Computers Math. Applic. Vol. 35, No. 11, pp. 75–82, 1998

0898-1221/98 $19.00 + 0.00

PII: S0898-1221(98)00085-6

Direct Methods for Spectral Approximations in Nonconforming Domain Decompositions

A. KARAGEORGHIS
Department of Mathematics and Statistics, University of Cyprus
P.O. Box 537, 1678 Nicosia, Cyprus

M. PAPRZYCKI
Department of Mathematics and Computer Science
University of Texas of the Permian Basin, Odessa, TX 79762, U.S.A.

(Received August 1997; accepted October 1997)

Abstract—The application of a conforming spectral collocation method to certain nonconforming domain decompositions leads to global matrices which have a particular block structure. We study the performance of various direct linear system solvers, some of which exploit this block structure, on a Cray J-916 vector computer, an SGI Power Challenge 8000, and an IBM RS6000 workstation.

Keywords—Spectral methods, Direct solvers, High performance computing.

1. INTRODUCTION

In this paper, we study the efficiency of various direct methods for the solution of the global systems resulting from spectral approximations for certain domain decompositions. In particular, we examine the systems resulting from conforming spectral approximations in nonconforming domain decompositions in rectangular domains, developed in [1]. When we say that the domain decompositions are nonconforming, we mean that the rectangular domains are divided into an odd number of subdomains whose interfaces are nonconforming in size. The spectral approximations which are used are conforming, that is, the solution is C^0 continuous at all points across the subdomain interfaces for second-order problems and C^1 continuous at all points across the subdomain interfaces for fourth-order problems. This type of decomposition is particularly useful when dealing with problems which contain boundary singularities (see, e.g., [2]). The matrices resulting from these approximations possess a particular block diagonal structure. This structure is exploited by using a capacitance-type technique [3], a banded system solver from the NAG Library [4], two versions of the sparse system solver UMFPACK [5–7], and combinations of the above.

2. DOMAIN DECOMPOSITION AND SPECTRAL APPROXIMATION

We consider the problem

$$\nabla^2\phi(x,y) = F(x,y), \qquad \text{on the rectangle } (\alpha,\beta)\times(a,b), \tag{2.1}$$

The computer time grant from the NCSA at Urbana-Champaign is kindly acknowledged.

subject to Dirichlet boundary conditions. We shall assume that the expressions for the boundary conditions are analytic and that they are well approximated by their truncated Taylor series. We shall therefore assume, henceforth, that these functions are polynomials.

For the partitions $\alpha = \alpha_0 < \alpha_1 < \alpha_2 < \cdots < \alpha_{N-1} < \alpha_N = \beta$ and $a = a_0 < a_1 < a_2 < \cdots < a_{N-1} < a_N = b$, $N \in \mathbb{N}$, we consider the decomposition D_{2N-1}: the rectangle $(\alpha, \beta) \times (a, b)$ is decomposed into $2N-1$ subdomains in the following way: for $k = 1, 2, \ldots, N-1$, subdomain $2k-1$ is the rectangle $(\alpha_{k-1}, \alpha_k) \times (a_{k-1}, a_N)$ and subdomain $2k$ is the rectangle $(\alpha_k, \alpha_N) \times (a_{k-1}, a_k)$. Subdomain $2N-1$ is the rectangle $(\alpha_{N-1}, \alpha_N) \times (a_{N-1}, a_N)$.

In each subdomain, the solution is approximated by

$$\phi_s(x, y) = \sum_{m=0}^{M_s} \sum_{n=0}^{N_s} \gamma_{mn}^s \hat{T}_m^s(x) \tilde{T}_n^s(y), \qquad s = 1, 2, \ldots, 2N-1, \tag{2.2}$$

where the functions $\hat{T}_m^s(x)$ and $\tilde{T}_n^s(y)$ are the shifted Chebyshev polynomials defined on the corresponding intervals of each region and the collocation points on each interval of each region (e.g., $\{x_i^s\}_{i=0}^{M_s}$) are the Gauss-Lobatto points [8,9]. We shall assume that $M_{2k} \leq \min\{M_{2k+1}, M_{2k+2}\}$ and that $N_{2k-1} \leq \min\{N_{2k}, N_{2k+1}\}$, $k = 1, 2, \ldots, N-1$. For the above problem and the domain decomposition D_{2N-1} ($N \in \mathbb{N}$), it can be shown that the collocation formulation of the spectral approximation (2.2), with the appropriate interface continuity conditions for the solution and its normal derivative, yields C^0 conforming approximations on all the subdomain interfaces (see [1]).

3. METHODS OF SOLUTION

3.1. Capacitance-Type Technique

The structure of the global matrix for the multidomain decomposition (the five element case) is of the form given in Figure 1.

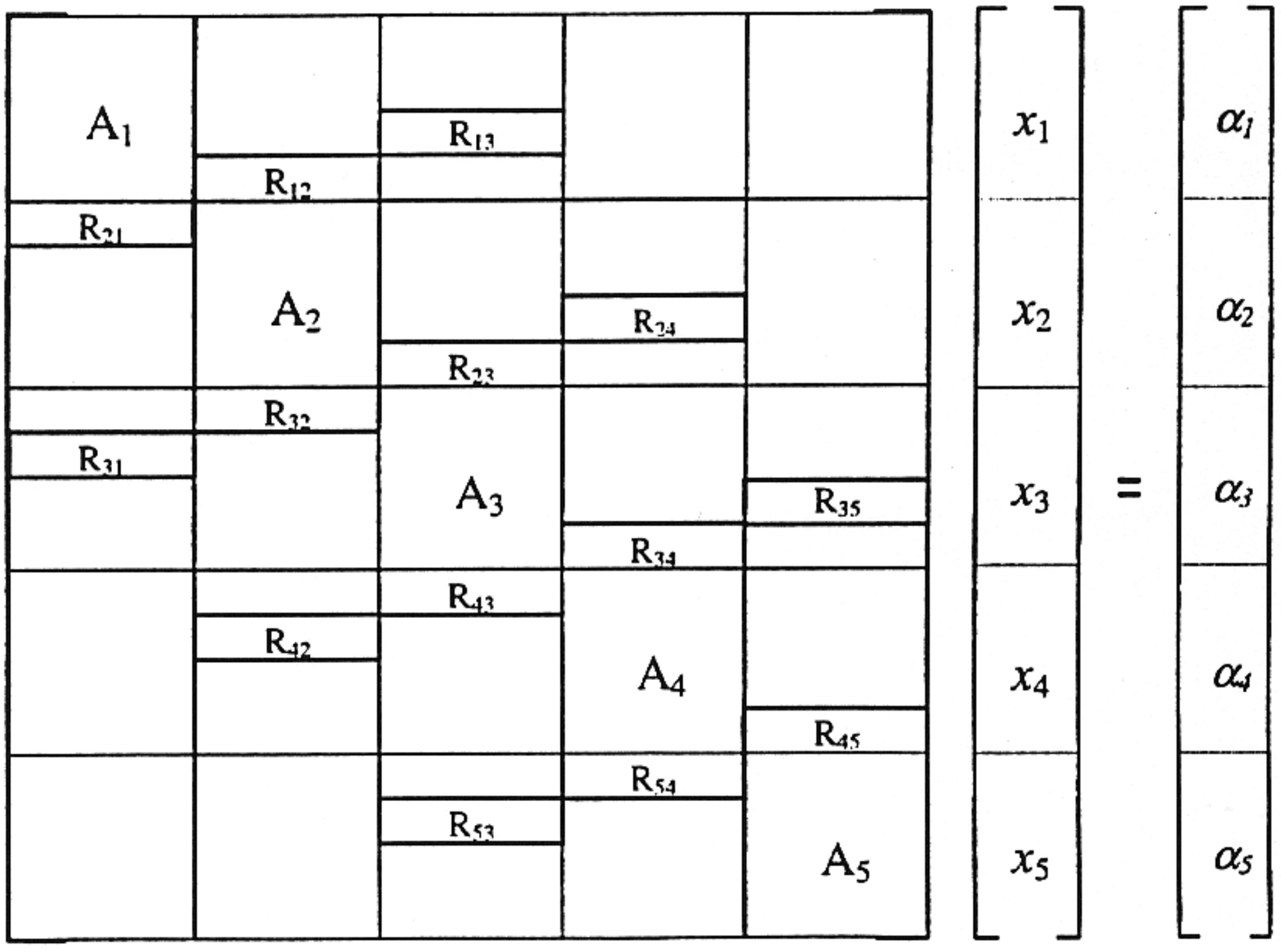

Figure 1. The linear system resulting from a five-element decomposition.

If we take $k_s = (M_s + 1)(N_s + 1), s = 1, 2, \ldots, L\ (= 2N - 1)$, the global matrix has dimension $\left(\sum_{i=1}^{L} k_i\right)^2$. The vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ contain the unknown coefficients in regions $1, 2, \ldots, L$, respectively. The matrices A_s, $s = 1, 2, \ldots, L$, have dimensions k_s, $s = 1, 2, \ldots, L$, respectively, and their rows correspond to the satisfaction of the boundary conditions, the governing equation and interface conditions in each region. The matrices R_{ij}, $i = 1, 2, \ldots, L$, $j = 1, 2, 3, 4$, correspond to the satisfaction of the interface conditions: R_{i1} corresponds to the satisfaction of the interface conditions between elements $i - 2$ and i, R_{i1} to the interface conditions between elements $i - 1$ and i, R_{i3} between $i+1$ and i, and R_{i4} between $i+2$ and i. Their dimensions are: R_{i1} is $l_1 \times k_{i-2}$, R_{i2} is $l_2 \times k_{i-1}$, R_{i3} is $l_3 \times k_{i+1}$, and R_{i4} is $l_4 \times k_{i+2}$, where $l_1 = M_i$, $l_2 = N_i$, and $l_3 + l_4 = M_i - 1$ if i is even, and $l_1 = N_i$, $l_2 = M_i$, and $l_3 + l_4 = N_i - 1$ if i is odd. Further, R_{L2} is $M_L - 1$ (or $N_L - 1$) $\times k_{L-1}$.

By denoting $R^*_{i1} = \left[\frac{0}{R_{i1}}\right]$, $R^*_{i2} = \left[\frac{R_{i2}}{0}\right]$, $R^*_{i3} = \left[\frac{0}{R_{i3}}\right]$, and $R^*_{i4} = \left[\frac{0}{R_{i4}}\right]$, $i = 1, 2, \ldots, L$, the global system may be rewritten as

$$A_1\mathbf{x}_1 + R^*_{12}\mathbf{x}_2 + R^*_{13}\mathbf{x}_3 = \alpha_1, \tag{3.1}$$

$$R^*_{21}\mathbf{x}_1 + A_2\mathbf{x}_2 + R^*_{23}\mathbf{x}_3 + R^*_{24}\mathbf{x}_4 = \alpha_2, \tag{3.2}$$

$$R^*_{31}\mathbf{x}_1 + R^*_{32}\mathbf{x}_2 + A_3\mathbf{x}_3 + R^*_{34}\mathbf{x}_4 + R^*_{35}\mathbf{x}_5 = \alpha_3, \tag{3.3}$$

$$R_{L-1,L-3}\mathbf{x}_{L-3} + R^*_{L-1,L-2}\mathbf{x}_{L-2} + A_{L-1}\mathbf{x}_{L-1} + R^*_{L-1,L}x_L = \alpha_{L-1}, \tag{3.L-1}$$

$$\vdots \qquad\qquad \vdots$$

$$R^*_{L,L-2}\mathbf{x}_{L-2} + R_{L,L-1}\mathbf{x}_{L-1} + A_L x_L = \alpha_L. \tag{3.L}$$

From (3.1) and (3.L), we may express $\mathbf{x}_1$ and $\mathbf{x}_L$ in terms of $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_{L-2}$, $\mathbf{x}_{L-1}$, respectively. We then substitute these expressions into (3.2)–(3.$L-1$), thus obtaining a system in terms of the unknown vectors $\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{L-1}$. This process is repeated until the system is reduced to a system of the form:

$$\bar{A}_{N-1}\mathbf{x}_{N-1} + \bar{R}_{N-1,N}\mathbf{x}_N + \bar{R}_{N-1,N+1}\mathbf{x}_{N+1} = \bar{\alpha}_{N-1}, \tag{3.L+1}$$

$$\bar{R}_{N,N-1}\mathbf{x}_{N-1} + \bar{A}_N\mathbf{x}_N + \bar{R}_{N,N+1}\mathbf{x}_{N+1} = \bar{\alpha}_N, \tag{3.L+2}$$

$$\bar{R}_{N+1,N-1}\mathbf{x}_{N-1} + \bar{R}_{N+1,N}\mathbf{x}_N + \bar{A}_{N+1}\bar{\mathbf{x}}_{N+1} = \bar{\alpha}_{N+1}. \tag{3.L+3}$$

The above system is a system of $k_{N-1} + k_N + k_{N+1}$ equations in $k_{N-1} + k_N + k_{N+1}$ unknowns which may solved to give $\mathbf{x}_{N-1}$, $\mathbf{x}_N$, and $\mathbf{x}_{N+1}$. The remaining unknown vectors may be obtained by back substitution.

3.2. Dense Solvers

We first examined the performance of two dense solvers, namely, the NAG routine F04ATF [4] and the LAPACK pair _GETRF-_GETRS [10]. The performance of these two solvers is examined in detail in [11]. The results we will present in this study are the ones obtained with the LAPACK pair as it is much more time efficient.

3.3. Banded Solvers

We also experimented with the banded solver pair F07BDF-F07BEF from the NAG Library. As can be seen from Figure 1, the parts of the global matrices which are enclosed in the banded system have a considerable degree of sparsity which cannot be exploited by the banded solvers. This degree of sparsity increases as the degree of the approximating polynomial increases, but remains independent of the number of elements in the decomposition (see also [11]).

3.4. Sparse Solvers

We examined the performance of some state-of-the-art general sparse solvers, namely, the UMFPACK Versions 1.1 [5,7] and 2.0 [6,7]. In cases when these led to a performance gain, we investigated the performance of a combination of the capacitance technique with the UMFPACK solvers. In order to exploit the sparsity of the capacitance matrix in the system $(3.L+1)$–$(3.L+3)$, we solved it using the UMFPACK solvers instead of the general dense solver.

4. EXPERIMENTAL RESULTS

4.1. Numerical Example

The performance of the various techniques described in Section 2 was tested on the following test problem:

$$\nabla^2\phi(x,y) = (y^2-1)e^x + (x^2-1)e^y + 2e^x + 2e^y, \qquad \text{on } (-1,1)^2,$$

subject to Dirichlet boundary conditions which correspond to the exact solution of this problem $\phi(x,y) = (y^2-1)e^x + (x^2-1)e^y$. We used the decomposition (in the notation of Section 2) $\alpha_i = \alpha_{i-1} + (1/2)(\alpha_N - \alpha_{i-1})$, $i = 1,2,\ldots,N-1$, $\alpha_0 = \alpha = -1$, $\alpha_N = \beta = 1$, and $a_i = a_{i-1} + (1/2)(a_N - a_{i-1})$, $i = 1,2,\ldots,N-1$, $a_0 = a = -1$, $a_N = b = 1$. We also took (in equation (2.3)) $M_s = N_s = n$, $s = 1,2,\ldots,L$. The total number of unknowns is therefore $L \times (n+1)^2$.

4.2. Implementation

The experiments were performed on a Cray J-916 vector computer, an SGI Power Challenge 8000, and an IBM RS6000-550. Timings on the Cray were obtained using the *perftrace* utility. Timings on the SGI were obtained using the *dtime* function, while on the RS6000 the timings were collected on an empty machine using the *time* function. Results are presented in seconds and each result is an average of multiple runs.

4.3. Performance on the Cray

Tables 1–3 summarize the performance of the solvers described above on the Cray J-916 for the five, seven, and nine element decompositions, respectively. For the five and seven element decompositions, only the results for Chebyshev polynomials of odd highest degrees are reported. Results for the nine domain decomposition are presented for polynomials of highest degrees 4–16.

As expected, the banded solver becomes more efficient when the number of elements in the decomposition increases (from 1.6 times faster for the largest five and seven element decompositions to about 2 times faster for the largest nine element decomposition). This relatively small improvement can be explained by the large bandwidth of the banded linear system and by the high degree of optimization of the dense solver (see also [12]). The latter is confirmed by the Mflop rates achieved by both solvers. The dense solver reaches 186 Mflops (approximately 96% of the practical peak performance [13]), while the banded solver reaches only 39 Mflops.

Table 1. Experimental results for the five subdomain decomposition.

n	Matrix Size	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2
5	180	0.17	0.15	0.11	0.21	0.20
7	320	0.38	0.32	0.21	0.57	0.59
9	500	0.84	0.65	0.40	1.31	1.32
11	720	1.92	1.40	0.79	2.58	3.30
13	980	4.12	2.83	1.49	4.65	4.49
15	1280	9.66	6.07	3.43	8.46	9.28

Table 2. Experimental results for the seven subdomain decomposition.

n	Matrix Size	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2
5	250	0.26	0.23	0.12	0.36	0.31
7	448	0.71	0.56	0.26	0.95	1.11
9	700	1.78	1.30	0.52	2.25	2.41
11	1008	4.46	3.08	1.12	4.24	5.23
13	1372	10.20	6.78	2.24	8.01	11.65
15	1792	24.00	15.00	5.18	14.96	17.72

Table 3. Experimental results for the nine subdomain decomposition.

n	Matrix Size	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2
4	225	0.23	0.21	0.11	0.31	0.26
5	324	0.39	0.33	0.18	0.51	0.43
6	441	0.66	0.53	0.31	0.96	0.81
7	576	1.18	0.86	0.55	1.40	1.34
8	729	1.96	1.38	0.88	2.35	2.15
9	900	3.30	2.27	1.33	3.15	3.48
10	1089	5.43	3.61	1.93	5.01	4.79
11	1269	8.73	5.70	2.82	6.28	7.08
12	1521	13.63	8.57	3.63	10.61	8.82
13	1764	20.62	12.81	5.91	11.96	13.37
14	2025	30.51	18.92	7.23	18.12	17.96
15	2304	48.47	29.01	26.00	23.31	24.53
16	2601	63.20	38.92	13.15	31.95	29.51

The gain from using the capacitance technique also increases as the number of elements in the decomposition increases. For the five element decomposition, the capacitance technique is about 1.74 times faster, while for the largest nine element decomposition, it is about 3 times more efficient than the banded solver. This efficiency is reached by fully utilizing the information about the structure of the linear system as the capacitance technique reaches only 112 Mflops (about 57% of the peak performance). The time increase for $n = 15$ can be related to memory bank conflicts as this degree of the polynomial generates blocks of sizes which are multiples of 16.

The performance of the general multifrontal solvers is comparable to the banded solver. This is a manifestation of a mismatch between the matrix reordering strategy of the UMFPACK and the vector processing architecture of the Cray (see also [3]). Only rarely do the UMFPACK codes reach more than 15 Mflops. It should also be pointed out that there is almost no difference between the two versions of UMFPACK. The UMFPACK documentation states that for some matrix structures Version 2.0 has about the same performance as Version 1.1 [7].

4.4 Performance on the SGI Power Challenge

Tables 4–6 summarize the performance of the solvers on the SGI Power Challenge 8000 for the five, seven, and nine element decompositions. For the five and seven element decompositions, only the results for Chebyshev polynomials of odd highest degrees are reported. Results for the nine element domain decomposition are presented for polynomials of highest degrees 4–16.

The results are somewhat surprising. The gain from using the banded solver decreases as the number of elements in the decomposition increases. For the largest five element decomposition, the banded solver is about 1.79 times faster, while for the largest nine element decomposition, it is about 1.28 times faster than the dense solver. These results need to be also compared with

Table 4. Experimental results for the five subdomain decomposition.

n	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2	UMF 1 /Cap	UMF 2 /Cap
5	0.05	0.05	0.03	0.06	0.06	0.05	0.04
7	0.15	0.12	0.09	0.16	0.18	0.11	0.14
9	0.42	0.33	0.19	0.40	0.41	0.27	0.28
11	1.13	0.83	0.47	0.86	1.21	0.56	0.85
13	3.19	2.08	0.99	1.66	1.85	1.10	1.78
15	8.99	5.02	2.16	3.27	4.00	2.23	3.41

Table 5. Experimental results for the seven subdomain decomposition.

n	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2	UMF 1 /Cap	UMF 2 /Cap
5	0.09	0.08	0.05	0.11	0.09	0.06	0.09
7	0.33	0.25	0.13	0.28	0.34	0.17	0.38
9	1.04	0.77	0.37	0.66	0.81	0.48	0.83
11	3.71	2.16	0.98	1.55	2.05	1.21	1.99
13	9.47	5.72	2.19	3.16	5.24	2.69	5.27
15	22.41	14.96	4.73	6.79	9.49	5.53	9.31

Table 6. Experimental results for the nine subdomain decomposition.

n	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2	UMF 1 /Cap	UMF 2 /Cap
4	0.07	0.07	0.04	0.09	0.08	0.04	0.04
5	0.16	0.13	0.08	0.15	0.12	0.07	0.07
6	0.31	0.24	0.15	0.27	0.24	0.14	0.14
7	0.61	0.45	0.28	0.42	0.43	0.24	0.24
8	1.10	0.90	0.51	0.66	0.69	0.41	0.42
9	2.39	1.56	0.88	1.05	1.02	0.71	0.71
10	4.22	2.61	1.48	1.83	1.92	1.15	1.22
11	7.78	4.57	2.43	2.30	2.77	1.87	1.83
12	12.50	8.21	3.81	3.74	4.31	2.88	2.92
13	20.54	13.24	5.81	5.13	5.77	4.11	4.17
14	33.72	20.82	8.58	9.92	8.19	6.39	6.80
15	47.09	31.41	12.67	11.39	12.69	9.08	8.72
16	71.47	55.72	18.18	12.22	18.17	10.38	14.44

these obtained on the Cray. While the practical peak performance of the Cray is about 195 Mflops, the same practical peak of the Power Challenge 8000 is about 290 Mflops [14]. The Cray, having about 1.49 times lower practical peak is (for the largest matrices) about 1.12 times faster on the dense solver, 1.41 times faster on the banded solver, and about 1.38 times faster on the capacitance technique. This suggests that both the dense and the banded solvers have not been fully optimized to take advantage of the Power Challenge architecture.

For the largest nine element decompositions, the capacitance technique is about 3 times faster than the banded solver, and UMFPACK v. 1.1 becomes the fastest solution method (both versions are comparable in performance). Here the RISC architecture of the Power Challenge is very well suited for the matrix reordering based UMFPACK solvers (the SGI is up to 2.5 times faster than the Cray). This result prompted us to combine the capacitance technique with the UMFPACK codes. The final combination is about 1.2 times faster than the plain UMFPACK itself.

4.5. Performance on the RS6000

Tables 7–9 present the timings of the solvers on the RS6000 for the five, seven, and nine element decompositions, respectively. For the five and seven element decompositions, only the results for Chebyshev polynomials of odd highest degrees are reported. Results for the nine domain decomposition are presented for polynomials of highest degrees 4–13 (which is the largest size that could fit into the memory of the workstation).

Table 7. Experimental results for the five subdomain decomposition.

n	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2	UMF 1 /Cap	UMF 2 /Cap
5	0.20	0.22	0.14	0.16	0.18	0.14	0.13
7	0.82	0.94	0.46	0.71	0.74	0.47	0.54
9	3.18	2.94	1.23	1.80	1.90	1.40	1.23
11	9.44	8.10	3.93	4.87	5.39	3.60	4.18
13	23.06	19.06	9.67	9.26	10.62	7.66	8.89

Table 8. Experimental results for the seven subdomain decomposition.

n	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2	UMF 1 /Cap	UMF 2 /Cap
5	0.40	0.43	0.17	0.33	0.30	0.16	0.20
7	1.80	1.81	0.69	1.12	1.35	0.75	0.79
9	6.40	5.85	2.25	3.01	3.83	2.36	2.33
11	19.38	16.23	6.73	7.63	9.47	6.60	6.26
13	46.44	38.84	17.17	18.87	24.40	16.01	14.78

Table 9. Experimental results for the nine subdomain decomposition.

n	Dense Solver	Banded Solver	Capac. Method	UMF 1	UMF 2	UMF 1 /Cap	UMF 2 /Cap
4	0.23	0.28	0.10	0.28	0.20	0.14	0.15
5	0.61	0.70	0.23	0.51	0.42	0.27	0.27
6	1.36	1.49	0.53	1.01	0.90	0.61	0.61
7	2.88	3.09	1.19	1.85	1.90	1.27	1.21
8	5.93	5.31	2.18	2.86	3.01	2.24	2.21
9	10.83	9.79	3.95	4.88	4.79	4.16	4.03
10	19.54	16.64	6.93	8.36	9.66	7.02	7.08
11	31.76	26.45	11.54	11.13	13.12	11.51	11.14
12	51.64	42.68	18.08	20.68	23.28	17.77	18.14
13	78.73	64.09	29.83	29.02	29.01	27.46	26.82

The results obtained here are quite similar to those of the SGI Power Challenge, which can be explained by the fact that both are RISC based architectures. At the same time, the gain from using the banded solver over the dense solver is about 1.2 times faster and is independent of the number of elements in the decomposition. Similarly, in all cases, the capacitance technique is about 2.13 times faster than the banded solver. The efficiency of the general solvers is quite similar to the capacitance technique, while the combination of the capacitance technique with the general solver (v. 1.1) becomes the most efficient solution method.

5. CONCLUSIONS

We present efficient direct methods for the solution of the global systems resulting from conforming spectral approximations for certain nonconforming domain decompositions. A compar-

ison of these methods is carried out for three high performance computing environments. The results indicate that a capacitance-type technique (which exploits the block structure of the global matrix) is the most efficient solution method on a vector computer. In the case of RISC based architectures, the capacitance technique should be combined with a state-of-the-art sparse solver such as Versions 1.1 and 2.0 of UMFPACK (which exploit the sparsity of the global matrix) for maximum efficiency. Our results also confirm that the processor speed should not be used to predict computer performance, as the much slower Cray easily outperforms the SGI Power Challenge due to its superior memory bandwidth and highly optimized BLAS kernels.

REFERENCES

1. A. Karageorghis and S. Sivaloganathan, Conforming spectral approximations for non-conforming domain decompositions, *Comput. Methods Appl. Mech. Engng.* (to appear).
2. A. Karageorghis, Conforming spectral domain decomposition techniques, In *Numerical Analysis and its Applications, Proceedings of the First International Workshop, WNAA'96*, Rousse, Bulgaria, 1996, (Edited by L. Vulkov, J. Wasniewski and P. Yalamov); *Lecture Notes in Computer Science*, No. 1196, pp. 220–226, Springer-Verlag, Berlin, (1997).
3. A. Karageorghis and M. Paprzycki, An efficient direct method for fully conforming spectral collocation schemes, *Numerical Algorithms* **12**, 309–319 (1996).
4. *Numerical Algorithms Group Library Mark 15*, NAG(UK) Ltd, Wilkinson House, Jordan Hill Road, Oxford, UK, (1992).
5. T.A. Davis, User's guide for the unsymmetric-pattern multifrontal package, Technical Report TR-95-004, CISE Department, University of Florida, (1995).
6. T.A. Davis and I.S. Duff, A combined unifrontal/multifrontal method for unsymmetric sparse matrices, Technical Report TR-95-020, CISE Department, University of Florida, (1995).
7. *UMFPACK, Versions 1.1 and 2.0*, available from `www.cis.ufl.edu/~davis`.
8. C. Canuto, M. Hussaini, A. Quarteroni and T. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, (1988).
9. J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, Springer-Verlag, New York, (1989).
10. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *LAPACK User's Guide*, SIAM, Philadelphia, (1992).
11. A. Karageorghis and M. Paprzycki, High performance solution of linear systems arising from conforming spectral approximations for non-conforming domain decompositions, In *Numerical Analysis and its Applications, Proceedings of the First International Workshop, WNAA'96*, Rousse, Bulgaria, 1996, (Edited by L. Vulkov, J. Wasniewski and P. Yalamov); *Lecture Notes in Computer Science*, No. 1196, pp. 373–379, Springer-Verlag, Berlin, (1997).
12. J.J. Dongarra, I.S. Duff, D.C. Sorensen and H. Van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, (1991).
13. M. Paprzycki and C. Cyphers, Multiplying matrices on the Cray—Practical considerations, *CHPC Newsletter* **6**, 77–82 (1991).
14. I. Bar-On and M. Paprzycki, An efficient algorithm for finding eigenvalues of complex symmetric matrices, *Computer Methods in Applied Mechanics and Engineering* (to appear).