

HIGH-PERFORMANCE PCG SOLVERS FOR FEM STRUCTURAL ANALYSIS

P. SAINT-GEORGES* AND G. WARZEE

Service des Milieux Continus, Université Libre de Bruxelles, 50 av. F.-D. Roosevelt, 1050 Bruxelles, Belgium

R. BEAUWENS AND Y. NOTAY†

Service de Métrologie Nucléaire, Université Libre de Bruxelles, 50 av. F.-D. Roosevelt, 1050 Bruxelles, Belgium

SUMMARY

The preconditioned conjugate gradient algorithm is a well-known and powerful method used to solve large sparse symmetric positive definite linear systems. Such systems are generated by the finite element discretization in structural analysis but users of finite elements in this context generally still rely on direct methods. It is our purpose in the present work to highlight the improvement brought forward by some new preconditioning techniques and show that the preconditioned conjugate gradient method performs better than efficient direct methods.

KEY WORDS: iterative methods for linear systems; preconditioning

1. INTRODUCTION

It seems generally accepted that Preconditioned Conjugate Gradient (PCG) solvers cannot compete with direct methods for solving realistic industrial problems arising from finite element (FE) discretizations in structural analysis. A recent comparison by Poole *et al.*¹ confirmed this view, leaving their low memory requirements as the only advantage that still should be granted to iterative solvers. As a matter of fact, most commercially available industrial codes use Iron's frontal method or a skyline LU factorization as solver. Our purpose here is to demonstrate that taking care of recent advances in the design of PCG algorithms reverses this conclusion at least for large problems.

Highly effective preconditioners have recently been obtained by dynamically controlled approximate factorizations of Stieltjes matrices, as described in Section 2. The field of application of these factorizations has further been extended to more general symmetric positive definite matrices by means of spectral equivalence techniques. The application of this extension to FE structural analysis is discussed in Section 3. Section 4 presents performance comparisons between our PCG algorithm and an efficient direct method, showing amongst others that the robustness of PCG does not deteriorate for problems offering discontinuities or high values of the Poisson ratio.

All the results in this paper are restricted to elastostatics leading after FE discretization to a system of linear equations of the form

$$\mathbf{Kq} = \mathbf{f} \quad (1)$$

* Supported by the IRSIA (Institut pour l'Encouragement de la Recherche Scientifique dans l'Industrie et l'Agriculture)

† Supported by the FNRS (Fonds National de la Recherche Scientifique)

where \mathbf{K} is the structural stiffness matrix, \mathbf{f} is the nodal load vector and \mathbf{q} is the nodal displacement vector.

2. PGG SOLVERS FOR STIELTJES MATRICES

PCG methods for solving (1) have now reached an elaborated stage of development and can almost be used in a black-box fashion when \mathbf{K} is a Stieltjes matrix.²⁻⁴ When \mathbf{K} is not a Stieltjes matrix, an additional so-called *reduction* step has to be introduced, by which \mathbf{K} is first preconditioned by an approximate Stieltjes matrix \mathbf{S} . The techniques for doing this have also seen recent developments, to be described in the next section.

Definition 1. \mathbf{M} is an M-matrix if \mathbf{M} is non-singular with non-positive off-diagonal entries and has a non-negative inverse.

Definition 2. \mathbf{S} is a Stieltjes matrix if \mathbf{S} is a symmetric M-matrix or, alternatively, positive definite with non-positive off-diagonal entries.

The PCG method for solving (1) is shown in Table I, where \mathbf{B} denotes the preconditioning matrix. At each iteration step, a linear system

$$\mathbf{B}\mathbf{h}^{k+1} = \mathbf{g}^{k+1} \quad (2)$$

has to be solved. Therefore, solving (2) must be cheap.

The number of iterations i_ε required to reduce the \mathbf{K} -norm of the initial error by a factor ε is bounded⁵ by

$$i_\varepsilon \leq \frac{1}{2} \sqrt{\kappa(\mathbf{B}^{-1}\mathbf{K})} \ln \frac{2}{\varepsilon} + 1 \quad (3)$$

where $\kappa(\mathbf{B}^{-1}\mathbf{K})$ denotes the ratio of the extreme eigenvalues of $\mathbf{B}^{-1}\mathbf{K}$

$$\kappa(\mathbf{B}^{-1}\mathbf{K}) = \frac{\lambda_N(\mathbf{B}^{-1}\mathbf{K})}{\lambda_1(\mathbf{B}^{-1}\mathbf{K})} \quad (4)$$

Table I. The PCG scheme

INITIALIZATION

$$\begin{aligned} \mathbf{q}^0 &= \mathbf{q}^{\text{init}} \\ \mathbf{g}^0 &= \mathbf{K}\mathbf{q}^0 - \mathbf{f} \\ \mathbf{d}^0 &= -\mathbf{g}^0 \\ \mathbf{h}^0 &= \mathbf{g}^0 \end{aligned}$$

ITERATION

$$\begin{aligned} \tau_k &= \frac{\mathbf{g}^{k\text{T}}\mathbf{h}^k}{\mathbf{d}^{k\text{T}}\mathbf{K}\mathbf{d}^k} \\ \mathbf{q}^{k+1} &= \mathbf{q}^k + \tau_k\mathbf{d}^k \\ \mathbf{g}^{k+1} &= \mathbf{g}^k + \tau_k\mathbf{K}\mathbf{d}^k \\ \mathbf{h}^{k+1} &= \mathbf{B}^{-1}\mathbf{g}^{k+1} \\ \beta_k &= \frac{\mathbf{g}^{k+1\text{T}}\mathbf{h}^{k+1}}{\mathbf{g}^{k\text{T}}\mathbf{h}^k} \\ \mathbf{d}^{k+1} &= -\mathbf{h}^{k+1} + \beta_k\mathbf{d}^k \end{aligned}$$

Table II. The IC scheme

INITIALIZATION	$\mathbf{U} = \text{offdiag}(\mathbf{K})$
	$\mathbf{P} = \text{diag}(\mathbf{K})$
ITERATION	
	For $r = 1, \dots, N - 1$
	For $i = r + 1, \dots, N$ such as $(r, i) \in \text{NZP}$
	temp = $\mathbf{u}_{ri}/\mathbf{p}_r$
	$\mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{ri}$
	For $j = i + 1, \dots, N$ such as $(r, j) \in \text{NZP}$
	If $(i, j) \in \text{FP}$ then $\mathbf{u}_{ij} = \mathbf{u}_{ij} - \text{temp } \mathbf{u}_{rj}$

called the spectral condition number of $\mathbf{B}^{-1}\mathbf{K}$ (a result sometimes called Meinardus bound as it refers to Reference 5). Therefore, \mathbf{B} should be a good spectral approximation of \mathbf{K} , meaning hereby that $\kappa(\mathbf{B}^{-1}\mathbf{K})$ should be as small as possible.

Both conditions put on \mathbf{B} (unexpensive resolution of equation (2) and small $\kappa(\mathbf{B}^{-1}\mathbf{K})$) can be met by using for \mathbf{B} an approximate factorization of \mathbf{K} but, as will be seen below, considerable care has to be taken in the choice of this approximate factorization.

To begin with, one may first try to use incomplete factorizations of \mathbf{K} , often called (somewhat abusively) *incomplete Cholesky* (IC) factorizations and derived from the exact factorization by ignoring updates of all entries (of the approximate factors) that do not belong to some given *fill-in pattern*.

This scheme is described in Table II, where

$$\mathbf{B} = \mathbf{U}^T \mathbf{P}^{-1} \mathbf{U} \quad (5)$$

\mathbf{U} being upper triangular and \mathbf{P} diagonal with $\mathbf{P} = \text{diag}(\mathbf{U})$. In this table, the differences between IC and the exact factorization are put in italic mode; the fill-in pattern is written FP and NZP denotes the nonzero pattern of $\mathbf{U} + \mathbf{U}^T$, that is the union of FP and the nonzero pattern of \mathbf{K} . An IC method is of low order when FP is small; a more precise definition of order will be given below.

IC methods were introduced hoping that the quality of the preconditioning would rapidly increase with FP. It was later proved⁶ that it never decreases when FP increases in the case of M-matrices (then in particular for Stieltjes matrices) on the basis of Woznicki's comparison theorem.⁷ But it was very early observed by Price and Varga⁸ that, on the contrary, a quite considerable increase of the fill-in leads to only a moderate improvement of the spectral condition number.

A first issue of the Price-Varga analysis is therefore that only low-order methods are of practical interest. In the present work, we shall only consider two fill-in levels:

1. order 0 $\equiv \text{FP}(0) = \{(i, j) \text{ with } i = j\}$;
2. order 1 $\equiv \text{FP}(1) = E(\mathbf{K})$,

where $E(\mathbf{K})$ denotes the edge set of the graph of the matrix \mathbf{K} , i.e. the set of couples (i, j) such that $\mathbf{k}_{ij} \neq 0$.

Higher orders of fill-in may be defined recursively by

$$FP(p + 1) = FP(p) \cup E(\mathbf{B}(p) - \mathbf{K}) \quad (6)$$

where $\mathbf{B}(p)$ denotes the IC factorization determined by $FP(p)$. Fractional orders may be used for intermediate levels. This definition differs only slightly from the Price–Varga convention^{6,8} and it is closer (although not identical) to the more usual ones.

It has to be mentioned that another issue of the Price–Varga analysis is that the IC factorization leads asymptotically to iteration numbers similar to those of the Jacobi preconditioning, with a better leading constant, but insufficient to represent a decisive improvement for large matrices.

Such an improvement came essentially from the works of Axelsson⁹ and Gustafsson¹⁰ who succeeded in analyzing a modification of the IC method, originally introduced by Buleev¹¹ and by which the diagonal entries of the triangular factors \mathbf{U} are modified such as to satisfy a row-sum criterion which may be written

$$\mathbf{B}\mathbf{x} = \mathbf{K}\mathbf{x} + \mathbf{A}\mathbf{D}\mathbf{x} \quad (7)$$

where \mathbf{x} is a positive vector such that $\mathbf{K}\mathbf{x} \geq \mathbf{0}$, $\mathbf{D} = \text{diag}(\mathbf{K})$ and $\mathbf{A} = (\lambda_i)$ is a non-negative diagonal matrix, called *perturbation* matrix. When \mathbf{K} is diagonally dominant, one may choose $\mathbf{x} = \mathbf{1}$ (the vector where all the components are unity) and this will be our choice in the following without further comment.

A heuristic way to justify this modification consists in regarding it as attempting to take care of the fill-in neglected in the IC scheme by moving it to the diagonal (so that the row-sum on each line has the same value as it would have for the exact factorization).

This heuristic explanation neglects the perturbation matrix \mathbf{A} which plays an essential role in the Axelsson–Gustafsson analysis but it turned out to be widely accepted and the corresponding method is described in Table III under the name of MIC method according to this widespread usage. We must insist, however, that it is in complete disagreement with the Axelsson–Gustafsson analysis (whose bound on $\kappa(\mathbf{B}^{-1}\mathbf{K})$ becomes infinite when $\mathbf{A} = \mathbf{0}$) and even that it neglects the main progress of their contributions, which was precisely to show the extreme sensitivity of (their bound on) the spectral condition number of the MIC preconditioning to the size and location of the perturbations λ_i .

Table III. The MIC scheme

INITIALIZATION

$$\mathbf{U} = \text{offdiag}(\mathbf{K})$$

$$\mathbf{P} = \text{diag}(\mathbf{K})$$

ITERATION

For $r = 1, \dots, N - 1$

For $i = r + 1, \dots, N$ such as $(r, i) \in \text{NZP}$

$$\text{temp} = \mathbf{u}_{r,i}/\mathbf{p}_r$$

$$\mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{r,i}$$

For $j = i + 1, \dots, N$ such as $(r, j) \in \text{NZP}$

$$\text{If } (i, j) \in \text{FP then } \mathbf{u}_{i,j} = \mathbf{u}_{i,j} - \text{temp } \mathbf{u}_{r,j}$$

$$\text{else } \mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{r,j}$$

$$\mathbf{p}_j = \mathbf{p}_j - \text{temp } \mathbf{u}_{r,j}$$

Table IV. The DMIC scheme

INITIALIZATION

$$\mathbf{U} = \text{offdiag}(\mathbf{K})$$

$$\mathbf{P} = \text{diag}(\mathbf{K})$$

ITERATION

For $r = 1, \dots, N - 1$

$$\tau_0 = -\frac{1}{\mathbf{p}_r} \sum_{i>r} \mathbf{u}_{ri}$$

If $\tau_0 > \tau$ then $\mathbf{p}_r = -\frac{1}{\tau} \sum_{i>r} \mathbf{u}_{ri}$

For $i = r + 1, \dots, N$ such as $(r, i) \in \text{NZP}$

$$\text{temp} = \mathbf{u}_{ri}/\mathbf{p}_r$$

$$\mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{ri}$$

For $j = i + 1, \dots, N$ such as $(r, j) \in \text{NZP}$

If $(i, j) \in \text{FP}$ then $\mathbf{u}_{ij} = \mathbf{u}_{ij} - \text{temp } \mathbf{u}_{rj}$

else $\mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{rj}$

$$\mathbf{p}_j = \mathbf{p}_j - \text{temp } \mathbf{u}_{rj}$$

Table V. The RIC scheme

INITIALIZATION

$$\mathbf{U} = \text{offdiag}(\mathbf{K})$$

$$\mathbf{P} = \text{diag}(\mathbf{K})$$

ITERATION

For $r = 1, \dots, N - 1$

For $i = r + 1, \dots, N$ such as $(r, i) \in \text{NZP}$

$$\text{temp} = \mathbf{u}_{ri}/\mathbf{p}_r$$

$$\mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{ri}$$

For $j = i + 1, \dots, N$ such as $(r, j) \in \text{NZP}$

If $(i, j) \in \text{FP}$ then $\mathbf{u}_{ij} = \mathbf{u}_{ij} - \text{temp } \mathbf{u}_{rj}$

else $\mathbf{p}_i = \mathbf{p}_i - \omega \text{temp } \mathbf{u}_{rj}$

$$\mathbf{p}_j = \mathbf{p}_j - \omega \text{temp } \mathbf{u}_{rj}$$

The best way to introduce these *modulated* perturbations became an active research area, leading to dynamic factorizations where a check is made at each stage of the factorization process, the issue of which determines the size of the corresponding perturbation.²⁻⁴

We shall consider here two dynamic factorization schemes, the DMIC scheme described in Table IV, which is a dynamic version of Gustafsson's statically perturbed MIC method^{10,12} (which must be distinguished from the unperturbed MIC scheme of Table III) and the DRIC method described in Table VI, which is a dynamic method derived from mixing the RIC and DMIC methods. The RIC method, due to Axelsson and Lindskog²⁶ is shown in Table V.

Table VI. The DRIC scheme

INITIALIZATION	$\mathbf{U} = \text{offdiag}(\mathbf{K})$ $\mathbf{P} = \text{diag}(\mathbf{K})$
ITERATION	
	For $r = 1, \dots, N - 1$
	$\tau_0 = -\frac{1}{\mathbf{p}_{r i > r}} \sum \mathbf{u}_{r i}$
	If $\tau_0 > \tau$ then $\omega = 2\tau/\tau_0 - 1$ else $\omega = 1$
	For $i = r + 1, \dots, N$ such as $(r, i) \in \text{NZP}$
	temp = $\mathbf{u}_{r i} / \mathbf{p}_r$
	$\mathbf{p}_i = \mathbf{p}_i - \text{temp } \mathbf{u}_{r i}$
	For $j = i + 1, \dots, N$ such as $(r, j) \in \text{NZP}$
	If $(i, j) \in \text{FP}$ then $\mathbf{u}_{i j} = \mathbf{u}_{i j} - \text{temp } \mathbf{u}_{r j}$
	else $\mathbf{p}_i = \mathbf{p}_i - \omega \text{temp } \mathbf{u}_{r j}$
	$\mathbf{p}_j = \mathbf{p}_j - \omega \text{temp } \mathbf{u}_{r j}$

These dynamic methods use an *a priori* given parameter τ and an upper spectral bound of $\mathbf{B}^{-1}\mathbf{K}$ to enforce the condition $\lambda_N(\mathbf{B}^{-1}\mathbf{K}) < (1 - \tau)^{-1}$ by inserting diagonal perturbations when (and only when) necessary. Both ends of the spectrum actually decrease with increased perturbations but the upper end is generally far more sensitive; as it is not feasible to control both, the best compromise is to control the upper end.

Numerical comparisons will be reported and analyzed in Section 4, where the parameter τ is defined by $1 - \tau = h_0$, where h_0 is a dimensionless measure of the mesh size, taken as

$$h_0 = \frac{hS}{4V} \approx \frac{1}{\sqrt[d]{\text{number of nodes}}} \quad (8)$$

where h is the average length of the sides of the elements and $4V/S$ is the so-called average chord length of the volume V (where the solution is sought) with boundary area S , for a problem with d spatial dimensions. In the RIC method, the parameter ω is similarly determined by letting $1 - \omega = h_0$.

The motivation for introducing RIC and DRIC methods in addition to DMIC arose from robustness considerations, particularly in presence of anisotropic discontinuities, where the DMIC scheme inserts too large perturbations. An alternate possibility, not covered here, was recently proposed by Magolu¹³ who introduced (in DMIC) a «dropping test» which simply cancels the perturbations in highly anisotropic regions.

The XIC notation used in the following refers to any of the IC, MIC, DMIC, RIC or DRIC factorizations.

3. REDUCTION TECHNIQUES

3.1. A two-step preconditioning approach

Stiffness matrices generated by FEM structural analyses are usually not Stieltjes matrices and non-positive pivots may then appear during the computation of the approximate factors.^{3,4,14,15} XIC preconditioners cannot anymore be considered as reliable. A simple remedy consists in adding appropriate positive pivots to the pivots encountered during the approximate factorization¹⁶ but there is no evidence that the resulting PCG method will still be efficient. A more reliable procedure is based on a two-step preconditioning approach:

1. *Reduction step*: Determine a Stieltjes matrix \mathbf{S} from the stiffness matrix \mathbf{K} ;
2. *Approximate factorization*: Determine an approximate factorization \mathbf{B} of \mathbf{S} by applying one of the XIC schemes described in the previous section to \mathbf{S} .

Since

$$\mathbf{B}^{-1}\mathbf{K} = \mathbf{B}^{-1}\mathbf{S}\mathbf{S}^{-1}\mathbf{K} \quad (9)$$

it is readily seen that

$$\kappa(\mathbf{B}^{-1}\mathbf{K}) \leq \kappa(\mathbf{S}^{-1}\mathbf{K})\kappa(\mathbf{B}^{-1}\mathbf{S}) \quad (10)$$

showing that the penalty incurred by the additional reduction step is bounded by $\kappa(\mathbf{S}^{-1}\mathbf{K})$ and thus independent of the parameters characterizing the problem whenever \mathbf{S} is spectrally equivalent to \mathbf{K} with respect to these parameters.

We recall here that two families of matrices $\mathbf{S}(p)$ and $\mathbf{K}(p)$ depending on some parameter p are called spectrally equivalent with respect to p when there exist two positive constants α, β independent of p such that, for any $\mathbf{x} \neq \mathbf{0}$,

$$\alpha \leq \frac{\mathbf{x}^T\mathbf{K}(p)\mathbf{x}}{\mathbf{x}^T\mathbf{S}(p)\mathbf{x}} \leq \beta \quad (11)$$

$\kappa(\mathbf{S}^{-1}\mathbf{K})$ is then bounded by β/α called the spectral equivalence bound. Our purpose in the present section is to show that \mathbf{S} can generally be chosen spectrally equivalent to \mathbf{K} with respect to the number NEL of finite elements as well as with respect to the size h of the finite elements.

3.2. Reduction schemes producing Stieltjes matrices from FE stiffness matrices

Beauwens and Wilmet¹⁵ are probably the first to use diagonal compensation (without giving it a name; it will be called here C-reduction) as reduction technique. Axelsson^{14,17} analyzed later the spectral properties of this method which consists in splitting a matrix \mathbf{A} to be reduced into two parts

$$\mathbf{A} = \underline{\mathbf{A}} - \bar{\mathbf{A}} \quad (12)$$

such that the off-diagonal part of $\underline{\mathbf{A}}$ contains only the negative entries of the off-diagonal part of \mathbf{A} :

$$\text{offdiag}(\underline{\mathbf{A}}) = \min(\text{offdiag}(\mathbf{A}), \mathbf{0}) \quad (13)$$

$$\text{offdiag}(\bar{\mathbf{A}}) = -\max(\text{offdiag}(\mathbf{A}), \mathbf{0}) \quad (14)$$

The diagonal part of $\underline{\mathbf{A}}$ is then computed by

$$\underline{\mathbf{A}}\mathbf{1} = \mathbf{A}\mathbf{1} \quad (15)$$

where $\mathbf{1} = \{1 \ 1 \ \dots \ 1\}^T$ is chosen here for convenience. The reduced matrix is obtained by taking: $\mathbf{A} = \mathbf{K}$ and $\mathbf{S} = \underline{\mathbf{A}}$.

Axelsson and Gustafsson¹⁸ developed also another reduction technique for 2-D membrane analyses, later extended to the study of 3-D solid structures by Shlafman and Efrat.¹⁹ First, the stiffness matrix is partitioned by grouping all the *degrees of freedom* (d.o.f.s) of the same type. The matrix below illustrates the case of a 3-D solid structure where each node has three associated d.o.f.s in the x , y and z directions, respectively;

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} & \mathbf{K}_{xz} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} & \mathbf{K}_{yz} \\ \mathbf{K}_{zx} & \mathbf{K}_{zy} & \mathbf{K}_{zz} \end{bmatrix} \quad (16)$$

A reduced matrix \mathbf{K}^D is then obtained by decoupling the do.f.s, that is, \mathbf{K}^D ignores any connection between d.o.f.s of different types. That decoupling will be called D-reduction.

$$\mathbf{K}^D = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yy} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{zz} \end{bmatrix} \quad (17)$$

Axelsson and Gustafsson¹⁸ and Shalfman and Efrat¹⁹ did not consider in their theoretical studies the case of matrices \mathbf{K}^D to which an approximate factorization cannot still apply. However, in most of the FEM structural analyses, \mathbf{K}^D is not a Stieltjes matrix. For this reason, the D-reduction is incomplete and must be followed by another reduction process: in our experiments, we used the C-reduction defined by equations (12)–(15) with $\mathbf{A} = \mathbf{K}^D$, $\underline{\mathbf{A}} = \underline{\mathbf{K}}^D$ and $\bar{\mathbf{A}} = \bar{\mathbf{K}}^D$.

We will refer in the following to that combination of reduction schemes as the DC-reduction. The reduced matrix is obtained by taking $\mathbf{S} = \underline{\mathbf{K}}^D$.

3.3. Spectral equivalence of \mathbf{K} and \mathbf{S} for the DC-reduction

Our purpose now is to prove the spectral equivalence of \mathbf{K} and $\mathbf{S} = \underline{\mathbf{K}}^D$ with respect to the number NEL and the size h of the finite elements. The whole set of matrices used in that proof are presented with their links in Figure 1. Matrix \mathbf{K} is formed by assembling the elementary stiffness matrices \mathbf{K}^e . The decoupling applies to these matrices as well as to the global matrix, generating elementary \mathbf{K}^{eD} matrices. The assembly of \mathbf{K}^{eD} restores the global \mathbf{K}^D matrix. The C-reduction may be applied to \mathbf{K}^D to form $\underline{\mathbf{K}}^D$ but also to each \mathbf{K}^{eD} , yielding elementary reduced matrices $\underline{\mathbf{K}}^{eD}$.

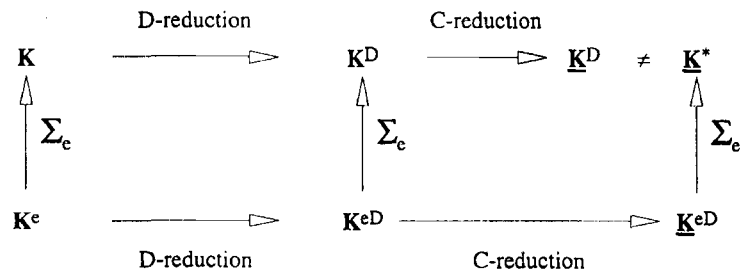


Figure 1. The reduction process and related matrices

The assembly of $\underline{\mathbf{K}}^{eD}$ brings forward a global matrix $\underline{\mathbf{K}}^*$; it can easily be seen that in general $\underline{\mathbf{K}}^* \neq \underline{\mathbf{K}}^D$.

The property of spectral equivalence with the DC-reduction is derived from the five theorems below.

Theorem 1. When C-reduction is applied to any symmetric positive definite matrix \mathbf{A} (resp. non-negative definite \mathbf{A}^e), the matrices $\underline{\mathbf{A}}$ and $\bar{\mathbf{A}}$ (resp. $\underline{\mathbf{A}}^e$ and $\bar{\mathbf{A}}^e$) are respectively, symmetric positive definite and non-negative definite (resp. both symmetric non-negative definite).

Proof. Symmetry is obvious. $\bar{\mathbf{A}}$ (resp. $\bar{\mathbf{A}}^e$) is non-negative definite from the Gershgorin theorem because $\bar{\mathbf{A}}\mathbf{1} = \mathbf{0}$ (resp. $\bar{\mathbf{A}}^e\mathbf{1} = \mathbf{0}$) and all off-diagonal entries of $\bar{\mathbf{A}}$ (resp. $\bar{\mathbf{A}}^e$) are non-positive. $\underline{\mathbf{A}}$ is positive (resp. $\underline{\mathbf{A}}^e$ is non-negative) definite because $\underline{\mathbf{A}} = \mathbf{A} + \bar{\mathbf{A}}$ (resp. $\underline{\mathbf{A}}^e = \mathbf{A}^e + \bar{\mathbf{A}}^e$). \square

Theorem 2. For a matrix $\mathbf{A} = \sum_e \mathbf{A}^e$ such that $\forall e$

$$\ker(\mathbf{A}^e) = \ker(\underline{\mathbf{A}}^e), \quad (18)$$

the matrices $\underline{\mathbf{A}}$ and \mathbf{A} are spectrally equivalent with respect to the number of finite element NEL, i.e. there exist $\alpha_A, \beta_A > 0$, independent of NEL such that for all $\mathbf{x} \neq \mathbf{0}$,

$$\alpha_A \mathbf{x}^T \underline{\mathbf{A}} \mathbf{x} \leq \mathbf{x}^T \mathbf{A} \mathbf{x} \leq \beta_A \mathbf{x}^T \underline{\mathbf{A}} \mathbf{x} \quad (19)$$

Proof. See appendix I. \square

Assumption (18) of Theorem 2 is always fulfilled when $\mathbf{A} = \mathbf{K}^D$, as shown in Appendix II. However, it is not the case when $\mathbf{A} = \mathbf{K}$: the in-plane rotation is a rigid body mode of several elementary stiffness matrices for 2-D membrane elements, for instance, and this vector does not belong to the kernel of the corresponding $\underline{\mathbf{A}}^e$. The C-reduction alone does not fulfill the spectral convergence properties (with our analysis), contrary to the DC-reduction. This is probably the origin of the better results obtained with the latter method in the numerical results of Section 4.

Therefore, Theorems 3 and 4 are introduced below.

Theorem 3. When D-reduction is applied to any symmetric positive definite matrix \mathbf{K} arising from a finite element discretization, the matrix \mathbf{K}^D is symmetric positive definite.

Proof. Straightforward since each block of \mathbf{K}^D is the stiffness matrix of the same FE problem with additional Dirichlet boundary conditions applied on the other blocks. \square

Theorem 4. (Axelsson and Gustafsson,¹⁸ Shlafman and Efrat¹⁹). For 2-D membrane and 3-D solid problems, \mathbf{K} and \mathbf{K}^D are spectrally equivalent, i.e. there exist two positive constants α_D, β_D independent of the number NEL and the size h of the finite elements such that

$$\alpha_D \mathbf{x}^T \mathbf{K}^D \mathbf{x} \leq \mathbf{x}^T \mathbf{K} \mathbf{x} \leq \beta_D \mathbf{x}^T \mathbf{K}^D \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^N \quad (20)$$

Proof. See Axelsson and Gustafsson¹⁸ for membrane elements, Shlafman and Efrat¹⁹ for membrane and 3-D solid elements. \square

Note that Shlafman and Efrat¹⁹ claim that this result is also valid for plate and shell problems but they provide a proof only for 3-D solid elements.

Thanks to Theorems 2 and 4, the spectral equivalence of $\mathbf{S} = \underline{\mathbf{K}}^D$ and \mathbf{K} with respect to NEL is obvious, with the spectral bounds

$$\alpha = \alpha_D \alpha_A \quad \text{and} \quad \beta = \beta_D \beta_A = \beta_D \quad (21)$$

The spectral equivalence bound β/α does not depend on the number of elements *nor on the way the elements are connected* (regular or non-regular mesh). Theorem 5 also shows that β/α does not

depend on the size of the finite elements h . Note that the assumptions of Theorem 5 are less restrictive than those of Theorem 4.

Theorem 5. For 2-D membranes, 3-D solids, beams and C1 continuity plates, the spectral bounds α_A and β_A are independent of the size h of the elements when C-reduction is applied to $\mathbf{A} = \mathbf{K}^D$.

Proof. See Appendix IV. \square

4. NUMERICAL RESULTS

All the techniques presented in Sections 2 and 3 have been implemented in our PCG solver, which allows the user to choose the preconditioner (IC, MIC, DMIC, RIC or DRIC), its order (0 or 1) and the reduction (C or DC).

4.1. Some important remarks about the numerical results

Remark 1. We compare our PCG solver to the implementation of *Iron's frontal solver* FRONT of the industrial software SAMCEF (V4), distributed by SAMTECH (Liège, Belgium), generally accepted to be an efficient direct solver and widely used for aeronautic and civil engineering applications. Some numerical tests aiming to compare SAMCEF and the well-known MA28 direct solver package of the Harwell Subroutine Library are presented in Appendix V.

Remark 2. It is well-known²⁰ that the numbering of the unknowns has a significant effect on PCG performances. We do not address this problem here, relying instead on a recent study by Notay²¹ who recommends the use of level orderings often called *Reverse Cuthill-McKee* (with respect to Reference 27 where one such ordering was introduced). As these orderings may be defined on arbitrary grids, we used one of them, described in²¹:

- (a) A level structure is built with one of the most connected nodes of the graph as starting node;
- (b) Within each level, nodes with the smallest value of the ratio ($\#$ unnumbered neighbours)/($\#$ neighbours) are numbered first;
- (c) The obtained ordering is reversed.

On the other hand, the FRONT solver needs a frontwidth minimization, for which Sloan's frontwidth reducer²² has been used.

Remark 3. One generally uses² a stopping criterion ensuring that the initial error (the norm of the initial gradient \mathbf{g}^0) has been reduced by a given factor E ($= 10^{-6}$ or 10^{-8}) but this may be unsatisfactory if the initial error is too large (or even too small). We use here the criterion

$$\mathbf{g}^{kT} \mathbf{h}^k \leq \frac{E^2}{1+E} \lambda_1(\mathbf{B}^{-1} \mathbf{K}) \mathbf{q}^{kT} \mathbf{f} \quad (22)$$

where $\lambda_1(\mathbf{B}^{-1} \mathbf{K})$, the lowest eigenvalue of $\mathbf{B}^{-1} \mathbf{K}$, can be easily computed from the parameters τ_k , β_k of the PCG algorithm (see Reference 23, p. 2.20–2.27). Notay²³ showed that the fulfilment of inequality (22) guarantees

$$\frac{\|\mathbf{q}_{\text{exact}} - \mathbf{q}^k\|}{\|\mathbf{q}_{\text{exact}}\|} \leq E \quad (23)$$

in the \mathbf{K} -norm. In the carrying out of the presented numerical results, nearly the same precision was obtained for both direct and iterative solvers (with $E = 10^{-8}$) thanks to this realistic stopping criterion.

Remark 4. The last two points relate to some details of implementation allowing significant reductions of the computations times:

- (i) The preconditioner and the system matrix are diagonally scaled by \mathbf{P} to shortcut all operations involving \mathbf{P} in the PCG iterations and to save the storage of a real vector. Because the diagonal scaling is performed only once, the benefit brought by that scaling increases if more than one right-hand side is considered;
- (ii) For order 0 preconditioners, Eisenstat's algorithm²⁴ applies \mathbf{P} as preconditioner for solving

$$(\mathbf{U}^{-\mathbf{T}}\mathbf{K}\mathbf{U}^{-1})(\mathbf{U}\mathbf{q}) = (\mathbf{U}^{-\mathbf{T}}\mathbf{f})$$

instead of applying \mathbf{B} as preconditioner for solving system (1). The so-modified PCG scheme produces at each iteration the same approximated solution as the original algorithm but allows significant CPU time savings.

4.2. The range of examples

The presented numerical experiments are restricted here to 2-D membrane and 3-D solid problems but the same kind of results have been obtained for plates and shells even if the theory has not yet been fully extended to handle these problems. The finite elements used in the following are:

- REM4, REM8 (REctangular Membrane 4- or 8-node elements);
- H8, H20 (Hexaedral solid 8- or 20-node elements).

4.3. Efficiency of the different preconditioning techniques on regular grids

For regular grids using one single type of FE, the number of PCG iterations n can be plotted as a function of the number of d.o.f.s N of the grid. The plotted curves tend to be straight lines (for $N \geq 500$ roughly) in bilogarithmic axes, which corresponds to the exponential law of equation (24),

$$n \approx N^e \tag{24}$$

In that law, the value of e depends on the preconditioner and the reduction technique that are chosen but also on the type of FE used for the discretization.

Table VII(a) gives the number of iterations n needed by all the IC-like order 0 preconditioners for increasing number of d.o.f.s N and for different types of FE over regular square and cubic grids. For each preconditioner, DC- and C-reductions were performed separately, giving rise to two values for n . Table VII(b) contains the same informations for order 1 preconditioners.

Comparing the number of iterations for the DC- and the C-reduction, it can be seen that the DC-reduction is much more efficient than the C-reduction. This conclusion is especially true for the MIC preconditioner, for which the number of iterations is increased by a factor 6 to 7 in some of our numerical tests when only the C-reduction was used (see e.g. REM8 for $N = 38,720$ in Table VII(a)). Let us remark that the choice of the reduction is more critical for the MIC preconditioner because of its lack of stability, as mentioned in Section 2.

In practice, increasing the order from 0 to 1 leads to a significant improvement in the aggregate for the MIC, DMIC and RIC factorizations but roughly no change is gathered in for IC and DRIC preconditioners. However, even for IC and DRIC, the numbers of iterations are sometimes different. These small differences betray that there is no mathematical equivalence between order 0 and 1 IC and DRIC preconditioners. It can only be guaranteed that the numbers of iterations are quite similar in the range of the considered examples.

Since the improvement due to an increase from order 0 to 1 is often small, especially for DC-reduced preconditioners, order 0 is to be recommended since it requires less storage and more efficient implementations of PCG (like Eisenstat's²⁴) can then be used.

Table VII(c) shows the values of the exponent e of equation (24) for all the preconditioners built with a DC-reduction. DMIC, RIC and DRIC factorizations exhibit the best asymptotical behaviour, having the smallest values for e . These preconditioners must thus be preferred to any other XIC factorization when very large systems are considered. The values of e seem to be too close to each other to conclude which is the best preconditioner amongst DMIC, RIC and DRIC. However, it can be noticed that

- (i) e is always smaller for DRIC than for RIC;
- (ii) e is sometimes smaller for DMIC than for DRIC;
- (iii) However, when DMIC(0) performs better than DRIC(0), DMIC(1) may perform worse than DRIC(0) and DRIC(1) (for H8 examples).

Table VII(a). Number of iterations obtained by performing order 0 preconditionings with DC- or C-reduction on regular grids

Element	N	IC(0)		MIC(0)		DMIC(0)		RIC(0)		DRIC(0)	
		C	DC	C	DC	C	DC	C	DC	C	DC
REM4	220	35	36	38	33	34	32	33	32	34	32
	840	67	61	80	49	64	49	58	45	60	45
	1860	98	87	144	61	98	60	84	55	87	57
	3280	130	113	222	74	143	72	113	66	122	65
	5100	161	139	320	85	184	82	144	75	153	74
	7320	195	165	419	98	233	91	167	83	190	82
	9940	225	192	564	106	279	99	211	92	234	88
	12960	257	217	690	116	337	105	243	101	275	94
	16380	294	244	826	128	393	113	280	109	231	101
REM8	640	96	97	71	54	68	52	73	59	70	54
	2480	178	170	137	69	110	62	113	70	111	59
	5520	264	246	226	87	157	74	158	77	155	68
	9760	346	317	350	103	215	84	202	88	197	77
	15200	439	396	487	115	274	95	246	97	249	85
	21840	523	473	660	129	337	104	287	106	300	95
	29680	610	549	871	142	403	114	333	115	358	102
	38720	688	626	1036	153	463	121	373	123	411	108
	48960		703		166		132		125		115
H8	540	43	43	42	41	38	41	35	36	37	37
	1344	60	57	69	49	55	46	49	43	50	42
	3630	88	74	117	64	83	53	72	50	75	50
	6084	104	86	154	73	103	57	84	57	90	54
	9450	120	98	203	82	124	62	98	61	125	57
	13872	137	110	252	91	146	65	114	68	122	61
	19494	153	123	316	102	168	69	130	72	140	64
H20	504	123	110	102	94	101	95	102	97	108	99
	1980	162	133	143	111	128	102	126	104	127	100
	3276	185	145	171	116	140	101	139	107	140	105
	5040	218	175	212	124	160	105	156	115	158	109
	7344	251	201	254	149	188	117	178	127	184	124

Table VII(b). Number of iterations obtained by performing order 1 preconditionings with DC- or C-reduction on regular grids

Element	N	IC(1)		MIC(1)		DMIC(1)		RIC(1)		DRIC(1)	
		C	DC	C	DC	C	DC	C	DC	C	DC
REM4	220	35	36	40	33	36	32	33	32	34	32
	840	67	61	90	49	69	49	63	45	60	45
	1860	98	87	167	62	112	60	94	55	87	57
	3280	130	113	263	74	159	72	128	66	122	65
	5100	161	139	398	85	204	82	165	75	153	74
	7320	195	165	532	95	266	92	196	83	190	82
	9940	225	192	691	108	325	99	245	92	234	88
	12960	257	217	878	116	392	105	282	101	275	95
16380	294	244	1094	128	456	113	329	109	322	101	
REM8	640	96	97	71	50	67	49	69	54	70	54
	2480	178	170	145	67	111	59	109	62	111	59
	5520	264	246	242	83	159	72	153	71	155	68
	9760	346	317	359	98	218	80	205	78	197	77
	15200	439	396	516	108	278	89	250	89	249	87
	21840	523	473	710	124	342	100	294	99	300	95
	29680	610	549	898	137	411	109	345	108	358	102
38720		626		143		113		113		108	
H8	540	43	43	39	38	37	37	35	36	37	37
	1344	60	57	64	46	56	42	50	42	50	42
	3630	88	74	111	59	82	50	76	50	75	50
	6084	104	86	150	66	102	54	90	55	90	54
	9450	120	98	191	76	125	58	107	60	106	57
	13872	137	110	239	84	147	62	122	65	123	61
	19494		123		93		65		69		64
H20	504	122	110	99	86	101	94	102	96	108	101
	1980	162	133	139	97	127	97	124	97	127	99
	3276	185	145	159	102	137	94	138	99	140	106
	5040	218	175	189	113	156	104	155	109	158	109
	7344	251	201	228	126	180	104	183	119	184	124

Table VII(c). Value of the exponent e of the DC-reduced preconditioners applied to regular grids for different types of FE

	Order 0 preconditioners				Order 1 preconditioners			
	REM4	REM8	H8	H20	REM4	REM8	H8	H20
IC	0.4482	0.4571	0.2887	0.2168	0.4482	0.4571	0.2887	0.2168
MIC	0.3137	0.2606	0.2549	0.1524	0.3129	0.2633	0.2499	0.1339
DMIC	0.2913	0.2130	0.1458	0.0636	0.2918	0.2126	0.1597	0.0375
RIC	0.2853	0.1811	0.1928	0.0909	0.2853	0.1879	0.1823	0.0715
DRIC	0.2671	0.1799	0.1544	0.0708	0.2680	0.1799	0.1544	0.0653

In addition, in presence of anisotropies, DRIC is also robust while DMIC presents some weakness, as shown by Notay.⁴ Since these tendencies will be confirmed for irregular grids, we consider that DRIC is the best possible choice for a black box solver and will be chosen for comparison purpose with the reference direct solver.

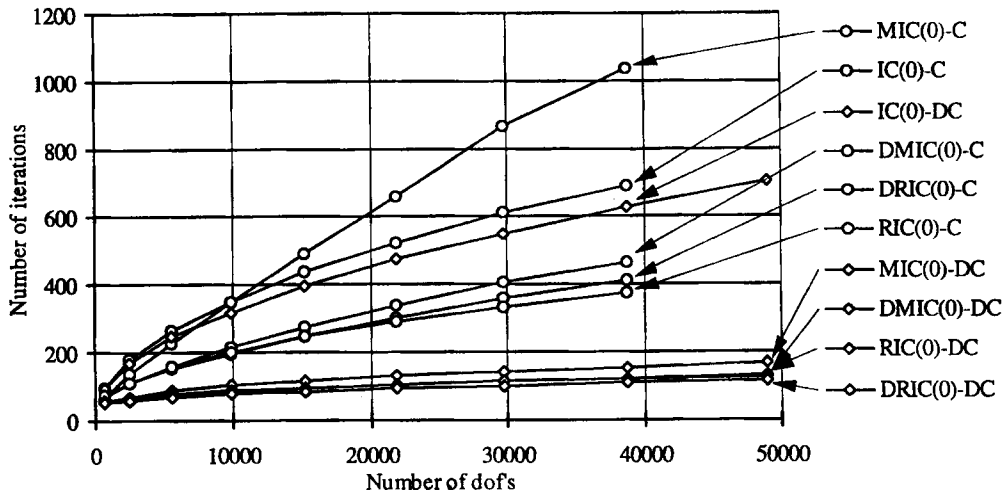


Figure 2. Number of iterations for different C- and DC-reduced order 0 preconditioners on REM8 regular grids

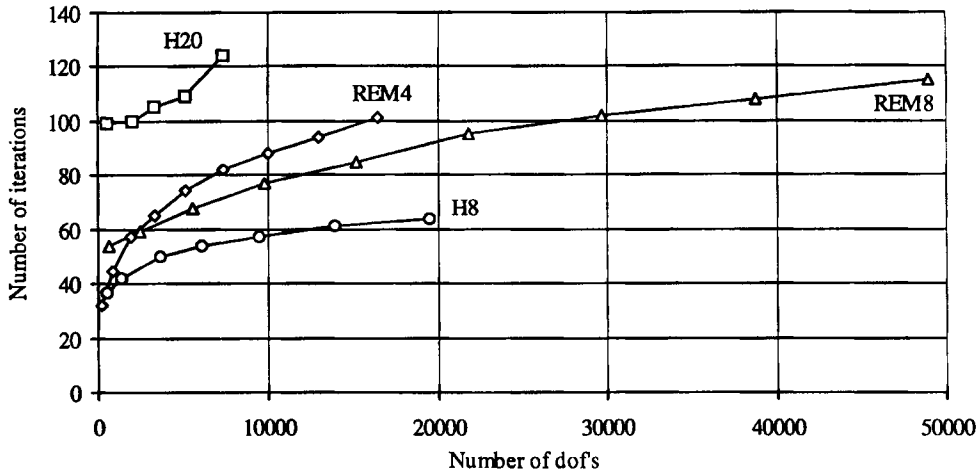


Figure 3. Number of iterations for the DRIC(0)-DC preconditioner on regular grids using various element types

Equation (24) is illustrated in Figure 2 where the number of iterations is plotted for REM8 grids for different preconditioners and in Figure 3 where the number of iterations is plotted for the DRIC(0)-DC preconditioner for different types of elements.

4.4. Comparison of iterative and direct solvers efficiencies on regular grids

For regular grids using the same type of finite element, the computational time t is only a function of the number of d.o.f.s N , following equation (25)

$$t \approx N^s \tag{25}$$

Table VIII(a). CPU times for the PCG-DRIC(0)-DC and FRONT solvers for regular grids (obtained on IBM 4381)

Example	N	PCG CPU times			FRONT
		(1) Iteration ^a	(2) Solving ^b	(3) Total ^c	Solving
REM4	220	0.405	0.575	0.838	1.108
	840	2.391	2.939	3.815	9.031
	1860	6.879	8.074	9.955	36.656
	3280	13.953	16.013	19.396	109.113
	5100	25.327	28.556	33.697	227.573
	7320	40.197	44.783	52.083	456.351
	9940	59.450	65.850	75.836	810.024
	12960	83.551	91.691	104.778	1497.427
	16380	112.169	122.484	138.985	2315.357
REM8	640	3.136	3.761	4.309	7.471
	2480	15.559	18.126	20.201	80.588
	5520	37.625	43.233	48.060	351.526
	9760	79.316	89.278	97.562	1138.358
	15200	130.478	145.766	158.703	2690.863
	21840	218.535	241.138	259.642	5250.614
	29680	307.051	336.728	363.226	10059.146
	38720	420.393	459.031	498.985	17542.381
	48960	582.388	632.292	689.003	29455.517
H8	540	3.099	4.060	5.427	27.175
	1344	10.861	13.439	17.377	242.312
	3630	36.960	44.575	56.450	2051.025
	6084	63.561	76.409	97.266	6859.362
	9450	107.673	128.957	162.874	19229.520
	13872	173.647	204.476	258.470	45299.115
	19494	256.410	300.513	400.626	93400.336
H20	504	13.142	14.764	16.864	34.911
	1980	63.954	71.805	83.854	765.153
	3276	117.534	131.415	152.714	2404.191
	5040	194.796	216.495	251.359	6497.124
	7344	326.620	358.411	422.500	15434.569

^a Column (1) = CPU for PCG iterations

^b Column (2) = Column (1) + CPU for reduction + approx. factorization + scaling

^c Column (3) = Column (2) + CPU for the assembly

Table VIII(b). Value of the exponent s for the FRONT and PCG-DRIC(0)-DC solvers

Solver	Type of element			
	REM4	REM8	H8	H20
FRONT	1.7830	1.9147	2.2674	2.2727
PCG-DRIC(0)-DC	1.1919	1.1675	1.1838	1.1926

In that law, the exponent s depends on the solver (FRONT or PCG). Table VIII(a) shows the CPU times needed by FRONT and PCG for which only the best order 0 preconditioner, that is DRIC(0) with the DC-reduction has been taken into account. These times were obtained on an IBM 4381 but our experience confirms that they are similar to those that would be obtained on

other workstations (IBM RS/6000, SUN SPARC, etc.). No tests were performed on supercomputers like Cray, that are used by a much fewer amount of industrial FEM users.

Three columns concern the PCG solver in Table VIII(a): the first one gives the CPU times associated to the iterative process, the second one adds to these main values the times needed to compute the reduction, the preconditioner and the scaling, and the third column shows the global times including the assembly of the system. This last time must be compared to the frontal solver time.

Table VIII(a) shows that even for small systems, the iterative scheme performs faster. The exponent s , given in Table VIII(b) for different types of FE, allows to extrapolate that conclusion to very large systems:

- (i) The value of s lies around 1.9 and 2.3 for the frontal method, respectively for 2-D and 3-D grids;
- (ii) For PCG-DRIC(0)-DC, the value of s remains almost unchanged in the 2-D and 3-D applications and lies around 1.18.

3-D structures are consequently very expensive in terms of (storage and) CPU times if FRONT is chosen as solver but this is not the case when PCG is chosen.

Let us recall here that the best result we could hope is to have $s = 1$ because the calculation time cannot increase less than the number of unknowns and note that DRIC(0)-DC yields values close to 1. Other iterative solvers like algebraic multilevel methods, combined with DC-reduction yield theoretically to $s = 1$ as asymptotical value. In any case, it is only an asymptotical behaviour and we do not have enough experience in dealing with multilevel methods to verify numerically their efficiency (i.e. to verify if they reach rapidly their asymptotical behaviour). Therefore, the values of s provided by DRIC(0)-DC seem satisfactory to us.

4.5. Performances of the different preconditioning techniques on non-regular grids

Excluding the influence of the frontwidth, the calculation times could still be extrapolated following equation (25) for the frontal method with the same exponent s but not for the PCG methods which are sensitive to mesh-specific parameters like element distortions, material discontinuities, For this reason, each numerical test must be considered separately.

The first non-regular grid tested is presented in Figure 4; it has been generated to study the membrane stress distribution in a parking floor. In that study, the plane stress hypothesis has been assumed and REM8 elements were used. In the following, we will refer to that grid as PARK. The second grid (Figure 5) has been built under the same assumptions; it is a REM8 discretization of a portion of a wall in a mansion (called MANS). 3-D grids were also tested, like a component of oven or an I-beam, using H8 elements. Those grids, named OVEN and BEAM, are represented in Figures 6 and 7.

Table IX(a) shows the number of iterations obtained by order 0 preconditioners, presented under the same format as in Table VII(a). Informations related to order 1 preconditioners are given in Table IX(b). It can be seen that DMIC, RIC and DRIC remain the most efficient schemes and that MIC exhibits again its lack of stability and poor convergence properties in the FE context. The use of DMIC or RIC order 1 preconditioners leads to small reductions of the number of iterations (about 10 per cent reduction for MANS, 8 per cent for PARK, 5 per cent for OVEN and no reduction for BEAM) compared to order 0 results, and the rate of convergence is simply not improved by increasing the order for the DRIC preconditioner.

Once more, DC-reduction is highly superior to the C-reduction. This allows to consider DRIC(0)-DC as the best preconditioning technique amongst those presented in this paper.

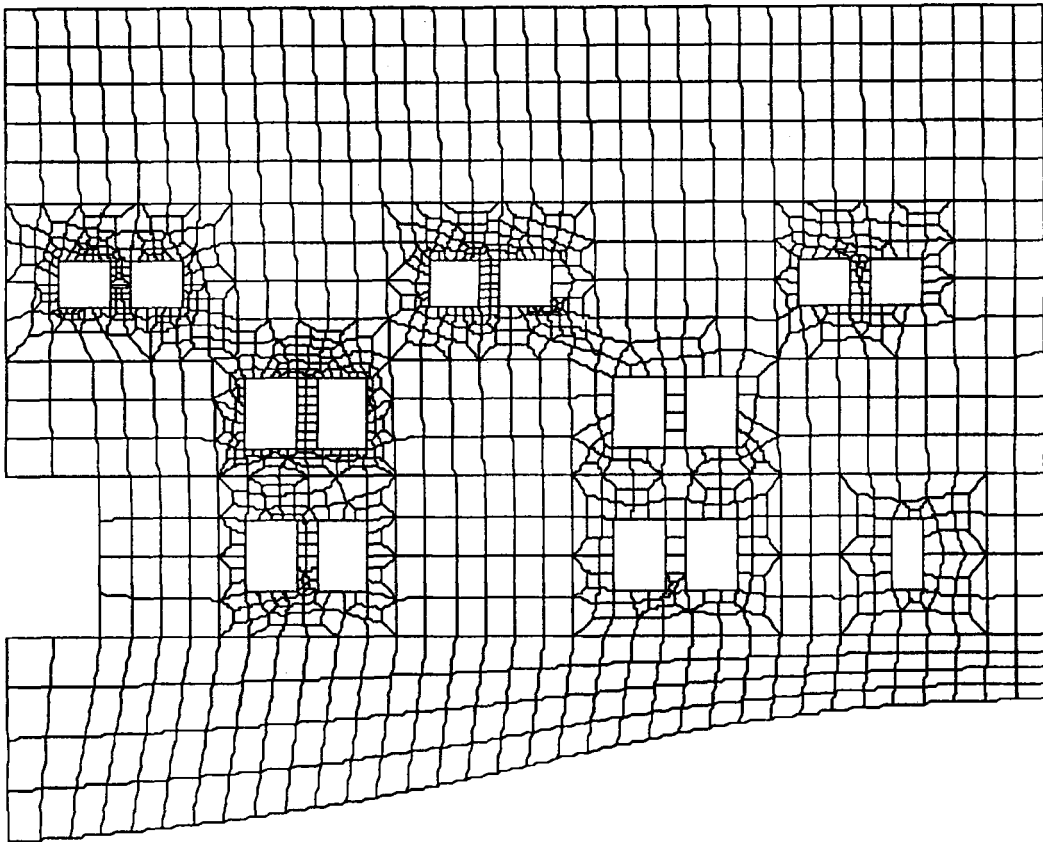


Figure 4. PARK: free mesh of a parking floor

4.6. Comparison of iterative and direct solvers efficiencies on non-regular grids

The same examples as in Section 4.5 are considered. Table X shows the total CPU times obtained for those examples by the frontal solver and the DRIC(0)-DC solver. The iterative solver remains quicker than the frontal one even for high gradients of elements sizes (MANS) or large element distortions (BEAM). Let us remark that the BEAM and OVEN grids have small frontwidths, which favour the frontal solver. PCG runs only two times faster than FRONT for the BEAM example.

4.7. The effect of material discontinuities

Material discontinuities have been generated as follows: regular square (REM4 and REM8) and cubic (H8 and H20) meshes are cut into two pieces of the same shape, containing the same number of elements, one having a Young's modulus 10 times greater than the other one. The number of iterations of our iterative solver on these non-uniform structures is compared in Table XI to that of the uniform structures of Table VII(a) for the DRIC(0)-DC preconditioner. The discontinuities have apparently no significant influence.

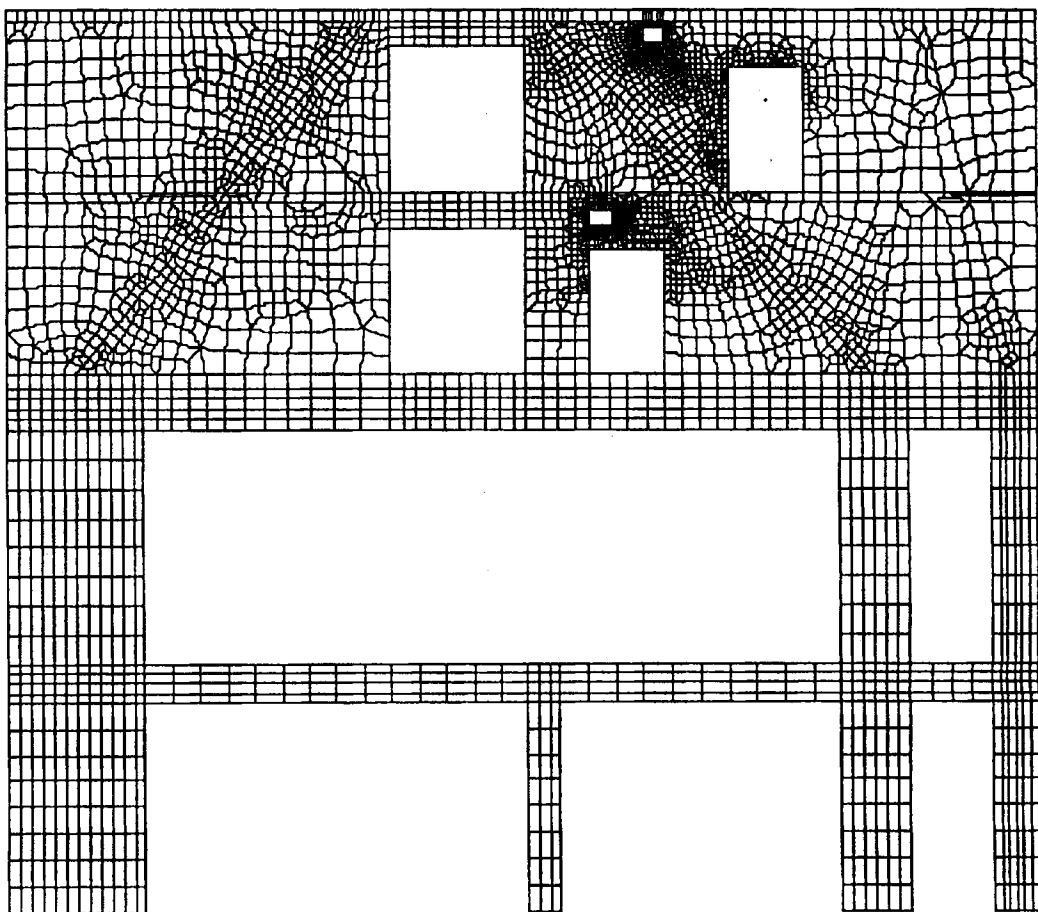


Figure 5. MANS: free mesh of a mansion wall

Table IX(a). Number of iterations obtained by performing order 0 preconditionings with DC- or C-reduction on non-regular grids

Example	N	IC(0)		MIC(0)		DMIC(0)		RIC(0)		DRIC(0)	
		C	DC	C	DC	C	DC	C	DC	C	DC
PARK	9067	453	449	*	*	503	191	422	196	423	184
MANS	24368	*	*	*	1042	1492	642	1279	680	1288	623
OVEN	23687	370	370	*	*	410	221	325	241	323	196
BEAM	19362	1169	1020	*	*	1199	677	1035	728	1073	649

* = more than 1501 iterations

Table IX(b). Number of iterations obtained by performing order 1 preconditionings with DC- or C-reduction on non-regular grids

Example	N	IC(1)		MIC(1)		DMIC(1)		RIC(1)		DRIC(1)	
		C	DC	C	DC	C	DC	C	DC	C	DC
PARK	9067	453	449	a	a	495	174	401	180	423	185
MANS	24368	a	a	a	947		594		608		621
OVEN	23687	370	370	a	a	399	210	347	236	325	195
BEAM	19362	1169	1020	a	a		679		725		647

^a = more than 1501 iterations

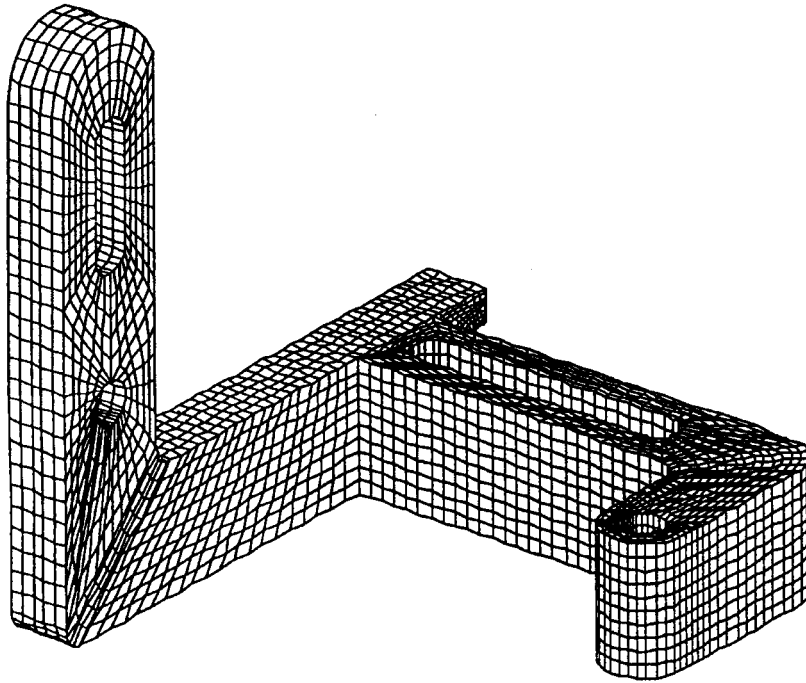


Figure 6. OVEN: free mesh of an oven component

Table X. CPU times for the PCG and FRONT solver for non-regular grids

Example	N	PCG CPU times			FRONT
		(1) Iteration ^a	(2) Solving ^b	(3) Total ^c	Solving
PARK	9067	164-060	173-339	180-378	5223-835
MANS	24368	1515-441	1539-534	1558-851	5292-520
OVEN	23687	578-947	610-050	673-929	9649-185
BEAM	19362	2277-499	2315-715	2384-131	4774-791

^a Column (1) = CPU for PCG iterations

^b Column (2) = Column (1) + CPU for reduction + approx. factorization + scaling

^c Column (3) = Column (2) + CPU for the assembly

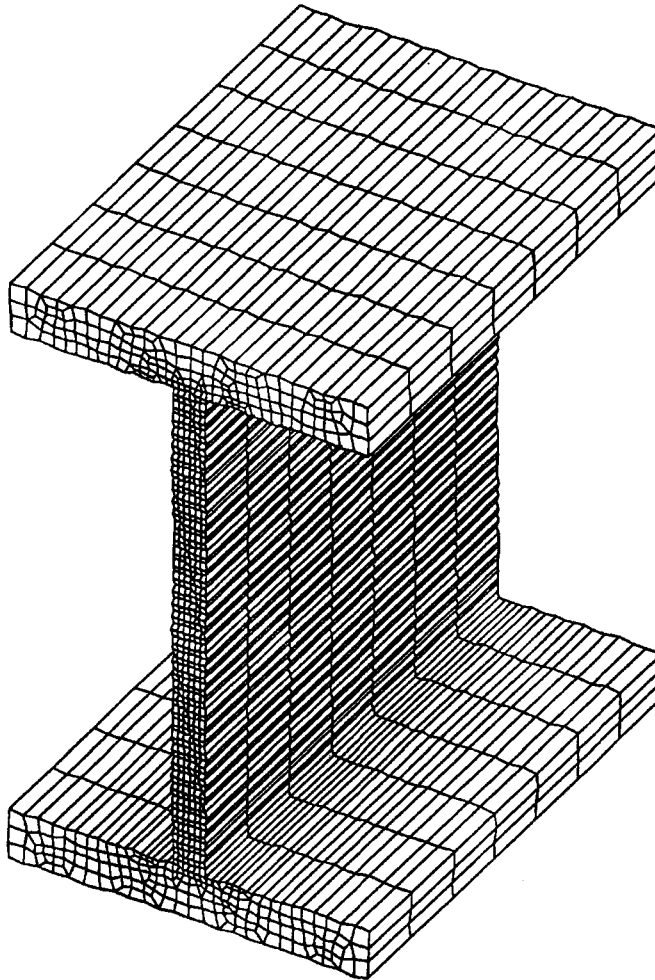


Figure 7. BEAM: free mesh of an I-beam

Table XI. Number of iterations of the DC-reduced order 0 IC-like preconditioners for uniform meshes and discontinuous meshes

Element	N	Mesh 1	Mesh 2
REM4	16380	101	103
REM8	38720	115	112
H8	19494	64	64
H20	7344	124	121

Mesh 1 = Uniform REM8 or H8 mesh with Young's modulus E_1

Mesh 2 = Discontinuous mesh: Young's modulus E_1 on the half mesh, $10 E_1$ on the other half

Table XII. Number of iterations for different values of the Poisson ratio ν with the DRIC(0)-DC preconditioner

ν	PARK	REM4 ($N = 388$)	REM4 ($N = 796$)
0.40000	184	45	55
0.49000	196	46	57
0.49900	197	46	57
0.49990	197	46	57
0.49999	198	46	57

4.8. The effect of the material properties and anisotropy

The quality of XIC preconditioners is admitted to be influenced by the material properties. Axelsson and Gustafsson¹⁸ have more particularly enhanced the effect of the Poisson ratio ν : in their experiments, values of ν near 0.5 deteriorate the spectral equivalence bound. This may be due to the presence of $(1 - 2\nu)$ in the Hooke matrix.

Table XII compares the effect of ν on the number of iterations for three of our grids, including a non-regular grid, with a DRIC(0)-DC preconditioner. It seems that thanks to the DC-reduction, the effect of ν on the number of iterations is fairly slight and can be neglected, which is an important improvement from the robustness point of view.

Following Notay,⁴ the robustness of the DRIC preconditioner extends to anisotropic problems, contrary to DMIC. Note that the Poisson ratio ν , when non-zero, introduces anisotropy in the discretized equations. However, additional experiments not presented here for brevity have shown that even when other XIC factorizations than DRIC are used, the effect of ν can be neglected; this highlights the fact that the insensitivity of the number of iterations with respect to ν is due to our DC reduction and not to the choice of DRIC amongst the other XIC schemes.

5. CONCLUSIONS

5.1. On the choice of an iterative scheme

Three aspects must be taken into account when choosing an approximate factorization for PCG preconditioning:

- (i) *The reduction*: the need for an efficient reduction technique in the FEM structural analysis context has been highlighted and reduction techniques yielding Stieltjes factorizable matrices have been presented and justified. The coupling of two reduction schemes, decoupling and diagonal compensation, leads to highly efficient preconditioners;
- (ii) *The approximate factorization*: the preconditioner DRIC recently developed by Notay⁴ has been presented and significant numerical tests show that also for structural FEM problems, it is one of the most powerful of the IC family of preconditioners: IC has weak convergence properties, MIC is not robust enough, RIC and DMIC sometimes perform better at order 1 but are more sensitive to grid non-uniformity, material discontinuities or anisotropy;

- (iii) *The order of the factorization:* the profit carried by medium-order preconditioners is often too low to make them seem attractive for problems arising from the FEM discretization in structural analysis, at least for DRIC-DC preconditioning. This allows us to use order 0 schemes that have small memory requirements.

The combination of DRIC(0) and the hybrid DC reduction leads to the lowest CPU times amongst the different popular preconditioning techniques referred here.

5.2. PCG versus frontal method

It is now accepted that the memory requirements of a PCG solver are much lower than those of a direct solver, but the comparison of CPU times feeds a great polemic for a lot of years. This is due to parameters that influence the convergence rate of the PCG solver (for a given problem), which are as follows:

- (i) *The type of finite elements:* This paper shows that PCG methods lead probably to one of the most efficient solvers for some types of finite element. This has been theoretically and numerically demonstrated for membrane and solid elements in this work but numerical results have been obtained separately for plate and shell problems;
- (ii) *The preconditioner:* The improvement brought forward by recently developed approximate factorization algorithms reduces dramatically the number of iterations;
- (iii) *The reduction* is generally ignored by several authors but this paper shows how to combine different reduction techniques to carry out efficient preconditioners in the structural analysis context;
- (iv) *The computer* on which the software codes are tested: the sparse data storage schemes and the ordering of the unknowns must be completely re-thought to run PCG on vector or parallel computers and get the same conclusions as here on a scalar computer when comparing PCG to FRONT. However, the most used computers in industry are scalar and mono-processor machines, on which our contribution allows to increase the size of the solved problems, now restricted by the use of direct methods.

DRIC(0)-DC computational times presented here are far smaller than those obtained by the frontal method, even if the grids are non-regular, with high element size gradients or shape distortions. This conclusion remains valid for 2-D structures and is reinforced for 3-D structures, and it seems that the PCG method must always be preferred to a direct method.

ACKNOWLEDGEMENTS

We are grateful to SAMTECH for having provided their software code SAMCEF and to the referees for their helpful comments that have widely contributed to the completion of this paper.

APPENDIX I: PROOF OF THEOREM 2

C-reduction applied to \mathbf{A} and \mathbf{A}^e requires equations (26)–(33) to be satisfied:

$$\mathbf{A} = \underline{\mathbf{A}} - \bar{\mathbf{A}} \quad (26)$$

$$\text{offdiag}(\underline{\mathbf{A}}) = \min(\text{offdiag}(\mathbf{A}), \mathbf{0}) \quad (27)$$

$$\text{offdiag}(\bar{\mathbf{A}}) = -\max(\text{offdiag}(\mathbf{A}), \mathbf{0}) \quad (28)$$

$$\underline{\mathbf{A}}\mathbf{1} = \mathbf{A}\mathbf{1} \quad (29)$$

$$\mathbf{A}^e = \underline{\mathbf{A}}^e - \bar{\mathbf{A}}^e \quad (30)$$

$$\text{offdiag}(\underline{\mathbf{A}}^e) = \min(\text{offdiag}(\mathbf{A}^e), \mathbf{0}) \quad (31)$$

$$\text{offdiag}(\bar{\mathbf{A}}^e) = -\max(\text{offdiag}(\mathbf{A}^e), \mathbf{0}) \quad (32)$$

$$\underline{\mathbf{A}}^e\mathbf{1} = \mathbf{A}^e\mathbf{1} \quad (33)$$

and the elementary matrices are tied to the global ones by equations (34) and (35)

$$\mathbf{A} = \sum_e \mathbf{A}^e \quad (34)$$

$$\underline{\mathbf{A}}^* = \sum_e \underline{\mathbf{A}}^e \quad (35)$$

Upper spectral equivalence bound β_A . Equation (26) yields

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T (\underline{\mathbf{A}} - \bar{\mathbf{A}}) \mathbf{x} \leq \mathbf{x}^T \underline{\mathbf{A}} \mathbf{x}$$

since $\bar{\mathbf{A}}$ is at least non-negative definite from Theorem 1. Then, in equation (19)

$$\beta_A = 1 \quad (36)$$

Lower spectral equivalence bound α_A . If assumption (18) is satisfied, there exists always, for any finite element of the structure, an $\alpha_e > 0$ that satisfies equation (37),

$$\alpha_e \mathbf{x}^T \underline{\mathbf{A}}^e \mathbf{x} \leq \mathbf{x}^T \mathbf{A}^e \mathbf{x} \quad (37)$$

Now, let

$$\alpha_{\min} = \min(\alpha_e) \quad (38)$$

Then,

$$\alpha_{\min} \mathbf{x}^T \underline{\mathbf{A}}^e \mathbf{x} \leq \mathbf{x}^T \mathbf{A}^e \mathbf{x} \quad (39)$$

implies by summing for all e

$$\alpha_{\min} \mathbf{x}^T \underline{\mathbf{A}}^* \mathbf{x} \leq \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (40)$$

or

$$\alpha_{\min} \mathbf{x}^T (\underline{\mathbf{A}} + (\underline{\mathbf{A}}^* - \underline{\mathbf{A}})) \mathbf{x} \leq \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (41)$$

If $(\underline{\mathbf{A}}^* - \underline{\mathbf{A}})$ is non-negative definite, which is proven in Appendix III,

$$\alpha_{\min} \mathbf{x}^T \underline{\mathbf{A}} \mathbf{x} \leq \mathbf{x}^T \mathbf{A} \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^N$$

and in equation (19)

$$\alpha_A = \alpha_{\min} \quad (42)$$

The upper bound does not depend on anything; the lowest bound does not depend on the number of elements. \square

APPENDIX II: PROOF OF ASSUMPTION (18): $\ker(\mathbf{A}^e) = \ker(\underline{\mathbf{A}}^e)$

The proof is made for $\mathbf{A}^e = \mathbf{K}^{eD}$. Without any loss of generality, let us consider a problem with two translation d.o.f.s x, y and one rotation d.o.f. θ ,

$$\mathbf{K}^{eD} = \sum_i [\mathbf{K}_{ii}^{eD}] = \begin{bmatrix} \mathbf{K}_{xx}^{eD} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yy}^{eD} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{\theta\theta}^{eD} \end{bmatrix} \tag{43}$$

$$\underline{\mathbf{K}}^{eD} = \sum_i [\underline{\mathbf{K}}_{ii}^{eD}] = \begin{bmatrix} \underline{\mathbf{K}}_{xx}^{eD} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{K}}_{yy}^{eD} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \underline{\mathbf{K}}_{\theta\theta}^{eD} \end{bmatrix} \tag{44}$$

where $[\mathbf{K}_{ii}^{eD}]$ and $[\underline{\mathbf{K}}_{ii}^{eD}]$ have the same entries as \mathbf{K}_{ii}^{eD} and $\underline{\mathbf{K}}_{ii}^{eD}$, padded with zeroes to have the same size as \mathbf{K}^{eD} and $\underline{\mathbf{K}}^{eD}$. Let us take a look at the kernel of each block:

Translation d.o.f.s: $\ker(\mathbf{K}_{ii}^{eD}) \subset \text{span}\{\mathbf{1}\}$. These kernels would be reduced to $\{\mathbf{0}\}$ if boundary conditions were applied to the corresponding d.o.f.s;

Rotation d.o.f.s: $\ker(\mathbf{K}_{\theta\theta}^{eD}) = \{\mathbf{0}\}$.

Moreover, equation (33) implies

$$\underline{\mathbf{K}}_{ii}^{eD} \mathbf{1} = \mathbf{K}_{ii}^{eD} \mathbf{1} \tag{45}$$

and therefore

$$\ker(\mathbf{K}^{eD}) \subset \ker(\underline{\mathbf{K}}^{eD}) \tag{46}$$

But since

$$\begin{aligned} \mathbf{x}^T \mathbf{K}^{eD} \mathbf{x} &= \mathbf{x}^T (\underline{\mathbf{K}}^{eD} - \bar{\mathbf{K}}^{eD}) \mathbf{x} \\ &\geq \mathbf{x}^T \underline{\mathbf{K}}^{eD} \mathbf{x} \end{aligned}$$

from Theorem 1, if $\mathbf{K}^{eD} \mathbf{x} = 0$, then $\underline{\mathbf{K}}^{eD} \mathbf{x} = 0$ and

$$\ker(\underline{\mathbf{K}}^{eD}) \subset \ker(\mathbf{K}^{eD}) \tag{47}$$

Equations (46) and (47) provide

$$\ker(\mathbf{K}^{eD}) = \ker(\underline{\mathbf{K}}^{eD}) \quad \square$$

APPENDIX III: $(\underline{\mathbf{A}}^* - \underline{\mathbf{A}})$ IS NONNEGATIVE DEFINITE

From equations (35), (33), (34) and (29) successively, one has

$$\underline{\mathbf{A}}^* \mathbf{1} = \sum_e \underline{\mathbf{A}}^e \mathbf{1} = \sum_e \mathbf{A}^e \mathbf{1} = \mathbf{A} \mathbf{1} = \underline{\mathbf{A}} \mathbf{1}$$

which leads to

$$-\sum_{i \neq j} (\underline{\mathbf{A}}^* - \underline{\mathbf{A}})_{ij} = (\underline{\mathbf{A}}^* - \underline{\mathbf{A}})_{ii} \tag{48}$$

On the other hand, it can be seen from equations (26)–(35) that

$$\begin{aligned}
 (\underline{\mathbf{A}}^*)_{ii} &= \sum_j \sum_e \max((\underline{\mathbf{A}}^e)_{ij}, 0) \\
 (\underline{\mathbf{A}})_{ii} &= \sum_j \max\left(\sum_e (\underline{\mathbf{A}}^e)_{ij}, 0\right) \\
 (\underline{\mathbf{A}}^*)_{ij} &= \sum_e \min((\underline{\mathbf{A}}^e)_{ij}, 0) \quad \text{for } i \neq j \\
 (\underline{\mathbf{A}})_{ij} &= \min\left(\sum_e (\underline{\mathbf{A}}^e)_{ij}, 0\right) \quad \text{for } i \neq j
 \end{aligned}$$

and therefore

$$(\underline{\mathbf{A}}^*)_{ij} \geq (\underline{\mathbf{A}})_{ij} \quad \text{for } i = j \tag{49}$$

$$(\underline{\mathbf{A}}^*)_{ij} \leq (\underline{\mathbf{A}})_{ij} \quad \text{for } i \neq j \tag{50}$$

and the conclusion follows from the Gershgorin theorem. □

APPENDIX IV: PROOF OF THEOREM 5

The spectral bounds α_A, β_A of Theorem 2 depend on the size of the elements h only through the parameter α_e , defined for each element. We prove here that at least for 2-D membranes, 3-D solids, beams and C1 continuity plates, α_e is independent of the size of the element e .

Proof for 2-D membrane, 3-D solid and C1 continuity plate elements. We use here the isoparametric transformation to map any finite element e on a given parent element p with a size ratio h , as shown in Figure 8. Let $(x, y, z), (\xi, \eta, \zeta)$ be the space co-ordinates in elements e and p , respectively, and

$$x_i, y_i, z_i = \pm h \tag{51}$$

are the nodes co-ordinates of element e .

Then²⁵

$$\mathbf{K}^e = \int_p (\mathbf{DN})^T \mathbf{H} (\mathbf{DN}) |\det \mathbf{J}| d\xi d\eta d\zeta \tag{52}$$

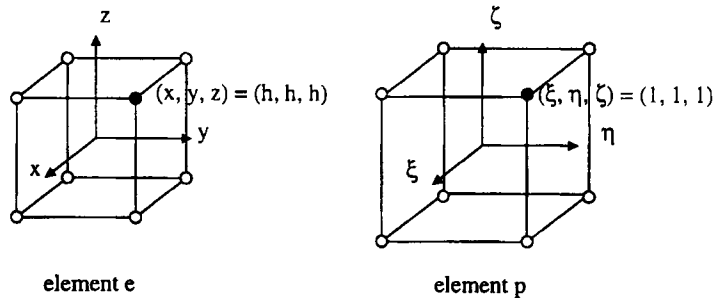


Figure 8. Isoparametric transformation with size ratio h

where \mathbf{H} is the Hooke matrix; \mathbf{N} is the matrix of the shape functions; \mathbf{J} is the Jacobian matrix of the isoparametric transformation that can be written, N_i being the shape functions at node i ,

$$\mathbf{J} = \begin{bmatrix} \partial_\xi x & \partial_\xi y & \partial_\xi z \\ \partial_\eta x & \partial_\eta y & \partial_\eta z \\ \partial_\zeta x & \partial_\zeta y & \partial_\zeta z \end{bmatrix} = \begin{bmatrix} \sum_i \partial_\xi N_i x_i & \sum_i \partial_\xi N_i y_i & \sum_i \partial_\xi N_i z_i \\ \sum_i \partial_\eta N_i x_i & \sum_i \partial_\eta N_i y_i & \sum_i \partial_\eta N_i z_i \\ \sum_i \partial_\zeta N_i x_i & \sum_i \partial_\zeta N_i y_i & \sum_i \partial_\zeta N_i z_i \end{bmatrix}$$

in which all entries are proportional to h thanks to equation (48). The value of $|\det \mathbf{J}|$ is then proportional to h^{\dim} with $\dim = 1, 2$ or 3 for 1-D, 2-D or 3-D finite elements respectively; \mathbf{D} is a matrix of differential operators using x -, y - and z -derivatives. As, for instance,

$$\partial_x = (\partial_{x\xi})\partial_\xi + (\partial_{x\eta})\partial_\eta + (\partial_{x\zeta})\partial_\zeta$$

in which all terms between brackets are entries of \mathbf{J}^{-1} , and thus proportional to h^{-1} , all first derivatives are proportional to h^{-1} . Moreover, order d derivatives are proportional to h^{-d} .

If \mathbf{D} is homogeneous (all the derivatives being of the same order) of order d , it follows from equation (52) that

$$(\mathbf{K}^e)_{h=h} = h^{\dim-2d}(\mathbf{K}^e)_{h=1} = h^{\dim-2d}\mathbf{K}^p$$

and obviously, for matrices $\mathbf{A}^e, \mathbf{A}^p$ computed from $\mathbf{K}^e, \mathbf{K}^p$ by a D-reduction (for instance),

$$(\mathbf{A}^e)_{h=h} = h^{\dim-2d}\mathbf{A}^p \tag{53}$$

$$(\underline{\mathbf{A}}^e)_{h=h} = h^{\dim-2d}\underline{\mathbf{A}}^p \tag{54}$$

Therefore, as it can always be written that for some positive α_e ,

$$\alpha_e \mathbf{x}^T \underline{\mathbf{A}}^p \mathbf{x} \leq \mathbf{x}^T \mathbf{A}^p \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^N$$

and

$$\alpha_e h^{\dim-2d} \mathbf{x}^T \underline{\mathbf{A}}^p \mathbf{x} \leq h^{\dim-2d} \mathbf{x}^T \mathbf{A}^p \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^N \tag{55}$$

Equations (53)–(55) allow to write equation (37) with the same α_e , independent of the size h of the element.

This proof is valid only if \mathbf{D} is homogeneous, like for 2-D membrane, 3-D solid or C1 continuity plate elements (see Reference 25). This condition is however not always fulfilled for C0 continuity plate elements (where some entries of \mathbf{D} are first order derivatives and some others are non-zero constants). □

Proof for beam elements. Note that the isoparametric transformation is not needed for beam elements since their matrices are easily established in structural axes. For instance, the stiffness matrix of a linear 2-D beam is

$$\mathbf{K}^e = \begin{bmatrix} A & -A & 0 & 0 & 0 & 0 \\ -A & A & 0 & 0 & 0 & 0 \\ 0 & 0 & B & -B & C & -C \\ 0 & 0 & -B & B & -C & -C \\ 0 & 0 & C & -C & 2D & D \\ 0 & 0 & -C & -C & D & 2D \end{bmatrix} \tag{56}$$

Table XIII. Comparison of FRONT and MA28 CPU times for some grids

Example	FRONT	MA28
PARK	24.21	79.90
MANS	116.32	528.70
BEAM	109.81	1895.80

with $A = E\Omega h^{-1}$, $B = 12EIh^{-3}$, $C = 6EIh^{-2}$ and $D = 2EIh^{-1}$, E being the Young's modulus, I the inertia of the beam and Ω its section.

The corresponding \mathbf{K}^{eD} and $\underline{\mathbf{K}}^{eD}$ are partitioned into three (2, 2) diagonal blocks $[\mathbf{K}_{ii}^{eD}]$ and $[\underline{\mathbf{K}}_{ii}^{eD}]$ corresponding to x , y and θ d.o.f.s, as proposed in equations (43) and (44). Each of these blocks contains only entries that are proportional to h^{-1} (x and θ blocks) or h^{-3} (y block). In equation (57)

$$\alpha_e \mathbf{x}^T [\underline{\mathbf{K}}_{ii}^{eD}] \mathbf{x} \leq \mathbf{x}^T [\mathbf{K}_{ii}^{eD}] \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^N, \forall i \quad (57)$$

the size parameter h can always be considered as a leading constant with a given exponent. The value of α_e in equation (57) is then independent of h . Equation (37) is finally obtained by summing equation (57) for all i , with respect to equations (43) and (44) and by taking $\mathbf{A}^e = \mathbf{K}^{eD}$. \square

APPENDIX V

The direct solver used here is the frontal solver of the industrial software code SAMCEF. Some CPU times given in Table XIII allow a comparison of FRONT and the frontal solver of the MA28 package of the Harwell Subroutine Library, which has a larger popularity in the research area. For technical reasons, the experiments could not have been made on an IBM 4381 with SAMCEF(V4); they were performed on a SUN SPARC20/514 with SAMCEF(V5) but it is expected to find the same qualitative conclusions. Partial pivoting was used for the MA28 solver.

It has been found that FRONT is always more efficient than MA28, which seems not to be optimized for industrial use.

REFERENCES

1. E. L. Poole, N. F. Knight and D. D. Davis Jr, 'High performance equation solvers and their impact on finite element analysis problems', *Int. j. numer. methods eng.*, **33**, 858–868 (1992).
2. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems—Theory and Computation*, Academic Press, New York, 1984.
3. R. Beauwens, 'Modified incomplete factorization strategies', in O. Axelsson and L. Kotolina (eds.), *PCG Methods*, Lecture Notes in Mathematics, **1457**, Springer, Berlin, 1990, pp. 1–16.
4. Y. Notay, 'A dynamic version of the RIC method', *Numer. Linear Algebra Appl.* **1**(4), (1994).
5. G. Meinardus, 'Ueber eine Verallgemeinerung einer Ungleichung von L. V. Kantorowitsch', *Numer. Math.* **5**, 14–23 (1963).
6. R. Beauwens, 'Factorization iterative methods, M-operators and H-operators', *Numer. Math.* **31**, 335–357 (1979).
7. S. Woznicki, 'Two-sweep iteration methods for solving large linear systems and their application to the numerical solution of multi-group multi-dimensional neutron diffusion equations', *Ph.D. Dissertation, Report 1447/CYF-RONET/PM/A*, Institute of Nuclear Research, Swieck, Poland, 1973.
8. H. S. Price and R. S. Varga, 'Incomplete primitive factorizations', unpublished manuscript, 1964.
9. O. Axelsson, 'A generalized SSOR method', *BIT* **13**, 443–467 (1972).
10. I. Gustafsson, 'A class of first order factorization methods', *BIT* **18**, 142–156 (1978).
11. N. I. Buleev, 'A numerical method for the solution of two-dimensional and three-dimensional equations of diffusion', *Math. Sb.* **51**, 227–238 (1960); English translation in *Report BNL-TR-551*, Bookhaven, National Laboratory, Upton, NY, 1973.

12. I. Gustafsson, 'Modified incomplete Cholesky (MIC) methods', in D. Evans (ed.), *Preconditioning Methods, Theory and Applications*, Gordon and Breach, NY, 1983, pp. 265–293.
13. M. M. Magolu, 'Taking advantage of the potentialities of dynamically modified block incomplete factorizations', *Report IT/IF/14-13*, Service de Métrologie Nucléaire, Université Libre de Bruxelles.
14. O. Axelsson and L. Kolotilina, 'Diagonally compensated reduction and related preconditioning methods', *Numer. Linear Algebra Appl.*, **1**(2), 155–177 (1994).
15. R. Beauwens and R. Wilmet, 'Conditioning analysis of positive definite matrices by approximate factorizations', *J. Comput. Appl. Math.*, **26**, 257–269 (1989).
16. J. K. Dickinson and P. A. Forsyth, 'Preconditioned conjugate gradient methods for three-dimensional linear elasticity', *Int. j. numer. methods eng.*, **37**, 2211–2234 (1994).
17. O. Axelsson, *Iterative solution methods*, Cambridge University Press, Cambridge, 1994.
18. O. Axelsson and I. Gustafsson, 'Iterative methods for the solution of the Navier equations of elasticity', *Comput. Methods Appl. Mech. Eng.*, **15**, 241–258 (1978).
19. S. Shlafman and I. Efrat, 'Using Korn's inequality for an efficient iterative solution of structural analysis problems', in R. Beauwens and P. de Groen (eds.), *Iterative Methods in Linear Algebra*, North-Holland, Amsterdam, 1992, pp. 575–581.
20. I. S. Duff and G. A. Meurant, 'The effect of ordering on preconditioned conjugate gradients', *BIT Comput. Sci. Numer. Math.*, **29**, 635–657 (1989).
21. Y. Notay, 'Ordering methods for approximate factorization preconditioning', Technical Report IT/IF/14-11, Service de Métrologie Nucléaire, Université Libre de Bruxelles, 1993.
22. S. W. Sloan, 'An algorithm for profile and wavefront reduction of sparse matrices', *Int. j. numer. methods eng.*, **23**, 239–251 (1986).
23. Y. Notay, 'Résolution itérative de systèmes linéaires par factorisations approchées', *Ph.D. Thesis*, Service de Métrologie Nucléaire, Université Libre de Bruxelles, Belgium, 1991.
24. S. Eisenstat, 'Efficient implementation of a class of preconditioned conjugate gradient methods', *SIAM J. Sci. Statist. Comput.*, **2**, 1–4 (1981).
25. O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, 4th edn, vol. 1 and 2, McGraw Hill, New York, 1989.
26. O. Axelsson and G. Lindskog, 'On the rate of convergence of the preconditioned conjugate gradient method', *Numer. Math.*, **48**, 499–523 (1986).
27. E. Cuthill and J. McKee, 'Reducing the bandwidth of sparse symmetric matrices', *Proc. 24th Nat. Conf. of the Assoc. for Computing Machinery*, Brandon Press, N.J., 1969, pp. 157–172.
28. M. R. Hestenes and E. Stiefel, 'Methods of conjugate gradient for solving linear systems', *J. Res. Nat. Bureau Standards Sect. B***49**, 409–436 (1952).