Preconditioned conjugate gradient method for the non-linear finite element analysis with particular reference to 3D reinforced concrete structures

M. Cervera, Y. C. Liu and E. Hinton
Department of Civil Engineering, University College of Swansea, Singleton Park, Swansea SA2 8PP, UK
(Received November 1985)

ABSTRACT

A hierarchically preconditioned conjugate gradient (PCG) method for finite element analysis is presented. Its use is demonstrated for the difficult problem of the non-linear analysis of 3D reinforced concrete structures. Examples highlight the dramatic savings in computer storage and more modest savings in solution times obtained using PCG especially for large problems.

INTRODUCTION

A major difficulty in three-dimensional finite element stress analysis is the amount of core-storage and computational time required to solve the equilibrium equations. For a typical 3D configuration, several thousands of degrees of freedom may be required to obtain reasonable accuracy. The CPU time needed to solve the linearized system of equations increases quadratically as the number of degrees of freedom increases. A bigger problem, however, is the core-storage required to store the assembled stiffness matrix. This is not only dependent on the size of the problem, but also on the finite element mesh topology.

Iterative solution techniques do not usually require the explicit assembly and storage of the global stiffness matrix. This results in a reduction of in-core and out-of-core computer storage which can make three-dimensional analysis of large systems feasible.

A hierarchically preconditioned conjugate gradient (PCG) method is employed in this paper for the non-linear analysis of three-dimensional reinforced concrete structures. If it is used in an incremental–iterative solution scheme to replace the direct solver then it is shown to save, not only core-storage, but also CPU time when slack convergence tolerances are used.

GOVERNING EQUATION

Non-linear equilibrium equations and their solution

Using the standard finite element procedures[1] the equations governing the non-linear behaviour of a structural system can be derived from the principle of virtual work. The resulting non-linear equations of equilibrium can be written as:

$$\boldsymbol{\psi} = \mathbf{f} - \mathbf{p}(\mathbf{d}) \tag{1}$$

where

$\boldsymbol{\psi}$ is the vector of residual forces,
$\mathbf{f}$ is the vector of external applied forces,
$\mathbf{p}(\mathbf{d})$ is the vector of internal resisting forces, and
$\mathbf{d}$ is the vector of nodal displacements.

In order to trace the entire response of the structure, an incremental solution procedure is usually adopted with a reasonable number of load increments. This incremental procedure must be associated with an iterative method to dissipate any residual forces. A typical method is shown in *Box 1*. The linearized iterative equations to be solved for each iteration during a typical load increment have the form (see 7 of *Box 1*):

$$\mathbf{K}_T\, \delta\mathbf{d} = \boldsymbol{\psi} \tag{2}$$

where

$\mathbf{K}_T$ is the tangential stiffness matrix, or an approximation to it,
$\delta\mathbf{d}$ is the vector of iterative displacements.

Box 1 Iterative procedure for typical increment n. NS chart

1. For increment n, set iteration number $i=1$
2. As initial solution guess, set $\mathbf{d}_i^n = \mathbf{d}^{n+1}$ and $\boldsymbol{\sigma}_i^n = \boldsymbol{\sigma}^{n+1}$ from last converged solution
3. Set current load level $\mathbf{f}^n = \lambda^n \mathbf{f}$ (λ^n is load parameter and $\mathbf{f}$ is reference load)
4. Evaluate residual forces
$$\boldsymbol{\psi}(\mathbf{d}_i^n) = \mathbf{f}^n - \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}_i^n \, d\Omega$$
5. Perform convergence check on norm of residual forces, displacements, energy, etc.

Has solution converged?

No

6. Update tangential stiffness $\mathbf{K}_{Ti}^n$ if necessary and factorize
7. Evaluate iterative displacements by backsubstitution
$$\delta\mathbf{d}_i^n = [\mathbf{K}_{Ti}^n]^{-1} \boldsymbol{\psi}_i^n$$
8. Update displacements by line search
$$\mathbf{d}_{i+1}^n = \mathbf{d}_i^n + s_i \delta\mathbf{d}_i^n$$
9. Update stresses $\boldsymbol{\sigma}_{i+1}^n$, set $i=i+1$ and go to 4

Yes

10. Set
$$\mathbf{d}^n = \mathbf{d}_i^n$$
$$\boldsymbol{\sigma}^n = \boldsymbol{\sigma}_i^n$$
11. Update increment $n=n+1$ and go to 1

Notes on Box 1

(a) For the first iteration of a load increment (see 4) $\boldsymbol{\psi}$ = (the increment of load, $\Delta\mathbf{f} = \mathbf{f}^n - \mathbf{f}^{n-1}$, + the residual forces from the last load increment.) Thus, there is no 'drift off' from the equilibrium curve as an attempt to satisfy equilibrium is made at all times.

(b) If $\mathbf{K}_{Ti}^n$ is always updated for each iteration, then this implies the Newton–Raphson (NR) method. If $\mathbf{K}_{Ti}^n$ is never updated after the initial linear stiffness is evaluated, then this implies the initial stiffness (IS) method. If $\mathbf{K}_{Ti}^n$ is updated only once per increment (or only occasionally, in response to changes in a solution monitoring parameter, such as the current stiffness parameter), then this implies the modified Newton–Raphson method (mNR).

0264–4401/86/030235–08$2.00

Usually these incremental–iterative techniques are very expensive as much effort is required in the solution of the system of (2) for each iteration performed. This is particularly true when a repeated calculation and factorization of the tangential stiffness matrix is necessary.

In the following sections a procedure to compute and solve (2), based on the preconditioned conjugate gradient method (PCG), is presented.

Disadvantages of direct solvers for large non-linear problems

The number of operations performed in an iteration i within a load increment for the traditional Newton–Raphson method can be summarized as follows:

	operations
(1) compute residual forces $\psi(\mathbf{d}_i)$	$0(nw)$
(2) compute tangential matrix $\mathbf{K}_{Ti}$	$0(nw)$
(3) solve equation	
(i) factorize $\mathbf{K}_{Ti}$	$0(nw^2)$
(ii) back substitution	$0(nw)$
(4) line search to find s_i	$0(n)$
(5) update solution $\mathbf{d}_{i+1}=\mathbf{d}_i+s_i\,\delta\mathbf{d}_i$	$0(n)$
(6) test convergence	$0(n)$

where w is the average half-bandwidth of the stiffness matrix $\mathbf{K}_{Ti}$ and n is the number of unknowns involved in the non-linear equation. It should be noted that the above operation counts are only estimates. Substantial differences in real effort exist between, for instance, (1) and (2). There could be an order of magnitude increase in step (2) over step (1). It is important to note that the actual number of operations is problem dependent and depends on the integration rule adopted, etc.

It can be observed that in a large problem the most expensive step of the above algorithm will be the factorization of $\mathbf{K}_{Ti}$. This is the reason for the introduction of methods such as the quasi-Newton method. Instead of recalculating the refactorizing $\mathbf{K}_{Ti}$ these methods perform an update of the factorization of $\mathbf{K}_{T1}$ or $\mathbf{K}_{T2}$. A notable example is the BFGS updating procedure.

Another disadvantage of using a direct solver for (2) is that the core storage requirements for storing $\mathbf{K}_{Ti}$ increase rapidly as the size of the problem increases. The core storage needed for $\mathbf{K}_{Ti}$ is of the order $0(nw)$ where the half-bandwidth w depends heavily on the finite element mesh topology. In three-dimensional analysis, w increases more rapidly than in two dimensions as the number of unknowns increase. If a frontal solver is used, then the increase of front-width leads to an increase of in-core and out-of-core storage.

It is generally necessary to perform some pre-processing to renumber either the nodes or the elements (depending on the type of direct solver used) to reduce the band-width or the front-width. However, in many large problems, even the optimum band or front-width may still be prohibitively large, requiring too much core storage or/and computational time for a given computer installation.

Preconditioned conjugate gradient method

The previous section has highlighted the need to use alternative solution techniques when solving large non-linear problems. Preconditioned iterative methods appear as suitable candidates. Dembo *et al.*[2] recently completed a study in which the mathematical feasibility of such methods was established. In the engineering field, Omid *et al.*[3] report good results obtained by combining the Lanczos method with the traditional Newton–Raphson method.

The most promising family of iterative methods are the preconditioned conjugate gradient methods (PCG). The relative popularity of the standard CG method compared with other iterative techniques was originally based on the fact that, in exact arithmetic, the method requires a maximum of n iterations to solve a system of n equations with n unknowns. Unfortunately, it was soon discovered that when the condition number of the system matrix is high, the method is strongly influenced by roundoff error, and requires much more than n iterations to converge. The introduction of suitable preconditioning can help to eliminate this difficulty.

Preconditioning. Consider the linear problem of the form:

$$\mathbf{Kp}=\mathbf{q} \tag{3}$$

which is equivalent to the minimization of the functional:

$$\Pi=\tfrac{1}{2}\mathbf{p}^{\mathrm{T}}\mathbf{Kp}-\mathbf{p}^{\mathrm{T}}\mathbf{q} \tag{4}$$

The number of iterations required to solve (3) using the conjugate gradient method depends on C (the condition number of $\mathbf{K}$) raised to some power. For many structural analysis problems, C may be too high to obtain an economic solution. However, the use of suitable change of variable from $\mathbf{p}$ to $\bar{\mathbf{p}}$ can remedy this problem. For example, consider the transformation:

$$\bar{\mathbf{p}}=\mathbf{L}^{\mathrm{T}}\mathbf{p} \tag{5}$$

in which $\mathbf{L}$ and $\mathbf{L}^{\mathrm{T}}$ are the Choleski factors of some approximate stiffness matrix $\mathbf{Q}$ such that:

$$\mathbf{Q}=\mathbf{LL}^{\mathrm{T}} \tag{6}$$

Insertion of (5) into (3) leads to:

$$\Pi=\tfrac{1}{2}\bar{\mathbf{p}}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\mathrm{T}}\bar{\mathbf{p}}-\bar{\mathbf{p}}^{\mathrm{T}}\mathbf{L}^{-1}\bar{\mathbf{q}} \tag{7}$$

which on subsequent minimization, i.e. setting:

$$\delta\Pi/\partial\bar{\mathbf{p}}=0 \tag{8}$$

gives

$$\bar{\mathbf{K}}\bar{\mathbf{p}}=\bar{\mathbf{q}} \tag{9}$$

in which

$$\bar{\mathbf{K}}=\mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\mathrm{T}} \tag{10}$$

and

$$\bar{\mathbf{q}}=\mathbf{L}^{-1}\mathbf{q} \tag{11}$$

As noted by Crisfield[4] if $\mathbf{Q}=\mathbf{K}$ then $\bar{\mathbf{K}}=\mathbf{I}$ and the condition number $C=1$ and the iterative solution performance is excellent but there is no benefit when compared with a direct solution. Thus, the main objective should be to find a matrix $\mathbf{Q}$ which is easier to factorize than $\mathbf{K}$ but which also leads to a matrix $\bar{\mathbf{K}}$ which is better conditioned than $\mathbf{K}$.

Preconditioned conjugate gradient algorithm. *Box 2* shows the preconditioned conjugate gradient algorithm. This algorithm may, of course, be used in (step 7 of *Box 1*)

Box 2 Preconditioned conjugate gradient method for $\mathbf{Kp}=\mathbf{q}$. NS chart

1. Set iteration number $k=1$
2. As initial solution guess, set $\mathbf{p}_k=\mathbf{p}^*$
3. Evaluate pre-conditioning matrix $\mathbf{Q}$ and factorize
4. Evaluate residual forces
$$\mathbf{g}_k=\mathbf{q}-\mathbf{K}\mathbf{p}_k$$
5. Perform convergence check on norm of residual forces, displacement, energy, etc.

Has solution converged?

No:

6. Compute first part of iterative solution
$$\delta_k^*=\mathbf{Q}^{-1}\mathbf{g}_k$$
7. Evaluate CG parameter
$$\beta_k=([\delta_k^*]^{\mathrm{T}}\mathbf{g}_k)/([\delta_{k-1}^*]^{\mathrm{T}}\mathbf{g}_{k-1})$$
8. Evaluate advancing direction
$$\delta_k=\delta_k^*+\beta_k\delta_{k-1}$$
9. Update solution using line search
$$\mathbf{p}_{k+1}=\mathbf{p}_k+\eta_k\delta_k$$
$$\eta_k=([\delta_k]^{\mathrm{T}}\mathbf{g}_k)/([\delta_k]^{\mathrm{T}}\mathbf{k}\delta_k)$$
10. Set $k=k+1$ and go to 3

Yes:

11. Set $\mathbf{p}=\mathbf{p}_k$ and terminate

the incremental–iterative solution for non-linear problems. Thus,

(i) $\mathbf{p}$ is equivalent to $\delta\mathbf{d}_{i+1}^n$
(ii) $\mathbf{K}$ is equivalent to $\mathbf{K}_{\mathrm{T}i}^n$
(iii) $\mathbf{q}$ is equivalent to ψ_i^n.

An apparent advantage of the method is the saving of core storage achieved because it is no longer necessary to assemble and store the global $\mathbf{K}$ matrix. The only time $\mathbf{K}$ appears in the algorithm $\mathbf{K}$ is through the matrix–vector product in *step 4*. Therefore, $\mathbf{K}$ need not be known explicitly and the product can be performed at the element level.

Additionally the performance of the algorithm is independent of the finite element mesh topology. It is not necessary to number the nodes or elements in an optimum ordering.

Heirarchical preconditioning. The hierarchical approach[5] is regarded as a major advance in finite elements. Apart from its merits of convenience for error estimates and mesh refinement[6], it provides a natural preconditioning matrix for the tangential equation. For linear analysis, Crisfield[4] has suggested the following form of preconditioning

$$\mathbf{Q}=\begin{bmatrix}\mathbf{K}_{\mathrm{LL}} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_{\mathrm{HH}}\end{bmatrix} \tag{12}$$

where $\mathbf{K}_{\mathrm{LL}}$ is the stiffness matrix related to lower order variables and $\mathbf{D}_{\mathrm{HH}}$ is the diagonal of $\mathbf{K}_{\mathrm{HH}}$, the stiffness matrix associated with the higher order hierarchical variables.

Our numerical experiments show[8] that the use of the CG method with preconditioning (12) is very effective even for highly ill-conditioned problems. Axelsson and Gustafsson[9] have shown that for 2D problems the condition number of matrix $\mathbf{Q}^{-1}\mathbf{K}$ is of order 0(1) if $\mathbf{Q}$ given in (12) is used and the number of elements in the mesh is enough to give reasonable accuracy. This means that the required number of iterations to achieve convergence will reach a maximum as the mesh is refined[7].

For non-linear problems, the following pre-conditioning is used

$$\mathbf{Q}=\begin{bmatrix}\mathbf{K}_{\mathrm{TLL}} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_{\mathrm{THH}}\end{bmatrix} \tag{13}$$

where the subscript T implies a tangential stiffness matrix. The preconditioning matrix is updated whenever the tangential stiffness matrix is updated. In this way, the preconditioning matrix is always appropriate for the equations currently being solved. This updating technique for the preconditioner has been strongly recommended by Axelsson[10] and more recently by Nakazawa *et al.*[11].

It is worth noting that $\mathbf{K}_{\mathrm{TLL}}$ has to be factorized, and this requires the use of a direct solver. Therefore, the algorithm is partially independent on the finite element mesh topology. However, the number of variables involved is dramatically reduced.

Tolerance. In the previous sections the PCG method has been suggested as an alternative technique for performing step 7 in *Box 1*. The iterative solution the linearized equation:

$$\delta\mathbf{d}_i^n=[\mathbf{K}_{\mathrm{T}i}^n]^{-1}\psi_i^n \tag{14}$$

becomes the inner loop within the iterative procedure required for establishing equilibrium during a typical increment in the non-linear problem.

The use of PCG method with a stringent convergence tolerance can result in the exact solution (apart from round off error) of the linearized system of equations and there would be no interaction with the outer non-linear iteration loop. However, when solving the linearized equations, one should not lose sight of the primary objective: solving the non-linear problem.

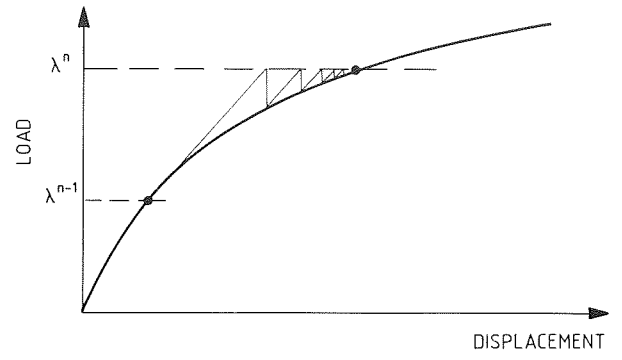
Owing to the linearization involved in writing (14) and the approximate nature of $\mathbf{K}_{\mathrm{T}}$, there is no need to reach the exact solution of the linearized system. This is illustrated in *Figure 1*. An approximation to the solution will be good enough as long as the overall algorithm is proved to be at least linearly convergent[12]. It can be proved[8] that the use of a PCG tolerance of $\varepsilon<0.5$ guarantees superlinear convergence, under certain circumstances. This allows the use of the PCG method for non-linear problems with quite slack tolerances, thus resulting in an important saving in computation time. Nour-Omid *et al.*[3] proposed the use of a tolerance of:

$$\varepsilon_a=\varepsilon_0(\|\psi^k\|/\|\psi^0\|)^q \tag{15}$$

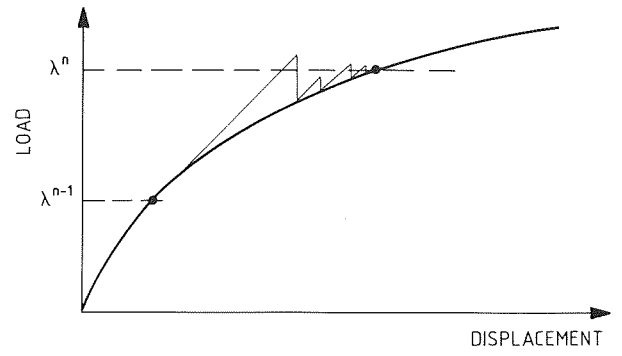
Here, the PCG tolerance is related to the convergence of the outer loop. However, our experience is that use of (15) is not necessary when solving concrete problems, where a loose tolerance (1–5%) is generally employed for the outer loop.

3D FINITE ELEMENT MODEL FOR REINFORCED CONCRETE STRUCTURES

The PCG method is now used to solve the linearized equations (see section 4 of *Box 1*) in an incremental–



(a) WITH A DIRECT SOLVER



(b) WITH AN ITERATIVE SOLVER AND A SLACK TOLERANCE

Figure 1 Approximate iterative solutions for incremental–iterative solution of non-linear equations

iterative analysis of a special class of difficult non-linear stress analysis problems: the 3D non-linear analysis of reinforced concrete structures. The sudden non-linearities in such problems (caused by cracking of the concrete as well as the smoother non-linear concrete crushing and reinforcing steel plasticity) provide a stringent test for the proposed solution algorithm. Furthermore, attention is focused on plate-type structures implying high element aspect ratios—a situation known to lead to poor conditioning in the element stiffness matrix. Poor conditioning reduces the rate of convergence of iterative methods such as the conjugate gradient method and thus this will provide an added difficulty. However, if the hierarchically preconditioned conjugate gradient method works well for this problem class, then it must be viewed as being fairly robust.

The finite element discretization and constitutive model adopted in the idealization of 3D reinforced concrete structures is now described briefly. A more detailed description is provided elsewhere[13].

Finite element model

Concrete solid. The 20 noded isoparametric element shown in *Figure 2* is used in this work. The derivation of the element stiffness matrix can be found in many finite element textbooks[14].

Hierarchical, rather than standard, interpolation functions[6] are used in the present formulation. Trilinear shape functions are used for the eight corner nodes and quadratic hierarchical shape functions are employed for the twelve mid-side nodes. The use of hierarchical shape functions provides an effective formulation for the iterative solving technique presented here. Furthermore, hierarchically derived stiffness matrices are better conditioned than the standard ones. This feature is an additional advantage when analysing plates and shells with three-dimensional elements.

To evaluate the volume integrals implicit in (2) numerical integration is used. A fifteen point integration rule[15] is employed. The location of the sampling points is shown in *Figure 3*. This particular rule is chosen because it requires less computational time than the usual $3\times3\times3$ Gauss rule, and unlike the reduced $2\times2\times2$ Gauss rule it does not produce any spurious mechanisms.

Reinforcing steel Perfect bond is assumed between the steel reinforcement and the surrounding concrete. The assumption of compatibility of displacements and strains between concrete and steel allows the reinforcement to be treated as part of the three-dimensional element. The steel stiffness is added to that of the concrete to obtain the total stiffness of the element.

Each set of reinforcing bars is smeared as a two-dimensional membrane 'layer' of equivalent thickness, placed inside the concrete element as shown in *Figure 4*. The steel is assumed to possess uniaxial material properties. The angle between the local tangent at each sampling point of the steel 'membrane' and the local coordinate system may be arbitrary. This treatment for the steel is identical to the procedure used for 'layered elements' usually employed in the analysis of reinforced plates and shells.

Constitutive laws

The constitutive laws here employed account for various types of material non-linearities in the concrete

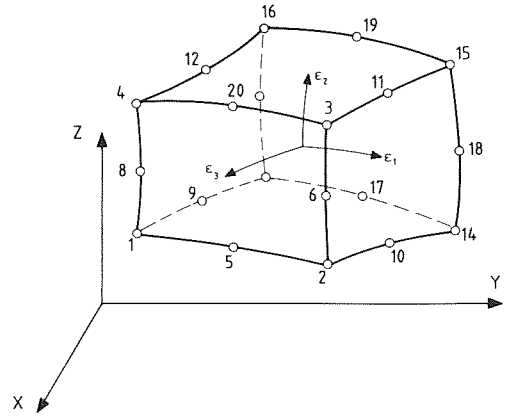


Figure 2 The 20-noded isoparametric brick element

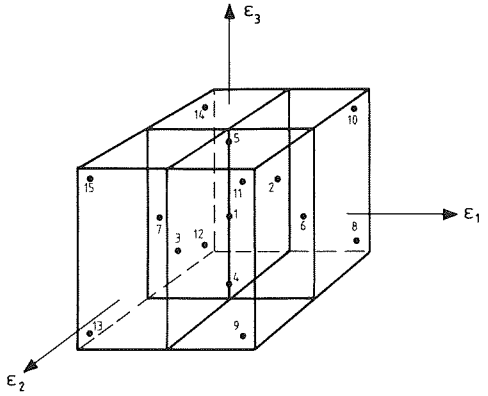


Figure 3 The 15-point integration rule for the 20-noded brick element

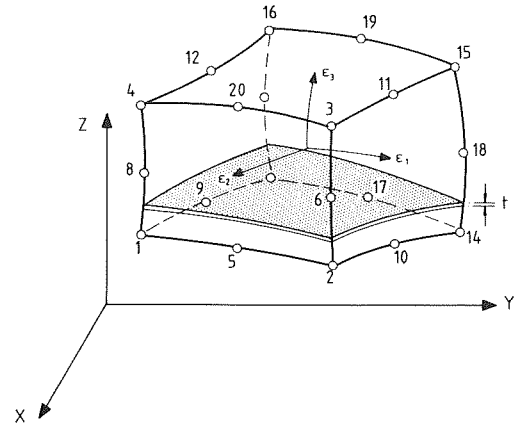


Figure 4 3D concrete element with smeared steel reinforcing layer

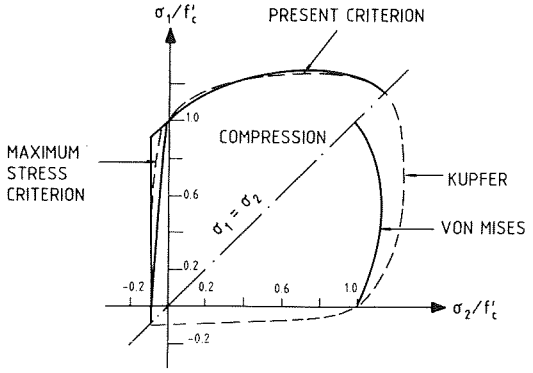


Figure 5 Compressive yield criterion for concrete (biaxial stress space)

and steel. The material behaviour is assumed to be independent of time.

Concrete model. (a) Compressive behaviour. Experimental evidence indicates that the stress–strain relationship for concrete is non-linear even for low stress levels. The inelastic deformation may be separated into recoverable and irrecoverable components. In the present model elasticity is used for the recoverable strain components, and a strain-hardening associated plasticity approach is employed to model the irrecoverable part of the deformation.

The yield function selected depends on the first two stress invariants[16] and can be written as:

$$f(I_1, J_2)=\{\alpha I_1+\beta(3J_2)\}^{1/2}=\sigma_0 \tag{16}$$

The values $\alpha=0.355\sigma_0$ and $\beta=1.355$ are suggested for an adequate fitting to experimental data[17] (see *Figure 5*).

A value of $\sigma_0=cf_c'$ (a typical value for c is 0.3) defines a surface limiting the elastic behaviour. When this surface is reached inelastic deformation begins and a hardening rule monitors the expansion of the yield surface under further loading. In this way, a whole family of 'loading surfaces' is defined. The hardening rule chosen is the conventional 'Madrid Parabola', which can be expressed as:

$$\sigma_0=-E_0\varepsilon_p+(2E_0\varepsilon_0\varepsilon_p)^{1/2} \tag{17}$$

where σ_0 is the effective stress, E_0 is the initial Young's modulus, ε_0 is the total strain at peak stress f_c' and ε_p is the plastic strain. Values $c=0.3$ and $\varepsilon_0=2f_c'/E_0$ provide good approximation to experimental test data[16] (see *Figure 6*).

Crushing of concrete is assumed to occur when a certain failure surface in the principal strain space is reached. The failure surface may be defined in a form which is similar to that of the yield surface so that:

$$\alpha I_1'+\beta(3J_2')=\varepsilon_u^2 \tag{18}$$

where I_1' and J_2' are strain invariants and ε_u is an ultimate total strain value extrapolated from uniaxial tests (typically $\varepsilon_u=0.003$–0.005).

(b) Tensile behaviour. The smeared crack approach is used to model cracking. The maximum stress criterion is employed: if the maximum principal stress exceeds a limiting value then a crack is formed in a plane orthogonal to the offending stress. Thereafter, concrete becomes orthotropic with local material axes coinciding with the principal stress directions. A maximum of two sets of cracks are allowed to open at each sampling point. The limiting value required to define the onset of cracking is established as follows,

(i) in the triaxial tension zone of the principal stress space:

$$\sigma_{i0}=f_t' \qquad i=1,\ 2,\ 3 \tag{19}$$

(ii) in the tension–tension–compression zone:

$$\sigma_{i0}=f_t'\left(1+\frac{\sigma_{i+1}}{f_c'}\right)\sigma_{i+1}\leqslant 0 \tag{20}$$

(iii) in the tension–compression–comcompression zone:

$$\sigma_{i0}=f_t'\left(1+\frac{\sigma_{i+1}}{f_c'}\right)\left(1+\frac{\sigma_{i+3}}{f_c'}\right) \qquad \sigma_{i+1},\ \sigma_{i+2}\leqslant 0 \tag{21}$$

The resulting cracking surfaces are shown in *Figure 7*.

After cracking, the tensile stress normal to the crack is released following the exponential curve:

$$\sigma=f_t'\,e^{-(\varepsilon-\varepsilon_0)/\alpha} \tag{22}$$

where f_t' is the concrete tensile strength, ε is the tensile strain across the crack, and $\varepsilon_0=f_t'/E_0$. The softening parameter α is chosen to be:

$$\alpha=G_f/l_c f_t' \tag{23}$$

where G_f is the fracture energy of concrete (a material property) and l_c is a characteristic length associated with

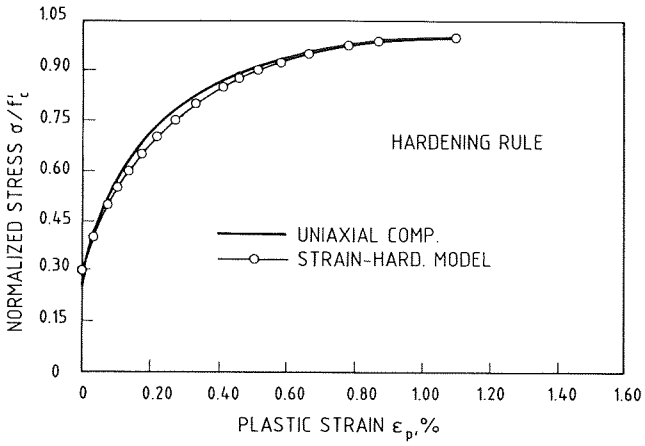


Figure 6 Comparison of hardening rule with inelastic uniaxial compressive strains of Kupfer

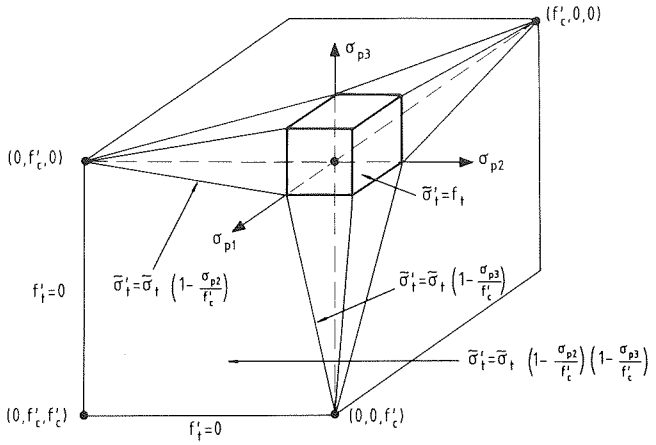


Figure 7 3D tension failure envelope for concrete model

the cracking sampling point. The dependence of the softening branch on the size of the element implies that the cracking model is mesh independent[18].

The shear modulus is also modified for cracked concrete. A shear retention factor $\beta = 0.5$ is used in this work.

Steel reinforcement model. The steel reinforcement is assumed to have uniaxial properties in the direction of the bars. A plasticity formulation is adopted in which linear isotropic hardening is employed after initial yielding. Unloading occurs elastically.

NUMERICAL EXAMPLES

In the previous sections some theoretical and computational aspects of the application of the PCG method to the analysis of reinforced concrete structures have been described. In this section two numerical examples are presented. The chosen structures are analysed with different finite element meshes and the resulting equations are solved using both a profile direct solver and a PCG iterative solver. Comparisons of CPU time and core storage needed are made. All the examples presented here were run in a VAX 11/750 under the VMS operating system, and using double precision arithmetic.

Bresler–Scordelis beam

The simply supported, reinforced concrete beam tested by Scordelis and Bresler[19], is shown in *Figure* 8. This beam has a span of 12 ft with cross-sectional dimension 21.75 in. × 9.00 in., no web reinforcement and a longitudinal reinforcement which consists of four bars with a total area of 4 in.

The beam is subjected to a central concentrated load P. The experimentally determined failure load is 58 kips, failure being caused by crushing of the concrete.

By taking advantage of symmetry the beam was analysed using two different meshes:

mesh (a) consists of 60 elements (5 along the span, 4 through the thickness and 3 in the width).
Total number of unknowns: 1153
Number of low order unknowns: 332

mesh (b) consists of 96 elements (8 × 4 × 3).
Total number of unknowns: 1792
Number of low order unknowns: 512

Material properties used in the analysis are:

elastic modulus of concrete,	$E_c =$	3,300 kips/in^2
elastic modulus of steel,	$E_s =$	27,800 kips/in^2
compressive strength of concrete,	$f'_c =$	3.16 kips/in^2
tensile strength of concrete,	$f'_t =$	0.33 kips/in^2
concrete fracture energy,	$G_f =$	0.0004 kips/in
yield stress of steel,	$f_y =$	60 kips/in^2
hardening parameter of steel,	$H' =$	0 kips/in^2
concrete Poisson's ratio,	$\nu =$	0.2
ultimate concrete compr. strain,	$\varepsilon_u =$	0.002

The non-linear response is traced using KT1 method, that is, a modified Newton–Raphson in which updating of the tangential stiffness matrix is only performed for the first iteration of each increment. Constant load increments (equal to half the cracking load) are applied throughout the analysis. A tolerance of 5% in the residual forces norm is chosen for the outer non-linear loop. A slack tolerance (=20%) is used for the PCG iterations.

Results comparing the performance of PCG against that of the direct solver (profile solver) are shown in *Table 1*.

It can be seen from these results that the PCG method is 2.33 times faster than the direct solver for the first mesh, and 3.56 times faster for the second. In fact, the total time for the PCG method with mesh (b) is identical to the total time for the use of the profile solver for mesh (a). The comparative efficiency of the PCG method over the direct method increases with the size of the problem. The time spent in equation solving increases less than linearly with the number of unknowns for the PCG method while it increases dramatically for the direct solver.

It is also worth noting that the length of the profile needed to store **K** (the only global array needed to be formed explicitly for the PCG method) is only about 10% of that needed for the whole of **K**.

The average number of outer non-linear iterations performed per increment was 3. A representative number of inner iterations to achieve the desired convergence with the iterative solver is 11.

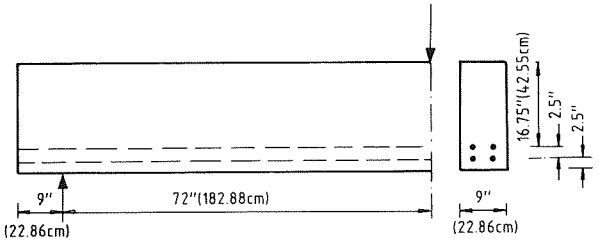


Figure 8 Bresler–Scordelis beam

Table 1 Comparison of CPU times and storage needs for Example 1—the reinforced concrete beam

	Mesh (a)		Mesh (b)	
	PCG	Profile	PCG	Profile
Total time (sec)	14,303	17,847	17,847	28,037
Eqn. solution time (sec)	3,215	7,509	3,796	13,527
Profile length	20,483	213,175	33,776	349,507

In *Figure 9* the experimental load–deflection curve at the midspan of the beam is compared with the predictions resulting from the present analysis. Results using the PCG and direct solution methods are identical in terms of accuracy.

Duddeck's slab

In this second example a corner supported slab tested by Duddeck *et al.*[20] is studied. Only transverse deflections are restrained at the corner supports.

The information on material properties is rather scarce. The values assumed by Figueiras[16] are again used here:

elastic modulus of concrete,	$E_c =$	16,400 N/mm^2
elastic modulus of steel,	$E_s =$	201,800 N/mm^2
compressive strength of concrete,	$f_c' =$	43 N/mm^2
tensile strength of concrete,	$f_t' =$	3 N/mm^2
concrete fracture energy,	$G_f =$	0.2 N/mm
yield stress of steel,	$f_y =$	670 N/mm^2
hardening parameter of steel,	$H' =$	0 N/mm^2
concrete Poisson's ratio,	$\nu =$	0
ultimate concrete compr. strain,	$\varepsilon_u =$	0.0035

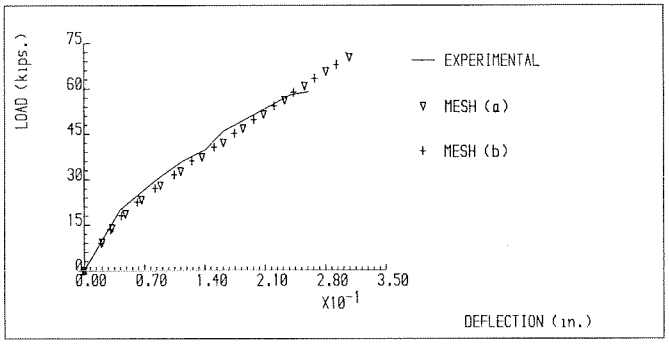


Figure 9 Load–displacement curve for Bresler–Scordelis beam

The slab dimensions and reinforcement details are shown in *Figure 10*. The thickness of the reinforcement is 0.193 mm for the top layers and 0.397 mm for the bottom layers. The slab was tested under a central concentrated load.

In the finite element analysis, by taking advantage of symmetry, it is only necessary to consider one quarter of the slab. Three regular meshes of equal finite elements are employed:

mesh (a)	consists of $3 \times 3 \times 1$ elements.	
	Total number of unknowns:	251
	Number of low order unknowns:	79
mesh (b)	consists of $6 \times 6 \times 1$ elements.	
mesh (b)	consists of $6 \times 6 \times 1$ elements.	
	Total number of unknowns:	878
	Number of low order unknowns:	265
mesh (c)	consists of $10 \times 10 \times 1$ elements.	
	Total number of unknowns:	2302
	Number of low order unknowns:	681

KT1 method is used again for the analysis, with load increments equal to half of the cracking load. The tolerance for the outer loop is 0.05 and the tolerance for the inner loop is set to 0.2.

Table 2 compares results for the iterative and direct solutions

This second example corroborates the conclusions drawn from the first one. The comparative efficiency of

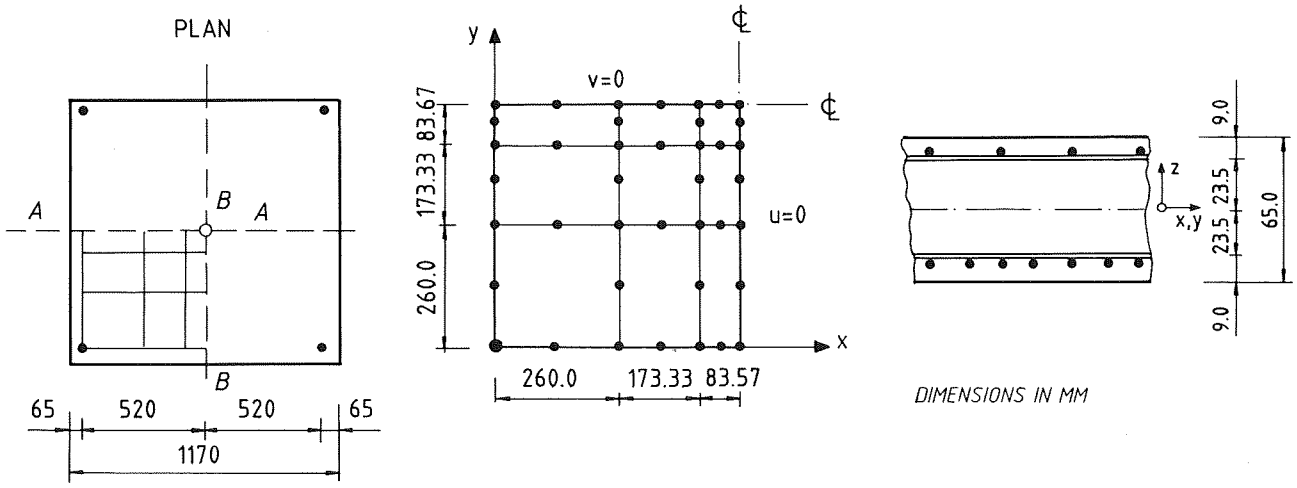


Figure 10 Duddeck's corner supported, reinforced concrete slab

Table 2 Comparison of CPU times and storage needs for Example 2—the reinforced concrete slab

	Mesh (a)		Mesh (b)		Mesh (c)	
	PCG	Profile	PCG	Profile	PCG	Profile
Total time (sec)	4,037	4,283	17,478	20,330	45,413	62,327
Equation solution time	636	234	2,164	3,453	7,615	21,595
Profile length	1,947	17,238	12,645	119,997	53,089	522,253

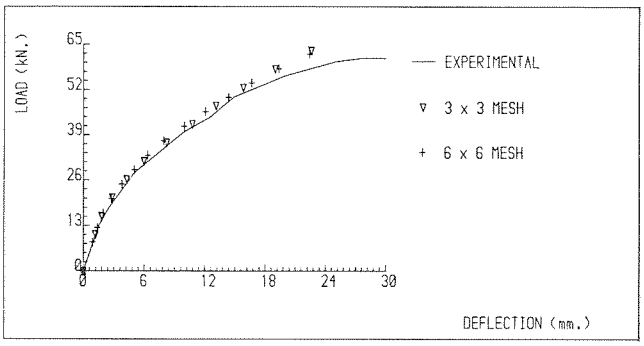


Figure 11 Load–displacement curve for Duddeck's slab

PCG and direct solvers depends heavily on the size of the problem: for mesh (a) the direct solver is almost three times faster than the iterative, but for mesh (c) the same ratio of speed is achieved in favour of the iterative solver. It is also interesting to note that for mesh (a), even though the direct solver is faster than the PCG solver, the total CPU time spent is less when the PCG method is employed. This is because convergence in the outer loop seems to be faster when an iterative solver with a slack tolerance is used to solve the iterative equilibrium equations, thus reducing the number of necessary non-linear iterations to be performed. Again the storage required for the PCG method is approximarely 10% of that required for the direct method.

For mesh (c) the number of inner iterations varies from 12 in the first increments to about 20 in the last increments of the analysis. This is caused by the progressive degradation of the stiffness due to cracking, and a corresponding increase in the condition number of the system of equations.

In *Figure 11* load–deflection curves for the three meshes are compared against experimental results. Excellent agreement is found through the whole loading process. Also the three meshes provide completely consistent results.

CONCLUSIONS

The paper shows the hierarchically preconditioned conjugate gradient method to be a suitable alternative to direct solvers for three-dimensional finite element analysis. The method is successfully used for the non-linear analysis of three-dimensional reinforced concrete structures. Numerical examples prove the efficiency of the method compared with a direct solver based on a profile storage scheme. Important savings are achieved both in storage requirements and in computing time.

REFERENCES

1 Owen, D. R. J. and Hinton, E. *Finite elements in plasticity: theory and practice*, Pineridge Press, Swansea (1984)
2 Dembo, R. S., Eisenstate, S. C. and Steihang, T. Inexact Newton methods, *SIAM J. Num. Anal.* **19**, 2 (1982)
3 Nour-Omid, B. A Newton–Lanczos method for solution of nonlinear finite element equations, *Report UCB/SESM-81/04*, University of California (1981)
4 Crisfield, M. A. New solution procedures for linear and nonlinear finite element analysis, *Transport and Road Research Lab. Report* (1983)
5 Zienkiewicz, O. C., Irons, B. M., Scott, F. C. and Campbell, J. 'Three-dimensional stress analysis, *Proc. IUTAM Symp. on High Speed Computing of Elastic Structures*, University of Liege Press (1971)
6 Zienkiewicz, O. C., Gago, J. P. S. R. and Kelly, D. W. The hierarchical concept in finite element analysis, *Comput. Struct.* **16**, (1–4) (1983)
7 Crisfield, M. A. Some recent research on numerical techniques for structural analysis, *NUMETA 85*, A. A. Balkema, Rotterdam, pp. 565–575 (1985)
8 Liu, Y. C. Iterative methods and finite elements, *PhD Thesis*, University of Wales (1985)
9 Axelsson, O. and Gustafsson, I. Preconditioning and two-level multigrid methods for arbitrary degree of approximation, *Math. Comp.* **40** (1983)
10 Axelsson, O. Solution of linear systems of equations: iterative methods, *Sparse Matrix Techniques*, Lecture notes in Mathematics, Vol. 572, Springer-Verlag, Berlin (1976)
11 Nakazawa, S., Nagtegaal, J. C. and Zienkiewicz, O. C. Iterative methods for mixed finite element equations, *Int. J. Num. Meth. Eng.* (1985) in press.
12 Dennis, J. R. and More, J. J. Quasi-Newton methods: motivation and theory, *SIAM Rev.* **19**, 46–86 (1977)
13 Cervera, M. and Hinton, E. Nonlinear, analysis of reinforced concrete plates and shells using a three dimensional model, *Research Report, University College of Swansea* (1985)
14 Hinton, E. and Owen, D. R. J. *An Introduction to finite element computations*, Pineridge Press, Swansea (1979)
15 Irons, B. M. Quadrature rules for brick based finite elements, *Int. J. Num. Meth. Eng.* **3**, 293–294 (1971)
16 Owen, D. R. J. and Figueiras, J. A. Ultimate load analysis of reinforced concrete plates and shells including geometric nonlinear effects, in *Finite element software for plates and shells*, Pineridge Press, Swansea (1984)
17 Kupfer, H., Hilsdorf, K. H. and Rush, H. Behaviour of concrete under biaxial stress, *ACI Proc.* **66**, 656–666 (1969)
18 Glemberg, R. Dynamic analysis of concrete structures, *Pub. 84:1*, Dept. of Structural Mechanics, Chalmers University of Technology (1984)
19 Franklin, H. A. Nonlinear analysis of reinforced concrete frames and panels, *Report No. SESM 70-5*, Dept. of Civil Eng., University of California, Berkeley (1970)
20 Duddeck, H., Griebenon, G. and Schaper, G. Material and time dependent non-linear behaviour of cracked concrete slabs, in *Nonlinear behaviour of reinforced concrete spatial structures, Vol.* 1. Preliminary Report, IASS Symp., Darmstadt (1978)