Efficient Chebyshev-Legendre Galerkin Methods for Elliptic Problems

Jie Shen*

Abstract

We introduce a new and efficient Chebyshev-Legendre Galerkin method for elliptic problems. The new method is based on a Legendre-Galerkin formulation, but only the Chebyshev-Gauss-Lobatto points are used in the computation. Hence, it enjoys advantages of both the Legendre-Galerkin and Chebyshev-Galerkin methods.

Key words: Chebyshev polynomial, Legendre polynomial, spectral-Galerkin method.

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

1 Introduction

The Legendre-Galerkin method [13] for self-adjoint elliptic equations leads to symmetric and simpler linear systems, but its efficiency is limited by the lack of fast transform between the physical space (values at the Legendre-Gauss-Lobatto points) and the spectral space (coefficients of the Legendre polynomials). Furthermore, the Legendre-Gauss-Lobatto (LGL) points are not available in an explicit form and their evaluations for large N (N being the order of polynomial space) may introduce significant roundoff errors (cf. [1]). On the other hand, the Chebyshev-Gauss-Lobatto (CGL) points are given explicitly and the transform between the physical space and the spectral space can be efficiently performed by using the Fast Fourier Transform (FFT). However, due to the non-uniform weight associated with the Chebyshev polynomials, the Chebyshev-Galerkin method [14] leads to non-symmetric (even for self-adjoint elliptic equations) and more complex linear systems. We introduce in this paper a very efficient Chebyshev-Legendre Galerkin method that has the advantages of both the Legendre- and Chebyshev-Galerkin methods. Our method has two essential features:

- It is based on a Legendre-Galerkin formulation which preserves the symmetry of the underlying problem. The basis functions of the approximation space are *compact combinations* of Legendre polynomials, determined by the order of the principle elliptic operator and the underlying boundary conditions. For problems with constant coefficients, the linear systems resulted from these *compact* basis functions are banded sparse matrices, similar to those arising from a finite difference discretization, which can be efficiently inverted.
- Only the coefficients of Legendre expansions and the values at the CGL points are used in the computation. Efficient algorithms are available to transform from the coefficients of Legendre expansions to the values at the CGL points and vice versa.

In the next section, we describe the Legendre- and Chebyshev-Galerkin methods for an one-dimensional model problem. In Section 3, we introduce the Chebyshev-Legendre Galerkin method and describe the fast Chebyshev-Legendre transform between the values at CGL points and the coefficients of Legendre expansion. In Section 4, we present some numerical results which demonstrate the efficiency of the new method.

2 Legendre-Galerkin and Chebyshev-Galerkin methods

To simplify the presentation, we consider the following one dimensional model problem:

$$(1)\qquad \alpha u - u_{xx} = f, \text{ in } I = (-1,1),$$

with the Robin type boundary condition

$$(2)\quad a_- u(-1) + b_- u_x(-1) = 0,\ a_+ u(1) + b_+ u_x(1) = 0.$$

*Department of Mathematics, Penn State University, University Park, PA 16802, USA. This work was supported in part by NSF grant DMS-9205300.

ICOSAHOM'95: Proceedings of the Third International Conference on Spectral and High Order Methods. ©1996 Houston Journal of Mathematics, University of Houston.

The problem(1-2) has a unique solution if

(3) $a_-^2 + b_-^2 \neq 0,\ a_-b_- \leq 0;\ a_+^2 + b_+^2 \neq 0,\ a_+b_+ \geq 0.$

Let S_N be the space of polynomials of degree less than or equal to N, and

$$X_N = \{v \in S_N :\ a_\pm v(\pm 1) + b_\pm v_x(\pm 1) = 0\}.$$

We denote I_N^l the operator of interpolation at the LGL points $\{\xi_i\}_{0\leq i\leq N}$ (ξ_i are the roots of the polynomial $(1-x^2)L_N'(x)$, where $L_N(x)$ is the Legendre polynomial of degree N), i.e. $I_N^l f \in S_N$ and

$$I_N^l f(\xi_i) = f(\xi_i),\ 0 \leq i \leq N.$$

Let $(u,v) = \int_{-1}^{1} uv\ dx$. Then, the *pseudo-spectral* Legendre-Galerkin method for (1-2) is: find $u_N \in X_N$ s.t. $\forall\, v_N \in X_N$,

(4) $$\alpha(u_N, v_N) - (D_{xx}u_N, v_N) = (I_N^l f, v_N).$$

Remark 2.1 *The above formulation is slightly different from the usual Galerkin method in the following aspects:*

- *f is replaced by its interpolant $I_N^l f$ to allow faster evaluation;*
- *the boundary condition (2) in all cases is strongly enforced as in a collocation scheme, while in case $b_-^2 + b_+^2 > 0$ the boundary condition (2) will only be satisfied as $N \to +\infty$ in a usual Galerkin method (cf. [2]).*

The actual solution procedure for (4) depends on the choice of basis functions for X_N. It is essential for the sake of efficiency to use *compact combinations* of orthogonal polynomials (with respect to the inner product $(\cdot,\cdot)$) as basis functions. To this end, we set

$$\phi_k(x) = L_k(x) + a_k L_{k+1}(x) + b_k L_{k+2}(x),\ k = 0,1,2,\dots$$

We will choose $\{a_k, b_k\}$ such that $\phi_k(x)$ verifies the boundary condition (2). 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 (2) leads to the following system for $\{a_k, b_k\}$:

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

The determinant of the above system is

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

We then conclude from (3) that $\mathrm{DET}_k \neq 0$ for any k. Hence, $\{a_k, b_k\}$ can be uniquely determined from (5).

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

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

Let us denote

$$f_k = (I_N^l f, \phi_k),\ \boldsymbol{f} = (f_0, f_1, \cdots, f_{N-2})^T;$$
$$u_N(x) = \sum_{n=0}^{N-2} v_n\phi_n(x),\ \boldsymbol{v} = (v_0, v_1, \cdots, v_{N-2})^T,$$
$$s_{kj} = (-D_{xx}\phi_k, \phi_j),\ m_{k,j} = (\phi_k, \phi_j),$$

and

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

Then the equation (4) is equivalent to the following linear system:

(6) $$(\alpha M + S)\boldsymbol{v} = \boldsymbol{f}.$$

By integration by parts and taking into account the boundary condition (2), we find that

$$\begin{aligned} s_{kj} &= -(\frac{d^2\phi_k}{dx^2}, \phi_j) \\ &= (D_x\phi_k, D_x\phi_j) + \frac{a_+}{b_+}\phi_k(1)\phi_j(1) - \frac{a_-}{b_-}\phi_k(-1)\phi_j(-1) \\ &= -(\phi_k, \frac{d^2\phi_j}{dx^2}), \end{aligned}$$

where $\frac{a_+}{b_+}$ (resp. $\frac{a_-}{b_-}$) should be replaced by zero when $b_+ = 0$ (resp. $b_- = 0$). Hence, using the orthogonality of the Legendre polynomials, it is easy to verify that the stiffness matrix S is a diagonal matrix with

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

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

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

and $m_{jk} = m_{kj}$.

In summary: given the values of f at LGL points $\{\xi_i\}_{0\le i\le N}$, we determine the values of u_N (solution of (4)) 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 $I_N^l f(x)$ from $\{f(\xi_i)\}_{i=0}^N$ (backward Legendre transform);
3. Evaluate $\boldsymbol{f}$ and solve $\boldsymbol{v}$ from (6);
4. Evaluate $u_N(\xi_j) = \sum_{i=0}^{N-2} v_i\phi_i(\xi_j)$, $j = 0, 1, \ldots, N$ (forward Legendre transform).

It is clear that step 3 can be done in $O(N)$ operations. However, each *Legendre transform* in steps 2 and 4 involves a matrix-by-vector product which will take about $2N^2$ arithmetic operations. To reduce the cost of the transforms between physical and spectral spaces, a natural choice is to use Chebyshev polynomials. We now describe briefly below the Chebyshev-Galerkin method for (1).

Let I_N^c be the interpolation operator at the CGL points $\{\eta_i = \cos(i\pi/N)\}_{0\le i\le N}$, i.e. $I_N^c f \in S_N$ and

$$I_N^c f(\eta_i) = f(\eta_i),\ 0 \le i \le N.$$

The *pseudo-spectral* Chebyshev-Galerkin method for (1-2) is: find $u_N \in X_N$ s.t. $\forall\, v_N \in X_N$,

$$(8) \qquad \alpha(u_N, v_N)_\omega - (D_{xx}u_N, v_N)_\omega = (I_N^c f, v_N)_\omega,$$

where $\omega(x) = (1-x^2)^{-\frac{1}{2}}$ and $(u,v)_\omega = \int_{-1}^{1} uv\omega\, dx$. As before, there exist unique $\{a_k, b_k\}$ such that

$$\psi_k(x) = T_k(x) + a_k T_{k+1}(x) + b_k T_{k+2}(x) \in X_N,$$

(where $T_k(x)$ is the Chebyshev polynomial of degree k) and

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

It is easy to see that the stiffness matrix S ($s_{ij} = (-D_{xx}\psi_j, \psi_i)_\omega$) is a upper triangular matrix and the mass matrix M ($m_{ij} = (\psi_j, \psi_i)_\omega$) is a symmetric positive definite penta-diagonal matrix. Although the matrix $\alpha M + S$ in (8) is not sparse, it can still be inverted at a cost comparable to invert a seven-diagonal matrix by exploiting the special structures of S [14]. Therefore, thanks to the fast transforms available to the Chebyshev expansions, the Chebyshev-Galerkin method is preferable for this specific problem. However, the Chebyshev-Galerkin method may not be the best choice due to the non-symmetry (which introduces considerable difficulties for its analysis and excludes the use of conjugate gradient type methods for problems with variable coefficients) and non-sparseness of the stiffness matrix (which excludes more efficient direct solvers such as the cyclic reduction [15] for problems in multi-dimensional domains).

Remark 2.2 *In case of the homogeneous Dirichlet boundary condition (i.e. $b_\pm = 0$) or the homogeneous Neumann boundary condition (i.e. $a_\pm = 0$), we have $a_k = 0$, $k = 0, 1, 2, \ldots$ These special cases are discussed in detail in [13, 14].*

For differential equations of order $2m$, we should choose the basis functions for the Legendre- and Chebyshev-Galerkin method to be of the form

$$\phi_k(x) = P_k(x) + a_{k,1}P_{k+1}(x) + \ldots + a_{k,2m}P_{k+2m}(x)$$

where $P_i(x)$ are Legendre or Chebyshev polynomials and $a_{i,j}$ should be determined by the underlying boundary conditions. See [13, 14] for the treatment of fourth-order problems.

For multi-dimensional problems, tensor products of one dimensional basis functions should be used. The resulting linear systems can be efficiently solved, for instance, by the matrix decomposition method [11, 9]. We refer to [13, 14] for more details.

3 Chebyshev-Legendre Galerkin method

To overcome the shortcomings of both the Legendre- and Chebyshev-Galerkin methods, we propose the following Chebyshev-Legendre Galerkin method for (1-2): find $u_N \in X_N$ s.t. $\forall\, v_N \in X_N$,

$$(9) \qquad \alpha(u_N, v_N) - (D_{xx}u_N, v_N) = (I_N^c f, v_N).$$

The only difference with (4) is that the Chebyshev interpolation operator I_N^c is used instead of the Legendre interpolation operator I_N^l. Hence, the solution procedure of (9) is essentially the same as that of (4) except that *Chebyshev-Legendre transforms* (between the value of a function at *CGL* points and the coefficients of its *Legendre* expansion) are needed instead of the *Legendre transforms.*

Recently Don and Gottlieb [6] introduced also a Chebyshev-Legendre collocation method for hyperbolic and parabolic equations. Their work is motivated by the fact that the hyperbolic equations are not well-posed in Chebyshev norm. In their method, the forcing term is appropriately penalized so that L^2 norm estimates can be obtained for the collocation method using Chebyshev points. Although the two methods share the same spirit which is to take advantages of both the Chebyshev and Legendre methods, the motivation and the implementation of the two methods are entirely different.

We now describe how the Chebyshev-Legendre transforms can be efficiently implemented. We split each Chebyshev-Legendre transform into two steps:

1. The transform between its values at CGL points and the coefficients of its Chebyshev expansion. This can be done by FFT in $\frac{5}{2}\log_2 N + 4N$ operations (cf. P. 502 in [2]).
2. The transform between the coefficients of the Chebyshev expansion and of the Legendre expansion.

This second transform has been addressed by Alpert and Rohklin [1]. They developed an $O(N)$ algorithm for the second transform for a prescribed precision. Their algorithm, as most algorithms based on the fast multipole method [7], is most attractive for very large N. For moderate N, the algorithm we describe 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

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

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

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

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

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

Then we have

$$\boldsymbol{f} = A\boldsymbol{g}, \ \boldsymbol{g} = B\boldsymbol{f}, \ AB = BA = I. \tag{10}$$

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

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

Hence, both A and B only have about $\frac{1}{4}N^2$ nonzero elements, and the cost of each transform between $\boldsymbol{f}$ and $\boldsymbol{g}$ 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 Table I for computational results).

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 Rohklin's algorithm [1]. 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:

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

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 (10) is extremely easy to implement, while the algorithm in [1] requires considerable programming effort.

We now turn our attention to problems with variable coefficients. Let us consider for instance the following nonseparable equation:

$$\begin{cases} -\text{div}(a(\boldsymbol{x})\nabla u) + b(\boldsymbol{x})u = f, \ \boldsymbol{x} \in \Omega = [-1,1]^d, \\ u|_{\partial\Omega} = 0, \end{cases} \tag{11}$$

where $d = 1$, 2 or 3, $a(\boldsymbol{x}) > 0$ and $b(\boldsymbol{x}) \geq 0$ for $\boldsymbol{x} \in [-1,1]^d$. We can apply the spectral-Galerkin method directly to this equation, but it is more efficient in most cases to make a change of dependent variable $v = \sqrt{a}u$ [4] which reduces (11) to:

$$(12) \quad \begin{cases} \mathcal{A}v \equiv -\Delta v + p(\boldsymbol{x})v = q, \; \boldsymbol{x} \in \Omega = [-1,1]^d, \\ v|_{\partial\Omega} = 0, \end{cases}$$

where $p(\boldsymbol{x}) = \frac{1}{\sqrt{a}}\Delta(\sqrt{a}) + \frac{1}{a}b(\boldsymbol{x})$ and $q(\boldsymbol{x}) = \frac{1}{\sqrt{a}}f$.

The unmodified spectral-Galerkin method is usually impractical for this type of problems. We propose instead the following *pseudo-spectral* Chebyshev-Legendre Galerkin method for (12): find $v_N \in X_N = \{w \in S_N : w|_{\partial\Omega} = 0\}$ s.t.

$$(13) \qquad (\mathcal{A}_{sp}v_N, w) = (I_N^c f, w), \forall w \in X_N,$$

where $\mathcal{A}_{sp}$ is defined

$$(\mathcal{A}_{sp}v, w) = (\nabla v, \nabla w) + (I_N^c(p(\boldsymbol{x})v), w),$$

and S_N is the space of polynomials of degree $\leq N$ in each variable. It is clear that the matrix corresponding to $\mathcal{A}_{sp}$ is full. Hence, the system (13) must be solved by using an iterative method. Preconditioned iterative methods have been successfully applied to the spectral-collocation systems with pre-conditioners based on either the finite difference approximations [12, 10] or finite elements approximations [5, 3]. Here we propose to use $\mathcal{H}_{sp}$ defined by

$$(\mathcal{H}_{sp}v, w) \equiv (\nabla v, \nabla w) + \alpha(v, w)$$

(for an appropriate $\alpha \geq 0$) as the preconditioner for $\mathcal{A}_{sp}$. This type of pre-conditioners was used in the finite difference context by Concus and Golub [4], and the convergence rate of a iterative scheme for (13) with this type of preconditioners is independent of the discretization parameter N. The preconditioning equation (i.e. $\mathcal{H}_{sp}v = g$) can be efficiently solved in $O(N)$ operations for $d = 1$ (see Section 2) and in $O(N^d(\log_2 N)^{d-1})$ operations for d =2 and 3 (see [15]). Since the evaluation of $\mathcal{A}_{sp}w$ for $w \in X_N$ can be done in $O(N^d \log_2 N)$ operations (see below), the equation (11) can be solved in general in $O(N^d(\log_2 N)^{d-1})$ operations.

Because of the pseudo-spectral treatment of the term pv_N, $\mathcal{A}_{sp}$ is not exactly symmetric. However, it is indeed a spectrally accurate approximation to the symmetric spectral operator $\tilde{\mathcal{A}}_{sp}$ defined by

$$(\tilde{\mathcal{A}}_{sp}v, w) = (\nabla v, \nabla w) + (p(\boldsymbol{x})v, w), \; \forall \, v, w \in X_N.$$

The numerical experiments indicate that preconditioned conjugate gradient type methods (see for instance [8]) can be applied to $\mathcal{A}_{sp}$ and converge significantly faster than preconditioned Richardson or preconditioned minimum residual methods.

The sacrifice for the exact symmetry is compensated by the fact that $\mathcal{A}_{sp}w$ can be efficiently evaluated. More precisely, given the coefficients of $w \in X_N$, we evaluate the action of $\mathcal{A}_{sp}w$ in X_N as follows (with the operation counts of each step in parentheses):

1. Compute the Legendre coefficients of $-\Delta w$; $(O(N^d))$
2. Perform the forward Chebyshev-Legendre transform (from the Legendre coefficients to the values at the CGL points) of w; $(O(N^d \log_2 N))$
3. Compute $p(\boldsymbol{x})w(\boldsymbol{x})$ at the CGL points and then take the backward Chebyshev-Legendre transform of $I_N^c(p(\boldsymbol{x})w(\boldsymbol{x}))$; $(O(N^d \log_2 N))$
4. Compute the action of $\mathcal{A}_{sp}w = -\Delta w + I_N^c(p(\boldsymbol{x})w(\boldsymbol{x}))$ in X_N. $(O(N^d))$

The total cost, dominated by the cost of the two Chebyshev-Legendre transforms, is of order $O(N^d \log_2 N)$.

4 Numerical results

In this section, we present and compare some numerical experiments on Legendre-Galerkin (LG), Chebyshev-Galerkin (CG) and Chebyshev-Legendre Galerkin (CLG) methods. All computations are performed in double precision on a SUN-Sparcl0 workstation Model-30 with standard optimization option "$-$O". All CPU times listed are in seconds. We first compare the costs of four different transforms:

1. Chebyshev transform (CT).
2. Legendre transform (LT). This is done in an optimal way: we pre-compute and store the transform matrix, and then use Fortran BLAS routine dgemm.f for the matrix-matrix product.
3. Chebyshev-Legendre transform I (CLT-I) by the $(\frac{5}{2}\log_2 N + 4N) + \frac{1}{2}N^2$ algorithm in Section 3.
4. Chebyshev-Legendre transform II (CLT-II) by the $(\frac{5}{2}\log_2 N + 4N) + O(N)$ algorithm in [1] with 16-digit accuracy.

FFTPACK routine cost.f (available at NETLIB) is used for the real cosine transform in CT, CLT-I and CLT-II. The

N	CT	LT	CLT-I	CLT-II
	1-D transform			
32 (1000)	.09	.23	0.17	*
64 (1000)	.16	1.04	0.59	.95
128 (1000)	.31	4.05	1.87	2.38
256 (1000)	.60	15.92	6.49	5.61
512 (100)	.13	6.32	2.41	1.46
1024 (100)	.36	25.81	9.51	3.36
	2-D transform			
16 (100)	.11	.21	.18	*
32 (100)	.43	1.45	0.93	*
64 (100)	2.47	14.74	6.59	12.29
128 (10)	.91	11.26	4.15	6.29
256 (10)	3.99	88.33	29.06	30.36
512 (1)	2.08	69.94	21.67	16.07

Table 1: CPU time for the three transforms.

N	LG	CLG	CG
32	0.04 (0.01)	0.04 (0.01)	0.03 (0.04)
64	0.36 (0.06)	0.2 (0.06)	0.14 (0.19)
128	2.75 (0.14)	1.39 (0.14)	1.64 (1.99)
256	20.13 (0.54)	8.53 (0.54)	11.93 (17.99)
512	151.87 (2.04)	45.27 (2.04)	118.13 (167.36)

Table 2: CPU time for the three Poisson solvers.

CPU time for the three transforms (excluding the initialization process) are tabulated in Table I where the number in parentheses is the number of transforms made. Notice that in the 2-D case, CLT-I is more efficient than LT for N as low as 16 and is three times faster than LT for $N = 256$, and CLT-I is more efficient than CLT-II for $N \leq 256$ (CLT-II may become more competitive if single precision is used [1]). Furthermore, CLT are more accurate than LT for N large, since LT may suffer from the round-off errors from the computation of LGL points (see Table 6 in [1] for more details). The CPU time for 3-D transforms behaves similarly as the 2-D transforms.

We now compare the costs of solving a 2-D Poisson equation by using the CG, LG and CLG methods. The CPU time for the initialization process (such as computing the eigenvectors of the 1-D second-order problem) is given in the parentheses. Evidently, CLG is more efficient than LG in all cases. Furthermore, CLG is comparable to the very efficient CG method for $N \leq 64$ and CLG becomes significantly more efficient than CG for $N \geq 128$.

Next, we report on the computational results for the 2-D non-separable equation (11). Two test problems are considered:

Example 1. $a(x,y) = \left(1+\alpha((x+1)^4+(y+1)^4)\right)^2$ and $b(x,y)=0$. The function $a(x,y)$ has very large variation over the domain but after the change of depedent variable $v = \sqrt{a}u$, the function $p(x,y)$ in (12) is still nonnegative and has much less variation than $a(x,y)$. Hence, fast convergence rate is expected for a wide range of α.

Example 2. $a(x,y) = (1.5+\sin(\alpha(x+y)))^2$ and $b(x,y)=0$. In this case, $p(x,y)$ in (12) is no longer a positive function and we have $-\frac{4}{5}\alpha^2 \leq p(x,y) \leq 4\alpha^2$. The system (12) is still positive definite because (11) is. But the convergence rate may deteorate for large α due to the nonpositivity of $p(x,y)$.

The two problems are solved by three different schemes:

1. (CL-PCG) the preconditioned conjugate gradient method applied to (13);

2. (CL-PCGS) the preconditioned conjugate gradient squared method [16] applied to (13);

3. (C-PCGS) the preconditioned conjugate gradient squared method applied to a Chebyshev-Galerkin approximation to (12)(similar to (13), but is formulated with a weighted inner product).

In Table 3, we list the number of iterations needed for 7-digit accuracy. A few remarks are in order. Firstly, all three schemes for the first example converge very rapidly, even with a very large α. The slower convergence for Example 2 with $\alpha = 5$ is attributed to the nonpositivity of $p(x,y)$. Note however that even though $-20 \leq p(x,y) \leq 100$ at $\alpha = 5$, the three schemes still converge with a relatively small number of interations. This is a strong indication that these interative schemes applied to spectral-Galerkin formulations are very robust. Secondly, even though the system (13) is not exactly symmetric, CL-PCG still converges very rapidly, although at a rate slower than that of CL-PCGS. However, the slower convergence rate is more than compensated by the fact that one iteration of PCGS costs twice as much as one iteration of PCG. Finally, the cost of one iteration of PCG (resp. PCGS) is about the same as the cost of solving one (resp. two) Poisson equation(s) by the respective method (cf. Table 2).

In summary, we have developed a very efficient Chebyshev-Legendre Galerkin method with quasi-optimal operation counts for solving elliptic equations.

Acknowledgments. The author would like to thank Dr. Alpert for providing his program for the Chebyshev-Legendre transform.

	CL-PCG	CL-PCGS	C-PCGS
N	Example 1:	$\alpha = 10, 1000$	
16	8, 10	5, 6	5, 7
32	8, 9	5, 6	5, 7
64	8, 8	5, 6	5, 7
	Example 2:	$\alpha = 2, 5$	
16	8, 23	5, 13	5, 14
32	8, 18	5, 13	5, 13
64	8, 17	5, 13	5, 13

Table 3: Number of iterations required for 7-digit accuracy.

References

[1] B.K. Alpert and V. Rokhlin. A fast algorithm for the evaluation of Legendre expansions. *SIAM J. Sci. Stat. Comput.*, 12:158–179, 1991.

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

[3] C. Canuto and A. Quarteroni. Preconditioner minimal residual methods for Chebyshev spectral calculations. *J. Comput. Phys.*, 60:315–337, 1985.

[4] P. Concus and G. H. Golub. Use of fast direct methods for the efficient numerical solution of nonseparable elliptic equations. *SIAM J. Numer. Anal.*, 10:1103–1120, 1973.

[5] M. O. Deville and E. H. Mund. Finite element preconditioning for pseudospectral solutions of elliptic problems. *SIAM J. Sci. Stat. Comput.*, 11:311–342, 1990.

[6] W.S. Don and D. Gottlieb. The Chebyshev-Legendre method: implementing Legendre methods on Chebyshev points. *SIAM J. Numer. Anal.*, 31:1519–1534, 1994.

[7] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[8] G. Golub and C. Van Loan. *Matrix Computations, second edition.* Johns Hopkins Press, Baltimore, 1989.

[9] D. B. Haidvogel and T. A. Zang. The accurate solution of Poisson's equation by expansion in Chebyshev polynomials. *J. Comput. Phys.*, 30:167–180, 1979.

[10] 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.

[11] R. E. Lynch, J. R. Rice, and D. H. Thomas. Direct solution of partial differential equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.

[12] S. A. Orszag. Spectral methods for complex geometries. *J.Comput.Phys.*, 37:70–92, 1980.

[13] Jie Shen. Efficient spectral-Galerkin method i. direct solvers for second- and fourth-order equations by using Legendre polynomials. *SIAM J. Sci. Comput.*, 15:1489–1505, 1994.

[14] 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.

[15] Jie Shen. On fast Poisson solver, inf-sup constant and iterative Stokes solver by Legendre Galerkin method. *J. Comput. Phys.*, 116:184–188, 1995.

[16] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 10:36–32, 1989.