

An efficient BE iterative-solver-based substructuring algorithm for 3D time-harmonic problems in elastodynamics

F.C. Araújo^{a,*}, C.J. Martins^a, W.J. Mansur^b

^aDepartment of Civil Engineering, Escola de Minas, UFOP, CEP 35400-000, Ouro Preto-MG, Brazil

^bPEC/COPPE/UFRRJ, CXP 68506, CEP 21945-970, Rio de Janeiro, Brazil

Received 18 May 2000; accepted 13 December 2000

Abstract

This work is concerned with the development of an efficient and general algorithm to solve frequency-domain problems modelled by the boundary element method based on a sub-region technique. A specific feature of the algorithm discussed here is that the global sparse matrix of the coupled system is implicitly considered, i.e. problem quantities are not condensed into interface variables. The proposed algorithm requires that only the block matrices with non-zero complex-valued coefficients be stored and manipulated during the analysis process. In addition, the efficiency of the technique presented is improved by using iterative solvers. The good performance of pre-conditioned iterative solvers for systems of equations having real-valued coefficients, well demonstrated in the literature, is confirmed for the present case where the system matrix coefficients are complex. The efficiency of the algorithm described here is verified by analysing a soil–machine foundation interaction problem. CPU time and accuracy are the parameters used for estimating the computational efficiency. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Boundary element method; 3D Elastodynamics; Frequency-dependent problems; Iterative solvers; Complex non-Hermitian matrices; Substructuring technique

1. Introduction

The pioneer work by Cruse [1] was the first to establish the mathematical basis for numerical modelling of elastic wave propagation problems by the boundary element method. Cruse [1] used integral equations in the Laplace transformed domain and a numerical algorithm of inversion due to Papoulis [2] to obtain time-domain responses. It is important to mention that presently Durbin's [3] inversion algorithm is employed rather than Papoulis' [2].

Cruse's [1] work inspired later on the development of BEM models based on the Fourier transform. The derivation of a BEM approach in Fourier domain was straightforward (see Ref. [4]) as such a formulation could be easily derived from Cruse's [1] work by replacing the complex Laplace parameter s by $i\omega$. FFT algorithms of inversion were used in this case to find time-domain solutions.

BEM procedures based either on Laplace or Fourier transforms (see Ref. [5]) have remained as important alternatives to time-domain BEM formulations (see Refs. [6,7,5]). They are useful alternatives to standard time-

domain BEM approaches and essential either when time-domain fundamental solutions have not been obtained but the corresponding frequency (or Laplace) domain ones are known, or when physical properties are frequency-dependent. A complete review of the main contributions up to 1996 to the topic boundary element methods in dynamic analysis can be found in Beskos [8,9].

The present paper is concerned with the development of a general and efficient BEM sub-region approach to analyse soil–machine foundation interaction. Only harmonic loads are considered here, however, transient excitations can also be dealt with as a trivial extension of the harmonic case [1,4]. Frequency-domain engineering analyses of infinite domain problems can be conveniently carried out by the boundary element method or by BEM/FEM coupling algorithms. Frequency-domain approaches deal with complex-valued variables that are commonly used to describe amplitude and phase angle of both loading and response. Real engineering analyses may require modelling non-homogeneous infinite media, e.g. soil–structure interaction problems. When the soil is composed of horizontal layers, simple procedures can be employed leading to very economic and accurate numerical models, e.g. models based on superposition of planes waves [10]. However,

* Corresponding author. Tel.: +55-31-559-1547; fax: +55-31-551-1988.
E-mail address: feelio@em.ufop.br (F.C. Araújo).

when layers are not horizontal, more general procedures must be used [11], BEM being one of the most suitable. In this case, the BE region must be discretized using the sub-region technique.

As it is well known, substructuring strategies extend considerably the range of applications of boundary-integral-based methods. General aspects of sub-regions techniques can be found in known textbooks on boundary element methods [12,13]. More specific aspects concerning substructuring techniques, which are mainly related to the efficient solution of the blocked, sparse, and unsymmetrically coupled final systems, are addressed to in a series of papers published in the last 20 years [14–19]. These works adopt either non-condensed or condensed strategies and are based on the use of direct solvers. In this paper, a non-condensed iterative-solver-based strategy is presented.

Efficient procedures must be used in order to analyse complex problems in small computers. The algorithm must use the minimum storage area and yet be fast. These two topics are addressed to in this paper. Storage area is reduced to a minimum by not storing null coefficients and iterative solvers are used to reduce CPU time. The good performance of iterative solvers for real-valued coefficient systems of Eqs. (20)–(23) is also confirmed for the present case, where the system matrix coefficients are complex.

2. Frequency-dependent boundary integral equations

When the analysis is frequency-dependent the originally time-dependent problem can be converted to an only-boundary value problem governed by

$$\begin{aligned}
 c_{ik}(\xi)U_i(\xi, \omega) + \int_{\Gamma} P_{ik}^*(\mathbf{x}; \xi, \omega)U_i(\mathbf{x}, \omega) d\Gamma(\mathbf{x}) \\
 = \int_{\Gamma} U_{ik}^*(x; \xi, \omega)P_i(x, \omega) d\Gamma(x) \\
 + \int_{\Omega} U_{ik}^*(x; \xi, \omega)B_i(x, \omega) d\Omega(x), \tag{1}
 \end{aligned}$$

where c_{ik} is the integral-equation jump term, equal to jump terms for elastostatics, and $U_{ik}^*(\mathbf{x}; \xi, \omega)$ and $P_{ik}^*(\mathbf{x}; \xi, \omega)$ are the frequency-dependent fundamental solutions given by

$$\begin{aligned}
 U_{ik}^*(\mathbf{x}; \xi, \omega) = \frac{1}{4\pi\rho r} \left\{ (3r_{,i}r_{,k} - \delta_{ik}) \right. \\
 \times \left[\frac{1}{r^2\omega^2} \left(e^{i\omega(r/c_2)} - e^{i\omega(r/c_1)} \right) \right. \\
 \left. - \frac{i}{r\omega} \left(\frac{1}{c_2} e^{i\omega(r/c_2)} - \frac{1}{c_1} e^{i\omega(r/c_1)} \right) \right] \\
 + r_{,i}r_{,k} \left(\frac{1}{c_1^2} e^{i\omega(r/c_1)} - \frac{1}{c_2^2} e^{i\omega(r/c_2)} \right) \\
 \left. + \delta_{ik} \left(\frac{1}{c_2^2} e^{i\omega(r/c_2)} \right) \right\} \tag{2}
 \end{aligned}$$

and

$$\begin{aligned}
 P_{ik}^*(\mathbf{x}; \xi, \omega) = \frac{n_m}{4\pi r^2} \\
 \times \left\{ -6c_2^2(5r_{,i}r_{,k}r_{,m} - \delta_{ik}r_{,m} - \delta_{km}r_{,j} - \delta_{mi}r_{,k}) \right. \\
 \times \left[\frac{1}{r^2\omega^2} \left(e^{i\omega(r/c_2)} - e^{i\omega(r/c_1)} \right) \right. \\
 \left. - \frac{i}{r\omega} \left(\frac{1}{c_2} e^{i\omega(r/c_2)} - \frac{1}{c_1} e^{i\omega(r/c_1)} \right) \right] + 2 \\
 \times (6r_{,i}r_{,k}r_{,m} - \delta_{ik}r_{,m} - \delta_{km}r_{,i} - \delta_{mi}r_{,k}) \\
 \times \left(e^{i\omega(r/c_2)} - \frac{c_2^2}{c_1^2} e^{i\omega(r/c_1)} \right) \\
 - 2\frac{i\omega}{c_2}r_{,i}r_{,k}r_{,m} \times \left(e^{i\omega(r/c_2)} - \frac{c_2^3}{c_1^3} e^{i\omega(r/c_1)} \right) \\
 - r_{,k}\delta_{im} \left(1 - 2\frac{c_2^2}{c_1^2} \right) \left(1 - i\omega\frac{r}{c_1} \right) e^{i\omega(r/c_1)} \\
 \left. - (r_{,m}\delta_{ik} + r_{,i}\delta_{km}) \left(1 - i\omega\frac{r}{c_2} \right) e^{i\omega(r/c_2)} \right\}. \tag{3}
 \end{aligned}$$

When known complex boundary conditions

$$\mathbf{U}(\mathbf{x}, \omega) = \bar{\mathbf{U}}(\mathbf{x}, \omega) \quad \text{if } x \in \Gamma_1, \tag{4}$$

and

$$\mathbf{P}(\mathbf{x}, \omega) = \bar{\mathbf{P}}(\mathbf{x}, \omega) \quad \text{if } x \in \Gamma_2, \tag{5}$$

are considered, the boundary integral Eq. (1) gives the complete boundary solution of the frequency-dependent problem (in terms of its complex amplitudes).

Adopting usual discretization procedures, the boundary integral Eq. (1) is converted to the following frequency-dependent system of algebraic equations:

$$\mathbf{H}(\omega)\mathbf{U}(\omega) = \mathbf{G}(\omega)\mathbf{P}(\omega), \tag{6}$$

after introducing the boundary conditions shown in Eqs. (4) and (5), Eq. (6) can be written as

$$\mathbf{A}(\omega)\mathbf{x}(\omega) = \mathbf{b}(\omega), \tag{7}$$

whose complex solution vector $\mathbf{x}(\omega)$ contains the unknown boundary values.

It should be noted that a special integration process based on a triangular polar co-ordinate transformation, optionally with use of integration subelements [7,24], was employed for obtaining the complex coefficients of the system of Eq. (6) associated with weakly singular integrals. Coefficients computed from Cauchy principal values — the diagonal block matrices coefficients (DBMC) — were evaluated

indirectly by means of the rigid body displacement criterion. In order to apply this criterion to frequency-dependent problems, the following relationship must be considered [25]:

$$\begin{aligned}
 & c_{ik}(\xi) + {}^{\text{dyn}}h_{ik}(\xi, \omega) \\
 &= \left[c_{ik}(\xi) + {}^{\text{st}}h_{ik}(\xi, \omega) \right] \\
 &+ \sum_{e=1}^{\text{nse}(\xi)} \int_{\Gamma_e} \left[P_{ik}^*(\mathbf{x}; \xi, \omega) - {}^{\text{st}}P_{ik}^*(\mathbf{x}; \xi) \right] \phi_{\xi}(r, s) d\Gamma(\mathbf{x}),
 \end{aligned} \tag{8}$$

where $\text{nse}(\xi)$ is the number of singular elements around ξ and

$$\phi_{\xi}(r, s)$$

is the shape function related to the singular point. Thus the diagonal block matrices coefficients for frequency-dependent problems are then determined from the sum of the DBMC for a similar elastostatic analysis — obtained by means of the rigid body displacement criterion — plus contributions of non-singular integrals as indicated by RHS of Eq. (8). Superscripts *st* and *dyn* on expression (8) stand for statics and dynamics, respectively.

Another important point to be observed is that in the case of semi-infinite domain problems a special mesh of enclosing elements must be considered in order to apply the rigid body displacement criterion [11]. Discrete Fourier transform (DFT) and inverse discrete Fourier transform (IDFT) algorithms can be employed to solve time-dependent problems in infinite or semi-infinite domains, however in this paper, only steady-state problems were considered.

3. The BE/BE coupling algorithm

In order to illustrate how the algorithm described in this

section works, the system matrix corresponding to a domain divided into the three sub-domains shown in Fig. 1, is depicted in Fig. 2. In fact the procedure followed in the present paper is quite general, i.e. one can have as many substructures as one wishes. Points 1 and 2 below, and the comments presented subsequently describe the most important features concerning the assemblage of the final system matrix $\mathbf{A}(\omega)$ shown in Eq. (7).

1. Compatibility and equilibrium conditions, which for instance at the interface between subregions *i* and *j* are given by

$$\mathbf{u}^i(\mathbf{x}) = \mathbf{u}^j(\mathbf{x}), \quad \text{if } \mathbf{x} \in \Gamma_{ij} \tag{9}$$

and

$$\mathbf{p}^i(\mathbf{x}) = -\mathbf{p}^j(\mathbf{x}), \quad \text{if } \mathbf{x} \in \Gamma_{ij} \tag{10}$$

must be introduced.

2. A search in order to identify the coupled nodes must be carried out. It should be noticed that in the proposed algorithm, a node cannot be coupled with more than one other. As described in Fig. 1, node *i* is coupled with node *j*, node *k* with node *l*, and node *m* with node *n*. By introducing, however, existing traction continuity conditions at nodes *i*, *j*, *k*, *l*, *m*, and *n* the corresponding columns in $\mathbf{G}(\omega)$ matrix can be superimposed, such that e.g. for the coupled domain shown in Fig. 1 (with three subdomains), at common interface nodes, indeed just two traction vectors (\mathbf{p}^i and \mathbf{p}^k) must be calculated. One has therefore, at the common node three equations, from which the unknown vectors

$$\mathbf{u}^i = \mathbf{u}^j = \mathbf{u}^k = \mathbf{u}^l = \mathbf{u}^m = \mathbf{u}^n, \tag{11a}$$

$$\mathbf{p}^i = -\mathbf{p}^j = -\mathbf{p}^m = \mathbf{p}^n, \tag{11b}$$

and

$$\mathbf{p}^k = -\mathbf{p}^l, \tag{11c}$$

can be calculated. In general it is valid: at a node common to *n* sub-regions, there are *n* equations for determining one

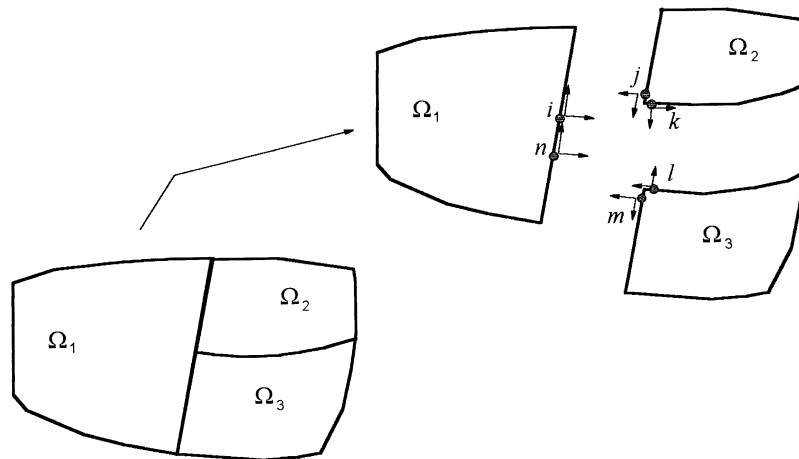


Fig. 1. Domain with three coupled subdomains.

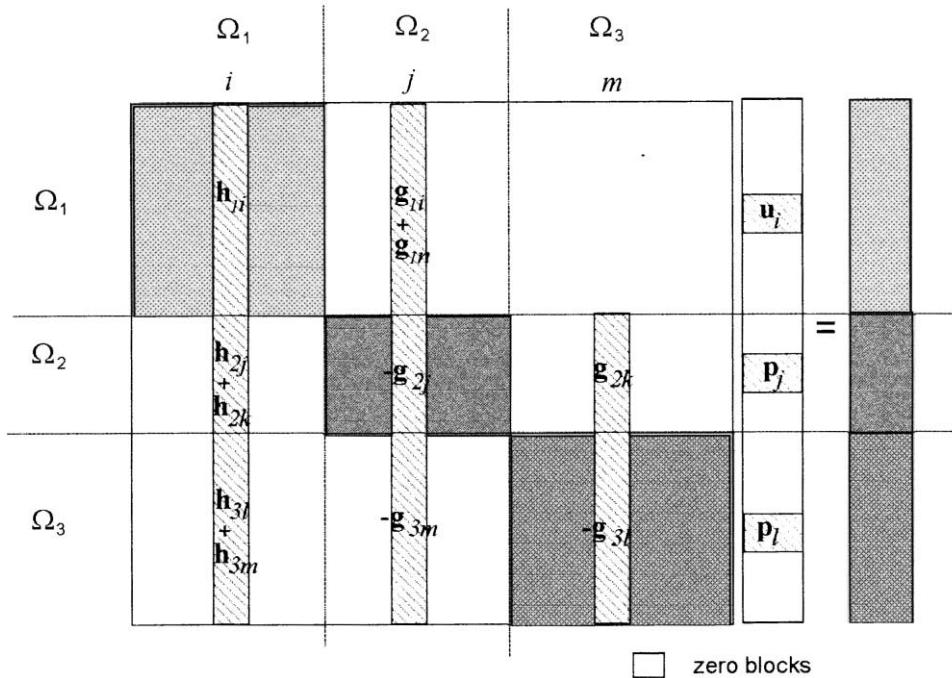


Fig. 2. The coupled system matrix.

displacement vector and $(n - 1)$ traction vectors. Thus, it is necessary that at least one sub-region be smooth at the common node, what represents no restriction to the coupling algorithm proposed here, as it is always possible to create an extra subregion with a smooth boundary, whenever necessary.

A simple and important detail of the present coupling formulation, mainly in what concerns the performance of the iterative schemes, is that, in order to obtain better convergence rates, one must use a scaling factor. The one used here is defined by

$$\bar{f} = \frac{1}{ns} \sum_{i=1}^{ns} f_i, \quad f_i = \frac{2G_i}{(1 - 2\nu_i)}, \quad (12)$$

where ns is the number of sub-regions and G_i and ν_i are, respectively, the Young's modulus and Poisson's ratio of the i th sub-region. \bar{f} , given by Eq. (12), is used to scale the coefficients of the \mathbf{G} -matrix, such that all coefficients of the resulting coupled system become of same order of magnitude. It should be also observed that as a matter of fact, the coupled domain depicted in Fig. 1 is not restricted to 2D problems, as it might seem to be; it refers to 3D problems as well, which in fact constitute exactly the class of problems being treated in this paper.

4. Iterative solvers for complex-valued coupled systems

Iterative procedures for obtaining the solution of systems of algebraic equations have played a very important role in the analysis of engineering problems. They can be of funda-

mental importance to minimise the high analyses costs normally involved in solving algebraic systems, which are sometimes sparse and have a number of equations varying from a few hundreds to a few millions. Important characteristics of iterative procedures that are closely related to their effectiveness are that, besides preserving matrix sparsity, they reduce substantially the CPU time in case of large systems of equations. The gain introduced by iterative solvers is more substantial for algorithms which employ parallel processing strategies. In this case, direct solvers are really inefficient. A comprehensive study on general aspects of iterative procedures, containing the main developments on this subject in the last decades (until the beginning of the 1990s) is presented by Hageman and Young [26] and Hackbusch [27].

Among the iterative schemes, those based on conjugate gradient acceleration procedures [26] have specially attracted the attention of the numerical analysis community. Conjugate gradient methods have a number of favourable properties, three of them being worthwhile mentioning here: (1) though an iterative scheme, convergence is achieved in a finite number of iterations if no round-off errors are present, (2) they converge at least as fast as the corresponding Chebyshev procedure, (3) no parameter estimate is necessary for obtaining optimal convergence rates.

Concerning BE analyses, nonsymmetric systems of algebraic equations are produced and therefore the development of iterative solution strategies naturally becomes a much more difficult task than in symmetric cases, as reported in works published during the 1980s [28–31]. In the late 1980s, and at the beginning of the 1990s, successful

applications of iterative solvers in BE analyses were reported in works by Araújo and Mansur [20–22,32], in which many different iterative techniques were considered. These works showed that the iterative strategies based on the biconjugate gradient (BiCG) and Lanczos procedures [20,22], with preconditioning, performed especially well in all analyses carried out. Another important point concerning these algorithms is that only information related to two previous iterations at most (Lanczos algorithm) is necessary. Other methods which require the entire history of iterations or, at least, a great deal of informations concerning previous iterations, have also been used successfully, e.g. idealised generalised conjugate gradient acceleration procedures and their corresponding truncated versions [32–35].

In the present paper, the biconjugate gradient algorithm, derived previously for real nonsymmetric and nonsingular matrices [20,22,32], is extended for complex-valued, non-Hermitian systems of algebraic equations.

4.1. Lanczos tridiagonalization algorithm

The starting point for deriving the biconjugated gradient algorithm employed here is the Lanczos tridiagonalization algorithm [32,36]. Starting from two known vectors \mathbf{c}^1 and ${}^*\mathbf{c}^1$, both in the N -dimensional complex space C^N , it is possible by means of the Lanczos tridiagonalization algorithm (in the same fashion as for real matrices [32,36]) to derive from \mathbf{A} and \mathbf{A}^T in $C^{N,N}$ two vector sequences $\{\mathbf{c}^{k+1}\}$ and $\{{}^*\mathbf{c}^{k+1}\}$, respectively given by

$$\delta_{k+1}\mathbf{c}^{k+1} = \mathbf{A}\mathbf{c}^k - \alpha_k\mathbf{c}^k - \beta_k\mathbf{c}^{k-1} \tag{13}$$

and

$${}^*\delta_{k+1}{}^*\mathbf{c}^{k+1} = \mathbf{A}^T{}^*\mathbf{c}^k - \alpha_k{}^*\mathbf{c}^k - {}^*\beta_k{}^*\mathbf{c}^{k-1}. \tag{14}$$

These are mutually orthogonal to each other, i.e. $\mathbf{c}^{k+1} \perp {}^*\mathbf{c}^1, {}^*\mathbf{c}^2, \dots, {}^*\mathbf{c}^k$ and ${}^*\mathbf{c}^{k+1} \perp \mathbf{c}^1, \mathbf{c}^2, \dots, \mathbf{c}^k$. It should be observed that these orthogonality conditions, can be used to demonstrate that these two vector sequences are linearly independent for $k \leq N$, N being the dimension of the

complex space in question. Thus the following property must be verified:

$$\mathbf{c}^{N+1} = {}^*\mathbf{c}^{N+1} = \mathbf{0} \in C^N. \tag{15}$$

The property of Lanczos vectors expressed by Eq. (15) is very important as it is exactly that one used for establishing the finite termination of the associated iterative schemes [26,32]. Moreover, if an usual inner product, i.e. not a Hermitian inner product, between complex-valued vectors is used, expressions for determining parameters α_k, β_k and ${}^*\beta_k$ in Eqs. (13) and (14) similar to those for real matrices are obtained; these parameters are computed by [32,36]

$$\alpha_k = \frac{{}^*\mathbf{c}^{k,T}\mathbf{A}\mathbf{c}^k}{{}^*\mathbf{c}^{k,T}\mathbf{c}^k}, \tag{16}$$

$$\beta_k = \frac{{}^*\mathbf{c}^{k-1,T}\mathbf{A}\mathbf{c}^k}{{}^*\mathbf{c}^{k-1,T}\mathbf{c}^{k-1}}, \tag{17}$$

$${}^*\beta_k = \frac{\mathbf{c}^{k-1,T}\mathbf{A}^T{}^*\mathbf{c}^k}{\mathbf{c}^{k-1,T}{}^*\mathbf{c}^{k-1}}. \tag{18}$$

4.2. Bi-conjugate iterative method

Following the same ideas presented by Araújo and Mansur [20–22] and Araújo [32], it is then possible, starting from Eqs. (13), (14), (16)–(18), to derive the Lanczos acceleration procedure and the bi-conjugate gradient method, the later being given by the following recursive expressions [26,32]:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \lambda_n\mathbf{p}^n, \tag{19}$$

$$\mathbf{p}^n = \begin{cases} \mathbf{r}^0, & \text{if } n = 0 \\ \mathbf{r}^n + \alpha_n\mathbf{p}^{n-1}, & \text{if } n \geq 1 \end{cases} \tag{20}$$

$$\mathbf{r}^n = \mathbf{r}^{n-1} - \lambda_{n-1}\mathbf{A}\mathbf{p}^{n-1}, \tag{21}$$

By considering now the auxiliary formulas related to \mathbf{A}^T ,

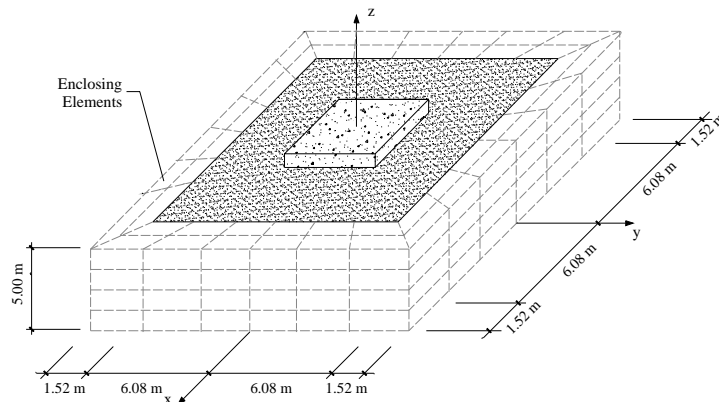


Fig. 3. Soil-foundation interaction problem.

which are,

$${}^* \mathbf{r}^n = {}^* \mathbf{r}^{n-1} - \lambda_{n-1} \mathbf{A}^T {}^* \mathbf{p}^{n-1}, \tag{22}$$

$${}^* \mathbf{p}^n = \begin{cases} {}^* \mathbf{r}^0 = \mathbf{r}^0, & \text{if } n = 0 \\ {}^* \mathbf{r}^n + \alpha_n {}^* \mathbf{p}^{n-1}, & \text{if } n \geq 1 \end{cases}, \tag{23}$$

and using the orthogonality condition of the Lanczos vectors (residue vectors) expressed by

$$\mathbf{r}^{i,T} \mathbf{r}^j = 0, \quad \text{if } i \neq j, \tag{24}$$

one can demonstrate [32] the \mathbf{A} -orthogonality condition of the search direction vectors \mathbf{p}^i ,

$${}^* \mathbf{p}^{i,T} \mathbf{A} \mathbf{p}^j = 0, \quad \text{if } i \neq j. \tag{25}$$

Moreover, with aid of these orthogonality conditions (relationships (24) and (25)), the following expressions for parameters λ_{n-1} and α_n are obtained [32]:

$$\lambda_{n-1} = \frac{{}^* \mathbf{r}^{n-1,T} \mathbf{r}^{n-1}}{{}^* \mathbf{p}^{n-1,T} \mathbf{A} \mathbf{p}^{n-1}} \tag{26}$$

$$\alpha_n = \frac{{}^* \mathbf{r}^{n,T} \mathbf{r}^n}{{}^* \mathbf{r}^{n-1,T} \mathbf{r}^{n-1}}. \tag{27}$$

Table 1
Response for $E_f = E_s$

Point	Displacement		Traction	
	$\Re(u_z)$	$\Im(u_z)$	$\Re(p_z)$	$\Im(p_z)$
A	-0.275×10^{-1}	-0.776×10^{-2}	$0.407 \times 10^{+7}$	$0.272 \times 10^{+5}$
B	-0.196×10^{-1}	-0.760×10^{-2}	$0.164 \times 10^{+8}$	$0.622 \times 10^{+6}$
C	-0.136×10^{-1}	-0.736×10^{-2}	$0.275 \times 10^{+8}$	$0.183 \times 10^{+7}$
D	-0.203×10^{-2}	-0.781×10^{-3}	–	–
E	-0.152×10^{-2}	-0.105×10^{-2}	–	–
F	-0.778×10^{-3}	-0.103×10^{-2}	–	–

Solvers based on iterative formulas (19)–(23) and (26), (27) are known as bi-conjugate gradient methods and were originally introduced by Fletcher [37].

As one can see with aid of property (24), it is naturally expected that the complex-valued residue vector \mathbf{r}^{n+1} becomes identically null for $n = N$ (c.f. Eq. (15)). This is indeed nothing else than the complete demonstration of convergence of the procedure. For ill-conditioned systems of algebraic equations, despite the finite termination property of the iterative scheme, as a consequence of truncation errors introduced during the computer data processing, it may happen that convergence is not achieved for $n \leq N$. It does happen that the larger the number of iterations required for reaching convergence, the larger the cumulative errors become, such that for slow convergence rates the procedure can definitively not converge. In order to accelerate the convergence rate of iterative schemes and to avoid cases where convergence is not achieved, some researchers have introduced preconditioning matrices in their iterative schemes [20-23,32,35,38], that are so called because they should improve the condition number of the system matrix (normally the spectral condition number is considered). Preconditioning which is quite relevant for the steepest descent procedure, in which the rate of convergence depends exclusively on the extreme eigenvalues [27,39] is also important for conjugate gradient procedures, in which, though the rate of convergence be dependent upon the complete distribution of the eigenvalues [27,39,40], it depends also upon the system matrix condition number. In the case of the bi-conjugate gradient procedure for non-Hermitian matrices, mathematical analysis concerning the rate of convergence is naturally a much more difficult issue, and is still a matter of current research. Despite this fact, following indeed the same ideas of acceleration or preconditioning considered in gradient

Table 2
CPU times and number of iterations for solving the system ($E_f = E_s$)

Solver	CPU time (s)	No. of iterations
Gauss elimination	$0.206 \times 10^{+4}$	–
Complex-J-BiCG	$0.206 \times 10^{+3}$	104
Real-J-BiCG	$0.343 \times 10^{+3}$	127

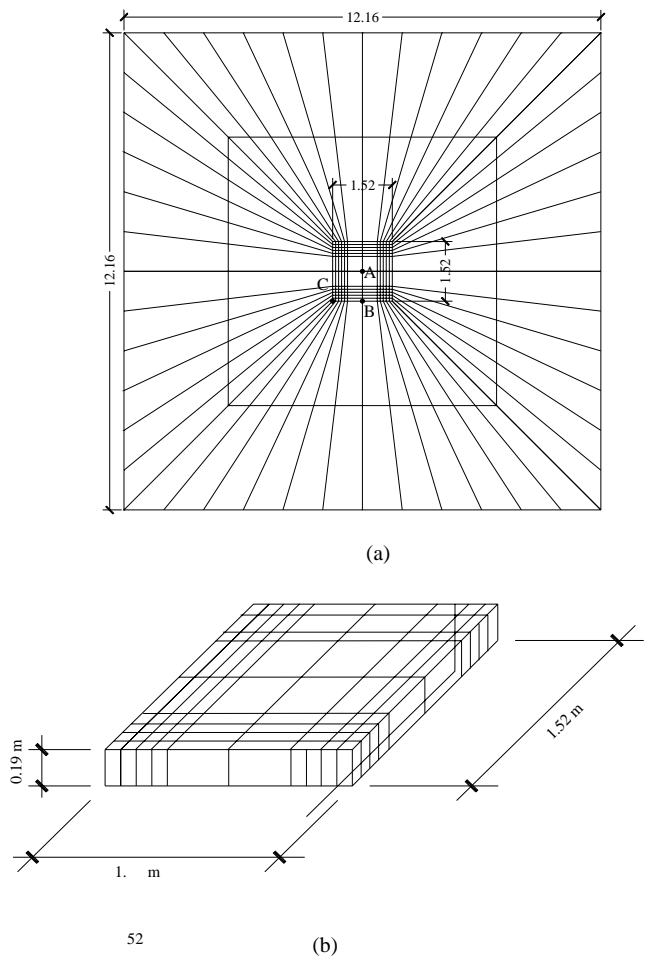


Fig. 4. BE meshes for each sub-region.

Table 3
Response for $E_f = 10E_s$

Point	Displacement		Traction	
	$\Re(u_z)$	$\Im(u_z)$	$\Re(p_z)$	$\Im(p_z)$
A	-0.255×10^{-1}	-0.744×10^{-2}	$0.368 \times 10^{+7}$	$-0.181 \times 10^{+5}$
B	-0.202×10^{-1}	-0.746×10^{-2}	$0.273 \times 10^{+8}$	$0.822 \times 10^{+6}$
C	-0.151×10^{-1}	-0.742×10^{-2}	$0.768 \times 10^{+8}$	$0.622 \times 10^{+7}$
D	-0.190×10^{-2}	-0.733×10^{-3}	–	–
E	-0.152×10^{-2}	-0.103×10^{-2}	–	–
F	-0.807×10^{-3}	-103×10^{-3}	–	–

schemes for symmetric matrices [26,27,32,39], satisfactory results in BEM analyses were obtained in the last ten years in applications of Lanczos-tridiagonalization-based iterative algorithms and generalised gradient methods by many researchers [20,22,32,35,38]. In this work, the only preconditioning matrix \mathbf{Q} used is the Jacobi splitting matrix, which is defined by the diagonal (complex-valued) of the system matrix.

4.3. The real version of the iterative scheme

An alternative procedure to iterative schemes for complex coefficient matrices is naturally their corresponding real versions. For the complex system indicated in Eq. (7), for instance, simple complex arithmetic operations can be carried out, giving as a result the following real equivalent system:

$$\begin{bmatrix} \Re(\mathbf{A}) & -\Im(\mathbf{A}) \\ \Im(\mathbf{A}) & \Re(\mathbf{A}) \end{bmatrix} \begin{Bmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{Bmatrix} = \begin{Bmatrix} \Re(\mathbf{b}) \\ \Im(\mathbf{b}) \end{Bmatrix}. \quad (28)$$

As the system in Eq. (28) is real, the usual solvers for real matrices can be applied. However, better results were obtained here by using the following diagonal preconditioning matrix \mathbf{Q} :

$$q_{ii} = \max[\Re(a_{ii}), \Im(a_{ii})] \quad (29)$$

4.4. Termination criterion

In order to terminate the iterative procedure for matrices with complex coefficients, the following criterion was adopted: the Euclidian norm of the real and imaginary parts of the residue vector at the current iteration, say $\|\Re(\mathbf{r}^n)\|$ and $\|\Im(\mathbf{r}^n)\|$, were calculated and compared with a certain tolerance ζ (a real positive number); whenever the norm of both parts were smaller than a fixed tolerance, the

Table 4
CPU times and number of iterations for solving the system ($E_f = 10E_s$)

Solver	CPU time (s)	No. of iterations
Gauss elimination	$0.206 \times 10^{+4}$	–
Complex-J-BiCG	$0.648 \times 10^{+3}$	328
Real-J-BiCG	$0.119 \times 10^{+4}$	441

Table 5
Response for ($E_f = 20E_s$)

Point	Displacement		Traction	
	$\Re(u_z)$	$\Im(u_z)$	$\Re(p_z)$	$\Im(p_z)$
A	-0.247×10^{-1}	-0.742×10^{-2}	$0.341 \times 10^{+7}$	$-0.292 \times 10^{+5}$
B	-0.204×10^{-1}	-0.745×10^{-2}	$0.302 \times 10^{+8}$	$0.685e \times 10^{+6}$
C	-0.159×10^{-1}	-0.744×10^{-2}	$0.981 \times 10^{+8}$	$0.666 \times 10^{+7}$
D	-0.191×10^{-2}	-0.728×10^{-3}	–	–
E	-0.155×10^{-2}	-0.103×10^{-2}	–	–
F	-0.832×10^{-3}	-0.104×10^{-2}	–	–

iterative procedure was terminated. One has therefore:

$$\text{if } \begin{cases} \|\Re(\mathbf{r}^n)\| < \zeta \\ \text{and} \\ \|\Im(\mathbf{r}^n)\| < \zeta \end{cases} \text{ then stop.} \quad (30)$$

For the real versions of the algorithms, see Eq. (28), a criterion equivalent to that given by expression (30) is adopted, such that the efficiency of both versions of the iterative scheme (the complex and the real) can be correctly compared one to another. It should be observed that $\|\Re(\mathbf{r}^n) + \Im(\mathbf{r}^n)\| < \zeta$ could also have been used as a termination criterion, however the criterion shown in expression (30) has been preferred in the present work.

5. Applications

The soil-foundation interaction problem depicted in Fig. 3 is analysed in order to check the performance of the coupling algorithm proposed. The problem is modelled with two sub-regions (one is the foundation and the other is the soil), such that the whole model contains 1303 nodes and 420 elements that corresponds to a complex-valued coupled system of 3909 equations (Fig. 4).

The soil parameters are $E_s = 2.0 \times 10^8 \text{ Nm}^{-2}$, $\nu_s = 0.35$ and $\rho_s = 1800 \text{ Kg m}^{-3}$. $\nu_f = 0.25$ and $\rho_f = 2500 \text{ Kg m}^{-3}$ were adopted for the foundation and four different values were adopted for its elasticity modulus: $E_f = E_s$, $E_f = 10E_s$, $E_f = 20E_s$ and $E_f = 40E_s$.

The square foundation side length and height are 1.52 and 0.19 m, respectively. The excitation loading considered was a harmonic distributed load of amplitude $4.0 \times 10^6 \text{ Nm}^{-2}$ and frequency 100 rad s^{-1} acting on the vertical direction at the top surface of the foundation. The results of the analysis in terms of displacements, tractions, CPU times

Table 6
CPU times and number of iterations for solving the system ($E_f = 20E_s$)

Solver	CPU time (s)	No. of iterations
Gauss elimination	$0.206 \times 10^{+4}$	–
Complex-J-BiCG	$0.880 \times 10^{+3}$	445
Real-J-BiCG	$0.216 \times 10^{+4}$	804

Table 7
Response for $E_f = 40E_s$

Point	Displacement		Traction	
	$\Re(u_z)$	$\Im(u_z)$	$\Re(p_z)$	$\Im(p_z)$
A	-0.239×10^{-1}	-0.743×10^{-2}	$0.308 \times 10^{+7}$	$-0.347 \times 10^{+5}$
B	-0.207×10^{-1}	-0.746×10^{-2}	$0.326 \times 10^{+8}$	$0.556 \times 10^{+6}$
C	-0.169×10^{-1}	-0.748×10^{-2}	$0.124 \times 10^{+9}$	$0.664 \times 10^{+7}$
D	-0.194×10^{-2}	-0.728×10^{-3}	–	–
E	-0.159×10^{-2}	-0.104×10^{-2}	–	–
F	-0.863×10^{-3}	-0.105×10^{-2}	–	–

Table 8
CPU times and number of iterations for solving the system ($E_f = 40E_s$)

Solver	CPU time (s)	No. of iterations
Gauss elimination	$0.206 \times 10^{+4}$	–
Complex-J-BiCG	$0.116 \times 10^{+4}$	588
Real-J-BiCG	$0.302 \times 10^{+4}$	1123

and number of iterations are shown on Tables 1–8. The points considered for showing the response are defined as follows: *A* is the point at the interface centre, *B* is the midpoint of an interface edge, *C* is the point at an interface corner, and *D*, *E*, and *F* are, respectively, soil points 1.0, 2.0 and 3.0 below *A*.

The direct solver considered is the standard Gauss elimination procedure without any pivoting strategy and that takes into account the zero blocks of the coupled system. Complex-J-BiCG and the Real-J-BiCG stand for the complex and real equivalent versions of the biconjugate gradient procedure with Jacobi preconditioner, respectively.

The termination criterion used for the iterative solver is that defined by Eq. (30), the tolerance being established by taking $\zeta = 10^{-5}$. The analyses were carried out in a personal computer with processor AMD K7 — 700 MHz and 640 Mbytes RAM, so that memory swapping (in/out-of-core transfer) was necessary.

6. Conclusions

The results presented in this paper show that the coupling algorithm developed, based on the use of iterative solvers, performs very well in what concerns response accuracy and convergence properties. No significant differences was found between results obtained with iterative solvers and those obtained by means of a coupling algorithm based on a direct solver, thus only one response concerning tractions and displacements is presented on Tables 1, 3, 5, and 7. An important fact to be observed, concerning the iterative solver analyses, is that the number of iterations required to reach convergence increased when the relationship between the elasticity modulus of the foundation and the soil increased. Based on the data shown on Tables 2 and 8 for instance, one can see that the Complex-J-BiCG iterative

solver is, for $E_f = E_s$, 10 times faster than the direct one (standard Gauss), and for $E_f = 40E_s$, only 1.8 times faster than that solver, the corresponding Real-J-BiCG solver being initially (for $E_f = E_s$) six times faster and finally (for $E_f = 40E_s$) even a little bit slower than the standard Gauss direct solver. This fact should not be considered as a drawback of the iterative solvers algorithms, rather it just hints that when $E_f = 40E_s$ a finer mesh is required to describe accurately the problem solution. This does happen because tractions amplitudes near the foundation edges increase considerably as the relationship E_f/E_s increases, such that in order to better describe the tractions over the interface a finer mesh near the edges is required. It should be also noted, that the scaling factor defined by Eq. (9), plays an important role for accelerating the convergence rate of the iterative solvers when E_f/E_s increases.

Other advantages of the iterative-solver-based coupling strategy presented here are that the zero blocks of coefficients are completely disregarded during the solution phase. It is therefore expected for coupled systems of order much higher than that one treated here, that iterative-solver-based coupling algorithms perform much better than direct-solver-based coupling strategies. Storage area required and consequently swapping time of direct solvers may restrict their use in more complex applications.

Specifically for the frequency-dependent soil-foundation interaction problem treated here, the Complex-J-BiCG solver was more efficient than the direct one for all relationships E_f/E_s considered, even though no pivoting strategy was employed in the latter solver. Concerning the iterative procedures, the complex version was more efficient than the real one; note that the number of iterations necessary to reach the convergence with the real version of the iterative scheme is greater than that of the complex one in all cases, and so it is the CPU time as well. One should observe that the preconditioning matrices in both versions are not the same (see Eq. (29)), such that the preconditioned versions of the iterative solvers are actually not identical; the preconditioning matrix given by Eq. (29) had a better performance than that Jacobi preconditioning matrix defined in the usual way for the equivalent real system of equations given in Eq. (28).

Finally, it is important to mention that the results presented in this paper hint that the complex version of iterative solvers should be used instead of the real equivalent versions, yet more numerical experiments are necessary in order to verify if this is always the case.

References

- [1] Cruse TA. The transient problem in classical Elastodynamics solved by integral equations. PhD thesis, University of Washington, 1967.
- [2] Papoulis A. A new method of inversion of Laplace transforms. Q Appl Math 1957;14(4):405–14.
- [3] Durbin F. Numerical inversion of Laplace transforms: an efficient improvement to Dubner and Abate's method. Comp J 1974;17:371–6.

- [4] Manolis GD. Dynamic response of underground structures. PhD thesis, University of Minnesota, 1980.
- [5] Manolis GD. A comparative study on three boundary element method approaches to problems in elastodynamics. *Int J Num Meth Engng* 1983;19:73–91.
- [6] Mansur WJ. A time-stepping technique to solve wave propagation problems using the boundary element method. PhD thesis, University of Southampton, 1983.
- [7] Araújo FC. Time-domain solution of three-dimensional linear problems of elastodynamics by means of a BE/FE coupling process (in German). PhD thesis, Technical University of Braunschweig, Germany, 1994.
- [8] Beskos DE. Boundary element methods in dynamic analysis. *Appl Mech Rev* 1987;40:1–23.
- [9] Beskos DE. Boundary element methods in dynamic analysis: part II (1986–1996). *Appl Mech Rev* 1997;50:149–97.
- [10] Wolf JP. Dynamic soil-structure interaction. Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [11] Araújo FC, Mansur WJ, Nishikava LK. Determination of 3D time domain responses in layered media by using a coupled BE/FE process. In: Kassab A, Brebbia CA, Chopra M, editors. *Boundary elements XX*, Orlando, Florida, USA, Southampton: Computational Mechanics Publications, 1998. p. 587–96.
- [12] Kane JH. Boundary element analysis in engineering continuum mechanics. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [13] Banerjee PK. The boundary element method in engineering. New York: McGraw-Hill, 1994.
- [14] Das PC. A disc based block elimination technique used for the solution of non-symmetrical fully populated matrix systems encountered in the boundary element method. In: *Proceedings of International Symposium on Recent Developments in Boundary Element Methods*. Southampton, UK, 1978;391–414.
- [15] Crotty JM. A block equation solver for large unsymmetric matrices arising in the boundary element method. *Int J Num Meth Engng* 1982;18:997–1017.
- [16] Kane JH, Kumar BL, Saigal S. An arbitrary condensing, non-condensing solution strategy for large scale, multi-zone boundary element analysis. *Comput Meth Appl Mech Engng* 1990;79:219–44.
- [17] Rigby RH, Allibadi MH. Out-of-core solver for large, multi-zone boundary element matrices. *Int J Num Meth Engng* 1995;38:1507–33.
- [18] Bialecki RA, Merkel M, Mews H, Kuhn G. In- and out-of-core BEM equation solver with parallel and non-linear options. *Int J Num Meth Engng* 1996;39:4215–42.
- [19] Ganguly S, Layton JB, Balakrishna C, Kane JH. A fully symmetric multi-zone Galerkin boundary element method. *Int J Num Meth Engng* 1999;44:991–1009.
- [20] Araújo FC, Mansur WJ. *Boundary elements XI*, Cambridge, USA, Southampton: Computational Mechanics Publications, 1989. p. 263–74.
- [21] Araújo FC, Mansur WJ, Malaghini JEB. Biconjugate gradient acceleration for large BEM systems of equations. *Boundary elements XII*, Sapporo, Japan, Southampton: Computational Mechanics Publications, 1990. p. 99–110.
- [22] Mansur WJ, Araújo FC, Malaghini JEB. Solution of BEM systems of equations via iterative techniques. *Int J Num Meth Engng* 1992;33:1823–41.
- [23] Barra LPS, Coutinho ALGA, Telles JCF, Mansur WJ. Multi-level hierarchical preconditioners for boundary element systems. *Engng Anal Boundary Elements* 1993;12:103–9.
- [24] Mang H, Li H, Han G. A new method for evaluating singular integrals in stress analysis of solids by the direct BEM. *Int J Num Meth Engng* 1985;21:2071–98.
- [25] Manolis GD, Beskos DE. *Boundary element methods in elastodynamics*. London: Unwin Hyman, 1988.
- [26] Hageman LA, Young DM. *Applied iterative methods*. New York: Academic Press, 1981.
- [27] Hackbusch W. *Iterative Lösung Grosser Schwachbesetzter Gleichungssysteme*, B.G., Teubner Stuttgart, 1991.
- [28] Doblare M. Three-dimensional formulation of the boundary element method with parabolic interpolation (in Spanish). PhD thesis, Polytechnical University of Madrid, Spain, 1981.
- [29] Bettess JA. Economical solution technique for boundary integral matrices. *Int J Num Meth Engng* 1985;19:1073–7.
- [30] Parreira P. Error analysis in the boundary element method in elasticity (in Portuguese). PhD Thesis, Technical University of Lisboa, Portugal, 1987.
- [31] Mullen R, Rencis JJ. Iterative methods for solving boundary element equations. *Comput Struct* 1987;25:713–23.
- [32] Araújo FC. Iterative techniques for solving linear systems of equations originated from the boundary element method (in Portuguese). MSc thesis, COPPE — Federal University of Rio de Janeiro, Brazil, 1989.
- [33] Young DM, Hayes LJ, Jea KC. Generalized conjugate gradient acceleration of iterative methods, Part I and II. Research Report CNA 162 and CNA 163, Center for Numerical Analysis, University of Texas at Austin, 1980.
- [34] Young DM, Jea KC. Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods. *Linear Algebra Appl* 1980;34:159–94.
- [35] Barra LPS, Coutinho ALGA, Mansur WJ, Telles JCF. Iterative solution of BEM equations by GMRES algorithm. *Comput Struct* 1992;44:1249–53.
- [36] Wilkinson JH. *The algebraic eigenvalue problem*. Oxford: Clarendon Press, 1965.
- [37] Fletcher R. *Conjugate gradient methods for indefinite systems*, Lecture notes in mathematics 506. Berlin: Springer, 1976.
- [38] Prasad KG, Kane JH, Keyes DE, Balakrishna C. Preconditioned Krylov solvers for BEA. *Int J Num Meth Engng* 1994;37:1651–72.
- [39] Axelsson O, Barker VA. *Finite element solution of boundary value problems — theory and computation*. New York: Academic Press, 1984.
- [40] van der Sluis A, van der Vorst HA. The rate of convergence of conjugate gradients. *Numerische Mathematik* 1986;48:543–60.