SIAM J. SCI. COMPUT.
Vol. 16, No. 1, pp. 74–87, January 1995

006

Efficient Spectral-Galerkin Method II. Direct Solvers of Second and Fourth Order Equations by Using Chebyshev Polynomials[*]

JIE SHEN[†]

Abstract. Efficient direct solvers based on the Chebyshev-Galerkin methods for second and fourth order equations are presented. They are based on appropriate base functions for the Galerkin formulation which lead to discrete systems with special structured matrices which can be efficiently inverted. Numerical results indicate that the direct solvers presented in this paper are significantly more accurate and efficient than that based on the Chebyshev-tau method.

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

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

1. Introduction.

This paper is the second in a series for developing efficient spectral-Galerkin methods for elliptic equations. In part I of this work [12], we described the direct solution techniques for the Helmholtz equation and the biharmonic equation by using Legendre-Galerkin approximations. These direct solvers are very efficient compared to the existing ones, especially for solving the biharmonic equation. Although using Legendre polynomials offers many advantage over using Chebyshev polynomials, such as symmetricity and sparseness of the matrices for the discrete systems, its obvious drawback is the lack of fast transform between the physical and spectral spaces, such as Fast Fourier Transforms (FFT) for Chebyshev polynomials. Since such transforms must be performed frequently when dealing with nonlinear equations, it is preferable to use Chebyshev polynomials when the resulting discrete systems can be solved with an accuracy and a number of operations comparable to the case when Legendre polynomials are used. It is our purpose in this paper to develop efficient Galerkin methods using Chebyshev polynomials.

As we mentioned in [12], although the theoretical analysis of Chebyshev-Galerkin method is the easiest to perform and the results are usually optimal among the three commonly used spectral methods, namely Galerkin, collocation and tau, its practical implementation is virtually unavailable in the literature due to the lack of appropriate bases. On the other hand, the tau method and the collocation method have been extensively used. The latter of course is more flexible and more suitable for problems with variable coefficients. However for problems with constant coefficients, and also for some problems with variable coefficients (e.g. the example of a nonseparable equation considered in [12]), the Chebyshev-tau method and Chebyshev-Galerkin method could be more efficient. It is clear that for the Chebyshev-Galerkin approximations we can construct similar special bases as in [12] for the Legendre-Galerkin approximations. But due to the nonuniform weight associated with the Chebyshev polynomials, the matrices of the resulting linear systems are usually not sparse as in the Legendre case. However, these matrices usually possess special structures which can be explored to derive efficient algorithms.

[*] Received by the editors February 22, 1993; accepted for publication (in revised form) November 4, 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).

For the general second order equations with constant coefficients, we are able to apply the matrix decomposition method to solve the discrete system arising from the Chebyshev-Galerkin approximation in about $(d-1)N^{d+1}$ arithmetic operations (where $d = 2$ or 3 is the dimension of the domain, N is the cutoff number of the Chebyshev series in each direction) with an accuracy comparable to that of the Legendre-Galerkin approximation and significantly better than that of the Chebyshev-tau approximation. This algorithm is clearly superior to the popular Chebyshev-tau algorithm [8] in terms of accuracy and efficiency as is demonstrated by the numerical examples in Section 5. It is also more efficient than the Legendre-Galerkin algorithm in [12] for solving equations with multiple right hand sides, thanks to the FFT.

For the fourth order equations, we are only able to derive an efficient and stable algorithm in the one dimensional case. Again thanks to the FFT, it is more efficient than the the Legendre-Galerkin method presented in [12]. It can be used in particular to solve the 2-D Stokes equations with periodic-nonperiodic boundary conditions. However, due to probably the non symmetricity associated to the nonuniform Chebyshev weight function, we are unable to extend the method of capacitance matrix used in [2] and [12] to solve the 2-D biharmonic equations.

The remainder of the paper is organized as follows: In the next section, we describe in detail our algorithms for solving the Helmholtz equations. In section 3, we consider the fourth order equations. In section 4, we extend our technique to more general problems. Finally in section 5, we present and compare some numerical results.

2. Helmholtz equations. In this section, we are interested in using the Chebyshev-Galerkin method to solve 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. More general second order problems will be treated in Section 4.

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

$$S_N = \text{ span}\{T_0(x),\ T_1(x),\ \ldots,\ T_N(x)\},\ \ V_N = \{v \in S_N :\ v(\pm 1) = 0\}.$$

Then the standard Chebyshev-Galerkin approximation to (2.1) is:
Find $u_N \in V_N^d$ such that

$$\alpha(u_N, v)_\omega - (\Delta u_N, v)_\omega = (f, v)_\omega,\ \forall\, \boldsymbol{v} \in V_N^d, \tag{2.2}$$

where $\omega(\boldsymbol{x}) = \Pi_{i=1}^d (1 - x_i^2)^{-\frac{1}{2}}$ and $(u, v)_\omega = \int_\Omega uv\omega d\boldsymbol{x}$ is the scalar product in the weighted space $L^2_\omega(\Omega)$. The norm in $L^2_\omega(\Omega)$ will be denoted by $\|\cdot\|_\omega$. Let us denote $H^s_\omega(\Omega)$ to be the weighted Sobolev spaces with the norm $\|v\|_{s,\omega}$. It is well known (cf. [4]) that for $\alpha \geq 0$, $s \geq 1$ and $u \in H^s_\omega(\Omega)$, the following optimal error estimates holds:

$$\|u - u_N\|_\omega + N\|u - u_N\|_{1,\omega} \leq C(s)N^{-s}\|u\|_{s,\omega}. \tag{2.3}$$

Although the approximation (2.2) achieves the optimal convergence rate, its practical value depends on the choice of a basis for V_N^d. It is essential for the sake of efficiency to choose an appropriate basis for V_N^d such that the resulting linear system is as simple as possible. However, to the best of the author's knowledge, the only basis available in the literatures (see for instance [7]) is:

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

with

$$\phi_k(x) = \begin{cases} T_k(x) - T_0(x), & k \text{ even} \\ T_k(x) - T_1(x), & k \text{ odd} \end{cases}.$$

Unfortunately this basis leads to a linear system with full matrix and hence its usage is virtually prohibited in practice. In the following, we shall construct appropriate bases so that the matrices of the resulting linear systems can be efficiently inverted.

2.1. One dimensional case. The following simple lemma is the key to the efficiency of our algorithms.

Lemma 2.1. *Let* $\phi_k(x) = T_k(x) - T_{k+2}(x)$, $a_{kj} = -(\phi_j''(x), \phi_k(x))_\omega$ *and* $b_{kj} = (\phi_j(x), \phi_k(x))_\omega$. *Then*

$$V_N = span\{\phi_0(x), \phi_1(x), \cdots, \phi_{N-2}(x)\}; \tag{2.4}$$

and

$$b_{kj} = b_{jk} = \begin{cases} \frac{(c_k+1)}{2}\pi, & j = k \\ -\frac{\pi}{2}, & j = k-2 \textit{ and } j = k+2 \\ 0, & \textit{Otherwise} \end{cases}; \tag{2.5}$$

and

$$a_{kj} = \begin{cases} 2\pi(k+1)(k+2), & j = k \\ 4\pi(k+1), & j = k+2, k+4, k+6, \cdots \\ 0, & j > k \textit{ or } j+k \textit{ odd} \end{cases}. \tag{2.6}$$

Proof. It is clear that $\phi_k(x) \in V_N$ and that $\{\phi_k(x)\}$ are linear independent. (2.4) then follows from the fact that $\dim V_N = N-1$.

The proof of (2.5)-(2.6) is based on the following well known properties of Chebyshev polynomials. We recall that the $\{T_n(x)\}_{n=0}^{n=\infty}$ form an orthogonal basis for $L^2_\omega(I)$ and

$$(T_i(x), T_j(x))_\omega = c_i \frac{\pi}{2}\delta_{ij}, \ \forall\, i,j \geq 0, \tag{2.7}$$

where $c_0 = 2$ and $c_i = 1$ for $i \geq 1$. We recall also that the following recurrence relation holds

$$2T_n(x) = \frac{T'_{n+1}(x)}{n+1} - \frac{T'_{n-1}(x)}{n-1}. \tag{2.8}$$

Note that $T_n(x)$ is a polynomial of degree n and therefore $T_n''(x) \in S_{N-2}$. More precisely

$$T_n''(x) = \sum_{\substack{k=0 \\ k+n \text{ even}}}^{n-2} \frac{1}{c_k} n(n^2-k^2)T_k(x). \tag{2.9}$$

Now (2.5) can be easily derived by using (2.7). Thanks to (2.9), we have

$$T''_{k+2}(x) = \frac{1}{c_k}(k+2)((k+2)^2-k^2)T_k(x) + \frac{1}{c_{k-2}}(k+2)((k+2)^2-(k-2)^2)T_{k-2}(x) + \cdots \tag{2.10}$$

It follows immediately from (2.10) and (2.7) that

$$-(\phi_k''(x), \phi_j(x))_\omega = 0, \ \text{ for } j > k \text{ or } j+k \text{ odd},$$

and

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

Setting $\phi_j''(x)=\sum_{n=0}^{j} d_nT_n(x)$, by a simple computation using (2.9), 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 $j=k+2,k+4,\cdots$, we find

$$-(\phi_j''(x),\phi_k(x))_\omega = d_k(T_k(x),T_k(x))_\omega - d_{k+2}(T_{k+2}(x),T_{k+2}(x))_\omega = 4\pi(k+1). \quad \square$$

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

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

$$\alpha(u_{\scriptscriptstyle N},\phi_k(x))_\omega-(u_{\scriptscriptstyle N}'',\phi_k(x))_\omega=(f,\phi_k(x))_\omega, k=0,1,\cdots,N-2. \tag{2.11}$$

Let us denote

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

$$u_{\scriptscriptstyle N}=\sum_{n=0}^{N-2}v_n\phi_n(x),\ \boldsymbol{v}=(v_0,v_1,\cdots,v_{N-2})^T;$$

and

$$B=(b_{kj})_{0\le k,j\le N-2}\,,\ A=(a_{kj})_{0\le k,j\le N-2}\,. \tag{2.12}$$

Then (2.11) is equivalent to the following matrix equation:

$$(\alpha B+A)\boldsymbol{v}=\boldsymbol{f}. \tag{2.13}$$

First of all, we observe that $a_{kj}=b_{kj}=0$ for $k+j$ odd. Hence the above system of order $N-1$ can be decoupled into two subsystems of order $N/2$ and $N/2-1$. The same argument can also be applied to multidimensional systems. In fact, the 2-D system (2.14) and 3-D system (2.19) can be respectively decoupled into four and eight subsystems. However it is as efficient and less tedious in coding, especially in multidimensional cases, to treat the original systems directly.

(i) $\alpha=0$: (2.13) reduces to $A\boldsymbol{v}=\boldsymbol{f}$. From Lemma 2.1, we see that A is an special upper triangular matrix whose nonzero off-diagonal elements in each row are equal to a constant. Hence this linear system can be solved by special backward substitution in about $4N$ arithmetic operations.

(ii) $\alpha\ne 0$: We form explicitly the LU factorization $\alpha B+A=LU$. Since the matrix B has only 3 nonzero diagonals, the elements in each row of U, except the diagonal and the nearest off diagonal elements, are equal to a constant. Consequently the linear system (2.13) can be solved with essentially the same number of operations as needed for solving a pentadiagonal system.

2.2. Two dimensional case. It is clear that

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

Let us denote

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

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

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

$$\alpha BUB + AUB + BUA^T = F, \tag{2.14}$$

where A and B are the matrices defined in (2.12). This equation can be solved in particular by the matrix decomposition method as in [3], [8] and [12]. To this end, we need to study the following eigenvalue problems:

$$-(\psi_{xx}, v)_\omega = \lambda(\psi, v)_\omega, \; \forall\, v \in V_N, \;\; \psi(\pm 1) = 0. \tag{2.15}$$

Denoting $\psi(x) = \sum_{n=0}^{N-2} x_n\phi_n(x)$, and taking $v = \phi_j(x)$ for $j = 0, 1, \cdots, N-2$, we find that (2.15) is equivalent to the following generalized eigenvalue problem:

$$A\boldsymbol{x} = \lambda B\boldsymbol{x}. \tag{2.16}$$

It is shown by Gottlieb and Lustman [6] that the eigenvalues of the problem (2.15) are all real positive.

Now let Λ be the diagonal matrix whose diagonal elements are the eigenvalues of $A^{-1}B$, and let E be the matrix formed by the eigenvectors of $A^{-1}B$, i.e. $A^{-1}BE = E\Lambda$. From Lemma 2.2, the diagonal elements of Λ are all real positive and E is a real matrix. Applying A^{-1} to (2.14), we obtain

$$\alpha A^{-1}BUB + UB + A^{-1}BUA^T = A^{-1}F.$$

Setting $U = EV$, the above equation becomes

$$\alpha E\Lambda VB + EVB + E\Lambda VA^T = A^{-1}F.$$

Now applying E^{-1} to the above equation, set $G = E^{-1}A^{-1}F$, we find

$$\alpha\Lambda VB + VB + \Lambda VA^T = G. \tag{2.17}$$

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 row of the equation (2.17) becomes:

$$(\alpha\lambda_p + 1)B\boldsymbol{v}_p + \lambda_p A\boldsymbol{v}_p = \boldsymbol{g}_p, \;\; p = 0, 1, \cdots, N-2. \tag{2.18}$$

which is equivalent to $N-1$ one dimensional equation of the form (2.13).

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

(0) Compute the eigenpairs Λ, E for $A^{-1}B$ and compute E^{-1};
(1) Compute $G = E^{-1}A^{-1}F$;
(2) obtain V by solving (2.18);
(3) Set $U = EV$.

We remark that $A^{-1}B$ can be decoupled into two submatrices of upper Hessenburg form onto which the QR method can be directly applied. Hence the CPU time and more importantly the *roundoff errors* of the preprocessing stage are significantly reduced compared to that of Chebyshev-tau method (see Section 5 below). The step

2 and $A^{-1}F$ takes $O(N^2)$ operations. Note that the matrices E and E^{-1} have alternating zero elements and hence the steps 1 and 3 can be performed in about N^3 arithmetic operations.

2.3. Three dimensional case. Let us denote

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

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) with $d = 3$ for $i, j, k = 0, 1 \cdots, N-2$, we find that it is equivalent to the following equation:

$$\begin{aligned} &\alpha b_{in}u_{nml}b_{jm}b_{kl} + a_{in}u_{nml}b_{jm}b_{kl} + b_{in}u_{nml}a_{jm}b_{kl} + b_{in}u_{nml}b_{jm}a_{kl} = f_{ijk}, \\ &i, j, k = 0, 1, \cdots, N-2, \end{aligned} \tag{2.19}$$

where we have used the conventional notation as a pair of repeated index implies a summation of the index from 0 to $N-2$. As in [12], we shall decompose (2.19) into $N-1$ 2-D systems of the form (2.14).

Let us denote a_{ij}^{-1} and e_{ij}^{-1} (not to be confused with $1/a_{ij}$ and $1/e_{ij}$) to be respectively the ijth entry of the matrices A^{-1} and E^{-1}, then by the definition of E and Λ in Section 2.2, we have

$$a_{ik}^{-1}b_{kn}e_{nq} = \lambda_q e_{iq}, \;\; e_{qi}^{-1}e_{ip} = \delta_{qp}, \;\; a_{qi}^{-1}a_{ip} = \delta_{qp}. \tag{2.20}$$

Multiply a_{ri}^{-1} to the equation (2.19), and set $u_{nml} = e_{nq}v_{qml}$, using the above relations, we derive

$$\alpha\lambda_q e_{rq}v_{qml}b_{jm}b_{kl} + e_{rq}v_{qml}b_{jm}b_{kl} + \lambda_q e_{rq}v_{qml}a_{jm}b_{kl} + \lambda_q e_{rq}v_{nml}b_{jm}a_{kl} = a_{ri}^{-1}f_{ijk}.$$

Multiply e_{pr}^{-1} to the above equation, we obtain

$$(\alpha\lambda_p + 1)v_{pml}b_{jm}b_{kl} + \lambda_p(v_{pml}a_{jm}b_{kl} + v_{pml}b_{jm}a_{kl}) = e_{pr}^{-1}a_{ri}^{-1}f_{ijk} \equiv g_{pjk}.$$

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

$$(\alpha\lambda_p + 1)BV^pB + \lambda_p(AV^pB + BV^pA^T) = G^p, \;\; p = 0, 1, \cdots, N-2. \tag{2.21}$$

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

In summary, the solution of (2.19) consists of the following steps:

(0) Pre-processing: compute the eigenpairs Λ and E of $A^{-1}B$ and compute E^{-1};
(1) Compute $g_{pjk} = e_{pr}^{-1}a_{ri}^{-1}f_{ijk}$ for $p, j, k = 0, 1, \cdots, N-2$;
(2) Obtain V^p by solving (2.21) for $p = 0, 1, \cdots, N-2$;
(3) Set $u_{nml} = e_{nq}v_{qml}$ for $n, m, l = 0, 1, \cdots, N-2$.

Step 2 consists of solving $N-1$ two-dimensional equations of the form (2.14). Hence it takes about N^4 operations. Steps 1 and 3 take about N^4 operations. Hence each particular solution of (2.19) can be obtained in about $2N^4$ operations.

3. Fourth order equations.

In this section, we consider the fourth order equation with the first boundary condition

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

where $\boldsymbol{n}$ is the normal vector to $\partial\Omega$ and $d = 1$ or 2.

Let us denote

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

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

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

It can be shown that for $\alpha, \beta > 0$ and $u \in H_\omega^s(\Omega) \cap H_{0,\omega}^2(\Omega)$ for $s \geq 2$, then the following optimal error estimate holds (see [11]):

$$\|u - u_{\scriptscriptstyle N}\|_\omega + N\|u - u_{\scriptscriptstyle N}\|_{1,\omega} + N^2\|u - u_{\scriptscriptstyle N}\|_{2,\omega} \leq C(s)N^{-s}\|u\|_{s,\omega}. \tag{3.3}$$

3.1. One dimensional case. It is obvious that $\dim(W_N) = N - 3$ and it is an easy matter to verify that a basis of W_N is given by

$$\psi_k(x) = T_k(x) - \frac{2(k+2)}{k+3}T_{k+2}(x) + \frac{k+1}{k+3}T_{k+4}(x),\ k = 0, 1, \cdots, N-4. \tag{3.4}$$

Setting

$$f_k = (f, \psi_k(x))_\omega,\ \boldsymbol{f} = (f_0, f_1, \cdots, f_{N-4})^T;$$

$$u_{\scriptscriptstyle N} = \sum_{n=0}^{N-4} v_n\psi_n(x),\ \boldsymbol{v} = (v_0, v_1, \cdots, v_{N-4})^T;$$

and

$$a_{kj} = (\psi_j''(x), (\psi_k(x)\omega)''),\ A = (a_{kj})_{0\leq k,j\leq N-4},$$

$$c_{kj} = (\psi_j'(x), (\psi_k(x)\omega)'),\ C = (c_{kj})_{0\leq k,j\leq N-4},$$

$$b_{kj} = (\psi_j(x), \psi_k(x))_\omega,\ B = (b_{kj})_{0\leq k,j\leq N-4}.$$

By setting $v = \psi_k(x)$ for $k = 0, 1, \cdots, N-4$ in (3.2) with $d = 1$, we find that it is equivalent to the following matrix form:

$$(A + \alpha C + \beta B)\boldsymbol{v} = \boldsymbol{f}. \tag{3.5}$$

Lemma 3.1. *The nonzero elements of A, B and C are*:

$$b_{kk} = \left(c_k + \frac{4(k+2)^2}{(k+3)^2} + \frac{(k+1)^2}{(k+3)^2}\right)\frac{\pi}{2},\ \textit{where } c_0 = 2,\ c_k = 1 \textit{ for } k \geq 1,$$

$$b_{kk+2} = b_{k+2k} = -\left(\frac{k+2}{k+3} + \frac{k+4}{k+5}\frac{k+1}{k+3}\right)\pi,\ \ b_{kk+4} = b_{k+4k} = \frac{k+1}{k+3}\frac{\pi}{2},$$

$$c_{kk} = -4\frac{k+1}{k+3}(k+2)^2\pi,$$

$$c_{kk-2} = 2(k-1)(k+2)\pi,\ \ c_{kk+2} = 2(k+1)(k+2)\pi,$$

$$a_{kk} = 8(k+1)^2(k+2)(k+4)\pi,$$

$$a_{kj} = \frac{8}{j+3}(k+1)(k+2)\left(k(k+4) + 3(j+2)^2\right)\pi,\ j = k+2, k+4, \cdots.$$

Proof. All the formulaes except the last one can be obtained by direct computations using the properties of Chebyshev polynomials. The computation of the last formula is extremely tedious by hand and we have resorted to the symbolic computation software *Mathematica*. □

Although the matrix A in the system (3.5) is not sparse but its special structure allows us to obtain the solution in $O(N)$ operations. Let us explain the procedure in the case $\alpha = \beta = 0$ in detail. The solution of $A\boldsymbol{v} = \boldsymbol{f}$ can be obtained by the backward substitution:

$$v_k = (f_k - \sum_{\substack{j=k+1 \\ k+j \text{ even}}}^{N-4} a_{kj}v_j)/a_{kk}, \quad k = 0, 1, \cdots, N-4. \tag{3.6}$$

Notice that a_{kj} can be factorized as

$$a_{kj} = p(k)q(j) + r(k)s(j), \ j = k+2, k+4, \cdots,$$

with

$$p(k) = 8k(k+1)(k+2)(k+4)\pi, \ q(j) = \frac{1}{j+3},$$

$$r(k) = 24(k+1)(k+2)\pi, \ s(j) = \frac{(j+2)^2}{j+3}.$$

Therefore

$$\sum_{\substack{j=k+1 \\ k+j \text{ even}}}^{N-4} a_{kj}v_j = \sum_{\substack{j=k+1 \\ k+j \text{ even}}}^{N-4} (p(k)q(j) + r(k)s(j))v_j$$

$$= p(k) \sum_{\substack{j=k+1 \\ k+j \text{ even}}}^{N-4} q(j)v_j + r(k) \sum_{\substack{j=k+1 \\ k+j \text{ even}}}^{N-4} s(j)v_j.$$

It is then clear that the above relation for $k = 0, 1, \cdots, N-4$ can be evaluated in $O(N)$ operations. Hence $\{v_k\}_{k=0}^{N-4}$ can be obtained in just $O(N)$ operations.

In the case $\alpha, \beta \neq 0$, we can form explicitly the LU factorization, i.e. $A + \alpha C + \beta B = LU$. Notice that the entries of U, excluding the diagonal and two nearest offdiagonals, are also factorizable as A. Consequently the system can still be solved in $O(N)$ operations.

Whenever spectral methods are used for solving the fourth order equations, one should be concerned with roundoff errors caused by potentially large condition numbers. As is pointed out in [7], the direct application of tau method to the fourth order equations leads to very ill conditioned system and is numerically unstable. Large condition numbers of order N^8 were also reported [5] for the spectral-collocation approximation to the 1-D fourth order equations. However the Chebyshev-Galerkin approximation presented above leads to systems with smaller condition numbers and is numerically stable. In table I, we list the condition numbers of A and the diagonally scaled matrix DA, where D is the inverse of diagonal matrix diag(A). It is clear that *cond*$(A) = (N^4)$ and *cond*$(DA) = O(N^2)$. Hence the propagation of roundoff errors should not be very significant. The numerical example presented in Section 5 confirms that the above algorithm is numerically stable.

Table I. Condition numbers for the 1-D fourth order equation

N	16	32	64	128	256
cond(A)	4.896E+3	1.046E+5	1.907E+6	3.241E+7	5.535E+8
cond(DA)	6.636E+1	2.674E+2	1.057E+3	4.174E+3	1.653E+4

Remark 3.1. *The second boundary condition, i.e.* $u(\pm 1) = u_{xx}(\pm 1) = 0$, *can also be considered. In this case a basis for the space* $\tilde{W}_N = \{v \in S_N : \; v(\pm 1) = v_{xx}(\pm 1) = 0\}$ *is given by*:

$$\psi_k(x) = T_k(x) - (1+e_k)T_{k+2}(x) + e_k T_{k+4}(x),\ k = 0, 1, \cdots, N-4,$$

with $e_k = \dfrac{(k+1)(2k^2+4k+3)}{(k+3)(2k^2+12k+19)}$.

Unfortunately the exact formula for $a_{kj} = (\psi_j''''(x), \psi_k(x))_\omega$ *derived by using Mathematica is extremely complicated and not factorizable as in the case of the first boundary condition. Hence the solution of the corresponding discrete system requires* $O(N^2)$ *operations. Furthermore the linear system is extremely ill conditioned such that the numerical solution makes little sense even at a relatively small* $N = 32$. *Therefore it is more efficient and reliable to treat the 1-D fourth order equation(when* $\beta = 0$) *with the second boundary condition as a decoupled system of two 1-D second order equations.*

3.2. Two dimensional case. Using the same notations as in Section 2.2 (with $\phi_k(x)$ replaced by $\psi_k(x)$, and $N-2$ replaced by $N-4$), taking $v = \psi_l(x)\psi_m(y)$ in (3.2) with $d = 2$ for $l, m = 0, 1, \cdots, N-4$, we find that it is equivalent to the following matrix equation:

$$\alpha BUB + \beta(CUB + BUC^T) + AUB + 2CUC^T + BUA^T = F. \tag{3.7}$$

As we noted in [12], this system can be efficiently solved by the matrix decomposition method if $A^{-1}B$ and $A^{-1}C$ are commutative. Unfortunately, we are not able to make $A^{-1}B$ and $A^{-1}C$ commutative by modifying a few rows of A, B and C, as we successfully did for Legendre-Galerkin approximation [12]. Hence we are not able to apply the method of capacitance matrix to solve the system (3.7). However the system can still be solved by using a special band elimination procedure in $O(N^4)$ operations (see a similar procedure in [10]). But this is certainly not competitive to the direct solver with complexity $2N^3$ by Legendre-Galerkin method presented in [12].

4. Some extensions. We have described in [12] how problems with nonhomogeneous boundary conditions can be efficiently transformed into problems with corresponding homogeneous boundary conditions. We have also explained in [12] how to treat certain non separable equations. Those techniques apply directly to the Chebyshev case as well. In this section, we shall present some other relevant extensions.

4.1. Other boundary conditions. For Neumann or Rabin type boundary conditions, we should construct special base functions satisfying the corresponding homogeneous boundary conditions. Let us consider for instance the 1-D equation with the homogeneous Neumann boundary condition:

$$-u_{xx} = f \text{ in } I,\ u_x(\pm 1) = 0. \tag{4.1}$$

Let $W_N = \left\{u \in S_N : \int_I u dx = 0,\ u_x(\pm 1) = 0\right\}$, and $\phi_k(x) = T_k(x) - \frac{k^2}{(k+2)^2}T_{k+2}(x)$. It can be easily shown that

$$W_N = \text{span}\{\phi_1(x), \cdots, \phi_{N-2}(x)\}. \tag{4.2}$$

Then the standard Chebyshev-Galerkin method for (4.1) is:

Find $u_N = \sum_{n=1}^{N-2} v_n \phi_n(x) \in W_N$ such that

$$-(u_N'', v)_\omega = (f, v)_\omega,\ \forall\, v \in W_N. \tag{4.3}$$

If we denote $a_{kj} = -(\phi_j''(x), \phi_k(x))_\omega$, by a direct computation as in Lemma 2.1, we can derive

$$a_{kj} = \begin{cases} 2\pi(k+1)k^2/(k+2), & j = k \\ 4\pi j^2(k+1)/(k+2)^2, & j = k+2, k+4, k+6, \cdots \\ 0, & j > k \text{ or } j+k \text{ odd} \end{cases} . \tag{4.4}$$

Now let $\tilde{a}_{kj} = a_{kj}/j^2$, $\tilde{A} = (\tilde{a}_{kj})_{1\le k,j\le N-2}$ and $g_j = (f, \phi_j(x))_\omega/j^2$. Then the equation (4.3) is equivalent to $\tilde{A}\boldsymbol{v} = \boldsymbol{g}$. We note that $\tilde{A}$ now has the same structure as the matrix A defined in (2.6). Therefore the system can be solved in $O(N)$ operations.

4.2. Equations with additional first order terms. Let us first consider the 1-D equation

$$\alpha u + \beta u_x - u_{xx} = f \quad \text{in } I \ u(\pm 1) = 0. \tag{4.5}$$

Let $c_{kj} = (\phi_j'(x), \phi_k(x))_\omega$ and $C = (c_{kj})_{0\le k,j\le N-2}$. A simple computation leads to

$$c_{kj} = \begin{cases} \pi(k+1) & j = k+1 \\ -\pi(k+1) & j = k-1 \\ 0, & \text{Otherwise} \end{cases} . \tag{4.6}$$

Hence the discrete system corresponding to (4.5) is:

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

where A and B are the matrices defined in Lemma 2.1. Hence the system can be solved in $O(N)$ operations.

Multidimensional problems can be efficiently handled as well. We consider for instance the 2-D equation

$$\alpha u + \beta u_y - \Delta u = f \quad \text{in } \Omega = I \times I \ , u|_{\partial\Omega} = 0. \tag{4.7}$$

Using the same notations as in Section 2.2, we find that the discrete equation corresponding to (4.7) is:

$$\alpha BUB + \beta BUC^T + AUB + BUA^T = F,$$

which can still be solved by the matrix decomposition method exactly as in the case $\beta = 0$ in Section 2.2.

4.3. 2-D Stokes equations with periodic-nonperiodic boundary conditions. We consider the 2-D Stokes equations

$$\begin{aligned} & -\Delta u + p_x = f, \ -\Delta v + p_y = g, \quad \text{in } \Omega, \\ & u_x + v_y = 0, \quad \text{in } \Omega, \end{aligned} \tag{4.8}$$

where $\Omega = [-1,1] \times [-\pi, \pi]$. We assume that the solutions (u, v, p) of the 2-D Stokes equations are periodic in the y-direction and satisfy the homogeneous Dirichlet boundary condition on the two vertical boundaries of Ω. We can then write

$$\begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=-\infty}^{+\infty} \begin{pmatrix} u^k(x) \\ v^k(x) \end{pmatrix} e^{iky}, \ \begin{pmatrix} f \\ g \end{pmatrix} = \sum_{k=-\infty}^{+\infty} \begin{pmatrix} f^k(x) \\ g^k(x) \end{pmatrix} e^{iky}, \ p = \sum_{k=-\infty}^{+\infty} p^k(x) e^{iky}.$$

Hence the 2-D Stokes equations can be split into a series of one dimensional systems:

$$-u^k_{xx}+k^2u^k+p^k_x=f^k,\ -v^k_{xx}+k^2v^k+ikp^k=g^k,\ x\in I,$$
$$u^k_x+ikv^k=0,\ x\in I,\ \ u^k(\pm1)=v^k(\pm1)=0,\ \ -\infty<k<+\infty.$$

For each k, we can easily eliminate v^k and p^k from the above system and the result is the fourth order equation:

$$u^k_{xxxx}-2k^2u^k_{xx}+k^4u^k=ikg^k_x+k^2f^k,\ x\in I,\ \ u^k(\pm1)=u^k_x(\pm1)=0. \tag{4.9}$$

Once u^k is known, v^k and p^k can be determined by

$$v^k=\frac{i}{k}u^k_x,\ \ p^k=\frac{i}{k}(g^k-k^2v^k+v^k_{xx}),\ \ k\neq0.$$

For $k=0$, we simply get $u^0=0,p^0=xf^0$ and $v^0=-\frac{f^0}{2}(x^2-1)$.

The equation (4.9) is exactly a fourth order equation of the form (3.1) with $d=1$. Hence its Chebyshev-Galerkin approximation can be efficiently implemented. For a different treatment of the equations (4.8) with periodic-nonperiodic boundary conditions, we refer to [1].

5. Numerical results. As noted by many authors (see [8] and [9]), when solving the discrete Helmholtz system by the matrix decomposition method, roundoff errors could be significant for large N since the accuracy of the algorithm relies on the accuracy of the matrix decomposition. Hence we shall first examine the roundoff errors of solving the discrete system associated with the 1-D Poisson equation by the matrix decomposition method. Numerical experiments indicate that the roundoff errors in the multidimensional computations by matrix decomposition method behave similarly as in the 1-D case.

Let $\boldsymbol{u}=(u_0,u_1,\cdots,u_{N-2})^T$ be a uniformly distributed random vector and we compute $\boldsymbol{f}=A\boldsymbol{u}$ where A is the matrix given in (2.6). Let $\boldsymbol{v}$ be the approximate solution to the system $A\boldsymbol{u}=\boldsymbol{f}$ obtained by using the matrix decomposition method, $\frac{\max_{0\le i\le N-2}|u_i-v_i|}{\max_{0\le i\le N-2}|u_i|}$ can then be regarded as the roundoff error of the procedure. In table II, we list the roundoff errors as described above by using the Chebyshev-Galerkin (CG), Legendre-Galerkin (LG) and Chebyshev-tau (CT) methods.

All computations are performed in double precision on SunSparc 2 workstation. LAPACK subroutines *dgeev, dstev* are used to compute the eigenvalue problems. VFFTPACK subroutines are used for FFT. The integer N represents the cutoff number in the Chebyshev or Legendre series.

Table II. Roundoff errors of the matrix decomposition methods

N	8	16	32	64	128	256
CG	3.36E-15	3.10E-15	4.88E-13	4.27E-11	1.54E-9	2.92E-9
CT	4.38E-13	2.44E-12	2.61E-11	1.28E-9	2.36E-8	7.86E-8
LG	1.73E-15	8.44E-15	1.02E-13	4.81E-11	4.49E-11	8.99E-11

Two remarks are in order. Firstly the roundoff errors of CT method are much more pronounced than that of CG and LG methods. Consequently the CG and LG methods are significantly more accurate than the CT method (see also table III). Secondly there were doubts [8] that the matrix decomposition method would not be suitable for computations with large N because the rapid increase of roundoff errors documented in [9] when N goes from 8 to 64. However our computations reveal that roundoff errors only increase slightly when N is further increased from 64 to 256. Hence all three methods, especially CG and LG methods, can be safely used for computations with large cutoff N as long as the computations are performed in double precision.

We now report on several numerical examples by CG, CT and LG methods. Let us remark that the pure spectral-Galerkin method is rarely used in practice. In fact the so called pseudospectral method is used to treat the right hand sides, i.e. we replace f by its polynomial interpolation over the set of Gauss-Lobatto points.

For the sake of comparison, we shall use again the two examples used in [8].

Example 1. The 2-D Poisson equation

$$-\Delta u = 2k^2 \sin(k\pi x)\sin(k\pi y), \text{ in } \Omega = I \times I, \quad u|_{\partial\Omega} = 0,$$

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

Example 2. The 2-D Poisson equation

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

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.

In table III, we list the maximum pointwise error of $u - u_N$ by LG, CT, CG methods.

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

Example	N	CG	CT	LG
1	16 (k=4)	5.22E-3	3.33E-2	2.93E-3
1	32 (k=4)	2.17E-12	4.77E-11	3.44E-13
1	64 (k=4)	6.11E-15	8.67E-13	5.55E-15
1	128 (k=32)	2.85E-9	9.15E-8	1.82E-9
1	256 (k=32)	2.79E-14	1.59E-12	3.39E-14
2	16	3.36E-6	3.52E-5	1.42E-6
2	32	1.27E-7	2.23E-6	7.48E-8

Table IV. Execution time and pre-processing time.

N	32	64	128
CG	0.05 (0.08)	0.26 (0.48)	1.96 (4.05)
CT	0.09 (0.13)	0.44 (0.54)	3.36 (5.20)
LG	0.10 (0.03)	0.64 (0.12)	6.96 (1.01)

In Table IV, we list the execution time for the three methods. The approximate preprocessing time is given in parentheses.

We note that in terms of accuracy and efficiency, the CG method is far more superior than the popular CT method. On the other hand, The accuracy of the CG method and the LG method is comparable. The LG method is probably the method of the choice if only one particular solution is needed. The CG method is more efficient for equations with multiple right hand side, as is the case for solving time dependent problems. However the LG method could be potentially accelerated by implementing the fast transforms mentioned in [12].

Example 3. The 1-D fourth order equation

$$\begin{aligned}&u_{xxxx} - \alpha u_{xx} + \beta u = -8\pi^2(16\pi^2 + \alpha)\cos(4\pi x) + \beta\sin(2\pi x),\ x \in I,\\ &u(\pm 1) = u_x(\pm 1) = 0,\end{aligned}$$

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

In table V, we list the maximum pointwise error of $u - u_N$ by the CG method with two typical choices of α, β. The results indicate that the spectral accuracy is achieved and that the effect of roundoff errors is very limited.

Table V. Maximum pointwise error of $u - u_N$ for Example 3 by CG method.

N	16	32	64	128
CG ($\alpha = \beta = 0$)	1.97E-2	9.42E-12	9.42E-14	4.68E-13
CG ($\alpha = 2 * N^2, \beta = N^4$)	5.99E-3	3.00E-12	5.38E-15	2.35E-14

Concluding remarks. We have presented in this paper some efficient direct solvers for the general second order equations and for the 1-D fourth order equations by using the Chebyshev-Galerkin approximation. Our algorithm for the second order equations is more accurate and more efficient than the Chebyshev-tau method. Furthermore the implementation of the algorithm is also relatively easier. Our algorithm for the 1-D fourth order equations is very efficient and numerically stable. It can be used in particular to solve the 2-D Navier-Stokes equations with periodic-nonperiodic boundary conditions. Unfortunately, we were not able to derive an competitive algorithm for the 2-D fourth order equations.

REFERENCES

[1] C. Bernardi, Y. Maday, and B. M'etivet, *Calcul de la pression dans la résolution spectrale du problème de Stokes*, La Recherche Aérospatiale, 1987.

[2] 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.

[3] 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.

[4] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1987.

[5] D. Funaro and W. Heinrichs, *Some results about the pseudospectral approximation of one dimensional fourth order problems*, Numer. Math., 58 (1990), pp. 399–418.

[6] D. Gottlieb and L. Lustman, *The spectrum of the Chebyshev collocation operator for the heat equation*, SIAM J. Numer. Anal., 20 (1983), pp. 909–921.

[7] D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM-CBMS, Philadelphia, 1977.

[8] 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.

[9] 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 (1984), pp. 115–128.

[10] W. HEINRICHS, *A stabilized treatment of the biharmonic operator with spectral methods*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 1162–1172.

[11] Y. MADAY AND B. M'ETIVET, *Chebyshev spectral approximation of Navier-Stokes equations in a two dimensional domain*, Model. Math. Anal. Numer., 21 (1986), pp. 93–123.

[12] J. SHEN, *Efficient spectral-Galerkin method I. Direct solvers for second- and fourth-order equations by using Legendre polynomials*, SIAM J. Sci. Comput., 15 (1994), pp. 1489–1505.