Transactions on Modelling and Simulation vol 12, © 1996 WIT Press, www.witpress.com, ISSN 1743-355X

Iterative solvers for tridimensional BEM algebraic systems of equations

F.P. Valente[a], H.L.G. Pina[b]
[a]*Department of Mathematics, ESTG,*
Instituto Politécnico da Guarda, 6300 Guarda, Portugal
[b]*Department of Mechanical Engineering,*
Instituto Superior Técnico, 1096 Lisboa Codex, Portugal

Abstract

A key issue in the Boundary Element Method is the solution of the associated system of algebraic equations. The matrices of this systems are dense and sometimes ill conditioned. For tridimensional problems, with large scale systems (several thousand of equations) direct methods like Gauss elimination become too expensive and iterative methods may be preferable.

For these problems there are already many algorithms, namely for general nonsymmetric systems. Most of them can be viewed as Lanczos or Conjugate Gradient-like solvers. Here we present some iterative techniques based on Conjugate Gradient solvers as Descent Methods (DM), Bi-Conjugate Gradients (Bi-CG), Conjugate Gradients Squared (CGS) and Bi-Conjugate Gradients Stab (Bi-CGstab) that seem to have the potential to be competitive solvers for BEM algebraic systems of equations, specialy when used with an appropriate preconditioner.

1 Introduction

The Conjugate Gradient method (CG), first described by Hestenes and Stiefel (1952), has been widely used for approximating the solution of large scale symmetric systems that arise, for example, in Computational Mechanics [2, 6, 10, 11, 18]. This method, which requires one matrix-vector product per iteration, can be viewed as a direct method that, in the absence of roundoff errors, gives the exact solution in at most n steps, n being the order of the matrix, or as an iterative solver that yields a good approximation to the exact solution in a few steps.

These iterative techniques have been widely employed for solving FEM systems of equations taking advantage of the fact that the system matrix is sparse and symmetric positive definite (SPD).

That is not the case with BEM matrices which are dense and sometimes ill conditioned. Those reasons make the CG algorithms not directly applicable and the solution of BEM systems of equations by iterative techniques is much more difficult [4, 13, 21].

However for general unsymmetric matrices there are some generalisations of CG available, such as Descent Methods [17], Bi-Conjugate Gradients [16], a Lanczos-type method introduced by Fletcher (1976) and which requires multiplication by the transpose matrix, Conjugate Gradient Squared [14], a faster converging variant that does not suffer from this drawback, introduced by Sonneveld (1984) and some others, as Generalized Conjugate Residual [19] and Orthomin [17, 20].

In some classes of problems, an irregular convergence behavior of CGS arises, notably when the iterations are started close to the solution. To correct this behavior Van der Vorst (1992) introduced the Bi-CGStab, a method with a convergence behavior much smoother, producing often much more accurate residual vectors and in most cases converging considerably faster then CGS [7, 9].

It is important to recall that a possible technique for solving nonsymmetric problems is to apply an iterative technique to the normal equations $A^t A\underline{x} = A^t\underline{b}$ in which the coefficient matrix is always symmetric and positive definite. This procedure of overcoming the problem of the unsymmetry of A has the disadvantage of being very expensive in memory requirements and time consumption. The condition number of A^tA is the square of the condition number of A, and so the convergence is slow, and in particular for ill conditioned systems roundoff may contaminate the results.

However, this method is guaranteed to converge anyhow (even in a finite number of iterations, neglecting roundoff), so it might be of value in situations where other methods that are cheaper per iteration fail.

In this paper the performance of different Conjugate Gradient type methods, for nonsymmetric systems, when applied in BEM tridimensional problems is analysed.

2 The Conjugate Gradient Method

Let A be a $n \times n$ SPD matrix and let $A\underline{x}=\underline{b}$, be the system to be solved. The solution of this system is equivalent to the minimization of the functional $\phi(\underline{x}) = \frac{1}{2}(\underline{x}, A\underline{x}) - (\underline{x}, \underline{b})$.

The minimum value of ϕ is $(\underline{b}, A^{-1}\underline{b})/2$ achieved by setting $\underline{x} = A^{-1}\underline{b}$.

For the minimization of ϕ we can use the strategy of the steepest descent: at a current point $\underline{x}_c$ the function ϕ decreases most rapidly in the direction of the negative gradient $-\bigtriangledown\phi(x_c) = \underline{b} - A\underline{x}_c$. If the residual of $\underline{x}_c$ ($\underline{r}_c = \underline{b} - A\underline{x}_c$) is nonzero, then $\phi(\underline{x}_c + \alpha\underline{r}_c) < \phi(\underline{x}_c)$ with α positive.

In the steepest descent method we set $\alpha = \frac{(\underline{r}_c, \underline{r}_c)}{(\underline{r}_c, A\underline{r}_c)}$ minimizing $\phi(\underline{x}_c + \alpha\underline{r}_c)$.

The initial approximation $\underline{x}_0$ may be 0, or if we have a better guess $\underline{x}^*$, this value.

Unfortunately the speed of convergence of the steepest descent method depends on the condition number $K(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ of matrix A, and is very slow for large $K(A)$ (λ_{max} and λ_{min} are the largest and the smallest eigenvalues of the SPD matrix).

The gradient directions that arise during the iteration process are not the best ones (they are too similar) slowing the process towards the minimum point. To avoid this situation we consider the sucessive minimization of ϕ along a set of directions $\{\underline{p}_1, \underline{p}_2, ...\}$ that do not necessarily correspond to the residuals $\{\underline{r}_0, \underline{r}_1, ...\}$.

Now to minimize $\phi(\underline{x}_{k-1}+\alpha\underline{p}_k)$ with respect to α we set $\alpha = \alpha_k = \frac{(\underline{p}_k,\underline{r}_{k-1})}{(\underline{p}_k,A\underline{p}_k)}$ and with this choice $\underline{x}_k = \underline{x}_{k-1} + \alpha_k\underline{p}_k$ and $\underline{x}_k \in \text{span}\ \{\underline{p}_1,\ldots\underline{p}_k\}$ $\quad((p_k, r_{k-1}) \neq 0)$.

The problem is the selection of the ideal vectors $\{\underline{p}_1,\underline{p}_2,\ldots\}$. A good approach is to choose linearly independent $\underline{p}_k$ with the property that each $\underline{x}_k$ solves

$$\min_{\underline{x} \in \text{span}\{\underline{p}_1,\ldots,\underline{p}_k\}} \phi(x) \qquad (1)$$

This would guarantee the global convergence and the finite termination too, because we must have $A\underline{x}_n = \underline{b}$.

The vector $\underline{p}_k$, solution of the one dimensional minimization problem min_α $\phi\ (\underline{x}_{k-1} + \alpha\underline{p}_k)$ is a partial solution of the K-dimensional minimization problem (1) too.

The vector $\underline{x}_k = \underline{x}_{k-1}+\alpha_k\underline{p}_k$ minimizes ϕ over the span of the search directions (the subspace span$\{\underline{p}_1,\ldots,\underline{p}_k\}$).

If it is possible to choose $\underline{p}_k$ such that $(\underline{p}_k, A\underline{p}_k) = 0$, $(\underline{p}_k,\underline{r}_{k-1}) \neq 0$ and to ensure a reduction in the size of the residuals choosing $\underline{p}_k$ in order to be the closest vector to $\underline{r}_{k-1}$ that is A-conjugate to $\underline{p}_1,\ldots,\underline{p}_{k-1}$ we have a first version of Conjugate Gradient method.

As $\underline{p}_k$ are nonzero vectors and nonzero A-conjugate vectors are linearly independent, either $\underline{r}_{k-1} = 0$ for some k$\leq$n, or we compute $\underline{x}_n$ which minimize ϕ over span$\{\underline{p}_1,\ldots,\underline{p}_n\} = R^n$ and so the finite termination of the problem (1) is guaranteed.

To compute $\underline{p}_k$ we set $\underline{p}_k = \underline{r}_{k+1} + \beta_k\underline{p}_{k-1}$ with $\beta_k = \frac{(\underline{p}_{k-1},A\underline{r}_{k-1})}{(\underline{p}k-1,A\underline{p}_{k-1})}$.

Applying A to both sides of $\underline{x}_k = \underline{x}_{k-1} + \alpha_k\underline{p}_k$, using the definition of the residual we get $\underline{r}_k = \underline{r}_{k-1} - \alpha_k A\underline{p}_k$ and after some calculation we obtain an efficient implementation of Conjugate Gradiente method:

Let A be a $n\times n$ SPD matrix, and $\underline{b} \in R^n$, then the following algorithm computes the solution of $A\underline{x} = \underline{b}$

```
k=0 ; x_0 = 0 ; r_0 = b
while r ≠ 0
        k=k+1
        if k=1
                p_1 = r_0
        else
                β_k = (r_{k-1}, r_{k-1}) / (r_{k-2}, r_{k-2})
                p_k = r_{k+1} + β_k p_{k-1}
        end
        α_k = (x_{k-1}, r_{k-1}) / (p_k, A p_k)
        x_k = x_{k-1} + α_k p_k ; r_k = r_{k-1} − α_k A p_k
end
x = x_k
```

(2)

This algorithm only requires one matrix-vector multiplication per iteration.

3 Conjugate Gradient-like Methods for Unsymmetric Systems

3.1 Descent Methods

Let A be a nonsymmetric $n \times n$ matrix with a positive definite symmetric part M ($M = \frac{(A+A^t)}{2}$).

We present four variants of descent methods that differ in their computational effort and storage requirements all of which have the following general form:

$$
\begin{aligned}
&\text{k=0 ; choose } \underline{x}_0 \text{ ; } \underline{r}_0 = \underline{b} - A\underline{x}_0 \text{ ; } \underline{p}_0 = \underline{r}_0 \\
&\text{While } \underline{r}_k \neq 0 \\
&\qquad \text{k=k+1} \\
&\qquad \alpha_k = \frac{(\underline{r}_k, A\underline{p}_k)}{(A\underline{p}_k, A\underline{p}_k)} \qquad (3) \\
&\qquad \underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k \text{ ; } \underline{r}_{k+1} = \underline{r}_k - \alpha_k A\underline{p}_k \\
&\qquad \text{Compute } \underline{p}_{k+1} \\
&\text{end} \\
&\underline{x}_k = \underline{x}_{k+1}
\end{aligned}
$$

The choice of α_k in (3) minimizes $\| \underline{r}_{k-1} \|_2 = \| \underline{b} - A(\underline{x}_k + \alpha \underline{p}_k) \|_2$ as a function of α, so that the Euclidean norm of the residual decreases at each step.

The four variants considered differ in the technique used to compute the new direction vector $\underline{p}_{k+1}$. The choice of this vector must result in a significant decrease in the norm of the residual $\| \underline{r}_{k+1} \|_2$ and must not require a large amount of computational work.

If we set
$$\underline{p}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{p}_k \qquad (4a)$$

where
$$\beta_k = \frac{(A\underline{r}_{k+1}, A\underline{p}_k)}{(A\underline{p}_k, A\underline{p}_k)} \qquad (4b)$$

we get another variant of CG known as the Conjugate Residual Method (CR).

Another option is to generate a set of A^tA-orthogonal directions using all the previous vectors $\{\underline{p}_k\}_{i=0}^k$ to compute $\underline{p}_{k+1}$

$$\underline{p}_{k+1} = \underline{r}_{k+1} + \sum_{i=0}^k \beta_i^{(k)} \underline{p}_i \qquad (5a)$$

where
$$\beta_i^{(k)} = -\frac{(A\underline{r}_{k+1}, A\underline{p}_i)}{(A\underline{p}_i, A\underline{p}_i)} \qquad (i \leq k) \qquad (5b)$$

This variant is known as the Generalized Conjugate Residual Method (GCR) and in the absence of roundoff errors gives the exact solution of the system in at most n iterations.

The work and storage requirements per iteration of GCR may be prohibitively high when n is large thus Vinsome (1976) has proposeded a modification of this method, significantly less expensive per iteration. Instead of making $\underline{p}_{k+1}$ A^tA-orthogonal to all preceding vectors $\{\underline{p}_i\}_{i=0}^k$ one can make $\underline{p}_{k+1}$ orthogonal to only the last l (≥ 0) vectors $\{\underline{p}_k\}_{i=k-l+1}^k$:

$$\underline{p}_{k+1} = \underline{r}_{k+1} + \sum_{i=k-l+1}^k \beta_i^{(k)} \underline{p}_i \qquad (l \geq 0) \qquad (6)$$

with $\{\beta_i^{(k)}\}_{i=k-l+1}^k$ defined in (5b). Only l direction vectors need to be saved. This variant is called Orthomin.

Finally, another alternative is to restart GCR periodically: every $l+1$ iterations, the current iterate $\underline{x}_i(l+1)$ is taken as the new starting guess (i counts the

number of restarts). This variant GCR(l) has the same storage requirements as Orthomin(l), the cost per iteration is however lower.

For the special case l=0, Orthomin(l) and GCR(l) are identical and

$$\underline{p}_{k+1} = \underline{r}_{k+1} \tag{7}$$

This method with very modest work and storage requirements is called Minimum Residual (MR).

3.2 Bi-Conjugate Gradient Method

A suitable algorithm for solving indefinite and unsymmetric systems is a different generalization of conjugate gradients, the Bi-conjugate Gradient algorithm, used by Lanczos as a mean of computing the eigenvalues of an unsymmetric matrix A. Two vectors $\underline{r}_1$ and $\underline{r}_1^*$ are given, and letting $\underline{p}_1 = \underline{r}_1$ and $\underline{p}_1^* = \underline{r}_1^*$, then for k=1,2,..., the following recurrence relations

$$\begin{aligned}
&\underline{r}_{k+1} = \underline{r}_k - \alpha_k A\underline{p}_k \; ; \qquad \underline{r}_{k+1}^* = \underline{r}_k^* - \alpha_k A^t \underline{p}_k^* \\
\text{with} \quad &\alpha_k = \frac{(\underline{r}_k, \underline{r}_k^*)}{(A\underline{p}_k, \underline{p}_k^*)} \\
\text{and} \quad &\underline{p}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{p}_k \; ; \qquad \underline{p}_{k+1}^* = \underline{r}_{k+1}^* + \beta_k \underline{p}_k^* \\
\text{with} \quad &\beta_k = \frac{(\underline{r}_{k+1}, \underline{r}_{k+1}^*)}{(\underline{r}_k, \underline{r}_k^*)}
\end{aligned} \tag{8}$$

are defined. The scalar α_k is choosen so as to force the biorthogonality condition $(\underline{r}_{k+1}, \underline{r}_k^*) = (\underline{r}_{k+1}^*, \underline{r}_k) = 0$ and β_k is choosen to force de biconjugacy condition $(\underline{p}_{k+1}^*, A\underline{p}_k) = (\underline{p}_{k+1} A^t \underline{p}_k^*) = 0$.

The algorithm (8) analogous to algorithm (2) has the feature that these two conditions also hold for any pair of vectors without having to be explicity enforced. Another characteristic of this algorithm is that it must terminate with $\underline{r}_{k+1} = \underline{r}_{k+1}^*$ in at most n iterations.

Starting from (8) it is possible to augment this algorithm in order to get the Bi-Conjugate Gradient method for solving the system $A\underline{x} = \underline{b}$, for general unsymmetric A.

We have:

$$\begin{aligned}
&k = 0; \underline{x}_0 = 0; \underline{r}_0 = \underline{b}_A \underline{x}_0 \\
&\underline{p}_0 = \underline{p}_0^* = \underline{r}_0 = \underline{r}_0^* \\
&\text{For k=0,1,2,...,m,...do} \\
&\qquad \alpha_k = \frac{(\underline{r}_k, \underline{r}_k^*)}{(A\underline{p}_k, \underline{p}_k^*)} \\
&\qquad \underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k \\
&\qquad \underline{r}_{k+1} = \underline{r}_k - \alpha_k A\underline{p}_k \; ; \; \underline{r}_{k+1}^* = \underline{r}_k^* - \alpha_k A^t \underline{p}_k^* \\
&\qquad \beta_k = \frac{(\underline{r}_{k+1}, \underline{r}_{k+1}^*)}{(\underline{r}_k, \underline{r}_k^*)} \\
&\qquad \underline{p}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{p}_k \; ; \; \underline{p}_{k+1}^* = \underline{r}_{k+1}^* + \beta_k \underline{p}_k^* \\
&\qquad \text{end} \\
&\underline{x}_k = \underline{x}_{k+1}
\end{aligned} \tag{9}$$

In this algorithm the explicit accurrence of A^t in the computation can be considered a disadvantage.

3.3 Conjugate Gradient Squared Method

The amount of work needed for the computation of the vectors $\underline{r}_k^*$ and $\underline{p}_k^*$ in (9) is about the same as that for the original CG method itself, so we observe that the computational work of the Bi-CG algorithm is twice the computational work for the CG algorithm.

In order to improve the situation Sonneveld (1984) derived a generalization of CG where the problem of the occurrence of A^t is avoided, via a polynomial equivalent of the CG algorithm. Taking the squares of computational steps it is possible to arrive at an algorithm for the squared polynomials and then to get a vectorial variant of this algorithm by substituition of A in the polynomials.

Let $\underline{x}_0$ be a starting estimate of the solution $\underline{x}$ of the nonsingular system $A\underline{x} = \underline{b}$ and let $\underline{r}_i$ be suitably chosen. Then the CGS algorithm reads as follow:

$$
\begin{aligned}
&\underline{r}_0 = \underline{b} - A\underline{x}_0 \\
&\underline{v}_0 = \underline{p}_0 = \underline{r}_i = \underline{r}_0 \\
&\text{For k=0,1,2,...,m,...,do} \\
&\qquad \alpha_k = \frac{(\underline{r}_i, \underline{r}_k)}{(\underline{r}_i, A\underline{p}_k)} \; ; \quad \underline{q}_{k+1} = \underline{v}_k - \alpha_k A\underline{p}_k \\
&\qquad \underline{x}_{k+1} = \underline{x}_k + \alpha_k(\underline{v}_k + \underline{q}_{k+1}) \\
&\qquad \underline{r}_{k+1} = \underline{r}_k - \alpha_k A(\underline{v}_k + \underline{q}_{k+1}) \qquad (10) \\
&\qquad \beta_k = \frac{(\underline{r}_i, \underline{r}_{k+1})}{(\underline{r}_i, \underline{r}_k)} \; ; \quad \underline{v}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{q}_{k+1} \\
&\qquad \underline{p}_{k+1} = \underline{v}_{k+1} + \beta_k(\underline{q}_{k+1} + \beta_k \underline{p}_k) \\
&\text{end} \\
&\underline{x} = \underline{x}_{k+1}
\end{aligned}
$$

Each step requires twice the amount of work necessary for original CG but not more than Bi-CG, and working with A^t is avoided.

3.4 Bi-CGStab

In many situations a quite irregular convergence behaviour of CGS occurs, in particular in situations when starting the iteration close to the solution. Motivated by this facts H. A. Van der Vorst (1992) proposed a more smoothly converging variant of Bi-CG but with the same attractive speed of convergence of CGS. The Bi-CGStab algorithm is derived via a polynomial equivalent identical to the Bi-CG scheme.

It reads as follows:

$$
\begin{aligned}
&\text{Let } \underline{x}_0 \text{be an initial guess; } \underline{r}_0 = \underline{b} - A\underline{x}_0 \\
&\underline{r}_0^* \text{ is an arbitrary vector, such that} \\
&(\underline{r}_0, \underline{r}_0^*) \neq 0, \text{ e. g. } \underline{r}_0^* = \underline{r}_0 \\
&\rho_0 = \alpha = \omega_0 = 1; \\
&\underline{v}_0 = \underline{p}_0 = 0; \\
&\text{For k=1,2,...,m,...do} \\
&\qquad \rho_k = (\underline{r}_0^*, \underline{r}_{k-1}) \; ; \quad \beta = \frac{(\rho_k/\rho_{k-1})}{(\alpha/\omega_{k-1})} \\
&\qquad \underline{p}_k = \underline{r}_{k-1} + \beta(\underline{p}_{k-1} - \omega_{k-1}\underline{v}_{k-1}) \qquad (11) \\
&\qquad \underline{v}_k = A\underline{p}_k \; ; \; \alpha = \frac{\rho_k}{(\underline{r}_0^*, \underline{v}_k)} \\
&\qquad \underline{s} = \underline{r}_{k-1} - \alpha\underline{v}_k; \quad \underline{t} = A\underline{s}; \quad \omega_k = \frac{(\underline{t}, \underline{s})}{(\underline{t}, \underline{t})}
\end{aligned}
$$

$$\underline{x}_k = \underline{x}_{k-1} + \alpha \underline{p}_k + \omega_k \underline{s} \; ; \; \underline{r}_k = \underline{s} - \omega_k \underline{t}$$

end

$$\underline{x} = \underline{x}_{k+1}$$

From the orthogonality property the algorithm (11) is guaranteed to have a finite termination in at most n iterations and the computational work per iteration step is identical to that of CGS.

4 Preconditioning

The preconditioning of symmetric indefinite and non-symmetric matrix systems is far from being a well established, effective or standard procedure, but it plays an important role in the performance of the methods [5, 8, 12].

In this section we refer the use of preconditioning with the aim of reduction of computational work required to obtain a good approximation to the solution. The literature on preconditioning for Conjugate Gradient type methods has been mainly concerned with its use on sparse matrices as those of finite differences and FEM. However the BEM matrices are dense, and so the use of standard incomplete factorization technique is not applicable.

In general we may say that the best preconditioner of a given system $A\underline{x} = \underline{b}$ is in some sense the inverse of A: the solution is then attained in only one iteration step. The problem is the amount of computational work needed for that in normal circumstances.

Considering this facts and that the simplest form of preconditioning is scaling the rows and columns of the matrix with the intention of obtaining a diagonal unit this technique was applied in the numerical examples of next section.

The scaling by the diagonal of the matrix A is in some respects optimal, since it approximatly minimizes the condition number of $D^{-1}A$ among all other diagonal scalings.

5 Numerical Results

In our experiments we consider two different examples to analyse the behavior with respect to accuracy and efficiency of the iterative techniques considered in this work when applied to the approximate solution of BEM systems. The experiments have been carried out in double precision floating point arithmetic (about 15 decimal places) on a HP9000/720.

The first problem is an external flow. We consider a very slow flow of an incompressible fluid about a solid sphere, Fig.1, that is analytically analysed in [15] and the results are compared with the exact solution.

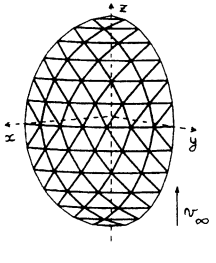

Fig.1 Flow of an incompressible fluid about a rigid sphere.

The second problem is a tridimensional heat flow problem with the geometrical definitions and boundary conditions shown in Fig.2 and we compare the results with the exact ones too.

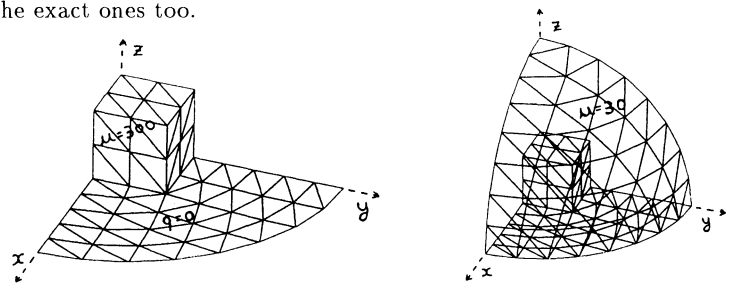

Fig.2 Heat flow problem (geometrical definitions and boundary conditions).

The matrices A(nxn) and $\underline{b}$ of the linear systems $A\underline{x} = \underline{b}$ to be solved are produced by a tridimensional Boundary Element computer program written in FORTRAN 77, after reordering the matrix equation $H\underline{u} = G\underline{q}$.

It is possible to use different types of triangular and quadrilateral elements and in our experiments we consider triangular elements with six nodes and quadrilateral elements with eight nodes. With the discretization used the resulting systems of equation have a dimension greater than one thousand.

The figures above show examples with a small number of nodes and elements (Fig.1 - 225 nodes and 98 elements, Fig.2 - 319 nodes and 142 elements) and possible symmetries were taken into account.

We employ the different CG type methods refered in this paper to obtain the solution of these systems and the results are presented here.

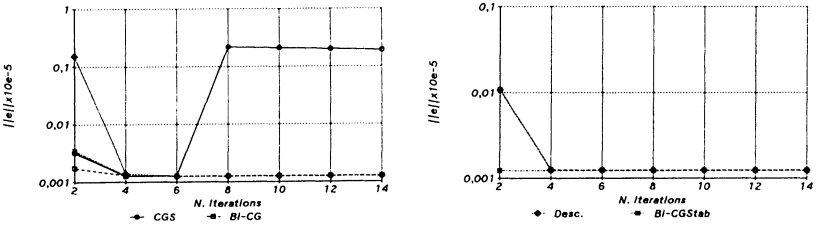

Fig.3 Results for the first problem (with and without preconditioning).

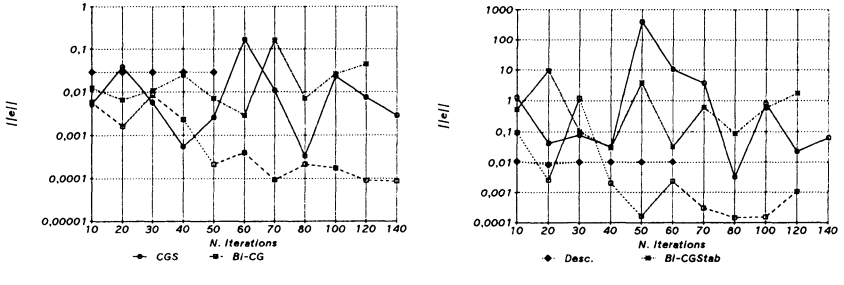

Fig.4 Results for the second problem (with and without preconditioning).

6 Conclusions

Some iterative techniques based on Conjugate Gradient solvers for the solution of the dense and unsymmetric systems of equations obtained by BEM have been presented.

From the results it becomes clear that these techniques perform differently, although in general well. For the first problem all techniques achieved a high convergence in a small number of iterations, with and without preconditioning. It seems that when the convergence is quickly achieved preconditioning is not an important factor.

For the second problem the iterative techniques performs differently and have more difficulty to achieve convergence. The results with preconditioning are in general better than without preconditioning and, of all iterative methods considered, the better one is the preconditioned Bi-CG.

One important conclusion is that the computational effort for these CG type methods obtain a good approximation to the solution, when applied in tridimensional BEM, is cheaper by a factor of ten or greater, depending on the dimension of the system, than the computational work needed by the Gauss elimination procedures, $O(n^3)$, an important factor for the solution of the large systems considered.

In order to exploit and confirm the potential of the methods, namely the assessment of the different preconditioners, further studies will be required.

References

[1] C. A. Brebia, J. C. Telles and L. C. Wrobel. *Boundary element techniques,* Springer-Verlag, Berlin, 1984.

[2] C. P. Jackson and P. Robinson. *A numerical study of various algorithms related to the preconditioned Conjugate Gradient method,* Int. J. Numer. Methods Engineering 21, 1315-1338, 1985.

[3] F. P. Valente. *Iterative methods for solving the systems of linear equations in BEM (in Portuguese),* Thesis for Prof. Coord., Instituto Politécnico da Guarda, Portugal, 1992.

[4] F. P. Valente and H. L. G. Pina. *Iterative solvers for BEM algebraic systems of equations,* Boundary Element Technology VIII, Computational Mechanics Publications, 1993.

[5] G. Brussino and V. Sonnad. *A comparison of direct and preconditioned iterative techniques for sparse, unsymmetric systems of linear equations,* Int. J. Numer. Methods in Enginnering, 28, 801-815, 1989.

[6] G. H. Golub and C. F. Van Loan. *Matrix computations,* The John Hopkins University Press, Baltimore, 1990.

[7] G. L. G. Sleijpen and D. R. Fokkema. *BiCGStab(l) for linear equations envolving unsymmetric matrices with complex spectrum,* Electronic Transactions on Numerical Analysis, 1, 11-32, 1993.

[8] H. A. Van Der Vorst. *High performance preconditioning,* SIAM, J. Sci. Stat. Computat., 10(6), 1174-1185, 1989.

[9] H. A. Van Der Vorst. *Bi-CGStab: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems,* SIAM, J. Sci. Stat. Computat., 13(2), 631-644, 1992.

[10] H. A. Van Der Vorst. *Parallel iterative solution methods for linear systems arising from discretized PDE's,* AGARD, 1995.

[11] H. P. Langtangen and A. Tveito. *A numerical comparison of Conjugate Gradient-like methods,* Communications in Applied Numerical Methods, 4, 793-798, 1988.

[12] H. P. Langtangen. *Conjugate Gradient methods and ILU preconditioning of non-symmetric matrix systems with arbitrary sparsity patterns,* Int. J. Numerical Methods in Fluids, 9, 213-233, 1989.

[13] J. H. Kane, D. E. Keyes e K. G. Prasad. *Iterative solution techniques in Boundary Element analysis,* Int. J. Numer. Methods in Engineering, 31, 1511-1536, 1991.

[14] P. Sonneveld. *CGS, A fast Lanczos-type solver for nonsymmetric linear systems,* SIAM, J. Sci. Stat. Computat., 10(1), 36-52, 1989.

[15] R. B. Bird, W. E. Stewart and E. N. Lightfoot. *Transport Phenomena,* John Wiley & Sons, New York, 1960.

[16] R. Fletcher. *Conjugate Gradient methods for indefinite systems,* Lecture Notes in Mathematics, 506, 73-89, Springer-Verlag, Berlin, 1976.

[17] S. C. Eisenstat, H. C. Elman and M. H. Schultz. *Variational iterative methods for nonsymmetric systems of linear equations,* SIAM, J. Num. Analysis, 20(2), 345-357, 1983.

[18] V. Faber and T. A. Manteuffel. *Orthogonal error methods,* SIAM, J. Num. Analysis, 24(1), 170-187, 1987.

[19] Y. Saad and M. H. Schultz. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM, J. Sci. Stat. Computat., 7(3), 856-864, 1986.

[20] Y. Saad. *The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems,* SIAM, J. Num. Analysis, 19(3), 485-506, 1982.

[21] W. J. Mansur, F. C. Araújo and J. E. B. Malaghini. *Solution of BEM systems of equations via iterative techniques,* Int. J. Numerical Methods in Engineering, 33, 1823-1841, 1992.