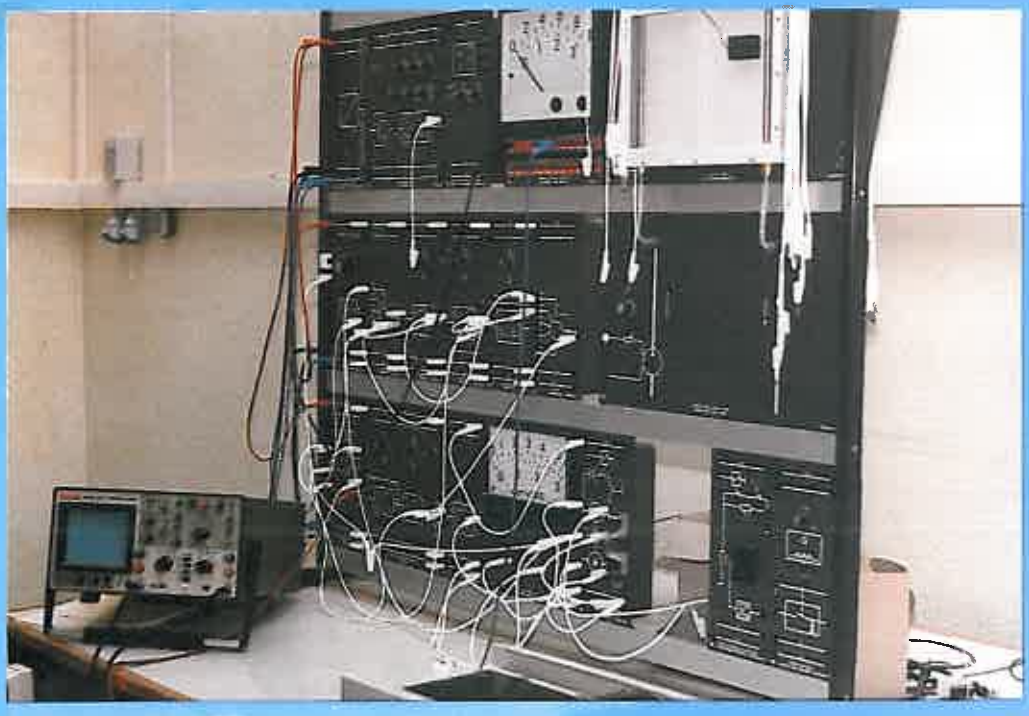
EDUCAÇÃO e TECNOLOGIA



Revista do Instituto Politécnico da Guarda

EDUCAÇÃO E TECNOLOGIA

Propriedade
Instituto Politécnico da Guarda

Director
João Bento Raimundo

Redacção
Serviços Centrais do I.P.G. - Av. Francisco Sá Carneiro nº 50
6300 Guarda
Telef. 222634 * Telecópia 222690

Composição
Gabinete Editorial do I.P.G.

Execução Gráfica e Impressão
Secção de Reprografia do I.P.G.

Periodicidade
Semestral

Tiragem
1.000 ex.

Depósito Legal
nº 17.981/87

nº XIV - Agosto de 1994

Evoluir e Agir

A valorização e o enriquecimento da nossa Revista tem sido uma preocupação constante, desde a sua primeira edição. Poderemos dizer que esta publicação tem caminhado a par com a própria evolução desta instituição de ensino superior, reflectindo a sua dinâmica, traduzindo a qualidade do ensino ministrado, incentivando a investigação, a edição de trabalhos inéditos, tracejando novas perspectivas.

Para além disso, e mercê da sua regularidade, do seu conteúdo, do seu contributo científico-cultural, a Revista "Educação e Tecnologia" é já hoje um título consagrado no contexto deste género de publicações, e com uma progressiva procura por parte de docentes, investigadores, homens de cultura e instituições.

É uma realidade que nos apraz registar. Sobretudo quando se trata de uma publicação, com estas características, editada no interior do País onde gera um diálogo cultural e onde intervém de forma idónea e responsável no processo subjacente ao papel do Instituto Politécnico da Guarda; instituição que no próximo ano lectivo aumentará substancialmente o seu número de alunos, que actualmente ultrapassa os três milhares.

É um número significativo, que confere à Guarda e à região toda uma vitalidade académica e social que honra os seus pergaminhos e as suas tradições estudantis de outrora, que a projecta, cada vez mais, no espaço nacional e europeu.

A Educação e a Tecnologia surgem, pois, como o quadro global em que se desenrola a actividade deste Instituto; daí que esta publicação seja sentida como um verdadeiro pilar e testemunho da sua acção, da sua capacidade interventiva. É sempre nesse sentido que continuaremos a caminhar.

João Raimundo
Presidente do IPG

ITERATIVE SOLVERS FOR *BEM* ALGEBRAIC SYSTEMS OF EQUATIONS

F. P. Valente*
H. L. G. Pina**

Abstract

Iterative techniques for the solution of the systems of algebric equations associated with the boundary element method are discussed in this paper. Due to the fact that the matrices associated with BEM are dense and sometimes ill conditioned, only direct methods like Gauss elimination, were used until recently to solve these systems. Here we present iterative techniques based on conjugate gradient solvers (CGS, Conjugate Gradient Squared and Bi-CG, Bi-Conjugate Gradients) that seem to have the potential to be competitive with direct methods.

Introduction

The conjugate gradient method (CG), first described by Hestenes and Stiefel (1952), has been widely used for approximating the solution of large sparse systems of linear

*Professor Coordenador na E.S.T.G., Instituto Politécnico da Guarda, Departamento de Matemática.
**Professor Catedratico do Instituto Superior Técnico, Departamento de Engenharia Mecanica.
Artigo apresentado no BETECH93 e publicado em Boundary Element Technology VIII, editado por Computational Mechanics Publications.

equations $\underline{A}\underline{x}=\underline{b}$, where A is an $(n \times n)$ real, symmetric positive-definite (SPD) matrix [3] [6] [12]. This method can be viewed as a direct method that, in the absence of roundoff errors, gives the exact solution in at most n steps, or as an iterative solver that gives a good approximation to the exact solution in a few steps.

Some essential properties of the CG method that makes it particularly suitable for large sparse systems are:

i) all references to A are in the form $A\underline{y}$ (matrix-vector product) yielding an algorithm which is cheap in both storage requirements and computing time;

ii) unlike most iterative methods, CG does not need estimation of any parameters.

These iterative techniques have been widely employed for solving FEM systems of equations, taking advantage of the fact that the system matrix is SPD.

Unfortunately, BEM matrices do not have these characteristics, which makes the CG algorithm not directly applicable and the solution of BEM systems of equations much more difficult.

However, there are some generalisations of CG available, such as Descent methods [13], Bi-Conjugate Gradients [11], Conjugate Gradient Squared [10], Generalized Conjugate Residual [9] [13], Orthomin [8] and Orthodir [12], etc., which are applicable to solve nonsymmetric systems of linear equations.

It is important to recall that a common technique for solving nonsymmetric problems is to apply the CG method to the normal equations

$$A^t A \underline{x} = A^t \underline{b} \qquad (1)$$

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 that the condition number of A^tA is the square of the condition number of A, and so the expected convergence will be 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 step fail.

In this paper the performance of different iterative methods when applied in BEM problems is analysed. Owing to the lack of studies about this subject different Conjugate Gradient type methods for nonsymmetric linear systems were considered.

1. The Conjugate Gradient method

Let A be an $(n \times n)$ SPD matrix and let $\underline{A}\underline{x}=\underline{b}$ be a system to be solved. The solution of this system is equivalent to minimize the functional $\phi(\underline{x})$ defined by $\phi(\underline{x}) = 1/2$

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

One possible strategy for minimizing ϕ is the method of steepest descent. At a current point $\underline{x}_c$ the function ϕ decreases most rapidly in the direction of the negative gradient $-\nabla\phi(\underline{x}_c) = \underline{b} - A\underline{x}_c$.

If $\underline{r}_c = \underline{b} - A\underline{x}_c$ (residual of $\underline{x}_c$) is nonzero, then there exists a positive α such that: $\phi(\underline{x}_c + \alpha\underline{r}_c) < \phi(\underline{x}_c)$

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

So we have the algorithm:

$$
\begin{array}{ll}
 & k = 0 \; ; \; \underline{x}_0 = 0 \; ; \; \underline{r}_0 = \underline{b} \\
 & \text{while } \underline{r}_k \neq 0 \\
 & \qquad k = k + 1 \\
(2) & \qquad \alpha_k = (\underline{r}_{k-1}, \underline{r}_{k-1})/(\underline{r}_{k-1}, A\underline{r}_{k-1}) \\
 & \qquad \underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{r}_{k-1} \\
 & \qquad \underline{r}_k = \underline{b} - A\underline{x}_k
\end{array}
$$

The initial guess $\underline{x}_0$ has not to be $\underline{0}$ and if we have a more appropriate guess $\underline{x}^*$ we merely replace $\underline{b}$ by $\underline{b} - A\underline{x}^*$.

The approximate solutions are then $\underline{x}_k + \underline{x}^*$.

Unfortunately the speed of convergence of the steepest descent method may be very slow if the condition number $K(A) = \lambda_{max}(A)/\lambda_{min}(A)$ is large. (λ_{max} and λ_{min} are the largest and smallest eigenvalues of the SPD matrix.)

The gradient directions that arise during the iteration are not the best ones (they are too similar) slowing progress towards the minimum point.

To avoid this situation we consider the sucessive minimization of ϕ along a set of directions $\{\underline{p}_1, \underline{p}_2, \ldots\}$ that do not necessarily correspond to the residuals $\{\underline{r}_0, \underline{r}_1, \ldots\}$.

Now to minimize $\phi(\underline{x}_{k-1} + \alpha\underline{p}_k)$ with respect to α we set

$$\alpha = \alpha_k = (\underline{p}_k, \underline{r}_{k-1})/(\underline{p}_k, A\underline{p}_k).$$

With this choice we have $\underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{p}_k$ and $\underline{x}_k \in \text{span}\{\underline{p}_1, \ldots, \underline{p}_k\}$ $((\underline{p}_k, \underline{r}_{k-1}) \neq 0)$

The problem is how to select the ideal vectors $\{\underline{p}_1, \underline{p}_2, \ldots\}$.

A good approach is to choose linearly independent $\underline{p}_i$ with the property that each $\underline{x}_k$ solves

$$\min \quad \phi(x) \qquad (3)$$

$$x \in \text{span}\{\underline{p}_1, \ldots, \underline{p}_k\}$$

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

The vector p_k , solution of the one dimensional minimization problem

$$\min_{\alpha} \phi(x_{k-1}+\alpha p_k)$$

is also a partial solution of the K-dimensional minimization problem (3).

With these conditions we obtain:

```
k = 0; x0 = 0; r0 = b
while rk ≠ 0
        k = k + 1
        choose pk ∈ span{Ap1,...,Apk-1}  so  (pk,rk-1)≠0
(4)     αk = (pk,rk-1)/(rk-1,Ark-1)
        xk = xk-1 + αk pk
        rk = b - Axk
end
```

The vector $x_k=x_{k-1}+\alpha_k p_k$ minimizes ϕ over the span of the search directions (the subspace span$\{p_1, \ldots, p_k\}$).

The algorithm (4) works if it is possible to choose p_k such that $(p_k,Ap_k)=0$ and $(p_k,r_{k-1})\neq 0$, which is the case, and, to ensure a reduction in the size of the residuals we must choose p_k in order to be the closest vector to r_{k-1} that is A-conjugate to $p_1,\ldots,p_{k-1}$.

So we have a first version of the method of conjugate gradient:

```
k = 0 ; x0 = 0 ; r0 = b
while  rk ≠ 0
        k = k + 1
        if  k = 1
                p1 = r0
        else
(5)             Let pk minimize ||p-rk-1||2 over all
                vectors p∈span{Ap1, ..., Apk-1}⊥
        end
        αk = (pk,rk-1)/(pk,Apk)
        xk = xk-1 + αk pk
        rk = b - Axk
end
x = xk
```

The finite termination of algorithm (5) is guaranteed since the p_k are nonzero and nonzero A-conjugate vectors are independent. Either $r_{k-1}=0$ for some $k \leq n$ or we compute x_n which minimizes ϕ over $\text{span}\{p_1,\ldots,p_n\} = R^n$.

To guarantee that (5) be an effective algorithm we need an efficient method for computing p_k. This can be done setting $p_k=r_{k+1}+\beta_k p_{k-1}$ with $\beta_k = =(p_{k-1},Ar_{k-1})/(p_{k-1},Ap_{k-1})$.

It seems that each loop needs three separate matrix-vector multiplications, but applying A to both sides of $x_k=x_{k-1}+\alpha_k p_k$ and using the definition of the residual we get

$$r_k = r_{k-1} - \alpha_k A p_k \qquad (6)$$

If we compute now the residuals recursively via (6) we have

$$(r_{k-1},(r_{k-1} = r_{k-2} - \alpha_{k-1} A p_{k-1}))$$
$$(r_{k-1},r_{k-1}) = -\alpha_{k-1} (r_{k-1} A p_{k-1})$$

and

$$(p_{k-1},(r_{k-1} = r_{k-2} - \alpha_{k-1} A p_{k-1}))$$
$$(r_{k-2},r_{k-2}) = \alpha_{k-1} (p_{k-1},A p_{k-1})$$

substituing into the formula for β_k, we obtain the more efficient inplementation (7)

Let A be an (nxn) symmetric positive definite matrix and $b \in R^n$,
then the following algorithm computes the solution of $Ax = b$.

(7)

```
k = 0 ; x_0 = 0 ; r_0 = b
while  r_k ≠ 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 = (r_{k-1},r_{k-1})/(p_k,Ap_k)
        x_k = x_{k-1} + α_k p_k
        r_k = r_{k-1} - α_k A p_k
end
x = x_k
```

As it can be seen in this algorithm only one matrix-vector multiplication is required per iteration.

There are other versions of the conjugate gradient method, but in all of them the rate of convergence is dependent

on the distribuition of the eigenvalues of the matrix **A** as we have already mentioned.

Here we present another possible version of the conjugate gradient method [3]

(8)

$k = 0$; $\underline{x}_k = 0$; $\underline{r}_k = \underline{p}_k = \underline{b} - A\underline{x}_k$
while $\underline{r}_k \neq 0$
$k = k + 1$
$\alpha_k = (\underline{p}_k, \underline{r}_k)/(\underline{p}_k, A\underline{p}_k)$
$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$
$\beta_k = -(\underline{r}_{k-1}, A\underline{p}_k)/(\underline{p}_k, A\underline{p}_k)$
$\underline{p}_{k+1} = \underline{r}_{k+1} + \beta \underline{p}_k$
end
$\underline{x} = \underline{x}_{k+1}$

As we have already referred, $\underline{x}_0$ has not to be **0**, and if we have a better guess it must be applied in the algorithms (7) and (8).

2. Descent methods, Bi-CG and CGS

As we have seen in the previous section, the conjugate gradient algorithms are only applicable when the systems have SPD matrices.

When this is not the case we have to work with some generalisation of CG .

In this section we will analyse three types of generalisation.

2.1. Descent methods

Let **A** be a nonsymmetric (**nxn**) matrix, with a positive-definite symmetric part **M** ($M=(A+A^t)/2$) and $\underline{b} \in \mathbf{R}^n$.

We will consider four variants [13], all of which have the following general form:

(9)

$k = 0$; choose $\underline{x}_0$; $\underline{r}_0 = \underline{b} - A\underline{x}_0$; $\underline{p}_0 = \underline{r}_0$
while $\underline{r}_k \neq 0$
$k = k + 1$
$\alpha_k = (\underline{r}_k, A\underline{p}_k)/(A\underline{p}_k, A\underline{p}_k)$
$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$
$\underline{r}_{k+1} = \underline{r}_k - \alpha_k A\underline{p}_k$
Compute $\underline{p}_{k+1}$
end
$\underline{x} = \underline{x}_{k+1}$

The choice of α_k in (9) minimizes

$\|r_{k+1}\|_2 = \|b-A(x_k+\alpha p_k)\|_2$ as a function of α, so that the Euclidean norm of the residual decreases at each step. The four variants differ in the technique used to compute the new direction vector p_{k+1}.

If we set $$p_{k+1} = r_{k+1} + \beta_k p_k \qquad (10a)$$

where $$\beta_k = (Ar_{k+1}, Ap_k)/(Ap_k, Ap_k) \qquad (10b)$$

we have another variant of CG known as the Conjugate Residual method (CR).

Another option is $$p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \beta_i^{(k)} p_i \qquad (11a)$$

where $$\beta_i^{(k)} = -(Ar_{k+1}, Ap_i)/(Ap_i, Ap_i) \qquad (i\le k) \qquad (11b)$$

This variant is known as the generalized conjugate residual method (GCR).

The work and storage requirements per iteration of GCR may be prohibitively high when n is large thus Vinsome (1976) has proposed a modification of this method, significantly less expensive per iteration. It is called Orthomin and in that variant we set

$$p_{k+1} = r_{k+1} + \sum_{i=k-l+1}^{k} \beta_i^{(k)} p_i \quad (l \ge 0) \qquad (12)$$

with $\{\beta_i^{(k)}\}_{i=k-l+1}^{k}$ defined as in (11b).

Finally, another alternative is to restart GCR periodically: every $l+1$ iterations, the current iterate $x_i(l+1)$ 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), however the cost per iteration is lower.

For the special case $l=0$, Orthomin(l) and GCR(l) are identical with $p_{k+1} = r_{k+1}$ and so we have a method with very modest work and storage requirements called Minimum Residual method (MR).

2.2 . Bi-CG - Bi-Conjugate Gradient Method

Another generalization for nonsymmetric linear systems is the so-called Bi-Conjugate Gradients algorithm (Bi-CG, [11]).

Let $A\underline{x}=\underline{b}$ be a nonsingular system and $\underline{x}_0$ an initial guess for the solution $\underline{x}$, with the corresponding residual $\underline{r}_0=\underline{b}-A\underline{x}_0$.

Then we have the following algorithm, based on a Lanczos-type method of computing the eigenvalues of an unsymmetric matrix A:

(13)

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

There are other possible versions of this method like this one [3]:

(14)

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

In all of them the explicit occurrence of A^t in the computation can be considered as a disadvantage.

2.3. CGS - Conjugate Gradient Squared method

This generalization for nonsymmetric linear systems where the problem of the occurrence of A^t does not occur is derived via a polynomial equivalent of the CG algorithm and reads as follows [10] :

$$
\begin{aligned}
&\underline{r}_0 = \underline{b} - A\underline{x}_0 \\
&\underline{v}_0 = \underline{p}_0 = \underline{r}\dot{i} = \underline{r}_0 \\
&\text{For} \quad k = 0,1,2,\ldots,m,\ldots \quad \text{do} \\
&\qquad \underline{q}_{k+1} = \underline{v}_k - \alpha_k A\underline{p}_k \\
&\qquad \underline{x}_{k+1} = \underline{x}_k + \alpha_k (\underline{v}_k + \underline{q}_{k+1}) \\
&\qquad \underline{r}_{k+1} = \underline{r}_k - \alpha_k A (\underline{v}_k + \underline{q}_{k+1}) \\
&\qquad \beta_k = (\underline{r}\dot{i}, \underline{r}_{k+1})/(\underline{r}\dot{i}, \underline{r}_k) \\
&\qquad \underline{v}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{q}_{k+1} \\
&\qquad \underline{p}_{k+1} = \underline{v}_{k+1} + \beta_k (\underline{q}_{k+1} + \beta_k \underline{p}_k) \\
&\text{end} \\
&\underline{x} = \underline{x}_{k+1}
\end{aligned}
\tag{15}
$$

(The vector $\underline{r}\dot{i}$ must be suitably chosen.)

3. Preconditioning

In this section we refer to the use of preconditioning with the aim of reduction of the number of iteration steps required to obtain a good approximation to the solution of $A\underline{x}=\underline{b}$. The literature on preconditioning for the conjugate gradient type methods has been mainly concerned with its use on sparse matrices. The BEM matrices do not have a sparsity pattern as they are denses, and so the use of standard incomplete factorization techniques is not applicable.

The best preconditioner, 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 that will be necessary to construct the inverse.

Considering these facts, the simplest form of preconditioning is scaling the rows and columns of the matrix **A** with the intention of obtaining a diagonal unit.

$$D^{-1} A \underline{x} = D^{-1} \underline{b}$$

The scaling by the diagonal of **A** is in some respects optimal, since it approximately minimizes the condition number of $D^{-1} A$ among all other diagonal scalings [6].

4. Numerical Results

Three test cases were designed and performed to analyse the behaviour with respect to accuracy and efficiency of the iterative techniques when applied to BEM problems.

Two of them (**1** and **2**) are potential problems with the geometrical definitions and boundary conditions shown in Fig. 1 (64 elements).

The other (**3**) is a two dimensional elastostatic problem with the geometrical definitions shown in Fig. 2 (34 elements).

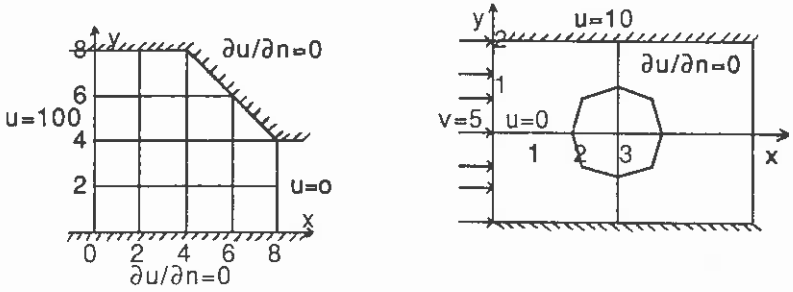


Fig. 1 Geometrical definitions and boundary conditions for cases 1 and 2

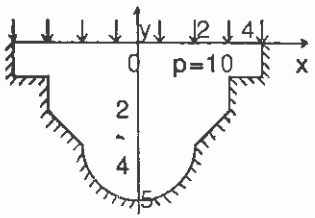


Fig. 2 Geometrical definitions for case 3

The matrices $A(n \times n)$ and $\underline{b} \in R^n$ of the linear systems $A\underline{x}=\underline{b}$ to be solved for the three test cases, are produced by two Boundary Element computer programs, after reordering the matrix equation $H\underline{u}=G\underline{q}$. One is for the potential problems and the other for the elastostatic problem which are described in [1], [2] . Both are in FORTRAN77 and use linear elements.

We try to employ all the CG type methods refered in this paper to the three cases.

In case 3, the matrix **A** of the system is dense, unsymmetric, but has a predominant diagonal. All the methods worked well, some (Bi-CG and CGS) performed very well yielding the solution in a small number of iterations; the others

not so well, they had some difficulty in achieving a good convergence.

For the potential problems (cases **1** and **2**) the iterative solution is much more difficult. The matrix **A** is dense, unsymmetric, has no dominant diagonal and is strongly ill-conditioned. As a consequence, most of the CG type methods did not work at all failing to get convergence. Only Bi-CG (algorithm **13**) and CGS (algorithm **15**) have been able to solve the systems corresponding to these problems.

In these cases, preconditioning reduced the number of iterations needed for convergence very significantly.

It is important to say that the methods that failed the convergence without preconditioning continue to fail convergence with preconditioning, they are not able to solve these systems.

The performance of the different methods for the three applications are shown in Fig. 3, 4 and 5, with and without preconditioning.

We compare the efficiency of the different iterative solvers with that of a direct Gauss elimination procedure, using the Euclidean norm of error at each iteration as an element of comparison and stoping the iterative procedure when the norm of error become stable.

For case 3, where some methods present a good performance and others not so good, we choose a comparison between the CG method and the CGS method.

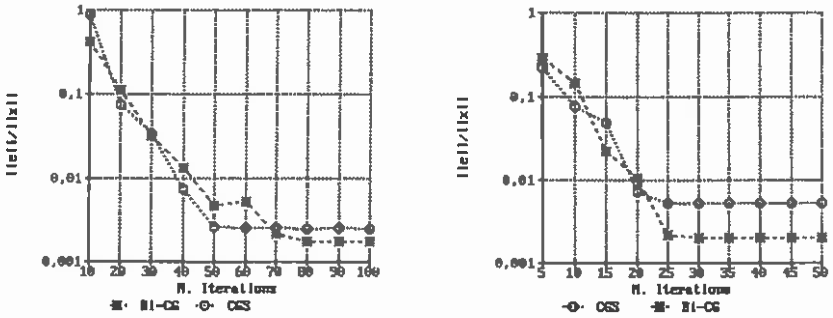


Fig. 3 Results for case 1 without and with preconditioning

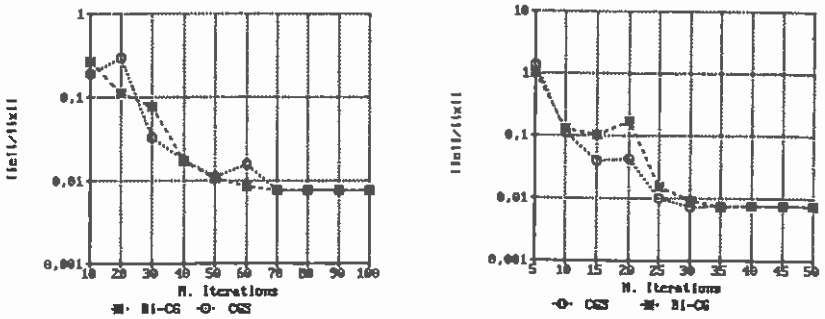


Fig. 4 Results for case 2 without and with preconditioning

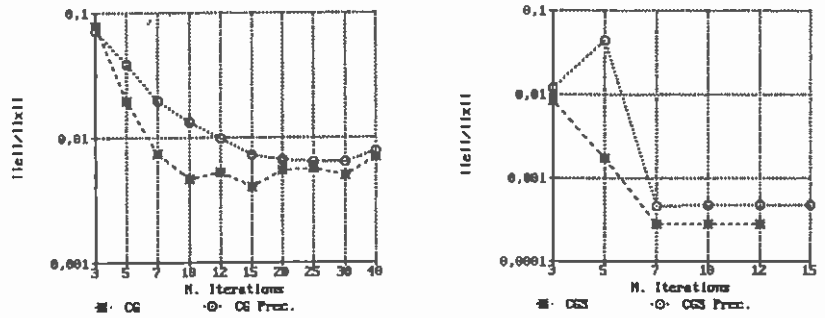


Fig. 5 Results for case 3 without and with preconditioning

The tests were performed on a HP9000/720 using simple precision.

5. Normal Equations

We have already seen that one way to get around the difficulties caused by the unsymmetry of A is solving the normal equations (1) : $A^tA\underline{x} = A^t\underline{b}$

This procedure has the disadvantages that we refered to in the intro-duction.

However, to analyse how it works we have applied this technique to the three test cases, for all the iterative methods considered in this paper. As was expected, all worked well and achieved the convergence, although the necessary number of iterations and the computational work became much greater.

As an example we present the results of the solution of normal equations for the CG method (algorithm **7**) with and without preconditioning, for cases **1** and **2**.

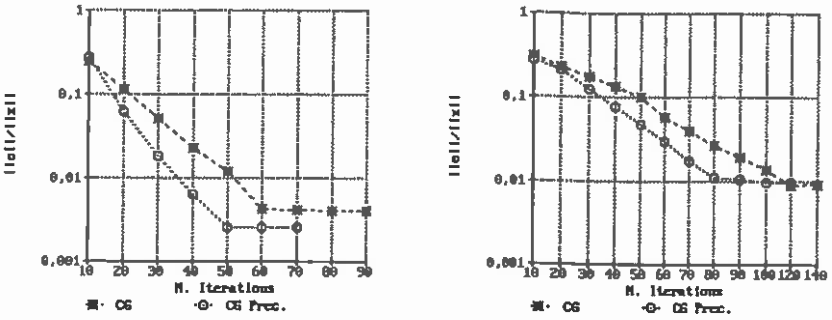


Fig. 6 Results of normal equations for cases 1 and 2

Conclusions

Some iterative methods for solving the dense, nonsymmetric and ill conditioned systems of equations obtained by the Boundary Element Method have been presented.

It has become clear that the several CG type methods presented perform differently when applied in BEM .

Only Bi-CG and CGS have been able to achieve convergence in the more difficult cases, that are the potential problems, and even in the others these two methods showed a better performance.

When convergence is achieved, preconditioning is an efficient technique for reducing the number of iterations to solution as can be seen in Fig. 3, 4 and 5. Only when the matrices present favorable characteristics (predominant diagonal), preconditioning seems to have no sensible effect.

The computational effort needed when Bi-CG and CGS are applied with preconditioning seems to be cheaper than the computational work needed for the Gauss elimination procedures (proportional to n^3 , n being the order of system matrix). This is an important factor for large three dimensional nonsymmetric BEM systems, where large computational savings can be expected.

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

References

C. A. Brebbia, *The boundary element method for engineers*, Pentech Press, London, 1984.

C. A. Brebbia, J. C. Telles and L. C. Wrobel, *Boundary element techniques*, Springer-Verlag, Berlin, 1984.

C. P. Jackson and P. C Robinson, *A numerical study of various algorithms related to the preconditioned conjugate gradient method*, Int. J. Numer. Meth. Engineering, 21, pp. 1315-1338, 1985.

D. Howard, W. M. Connolley and J. S. Rollett, *Unsymmetric conjugate gradient methods and sparse direct methods in finite element flow simulation*, Int. J. Numer. Meth. in Fluids, 10, pp. 925-945, 1990.

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

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

H. A. Van Der Vorst and K. Dekker, *Conjugate gradient type methods and preconditioning*, J. Comp. Appl. Mathematics, 24, pp. 73-87, 1988.

J. H. Kane, D. E. Keyes and K. Guru Prasad, *Iterative solution techniques in boundary element analysis*, Int. J. Numer. Meth. in Engineering, 31, pp. 1511-1536, 1991.

P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, J. Sci. Stat. Comput, 10(1), pp. 36-52, 1989.

R. Fletcher, *Conjugate gradient methods for indefinite systems*, Lecture Notes in Mathematics, Vol. 506, pp. 73-89, Springer-Verlag, Berlin, 1976.

S. F. Ashby, T. A. Manteuffel and P. E. Saylor, *A taxanomy for conjugate gradient methods*, J. Numerical Analysis, 27(6), pp. 1542-1568, 1990.

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

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

Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, J. Sci. Stat. Comput., 7(3), pp. 856-859, 1989.

W. J. Mansur, F. C. Araújo and J. E. B. Malaghini, *Solution of BEM systems of equations via iterative techniques*, Int. J. Numer. Meth. Engineering, 33, pp. 1823-1841, 1992.