- [14] S. D. Kim and S. V. Parter. Preconditioning Chebyshev spectral collocation by finite difference operators. *SIAM J. Numer. Anal.*, 34, No. 3:939–958, 1997.
- [15] D. Kincaid and E. W. Cheney. *Numerical Analysis, Mathematics of Scientific Computing*. Brooks/Cole, 2nd edition, 1996.
- [16] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser, Basel, 2nd edition, 1992.
- [17] S. A. Orszag. Spectral methods for complex geometries. *J. Comput. Phys.*, 37:70–92, 1980.
- [18] R. D. Richtmyper and K. W. Morton. *Difference Methods for Initial-Value Problems*. Interscience, New York, 2nd edition, 1967.
- [19] T. J. Rivlin. *The Chebyshev Polynomials*. Jong Wiley and Sons, 1974.
- [20] Jie Shen. Efficient spectral-Galerkin method II. direct solvers for second- and fourth-order equations by using Chebyshev polynomials. *SIAM J. Sci. Comput.*, 16:74–87, 1995.

- [21] Jie Shen. Efficient Chebyshev-Legendre Galerkin methods for elliptic problems. In A. V. Ilin and R. Scott, editors, *Proceedings of ICOSAHOM'95*, pages 233–240. Houston J. Math., 1996.