 ICCES, vol.2, no.2, pp.41-46, 2007

On the efficiency of generic BE substructuring algorithms based on Krylov solvers

F.C. Araújo[1] and L.J. Gray[2]

Summary

This paper is concerned with the solution of boundary-element models based on substructuring. Structured matrix-vector products and the matrix-copy option are proposed to increase the efficiency of algorithms based on Krylov solvers. The former technique was designed to avoid the excessive number of conditional tests during solver iterations, and the latter one, to avoid the repeated calculation of coefficient matrices for identical subregions. Potential applications of the algorithm to composite materials, and to develop parallel codes, are noted.

Introduction

Mainly in the 90s, iterative Krylov solvers began to be widely considered in the development of computer codes to solve engineering problems, including those based on Boundary Element Methods. The major advantage of this kind of solver is their efficiency for solving large-order systems and their suitableness for developing parallel codes. In Computational-Fluid-Dynamics (CFD) simulations, for instance, they have been commonly applied to develop parallel finite-element codes [1-3].

For Boundary Element Methods, Krylov solvers have also been successfully applied, in particular for substructuring algorithms [4-5]. In this connection, optimized data structures for a generic number of coupled subdomains, with complete exclusion of zero blocks, have been proposed [6-9]. This paper considers two further improvements to these algorithms: the structuring of matrix-vector products (SMVP) involved in the iterative solvers and the implementation of the matrix-copy option (MCO). The first technique was designed to exclude the many conditional tests, necessary when the matrix-vector products are left unstructured (UNSMVP). The second technique is designed to avoid calculating and assembling, repeatedly, the coefficient matrices for identical subregions.

Generic Coupling of BE Models

As an iterative solver does not transform the coefficient matrix, formats can be devised that reduce memory requirements and solution CPU time. Here, a subregion-by-subregion data structure, described below, is used.

Generically (for n_s subregions), the corresponding subsystems of boundary el-

[1]Department of Civil Engineering, UFOP, 35400-000 Ouro Preto-MG, Brazil; at ORNL under ORISE/HERE Faculty Program.

[2]Computer Science and Mathematics Division, ORNL, Oak Ridge, P.O. Box 2008, USA.

ement equations can be written as

$$\sum_{m=1}^{i-1}(\mathbf{H}_{im}\mathbf{u}_{mi}-\mathbf{G}_{im}\mathbf{p}_{im})+\mathbf{A}_{ii}\mathbf{x}_i+\sum_{m=i+1}^{n}(\mathbf{H}_{im}\mathbf{u}_{im}+\mathbf{G}_{im}\mathbf{p}_{mi})=\mathbf{B}_{ii}\mathbf{y}_i,\quad i=1,n_s, \quad (1)$$

where, $\mathbf{H}_{ij}$ and $\mathbf{G}_{ij}$ denote the usual BE matrices obtained for source points pertaining to subregion Ω_i and associated respectively with the boundary vectors $\mathbf{u}_{ij}$ and $\mathbf{p}_{ij}$ at Γ_{ij}. Here Γ_{ij} with $i \neq j$ is the interface between Ω_i and Ω_j, and Γ_{ii} is the outer boundary of Ω_i. In the subregion-by-subregion data structure considered herein, these subsystems are not explicitly assembled into a global system, but separately stored. Thus, the many zero blocks, unavoidably present at these systems, are completely excluded [9]. Moreover, it is assumed that discontinuous boundary elements are employed, so that compatibility and equilibrium conditions at the interfaces,

$$\begin{cases} \mathbf{u}_{ij}=\mathbf{u}_{ji} \\ \mathbf{p}_{ij}=-\mathbf{p}_{ji} \end{cases} \quad \text{at} \quad \Gamma_{ij} \qquad (2)$$

only have to be imposed pairwise. This enormously simplify the treatment of inner edges and corners [8].

The first topic addressed in this work is the structuring of the matrix-vector products employed by the iterative solver. Herein, the J-BiCG solver [4, 6] is adopted, and the matrix-vector products are of the form $(\mathbf{D}^{-1}\mathbf{A})\mathbf{p}_j$ and $(\mathbf{D}^{-1}\mathbf{A})^T\mathbf{p}_j^*$, where $\mathbf{A}$ and $\mathbf{D}$ are the global matrix of the coupled system and its corresponding diagonal matrix (Jacobi preconditioning), respectively. In previous versions of the algorithm [9], unstructured matrix-vector products (UNSMVP) were employed. This means that no column re-ordering of the subregion matrices were carried out, and thus, conditional tests to identify the type of boundary condition present at each degree of freedom (interface or prescribed boundary value) had to be performed at every iteration of the solver.

By structured matrix-vector products (SMVP) it is meant that the columns of the matrix of a given subregion, say Ω_i, are re-ordered so as to group its coefficients into separate blocks. Three blocks are adopted: one associated with interfaces Γ_{ij} for which $i > j$, a second associated with the outer boundary Γ_{ii} (at which boundary values are prescribed), and a last one associated with interfaces Γ_{ij} for which $i < j$ (see equation 2). This data structure is precisely that indicated in equation (1). Thus, matrices $\mathbf{A}$ and $\mathbf{B}$ for the i-*th* subregion, after interchanging columns for introducing the boundary conditions at Γ_{ii}, have the following generic aspects:

$$\begin{array}{c|c|c|c|} & \overleftarrow{\qquad\text{block1}\qquad} & \overleftarrow{\text{block2}} & \overleftarrow{\qquad\text{block3}\qquad} \\ H_i=[& H_{i,1}\cdots H_{i,i-1} & A_{ii} & H_{i,i+1}\cdots H_{i,n} \;] \\ G_i=[& G_{i,1}\cdots G_{i,i-1} & B_{ii} & H_{i,i+1}\cdots H_{i,n} \;] \end{array} \qquad (3)$$

In the code, each block is identified by an initial and a final column, calculated for each subregion matrix.

The second modification is the implementation of the matrix-copy option (MCO). This will prove useful for problems involving identical subregions. Two examples are identical inclusions or layers in composite materials (Figure 1). By employing the matrix-copy option, a matrix for a subdomain only needs to be assembled once and read from memory when needed. In this way, CPU time and memory may be saved. Note that for solid mechanics problems, rotation transformation of the copied matrices may be also necessary.

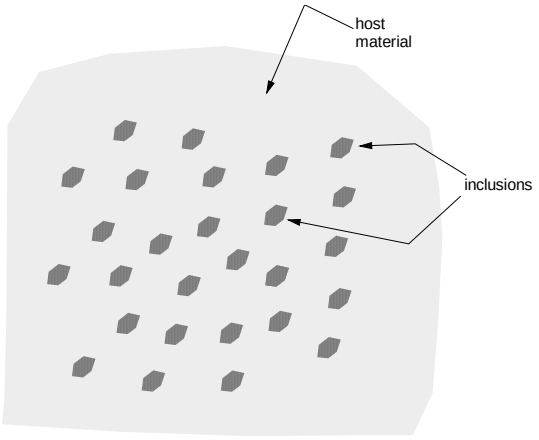

Figure 1: Composite material

Results and Discussion

To demonstrate the effectiveness of the techniques, the holed rod shown in Figure 2 is analyzed. Its length is 1500 *mm*, and its cross section has width and height 140 mm and 225 mm respectively. The rod-wall thickness is 50 *mm*. The elasticity modulus and Poisson's ratio are $E = 205 \times 10^3 \quad MPa$ and $\nu = 0.30$ respectively. The rod is subjected to a normal pressure of 1.0 *MPa*. In Figure 2(a) and 2(b), the general aspect of the entire mesh and the coupled subdomains may be seen. In Figure 2(c), details of each subregion are shown. In sum, the BE model is composed of six identical subregions, each one with 144 boundary elements and 432 nodes, corresponding to a total of 1296 degrees of freedom per subregion and of 7776 for the global system.

For the rod dimensions chosen, the response is well described by one-dimensional bar theory. The results agree very well with the analytical bar solution, with the error less than 0.4%, and are not presented here. The number of iterations and CPU times required for the solution are given in Table 1, where n_{it} denotes the number of iterations and n the system order. In Table 2, the matrix-assembly CPU times

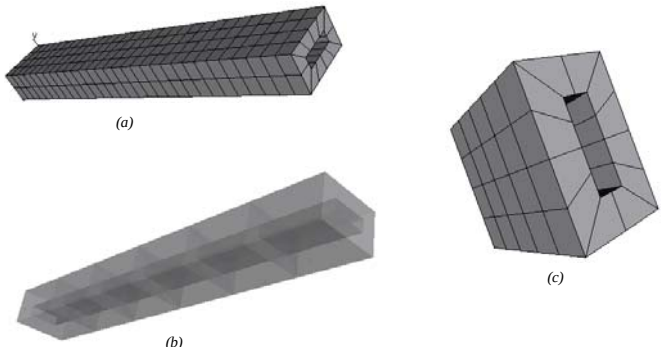

Figure 2: BE mesh adopted

using and not using matrix-copy option (MCO) are shown. The solver has been stopped when $\|\delta\|_2 / \left\|\mathbf{D}^{-1}\mathbf{b}\right\|_2 < 10^{-5}$.

For this single test, a CPU time reduction per iteration of about 15% is obtained using the SMVP option (see Table 1). The small variation between the number of iterations using SMVP and UNSMVP can be expected, as the order of

Table 1: Efficiency parameters

	n_{it}	*solver CPU time (s)*	*CPU time*$/n_{it}$	n_{it}/n
SMVP	2141	242	0.11	0.275
UNSMVP	2165	290	0.13	0.278

Table 2: Matrix-assembly CPU time (s)

	with no MCO	with MCO
SMPV	87	18

operations are different in these strategies, and arithmetic properties like the associativity of addition are not valid. Therefore, to measure the CPU-time performance difference, the relation *CPU time*$/n_{it}$ was adopted. The relations n_{it}/n measured hints the good performance of the J-BiCG solver. As seen, the CPU time for assembling the global coupled matrix not using MCO (matrix-copy option) is about five times that using MCO.

Conclusions

For the numerical test, the efficiency of the coupling algorithm improved by adopting structured matrix-vector products (about 15%). Of course, for larger problems, this increase in efficiency will be more significant, as more conditional tests would have to be performed. Concerning the matrix-copy option (MCO), it considerably reduced, as expected, the assembly time of the global system, and will be likely very useful for modeling composite materials. Furthermore, if the matrix-copy option is allied with unstructured matrix-vector products, memory space may be substantially reduced. As a natural consequence of the domain-decomposition

strategy considered, the algorithm is highly suitable for developing parallel BE codes.

Acknowledgement

This research was supported in part by the Applied Mathematical Sciences Research Program of the Office of Mathematical, Information, and Computational Sciences, U.S. Department of Energy under contract DE-AC05-00OR22725 with UT-Batelle, LLC, in part, by the Brazilian Research Council, CNPq, and in part, by the U.S. Department of Energy Higher Education Research Experiences (HERE) Program, administered by the Oak Ridge Institute for Science and Education, ORISE.

References

1. Baum, J. D., Luo, H. and Löhner, R. (1994): "Edge-based finite element scheme for Euler equations", *AIAA J.*, Vol. 32, pp. 1183-1190.
2. Dutto, L. C., Lepage, C. Y. and Habashi, W.G. (2000): "Effect of the storage format of sparse linear systems on parallel CFD computations", *Comput. Meth. Appl. Mech. Engrg.,* Vol 188, pp. 441-453.
3. Catabriga, L., Coutinho, A. L. G. A. (2002): "Implicit SUPG solution of Euler equations using edge-based data structures", *Comput. Meth. Appl. Mech. Engrg.*, Vol. 191, pp. 3477-3490.
4. Mansur, W. J., Araújo, F. C. and Malaghini, J. E. B. (1992): "Solution of BEM systems of equations via iterative techniques", *Int. J. Numer. Methods Engrg.*, Vol. 33, pp. 1823-1841.
5. Prasad, K. G., Kane, J. H., Keyes, D. E. and Balakrishna, C. (1994): "Preconditioned Krylov solvers for BEA", *Int. J. Numer. Methods Engrg.*, Vol. 37, pp. 1651-1672.
6. Araújo, F. C., Martins, C. J. and Mansur, W. J. (2001): "An efficient BE iterative-solver-based substructuring algorithm for 3D time-harmonic problems in elastodynamics", *Engineering Analysis with Boundary Elements*, Vol. 25, pp. 795-803.
7. Araújo, F. C., Dors, C. and Mansur, W. J. (2003): "Dynamic cross-interaction between footings - 3D frequency-domain parallelized analysis via the BEM", In: *Proc. International Conference on Boundary Element Techniques IV*, Granada,. Queen Mary, University of London: London, Vol. I, pp. 253-258.
8. Araújo, F. C., Dors, C., Martins, C. J. and Mansur, W.J. (2004): "New developments on BE/BE multi-zone algorithms based on Krylov solvers – Applications to 3D frequency–dependent problems", *J. Braz. Soc. Mech. Sc. & Engrg*., Vol. 26, pp. 231-248.

9. Araújo, F. C., Silva, K. I. and Telles, J. C. F. (in press): "Generic domain decomposition and iterative solvers for 3D BEM problems", *Int. J. Numer. Methods Engrg.*