



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 160 (1998) 101–114

**Computer methods
in applied
mechanics and
engineering**

Efficient iterative solution of constrained finite element analyses

Pascal Saint-Georges*, Yvan Notay, Guy Warzée

*Continuum Mechanics Dept. CP 194/5 & Nuclear Metrology Dept. CP 165 Université Libre de Bruxelles, 50 av. F.D. Roosevelt,
B-1050 Bruxelles, Belgium*

Received 19 August 1996; accepted 18 August 1997

Abstract

This contribution describes how iterative solvers can meet specific requirements of industrial FE analyses, focusing on the case frequently met where the unknowns are subject to linear equality constraints. Standard iterative methods designed to deal with that kind of problem suffer from a significant overhead with respect to the CPU times involved in the solution of unconstrained problems, whereas the performance of direct solvers traditionally used is not affected. Here we propose a subspace projection method that allows the use of any iterative scheme able to solve the unconstrained problem, with the same preconditioner. We also highlight that there is no loss of efficiency due to the presence of linear constraints. © 1998 Elsevier Science S.A. All rights reserved.

1. Introduction

The finite element (FE) discretisation of a linear elastic structure submitted to nodal loads f leads to the problem of finding the displacement vector \tilde{q} that minimises the total energy W^* amongst all possible displacement vectors $q \in \mathcal{R}^n$. The total energy functional is defined as

$$W^*(q) = \frac{1}{2} \langle q, Kq \rangle - \langle q, f \rangle \quad (1)$$

where K is the stiffness matrix of order n and $\langle u, v \rangle$ denotes the inner product of two vectors u and v . The boundary conditions are assumed to be included in the expression of the matrix, e.g. by removing the corresponding lines and columns for the Dirichlet boundary conditions. In many circumstances some unknowns are linked by additional linear equations written as

$$Cq = c \quad (2)$$

with C figuring the c by n rectangular constraint matrix of rank c (redundant constraints are supposed to have been eliminated in a former stage). Such linear constraints occur for instance

- when one or more regions behave as rigid bodies, C and c are obtained by prescribing constant distances between all nodes belonging to that region;
- for the connection of parts of the structure that have been meshed separately and do not satisfy the continuity conditions of the grid;
- when the use of different levels of refinement in different regions introduces ‘slave nodes’ in the fine mesh,

* Corresponding author.

- at which the values of the unknowns are determined by interpolation of the solution at the neighbouring ‘master nodes’ of the coarse mesh;
- the prescription of cyclic periodicity conditions on a mesh that represents only a part of the whole structure, as illustrated in an example of Section 4.
 - for the modelling of an elastic soil by prescribing the average displacement of the nodes lying onto the soil;
 - if the value of the displacement following a given direction d normal or parallel to the edge of the structure at some node N has to be prescribed, leading to a boundary condition of the form

$$\mathbf{q}(N_x) \cos(x, d) + \mathbf{q}(N_y) \cos(y, d) + \mathbf{q}(N_z) \cos(z, d) = \text{prescribed value} \quad (3)$$

where $\cos(i, d)$ denotes the cosine of the angle between directions i and d . A boundary condition like (3) would not be necessary if local axes were introduced at node N but expressing the stiffness matrix in local axes hinders the use of some very efficient preconditioners [19,21].

When there is no constraint, the vector of displacements $\tilde{\mathbf{q}}$ that minimises (1) is the solution of the (symmetric positive definite) system

$$\mathbf{K}\tilde{\mathbf{q}} = \mathbf{f} \quad (4)$$

Industrial finite element (FE) softwares generally hang on a frontal or skyline solution procedure to solve Eq. (4). The motivation for this choice is related to the well-known robustness of direct solvers, which however, have such large disk requirements that the size of the problems that can be solved is limited to roughly 100 000 unknowns on current engineering workstations. The demand for solvers that are economical with respect to disk space has grown in the last few years and iterative solution techniques have become an important field of research for industrial software developers.

Some commercial softwares already include iterative solvers for well-conditioned systems (e.g. in thermics) like MARC K5/K6 [14,25] or COSMOS/FFE [4]. Others address elasticity problems but exhibit lack of robustness in some cases (for critical values of the Poisson ratio for instance) like NASTRAN [4,12], or use iterative schemes that could be enhanced by a better preconditioning [20] like DIANA [24].

The solver of the authors [20,21], currently implemented in the industrial FE software SAMCEF [22], has now reached an elaborate stage of development and has overcome the drawbacks mentioned above, outperforming high-quality direct solvers thanks to high-performance preconditioners. In the presence of linear constraints (2), the comparison generally becomes less favourable to iterative solvers. Indeed, direct solvers are easily adapted to compute the solution of the constrained system without generating any significant increase of the solution time. On the other hand, an overview of standard techniques leads to the conclusion that iterative solvers suffer from a significant overhead even for a small number of constraints. This led us on the way to the development of a subspace projection technique. Whenever combined with the unpreconditioned conjugate gradient algorithm, the latter reduces to the projected gradient method which has gained popularity in the field of constrained optimisation [9]. This method appears not to have been proposed before in the present context. Moreover, because we do not have to deal with inequality constraints, our approach is more general in the sense that it can be linked to any iterative scheme and naturally allows the introduction of preconditioning. In particular, we show that it is possible to use the same preconditioner as for solving the unconstrained problem (4) with a similar convergence rate. Hence, subspace projection makes iterative solvers as competitive for both constrained and unconstrained problems.

An overview of standard techniques for dealing with constrained problems is made in Section 2. Section 3 gives a simple presentation of the subspace projection method. Numerical results are presented in Section 4.

2. Standard methods for constrained problems

2.1. Elimination of dependent unknowns

A first way to deal with constraints is to split the unknowns of \mathbf{q} into two subsets grouping dependent and independent unknowns \mathbf{q}_d and \mathbf{q}_i , defined by rewriting Eq. (2) as

$$\mathbf{q}_d = \mathbf{A}\mathbf{q}_i + \mathbf{a} \quad (5)$$

where A is a c by $n-c$ dependency matrix computed from C . The matrix K and nodal loads f are split accordingly,

$$K = \begin{bmatrix} K_{ii} & K_{id} \\ K_{di} & K_{dd} \end{bmatrix}; \quad f = \begin{Bmatrix} f_i \\ f_d \end{Bmatrix} \tag{6}$$

When K , f and q are replaced into Eq. (1), the elimination of the dependent unknowns thanks to Eq. (5) gives

$$W_A(q_i) = \frac{1}{2} \langle q_i, K_A q_i \rangle - \langle q_i, f_A \rangle \tag{7}$$

with

$$K_A = K_{ii} + K_{id}A + A^t K_{di} + A^t K_{dd}A \tag{8}$$

and f_A is a function of f_i , f_d , A and a . The problem of minimising (1) under constraints (2) has been transformed into the unconstrained minimisation of (7), which is equivalent to solving system (9),

$$K_A \tilde{q}_i = f_A \tag{9}$$

When considering the solution of (9) by a direct solver, the main difficulty is the computation of the factorisation of K_A which may be impossible since K_A is generally much denser than K —a major bottleneck for large FE models—and difficult to form explicitly within reasonable time and disk resources. Direct solvers are therefore employed mainly if the dependency (5) can be written at the element level. On the other hand, an iterative scheme can always be used but gives rise to the problem of finding a (good) preconditioner for K_A .

2.2. Lagrange multipliers

Unlike the dependent/independent unknowns splitting, the Lagrange multiplier technique is widely used. The introduction of a set of additional unknowns λ leads to the unconstrained minimisation of

$$W_L(q_L) = \frac{1}{2} \langle q_L, K_L q_L \rangle - \langle q_L, f_L \rangle \tag{10}$$

with

$$K_L = \begin{bmatrix} K & C^t \\ C & 0 \end{bmatrix}; \quad q_L = \begin{Bmatrix} q \\ \lambda \end{Bmatrix}; \quad f_L = \begin{Bmatrix} f \\ c \end{Bmatrix}$$

The minimiser \tilde{q}_L of (10) is the solution of the system

$$K_L \tilde{q}_L = f_L \tag{11}$$

Assuming that K_L is regular, a solver based on Gaussian elimination is able to find its unique solution provided the pivots are carefully chosen, which is easily obtained by leaving the c last lines of K_L corresponding to the constraints for the end of the factorisation. It is readily seen that a small number c of constraints will not significantly slow down the direct solver when compared to the unconstrained case. On the contrary, an iterative solver is dramatically affected by the presence of Lagrange multipliers, mainly because the new system matrix is indefinite.

Classical iterative methods for dealing with systems like (11) may be subdivided into three families:

- (1) Methods resorting to inner–outer iterations, based on the (semi-)positive definiteness of K that makes a system with matrix K efficient to be solved by an inner iteration. Uzawa’s method [1,13] and Axelsson’s regularisation [3] are typical techniques. They are not satisfactory in the present context since m outer iterations require m resolutions of a system with matrix K , meaning that the solution of the constrained system is m times slower than that of the unconstrained one. Such an overhead is unacceptable when there are only a few constraints with respect to the number of unknowns \tilde{q} ; besides, these methods have been proposed for systems of the form (11) presenting a large number of unknowns $\tilde{\lambda}$ in the second block.
- (2) Preconditioning of system (11) such that the preconditioned system is positive definite. Ashby et al. [2] propose a polynomial preconditioning scheme but this kind of preconditioner, if effective for the purpose

considered here, is known to yield a poorly conditioned system and in most cases the number of iterations is very large. A second preconditioning step is therefore suggested by the authors but as the new system matrix is not explicitly formed, efficient preconditioning is a considerably more difficult problem than preconditioning a matrix like \mathbf{K} . Block factorisation has been used instead of polynomial preconditioning by Axelsson [3] and Saint-Georges [19] but the obtained preconditioner has not been found robust enough.

- (3) The application of some generalised CG method able to cope with indefinite systems, like BiCG [6], BiCG-STAB [23], QMR [7], GMRES [18], SYMMLQ [16]. However, even with an ‘optimal’ method like GMRES or SYMMLQ the rate of convergence is not as attractive as for positive definite systems. Finding an efficient preconditioner is also a much more difficult task for which several possibilities have been investigated by the authors in [19] but none of them were found satisfactory.

3. The subspace projection method

3.1. Subspace projection

As a conclusion to the previous section, the efficiency of an iterative solver deteriorates in the presence of linear constraints if dependent unknowns are eliminated or Lagrange multipliers are introduced, while a direct solver is not affected. This bottleneck is overcome thanks to the subspace projection method, whose origin goes back to works in optimisation methods [5,9,17]. The presentation given in this section proceeds in two steps: first, the problem is reduced to a constrained minimisation with homogeneous constraints; second, this latter is transformed into an unconstrained minimisation whose solution is sought only in the subspace containing all (but only) vectors that satisfy the homogeneous constraints. This subspace \mathcal{R} of \mathcal{R}^n , is then defined by

$$\mathcal{R} \equiv \{\mathbf{q} \in \mathcal{R}^n \mid \mathbf{C}\mathbf{q} = \mathbf{0}\} \quad (12)$$

Assume that one has found one vector \mathbf{q}_c satisfying

$$\mathbf{C}\mathbf{q}_c = \mathbf{c}$$

It is always possible to decompose any vector $\mathbf{q} \in \mathcal{R}^n$ as

$$\mathbf{q} = \mathbf{q}_0 + \mathbf{q}_c \quad (13)$$

with

$$\begin{aligned} \mathbf{C}\mathbf{q} &= \mathbf{C}\mathbf{q}_0 + \mathbf{C}\mathbf{q}_c \\ &= \mathbf{C}\mathbf{q}_0 + \mathbf{c} \end{aligned}$$

so that \mathbf{q} satisfies the constraints (2) if and only if $\mathbf{C}\mathbf{q}_0 = \mathbf{0}$, i.e. if

$$\mathbf{q}_0 \in \mathcal{R} \quad (14)$$

According to Eq. (13),

$$\begin{aligned} \mathbf{W}^*(\mathbf{q}) &= \frac{1}{2} \langle \mathbf{q}_0 + \mathbf{q}_c, \mathbf{K}(\mathbf{q}_0 + \mathbf{q}_c) \rangle - \langle \mathbf{q}_0 + \mathbf{q}_c, \mathbf{f} \rangle \\ &= \frac{1}{2} \langle \mathbf{q}_0, \mathbf{K}\mathbf{q}_0 \rangle + \langle \mathbf{q}_0, \mathbf{K}\mathbf{q}_c \rangle - \langle \mathbf{q}_0, \mathbf{f} \rangle + \frac{1}{2} \langle \mathbf{q}_c, \mathbf{K}\mathbf{q}_c \rangle - \langle \mathbf{q}_c, \mathbf{f} \rangle \\ &= \frac{1}{2} \langle \mathbf{q}_0, \mathbf{K}\mathbf{q}_0 \rangle - \langle \mathbf{q}_0, \mathbf{f} - \mathbf{K}\mathbf{q}_c \rangle + \text{constant} \end{aligned} \quad (15)$$

where the constant term does not depend on \mathbf{q} . This defines

$$\underline{\mathbf{W}}^*(\mathbf{q}_0) = \frac{1}{2} \langle \mathbf{q}_0, \mathbf{K}\mathbf{q}_0 \rangle - \langle \mathbf{q}_0, \mathbf{f} - \mathbf{K}\mathbf{q}_c \rangle$$

and the problem of finding the minimum of $W^*(q)$ while satisfying the constraints (2) is transformed into the search for a vector q_0 that minimises $\underline{W}^*(q_0)$ while q_0 satisfies the homogeneous constraints (14).

Now, let \mathcal{P} be some linear projector from \mathcal{R}^n to $\underline{\mathcal{R}}$, i.e. some matrix satisfying

$$\forall q \in \mathcal{R}, \quad \mathcal{P}q = q_0 \quad \text{with } q_0 \in \underline{\mathcal{R}}; \quad \mathcal{P}q_c = 0 \tag{16}$$

according to (13), which implies

$$\forall q_0 \in \underline{\mathcal{R}}: \quad \mathcal{P}q_0 = q_0$$

The new problem is then

$$\text{find } \hat{q} \in \mathcal{R}^n \text{ such that } \underline{W}^*(\mathcal{P}\hat{q}) \leq \underline{W}^*(\mathcal{P}q) \quad \forall q \in \mathcal{R}^n \tag{17}$$

$$\text{and set } \tilde{q} = \mathcal{P}\hat{q} + q_c \tag{18}$$

However,

$$\begin{aligned} \underline{W}^*(\mathcal{P}q) &= \frac{1}{2} \langle \mathcal{P}q, K\mathcal{P}q \rangle - \langle \mathcal{P}q, f - Kq_c \rangle \\ &= \frac{1}{2} \langle q, \mathcal{P}^t K \mathcal{P}q \rangle - \langle q, \mathcal{P}^t (f - Kq_c) \rangle \\ &= \frac{1}{2} \langle q, K_p q \rangle - \langle q, f_p \rangle \end{aligned}$$

with

$$K_p = \mathcal{P}^t K \mathcal{P}; \quad f_p = \mathcal{P}^t (f - Kq_c) \tag{19}$$

Therefore, the minimiser \hat{q} of problem (17) can rather be computed by solving the system

$$K_p \hat{q} = f_p \tag{20}$$

Note that K_p is a singular matrix because its product with any vector x such that $x = (I - \mathcal{P})x$ is zero.

3.2. The proposed algorithm for the solution of constrained problems

Keeping in mind that we aim at using an iterative scheme for the solution of (20), products with matrix K_p will have to be performed repeatedly. So a practical requirement for selecting \mathcal{P} is that the multiplication by \mathcal{P} and \mathcal{P}^t has to remain cheap. In this view an interesting family of projectors is

$$\mathcal{P} = I - \Gamma^t (C\Gamma^t)^{-1} C \tag{21}$$

where Γ is some (sparse) c by n matrix of rank c . We leave to the reader the easy demonstration that \mathcal{P} satisfies requirement (16). With this choice, an admissible vector q_c is obtained by

$$q_c = \Gamma^t (C\Gamma^t)^{-1} c \tag{22}$$

Thus, the proposed solution procedure is

- (a) choose some Γ and build $C\Gamma^t$ explicitly, which is cheap provided both matrices are sparse and $c \ll n$;
- (b) store the triangular factors of the LU factorisation of $C\Gamma^t$;
- (c) compute q_c and f_p by Eqs. (22) and (19) respectively;
- (d) compute a solution \hat{q} of the singular system (20) by some relevant iterative method;
- (e) compute the solution \tilde{q} of the constrained minimisation problem via Eq. (18).

3.3. Choosing the solver

Computing \hat{q} at step (d) is achieved by any iterative method able to deal with singular consistent (positive semi-definite) systems but the conjugate gradient method [11,15] is especially attractive because of its optimal convergence properties [10], mainly due to the fact that some norm of the error is minimised at each iteration.

Moreover, the number of iterations i_ε required to reduce the error under a given value ε is bounded following

$$i_\varepsilon \leq \frac{1}{2} \sqrt{\kappa(\mathbf{B}^{-1}\mathbf{K})} \log \frac{2}{\varepsilon} + 1 \quad (23)$$

where \mathbf{B} is a preconditioning matrix and $\kappa(\mathbf{B}^{-1}\mathbf{K})$ is the condition number of the preconditioned system, i.e. the ratio of the largest to the least eigenvalue of $\mathbf{B}^{-1}\mathbf{K}$. The number of iterations is in practice much lower than suggested by the upper bound (23) when there are large ‘gaps’ in the spectrum of $\mathbf{B}^{-1}\mathbf{K}$ while the bound is sharp when the spectrum tends to be continuous. The clustering of the eigenvalues thus favours the so-called *super-convergence* of the conjugate gradient method.

3.4. Preconditioning and choosing the projector

Finite element matrices are, in general, not well-conditioned and good convergence will be obtained at step (d) only if an efficient preconditioning can be applied. High-quality preconditioners are available for unconstrained FE analyses [20]; the current section gives valuable results about the efficiency of these preconditioners in the constrained case. Since the quality of a preconditioner \mathbf{B} for a matrix \mathbf{K} depends essentially on the clustering of the eigenvalues of the preconditioned system matrix $\mathbf{B}^{-1}\mathbf{K}$, it is worth analysing how the extreme eigenvalues of $\mathbf{B}^{-1}\mathbf{K}_p$ can be bounded in terms of those of $\mathbf{B}^{-1}\mathbf{K}$.

It is shown in Appendix A that

$$\nu_{\max}(\mathbf{B}^{-1}\mathbf{K}_p) \leq \nu_{\max}(\mathbf{B}^{-1}\mathbf{K})(1 - \gamma^2)^{-1} \quad (24)$$

$$\nu_{\min}(\mathbf{B}^{-1}\mathbf{K}_p) \geq \nu_{\min}(\mathbf{B}^{-1}\mathbf{K}) \quad (25)$$

where γ is the lowest of both Cauchy–Buniakowskii–Schwarz constants γ_K and γ_B defined by

$$\gamma_K = \max_{\substack{u \in \mathcal{R}(\mathcal{P}) \\ v \in \mathcal{R}(\mathbf{I} - \mathcal{P})}} \frac{\langle u, \mathbf{K}v \rangle}{\sqrt{\langle u, \mathbf{K}u \rangle \langle v, \mathbf{K}v \rangle}}$$

$$\gamma_B = \max_{\substack{u \in \mathcal{R}(\mathcal{P}) \\ v \in \mathcal{R}(\mathbf{I} - \mathcal{P})}} \frac{\langle u, \mathbf{B}v \rangle}{\sqrt{\langle u, \mathbf{B}u \rangle \langle v, \mathbf{B}v \rangle}}$$

and where $\nu_{\max}(\cdot)$, $\nu_{\min}(\cdot)$ denotes the largest and the lowest *non-zero* eigenvalue (the modes corresponding to the zero eigenvalue are not ‘active’ when solving a consistent singular system by, e.g. the conjugate gradient method). In the expression of γ_K and γ_B , $\mathcal{R}(\cdot)$ is the *range* of a given matrix.

In the light of Ineq. (23), (24) and (25), an ‘ideal’ choice for the projector would be obtained by letting $\mathbf{I} = \mathbf{C}\mathbf{K}^{-1}$ or $\mathbf{I} = \mathbf{C}\mathbf{B}^{-1}$ which implies either γ_K or γ_B to be zero. These choices are however of no practical interest due to the large overhead that would be spent in steps (a) and (b). In our implementation of the method we have preferred to set $\mathbf{I} = \mathbf{C}$ for which no deterioration of the conditioning properties was observed compared to the unconstrained case, as emphasised by the numerical results of Section 4.

With the latter choice of \mathbf{I} and using the unpreconditioned conjugate gradient solution algorithm, our solution procedure becomes equivalent to the projected gradient method well known in the field of constrained optimisation [5,9,17]. Our approach is, however, more simple and somewhat more general because it clearly separates the projection features from the choice of the iterative procedure. Note also, that the starting point in algorithms used in [5,9,17] is a modification of an existing iterative algorithm in order to satisfy the constraints at each step, motivated by the need to also deal with linear inequality constraints. Such a modification requires to re-establish all the properties of the considered iterative algorithm, which is avoided here since our iterative scheme remains unchanged. Moreover, the mixing between projection and iterative solution method makes it unclear how preconditioning could be introduced without requiring again a careful analysis of the algorithm, whereas preconditioning is natural in our approach.

4. Numerical results

In all numerical experiments described in this section, the DRIC(1)-DC factorisation introduced in [21] is used as a preconditioner. The iterations are stopped when the error is reduced under 10^{-8} and the measure of the error chosen is the standard residual-based error, i.e. the ratio of the L_2 -norm of the residual of the preconditioned system and the L_2 -norm of the right-hand side.

4.1. Regular meshes

A first set of simple test problems are regular and uniform 2D meshes of plane stress bilinear finite elements. The bottom corners of the square structure are fixed, a vertical nodal load is applied at the centre of the top side. Linear equality constraints prescribe the average value of the displacements of a number of randomly chosen series of five nodes located on a same vertical or horizontal line. Increasing the number of series raises the number of constraints. Fig. 1 shows the effect of a growing number of constraints for a problem of fixed size with 5198 degrees of freedom. The number of iterations slightly increases with the number of constraints but this effect is negligible and the number of iterations remains smaller than that required for solving the unconstrained problem. The reason for this better convergence on the constrained system is investigated in the numerical experiments of Subsections 4.2 and 4.3. In Fig. 2, the number of constraints is kept small and the number of unknowns varies. The same comment as for Fig. 1 applies.

4.2. Cyclic periodicity constraints

More significant results are obtained on industrial benchmarks. The test problem represented in Fig. 3 is a fourth of turbine blade submitted to torsion. The base side is fully fixed, the loads are applied on the top side. Only a fourth of the structure is meshed to save CPU- and memory requirements during the FE static linear analysis, implying that some periodicity conditions are needed along the vertical cut edges. Here the cut edges are normal to the x - and y -global axes, so no local axes are needed and these conditions can be expressed as follows (if l and r denote the left and right cut edges respectively):

$$q_x(\text{node}_l) - q_y(\text{node}_r) = 0 \tag{26}$$

$$q_y(\text{node}_l) + q_x(\text{node}_r) = 0 \tag{27}$$

$$q_z(\text{node}_l) - q_z(\text{node}_r) = 0 \tag{28}$$

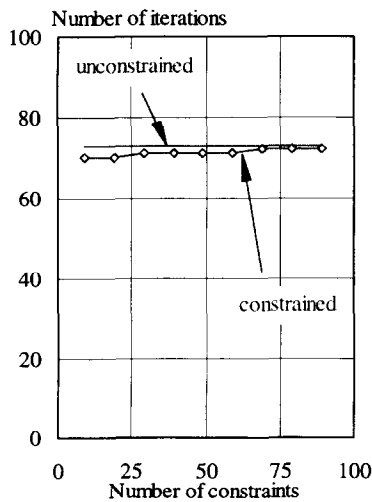


Fig. 1. Effect of the number of constraints on the convergence.

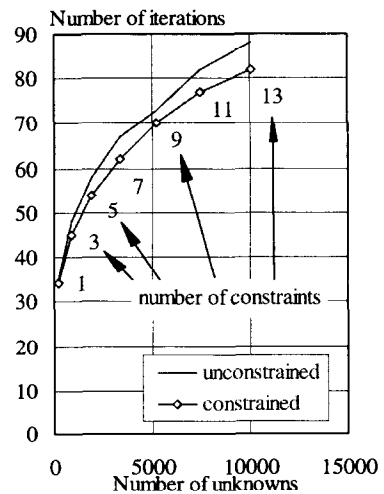


Fig. 2. Effect of the number of unknowns and constraints on the convergence.

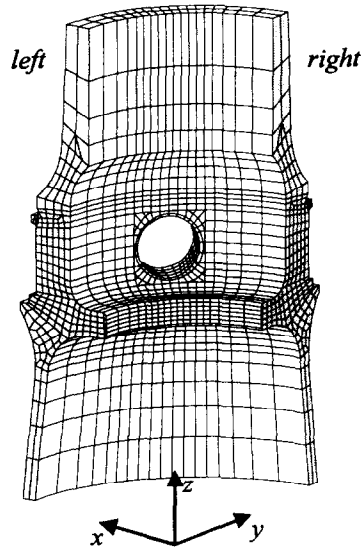


Fig. 3. The shaft.

In the considered test example, which counts 51 162 unknowns, there are 2043 such constraints.

These linear equality constraints can also be interpreted as assembly directives: unknowns $q_x(\text{node}_i)$ and $q_y(\text{node}_j)$ (are the same) which can be taken into account by giving them the same unknown number; the corresponding entries in \mathbf{K} are then summed up. On the contrary, the entries of \mathbf{K} related to unknowns $q_y(\text{node}_i)$ and $q_x(\text{node}_j)$ are subtracted. The efficiency of subspace projection is measured more precisely here by a comparison of the two available ways to take periodicity conditions into account.

The numerical results related to this benchmark (number of iterations and condition number of the preconditioned system) are reported in Table 1. It is readily seen that the number of iterations is larger when the cyclic periodicity of the structure is taken into account via linear constraints (LCE) instead of during the assembly (ASS) but this loss of efficiency (about 25% in terms of number of iterations) remains fairly reasonable. It has also to be noticed that the system is smaller (49 119 unknowns instead of 51 162) when periodicity conditions are forced through the assembly process.

The number of iterations and condition number are, however, much larger in the unconstrained case, which would tend to show that solving a constrained problem is less time-consuming than solving an unconstrained one. This has to be related to the well-known fact that the conditioning is better when the number of Dirichlet boundary conditions is increased since the constraints are boundary conditions on the displacements. This trend is illustrated in Table 1 where the line marked FIX shows the number of iterations and condition number when the displacements on the cut edges are fixed to zero; the best convergence is obtained in this case. This remark highlights the difficulty of comparing the performance of an iterative solver when the boundary conditions vary and comparisons of constrained/unconstrained structures are to be made with very special care.

The plot of Fig. 4 representing the spectrum of the preconditioned system confirms our theoretical result: Ineq. (24) and (25) are both satisfied. This figure highlights the better positioning of the smallest eigenvalues for

Table 1
Number of iterations and condition numbers for the shaft benchmark

Example	Number of iterations	Condition number
unconstrained	865	$1.89 \cdot 10^5$
ASS	319	$3.22 \cdot 10^3$
LCE	425	$6.20 \cdot 10^4$
FIX	218	$6.62 \cdot 10^2$

ASS: cyclic periodicity guaranteed via the assembly; LCE: linear constraints on the cut edges; FIX: Dirichlet boundary conditions on the cut edges.

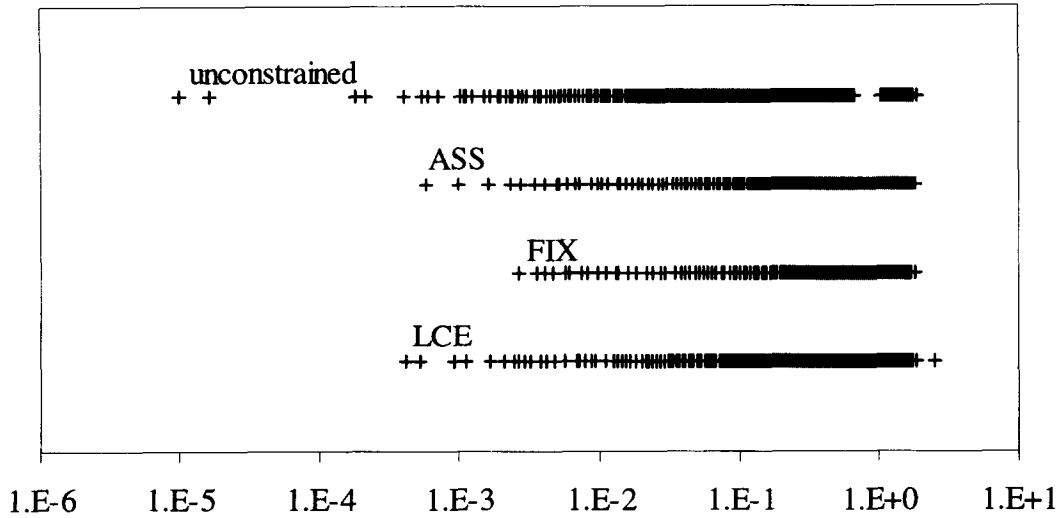


Fig. 4. Spectrum of the preconditioned system for the shaft benchmark.

the constrained problem than for the unconstrained one. On the other hand, the largest eigenvalues are not shifted by a too large amount, so the value of γ in (24) is reasonably small.

4.3. Rigid body parts

The last example depicted by Fig. 5 illustrates the footing of a concrete pile of bridge embedded in a soil composed of three layers with different material characteristics. All nonlinear effects are neglected in this simplified academic model; the contact and friction conditions between the pile and the soil are not taken into account (the displacements are continuous through any material discontinuities). The footing is introduced to smooth the transfer of the vertical stresses at the pile/soil interface.

For simplicity, the discretisation is regular and uniform through the whole model, with 17 280 tri-linear solid elements and 48 610 unknowns. The displacements of the vertical and bottom boundaries of the studied region of the soil are prescribed as zero. The loads, applied on all nodes at the top of the pile, are parallel to the (y, z)

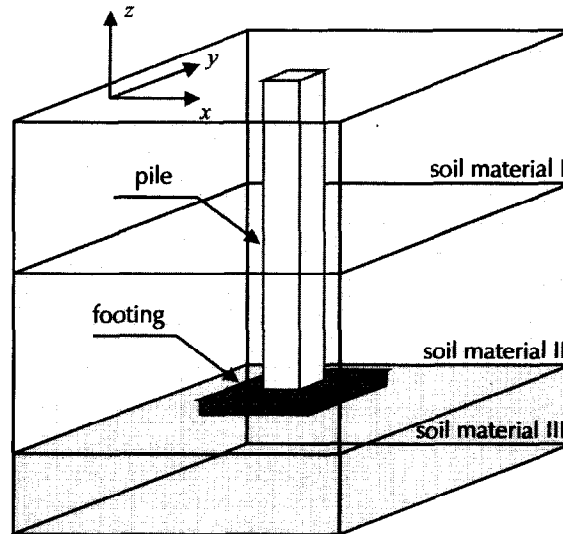


Fig. 5. The pile footing benchmark.

Table 2

Number of iterations and condition numbers for the pile footing benchmark (702 linear equality constraints)

Example	Number of iterations	Condition number
unconstrained	247	$4.33 \cdot 10^3$
LCE	156	$1.75 \cdot 10^4$

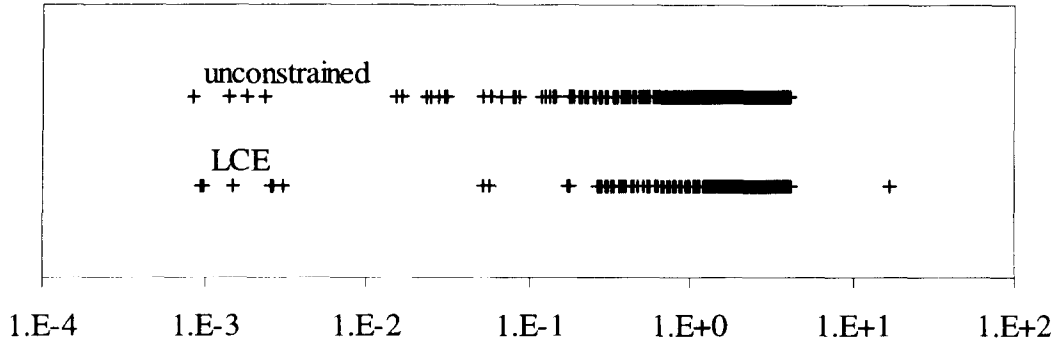


Fig. 6. Spectrum of the preconditioned system for the pile footing problem, in the constrained and unconstrained cases.

plane to study the bending of the pile. The footing is made of reinforced concrete, considered here to have an infinite stiffness to avoid the modelling of a such a composite material. This condition is ensured by prescribing a constant distance between the nodes in the footing: by choosing a *master* node m , the displacement of the remaining *slave* nodes s of this area satisfy

$$\mathbf{q}_s = \mathbf{q}_m + \boldsymbol{\omega}_m \times \mathbf{d}(m, s) \tag{29}$$

where

- \mathbf{q}_s and \mathbf{q}_m are the translational displacements of nodes s and m respectively;
- $\boldsymbol{\omega}_m$ is the rotational displacement of node m ;
- $\mathbf{d}(m, s)$ is a vector containing the difference of the co-ordinates of points s and m , pointing from m to s ;
- \times denotes the vector product: $(\mathbf{v} \times \mathbf{w})_k = \mathbf{v}_i \mathbf{w}_j \delta_{ijk}$ if δ_{ijk} is the Kronecker symbol.

Of course, rotational displacements are not present in a 3D discretisation with only solid finite elements. The existence of $\boldsymbol{\omega}_m$ is enforced by setting one beam element between the master node and each of its slave nodes. These beams and the linear constraints (29) are automatically generated in industrial FE softwares like SAMCEF [22]. Note that, contrary to periodicity conditions, the linear constraints considered here cannot be taken into account during the assembly.

The numerical results given in Table 2 show that the condition number is larger in the constrained case, due to our choice of \mathbf{F} as discussed in Section 3. Complementary information is given in Fig. 6 where the spectrum of the preconditioned system is represented. It appears that despite the less attractive width of the spectrum in the constrained case, only a few eigenvalues are located beyond the largest eigenvalue of the unconstrained problem and they are concentrated and isolated in a very restricted area. Moreover, the gaps between clusters of eigenvalues are larger which favours the convergence and explains the smaller number of iterations obtained in the constrained case.

5. Conclusions

Some methods used to solve FE problems including linear equality constraints have been briefly discussed and subspace projection has been shown to be the most efficient one. This latter case has been described and it has been highlighted that this technique is independent of the iterative algorithm chosen to solve the projected system. Only a few modifications of an existing solver are required, consisting mainly in the implementation of a procedure to compute the projection of any given vector.

The main contribution of this paper is a discussion of the preconditioning of the projected system. Theoretical

results relating the position of the extreme eigenvalues of the projected system to those of the unconstrained problem have been provided. Numerical experiments have confirmed the analysis of the bounds of the spectrum and constrained problems have been shown to be at least as well-conditioned as unconstrained ones. Relying on these numerical experiments, our proposed CPU and memory saving projector has been found very satisfactory, despite not being optimal with respect to the location of the upper eigenvalue.

Acknowledgments

This work has been funded by the IRSIA (Institut pour l’encouragement de la Recherche Scientifique dans l’Industrie et l’Agriculture), the FRIA (Fonds pour la formation à la Recherche dans l’Industrie et l’Agriculture) and the FNRS (Fonds National de la Recherche Scientifique). The authors are also most appreciative for the support of SAMTECH S.A. (Liège, Belgium) and wish to thank TECHSPACE AERO S.A. (Herstal, Belgium) for providing the benchmark of the turbine shaft.

Appendix A

Theorems 2 and 3 below demonstrate how Ineq. (25) and (24) have been obtained. Theorem 3 uses Lemma 1 as a pre-requisite for introducing the Cauchy–Buniakowskii–Schwarz constants. The specific notations used in this appendix are:

- $\mathcal{R}(\cdot)$ and $\mathcal{N}(\cdot)$ for the range and the null-space of a matrix respectively;
- $(\cdot)^\perp$ for a subspace orthogonal to a given subspace;
- $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{B}} = \langle \mathbf{u}, \mathbf{B}\mathbf{v} \rangle$ for the \mathbf{B} -inner product if \mathbf{B} is some symmetric positive definite matrix.

LEMMA 1. Given a linear projector \mathcal{P} , a symmetric positive definite matrix \mathbf{K} and a symmetric positive definite preconditioner \mathbf{B} . If $\gamma_{\mathbf{K}}$ and $\gamma_{\mathbf{B}}$ are defined by

$$\gamma_{\mathbf{K}} = \max_{\substack{\mathbf{u} \in \mathcal{R}(\mathcal{P}) \\ \mathbf{v} \in \mathcal{R}(\mathbf{I} - \mathcal{P})}} \frac{\langle \mathbf{u}, \mathbf{K}\mathbf{v} \rangle}{\sqrt{\langle \mathbf{u}, \mathbf{K}\mathbf{u} \rangle \langle \mathbf{v}, \mathbf{K}\mathbf{v} \rangle}}$$

$$\gamma_{\mathbf{B}} = \max_{\substack{\mathbf{u} \in \mathcal{R}(\mathcal{P}) \\ \mathbf{v} \in \mathcal{R}(\mathbf{I} - \mathcal{P})}} \frac{\langle \mathbf{u}, \mathbf{B}\mathbf{v} \rangle}{\sqrt{\langle \mathbf{u}, \mathbf{B}\mathbf{u} \rangle \langle \mathbf{v}, \mathbf{B}\mathbf{v} \rangle}}$$

one has

$$\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle \geq \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle (1 - \gamma_{\mathbf{B}}^2) \tag{A.1}$$

and

$$\langle \mathbf{q}, \mathbf{K}\mathbf{q} \rangle \geq \langle \mathcal{P}\mathbf{q}, \mathbf{K}\mathcal{P}\mathbf{q} \rangle (1 - \gamma_{\mathbf{K}}^2) \tag{A.2}$$

PROOF. Thanks to Eq. (A.5),

$$\begin{aligned} \langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle &= \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle + \langle (\mathbf{I} - \mathcal{P})\mathbf{q}, \mathbf{B}(\mathbf{I} - \mathcal{P})\mathbf{q} \rangle + 2\langle (\mathbf{I} - \mathcal{P})\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle \\ &\geq \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle + \langle (\mathbf{I} - \mathcal{P})\mathbf{q}, \mathbf{B}(\mathbf{I} - \mathcal{P})\mathbf{q} \rangle - 2\gamma_{\mathbf{B}} \sqrt{\langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle \langle (\mathbf{I} - \mathcal{P})\mathbf{q}, \mathbf{B}(\mathbf{I} - \mathcal{P})\mathbf{q} \rangle} \\ &\geq \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle (1 + \xi^2 - 2\xi\gamma_{\mathbf{B}}) \end{aligned}$$

saved

$$\xi^2 = \frac{\langle (\mathbf{I} - \mathcal{P})\mathbf{q}, \mathbf{B}(\mathbf{I} - \mathcal{P})\mathbf{q} \rangle}{\langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle}$$

Moreover,

$$\begin{aligned}
1 - \gamma_B^2 &= 1 + \xi^2 - 2\xi\gamma_B - \xi^2 + 2\xi\gamma_B - \gamma_B^2 \\
&= (1 + \xi^2 + 2\xi\gamma_B) - (\xi - \gamma_B)^2 \\
&\leq 1 + \xi^2 + 2\xi\gamma_B
\end{aligned}$$

which leads to Eq. (A.1). The same process applies to get Eq. (A.2) \square

THEOREM 2. Given a linear projector \mathcal{P} , a symmetric positive definite matrix \mathbf{K} and a symmetric positive definite preconditioner \mathbf{B} ,

$$\nu_{\min}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) \geq \nu_{\min}(\mathbf{B}^{-1}\mathbf{K}) \quad (\text{A.3})$$

PROOF. One starts with the definition of the least non-zero eigenvalue as established in [15],

$$\nu_{\min}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) = \min_{\mathbf{q} \in \mathcal{R}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P})} \frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle}$$

But

$$\begin{aligned}
\mathbf{q} \in \mathcal{R}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) &\Leftrightarrow \mathbf{B}\mathbf{q} \in \mathcal{R}(\mathcal{P}^t) \\
&\Leftrightarrow \mathbf{B}\mathbf{q} \in \mathcal{N}(\mathcal{P})^\perp \\
&\Leftrightarrow \forall \mathbf{z} \in \mathcal{N}(\mathcal{P}), \quad \langle \mathbf{z}, \mathbf{B}\mathbf{q} \rangle = 0
\end{aligned} \quad (\text{A.4})$$

Since

$$\mathbf{q} = \mathcal{P}\mathbf{q} + (\mathbf{I} - \mathcal{P})\mathbf{q} \quad (\text{A.5})$$

then

$$\mathcal{P}\mathbf{q} = \mathcal{P}\mathbf{q} + \mathcal{P}(\mathbf{I} - \mathcal{P})\mathbf{q}$$

in which the second term vanishes, so that in Eq. (A.5), $(\mathbf{I} - \mathcal{P})\mathbf{q}$ is the component of \mathbf{q} that belongs to $\mathcal{N}(\mathcal{P})$. Therefore,

$$\begin{aligned}
\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle &= \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathbf{q} \rangle + \langle (\mathbf{I} - \mathcal{P})\mathbf{q}, \mathbf{B}\mathbf{q} \rangle \\
&= \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathbf{q} \rangle \text{ due to (A.4)} \\
&= \langle \mathcal{P}\mathbf{q}, \mathbf{q} \rangle_{\mathbf{B}} \text{ because } \mathbf{B} \text{ is symmetric positive definite and then defines an inner product} \\
&\leq \sqrt{\langle \mathcal{P}\mathbf{q}, \mathcal{P}\mathbf{q} \rangle_{\mathbf{B}}} \sqrt{\langle \mathbf{q}, \mathbf{q} \rangle_{\mathbf{B}}}
\end{aligned}$$

which implies

$$\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle \leq \langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle$$

On the other hand,

$$\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle \leq \langle \mathcal{P}\mathbf{q}, \mathbf{K}\mathcal{P}\mathbf{q} \rangle$$

and

$$\min_{\mathbf{q} \in \mathcal{R}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P})} \frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle} \geq \min_{\mathbf{q} \in \mathcal{R}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P})} \frac{\langle \mathcal{P}\mathbf{q}, \mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathcal{P}\mathbf{q}, \mathbf{B}\mathcal{P}\mathbf{q} \rangle} \geq \min_{\mathbf{q} \in \mathcal{R}^n} \frac{\langle \mathbf{q}, \mathbf{K}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle}$$

which leads to Ineq. (A.3). \square

THEOREM 3. Given a linear projector \mathcal{P} , a symmetric positive definite matrix \mathbf{K} and a symmetric positive definite preconditioner \mathbf{B} ,

$$\nu_{\max}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) \leq \nu_{\max}(\mathbf{B}^{-1}\mathbf{K})(1 - \gamma^2)^{-1} \quad (\text{A.6})$$

Proof. As in the proof of Theorem 2,

$$\nu_{\max}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) = \max_{\mathbf{q} \in \mathcal{R}(\mathbf{B}^{-1}\mathcal{P}\mathbf{K}\mathcal{P})} \frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle}$$

One has

$$\frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle} = \frac{\langle \mathbf{q}, \mathbf{K}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle} \frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{K}\mathbf{q} \rangle}$$

The first term of this product is always lower or equal to $\nu_{\max}(\mathbf{B}^{-1}\mathbf{K})$ and Eq. (A.2) provides a bound for the second term, so that

$$\nu_{\max}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) \leq \nu_{\max}(\mathbf{B}^{-1}\mathbf{K})(1 - \gamma_K^2)^{-1} \tag{A.7}$$

Moreover,

$$\frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle} = \frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{K}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathcal{P}^t\mathbf{B}\mathcal{P}\mathbf{q} \rangle} \frac{\langle \mathbf{q}, \mathcal{P}^t\mathbf{B}\mathcal{P}\mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{B}\mathbf{q} \rangle}$$

The maximum value of the first term of the product is always lower or equal to $\nu_{\max}(\mathbf{B}^{-1}\mathbf{K})$ and Eq. (A.1) bounds upperly the second term to yield

$$\nu_{\max}(\mathbf{B}^{-1}\mathcal{P}^t\mathbf{K}\mathcal{P}) \leq \nu_{\max}(\mathbf{B}^{-1}\mathbf{K})(1 - \gamma_B^2)^{-1} \tag{A.8}$$

and if

$$\gamma = \min(\gamma_K, \gamma_B)$$

both Eqs. (A.7) and (A.8) combine in Ineq. (A.6). \square

References

- [1] K. Arrow, L. Hurwicz and H. Uzawa, *Studies in Nonlinear Programming* (Stanford University Press, 1958).
- [2] S.F. Ashby, T.A. Manteuffel and P.E. Saylor, Adaptive polynomial preconditioning for Hermitian indefinite linear systems, *BIT* 29 (1989) 583–609.
- [3] O. Axelsson, *Numerical Algorithms for Indefinite Problems, Elliptic Solvers II* (Academic Press, 1984) 219–232.
- [4] BENCHMARK, *Iterative FEA Solvers, Benchmark (NAFEMS)* (March 1994), 14–15.
- [5] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, *Comput. J.* 7 (1964) 149–154.
- [6] R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems, Lecture Notes in Mathematics, Vol. 506* (Springer Verlag, 1976) 73–89.
- [7] R.W. Freund and N.M. Nachtigal, QMR: A quasi-minimal residual method for non-hermitian linear systems, *Numer. Math.* 60 (1991) 315–339.
- [8] K. Georgiev, A. Baltov and S. Margenov, *Some Practical Requirements to HIPERGEOS Software Applications in the Bridge Engineering*, Technical report, MOST XXI Ltd, Sofia, Bulgaria, 1995.
- [9] D. Goldfarb, Extension of Davidon’s variable metric method to maximization under linear inequality and equality constraints, *SIAM J. Appl. Math.* 17 (1969) 739–764.
- [10] M.R. Hestenes and E. Stiefel, Methods of conjugate gradient for solving linear systems, *J. Res. Nat. Bur. Standard Sect. B49* (1952) 409–436.
- [11] E.F. Kaasschieter, Preconditioned conjugate gradients for solving singular systems, *J. Comput. Appl. Math.* 24 (1988) 265–275.
- [12] L. Komzsik and P. Poschmann, *Iterative Solution Techniques for Finite Element Applications, Finite Elements in Analysis and Design, Vol. 14* (Elsevier, Amsterdam, London, NY, Tokyo 1993) 373–379.
- [13] U. Langer and W. Queck, *Preconditioned Uzawa-type Iterative Methods for Solving Mixed Finite Element Equations* (Wissenschaftliche Schriftenreihe der Technischen Universität Karl-Marx-Stadt, 1987).
- [14] MARC Analysis Research Corporation, *On the MARC, The MARC newsletter*, December 1993.
- [15] Y. Notay, Polynomial acceleration of iterative schemes associated with subproper splittings, *J. Comput. Appl. Math.* 24 (1988) 153–167.
- [16] C.C. Paige and M.A. Saunders, Solution of sparse indefinite systems of linear equations, *SIAM J. Numer. Anal.* 12 (1975) 617–629.
- [17] M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Comput. J.* 7 (1964) 155–162.
- [18] Y. Saad and M.H. Schultz, GMRES—A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (1986) 856–869.

- [19] P. Saint-Georges, Iterative solution of linear systems for FEM structural analysis, Technical Report SMC95/05 (Service des Milieux Continus, Université Libre de Bruxelles) intermediate IRSIA report 3, 1995.
- [20] P. Saint-Georges, G. Warzée, Y. Notay and R. Beauwens, Fast iterative solvers for FE analysis in general and shell analysis in particular, Proc. 3rd Int. Conf. on Computational Structures Technology, Budapest, 21–23 August 1996.
- [21] P. Saint-Georges, G. Warzée, Y. Notay and R. Beauwens, High-performance PCG solver for FEM structural analyses, *Int. J. Numer. Methods Engrg.* 39 (1996) 1313–1340.
- [22] SAMCEF V6.0, Système d'Analyse de Milieux Continus par Eléments Finis, Mode d'Emploi (SAMTECH, Liège, 1994).
- [23] H.A. Van Der Vorst, Bi-CGSTAB : A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.* 13 (1992) 631–644.
- [24] M.B. Van Gijzen, Iterative Linear Equation Solvers, Benchmark (NAFEMS) (July 1995) 24–25.
- [25] T.B. Wertheimer, MARC K5 iterative solution procedures, Benchmark (NAFEMS) (June 1994) 8–10.