An efficient preconditioned iterative solver for solving a coupled fluid structure interaction problem

Martyn D. Hughes* and Ke Chen¶

Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 7ZL, UK.

Abstract

Modelling the interaction of an acoustic field in a fluid and an elastic structure submerged in the fluid leads to a system of complex linear equations with a complicated sparsity structure and, for higher wavenumbers and adequate modelling, the systems are very large. Direct methods are not practical. Preconditioned iterative methods, which are suitable for single operator equations, are not immediately applicable to the coupled case. This paper proposes a block diagonal preconditioner of the sparse approximate inverse (SPAI) type that can accelerate the convergence of a Krylov iterative solver for the coupled system. Moreover, the proposed preconditioner can properly and implicitly scale the coupled matrix. Some numerical results are presented to demonstrate the effectiveness of the new method.

Keywords: Coupled linear systems, ill-conditioning, iterative methods, preconditioning, sparse approximate inverses, *a priori* patterns, sparsification, diagonal blocks.

C. R. Categories: G1.3; G1.8

1 Introduction

This work is concerned with the solution of the linear systems of equations arising from modelling the interaction of an acoustic field in a fluid medium (such as the sea) with an elastic structure (such as a submarine or other vessel). Refer to [1, 17, 12].

The acoustic field is governed by the small amplitude linear wave equation which is reduced to a Helmholtz equation when considered as single frequency harmonic waves [5]. The Helmholtz equation in the exterior of an elastic structure is formulated in equivalent integral equation form over the surface of the structure to avoid the problem of the infinite fluid domain. This has the added advantage that the acoustic problem is reduced from a three-dimensional problem to a two-dimensional one while the Sommerfeld boundary condition at infinity is satisfied exactly, as is well known. The integral equation is then reduced to discrete form using the boundary element method (BEM). The displacement of an elastic structure under the influence of the applied and acoustic field is modelled by a linear elasticity equation which is in turn solved using the finite element method (FEM). These models are coupled by insisting that the fluid particle velocity normal to the surface is continuous [1].

As far as fast solution is concerned, some special cases of simpler geometries have been considered before. In the case of axisymmetric structures, further dimensional reduction can be done and leads to relatively smaller linear systems [1]. In particular, the Schur complements idea has been tried (i.e. either the structural equation or the surface Helmholtz equation is eliminated first). However, for the fully three-dimensional case, the coupled system is too large to use any direct solution methods, especially for medium and large wavenumbers. Iterative methods with standard preconditioners converge too slowly, or without suitable preconditioning not at all (see our experiments later). One way to make use of existing methods is to scale the coupled system first using diagonal matrices before considering preconditioning,

*Financial support of an EPSRC CASE award (with QinetiQ, UK) is gratefully acknowledged.

¶E-mail: k.chen@liverpool.ac.uk, URL: http://www.liv.ac.uk/~cmchenke (for correspondence).

©Int. J. Comput. Math., Vol 81 (5), pp.583-594, (2004).

achieving some success [2]. Here continuing the work of [2], we take a different approach that implements scaling implicitly.

We propose to use a block diagonal preconditioner of the SPAI type to accelerate the convergence of the Krylov iterative solution methods, mainly the generalized minimal residual method (GMRES [14, 6]) and the conjugate gradients squared method (CGS [16, 6, 15]).

It should be remarked that, although our particular coupling problem arising from structural acoustics is somewhat typical of a larger class of coupling interaction problems, there exist many other related but different models and methods in this general field; refer to [7, 9, 13, 18].

2 The model problem and solution methods

We first briefly describe the models used for modelling our fluid and structure interaction problem in three dimensions. It consists of a vibrating elastic structure and an acoustic field exterior to this structure. Refer to [1, 17].

The behaviour of the elastic structure under the influence of the applied and acoustic fields is modelled by linear elasticity. Using a finite element method, either through discretizing the structural differential equation from beam theory or by energy conservation of the structure body, a structural equation can be written. As in [1] we take the latter approach to derive a linear system, which is of the form

$$(K - \omega^2 M)\underline{q} = \underline{f}, \tag{1}$$

where K is the stiffness matrix, M the mass matrix, $\underline{q}$ the displacement, and $\underline{f} = \underline{f}^k + \underline{f}^\phi$ the total load due to the applied forces ($\underline{f}^k$) and the fluid pressure ($\underline{f}^\phi$). Here ω is the time harmonic frequency for the kinetic energy of the structure and the stiffness matrix contains a large Lame constant λ.

In our discretization scheme, we approximate the structure surface using piecewise quadratic surfaces defined over triangular elements. Further, prism-shaped finite elements are formed by projection from these surfaces to the centre of the structure and we define piecewise quadratic interpolation functions on the triangular faces and piecewise linear functions on the rectangular faces to approximate $\underline{q}$. Overall, the matrices in (1) are real, symmetric, and sparse.

For the corresponding acoustic field, the use of single frequency harmonic waves of the form $\Phi(p,t) = \phi(p)e^{-i\omega t}$, where $\Phi(p,t)$ is the excess pressure at the point p at time t, reduces the linear wave equation (governing Φ) to a Helmholtz equation [5, 1]

$$\nabla^2\phi(p) + k^2\phi(p) = 0, \tag{2}$$

where $k = \omega/c$ is the acoustic wave number, c being the wave speed. To ensure that all waves are outgoing at infinity, we use the Sommerfeld radiation condition ($r = |p - q|$ with $p, q \in \mathrm{R}^3$)

$$\lim_{r\to\infty} r\left\{\frac{\partial\phi(r)}{\partial r} - ik\phi(r)\right\} = 0.$$

Formulating the Helmholtz equation in equivalent boundary integral equation form over the surface of the structure gives

$$\int_S \left\{\phi(q)\frac{\partial G_k(p,q)}{\partial n_q} - G_k(p,q)\frac{\partial\phi(q)}{\partial n_q}\right\} dS_q = \frac{1}{2}\phi(p),$$

where $G_k(p,q) = \frac{e^{ik|p-q|}}{4\pi|p-q|}$ is the free-space Green's function for the Helmholtz equation. To avoid problems of non-existence and non-uniqueness of the solutions of the integral equation at the natural frequencies of the structure, we use the Burton and Miller formulation,

$$\left(-\frac{1}{2}\mathcal{I} + \mathcal{M}_k + \alpha\mathcal{N}_k\right)\phi = \left[\mathcal{L}_k + \alpha\left(\frac{1}{2}\mathcal{I} + \mathcal{M}_k^T\right)\right]\frac{\partial\phi}{\partial n}, \tag{3}$$

where $\mathcal{L}_k$ and $\mathcal{M}_k$ are the single and double layer Helmholtz potential operators, $\mathcal{M}_k^T$ and $\mathcal{N}_k$ are their normal derivatives, and α is a coupling parameter whose imaginary part must be nonzero. The above integral equation is discretized using the boundary element (BE) method. We use the collocation method,

with triangular elements for the surface of the structure and piecewise constant interpolation functions for the solution, which discretizes (3) as a linear system of the form

$$R\underline{\phi} = i\omega\rho B\underline{v} - \underline{c}, \tag{4}$$

where ρ is the fluid density, $\underline{\phi}$ is the pressure, $\underline{v}$ the velocity, and $\underline{c}$ the incident wave pressure. The matrices in this system are dense and complex (non-Hermitian).

For the coupled system, we require that the fluid particle velocity be continuous at the surface of the structure to couple the BE and FE systems. The load due to the fluid pressure, $\underline{f}^{\phi}$, can be written in terms of the pressure potential $\underline{\phi}$ as

$$\underline{f}^{\phi} = -L\underline{\phi} \qquad \text{with} \qquad \underline{f}^{\phi} = \underline{f} - \underline{f}^{k},$$

where L is a matrix derived from the basis functions used in the BE and FE analyses. Also, the velocity, $\underline{v}$, can be written in terms of the displacement, $\underline{q}$, as

$$\underline{v} = -i\omega L'\underline{q},$$

where L' is a matrix derived from the basis functions used in the FE analysis. Then the coupled problem is to solve, simultaneously, the equations

$$\begin{cases} R\underline{\phi} = i\omega\rho B\underline{v} - \underline{c}, \\ (K - \omega^2 M)\underline{q} = \underline{f}^{k} - L\underline{\phi}. \end{cases} \tag{5}$$

This gives the partitioned system for $\underline{\phi}$ and $\underline{q}$

$$\begin{bmatrix} R & -\omega^2\rho BL' \\ L & K - \omega^2 M \end{bmatrix} \begin{bmatrix} \underline{\phi} \\ \underline{q} \end{bmatrix} = \begin{bmatrix} -\underline{c} \\ \underline{f}^{k} \end{bmatrix},$$

which will be written as the generic equation

$$Ax = b. \tag{6}$$

A typical matrix of this system has the pattern of nonzeros as in Fig.1. Diagrammatically, the block

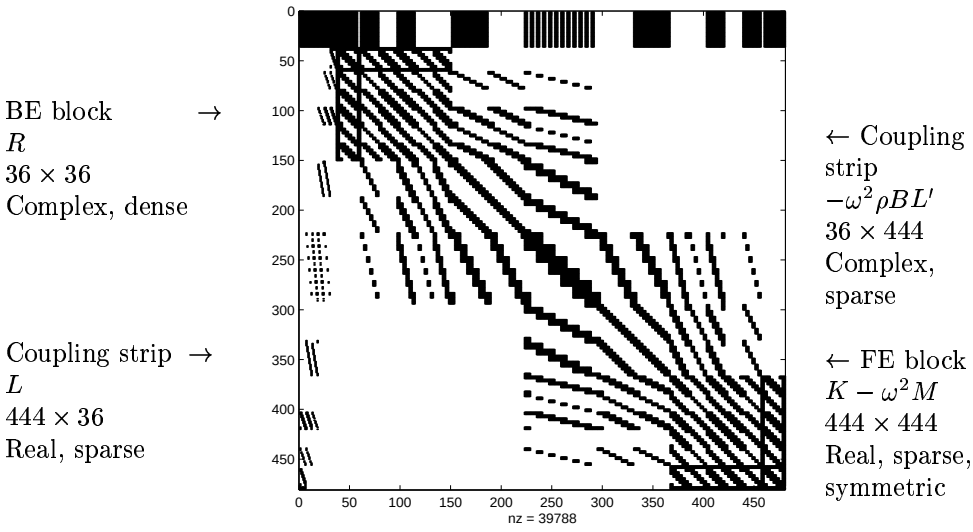

Figure 1: Pattern of the coupled matrix A

structure of the matrix is

$$\left[\begin{array}{l|l} \textbf{BE} & \textbf{BEC} \\ \text{Boundary element block} & \text{Coupling strip} \\ \text{(small, dense} & \text{(fairly full,} \\ \text{complex)} & \text{complex)} \\ \hline \textbf{FEC} & \textbf{FE} \\ \text{Coupling strip} & \text{Finite element block} \\ \text{(sparse, real)} & \text{(sparse, real,} \\ & \text{symmetric)} \end{array}\right]. \tag{7}$$

In this study, we use two iterative methods — GMRES and CGS. For convenience, we describe the methods in the following algorithms (with GMRES(m) denoting the restarted GMRES, see [14]) for solving (6) via $APy = b,\ x = Py$ with matrix P to be specified later.

Algorithm 1 (CGS iterative solver with right preconditioning)
1. Set the initial guess, y_0, and the maximum number of iterations, MAXIT
2. Compute the initial residual $r_0 = b - APy_0$ and set $p_0 = u_0 = r_0$
3. for $j = 0, 1, \ldots$ MAXIT unless $\|r_j\|/\|r_0\| \leq TOL$
4. $p_j = Pp_j, \quad \alpha_j = (r_j, r_0)/(Ap_j, r_0), \quad q_j = u_j - \alpha_j Ap_j \quad and \quad s_j = u_j + q_j$
5. $y_{j+1} = y_j + \alpha s_j, \quad s_j = Ps_j, \quad r_{j+1} = r_j - \alpha_j As_j$
6. $\beta_j = (r_{j+1}, r_0)/(r_j, r_0), \quad u_{j+1} = r_{j+1} + \beta_j q_j, \quad p_{j+1} = u_{j+1} + \beta_j(q_j + \beta_j p_j)$
7. end
8. Accept the solution $x = Py$.

Algorithm 2 (GMRES iterative solver with right preconditioning)
1. Set the initial guess y_0, the restart value m, and the maximum number of iterations MAXIT
2. Compute $r_0 = b - APy_0, \beta = ||r_0||_2$, and $v_1 = r_0/\beta$
3. for $k = 1, \ldots,$ MAXIT unless exit from the loops (global)
4. for $j = 1, \ldots, m$ (outer loop)
5. Compute $v_j = Pv_j$ and $w = Av_j$
6. for $i = 1, \ldots, j$ (inner loop)
7. $h_{i,j} = (w, v_i) = w^H v_i$
8. $w = w - h_{i,j} v_i$
9. end i (inner loop)
10. Compute $h_{j+1,j} = ||w||_2$, $v_{j+1} = w/h_{j+1,j}$ and define $V_m = [v_1, \ldots, v_m]$, $H_m = \{h_{i,j}\}_{1 \leq j \leq m}^{1 \leq i \leq m+1}$
11. end j (outer loop)
12. Solve $y_m = argmin_y ||\beta e_1 - H_m y||_2$. Compute $y_m = y_0 + V_m y_m$, $x = Py_m$ and $r = b - Ax$
13. Accept the solution x and exit if $\beta/\|r_0\|_2 \leq TOL$ with $\beta = \|r\|_2$
14. Set $v_1 = r/\beta$ and $y_0 = y_m$.
15. end k (global).

There is no convergence of either method for our coupled system without any preconditioning (i.e. setting $P = I$). This may be because the eigenvalues of matrix A are too spread out. Ideally, the eigenvalues should be clustered, say at 1, and there should be few near zero. Preconditioning can achieve this.

We first consider the simplest preconditioners. These are the inverses of sparse approximations to A, such as the matrix with just the diagonal entries, or block diagonal entries, of A, or incomplete factorizations of A. It turns out that these do improve the convergence of iterative solvers but they are not robust. For example, a simple diagonal preconditioner produces slow convergence for GMRES (instead of no convergence) but not CGS. Scaling of the matrix on the right by a diagonal matrix

$$\left[\begin{array}{cc} I & 0 \\ 0 & D_2 \end{array}\right],$$

where I is the identity matrix of the size of the BE part R and D_2 is a diagonal matrix of the size of the FE part, with diagonal entries 10^{-11} to equalize the maximum size of the entries in both parts, also produces slow convergence of GMRES (as observed in [2]) but not CGS. The table below shows the number of iterations required for GMRES(∞) to converge to a tolerance of 10^{-5} using the diagonal inverse matrix alone (diag), and the scaling matrix alone (scale). Clearly both ideas are useful but we shall explore further improvements.

Size	*Method*	*GMRES*	*Size*	*Method*	*GMRES*	*Size*	*Method*	*GMRES*
480	scale	51	702	scale	87	2754	scale	170
	diag	42		diag	79		diag	94

3 Sparse approximate inverses and *a priori* patterns

The performance of iterative solvers can be greatly improved using more sophisticated preconditioners, including ones involving approximations to the inverse of A itself. We are using such preconditioners, generated by an algorithm called SPAI (SParse Approximate Inverse). A major advantage of this method, particularly for this problem, is that it is easily parallelizable.

The method can be used to generate a left or a right preconditioner; consider the case of a right preconditioner. The method determines how good an approximation P is to A^{-1} by minimizing the Frobenius norm

$$||AP - I||_F. \tag{8}$$

From the definition of the Frobenius norm, this minimization problem separates into n independent least squares problems

$$\min_{m_k} ||Am_k - e_k||_2, \qquad k = 1, 2, \ldots, n,$$

where m_k and e_k are the kth columns of P and I respectively. For sparse A (as in this case) and P these problems are small and quick to solve.

Hence the method is implemented column-by-column (or row-by-row for a left preconditioner) and so is easy to parallelize. Run sequentially, the method is not competitive with other methods for speed but it can be when run in parallel. Separation of the columns/rows also means that effort can be concentrated on the difficult ones, which may be a particular advantage for this problem. We can adopt the well-known implementation of an adaptive approach as in [10]. In this method we start with a simple pattern for P (usually diagonal) and add a few entries at a time (usually 5), chosen by a minimization procedure for the error involved in adding the entries to the column, until an error tolerance or a maximum number of nonzeros in the column has been reached. The choice of entry positions can be restricted, for example to those of A. It is important that the total number of nonzeros per column is kept fairly small; just one full column defeats the object since more work is involved in this case than in a direct method of solution.

Alternatively, if we can find a good sparsity structure for the approximate inverse beforehand, this can be used as an *a priori* pattern, with the method being used to solve the least squares problem for all the corresponding entries in a column at once. This avoids the adaptive procedure and so can be much faster, but it is important that the *a priori* pattern is sufficiently sparse. This is an inherent problem since the true inverse is almost certain to be full. See Refs. [11, 3, 4]. To choose a good *a priori* pattern, we need to know which of the entries in the true inverse of A are the most important. We can use the theory of the relationship between the pattern of A, denoted by $S(A)$, and the pattern of its inverse, denoted by $S(A^{-1})$. From the characteristic polynomial of A, we have

$$\alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \alpha_3 A^3 + \cdots + \alpha_n A^n = 0,$$

where the α's are the coefficients in the polynomial. Multiplying by A^{-1} and rearranging, we have

$$A^{-1} = -(\alpha_1 I + \alpha_2 A + \alpha_3 A^2 + \alpha_4 A^3 + \cdots + \alpha_n A^{n-1})/\alpha_0.$$

Hence the pattern of A^{-1} is contained in the pattern $\bigcup_{j=0}^{n-1} S(A^j)$, or

$$S(A^{-1}) \subseteq S((I + A)^{n-1})$$

so we can use $S((I + A)^m)$ for small m as an approximate pattern for A^{-1}, and hence as an *a priori* pattern. Provided that A has a full diagonal, we can thus use A, A^2, A^3 etc as *a priori* patterns.

Unless the matrix is very sparse, it is clearly necessary to remove some of the entries from the matrix before using it or its powers as *a priori* patterns, in order to avoid the pattern being too dense. It is obviously essential for full matrices. This can be done using various methods, including a simple dropping strategy based on a threshold using the mean of the entries, either globally or per column. This process is called sparsification. In our case, this will be done for both the BE (complex) and FE (real) parts. The outline of the *a priori* SPAI method for a generic $n \times n$ matrix A is as follows.

Algorithm 3 (The *a priori* SPAI algorithm)
1. for $k = 1, \ldots, n$ (loop through the cols of A)
2. Determine the indices of nonzero entries in the k^{th} column of the pattern matrix to form the vector $\mathcal{J}$
3. Determine the unique indices of the nonzero entries of the $\mathcal{J}$ cols of A to form the vector $\mathcal{I}$
4. Form the reduced matrix $\hat{A} = A(\mathcal{I}, \mathcal{J})$ and the reduced identity vector $\hat{e} = e_k(\mathcal{I})$
5. Solve for the k^{th} column, $\hat{m}_k$, of the reduced approximate inverse matrix, M, from

$$\min_{\hat{m}_k} ||\hat{A}\hat{m}_k - \hat{e}_k||_2$$

6. Set the appropriate entries of the k^{th} column of M to $\hat{m}_k$
7. end

4 A NEW BLOCK DIAGONAL PRECONDITIONER

Rather than using SPAI on the whole matrix, we propose to take advantage of the block structure of the matrix and apply the *a priori* SPAI method for each of the two diagonal blocks. Thus we obtain an approximate block diagonal inverse preconditioner. The overall block diagonal preconditioner takes the following form

$$P = \begin{bmatrix} P_1 & \\ & P_2 \end{bmatrix} = \left[\begin{array}{c|c} \begin{array}{l}\textbf{BEINV}\\ \text{SPAI approximate}\\ \text{inverse for BE block}\end{array} & \textbf{0} \\ \hline \textbf{0} & \begin{array}{l}\textbf{FEINV}\\ \text{SPAI approximate}\\ \text{inverse for FE block}\end{array} \end{array}\right] \tag{9}$$

from solving (see (7))

$$\min_{P_1} \|\mathbf{BE}P_1 - I\|_F \quad \text{and} \quad \min_{P_2} \|\mathbf{FE}P_2 - I\|_F.$$

Here the *a priori* patterns for P_1, P_2 are based on the patterns of the original blocks, with sparsification using a dropping strategy based on the global mean of the absolute values of the entries.

For cases where the (full) BE part is small compared to the (sparse) FE part, we can afford to fill this completely to produce a very accurate inverse, while still keeping the overall number of nonzeros in P small. However, it is more efficient with SPAI to use sparsification to reduce the number of nonzeros, when both blocks of A are relatively large.

Now we remark on the implicit scaling done by our preconditioner P. Clearly from (7) and above, **FE** $P_2 = (K - \omega^2 M)P_2 \approx I$. Suppose ℓ is an ideal scaling constant and $(\bar{K} - \omega^2\bar{M}) = (K - \omega^2 M)/\ell$ is a better scaled matrix in relation to **BE** in (7). If we try to construct an SPAI block preconditioner for the scaled matrix, we will get

$$(\bar{K} - \omega^2\bar{M})\bar{P}_2 = (\bar{K} - \omega^2\bar{M})\ell\frac{1}{\ell}\bar{P}_2 = (K - \omega^2 M)\frac{1}{\ell}\bar{P}_2 \approx I.$$

Therefore, we see that $P_2 = \bar{P}_2/\ell$ assuming that we use the same *a priori* pattern and hence constants in **FE** will not affect the quality of SPAI. The main point here is that ℓ needs not to be known at all. That is, preconditioner P implements an implicit scaling for A.

The resulting preconditioner is as effective as those using the *a priori* method for the whole matrix, but the new method is considerably faster and there are less nonzeros in the preconditioner. There is obvious scope for parallelization of the SPAI for the BE and FE parts for high resolution simulations.

5 NUMERICAL RESULTS

We have carried out extensive tests for the block diagonal SPAI method with sparse vector storage throughout [8]. The method has also enabled us to test larger systems than was previously possible, by virtue of the reduced memory requirements of the new method and better efficiency.

The numerical experiments are conducted for a steel concentric sphere, with inner radius 0.75 m and outer radius 1 m, in water. Only the outer spherical surface vibrates and interacts with the water

medium. The physical parameters for all the cases were as follows:

Young's modulus = 209×10^9 Nm^{-2},
Poisson's ratio = 0.3,
Density of steel = 7800 Kgm^{-3},
Density of water = 1000 Kgm^{-3},
Speed of sound in water = 1500 ms^{-1}.
Frequency tested $\omega = 1500$ rad/sec (wave number $k = 1$).

The tolerance for all the iterative methods was 10^{-5} for all cases and GMRES was restarted as shown. The matrices were not scaled.

The headings for the columns in the tables have the following meanings:

- CASE

  The size of the matrix, n, together with the sizes of the BE, n_{BE}, and FE, n_{FE}, blocks

- FR BE and FR FE

  The numbers by which the means of the absolute values of the nonzero entries of the BE and FE blocks respectively were multiplied to give the threshold value above which positions were retained for the *a priori* pattern.

- NNZBE/COL and NNZFE/COL

  The average number of nonzeros per column in the *a priori* patterns, and hence in the BE and FE approximate inverses, for the BE and FE blocks, respectively.

- BE TIME and FE TIME

  The total times taken for the generation of the approximate inverses for the BE and FE blocks respectively. In some cases, times for the previous method involving assembly of the coupled matrix before SPAI are shown in brackets for comparison.

- CGS ITS and CGS TIME

  The number of iterations for CGS to converge to the required tolerance, and the total time taken.

- GM(m) ITS and GM(m) TIME

  Total number of inner iterations required for GMRES(m), where m is the restart value, to converge to the required tolerance, and the total time taken. The time taken includes the time for the total number of inner iterations within which convergence occurred so that, for example, if GMRES(50) converged in a total of $55 = 1 \times 50 + 5$ inner iterations, the iterations stop after the fifth inner iteration in the second outer iteration.

Table I: Test cases and results from the new block SPAI preconditioner

CASE n (n_{BE}, n_{FE})	FR BE	FR FE	NNZ BE/ COL	NNZ FE/ COL	BE TIME	FE TIME	CGS ITS	CGS TIME	GM(50) ITS	GM(50) TIME
480	0	0	36	68	0.075 (.06)	8.39 (8.07)	9	0.31	11	0.4
(36, 444)	0.1	0.1	34	56	0.07 (.05)	5.91 (5.69)	12	0.39	16	0.54
702	0	0	36	78	0.074	26.52	14	0.92	17	1.18
(36, 666)	0.1	0.1	34	64	0.067	18.16	23	1.42	26	1.69
1884	0	0	144	69	11.05 (9.82)	46.56 (52)	16	2.48	18	2.94
	2	0.1	9	54	0.18	29.66	25	3.43	30	4.54
2754	0	0	144	80	10.91	143.58	23	6.76	25	7.92
(144, 2610)	0.1	0.1	82	67	4.53	101.9	30	8.17	35	10.13
	2	0.1	9	67	0.18	101.12	33	8.89	36	10.83
7500	0	0	576	69	2591 (2318)	271 (442)	32	32.45	35	37.87
(576, 6924)	0.1	0.1	179	48	340 (315)	146 (271)	43	37	54	51.04
	2	0.1	17	48	7.36	145	50	40.51	55	49.84

Tables I and II clearly demonstrate that the new preconditioner is quite effective for the CGS and GMRES(m) method. There is a fine balance between thresholds and convergence. Without further fine

Table II: Larger test cases and results from the new block SPAI preconditioner

CASE n (n_{BE}, n_{FE})	FR BE	FR FE	NNZ BE/ COL	NNZ FE/ COL	BE TIME	FE TIME	CGS ITS	CGS TIME	GM (100) ITS	GM (100) TIME
10962	0	0	576	80	2400	781	32	50.85	38	64.64
(576, 10386)	0.1	0.1	179	63	341	524	57	80	59	91.96
	..	0.5	..	35	..	193	65	83	67	97.4
	2	0.1	17	63	7.55	524	>5000		60	90.89
29964	0	0	2304	69	1.5e6	2376	66	1039	71	1059
(2304, 27660)	0.1	0.1	393	45	45234	1393	>1000		135	1761
	..	1	..	20	44262	550	>1000		210	2657
	..	2	..	9	44377	275	>1000		333	4202

tuning, we would suggest the value of 0.1 for both "FR BE" and "FR FE". Not surprisingly, CGS is less robust than GMRES. As mentioned, these experiments were conducted in Fortran on a Sun Blade 100 sparc-workstation. For Table II, direct solvers cannot be set up and attempted because of memory limitation.

Finally, we make a comparison with other related methods. Table III shows some results for simple diagonal preconditioners and ILU (incomplete LU decomposition) preconditioners using the pattern of the original matrix with comparable number of nonzeros. In Table III, these results were obtained using Matlab 5 using the full pattern of the coupled matrix (i.e. "FR BE", "FR FE" = 0 in Table I) as the *a priori* pattern are shown for comparison. Here GMRES(∞) with no restart is used.

Table III: Comparison results with other methods.

CASE	METHOD	GMRES ITS	GMRES FLOPS	CGS ITS	CGS FLOPS	METHOD FLOPS
480	SPAI	11	1.44e7	8	1.34e7	1.1e9
	ILU	27	3.22e7	16	2.66e7	8.1e6
	DIAG	42	4.0e7	DNC		
702	SPAI	17	3.6e7	13	3.38e7	3.48e9
	ILU	45	8.9e7	36	9.33e7	1.55e7
	DIAG	79	1.38e8	DNC		
1884	SPAI	18	1.3e8	15	1.4e8	6.72e9
	ILU	47	3.67e8	33	3.63e8	1.08e8
	DIAG	73	4.5e8	DNC		
2754	SPAI	25	2.67e8	22	2.87e8	1.87e10
	ILU	55	5.95e8	28	4.1e8	1.4e8
	DIAG	94	8.6e8	DNC		
7500	SPAI	35	2.2e9	31	2.6e9	3.23e11
	ILU	108	8.49e9	63	7.16e9	5.21e9
	DIAG	154	9.2e9	>2000		

Note: No convergence (denoted by DNC) is also observed for cases without any preconditioning.

The comparison results show that the use of simple diagonal and ILU preconditioners can enable GMRES to converge and CGS to converge sometimes. However, the performance is generally not as good as our suggested SPAI and the ILU method does not have the advantage of being easily parallelized. Thus, although ILU appears to be doing well in preconditioner set up from the last column of Table III, use of parallel methods will reduce that for SPAI dramatically.

Appendix — An all-real block diagonal SPAI algorithm

In our BE-FE formulation (7), there exist a mixture of real (FE) and complex (BE) matrices. However, if so desired, we can write an algorithm by splitting the complex parts of the system into their separate real and imaginary parts and equating the real and imaginary parts on both sides to give a new and all-real system, as follows.

Denote the original system as

$$\begin{bmatrix} R & L \\ C & F \end{bmatrix} \begin{bmatrix} \phi \\ q \end{bmatrix} = \begin{bmatrix} c \\ f \end{bmatrix}. \tag{10}$$

Splitting the complex blocks into their real and imaginary parts, the system can be written as

$$\begin{bmatrix} R_1 + iR_2 & L_1 + iL_2 \\ C & F \end{bmatrix} \begin{bmatrix} \phi_1 + i\phi_2 \\ q_1 + iq_2 \end{bmatrix} = \begin{bmatrix} c_1 + ic_2 \\ f \end{bmatrix}. \tag{11}$$

Expanding this system, we have two equations

$$(R_1 + iR_2)(\phi_1 + i\phi_2) + (L_1 + iL_2)(q_1 + iq_2) = c_1 + ic_2 \tag{12}$$
$$C(\phi_1 + i\phi_2) + F(q_1 + iq_2) = f \tag{13}$$

and, multiplying out,

$$R_1\phi_1 + iR_2\phi_1 + iR_1\phi_2 - R_2\phi_2 + L_1q_1 + iL_2q_1 + iL_1q_2 - L_2q_2 = c_1 + ic_2$$
$$C\phi_1 + iC\phi_2 + Fq_1 + iFq_2 = f.$$

Equating the real and imaginary parts on both sides, we get four equations

$$R_1\phi_1 - R_2\phi_2 + L_1q_1 - L_2q_2 = c_1 \tag{14}$$
$$R_2\phi_1 + R_1\phi_2 + L_2q_1 + L_1q_2 = c_2 \tag{15}$$
$$C\phi_1 + Fq_1 = f \tag{16}$$
$$C\phi_2 + Fq_2 = 0, \tag{17}$$

and these can be written as the new system

$$\left[\begin{array}{cc|cc} R_1 & -R_2 & L_1 & -L_2 \\ R_2 & R_1 & L_2 & L_1 \\ \hline C & 0 & F & 0 \\ 0 & C & 0 & F \end{array}\right] \left[\begin{array}{c} \phi_1 \\ \phi_2 \\ \hline q_1 \\ q_2 \end{array}\right] = \left[\begin{array}{c} c_1 \\ c_2 \\ \hline f \\ 0 \end{array}\right] \tag{18}$$

If the original BE and FE blocks are of size $n_{BE} \times n_{BE}$ and $n_{FE} \times n_{FE}$, respectively, the sizes of the blocks in this new system are

$$\left[\begin{array}{c|c} 2n_{BE} \times 2n_{BE} & 2n_{BE} \times 2n_{FE} \\ \hline 2n_{FE} \times 2n_{BE} & 2n_{FE} \times 2n_{FE} \end{array}\right]. \tag{19}$$

The structure of the matrix of this system gives new possibilities for the block diagonal SPAI method and for sparsification, in all-real arithmetic.

6 Conclusions

The coupled system from using boundary and finite element methods to model the dynamic fluid-structure interaction problem cannot be solved by iterative methods without preconditioning and suitable scaling. In this paper we have proposed a block diagonal preconditioner of the approximate inverse type and demonstrated that the method enables CGS and GMRES to converge quickly. The method is also compared favourably with the ILU preconditioner. Further work involving high order boundary elements and parallel computing is under way.

References

[1] S. Amini, P. J. Harris and D. T. Wilton, *Coupled Boundary and Finite Element Methods for the Solution of the Dynamic Fluid-Structure Interaction Problem*, Springer-Verlag, 1992.

[2] K. Chen and P. J. Harris, *On preconditioned iterative solution of the coupled block linear systems arising from the three-dimensional fluid-structure interaction problem*, Proc. 3rd UK BIE conf., Brighton University, UK (2001), pp 139-148.

[3] E. Chow, *A Priori Sparsity Patterns for Parallel Sparse Approximate Inverse Preconditioners*, SIAM J. Sci. Comput., Vol. 21, No. 5 (2000), pp. 1804-1822.

[4] E. Chow and Y. Saad, *Approximate Inverse Preconditioners via Sparse-Sparse Iterations*, SIAM J. Sci. Comput., Vol. 19 (1998), pp. 995-1023.

[5] D. Colton D. and R. Kress, *Integral equation methods in scattering theory*, John Wiley and Sons, New York (1983).

[6] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, USA, 1997.

[7] Q. Du, M. D. Gunzburger, L. S. Hou and J. Lee, *Analysis of a linear fluid-structure interaction problem*, Discrete Cont. Dyna. Sys., Vol. 9, No. 3 (2003), pp 633-650.

[8] I. S. Duff, A. M. Erisman and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

[9] C. Farhat, M. Lesoinne and P. LeTallec, *Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces*, Comput. Methods Appl. Mech. Engng., Vol. 157 (1998), pp 95-114.

[10] M. J. Grote and T. Huckle, *Parallel Preconditioning with Sparse Approximate Inverses*, SIAM J. Sci. Comput., Vol. 18, No. 3 (1997), pp. 838-853.

[11] T. Huckle, *Approximate Sparsity Patterns for the Inverse of a Matrix and Preconditioning*, Appl. Numer. Math., Vol. 30 (1999), pp 291-303.

[12] M. Malhotra, R. W. Freund and P. M. Pinsky, *Iterative solution of multiple radiation and scattering problems in structural acoustics using a block QMR algorithm*, Comput. Methods Appl. Mech. Engng., Vol. 146 (1997), pp 173-196.

[13] C. Rajakumar and A. Ali, *Boundary element-finite element coupled eigenanalysis of fluid-structure systems*, Int. J. Numer. Methods Engng., Vol.39, No. 10 (1996), pp 1625-1634.

[14] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, Mass, 1996.

[15] J R. Shewchuck, *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*, Carnegie Mellon University, Pittsburgh, 1994.

[16] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM. J. Sci. Statist. Comput., Vol. 10 (1989), pp 36-52.

[17] D. T. Wilton, *A numerical approach to some problems in acoustic radiation: Part C — the coupled structure fluid interaction problem*, MoD Research Report 3, Department of Mathematics, University of Dundee, UK (1974).

[18] G. Y. Yu, *Symmetric collocation BEM/FEM coupling procedure for 2D dynamic structural acoustic interaction problems*, Comput. Mech., Vol. 29 (2002), pp 191-198.