Computers and Concrete, Vol. 2, No. 5 (2005) 345-366
DOI: http://dx.doi.org/10.12989/cac.2005.2.5.345

Assessment of computational performance for a vector parallel implementation: 3D probabilistic model discrete cracking in concrete

Carmen N. M. Paz†, José L. D. Alves‡ and Nelson F. F. Ebecken‡†

LAMCE/COPPE/UFRJ, Program of Civil Engineering, PO Box 68552, 21944-900, Brazil
(Received December 23, 2004, Accepted September 22, 2005)

Abstract. This work presents an assessment of the computational performance of a vector-parallel implementation of probabilistic model for concrete cracking in 3D. This paper shows the continuing efforts towards code optimization as reported in earlier works Paz, *et al.* (2002a,b and 2003). The probabilistic crack approach is based on the direct Monte Carlo method. Cracking is accounted by means of 3D interface elements. This approach considers that all nonlinearities are restricted to interface elements modeling cracks. The heterogeneity governs the overall cracking behavior and related size effects on concrete fracture. Computational kernels in the implementation are the inexact Newton iterative driver to solve the non-linear problem and a preconditioned conjugate gradient (PCG) driver to solve linearized equations, using an element by element (EBE) strategy to compute matrix-vector products. In particular the paper analyzes code behavior using OpenMP directives in parallel vector processors (PVP), such as the CRAY SV1 and CRAY T94. The impact of the memory architecture on code performance, and also some strategies devised to circumvent this issue are addressed by numerical experiment.

Keywords: high performance computing; openMP directives; vector-parallel performance analysis; non linear analysis; probabilistic discrete cracking in concrete; 3D finite elements; size effects; heterogeneity material.

1. Introduction

Many researchers made significant contributions about fracture mechanics of concrete and size effects in the last decades: Bazant (1976), Hillerborg (1976), Peterson (1981), Carpinteri (1986), Elices, *et al*. (1989), Planas, *et al*. (1989), Hu (1991), Mihashi, *et al*. (1991), Tschegg, *et al*. (1992), Guinea, *et al*. (1994), Wittmann (1995). The advance of the computational technique made possible the simulation of heterogeneous materials as found in: Carpinteri, *et al*. (1994, 2003), Bazant (1997, 2002); Rossi, *et al* (1994), Fairbairn, *et al* (1999), Paz, *et al*. (2002a,b and 2003), Van Mier, *et al*. (2003). The probabilistic crack approach, based on the direct Monte Carlo method was developed by Rossi and co-workers (1987), in a 2D framework.

This work presents the continuing efforts towards code optimization as reported in earlier works by Paz, *et al.* (2002a,b and 2003). A probabilistic crack approach, based on the Monte Carlo method, is fully parallelized in a 3D finite element code. The cracking scheme used is discrete crack in nature,

† Corresponding Author, E-mail: mena@lamce.ufrj.br
‡ E-mail: jalves@lamce.ufrj.br
‡† E-mail: nelson@ntt.ufrj.br

i.e., it is by of 3D interface elements. In this approach, the heterogeneity of the material is taken into account by considering the properties to vary spatially according to a normal distribution.

The optimized implementation of probabilistic model for the simulation of cracking in concrete structures was presented. This model is based on the assumption that some particularities of the cracking behavior of concrete, such as strain softening, cracking evolution and size-effects are derived from the heterogeneous characteristics of the material.

The probabilistic methodology presented in this paper corresponds to the 3D analysis of a strongly nonlinear material that develops cracking. In addition, the finite elements analysis must be called several times within a Monte Carlo simulation. The examples presented in this paper show that the model is capable of simulating the crack opening and the crack pattern.

Performance analyzes were carried out for the two implementations (original and modified version of the code), both in the CRAY SV1 and CRAY T94. The computational performance will be made in uniaxial tension test (dog-bone shaped). In this example the size/scale effects on fracture of concrete will also be studied.

This work presents FPST (four-point shear test with one notch) a large example which was only possible to analyze with the modified version, as described herein, of the original code. In this example the behavior of the problem and parallel-vector performance were analyzed.

Due to the complexity of the example (FPST) that involves large number of interface elements, non-linearity behavior of the problem is increasing. It was not possible to obtain the performance results for a single CPU for the original version of the code. The job time with original code exceeded a practical limit on CRAY SV1 in single CPU.

The remaining of this work is organized as follows: in sections 2 and 3 the work briefly describes respectively the probabilistic model and the 3D interface element; section 4 describes the computational strategies employed; section 5 addresses the numerical examples used in the performance experiment; section 6 presents computational performance results; Section 7 closes presenting some conclusions drawn from the computational experiments.

2. Probabilistic modeling of concrete cracking

The heterogeneity governs the overall cracking behavior and related size effects on concrete fracture. The probabilistic crack approach, based on the direct Monte Carlo method and developed by Rossi, *et al.* (1987, 1994 and 1997), in a 2D framework, takes this stochastic process into account by assigning in the finite element model, randomly distributed material properties, such as tensile strength (f_{ct}) and Young's modulus (E_c) to both solid and contact elements interfacing the former, that is, a discrete crack approach. This approach considers that all nonlinearities are restricted to interface elements modeling cracks. Therefore, the stochastic process is introduced at the local scale of the material, by considering that cracks are created within the concrete with different energy dissipation depending on the spatial distribution of constituents and initial defects. The local behavior is assumed to obey a perfect elastic brittle material law. Thus, the random distribution of local cracking energies can be replaced by a random distribution of local strengths.

The present probabilistic model involves a number of mechanical properties of the material to be determined, which constitutes the modeling data. From a large number of uniaxial tensile tests, it was found that a normal law describes rather well the experimental distribution of the relevant material data (Rossi, *et al.* 1994). These characteristics are the means of the tensile strength ($f_{ct,\mu}$)

and of the Young's modulus (E_μ); the standard deviations of the tensile strength ($f_{ct,\sigma}$) and of the Young's modulus (E_σ). The following analytical expressions were proposed:

$$f_{ct,\,\mu} = 6.5 \cdot C \cdot (V_T/V_g)^{-a}; \qquad f_{ct,\sigma} = 0.35 \cdot f_{ct,\mu}(V_T/V_g)^{-b} \tag{1}$$

$$E_\mu = E\,; \qquad E_\sigma/E = 0.15(V_t/V_g)^{-c} \tag{2}$$

where V_T is the volume of the two finite elements contiguous to an individual contact element of the mesh; V_g is the volume of the coarsest aggregate; E is the mean value of the Young's modulus which does not exhibit significant volume effects; and *a, b, c* constants related to cylinder compressive strength f_c

$$a = 0.25 - 3.6 \cdot 10^{-3}(f_c/C) + 1.3 \cdot 10^{-5}(f_c/C)^2 \tag{3}$$

$$b = 4.5 \cdot 10^{-2} + 4.5 \cdot 10^{-3}(f_c/C) + 1.8 \cdot 10^{-5}(f_c/C)^2 \tag{4}$$

$$c = 0.116 + 2.7 \cdot 10^{-3}(f_c/C) - 3.4 \cdot 10^{-6}(f_c/C)^2 \tag{5}$$

with C=1 MPa. In these expressions, the compressive strength f_c represents the quality of the concrete matrix, while the volume of the coarsest aggregate V_g, refers to the elementary material heterogeneity.

Fig. 1 shows, in a double logarithmic space, the tensile strength domain as a function of the volume ratio V_t/V_g for a concrete of compressive strength f_c=60 MPa, calculated from Eq. (1). The figure shows that the smaller the scale of observation (respectively the modeling scale), the larger the fluctuation of the local mechanical properties, and thus the (modeled) heterogeneity of the matter. In other words, the finer the mesh, the greater the modeled heterogeneity in terms of Young's modulus and tensile strength. In this way, the probabilistic approach aims at modeling size

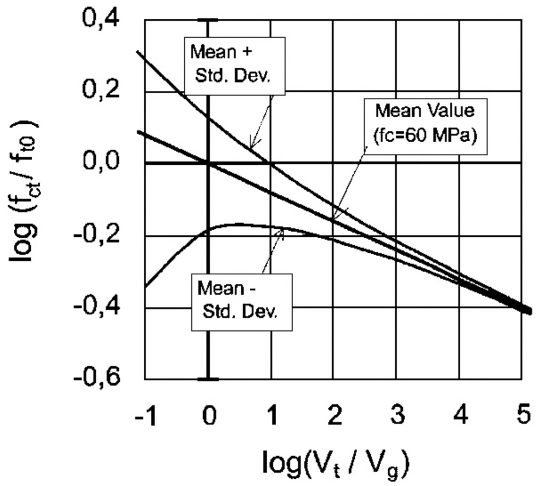


Fig. 1 The tensile strength domain of concrete according to the probabilistic crack approach, Rossi, *et al.* (1987)

effects by bridging the gap between the strength theories and linear fracture mechanics: for small volume-ratios, the strength criteria are enriched by the randomness of material strength (Weilbull 1939); for larger sizes, both the mean value and the standard deviation decrease, i.e., the Weibull-type statistical size effects decrease. This is in agreement with Bazant's (1997) asymptotic analysis of quasi-brittle fracture. Provided that the elementary material volume (i.e. volume of the solid finite element V_T) remains small with respect to the structural volume, the numerical results are mesh independent (Rossi, *et al.* 1996).

The mesh has v volume elements (tetrahedral element) and i interface elements. Each interface element follows an elastic-brittle constitutive law characterized by an individual tensile strength $f_{ct,\,i}$. The volume elements are elastic and have individual Young's modulus referenced by E_v.

Following Rossi, *et al.* (1994) the individual local tensile strengths and Young's modulus are represented by normal distributions having the densities:

$$g_f(f_{ct}) = \frac{1}{f_{ct,\sigma}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{f_{ct,\mu}}{f_{ct,\sigma}}\right)^2\right] \tag{6}$$

$$g_E(E) = \frac{1}{E_\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{E-E_\mu}{E_\sigma}\right)^2\right] \tag{7}$$

where $g_f(f_{ct})$ and $g_E(E)$ are density functions for the tensile strength f_{ct} and the Young's modulus E, respectively; while x_μ and x_σ denote the mean and standard deviation of the distribution of quantity x. For the problem at hand, it is possible to find a sample of i values $f_{ct,i}$, each value corresponding to an interface element i, and v values E_v, each value corresponding to a volume element v, by using a standard routine for generation of random numbers for a given normal distribution (Press, *et al*. 1992).

The solution for this probabilistic approach is obtained by means of a Monte Carlo simulation. A number of n samples are generated for a given normal distribution and some characteristic responses of the structure are computed; for example, stress crack-width $\sigma - w$ curve or load displacement P-δ curve. This direct Monte Carlo procedure is sketched in Fig 2. Let the jth sample correspond to the jth P-δ curve. This jth P-δ curve is composed of discrete values, P_k^j and δ_k^j,

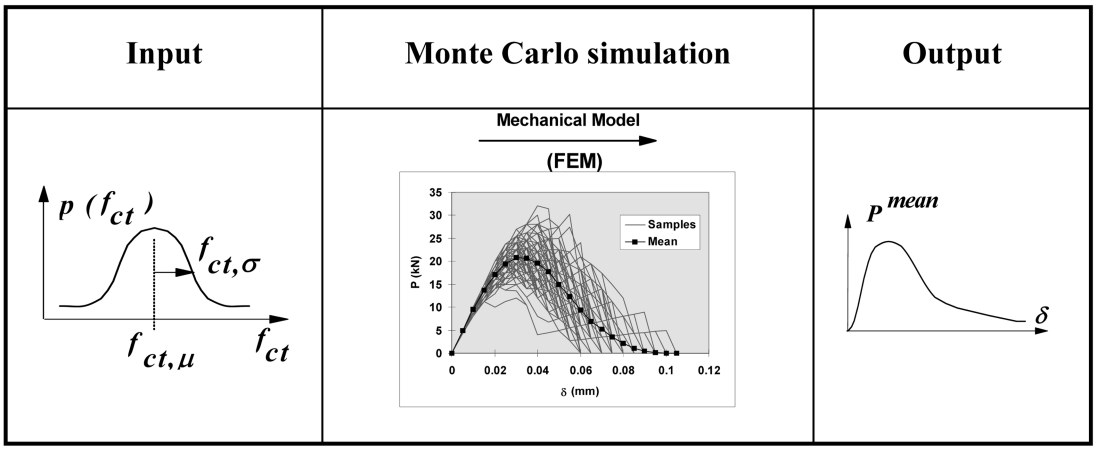


Fig. 2 Monte Carlo simulation

where the superscript j indicates the sample and the subscript k the discrete value of the P-δ curve. The same number of discrete δ-value is assumed. The mean curve composed by pairs (P_k^{mean}, δ_k) then simply reads

$$P_k^{mean,j} = \frac{1}{j}\sum_{l=1}^{j} P_k^l \tag{8}$$

The Monte Carlo simulation is stopped when

$$\left| P_k^{mean} - P_k^{mean,j-1} \right| \leq tol \tag{9}$$

where *tol* is the prescribed tolerance to check the convergence of the procedure. With the convergence of the procedure, the total number of samples is set to $n=j$. This total number of samples n, corresponding to a Monte Carlo converged simulation, clearly depends on *tol*, which is a measure of the precision required by the analysis. It also depends on the heterogeneity of the material represented by the standard deviation. The more heterogeneous the material is the greater is the number of samples to obtain a converged solution. Experience indicates that 15 to 30 samples are sufficient to obtain a converged P-δ curve.

For a given normal distribution characterized by the mean and standard deviation of the tensile strength, the described direct Monte Carlo procedure delivers a P-δ mean curve which is characterized by the P_k^j and δ_k^j values.

3. Three-dimensional interface elements for modeling discrete cracking

The finite element cracking model is a discrete model for which volume elements are assumed elastic and cracking only occurs in elastic-brittle interface elements (Fig. 3b) placed between two neighboring surfaces of the volume elements. The three dimension interface elements depicted in Fig. 3(a) can be thought as triangular base prisms connecting adjacent faces of neighboring

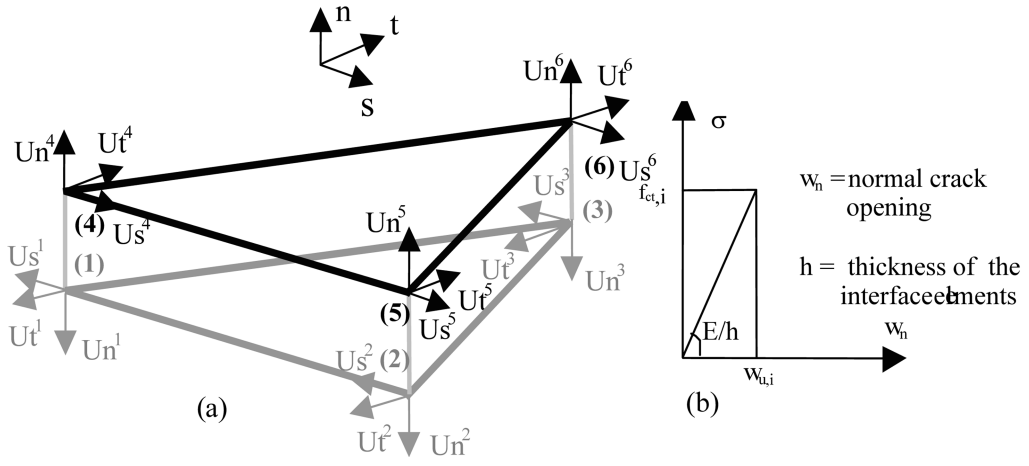


Fig. 3 (a) An interface element and its degrees of freedom in a local system and (b) elastic-brittle contact law

tetrahedra. These elements are formulated to represent relative displacements between the triangular faces to simulate crack opening. The formulation of the interface element was developed by Paz, *et al*.

The constitutive law of the 3D interface element is defined by Eq. (10) for non cracked elastic state characterized by $\sigma_n < f_{ct,i}$. When the tensile strength is reached the elements attains the cracked stage and modulus E_c and G_c are set to zero (Fig. 3b).

$$\Delta\boldsymbol{\sigma} = \mathbf{D}_{cr}\Delta\mathbf{w} = \begin{Bmatrix} \Delta\sigma_n \\ \Delta\sigma_s \\ \Delta\sigma_t \end{Bmatrix} = \begin{bmatrix} E_c/h & 0 & 0 \\ 0 & G_c/h & 0 \\ 0 & 0 & G_c/h \end{bmatrix} \begin{Bmatrix} \Delta w_n \\ \Delta w_s \\ \Delta w_t \end{Bmatrix} \tag{10}$$

In Eq. (10) the subscript n stands for normal, while s and t indicate tangential direction respective to the crack plane, w are the relative displacements between the two faces of the interface element, h is the width of interface element, E_c and G_c are respectively the normal (Young's) and the shear modulus along crack plane.

Eqs. (10) and Fig. 3(b) define the elastic-brittle constitutive behavior. The thickness h plays the role of a penalization parameter and should be conveniently chosen not to affect the solution.

In order to select the element we define the ratio ξ^e;

$$\xi^e = \frac{\sigma^e_n}{f^e_{ct}} \tag{11}$$

where the superscript e indicates an interface element; σ^e_n is the normal stress to the crack plane, and f^e_{ct} tensile strength for interface element. At each iteration only the element presenting the largest $\xi^e > 1$ is allowed to develop a crack, i.e., E_c=G_c=0. The remaining interface elements for which $\xi^e > 1$ are kept with the last non zero E_c and G_c. This approach renders a robust equilibrium path until reaching the limit load.

The kinematic relation for the interface element is;

$$\Delta\mathbf{w} = \mathbf{B}\Delta\mathbf{a}^e_l \tag{12}$$

where

$$\mathbf{B} = \frac{1}{3}\begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$\Delta\mathbf{w}$ is the crack opening incremental and $\Delta\mathbf{a}^e_1$ is the vector incremental nodal displacements for the interface element.

Applying a standard displacement based F. E. formulation, the resulting tangent stiffness matrix for the interface element is given is:

$$\mathbf{K}^e_{Intf} = \int_\Omega \mathbf{B}^T\mathbf{D}_{cr}\boldsymbol{B}d\Omega \tag{14}$$

The interface elements are generated contiguous to the faces of selected tetrahedra elements. This selection is performed by the user, defining a 3D box inside the mesh that contains the target elements.

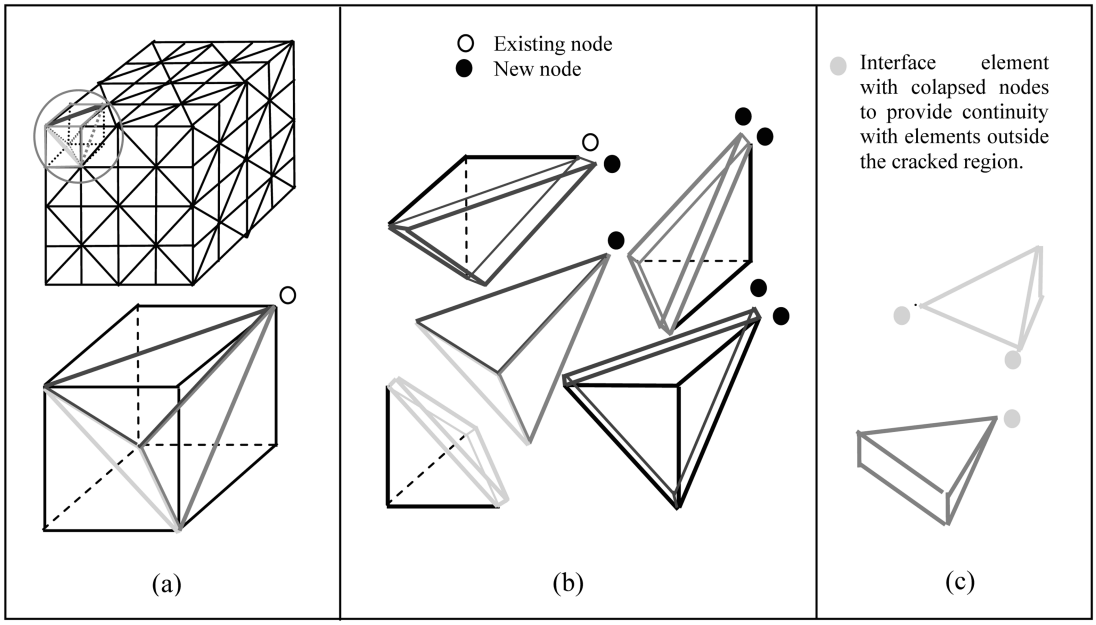


Fig. 4 Interface element mesh generation

3.1. Three-dimensional interface elements mesh generation

Fracturing is modeled by 3D interface elements generated in a previously defined cracked region. This selection is performed by the user defining a 3D box inside the mesh that contains the target elements. Once having a 3D mesh (either structured or not) of tetrahedra, the cracking region is defined by a bounding box. The interface elements then are generated contiguous to the faces of the tetrahedra within the cracking region (developed by Paz, *et al.*)

The procedure initially establishes the neighborhoods of the faces, then it maps how many elements share each node. Later it creates the nodes necessary to the interface elements, all nodes having the same coordinates of the neighboring node. After this all nodes with the same coordinates are visited, and the first element of the loop takes the existing node of the initial mesh of tetrahedra (see Fig. 4a) and, for the other elements that share this node, a new node numbering (see Fig. 4b) is introduced. Later connectivities are created, introducing the interface elements according to an ordering previously established.

Interface elements with collapsed nodes provide continuity to the elements outside the cracked region (see Fig. 4c). These elements are implemented using an artifice that allows the use of elements with six nodes. This artifice consists of multiple references for the same collapsed nodes duplicating the node numbering for the elements outside the 3D box.

4. Computational strategies

In this work we employ the Inexact Newton Method (Kelley 1995) to solve the resulting

Given u_{tol}, r_{tol}, η_i relative and residual tolerance.

Compute stiffness tetrahedra matrix $\mathbf{K}_{Tetra}$

do k = 1,2….., number of load increments do

Compute external forces vector

$$\mathbf{F}_{ext}^{k} = \mathbf{F}_{nodal}^{K} + \mathbf{F}_{volume} + \mathbf{F}_{\bar{\sigma}} - \left(\mathbf{K}_{tetra}\, \bar{\mathbf{U}}_k + \mathbf{K}_{Intef}\, \bar{\mathbf{U}}_k \right)$$

do i = 1,2 ..., while convergence

Compute internal forces vector,

$$\mathbf{F}_{\text{int}}^{i} = \left(\mathbf{F}_{\text{int}}^{i} \right)_{Tetra} + \left(\mathbf{F}_{\text{int}}^{i} \right)_{Intf}$$

Compute residual vector,

$$\boldsymbol{\psi}^{i} = \mathbf{F}_{\text{int}}^{i} - \mathbf{F}_{ext}^{i}$$

Update stiffness interface matrix $\mathbf{K}_{\text{Intf}}^{\text{i}}$

$$\mathbf{A}^{i} = \mathbf{K}_{Tetra} + \mathbf{K}_{Intf}^{i}$$

Compute tolerance for iterative driver, η_i

Solver: $\mathbf{A}^{i}\ \Delta\mathbf{u} = \boldsymbol{\psi}^{i}$ for tolerance η_i

Update solution,

$\mathbf{U} = \mathbf{U} + \Delta\mathbf{u}$

if $\dfrac{\| \Delta\mathbf{u} \|}{\| \mathbf{U} \|} \le utol$ and $\dfrac{\| \Delta\boldsymbol{\psi}^{i} \|}{\| \mathbf{F}_{ext}^{k} \|} \le rtol$ then convergence

end while i.

end do k.

Box 1 Inexact Newton algorithm

nonlinear set of equations at each load or displacement increment. In the Inexact Newton Method, at each nonlinear iteration, a linear system of finite element equations is approximately solved by the preconditioned (nodal block diagonal) conjugate gradient method (PCG). The outline of the nonlinear solution algorithm is presented in Box 1.

Note that in $\mathbf{F}_{ext}^{k}$ we account for nodal forces, body forces and prescribed displacements and stresses $\bar{\mathbf{U}}, \bar{\boldsymbol{\sigma}}$. The total internal forces vector $\mathbf{F}_{int}^{i}$ is the sum of the tetrahedra element vector internal forces $(\mathbf{F}_{int}^{i})_{Tetra}$ plus the interface element internal forces vector $(\mathbf{F}_{int}^{i})_{Intf}$.

The computational kernel for a PCG driver is the matrix-vector products **Ap** where **A** is global stiffness and **p** is the vector of residuals (Kelley 1995). In our implementation, following the EBE approach of Hughes (1987), the global matrix **A** is never assembled when used as a linear operator. Taking each element contribution to this product, considering both solid and interface elements, the products **Ap** can be recast as follows:

$$\mathbf{Ap} = \sum_{i=1}^{N_{tetra}} (\mathbf{K}_{Tetra}\, \mathbf{p}_i) + \sum_{j=1}^{N_{Intf}} (\mathbf{K}_{Intf}\, \mathbf{p}_j) \tag{15}$$

where, N_{tetra} is the number of tetrahedra, N_{intf} is the number of interface elements, $\mathbf{K}_{Tetra}$ and $\mathbf{K}_{Intf}$ are respectively element matrices for the tetrahedra and interface; $\mathbf{p}_i$ and $\mathbf{p}_j$ are the components of $\mathbf{p}$ restricted to the degrees of freedom of each element type.

Stiffness matrices for tetrahedra are computed and stored at the beginning of the analysis since they are elastic, while the stiffness matrices for interface elements are updated at every nonlinear iteration. These arrays are stored accounting for the symmetric and rank deficiency of the unassembled element matrices yielding 78 stored coefficients for tetrahedral and for the interface element only 18 coefficients are stored, exploring the particular structure of the discrete gradient operator. The mesh coloring algorithm of Hughes (1987) was extended to block both solid and interface elements into disjoint groups thus enabling full vectorization and parallelization of the operations involved in Eq. (15). An approximate solution is obtained when the Inexact Newton termination criterion is satisfied, that is.

$$\left\| \mathbf{A}^i \Delta\mathbf{u} - \Psi^i \right\| \le \eta_i \left\| \Psi^i \right\| \tag{16}$$

Tolerance η_i may be kept fixed throughout the nonlinear iterations or may be adaptively selected as the iterations progress towards the solution. Our choice for η_i follows the criteria suggested by Kelley (1995).

Several preconditioner options are available ranging from the nodal block diagonal up to incomplete factorizations. In this work, we restricted to nodal block diagonal.

According to the above algorithm, an approximate solution is obtained when the Inexact Newton termination criterion is satisfied, that is, when,

The selection for η_i follows Kelley (1995), based on a measure of how far the nonlinear iteration is from the solution, that is,

$$\eta_i^A = \min\left(\eta_{\max}, \gamma \frac{\left\| \psi^i \right\|^2}{\left\| \psi^o \right\|^2} \right), \qquad 0 < \gamma < 1 \tag{17}$$

And finally $\eta_i = \max(\eta_{\min}, \eta_i^A)$ Kelley (1995) has shown general convergence properties when Eq. (17) is used.

Our experience indicates that selecting $\eta_{\max}$, =0.1 and $10^{-3} \le \eta_{\min} \le 10^{-6}$ for *utol* and *rtol* in the usual range, that is, 10^{-3} tol, is enough for practical engineering computations.

4.1. Code parallelization the implementation made extensive use of the OpenMP directives

The !$OMP PARALLEL directive creates (i.e. opens) the parallel region and the !$OMP END PARALLEL directive destroys (i.e. closes) the parallel region. Each variable inside this parallel region may have one of the following three basic attributes: SHARED single storage location in memory for the duration of the parallel construct, PRIVATE multiple storage locations in memory for duration of the parallel construct and REDUCTION for both private and shared storage behavior.

In the modified code, critical regions were restructured in order to circumvent the bottle-necks that inhibit parallelism, and in those cases using CRAY libraries.

In order to illustrate the modifications introduced, the pseudo-code fragment of internal forces evaluation for interface element is listed in Box 2. Our experience indicates that to increase

```
subroutine (Fint)Intf
Dimension Fintf(18,nel_intf)
!$OMP PARALLEL DO PRIVATE (iel)
!$OMP+ DEFAULT (PRIVATE)
!$OMP+ SHARED (fintf,plas,All_stress,nel_intf,......)
cdir$ ivdep
      do iel = 1, nel_intf
         .
         fintf(1,iel)   = ...
         .
         fintf(18,iel)  = ...
         All_stress(iel) = max(1.d0,stress(1,iel)/fcti(iel))
      end do
!$OMP END PARALLEL DO

!$OMP PARALLEL DO PRIVATE (iel) REDUCTION(max:stress_max)
!$OMP+ SHARED (nel_intf,All_stress)
cdir$ ivdep
       do iel=1,nel_intf
          stress_max = max(stress_max,All_stress(iel))
       enddo
!$OMP END PARALLEL DO

       call wheneq(nel_intf,All_stress,1,razmax,index(1),nind)
       ielmax = index(nind)
       if(All_stress(ielmax).gt.1.d0) then
          plas(ielmax)= 0.d0
          nel_crack = nel_crack + 1
          fintf(1,ielmax)  = 0.d0
          .
          fintf(18,ielmax) = 0.d0
       end if
```

Box 2 Pseudo code fragment of the modified code: Subroutine for computing internal force for the interface elements

robustness of nonlinear solution process we have to limit only one interface element to "crack" at each nonlinear iteration.

The search algorithm was divided into two loops: the first evaluates the all element stresses (All_stress(iel)); the second points out the maximum stress (stress_max) and the correspondent element is pointed out with the aid of routine wheneq subroutine from Cray/F90 library. This procedure avoids critical memory references accomplishing the internal force correction outside the referred parallel loop.

Parallel performance was assessed using Atexpert tool. According to Atexpert tool this subroutine appears to be 99.8 percent parallel and 0.2 percent serial. In the evaluation of the subroutine Amdahl's Law predicts of was achieve a 11.9 times speed-up on 12 CPU's and A 10.7 speed-up is predicted with 12 CPU's on a dedicated system.

4.2. Memory access strategies in the PVP machine

This work assesses the behavior of the code in two PVP machines: a) the CRAY T94, which has

```
Subroutine Reverse Cuthill-McKee
   1. Finding a starting vertex: Determine a starting vertex r and assign r to x1.
   2. Main part: do i=1,...,N,
          find all the unnumbered neighbors of the vertex xi and number them in
           increasing order of degree.
   3. Reversing ordering: The reverse Cuthill-McKee ordering is given by
        y1,...,yN where yi is  xN-i+1 do i = 1,...,N.
```

Box 3 A pseudo code fragment of the modified code: Subroutine Reverse Cuthill-McKee

```
Subroutines Smatv-tetra and Smatv-intf
Dimension  Ap (neq),  K_tetra (nel_tetra,78),  lm(nel_tetra,12)
Dimension  Ap (neq),  K_intf  (nel_intf,171),  lm(nel_intf,18 )
Loop over elements blocks
do iblock = 1, nelblock
      nvec= ielblock (iblock)
!$OMP PARALLEL DO
!$OMP+ DEFAULT (PRIVATE)
!$OMP+ SHARED (nvect, Ap, p, K_e , lm ,)
cdir$ ivdep
   do i = 1, nvec
      grather global componets of  p to local elements
      compute and assembly elements contribution to product Ap
   end do
!$OMP END PARALLEL DO
end loop over the blocks
```

Box 4 A pseudo code fragment of the original code: Subroutines Smatv-tetra and Smatv-intf

4 CPU's and peak performance for a single CPU of 1.8 Gflop/s, b) the CRAY SV1, which has 12 CPU's and 1.2 Gflop/s per CPU's. However, memory architecture is different: CRAY SV1 has cache-memory hierarchy, while T94 has not.

In order to improve the strategies for memory access, several alternatives were tested, and the following changes have been consolidated:

To optimize memory access it was necessary to reorganize the mesh data trying to improve data locality for the EBE computations. The mesh generation for tetrahedra and interfaces is done in distinct generators. For that reason an arbitrary node numbering is formed. To overcome this problem the Reverse Cuthill-McKee algorithm RCM (1972) was used to reorder the nodal graph (implemented in modified code, show in Box 3).

A well known fact in an EBE interative driver is that the most time consuming computational kernel is the matrix-vector product. In the implementation adressed herein, these kernels are implemented by the routines **Smatv-intf** and **Smatv-tetra**, (original code, show in Box 4) respectively for the interface and tetrahedra, as suggested by Eq. (15).

In previous works (Paz, *et al.* 2002a,b and 2003) it was observed that the matrix-vector products routines are most time consuming. The strategy of the operations was changed, these routines have been modified partitioning each routine in two, as for tetrahedral as for interface elements, **Smatv1-tetra**, **Smatv1-intf** and **Assemb-tetra** and **Assemb-intf**. Shown in Box 5 and 6 to follow:

As can be observed, there occurred switching array dimension and splitting the original loops into two loops: one exclusive for the matrix-vector products and another for the assembly phase.

```
Subroutines Smatv1-tetra and Smatv1-intf
Dimension  Ape(12,nel_tetra), p(neq), Ktetra (78,nel_tetra), lm(12,nel_tetra)
Dimension  Ape(18,nel_intf ), p(neq), Kintf  (171,nel_intf), lm(18,nel_intf )
!$OMP PARALLEL DO
!$OMP+ DEFAULT (PRIVATE)
!$OMP+ SHARED (nel, p, Ape, Ke)
Loop over elements
do i = 1, nel
     grather global componets of  p to local elements
     compute and store elements contribution to product Ape
end do
!$OMP END PARALLEL DO
end loop over elements
```

Box 5 A pseudo code fragment of the modified code: Subroutines Smatv1-tetra and Smatv1-intf

```
Subroutines Assemb-tetra and Assemb-intf
Dimension  Ap (neq),  Ape(12,nel_tetra)
Dimension  Ap (neq),  Ape(18,nel_intf)
Loop over elements blocks
do iblock = 1, nelblock
     nvec= ielblock (iblock)
!$OMP PARALLEL DO
!$OMP+ DEFAULT (PRIVATE )
!$OMP+ SHARED (nvect, Ap, Ape)
cdir$ ivdep
   do i = 1, nvec
      Ap(neq1) = Ap(neq1)  + Ape (1,i)
      .          .
      Ap(neq18)= Ap(neq18) + Ape (18,i)
   end do
!$OMP END PARALLEL DO
end loop over the blocks
```

Box 6 A pseudo code fragment of the modified code: Subroutines Assemb-tetra and Assemb-intf

5. Numerical examples

In order to validate de implemented code, this work employed two cases: the first case of the uniaxial tension tests on dog-bone shaped concrete specimens by Carpinteri, *et al.* (2003); and the other case of the four-point shear test with one notch by Carpinteri, *et al.* (2003).

5.1. Uniaxial tension test: dog-bone shaped specimen

The first case (Carpinteri, *et al.* 2003. Fig. 5) dog-bone shaped geometric characteristics are with unstructured three meshes (Fig. 6a, b, c and Table 1). A uniform displacement field is applied in 30 incremental steps, at the top surface as shown in Fig. 5. Concrete with maximum aggregate diameter of 8.00 mm was used (to compute V_g the volume of coarsest aggregate Eqs. 1-2).

The curves stress-strain for meshes A, B and C show in Figs. 7(a), (b), (c) concern with Monte-Carlo simulations using 20 samples. Fig. 8 presents the comparison of numerical results; Curves

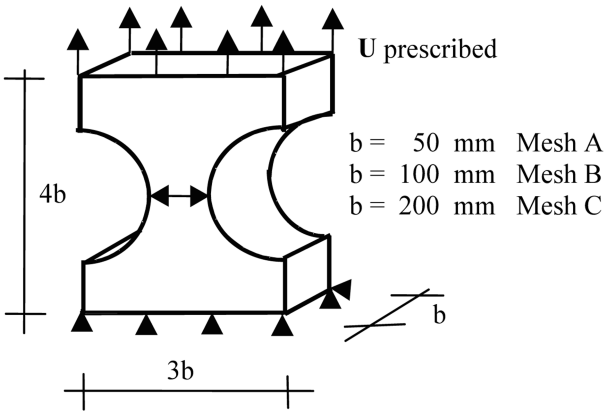


Fig. 5 Uniaxial tension test specimens, dimensions and geometry, dog-bone shaped concrete specimen by Carpinteri, *et al.* (2003)

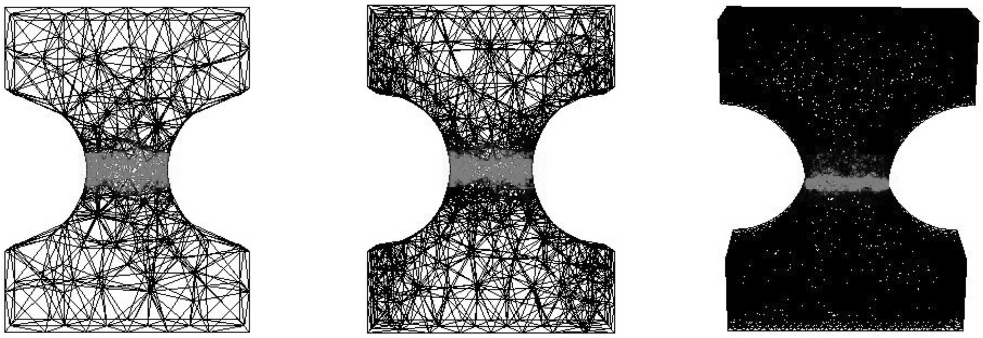


Fig. 6 Representations of the computational meshes tetrahedra (black) and interface elements (gray): (a) Mesh A, (b) Mesh B and (c) Mesh C

Table 1 Details of meshes and material

Uniaxial tension testes	Nº of total elements	Nº of tetrahedral	Nº of interface	Nº of nodes	F_{ct} MPa	E_c MPa
Mesh A	2040	1326	714	1232	4.5	42000
Mesh B	5987	4468	1519	2746	3.5	42000
Mesh C	33931	30292	3639	11815	2.5	42000

means Monte Carlo simulations and Carpinteri, *et al.* (2003): Stress vs. strain. Fig. 9 presents the crack configuration for a given sample at a stage corresponding to the softening branch of the stress-strain curve (Mesh B).

The following conclusion based on the results presented in this example: Size-scale effect experiments using dog-bone shaped concrete specimens have shown a decrease of nominal strength with increasing specimen size. This model is able to predict the size effects even in tests where the classical approach fails, e.g., the uniaxial tension test. Size effect in uniaxial tensile tests can therefore be seen as consequence of the heterogeneous microstructure of concrete. The

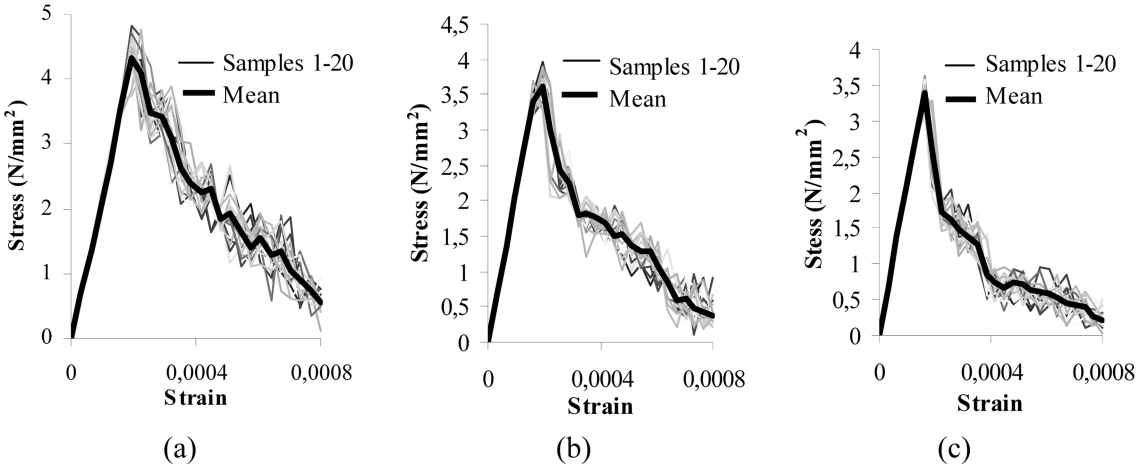


Fig. 7 Results for the complete Monte Carlo simulation for uniaxial tension testes: Stress vs. strain. (a) Mesh A, (b) Mesh B and (c) Mesh C

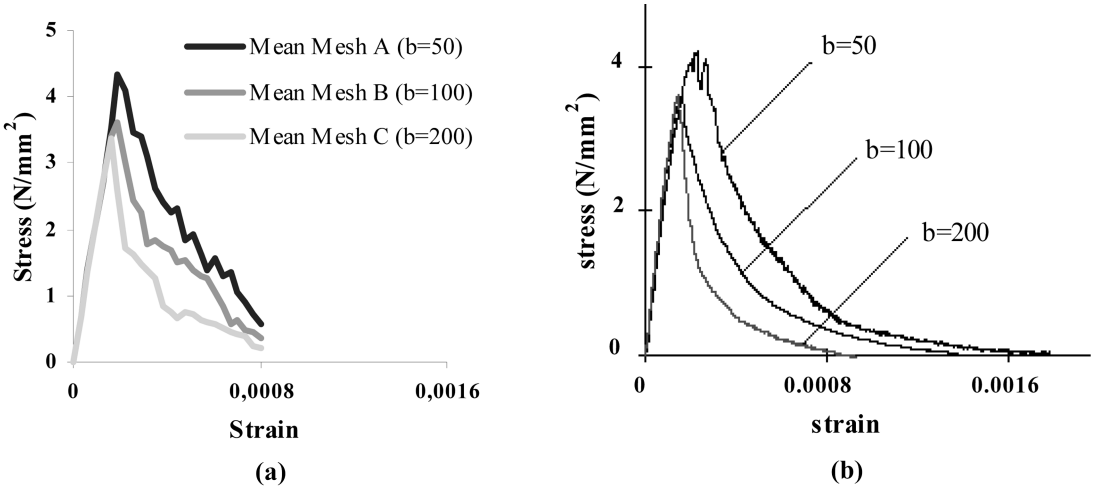
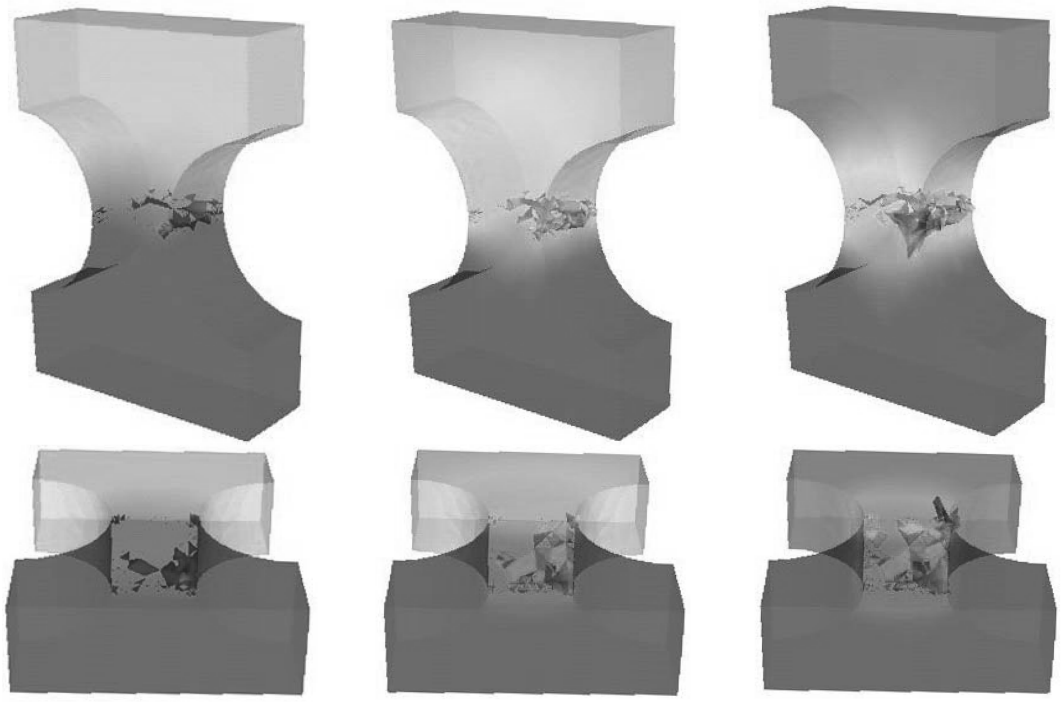


Fig. 8 Comparison of numerical results, Stress vs. strain: (a) Curves means Monte Carlo simulations and (b) Carpinteri, *et al.* (2003)

heterogeneity governs the overall cracking behavior and related size effects on concrete fractures. The determination of fracture properties of concrete by means of the uniaxial tension has been described. The uniaxial tension test is the most fundamental test to determine the fracture proprieties of a material.

5.2. Four-point shear test with one notch FSTP

In order to validate de implemented code, we employed the four-point shear test (FSTP) with one notch. The case as reported in Carpinteri, *et al.* (2003), has regular geometric characteristics as shown in Fig. 10. The Fig. 11 (Table 2) displays the unstructured computational mesh for the test.



(a) (b) (c)

Fig. 9 Crack evolution for numerical simulation-Mesh B: (a) step 10, (b) step 20 and (c) step 30

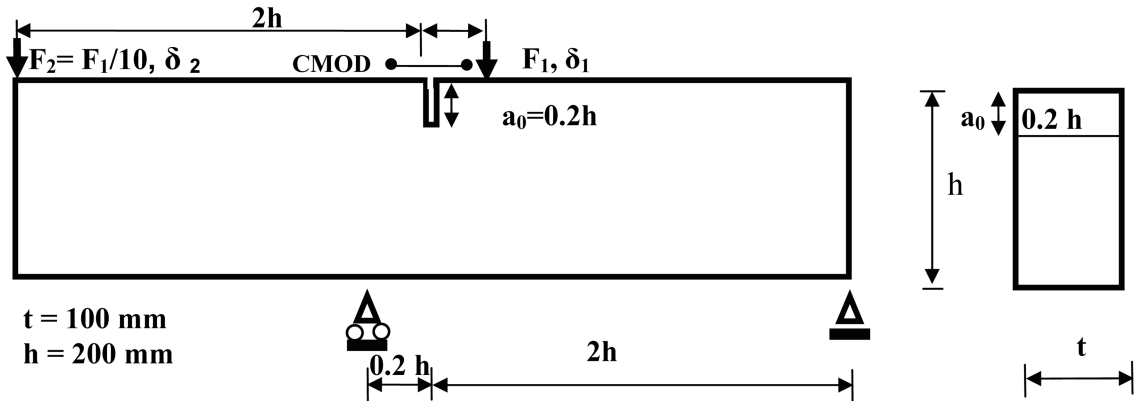


Fig. 10 Four-point shear test with one notch FSTP; specimen, geometry, dimensions: Carpinteri, *et al.* (2003)

A uniform displacement field is applied in 30 incremental steps, at the top surface as shown in Fig. 10. Concrete with maximum aggregate diameter of 8.00 mm was used (to compute V_g the volume of coarsest aggregate Eq. 1-2).

The curves load vs. CMOD (crack mouth opening displacement) for beam FPST are shown in Fig. 12(a) concern with Monte-Carlo simulations using 15 samples and Fig. 12(b) concern the results of Carpinteri, *et al.* (2003). Fig. 13 presents the crack configuration for a given sample at a stage corresponding to the softening branch of the load vs. CMOD curve

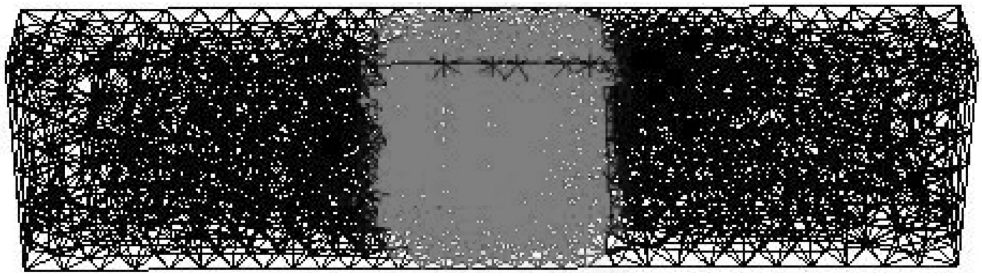


Fig. 11 Representations of the computational mesh tetrahedra (black) and interface elements (gray) for the simulation of a four-point shear test with one notch FSTP

Table 2 Details of the mesh and material

	Nº of total elements	Nº of tetrahedra	Nº of interface	Nº of nodes	f_{ct} MPa	E_c MPa
Beam FPST	25756	12373	13383	23401	2.0	40000

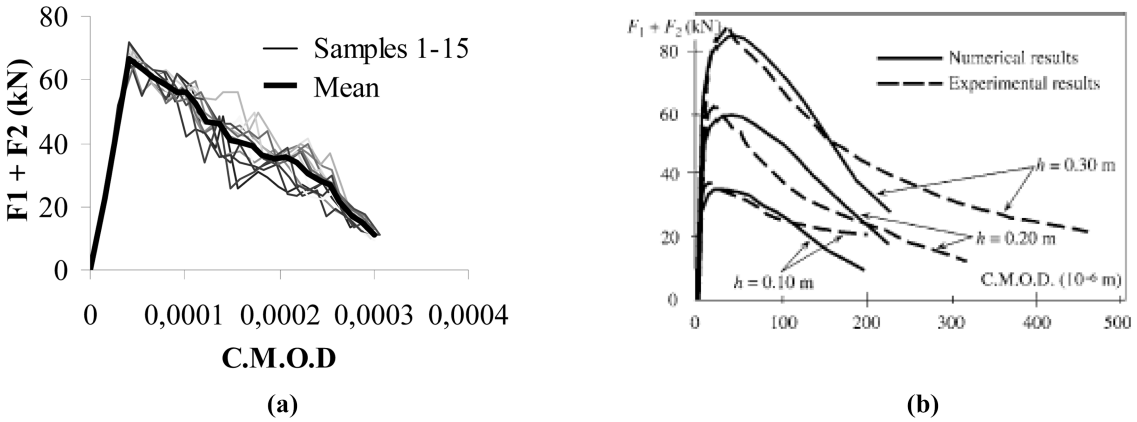


Fig. 12 (a) Results for the complete Monte Carlo simulation. load vs. CMOD (h = 0.20). (b) Results Capinteri, *et al.* (2003) load vs. CMOD

This work presents a large FPST example which was only possible to analyze with the modified version of the original code as described herein. Due to the complexity of the example the four-point shear test with one notch that involves large number of interface elements, increasing non-linearity behavior of the problem.

6. Computational performance

A comparative code performance was carried out for the two implementations (original and modified version of the code), both in the CRAY SV1 and CRAY T94. The computational performance was evaluated through the uniaxial tension test problem using the dog-bone shaped

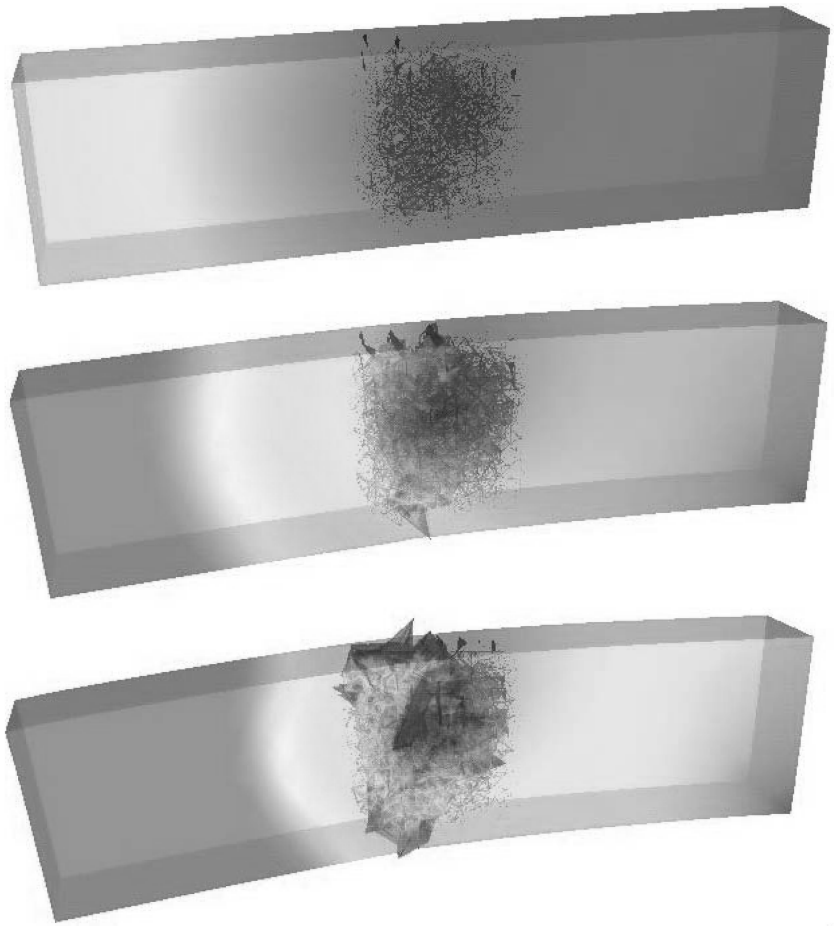


Fig. 13 Crack evolution for numerical simulation

model with mesh C, as described in sec. 5.1.

Fig. 14 displays results for a single CPU run of the original and modified code on the CRAY SV1, showing (Fig. 14a) the observed floating point operations rate (Mflop/s) for the two most computationally intensive subroutines (**Smatv1-intf** and **Smatv1-tetra**) and for 1 sample with 30 steps the whole job, in Fig. 14b showing for subroutines the average time in seconds per call, in Fig 14(c) displays CPU time whole job. As expected, the most intensive subroutines implement the EBE matrix-vector products for interface elements and tetrahedra. In conjunction they are responsible for over 90% of the CPU time in the original code implementation.

Since the modified code implements the assembly and matrix-vector product in separate subroutines, the values presented on Fig. 14(d) have to be added to the performance values obtained for assembly subroutines namely **Assem-intf** and **Assem-tetra** respectively so as to evaluate the speed-up over original code. It was observed an overall speed-up by a factor of 1.57-1.58 (respectively for the interface and tetrahedral) for the EBE computations. Overall code performance results can be seen in Fig. 14(d), where Mflop/s and CPU time for the whole job are displayed. The modified code achieved an overall speed-up of 1.64.

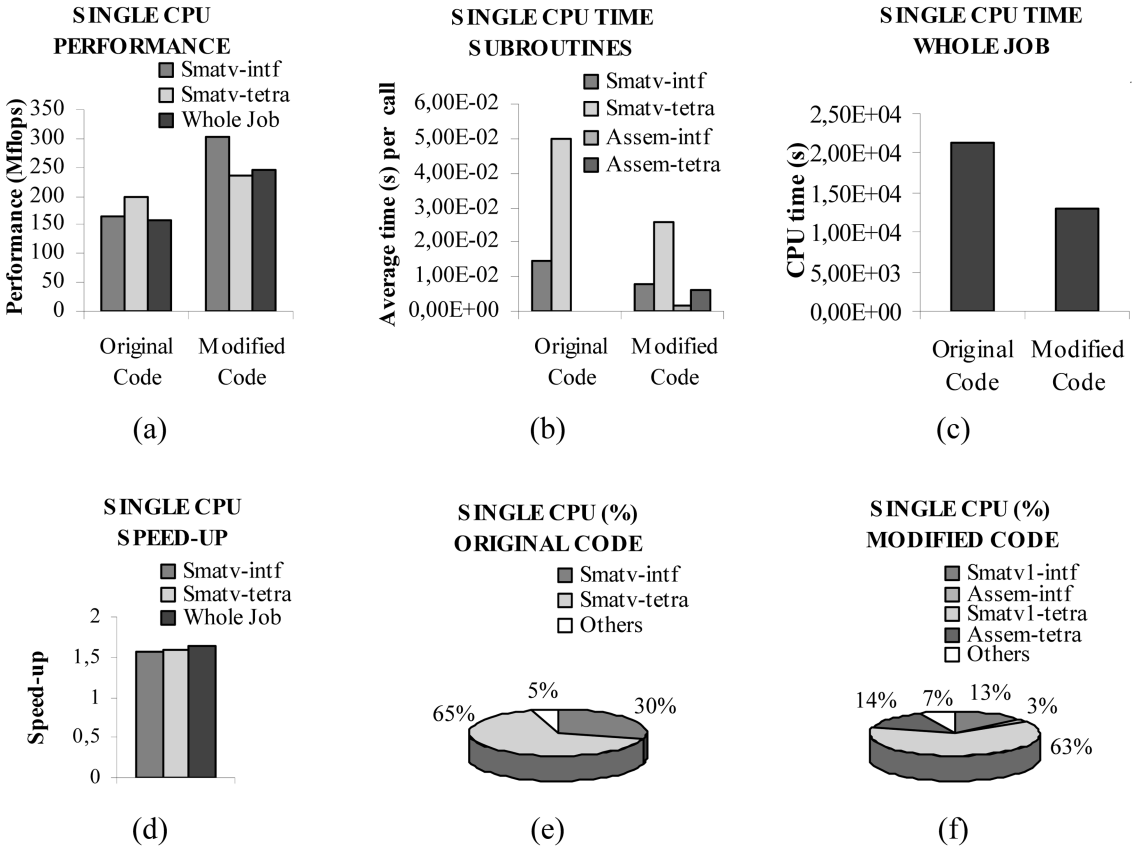


Fig. 14 Performance analysis in single CPU for CRAY SV1-Uniaxial tension test mesh C, for the original and modified code. (a) The top subroutines and for 1 sample with 30 steps-whole job, (b) Average time in seconds per call, (c) CPU time whole job, (d) Speed-up, (e) and (f) Single CPU (%) to the whole job

Figs. 14(e), (f) present results for a single CPU the fraction of CPU time (%), relative to the whole job.

The same tests were carried out in the CRAY T94 Figs. 15(a)-(f) display equivalent results as the previous Figs. 14(a)-(f) this time for the CRAY T94 platform. The performance results for the modified code are equivalent to the results obtained in the CRAY SV1. It was observed an overall speed-up by a factor of 1.69-1.67 (respectively for the interface and tetrahedra) for the EBE computations. Overall code performance can be seen in Fig. 15(d), where Mflop/s and CPU time for the whole job are displayed. The modified code achieved an overall 1.65 fold speed-up.

Figs. 15(e), (f) present results for a single CPU the fraction of CPU time (%), relative to the whole job.

As observed in the experiments, the code modifications provided significant speed-up in a single CPU (~1.7 on average for the whole job). It should be mentioned that the original code was already developed for PVP machines.

Next, we present the performance analysis for the four-point shear test with one notch (FPST) test problem. Due to the complexity of the example that involves large number of interface elements,

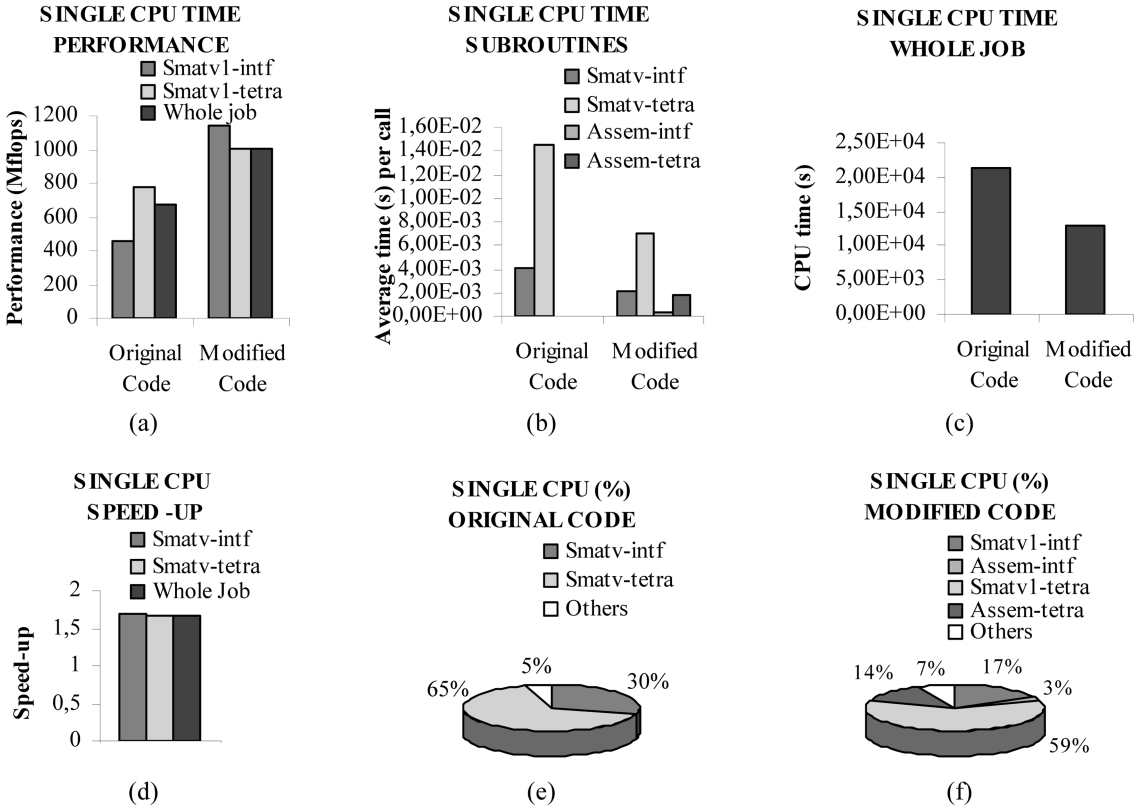


Fig. 15 Performance analysis in single CPU for CRAY T94- Uniaxial tension test mesh C, for the original and modified code. (a) The top subroutines and for 1 sample with 30 steps-whole job, (b) Average time in seconds per call, (c) CPU time whole job, (d) Speed-up, (e) and (f) Single CPU (%) to the whole job

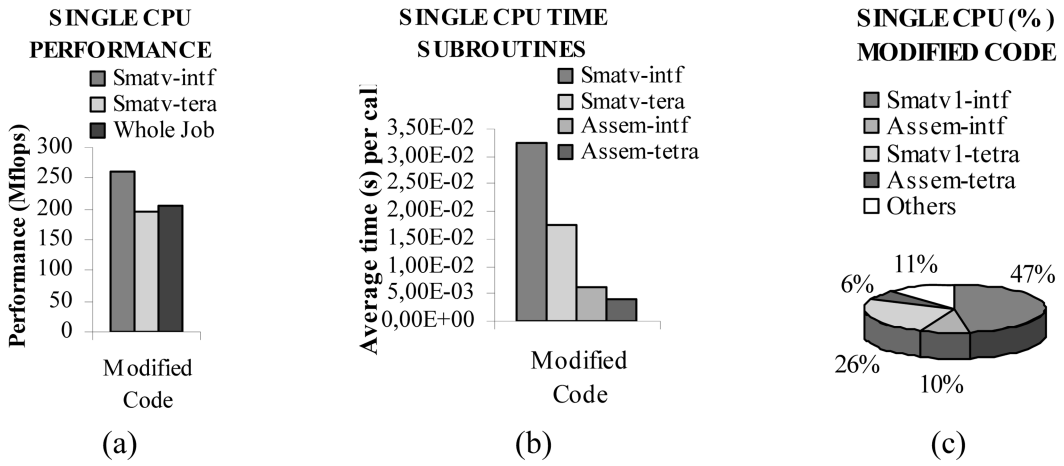


Fig. 16 Performance analysis in single CPU for CRAY-SV1 (modified code). Four-point shear test with one notch FPST. (a) the top subroutines and 1 sample with 30 steps whole job (b) Average time (s) per call, (c) CPU time (s) whole job and (d) Single CPU (%) to the whole job

Table 3 Summary of the ATEXPERT's Report for the 4 dominant loops CRAY SV1

ROUTINES	% Parallel	Amdahl's law Speed-up	Predicted Speed-up
Smatv1-intf	99.8	11.5	11.8
Smatv1-tetra	97.8	10.5	10.7
Assemb-intf	99.8	11.4	11.8
Assemb-tetra	100.0	11.4	12.0

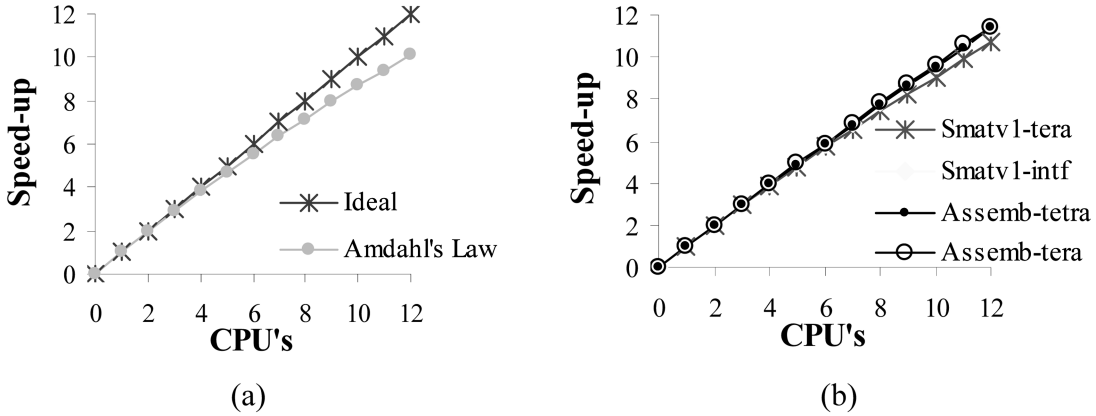


Fig. 17 Parallel speed-up charts according to Atexpert tool: (a) Whole job (b) The top 4 subroutines

increasing non-linearity behavior of the problem, it was not possible to obtain the performance results for a single CPU for the original version of the code. The job time with original code exceed a practical limit on CRAY SV1 in single CPU. This work presents a large example which was only possible to analyze with the modified version. Figs. 16(a)-(d) display equivalent results as the previous Figs. 14(a), (b), (c), (f), these results for the CRAY SV1 platform. The CPU time of the vectorized single processor run for CRAY SV1 is 2.32E+4 seconds for the whole job.

Parallel performance was assessed using CRAY's Atexpert tool. The results for a test run for four-point shear test with one notch (FPST) are shown in Table 3 and Fig. 17, for the top four routines. Results displayed consider the amount of parallelism (%), the corresponding Speed-up predicted by Amdahl's and the expected speed-up on a dedicated run on a 12 CPU system, as predicted by Atexpert tool.

According to Atexpert tool this program appears to be 99.0 percent parallel and 1.0 percent serial. Amdahl's Law predicts the program could expect to achieve a 10.8 times speed-up on 12 CPU's and a 10.1 speed-up is predicted with 12 CPU's on dedicated system. Parallelism was not affected by these modifications and predicted parallel speed-ups are encouraging.

The probabilistic methodology study in this paper corresponds to the 3D analysis of a strongly nonlinear material that develops cracks. In addition, the finite elements analysis must be called several times within a Monte Carlo simulation. The modified version of the code was optimized in such a way that the simulation time did not exceed a practical limit.

7. Conclusions

This work presented the key issues regarding the implementation and optimization of a computational code for three-dimensional analysis of probabilistic discrete cracking in concrete. Basic guidelines for the implementation considered as target computational platforms shared memory vector processors (PVP's) such as Cray SV1 and Cray T94. Optimization techniques included algorithm and data structure reorganization in order to allow the analysis of large scale models. Two test problems were used to validate the model and access the implementation performance. In both cases, the obtained results showed good agreement with results available in literature. As observed in the experiments, the introduced code modifications provided significant advance in more complex models, achieving good vector-parallel performance. As an attempt to enhance data localization, a critical aspect in systems with memory hierarchy, data reorganization was enforced in the implementation through a graph reordering technique (i.e. Reverse Cuthill-McKee). However, performance bottlenecks related to data access in architectures with memory hierarchy were still observed and techniques for circumventing this issue are currently under further investigation.

Acknowledgements

Computer time for the CRAY SV1 was provided by High Performance Computing Center NACAD/COPPE/UFRJ. Graphic resources were provided by Laboratory of Computational Methods in Engineering of the Program of Civil Engineering LAMCE/COPPE/UFRJ. The CESUP/UFRGS is gratefully acknowledged for the computer time provided in the CRAY T94. The authors are also indebted to Dr Nick Chepurniy formerly manager and support from CRAY.

References

Bazant, Z. P. (1976), "Instability, ductility, and size effects in strain-softening concrete", *J. Eng. Mech. Div.*, *ASCE*, **102**(EM2): 331-44; Disc. 103:357-8, 104:505-2.

Bazant, Z. P. (1997), "Scaling of quasi-brittle fracture: hypoteses of invasive and lacunar fractality, their critique and Weilbull connection", *Int. J. Fract.,* **83**(1), 41-65.

Bazant, Z. P. (2002), "Concrete fracture models: testing and practice", *Eng. Fract. Mech.,* **69,** 165-202.

Hillerborg, A., Modéer, M. and Petersson, P. E. (1976), "Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements", *Cement Concr. Res.,* **6**, 773-82.

Carpinteri, A., Chiaia, B. and Ferro, G. (1994) "Multifractal scaling law for the nominal strength variation of concrete structures", *Size Effects in Concrete Structures,* Mihashi, M., Okamura, H, Bazant, ZP., editors. London E & FN Spon 193-206.

Carpinteri, A., Cornetti, P., Barpi, F. and Valente, S. (2003), "Cohesive crack model description of ductile to brittle size-scale transition: dimensional analysis vs. renormalization group theory", *Eng. Fract. Mech.,* **70**, 1809-1839.

Cuthill, E. (1972), "Several strategies for reducing the bandwidth of matrices", *Sparse Matrices and Their Applications*, D. J. Rose and R. A. Willoughby (Eds.), Plenum Press, New York.

Elices, M. and Planas, J. (1989), "Material models", *Fracture Mechanics of Concrete Structures*, Elfegren L. editor, London: Chapman and Hall 16-66 chaper 3.

Fairbairn, E. M. R., Paz, C. N. M., Ebecken, N. F. F. and Ulm, F-J. (1999), "Use of neural networks for fitting

of FE probabilistic scaling model parameters", *Int. J. Fract.,* **95**, 315-324.
Guineas, G. V. Elices, M. and Planas, J. (1994), "A general bilinear fit for the softening curve of concrete", *Mater. Struct*., **27** 99-105.
Hu, X. Z. and Wittmann, F. H. (1991), "An analytical method to determine the bridging stress transferred within the fracture process zone", *Cement Concrete,* **21**, 1118-28.
Hughes, T. J. R. (1987), "Algorithms for parabolic problems, element-by-element (EBE) implicit methods", *The Finite Element Method Analysis*, Chapter 8, 483-489, New Jersey: Prentice-Hall.
Kelley, C. T. (1995), "Iterative methods for linear and nonlinear equations, frontiers in applied mathematics", SIAM, *Society for Industrial and Applied Mathematics*, Philadelphia.
Mihashi, H., Nomura, N., Izumi, M. and Wittmann, F. H. (1991) "Size dependence of fracture energy of concrete", *Fracture Process in Concrete Rocks and Ceramics*, van Mier, Rost, Bakker, editors. 441-50.
Paz, C. N. M., Martha, L. F., Alves, J. L. D., Fairbairn, E. M. R., Ebecken, N. F. F. and Coutinho, A. L. G. A. (2002a), "Parallel implementation and development of a probabilistic model for 3D discrete cracking concrete", *Proceedings of the Fifth World Congress on Computational Mechanics WCCM V*, July 7-12, Viena Austria Eds: H.A. Mang, F.G. Rammerstorfer, J Eberhardsteiner, J., Publisher: Viena University of tecnologi, Austria, ISBN 3-9501554-0-6, http:wccm.tuwien.ac.at.
Paz, C. N. M., Martha, L. F., Alves, J. L. D., Fairbairn, E. M. R., Ebecken, N. F. F. and Coutinho, A. L. G. A. (2002b), "3D simulation of concrete cracking: probabilistic formulation in parallel environment". *The Sixth International Conference on Computational Structures Technology*, Prague-Czech Republic: B. H. V. Topping and Bittnar, **1**, 1-19.
Paz, C. N. M., Martha, L. F., Alves, J. L. D., Fairbairn, E. M. R., Ebecken, N. F. F. and Coutinho, A. L. G. A. (2003), "Parallel implementation for probabilistic analysis of 3D discrete cracking concrete", *Lecture Notes in Computer Science,* ISBN 3-5540-00852-7, **2565,** 79-93.
Petersson, P. E. (1981), "Crack growth an developedment of fracture zones in plain concrete and similar materials", Report TVBM-1006 Division of Building Materials, Lund Institute of Tecnology, Lund, Sweden.
Planas, J., Elices, M., Mazars, J. and Bazant, Z. P. (1989), *Editors Craking and Damage.* London: Elsevier, 462-76.
Press, W. H., Teukolski, S., Vetterling, W. T. and Flannery, B. (1992), *Numerical Recipes*, Cambridge University Press.
Prado, E. P. and Vnd Mier, J. G. M. (2003), "Effect of particle structure on model I fracture process in concrete", *Eng. Fracture Mech.,* **70**, 1793-1807.
Rossi, P. and Richer, S. (1987), "Numerical modeling of concrete cracking based on a stochastic approach", *Mater. and Struct., RILEM,* **20**, 334-337.
Rossi, P., Wu, X., Maou, Fle and Belloc, A. S. (1994), "Scale effect on concrete in tension", *Mater. Struct., RILEM*, **27**, 437-444.
Rossi, P., Ulm, F.-J. and Hachi, F. (1996), "Compressive behavior of concrete: physical mechanisms and modeling", *J. Eng. Mech.*, *ASCE*, **122**(11), 1038-1043.
Rossi, P. and Ulm, F.-J. (1997), "Size effects in the biaxial behavior of concrete: physical mechanisms and modeling", *Mat. Struct., RILEM*, **30,** 210-216.
Schlangen, E. and Van Mier, J. G. M. (1992), "Micromechanical analysis of fracture of concrete", *Int. J. Damage Mech.*, **1**(4), 435-454.
Tschegg, E., Kreuzer, H. and Zelenzny M. "Fracture in concrete under biaxial loadings-numerical evaluation of wedge-splitting results", *Fracture of Mechanics of Concrete Structures*, Bazant ZP, editors. London: Elsevier 455-60.
Van Mier, J. G. M. and Van Vliet, M. R. A. (2003), "Influence of microstructure of concrete on size/scale efects in tensile fracture", *Eng. Fract. Mech.,* **70**, 2281-2306.
Weibull, W. (1939), "A statical theory of the strength of materials", *Proc Royal Swedish Intitute of Engineering Research*, **151**.

CC