NOTE

On Fast Direct Poisson Solver, INF-SUP Constant and Iterative Stokes Solver by Legendre–Galerkin Method*

1. INTRODUCTION

In a sequence of recent works [12, 11], we have presented efficient direct solvers, based on the Legendre– and Chebyshev–Galerkin methods, for the second- and fourth-order elliptic equations with constant coefficients. The complexity of these direct solvers is a small multiple of N^{d+1}, where $d = 2$ or 3 and N is the cutoff number of the polynomial expansion in each direction. These direct solvers were all based on the matrix decomposition method and, hence, did not take full advantage of the special structures of the matrices obtained from the Legendre–Galerkin discretization. We shall see that in the two-dimensional case, more efficient algorithms can be constructed by further exploring the matrix structures.

As the title suggests, the aim of this note is twofold: (i) we shall present a fast direct 2D Poisson solver, based on the Legendre–Galerkin approximation, whose complexity is of order $O(N^2 \log_2 N)$ (where N is the cutoff number of the Legendre expansion in each direction); (ii) we shall study numerically the asymptotic behavior of the inf-sup constants for a sequence of the discretized Stokes systems. Our results indicate in particular that the iterative Stokes solver, more precisely the conjugate gradient Uzawa algorithm, has a complexity of order $O(N^5/2 \log_2 N)$ for a sequence of discretized 2D Stokes systems. Since the convergence rate of the Legendre–Galerkin approximations is exponential for problems with smooth solutions, the algorithms presented below should be very competitive for the specified problems.

2. FAST DIRECT POISSON SOLVER

Let $\Omega = [-1, 1]^2$ and $L_n(x)$ be the nth-degree Legendre polynomial. We denote

$$S_N = \operatorname{span}\{L_i(x)L_j(y) : i, j = 0, 1, \ldots, N\},$$
$$X_N = \{v \in S_N : v|_{\partial\Omega} = 0\}.$$

Then the standard Legendre–Galerkin approximation to the two-dimensional Helmholtz equation

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

is to find $u_N \in X_N$ such that

$$\alpha(u_N, v) + (\nabla u_N, \nabla v) = (f, v), \forall v \in X_N, \tag{2.2}$$

where $(u, v) = \int_\Omega uv \, dx \, dy$ is the scalar product in $L^2(\Omega)$.

It is easy to verify that (see Lemma 2.1 in [12])

$$X_N = \operatorname{span}\{\phi_i(x)\phi_j(y) : i, j = 0, 1, \ldots, N-2\},$$

where $\phi_k(\cdot) = (1/\sqrt{4k+6})\,(L_k(\cdot) - L_{k+2}(\cdot))$. Furthermore, setting

$$a_{jk} = \int_{-1}^{1} \phi_k'(x)\phi_j'(x)\, dx, \quad b_{jk} = \int_{-1}^{1} \phi_k(x)\phi_j(x)\, dx,$$

then we have

$$a_{jk} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases},$$

$$b_{jk} = b_{kj} = \begin{cases} c_k c_j\left(\dfrac{2}{2j+1} + \dfrac{2}{2j+5}\right), & k = j \\ -c_k c_j \dfrac{2}{2k+1}, & k = j+2 \\ 0, & \text{otherwise}, \end{cases} \tag{2.3}$$

where $C_k = (1/\sqrt{4k+6})$.

Let us now 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

$$U = (u_{kj})_{k,j=0,1,\ldots,N-2}, \quad F = (f_{kj})_{k,j=0,1,\ldots,N-2},$$
$$B = (b_{kj})_{k,j=0,1,\ldots,N-2}.$$

* This work was supported in part by NSF Grant DMS-9205300.

0021-9991/95 $6.00

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

$$\alpha BUB + UB + BU = F, \tag{2.4}$$

which can also be written in the tensor product form

$$(\alpha B \otimes B + \mathrm{I} \otimes B + B \otimes I)\mathbf{u} = \mathbf{f}, \tag{2.5}$$

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

$$\mathbf{f} = (f_{00}, f_{10}, \ldots, f_{q0}; f_{01} \ldots, f_{q1}; \ldots; f_{0q}, \ldots, f_{qq})^{\mathrm{T}}, \tag{2.6}$$

where $q = N - 2$ and $\otimes$ denotes the tensor product of matrices, i.e., $A \otimes B = (Ab_{ij})_{i,j=0,1,\ldots,q}$. Note that we generally use a capital letter to denote a matrix and a boldface letter to denote a vector.

Since B has alternating zero and nonzero elements, the system (2.5) can be split into four subsystems. Namely, Let $I^{(1)}$ (resp. $I^{(2)}$) be the identity matrix of order $[N/2]$ (resp. $[N/2] - 1$); let

$$b_{ij}^{(1)} = b_{2i,2j}, \qquad B^{(1)} = (b_{ij}^{(1)})_{i,j=0,1,\ldots,[N/2]},$$
$$b_{ij}^{(2)} = b_{2i-1,2j-1}, \quad B^{(2)} = (b_{ij}^{(2)})_{i,j=0,1,\ldots,[N/2]-1}.$$

Thanks to (2.3), $B^{(1)}$ and $B^{(2)}$ are both symmetric tridiagonal matrices. Let $\mathbf{u}^{(e,e)}$, $\mathbf{u}^{(e,o)}$, $\mathbf{u}^{(o,e)}$, and $\mathbf{u}^{(o,o)}$ (resp. $\mathbf{f}^{(e,e)}$, $\mathbf{f}^{(e,o)}$, $\mathbf{f}^{(o,e)}$, and $\mathbf{f}^{(o,o)}$) consist of respectively the even–even, even–odd, odd–even, and odd–odd components of U (resp. F). Then the four subsystems are

$$\begin{aligned}
(\alpha B^{(1)} \otimes B^{(1)} + I^{(1)} \otimes B^{(1)} + B^{(1)} \otimes I^{(1)})\mathbf{u}^{(e,e)} &= \mathbf{f}^{(e,e)},\\
(\alpha B^{(1)} \otimes B^{(2)} + I^{(1)} \otimes B^{(2)} + B^{(1)} \otimes I^{(2)})\mathbf{u}^{(e,o)} &= \mathbf{f}^{(e,o)},\\
(\alpha B^{(2)} \otimes B^{(1)} + I^{(2)} \otimes B^{(1)} + B^{(2)} \otimes I^{(1)})\mathbf{u}^{(o,e)} &= \mathbf{f}^{(o,e)},\\
(\alpha B^{(2)} \otimes B^{(2)} + I^{(2)} \otimes B^{(2)} + B^{(2)} \otimes I^{(2)})\mathbf{u}^{(o,o)} &= \mathbf{f}^{(o,o)}.
\end{aligned} \tag{2.7}$$

It is now clear that the matrices of the four subsystems are in fact block-tridiagonal. Furthermore, in case $\alpha = 0$, they have exactly the same structure as the matrix generated by the standard five-point finite difference discretization to separable second-order elliptic equations with variable coefficients (cf. [14]). Hence in particular the four subsystems can be solved by the extended cyclic reduction method developed by Swarztrauber [14] in $O(N^2 \log_2 N)$ operations. In fact, the subroutine blktri.f in FISHPACK (written by J. Adams, P. N. Swarztrauber, and R. Sweet, available via netlib) can be directly used to solve the above subsystems.

Remark 1. In case $\alpha \neq 0$, each of the systems (2.7) correspond to a block-tridiagonal matrix in which each block is a tridiagonal matrix. Hence blktri.f cannot be directly used in this case. However, we note that (see (2.18) in [14]) the structure of the systems with $\alpha \neq 0$ is exactly the same as that of the systems obtained after one cyclic reduction to the systems with $\alpha = 0$. Therefore the systems (2.7) with $\alpha \neq 0$ can still be solved in $O(N^2 \log_2 N)$ operations.

We now compare in Table I the efficiency of the above cyclic reduction algorithm (denoted by the first solver), whose operation count is about $32N^2 \log_2 N - 11N^2$ (cf. Section 4 in [14]), with the direct Poisson solver (denoted by the second solver), based on the matrix decomposition method presented in [12], whose operation count is about $2N^3$ (cf. Section 2.2 in [12]). Note that only the CPU time in seconds for solving the system (2.4) is reported. The preprocessing CPU time in seconds is reported in parentheses. All the computations here and later are performed on a Sun Sparc-10 workstation in double precision, except the case $N = 32$ below for which the computation is done on a Sun Sparc-1$^+$ to highlight the comparison.

One concludes from Table I that for $N \lesssim 80$ the direct solver in [12] is more efficient than the present direct solver; and that for $N > 80$ the latter becomes more efficient.

3. INF-SUP CONSTANT AND ITERATIVE STOKES SOLVER

We shall take advantage of the fast Poisson solver developed above to construct an efficient algorithm for solving the 2D Stokes equations:

$$\begin{gathered}
-\Delta\mathbf{u} + \nabla p = \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega = [-1, 1]^2,\\
u|_{\partial\Omega} = 0, \quad \int_\Omega p\,dx\,dy = 0.
\end{gathered} \tag{3.1}$$

We shall consider a sequence ($m = N, N-1, N-2, \ldots$) of Legendre–Galerkin approximations for (3.1): Find $(\mathbf{u}_{N,m}, p_{N,m}) \in \mathbf{X}_N \times S_m \cap L_0^2(\Omega)$ such that

$$\begin{aligned}
(\nabla\mathbf{u}_{N,m}, \nabla\mathbf{v}) - (\nabla \cdot \mathbf{v}, P_{N,m}) = (\mathbf{f}, \mathbf{v}), \quad &\forall \mathbf{v} \in \mathbf{X}_N;\\
(\nabla \cdot \mathbf{u}_{N,m}, q) = 0, \quad &\forall q \in S_m \cap L_0^2(\Omega),
\end{aligned} \tag{3.2}$$

where $\mathbf{X}_N = X_N \times X_N$ and $L_0^2(\Omega) = \{q \in L^2(\Omega) : \int_\Omega q\,dx\,dy = 0\}$.

Remark 2. The case $m = N$ was first studied in [3] in the context of the collocation method. The case $m = N - 1$ was first used [2] in the context of the collocation method on staggered grids. The case $m = N - 2$ was first introduced in [13] in the context of the tau method and in [10] in the context of the spectral element method.

It is well known that (3.2) admits a unique solution if and only if the set

$$Z_{N,m} = \{q \in S_m \cap L_0^2(\Omega) : (\nabla \cdot \mathbf{u}, q) = 0, \forall \mathbf{u} \in \mathbf{X}_N\}$$

is reduced to $\{\mathbf{0}\}$. However, the following results are now well known (see, for instance, [3]).

TABLE I

CPU Comparison of Two Direct Poisson Solvers

N	32[a]	64	80	128	160	256
1st solver	0.12(0.07)	0.11(0.04)	0.18(0.07)	0.55(0.14)	0.88(0.22)	2.78(0.54)
2nd solver	0.06(0.04)	0.09(0.05)	0.17(0.09)	0.99(0.38)	1.33(0.72)	7.63(2.64)

[a] Sun Sparc-1^+ is used in this case.

LEMMA 1.

$$
\begin{aligned}
Z_{N,N} &= \text{span}\{L_N(x), L_N(y), L_N(x)L_N(y),\\
&\qquad L'_{N+1}(x)L'_{N+1}(y), L'_N(x)L'_{N+1}(y),\\
&\qquad L'_{N+1}(x)L'_N(y), L'_N(x)L'_N(y)\};\\
Z_{N,N-1} &= \text{span}\{L'_N(x)L'_N(y)\};\\
Z_{N,m} &= \{\mathbf{0}\}, \quad \forall 0 \le m \le N-2.
\end{aligned}
$$

Hence let $M_{N,m}$ be a supplement of $Z_{N,m}$ in $S_m \cap L_0^2(\Omega)$, i.e., $M_{N,m} \oplus Z_{N,m} = S_m \cap L_0^2(\Omega)$. Then by the construction of $M_{N,m}$, the new approximation,

$$
\begin{aligned}
(\nabla \mathbf{u}_{N,m}, \nabla \mathbf{v}) - (\nabla \cdot \mathbf{v}, p_{N,m}) &= (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{X}_N;\\
(\nabla \cdot \mathbf{u}_{N,m}, q) &= 0, \quad \forall q \in M_{N,m}, \qquad (3.3)
\end{aligned}
$$

admits a unique solution in $\mathbf{X}_N \times M_{N,m}$. Or equivalently, there exists $\beta_{N,m} > 0$ such that (see, for instance, [4, 5])

$$
\inf_{q \in M_{N,m}} \sup_{\mathbf{v} \in X_N} \frac{(\nabla \cdot \mathbf{v}, q)}{\|\mathbf{u}\|_{\mathbf{H}^1(\Omega)} \|q\|_{L^2(\Omega)}} = \beta_{N,m}. \qquad (3.4)
$$

$\beta_{N,m}$ is usually referred to as Babuska–Brezzi's inf-sup constant for the system (3.3). The behavior of the inf-sup constant $\beta_{N,m}$ is important not only because (cf. [5]) the system (3.3) admits a unique solution if and only if $\beta_{N,m} > 0$, but also because (i) the error estimate for the pressure approximation in (3.3) is optimal if and only if $\beta_{N,m}$ is independent of (N, m) and otherwise there is a loss of accuracy of the order $\beta_{N,m}^{-1}$ for the pressure approximation; (ii) the number of iterations required in the conjugate gradient Uzawa method is proportional to $\beta_{N,m}$ (see below for more details).

There have been many efforts for determining numerically and theoretically the inf-sup constant (see, for instance, [3, 10, 6–8]). Most of the theoretical results in these references provided only lower bounds for the inf-sup constants, and these lower bounds usually more or less under-estimated the inf-sup constants. However, the author was informed, when this work was complete, by S. Jensen that recently Maday *et al.* [9] succeeded in proving that $\beta_{N,N-2} \sim CN^{(1-d)/2}$. On the other hand, to the author's knowledge, the available numerical results are limited to relatively small N's ($N \lesssim 20$), which are far from reaching the asymptotic range of the inf-sup constants. We shall try to compute numerically below the inf-sup constants $\beta_{N,m}$ for a range of (N,m) large enough to exhibit clearly the asymptotic decay rate of $\beta_{N,m}$.

It is certainly very difficult to compute $\beta_{N,m}$ directly from the definition (3.4). To facilitate the task, we shall use instead a different interpretation of $\beta_{N,m}$. By using the Legendre polynomials $\{L_i(x)L_j(y) : i,\ j = 0, 1, ..., m\}$ as base functions for S_m and the combinations of Legendre polynomials $\{\phi_i(x)\phi_j(y) : i,\ j = 0, 1, ..., N-2\}$ as base functions for $\mathbf{X}_N$, we can rewrite (3.2) in the matrix form

$$
\begin{aligned}
A_N w_{N,m} + B_{N,m}^{\mathrm{T}} r_{N,m} &= \tilde{f}_N;\\
B_{N,m}, w_{N,m} &= 0, \qquad (3.5)
\end{aligned}
$$

where $w_{N,m}$ and $r_{N,m}$ denote the vectors formed by the coefficients of the aforementioned base expansions of respectively, $\mathbf{u}_{N,m}$ and $p_{N,m}$; $\tilde{f}_N$ is the vector formed by $\{\tilde{\mathbf{f}}_{ij} : i, j = 0, 1, ..., N-2\}$ with $\tilde{\mathbf{f}}_{i,j} = (\mathbf{f}, \phi_i(x)\phi_j(y))$; A_N is the discrete Laplacian operator; $B_{N,m}$ is the discrete negative divergence operator; and its transpose, $B_{N,m}^{\mathrm{T}}$, is the discrete gradient operator.

It is obvious that A_N is positive definite and, hence, invertible. Therefore we can eliminate $w_{N,m}$ from (3.5) to obtain an equation only for $r_{N,m}$,

$$
D_{N,m} r_{N,m} \equiv (B_{N,m} A_N^{-1} B_{N,m}^{\mathrm{T}}) r_{N,m} = B_{N,m} A_N^{-1} \tilde{f}_N. \qquad (3.6)
$$

Remark 3. The generalized Stokes system, obtained by replacing $-\Delta$ by the Helmholtz operator $\alpha I - \Delta$, can be treated similarly. We only have to replace A_N in (3.3) and A_N^{-1} in (3.6) by respectively the discrete Helmholtz operator and its inverse.

The matrix $D_{N,m}$ is often referred as the discretization of the Uzawa operator $-\nabla \cdot (-\Delta)^{-1} \nabla$, since the Uzawa algorithm introduced in [1] can be interpreted as a gradient method applied to the system (3.6).

Clearly $D_{N,m}$ is symmetric and at least semi-positive definite. Furthermore, we have the following results for $D_{N,m}$ (see, for instance, Chapter 2 in [5]).

LEMMA 2. *Let $\lambda_{\min}^{(N,m)}$ (resp. $\lambda_{\max}^{(N,m)}$) be the smallest positive (resp. largest) eigenvalue of $D_{N,m}$. Then*

TABLE II

The Inverse of $\lambda_{\min}^{(N,m)}$, i.e., $\mathrm{cond}(D_{N,m})$

m \ N	8	16	32	64	128	256	512	$N \gg 1$
N	105.1	416.8	1662	6642	26561	—	—	$\sim 1.62N^2$
$N-1$	6.204	8.760	13.49	22.39	39.54	73.01	142.4	$\sim 0.27N$
$N-2$	4.608	5.444	7.229	11.36	19.79	36.52	69.57	$\sim 0.13N$
$N-3$	4.548	5.210	6.803	10.91	19.36	36.08	69.06	$\sim 0.13N$

$$\lambda_{\max}^{(N,m)} = 1, \quad \lambda_{\min}^{(N,m)} = \beta_{N,m}^2;$$

and $\dim(Z_{N,m})$ *equals to the number of zero eigenvalues of* $D_{N,m}$.

Thus we can obtain $\beta_{N,m}$ by computing $\lambda_{\min}^{(N,m)}$ of $D_{N,m}$. However, due to the presence of A_N^{-1} in the definition of $D_{N,m}$, one cannot expect $D_{N,m}$ to be a sparse matrix. Hence one must try to solve the system (3.6) or to compute its eigenvalues without forming explicitly the matrix $D_{N,m}$. Note that the matrix-vector product $D_{N,m} r_{N,m}$ can be easily performed in $O(N^2 \log_2 N)$ operations, thanks to the fast direct Poisson solver developed above. Therefore the system (3.6) can be solved in particular by the conjugate gradient method, and thanks to Lemma 2, the number of iterations required in the conjugate gradient method scales as $(\lambda_{\min}^{(N,m)})^{-1/2} = \beta_{N,m}^{-1}$. Hence the efficiency of this iterative Stokes solver (note that once the solution $r_{N,m}$ of (3.6) is known, we can obtain $w_{N,m}$ from (3.3) by solving a Poisson equation) depends greatly on the behavior of $\beta_{N,m}$, or equivalently, $\lambda_{\min}^{(N,m)}$.

To compute $\lambda_{\min}^{(N,m)}$ from (3.6) without forming explicitly the matrix $D_{N,m}$, one can of course employ the power method, as is used in [8]. However, the convergence rate is usually very slow and, hence, prohibits the computation for large (N, m). A better strategy is to use the block-Lanczos algorithm for computing a few extreme eigenvalues of a large symmetric system. The algorithm usually yields the desired results with a relatively small number of matrix-vector products. The algorithm was particularly implemented by D. Scott in a public domain package LASO (available via netlib). In Table II, we present the condition numbers of $D_{N,m}$ (denoted by $\mathrm{cond}(D_{N,m})$) with $m = N, N-1, N-2, N-3$ as computed by using the subroutine dnlaso.f in LASO.

We note that for $m = N$, the computational asymptotic rate $\beta_{N,N} = \dfrac{1}{\sqrt{1.62N}}$ is very accurate for all N reported. It also suggests that the pair $\mathbf{X}_N \times S_N$ is not suitable for use with the iterative Uzawa scheme, since $\mathrm{cond}(D_{N,N}) \sim 1.62N^2$ is too large. On the other hand, for $m \leq N-1$, the asymptotic rates for $\beta_{N,m}$ are all of order $N^{-1/2}$. Furthermore, the inf-sup constants $\beta_{N,m}$ (for $m \leq N-1$) do not enter the asymptotic range until $N \sim 512$; for small N, their decay rates are significantly slower, as noted also in [8]. One should also note that

$$\mathrm{cond}(D_{N,m-3}) \approx \mathrm{cond}(D_{N,m-2});$$

$$\mathrm{cond}(D_{N,m-1}) \approx 2\,\mathrm{cond}(D_{N,m-2}), \quad \text{for } N > 32.$$

This suggests that this is of no interest to use $m \leq N-3$ and that $m = N-2$ is probably the best choice if one uses the conjugate gradient Uzawa scheme to solve the discrete system (3.3). But when a small N is used, such as in the spectral element applications, the case $m = N-1$ is also worth considering since it leads to a relatively more accurate pressure approximation.

We finally note that taking into account the complexity of the fast Poisson solver and the asymptotic growth rates of $\beta_{N,m}^{-1}$, the conjugate gradient Uzawa scheme for the discrete Stokes system with $m \leq N-1$ is of complexity $O(N^{5/2}\log_2 N)$. We mention that often the number of iterations required in the conjugate gradient Uzawa scheme grows significantly slower than $\beta_{N,m}^{-1}$. In Table III, we have tabulated the number of conjugate gradient iterations required for seven-digit accuracy for two concrete examples in the case $m = N-2$:

Ex. 1. $f(x, y)$ is chosen so that the exact solution $P(x, y) = \cos(\pi x)\cos(\pi y)$;

Ex. 2: $f(x, y)$ is a uniformly distributed random function.

In both cases, the starting guess was taken to be zero.

Remark 4. The algorithm in this section can be naturally extended to the three-dimensional case. However, the decay rate of the inf-sup constant in the 3D case is significantly faster than in the 2D case. To illustrate the situation, we have computed $\mathrm{cond}(D_{N,m})$ (see Table IV) for the more interesting case, $m = N-2$. Although we were not able to reach the asymptotic range because of the limitation of the computing

TABLE III

Number of Conjugate Gradient Iterations

N	8	16	32	64	128	256
Ex. 1	9	10	12	12	13	15
Ex. 2	13	13	15	15	16	17

TABLE IV

cond($D_{N,N-2}$) in 3D Case

N	8	16	32	40	64	80	128
cond($D_{N,N-2}$)	11.34	22.01	52.64	72.41	149.2	208.4	469.1

resource, we observe that for N increases from 8 up to 128, cond($D_{N,N-2}$) is steadily growing toward the theoretical asymptotic rate $O(N^2)$. This implies that the number of conjugate gradient iterations required grows like $O(N)$. Hence, this iterative Stokes solver in the 3D case is not as attractive as in the 2D case, especially when a large N has to be used, although it is probably still the most viable technique at the present time for solving the 3D Stokes system with a spectral discretization.

CONCLUDING REMARKS

We have presented in this paper a fast Poisson solver and an iterative Stokes solver, based on the Legendre–Galerkin approximations, whose complexities are respectively $O(N^2 \log_2 N)$ and $O(N^{5/2} \log_2 N)$ in a two-dimensional rectangular domain. Taking into account the spectral accuracy of the Legendre–Galerkin approximations, we conclude that these algorithms are very valuable and competitive for the specified problems.

We have also computed numerically the inf-sup constants of a sequence of discretized Stokes systems for a large range of (N, m). The results exhibit not only the quantitative but also the qualitative asymptotic behavior of the inf-sup constants. The results may serve, in particular, as a reference for users of spectral methods to choose an appropriate pair of discretization spaces for the velocity and the pressure in the Stokes problem.

REFERENCES

1. K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Nonlinear Programming* (Stanford Univ. Press, Stanford, 1958).
2. C. Bernardi and Y. Maday, *Int. J. Numer. Methods Fluids* **8,** 537 (1988).
3. C. Bernardi, Y. Maday, and B. Métivet, *Rech. Aerosp.* (1987).
4. F. Brezzi, *RAIRO Anal. Numer.* **8-R2,** 129 (1974).
5. F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods* (Springer-Verlag, New York/Heidelberg/Berlin, 1991).
6. S. Jensen, *Numer. Math.* **59,** 581 (1991).
7. S. Jensen and M. Vogelius, *Model. Math. Anal. Numer.* **24,** 737 (1990).
8. Y. Maday, D. Meiron, A. T. Patera, and E. M. Ronquist, *SIAM J. Sci. Comput.* **14,** 310 (1993).
9. Y. Maday, A. T. Patera, and E. M. Ronquist, The $P_N \times P_{N-2}$ method for the approximation of the Stokes problem, 1992, preprint.
10. E. M. Ronquist, Ph.D. thesis, MIT, 1988 (unpublished).
11. J. Shen, *SIAM J. Sci. Comput.* **16,** No. 1 (1995).
12. J. Shen, *SIAM J. Sci. Comput.* **15,** No. 6 (1994).
13. J. Shen, *Math. Model. Num. Anal.* **22,** 677 (1988).
14. P. N. Swarztrauber, *SIAM J. Numer. Anal.* **11,** 1136 (1974).

Received July 27, 1993

JIE SHEN

Department of Mathematics
Pennsylvania State University
University Park, Pennsylvania 16802