SIAM J. SCI. COMPUT.
Vol. 15, No. 6, pp. 1489–1505, November 1994

013

Efficient Spectral-Galerkin Method I. Direct Solvers for the Second and Fourth Order Equations Using Legendre Polynomials*

JIE SHEN†

Abstract. We present some efficient algorithms based on the Legendre-Galerkin approximations for the direct solution of the second and fourth order elliptic equations. The key to the efficiency of our algorithms is to construct appropriate base functions, which lead to systems with sparse matrices for the discrete variational formulations. The complexities of the algorithms are a small multiple of N^{d+1} operations for a d dimensional domain with $(N-1)^d$ unknowns, while the convergence rates of the algorithms are exponential for problems with smooth solutions. In addition, the algorithms can be effectively parallelized since the bottlenecks of the algorithms are matrix-matrix multiplications.

Key words. spectral-Galerkin method, Legendre polynomial, Helmholtz equation, biharmonic equation, direct solver

AMS subject classifications. 65N35, 65N22, 65F05, 35J05

1. Introduction.

This article is the first in a series for developing efficient spectral Galerkin algorithms for elliptic problems. The spectral method employs global polynomials as the trial functions for the discretization of partial differential equations. It provides very accurate approximations with a relatively small number of unknowns. Consequently it has gained increasing popularity in the last two decades, especially in the field of computational fluid dynamics (see [11], [8] and the references therein).

The use of different test functions in a variational formulation leads to three most commonly used spectral schemes, namely, the Galerkin, tau and collocation versions. In the collocation method, we work in the physical space – a set of collocation points; while in the Galerkin and tau methods, we work in the spectral space – the coefficients of the polynomial series. The Galerkin and collocation methods usually lead to optimal error estimates; while the tau method, being a modification of the Galerkin method, leads to non symmetric variational formulations and only sub-optimal error estimates are available (see for instance [19] and [20]). Gottlieb and Orszag in their pioneer book [11] presented an efficient Chebyshev-tau method; on the other hand, they presented a basis for the Galerkin method which leads to full matrices and its application in practice is prohibited. It is surprising that virtually no effort has been made on constructing appropriate bases (other than the Lagrangian interpolant basis) for the spectral-Galerkin method. Consequently the tau method along with the collocation method (the later being more natural for problems with variable coefficients) have been the focus of a great number of research papers (see [8] and the reference therein), while the Galerkin method, being more authentic and more accurate than the tau method, has draw less attention. We should point out that the spectral element method, developed by Patera and his group, is in fact a spectral-Galerkin method (see the survey paper [13]). However, the spectral element method, in the case of a single domain, differs from the spectral-Galerkin method to be presented in this work in two aspects: (i) a Gaussian quadrature formula is used instead of exact integration;

* Received by the editors Junuary 20, 1993; accepted for publication (in revised form) September 29, 1993. This work was supported in part by NSF grant DMS-9205300.

† Department of Mathematics, Penn State University, University Park, Pennsylvania 16802 (shen_j@math.psu.edu).

(ii) a Lagrangian interpolant basis is used instead of the basis by simple combinations of Legendre polynomials.

We shall present in this article appropriate bases for the Legendre Galerkin method applied to second order and fourth order elliptic equations with various boundary conditions. The resulting discrete systems have sparse matrices similar to the ones by finite difference discretizations. We shall develop efficient direct solution techniques for solving these discrete systems.

For the second order equations, the complexity of our algorithm is a small multiple of N^{d+1} operations for a d dimensional domain with $(N-1)^d$ unknowns. Since its pre-processing time is negligible, it is definitely the method of choice among the existing spectral algorithms for solving one particular equations. For solving equations with multiple right-hand sides, it is as efficient as and more accurate than the Legendre-tau method; and it is also very competitive to the Chebyshev-tau and Chebyshev-collocation methods. Furthermore, the roundoff errors in the Legendre-Galerkin approximation are considerably less appreciable than that in other spectral approximations.

Although the solution techniques for fourth order equations by finite difference methods are well developed (see [5] for a direct solver and [3] for a more efficient iterative solver), there are not many results available for the fourth order equations by spectral methods, which are more difficult to analyze and to solve numerically. We mention here the theoretical work in [10] and [2] for one dimensional fourth order equations, and in [1] for two dimensional fourth order equations. However how to efficiently solve the resulting discrete systems is not clear, and only sub-optimal error estimates are available for the collocation methods presented in [1].

We shall develop an efficient direct solver for the fourth order equations based on the Legendre-Galerkin method. In the two dimensional case, it has a complexity of a small multiple of $O(N^3)$ operations. In other words, we can solve fourth order equations with an amount of work comparable to that of solving second order equations. Hence, our algorithm should be clearly more efficient than the existing spectral methods for fourth order equations (see for instance [24], [1]). Furthermore, it is clear that the Legendre-Galerkin approximation to the fourth order equations leads to optimal error estimates. One of the application of this algorithm is for solving the two dimensional Stokes equations, since the stream function formulation of the 2-D Stokes equations is the 2-D biharmonic equation with Dirichlet boundary conditions. Let us mention that the bases we developed here are also suitable for decomposing the solution into a high frequency part and a low frequency part as is required in the implementation of the Nonlinear Galerkin Method [22].

A drawback of using Legendre polynomials is of course the lack of a fast transform between the physical space and the spectral space. Hence, in the second part of the series [21], we shall study the spectral-Galerkin method by using the Chebyshev polynomials, for which the Fast Fourier Transform (FFT) is applicable. Let us mention that it is hopeful that an efficient transform method for the Legendre polynomials can be eventually developed, at least for N sufficiently large, by using the ideas presented in [17] and [4].

We note that the algorithms presented here are mostly appropriate for simple geometries and for constant coefficient problems. We shall briefly address how to treat the variable coefficient case and leave the extension to more complex geometries to a future investigation. We should mention that the complexities of our algorithms are not optimal with respect to the number of unknowns. However thanks to the spectral

accuracy and the very small constants in front of the $O(N^{d+1})$ operation counts, our algorithms should be very competitive, at least for smooth functions or N not too large, to the high-precision algorithms with optimal or near-optimal complexity (cf. [18], [15], [16]).

The remainder of the article is organized as follows: In the next section, we consider the second order equations. In Section 3, we study the fourth order equations. In Section 4, we point out some immediate extensions. Finally in Section 5, we present and compare some numerical results.

2. Second order equations. In this section, we are interested in solving the Helmholtz equation

$$\alpha u - \Delta u = f \ \text{ in } \ \Omega = I^d, \ \ u|_{\partial\Omega} = 0, \tag{2.1}$$

where $I = (-1, 1)$ and $d = 1$, 2 or 3, by using the Legendre Galerkin method. The extension to more general problems will be discussed in Section 4.

Let us first introduce some basic notations which will be used in the sequel. We denote by $L_n(x)$ the nth degree Legendre polynomial, and we set

$$S_N = \ \text{span}\{L_0(x),\ L_1(x),\ \ldots,\ L_N(x)\}, \ \ V_N = \{v \in S_N :\ v(\pm 1) = 0\}.$$

Then the standard Legendre-Galerkin approximation to (2.1) is:
Find $u_{_N} \in V_N^d$ such that

$$\alpha(u_{_N}, v) + (\nabla u_{_N}, \nabla v) = (f, v),\ \forall\ \boldsymbol{v} \in V_N^d, \tag{2.2}$$

where $(u, v) = \int_\Omega uvd\boldsymbol{x}$ is the scalar product in $L^2(\Omega)$, and its norm will be denoted by $\|\cdot\|$. The approximation property of (2.2) is best described by using the Sobolev spaces. Let us denote $H^s(\Omega)$ to be the usual Sobolev spaces with the norm $\|v\|_s$. It is well known (cf. [8]) that for $\alpha \geq 0$, $s \geq 1$ and $u \in H^s(\Omega)$, the following optimal error estimates holds:

$$\|u - u_{_N}\| + N\|u - u_{_N}\|_1 \leq C(s)N^{-s}\|u\|_s. \tag{2.3}$$

Although the approximation (2.2) is highly accurate, its practical value depends on the choice of a basis for V_N^d.

2.1. One dimensional case. The crucial task is to choose an appropriate basis for V_N such that the linear system resulting from (2.2) is as simple as possible. Let us mention that the Lagrangian interpolant (or nodal) basis used in the collocation and spectral element formulations, and the basis

$$V_N = \text{span}\{\phi_2(x), \phi_3(x), \cdots, \phi_N(x)\}$$

with

$$\phi_k(x) = \begin{cases} L_k(x) - L_0(x), & k \text{ even} \\ L_k(x) - L_1(x), & k \text{ odd} \end{cases}; \ \text{ or } \ \phi_k(x) = (1 - x^2)L_{k-2}(x)$$

all lead to linear systems with full matrices even in the simplest case $\alpha = 0$. However a clever choice of the basis would lead to a linear system with a sparse matrix.

We recall that the $\{L_n(x)\}$ satisfy the orthogonality relation

$$(L_k(x), L_j(x)) = \frac{2}{2k+1}\delta_{ij},\ \forall\ k, j \geq 0, \tag{2.4}$$

and the recurrence relation

$$(2k+1)L_k(x) = L'_{k+1}(x) - L'_{k-1}(x). \tag{2.5}$$

We recall also that $L_n(x)$ is a polynomial of degree n and therefore $L_n''(x) \in S_{n-2}$, more precisely

$$L_n''(x) = \sum_{\substack{k=0 \\ k+n \text{ even}}}^{n-2} (k+\frac{1}{2})[n(n+1)-k(k+1)]L_k(x). \tag{2.6}$$

The following lemma is the key to the efficiency of our algorithms.

LEMMA 2.1. *Let us denote*

$$c_k = \frac{1}{\sqrt{4k+6}}, \quad \phi_k(x) = c_k(L_k(x) - L_{k+2}(x)), \tag{2.7}$$
$$a_{jk} = (\phi_k'(x), \phi_j'(x)), \quad b_{jk} = (\phi_k(x), \phi_j(x)),$$

then

$$a_{jk} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases}, \quad b_{jk} = b_{kj} = \begin{cases} c_k c_j(\frac{2}{2j+1} + \frac{2}{2j+5}), & k = j \\ -c_k c_j \frac{2}{2k+1}, & k = j+2 \\ 0, & \textit{Otherwise} \end{cases}, \tag{2.8}$$

and

$$V_N = \textit{span}\{\phi_0(x), \phi_1(x), \cdots, \phi_{N-2}(x)\}.$$

Proof. Since $L_k(\pm 1) = (\pm 1)^k$, it follows that $\phi_k(x) \in V_N$ for $k = 0, 1, \cdots, N-2$. On the other hand, it is clear that $\{\phi_k(x)\}$ are linear independent and that $\dim(V_N) = N-1$. Hence,

$$V_N = \text{span}\{\phi_0(x), \phi_1(x), \cdots, \phi_{N-2}(x)\}.$$

Notice that

$$a_{jk} = (\phi_k'(x), \phi_j'(x)) = -(\phi_k''(x), \phi_j(x)) = -(\phi_k(x), \phi_j''(x)),$$

by using (2.4)- (2.5), one can easily derive (2.8). □

REMARK 2.1. *It is transparent that base functions similar to that in*(2.7) *can be constructed by using Chebyshev or other Jacoby polynomials. However, whether they would also lead to efficient algorithms needs more investigations due to the non-uniform weight in their orthogonal relations*(*see [21]*).

In the following, we shall use capital letters to denote matrices or two dimensional arrays, and bold face letters to denote column vectors.

It is now clear that (2.2) (with $d = 1$) is equivalent to

$$\alpha(u_N, \phi_k(x)) + (u_N', \phi_k'(x)) = (f, \phi_k(x)), k = 0, 1, \cdots, N-2. \tag{2.9}$$

Let us denote

$$f_k = (f, \phi_k(x)), \ \boldsymbol{f} = (f_0, f_1, \cdots, f_{N-2})^T;$$

$$u_N = \sum_{n=0}^{N-2} v_n \phi_n(x), \ \boldsymbol{v} = (v_0, v_1, \cdots, v_{N-2})^T, \text{ and } B = (b_{kj})_{0 \leq k,j \leq N-2}. \tag{2.10}$$

Then (2.9) is equivalent to the following matrix equation:

$$(\alpha B + I)\boldsymbol{v} = \boldsymbol{f}. \tag{2.11}$$

Since $b_{kj} = 0$ for $k \neq j$ and $k \neq j \pm 2$, we observe that B (resp. The system (2.11) with $\alpha \neq 0$) can be decoupled into two tridiagonal submatrices (resp. two tridiagonal

subsystems for the odd and even components of $\boldsymbol{v}$). Notice also that the system (2.11) reduces to a diagonal system for $\alpha = 0$.

REMARK 2.2. *The equation*

$$\alpha u + \beta u_x - u_{xx} = f, \quad u(\pm 1) = 0 \tag{2.12}$$

can be easily handled as well. In fact let $c_{jk} = (D_x\phi_k(x), \phi_j(x))$ *and* $C = (c_{kj})$. *A simple computation leads to*

$$c_{jk} = -c_{kj} = \begin{cases} 2c_k c_j, & k = j+1 \\ 0, & \textit{Otherwise} \end{cases}. \tag{2.13}$$

Hence, the discrete system corresponding to(2.12) *is*:

$$(\alpha B + \beta C + I)\boldsymbol{v} = \boldsymbol{f},$$

which is simply a pentadiagonal system.

2.2. Two dimensional case. It is clear that

$$V_N^2 = \text{span}\{\phi_i(x)\phi_j(y) : i, j = 0, 1, \cdots, N-2\}.$$

Let us denote

$$u_{_N} = \sum_{k,j=0}^{N-2} u_{kj}\phi_k(x)\phi_j(y), \quad f_{kj} = (f, \phi_k(x)\phi_j(y)),$$

and

$$U = (u_{kj})_{k,j=0,1,\cdots,N-2}, \; F = (f_{kj})_{k,j=0,1,\cdots,N-2}.$$

Taking $v = \phi_l(x)\phi_m(y)$ in (2.2) for $l, m = 0, 1\cdots, N-2$, we find that (2.2) is equivalent to the following matrix equation:

$$\alpha BUB + UB + BU = F, \tag{2.14}$$

where B are the matrices defined in (2.10).

This equation can be solved in particular by the matrix decomposition method described in [7], which is better known in the field of spectral methods as the matrix diagonalization method [12]. To this end, let Λ be the diagonal matrix whose diagonal entries $\{\lambda_p\}$ are the eigenvalues of B, and let E be the orthonormal matrix formed by the eigenvectors of B (since B is symmetric), i.e. $E^T BE = \Lambda$. Now setting $U = EV$, the equation (2.14) becomes

$$\alpha E\Lambda VB + EVB + E\Lambda V = F.$$

Multiply E^T to the above equation, we arrive to

$$\alpha\Lambda VB + VB + \Lambda V = E^T F \equiv G.$$

The transpose of the above equation reads

$$\alpha BV^T\Lambda + BV^T + V^T\Lambda = G^T. \tag{2.15}$$

Let $\boldsymbol{v}_p = (v_{p0}, v_{p1}, \cdots, v_{pN-2})^T$ and $\boldsymbol{g}_p = (g_{p0}, g_{p1}, \cdots, g_{pN-2})^T$ for $p = 0, 1, \cdots, N-2$. Then the pth column of the equation (2.15) can be written as:

$$((\alpha\lambda_p + 1)B + \lambda_p I)\,\boldsymbol{v}_p = \boldsymbol{g}_p, \quad p = 0, 1, \cdots, N-2, \tag{2.16}$$

which is equivalent to $N-1$ one dimensional equations of the form (2.11).

In summary, the solution of (2.14) consists of the following steps:

(0) Pre-processing: compute the eigenvalues and eigenvectors of B;
(1) Compute $G = E^T F$;
(2) Obtain V by solving (2.16);
(3) Set $U = EV$.

since B can be split into two symmetric tridiagonal submatrices, the eigenvalues and eigenvectors of B can be easily computed in $O(N^2)$ operations. The step 2 consists of solving $N-1$ tridiagonal systems of order $N-1$. From the structure of B, we realize that $e_{kj} = 0$ for $k+j$ odd. Hence the amount of work for the matrix multiplications can be reduced by a factor of 2. Consequently the steps 1 and 3 take a total of $2(N-1)^3$ arithmetic operations. Therefore, for each right-hand side, the system (2.14) can be solved in about $2N^3$ operations.

We remark that for each particular solution of (2.1), the Legendre Galerkin method described above takes at most the same amount of operations as the Legendre-tau method. However thanks to the symmetry of the Galerkin procedure, the complexity of the pre-processing stage in the Galerkin case is an order of magnitude less (in terms of N) than in the tau (and the collocation) case. Furthermore the error estimates of the Galerkin approximation is optimal (see (2.3)) while that of the tau approximation is not (see for instance [20] and [19]).

REMARK 2.3. *As in Remark 2.2, the equation*(2.1) *with a first order term* βu_y *can be efficiently treated. In this case, the discrete equation*(2.14) *should be replaced by*

$$\alpha BUB + \beta BUC + UB + BU = F,$$

which can still be solved by the matrix diagonalization method. By the same token, we can treat the equation(2.1) *with a first order term* βu_x.

REMARK 2.4. *We note that the bottleneck of the algorithm is the two matrix-matrix multiplications in steps 1 and 3, which can be effectively parallelized and its complexity can be improved to* $O(N^{d+1}/p)$ *with p parallel processors. The same remark holds for other algorithms to be presented later in this paper.*

2.3. Three dimensional case. The three dimensional Helmholtz equation can also be efficiently solved by the above matrix diagonalization method. However, the formulation of the algorithm requires a little extra care. Let us denote

$$u_{_N} = \sum_{n,m,l=0}^{N-2} u_{nml}\phi_n(x)\phi_m(y)\phi_l(z), \quad f_{ijk} = (f, \phi_i(x)\phi_j(y)\phi_k(z)).$$

since

$$V_N^3 = \text{span}\{\phi_i(x)\phi_j(y)\phi_k(z) : i,j,k = 0,1,\cdots,N-2\},$$

taking $v = \phi_i(x)\phi_j(y)\phi_k(z)$ in (2.2) for $i,j,k = 0,1\cdots,N-2$, we find that (2.2) with $d=3$ is equivalent to the following equation:

(2.17)
$$\alpha b_{in}u_{nml}b_{jm}b_{kl} + u_{iml}b_{jm}b_{kl} + b_{in}u_{njl}b_{kl} + b_{in}u_{nmk}b_{jm} = f_{ijk},\ i,j,k = 0,1,\cdots,N-2,$$

where we have used the conventional notation such that a pair of repeated index imply a summation of the index from 0 to $N-2$. Hence, by the definition of E and Λ in Section 2.2, we have

$$b_{in}e_{nq} = \lambda_q e_{iq},\ e_{iq}e_{ip} = \delta_{qp}.$$

Set $u_{nml} = e_{nq}v_{qml}$ and use the above relation, we can rewrite the equation (2.17) as

$$\alpha\lambda_q e_{iq}v_{qml}b_{jm}b_{kl} + e_{iq}v_{qml}b_{jm}b_{kl} + \lambda_q e_{iq}v_{qjl}b_{kl} + \lambda_q e_{iq}v_{nmk}b_{jm} = f_{ijk}, \quad i,j,k = 0,1,\cdots,N-2.$$

Multiply e_{ip} to the above equation, we obtain

$$(\alpha\lambda_p+1)v_{pml}b_{jm}b_{kl} + \lambda_p(v_{pjl}b_{kl} + v_{pmk}b_{jm}) = e_{ip}f_{ijk} \equiv g_{pjk},\ p,j,k = 0,1,\cdots,N-2.$$

Now set $V^p = (v_{pml})_{0\le m,l\le N-2}$ and $G^p = (g_{pml})_{0\le m,l\le N-2}$, we can rewrite the above equation as

$$(\alpha\lambda_p + 1)BV^pB + \lambda_p(V^pB + BV^p) = G^p,\ p = 0,1,\cdots,N-2. \tag{2.18}$$

For each p, the above equation corresponds to a two dimensional equation of the form (2.14).

In summary, the solution of (2.17) consists of the following steps:

(0) Pre-processing: compute the eigenvalues and eigenvectors of B;
(1) Compute $g_{pjk} = e_{ip}f_{ijk}$ for $p,j,k = 0,1,\cdots,N-2$;
(2) Obtain V^p by solving (2.18) for $p = 0,1,\cdots,N-2$;
(3) Set $u_{nml} = e_{nq}v_{qml}$ for $n,m,l = 0,1,\cdots,N-2$.

The step 2 consists of solving $N-1$ two-dimensional equations of the form (2.14). Hence, it takes about $2N^4$ operations. The steps 1 and 3 take $2N^4$ operations. Therefore, The system (2.17) can be solved in about $4N^4$ operations.

3. Fourth order equations. In this section, we consider the fourth order equation

$$\alpha u - \beta\Delta u + \Delta^2 u = f \ \text{ in } \Omega = I^d,\ u|_{\partial\Omega} = \frac{\partial u}{\partial \boldsymbol{n}}|_{\partial\Omega} = 0, \tag{3.1}$$

where $\boldsymbol{n}$ is the normal vector to $\partial\Omega$.

Let

$$W_N = \{v \in S_N : v(\pm 1) = v_x(\pm 1) = 0\}.$$

Then the Legendre-Galerkin approximation of (3.1) consists of finding $u_{_N} \in W_N^d$ such that

$$\alpha(u_{_N}, v) + \beta(\nabla u_{_N}, \nabla v) + (\Delta u_{_N}, \Delta v) = (f, v),\ \forall\, v \in W_N^d. \tag{3.2}$$

It can be shown that for $\alpha, \beta > 0$ and $u \in H^s(\Omega)\cap H_0^2(\Omega)$ for $s \ge 2$, then the following optimal error estimate holds:

$$\|u - u_{_N}\| + N\|u - u_{_N}\|_1 + N^2\|u - u_{_N}\|_2 \le C(s)N^{-s}\|u\|_s. \tag{3.3}$$

3.1. One dimensional case. The equation (3.1) in the one dimensional case can serve as a model for the *clamped rod* problem. It can also serve as a model for the 2-D Stokes equations in the primitive variable formulation or in the stream function formulation, subject to the periodic boundary condition in the y direction and Dirichlet boundary condition in the x direction. A semi-implicit time discretization of the important Kuramoto-Sivashinsky equation modeling a flame propagation ([23]) is also of the form (3.1) with $d = 1$.

LEMMA 3.1. *Let* $d_k = \dfrac{1}{\sqrt{2(2k+3)^2(2k+5)}}$ *and*

$$\psi_k(x) = d_k(L_k(x) - \frac{2(2k+5)}{2k+7}L_{k+2}(x) + \frac{2k+3}{2k+7}L_{k+4}(x)),\ k = 0,1,\cdots,N-4. \tag{3.4}$$

Then

$$W_N = span\{\psi_0(x), \psi_1(x), \cdots, \psi_{N-4}(x)\}.$$

Furthermore, we have

$$a_{kj} = (\psi_j''(x), \psi_k''(x)) = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases}, \tag{3.5}$$

and the only non zero elements of $b_{kj} = (\psi_j(x), \psi_k(x))$, $c_{kj} = (\psi_j'(x), \psi_k'(x))$ *are*:

$$\begin{aligned} b_{kk} =& d_k^2(e_k + h_k^2 e_{k+2} + g_k^2 e_{k+4}) \\ b_{kk+2} =& b_{k+2k} = d_k d_{k+2}(h_k e_{k+2} + g_k h_{k+2} e_{k+4}) \\ b_{kk+4} =& b_{k+4k} = d_k d_{k+4} g_k e_{k+4} \\ c_{kk} =& -2(2k+3)d_k^2 h_k \\ c_{kk+2} =& c_{k+2k} = -2(2k+3)d_k d_{k+2} \end{aligned} \quad , \tag{3.6}$$

where

$$e_k = \frac{2}{2k+1},\ g_k = \frac{2k+3}{2k+7},\ h_k = -(1+g_k).$$

Proof. Since $L_k(\pm 1) = (\pm 1)^k$ and $L_k'(\pm 1) = \frac{1}{2}(\pm 1)^{k-1}k(k+1)$, we can readily check that $\psi_k(x) \in W_N$. On the other hand it is clear that $\{\psi_i(x)\}$ are linear independent and the dimension of W_N is $N-3$. Hence

$$W_N = \text{span}\{\psi_0(x), \psi_1(x), \cdots, \psi_{N-4}(x)\}.$$

(3.5) and (3.6) can be derived by direct computations using (2.4)-(2.6) and integration by part. □

To simplify the notation, we denote hereafter $q = N-4$. Let us denote now

$$B = (b_{kj})_{0\le k,j\le q}\,,\ \ C = (c_{kj})_{0\le k,j\le q}\,,$$

$$f_k = (f, \psi_k(x)),\ \boldsymbol{f} = (f_0, f_1, \cdots, f_q)^T;$$

$$u_{_N} = \sum_{n=0}^{q} v_n \psi_n(x),\ \boldsymbol{v} = (v_0, v_1, \cdots, v_q)^T,$$

we find that (3.2) with $d = 1$ is equivalent to the following matrix equation:

$$(\alpha B + \beta C + I)\boldsymbol{v} = \boldsymbol{f}. \tag{3.7}$$

It is obvious that B and C are symmetric positive definite matrices. Furthermore, B can be split into two pentadiagonal submatrices and C can be split into two tridiagonal submatrices. Hence, the system can be efficiently solved.

3.2. Two dimensional case. The equation (3.1) in the two dimensional case with $\alpha = \beta = 0$ is the well-known biharmonic equation. It has many important applications. In particular, it is known as a model for the plate problem; also the steam-function of a flow governed by two dimensional Stokes equations is a solution of the equation (3.1). The case with $\alpha \neq 0$ is also important for time dependent problems.

It is obvious that

$$W_N^2 = \text{span}\{\psi_i(x)\psi_j(y) : i, j = 0, 1, \cdots, q\}.$$

Using the same notations as in Section 2.2 (with $\phi_k(x)$ replaced by $\psi_k(x)$, and $N-2$ replaced by q), taking $v = \psi_l(x)\psi_m(y)$ in (3.2) for $l, m = 0, 1, \cdots, q$, we find that (3.2) is equivalent to the following matrix equation:

$$\alpha BUB + \beta(CUB + BUC) + BU + 2CUC + UB = F. \tag{3.8}$$

We can also rewrite the above equation in the following form using the tensor product notation:

$$D\boldsymbol{u} \equiv (\alpha B \otimes B + \beta(C \otimes B + B \otimes C) + B \otimes I + 2C \otimes C + I \otimes B)\boldsymbol{u} = \boldsymbol{f}, \tag{3.9}$$

where $\boldsymbol{f}$ and $\boldsymbol{u}$ are F and U written in the form of a column vector, i.e.

$$\boldsymbol{f} = (f_{00}, f_{10}, \cdots, f_{q0}; f_{01} \cdots, f_{q1}; \cdots; f_{0q}, \cdots, f_{qq})^T, \tag{3.10}$$

and $\otimes$ denotes the tensor product of matrices, i.e. $A \otimes B = (Ab_{ij})_{i,j=0,1,\cdots,q}$.

Let us first remark that if $BC = CB$, then the equation (3.8) can be efficiently solved by using the matrix diagonalization method (see below). But unfortunately due to the fact that the equation (3.1) is not separable, we have $BC \neq CB$. However $BC - CB$ only has eight non zero entries at positions $(0,2), (1,3), (q-3, q-1), (q-2, q)$ and their symmetric counterpart $(2,0), (3,1), (q-1, q-3), (q, q-2)$. These non zero entries can be eliminated by modifying the following four entries of the matrix B:

$$\begin{aligned}
b_{00} &\longrightarrow \tilde{b}_{00} = (c_{00}b_{02} + c_{02}b_{22} - b_{02}c_{22} - b_{04}c_{42})/c_{02};\\
b_{11} &\longrightarrow \tilde{b}_{11} = (c_{11}b_{13} + c_{13}b_{33} - b_{13}c_{33} - b_{15}c_{53})/c_{13};\\
b_{q-1q-1} &\longrightarrow \tilde{b}_{q-1q-1} = (c_{q-1q-1}b_{q-1q-3} + c_{q-1q-3}b_{q-3q-3}\\
&\qquad - b_{q-1q-3}c_{q-3q-3} - b_{q-1N-9}c_{N-9q-3})/c_{q-1q-3};\\
b_{qq} &\longrightarrow \tilde{b}_{qq} = (c_{qq}b_{qq-2} + c_{qq-2}b_{q-2q-2} - b_{qq-2}c_{q-2q-2} - b_{qq-4}c_{q-4q-2})/c_{qq-2}.
\end{aligned}$$

Setting $\tilde{B} = (\tilde{b}_{ij})$ with $\tilde{b}_{ij} = b_{ij}$ for $(i,j) \neq (0,0), (1,1), (q-1, q-1), (q,q)$, we have

$$\tilde{B}C = C\tilde{B}.$$

Now let us explain how to efficiently solve the equation:

$$\alpha\tilde{B}UB + \beta(CUB + \tilde{B}UC) + \tilde{B}U + 2CUC + UB = F. \tag{3.11}$$

Since $\tilde{B}$ and C are symmetric and $\tilde{B}C = C\tilde{B}$, it is well known that there exists an orthonormal matrix E such that

$$\tilde{B}E = E\Lambda, \quad CE = E\Sigma, \tag{3.12}$$

and Λ and Σ are real diagonal matrices. The matrix E is the set of eigenvectors of $\tilde{B}$ and C, the diagonal entries $\{\lambda_p\}$ and $\{\sigma_p\}$ of Λ and Σ are respectively eigenvalues of $\tilde{B}$ and C. Hence, setting $U = EV$ in (3.8), using (3.12), we obtain:

$$\alpha E\Lambda VB + \beta E(\Sigma VB + \Lambda VC) + E\Lambda V + 2E\Sigma VC + EVB = F.$$

Multiplying E^T to the above equation, we get

$$\alpha\Lambda VB + \beta(\Sigma VB + \Lambda VC) + \Lambda V + 2\Sigma VC + VB = E^TF \equiv G. \tag{3.13}$$

As in Section 2.2, let $\boldsymbol{v}_p = (v_{p0}, v_{p1}, \cdots, v_{pq})^T$ and $\boldsymbol{g}_p = (g_{p0}, g_{p1}, \cdots, g_{pq})^T$ for $p = 0, 1, \cdots, q$. Then the pth row of the equation (3.13) is:

$$\{(\alpha\lambda_p + \beta\sigma_p + 1)B + (\beta\lambda_p + 2\sigma_p)C + \lambda_pI\}\,\boldsymbol{v}_p = \boldsymbol{g}_p, \quad p = 0, 1, \cdots, q. \tag{3.14}$$

For each p, the above equation is simply an one dimensional equation of the form (3.7).

In summary, the solution of (3.11) consists of the following steps:

(0) Pre-processing: compute the eigenvalues and eigenvectors Λ, Σ, E of $\tilde{B}$ and C;
(1) Compute $G = E^T F$;
(2) Obtain $\boldsymbol{v}_p$ by solving (3.14);
(3) Set $U = EV$.

By a classical result on tridiagonal matrices (see Section 3.7 of [14]), we derive that the eigenvalues of C are all distinct. Consequently an eigenvector of C is automatically an eigenvector of $\tilde{B}$. Therefore for the pre-processing stage, we only have to compute the eigenvalues and the eigenvectors Σ and E of C, Λ can then be determined by the relation $\tilde{B}E = E\Lambda$. Since C can be split into two tridiagonal submatrices, the preprocessing stage only takes $O(N^2)$ operations. As before the steps 1 through 3 take about $2N^3$ operations.

Now let us describe how to use the fast solver for (3.11) to solve the original system (3.8) or equivalently (3.9). To this end, we rewrite (3.11) by using the tensor product notation:

$$\tilde{D}u \equiv \left(\alpha\tilde{B}\otimes B + \beta(C\otimes B + \tilde{B}\otimes C) + \tilde{B}\otimes I + 2C\otimes C + I\otimes B\right)\boldsymbol{u} = \boldsymbol{f}. \tag{3.15}$$

it is easy to see that D and $\tilde{D}$ differ at only $4(q+1)$ rows, more precisely they differ at the rows: $(i-1)*q+1, (i-1)*q+2, i*q-1, i*q$ for $i = 1, 2\cdots q$. Following the idea in [6], we can use the fast solver for (3.15) to solve the equation (3.9) by the method of capacitance matrix. For the readers' convenience, we briefly describe the method of capacitance matrix below:

Without loss of generality, we assume that the first p rows of D are changed to obtain $\tilde{D}$. Although this is not the case here, but the same result can be achieved by using an implicit indexing scheme. Partition D, $\tilde{D}$ and $\boldsymbol{f}$ in the form

$$D = \begin{pmatrix} D_1 \\ D_2 \end{pmatrix},\ \tilde{D} = \begin{pmatrix} \tilde{D}_1 \\ D_2 \end{pmatrix},\ \boldsymbol{f} = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{pmatrix},$$

where D_1 and $\tilde{D}_1$ are $p\times r$ matrices with $r = (q+1)^2$ to be the order of the matrix D, and $\boldsymbol{f}_1$ is a vector of length p. Then the equation (3.9) can be solved as follows:

(0) Pre-processing: compute the $p\times p$ capacitance matrix

$$P = D_1\tilde{D}^{-1}\begin{pmatrix} I \\ 0 \end{pmatrix}, \tag{3.16}$$

where I is the $p\times p$ identity matrix; factor P into a LU form;
(1) Compute $\boldsymbol{v}_1 = D_1\tilde{D}^{-1}\boldsymbol{f}$;
(2) Solve $P\boldsymbol{w}_1 = \boldsymbol{f}_1 - \boldsymbol{v}_1$;
(3) Solve $\tilde{D}\boldsymbol{u} = \boldsymbol{f} + \begin{pmatrix} \boldsymbol{w}_1 \\ 0 \end{pmatrix}$.

One can check that u is in fact the solution of the equation $D\boldsymbol{u} = \boldsymbol{f}$.

It is clear that the steps 1 through 3 can be performed in $4N^3 + O(N^2)$ operations. The pre-processing stage, i.e. the computation of the capacitance matrix is however a little tricky. As explained in [5], one should take advantage of the sparseness of D_1 and $\begin{pmatrix} I \\ 0 \end{pmatrix}$ in the construction of P so that the preprocessing stage can be done in $O(pr + p^3) = O(N^3)$ operations. We refer to [5] for more details in this aspect.

However, unlike in the finite difference case [5], the capacitance matrix is no longer symmetric positive definite. For we can write $D = \tilde{D} + \Delta$ with

$$\Delta = \begin{pmatrix} E, 0, \cdots, 0 \\ 0, E, \cdots, 0 \\ \vdots \ \vdots \\ 0, \cdots, 0, E \end{pmatrix}$$

where E is a diagonal matrix of order q with four nonzero entries at positions $(1,1)$, $(2,2)$, $(q-1, q-1)$ and (q,q). Hence, $\tilde{D}^{-1}D = I + \tilde{D}^{-1}\Delta$ and

$$P = I + \bar{\Delta}\bar{D}, \tag{3.17}$$

where $\bar{\Delta}$ is the diagonal matrix with the nonzero entries of Δ and $\bar{D}$ is the part of $\tilde{D}^{-1}$ corresponding to the entries in positions: $(i,j) : j = 0, 1, \cdots, q;\ i = 0, 1, q-1, q$. $\tilde{D}$ is symmetric positive definite so is $\bar{D}$. It turns out that all the nonzero entries in Δ are positive. Hence P is not symmetric but it is similar to the symmetric positive definite matrix $\bar{\Delta}^{-\frac{1}{2}} + \bar{\Delta}^{\frac{1}{2}}\bar{D}\bar{\Delta}^{\frac{1}{2}}$. Furthermore, we note that D and $\tilde{D}$ have alternating zero and nonzero elements, hence the system (3.9) (resp. the capacitance matrix P) can be decoupled into four subsystems (resp. four submatrices of order q or $q+1$). This would reduce the cost of pre-processing by a factor of 16.

Finally, let us mention that three dimensional fourth order equations can also be treated by the above procedure.

4. Miscellaneous extensions. The Legendre Galerkin method described above can be applied to more general problems. In this section, we describe several immediately extensions.

4.1. Robin type boundary conditions. When other boundary conditions are prescribed, it is necessary to construct a basis incorporating the boundary conditions. For the sake of simplicity, we only consider the one dimensional equation

$$\alpha u - u_{xx} = f, \text{ in } I,$$

with the Robin type boundary condition

$$a_{\pm}u(\pm 1) + b_{\pm}u_x(\pm 1) = 0.$$

Let $W_N = \{v \in S_N : a_{\pm}v(\pm 1) + b_{\pm}v_x(\pm 1) = 0\}$, then the standard Legendre Galerkin approximation to the above equation is:
Find $u_{_N} \in W_N$ such that

$$\alpha(u_{_N}, v) + (D_x u_{_N}, D_x v) = (f, v),\ \forall\ v \in W_N. \tag{4.1}$$

As before, we can find an appropriate basis for W_N by setting

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

where a_k and b_k are the unique constants such that $\phi_k(x) \in W_N$. Therefore

$$W_N = \text{span}\{\phi_0(x), \phi_1(x), \cdots, \phi_{N-2}(x)\}.$$

Taking $v = \phi_k(x)$ in (4.1) for $k = 0, 1, \cdots, N-2$, it can be readily checked that the matrix corresponding to the discrete system (4.1) is a symmetric pentadiagonal matrix.

In the special case where $a_\pm = 0$ and $b_\pm = 1$ (i.e. the homogeneous Neumann boundary condition), we have in particular $a_k = 0$ and $b_k = -\frac{k(k+1)}{(k+2)(k+3)}$. In this particular case the matrix can be decoupled into two tridiagonal submatrices.

Let us mention that we can also construct special basis for the fourth order problems with the condition $u|_{\partial\Omega} = \Delta u|_{\partial\Omega} = 0$.

4.2. Non homogeneous boundary conditions. We can always modify the right-hand side to take care of the non homogeneous boundary conditions. Let us consider for instance the two dimensional Helmholtz equation (2.1). Non homogeneous boundary conditions for fourth order equations can be treated similarly.

If the solution u of (2.1) is subject to the non homogeneous Dirichlet boundary condition:

$$u(\pm 1, y) = a_\pm(y),\ u(x, \pm 1) = b_\pm(x),$$

we proceed as follows:

Setting

$$u^1(x,y) = \frac{b_+(x) - b_-(x)}{2} y + \frac{b_+(x) + b_-(x)}{2},\ \ \tilde{a}_\pm(y) = a_\pm(y) - u^1(\pm 1, y).$$

By construction, we have $\tilde{a}_\pm(\pm 1) = 0$.

Then setting

$$u^2(x,y) = \frac{\tilde{a}_+(y) - \tilde{a}_-(y)}{2} x + \frac{\tilde{a}_+(y) + \tilde{a}_-(y)}{2},\ \ \tilde{u} = u - u^1 - u^2.$$

One can readily check that

$$\tilde{u}(\pm 1, y) = \tilde{u}(x, \pm 1) = 0.$$

Hence it suffices to solve the following modified Helmholtz equation:

$$\alpha\tilde{u} - \Delta\tilde{u} = f - \alpha(u^1 + u^2) + \Delta(u^1 + u^2) \ \text{ in } \ \Omega = I \times I,\ \ \tilde{u}|_{\partial\Omega} = 0.$$

We note that the extra term $\Delta(u^1 + u^2)$ on the right hind side can be obtained in only $O(N)$ operations by using the recursive relation (2.5).

4.3. Non separable elliptic equations. For non separable equations of the form

$$\begin{aligned} Lu \equiv -\nabla \cdot [a(x,y)\nabla u] + b(x,y)u = f(x,y),\ \text{in } \Omega = I \times I, \\ u|_{\partial\Omega} = g(x,y), \end{aligned} \tag{4.2}$$

or more generally for problems with variable coefficients, the Legendre-Galerkin methods lead to systems with dense matrices, and can not be efficiently solved by a direct method. However the fast direct methods for constant coefficient problems developed in the previous sections can be used to efficiently solve the equation (4.2) by using an iterative procedure proposed in [9]. The heart of this iterative procedure is the utilization of the fast solver for the Helmholtz equation, regardless the type of discretization employed. It is shown in [9] that the rate of the convergence of the iterative procedure is independent of the number of modes used in each direction. Therefore an equation of the form (4.2) can be solved in $O(N^3)$ operations.

5. Numerical results. We report in this section several numerical examples by using the algorithms presented in the previous sections.

Let us remark that the pure spectral-Galerkin method is rarely used in practice, since for a general right-hand side function f we are not able to compute **exactly**

its representation by Legendre polynomials. Instead, the so called pseudo-spectral method is used to treat the right-hand side (or the product of two functions), i.e. we replace f by $I_N f$ in the computation, where $I_N f$ being the interpolation of f over the set of Gauss-Lobatto points. This would introduce an extra error term of order $C(\sigma)N^{-\sigma}\|f\|_\sigma$ for $f \in H^\sigma(\Omega)$ (see [8]) in the estimates (2.3) and (3.3). Hence, the spectral accuracy is still retained.

Unfortunately in the Legendre case, we can not use FFT to transform from the physical space to the spectral space and vice versa. Instead we shall use direct matrix multiplication, which takes $2dN^{d+1}$ operations for each transformation in a d dimensional domain. Hence for each particular solution of the Helmholtz equation and the biharmonic equation, $2dN^{d+1}$ additional operations are needed to transform $I_N f$ to the spectral space and to transform the solution u_N to the physical space. However let us remark that this additional amount of operations can be eventually reduced, at least for N sufficiently large, by using the fast transform developed by Orszag [17], to $O(N^d(\log_2 N)^2)$ operations, or by the fast multipole method presented in [4], to $O(N^d \log_2 N)$ operations. We note also that this additional amount of work is not needed if one is content with the solution in spectral space for right-hand side given in spectral space.

We consider first the two examples considered in [12].

Example 1. The Poisson equation

$$-\Delta u = 32\sin(4\pi x)\sin(4\pi y),\ \text{in } \Omega = I \times I,\ \ u|_{\partial\Omega} = 0,$$

with a smooth exact solution $u(x, y) = \sin(4\pi x)\sin(4\pi y)$.

Example 2. The Poisson equation

$$-\Delta u = 1,\ \text{in } \Omega = I \times I,\ \ u|_{\partial\Omega} = 0,$$

with an exact solution

$$u(x,y) = -\frac{64}{\pi^4} \sum_{\substack{n,m=1 \\ n,m \text{ odd}}}^{\infty} (-1)^{\frac{n+m}{2}} \frac{\cos(\frac{n\pi x}{2})\cos(\frac{m\pi y}{2})}{nm(n^2+m^2)},$$

which has singularities at the four corners.

We shall compare the Legendre-Galerkin method (LGM) with the Chebyshev-tau (CTM) and Chebyshev-collocation (CCM) methods.

In Table I, we list the maximum pointwise error of $u - u_N$ by LGM, CTM, CCM and the second and fourth order finite difference (FD2 and FD4) methods.

Table I. Maximum pointwise error of $u - u_N$ for examples 1 and 2.

Example	N	LGM	CTM	CCM	FD4	FD2
1	16	2.93E-3	3.33E-2	5.25E-3	2.81E-2	2.34E-1
1	32	3.44E-13	4.77E-11	2.12E-12	1.62E-3	5.30E-2
1	64	5.55E-15	8.67E-13	1.55E-13	9.97E-5	1.30E-2
1	128	6.88E-15	2.00E-12	1.56E-13	6.21E-6	3.22E-3
2	16	1.42E-6	3.52E-5	7.47E-7	7.17E-6	9.02E-4
2	32	7.48E-8	2.23E-6	5.51E-8	1.79E-6	2.26E-4

We notice that for the first example, all three spectral methods converge exponentially but the LGM is considerably more accurate than the CTM and CCM. For the second example which has corner singularities, the LGM is as accurate as

the CCM and is still clearly more accurate than the CTM. We note that the main source of roundoff errors comes from the matrix decomposition. Since decomposing a symmetric tridiagonal matrix in LGM is much less affected by roundoff errors than decomposing a full matrix in CTM and CCM, the roundoff errors in LGM are much less pronounced than that of CTM and CCM. This is confirmed by the numerical results in Table I.

In Table II, we list the execution time in seconds on Sun-Sparc2 for the first example by the three spectral methods. The approximate pre-processing times are given in parentheses. The programs were written in Fortran and compiled with the option $-O$. LAPACK routines were used for matrix-matrix multiplications and for solving the eigenvalue problems.

Table II. Execution time and pre-processing time for example 1.

Example	N	LGM	CTM	CCM
1	32	0.10 (0.03)	0.09 (0.13)	0.05 (0.27)
1	64	0.64 (0.12)	0.44 (0.54)	0.43 (2.44)
1	128	6.96 (1.01)	3.36 (5.20)	4.66 (19.58)

We note that for a particular solution of the Helmholtz equation, LGM is clearly the method of choice since its pre-processing time is negligible. For multiple solutions, the LGM is still competitive to CTM and CCM for N up to the range of 48 to 64.[1] We remark also that it is not very appropriate to compare the execution time of the collocation method which provides only the physical representation, to that of the Galerkin and tau methods which provide both the physical and spectral representations, since derivatives of functions, which are often needed for nonlinear problems, can be efficiently evaluated in spectral representations. We note that most of the computing time in LGM is used for transforming from physical representation to spectral representation and vice versa. Hence the CPU in LGM can be greatly reduced if one only needs the spectral representation.

Example 3. The 2-D biharmonic equation in $\Omega = I \times I$:

$$\Delta^2 u = 128\pi^4(\cos(4\pi x)\cos(4\pi y) - \cos(4\pi x)\sin(2\pi y)^2 - \cos(4\pi y)\sin(2\pi x)^2)$$
$$u|_{\partial\Omega} = \frac{\partial u}{\partial \boldsymbol{n}}|_{\partial\Omega} = 0,$$

with a smooth exact solution $u(x, y) = (\sin(2\pi x)\sin(2\pi y))^2$.

In Table III, we list in the second row the maximum pointwise error of $u - u_N$; in the third row, we list the execution time for Example 3 with approximate pre-processing time in parentheses; in the last row, we list the condition number of the capacitance matrix P.

Table III. Results for example 3.

N	16	32	64	128
L^∞ error	1.48E-2	7.45E-12	2.04E-14	2.81E-14
CPU (Pre-P)	0.02 (0.08)	0.12 (0.60)	0.82 (5.15)	8.28 (45.03)
Cond. no. of P	115.524	1608.96	24772.8	393214

[1] This range may be increased for highly parallel computers or when a fast transform method is employed.

It is obvious that the approximate solutions converge exponentially to the exact solution. We note that the condition numbers of P grow like $O(N^4)$. Hence the inversion of P is probably the main source of roundoff errors observed for N large. It is worth noting that the execution time for solving a 2-D biharmonic equation is only about 20% more than that for solving a 2-D Poisson equation.

The last example we consider is the non separable equation (4.2) with

$$a(x,y) = [1 + \frac{1}{4}((x+1)^4 + (y+1)^4)]^2, \quad u(x,y) = \sin(\pi x)\sin(\pi y).$$

For the details of the iterative scheme, we refer to [9]. Note however that we used the shifting factor $K \approx \|\frac{\Delta\sqrt{a(x,y)}}{\sqrt{a(x,y)}}\|_{L^2(\Omega)}$, which seemed to give a slightly better convergence rate, instead of $K \approx \frac{1}{2}(\min \frac{\Delta\sqrt{a(x,y)}}{\sqrt{a(x,y)}} + \max \frac{\Delta\sqrt{a(x,y)}}{\sqrt{a(x,y)}})$ suggested in [9]. Note also that the preconditioned conjugate gradient method, as opposed to the simple Richardson iteration proposed in [9], can be used to accelerate the convergence rate. Various iterative methods along with treatments for more general variable coefficient problems will be investigated in a future work.

Table IV. Results for the non separable equation.

N	16	32	64	128
$\max(u_N^7 - u_N^6)$	2.93E-7	4.87E-7	6.15E-7	6.83E-7
$\max(u_N^7 - u)$	2.47E-8	4.65E-8	6.16E-8	6.98E-8
CPU	0.09	0.62	4.51	45.06

We have summarized In Table IV the results after 7 Richardson iterations with the initial guess $u_N^0 \equiv 0$. The results clearly demonstrate that the convergence rate of the scheme is independent of N. Note however that due to the expensive transformations required by the Galerkin method, a collocation method is probably preferable for this problem, especially when N is large.

Concluding remarks. We have presented in this paper a systematic way to construct appropriate bases for the Legendre-Galerkin method applied to constant coefficient elliptic problems. We have also developed efficient direct solvers whose complexities are a small multiple of N^{d+1} operations in a d dimensional domain with $(N-1)^d$ unknowns. Furthermore, the algorithms can be effectively parallelized. Taking into account the fact that the convergence rate of the Legendre-Galerkin method is exponential, we conclude that our algorithms are very valuable for the specific problems considered in this article. To the best of the author's knowledge, our algorithm for the fourth order equations is the first fast direct solver with spectral discretization. Our direct solver for one particular second order equations is probably the most efficient among the existing spectral methods. Our direct solver for the second order equations with multiple right-hand sides is also very competitive to the direct solvers by Chebyshev-tau, Chebyshev-collocation and spectral element methods (see [12], [8], [18]).

We note that similar techniques can be applied to Galerkin method using Chebyshev polynomials (cf. [21]) or other Jacoby polynomials.

REFERENCES

[1] C. BERNARDI, G. COPPOLETTA, and Y. MADAY, *Some spectral approximations of two-dimensional fourth-order problems*, Math. Comp., 59 (1992), pp. 63–76.

[2] C. BERNARDI AND Y. MADAY, *Some spectral approximations of one dimensional fourth-order problems*, in Progress in Approximation Theory, P. Nevai and A. Pinkus, eds., Academic Press, 1991, pp. 43–116.

[3] P. BJORSTAD, *Fast numerical solution of the biharmonic Dirichlet problem on rectangles*, SIAM J. Numer. Anal., 20 (1983), pp. 59–71.

[4] J. P. BOYD, *Multipole expansions and pseudospectral cardinal functions*: a new generation of the fast Fourier transform, J. Comput. Phys., 103 (1992), pp. 184–186.

[5] B. L. BUZBEE AND F. W. DORR, *The direct solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions*, SIAM J. Numer. Anal., 11 (1974), pp. 753–763.

[6] B. L. BUZBEE, F. W. DORR, J. A. GEORGE, and G. H. GOLUB, *The direct solution of the discrete Poisson's equation on irregular domains*, SIAM J. Numer. Anal., 8 (1971), pp. 722–736.

[7] B. L. BUZBEE, G. H. GOLUB, and C. W. NIELSON, *On direct methods for solving Poisson's equations*, SIAM J. Numer. Anal., 7 (1970), pp. 627–656.

[8] C. CANUTO, M. Y. HUSSAINI, A. QUARTERONI, and T. A. ZANG, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1987.

[9] P. CONCUS AND G. H. GOLUB, *Use of fast direct methods for the efficient numerical solution of nonseparable elliptic equations*, SIAM J. Numer. Anal., 10 (1973), pp. 1103–1120.

[10] D. FUNARO AND W. HEINRICHS, *Some results about the pseudospectral approximation of one dimensional fourth order problems*, Numer. Math., 58 (1990), pp. 399–418.

[11] D. GOTTLIEB AND S. A. ORSZAG, *Numerical Analysis of Spectral Methods*: Theory and Applications, SIAM-CBMS, Philadelphia, 1977.

[12] D. B. HAIDVOGEL AND T. A. ZANG, *The accurate solution of Poisson's equation by expansion in Chebyshev polynomials* , J. Comput. Phys., 30 (1979), pp. 167–180.

[13] Y. MADAY AND A. T. PATERA, *Spectral element methods for the incompressible Navier-Stokes equations*, in State-of-the-art surveys on computational mechanics, A. K. Noor and J. T. Oden, eds., 1989, pp. 71–143.

[14] M. MARCUS AND H. MINC, *A Survey of Matrix Theory and Matrix Inequalities*, Prindle, Weber and Schmidt, Boston, 1964.

[15] A. MAYO, *The fast solution of Poisson's and the biharmonic equations on general regions* , SIAM J. Numer. Anal., 21 (1984), pp. 285–299.

[16] A. MAYO AND A. GREENBAUM, *Fast iterative solution of Poisson's and the biharmonic equations on irregular regions*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 101–118.

[17] S. A. ORSZAG, *Fast eigenfunction transforms*, Science and Computers: Advances in mathematics supplementary studies, 10 (1986), pp. 23–30.

[18] A. T. PATERA, *Fast direct Poisson solvers for high-order finite element discretizations in rectangularly decomposable domains*, J. Comput. Phys., 65 (1986), pp. 474–480.

[19] G. SACCHI-LANDRIANI, *Spectral tau approximation of the two-dimensional Stokes problem*, Numer. Math., 23 (1988), pp. 683–699.

[20] J. SHEN, *A spectral-tau approximation for the Stokes and Navier-Stokes equations*, Math. Model. Num. Anal., 22 (1988), pp. 677–693.

[21] ———, *Efficient spectral-Galerkin method II. Direct solvers for second- and fourth-order equations by using Chebyshev polynomials*, SIAM J. Sci. Comput., 16 (1995), pp. 74–87.

[22] J. SHEN AND R. TEMAM, *Nonlinear Galerkin methods using Chebyshev or Legendre polynomials I. One dimensional case*, SIAM J. Numer. Anal., 32 (1995), pp. 215–234.

[23] G. SIVASHINSKY, *Nonlinear analysis of hydrodynamic instability in laminar flames, Part I. Derivation of basic equations*, Acta Astronaut., 4 (1977), pp. 1177–1206.

[24] R. W. de VRIES AND P. J. ZANDBERGEN, *The numerical solution of the biharmonic equation, using a spectral multigrid method*, in Advances in Fluid Dynamics, W. F. Ballhaus and M. Y. Hussaini, eds., 1989, pp. 25–35.